PREDICTIVE BIG DATA ANALYTICS ON HIGH-DIMENSIONAL DATA

KYI LAI LAI KHINE

UNIVERSITY OF COMPUTER STUDIES, YANGON

October, 2019

Predictive Big Data Analytics on High-Dimensional Data

Kyi Lai Lai Khine

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

October, 2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

…..……………………………   ……………........…………………………

Date   Kyi Lai Lai Khine

ACKNOWLEDGEMENTS

First of all, I would like to thank the Minister, the Ministry of Education for full support during the Ph.D. course at the University of Computer Studies, Yangon.

I would like to express very special thanks to Dr. Mie Mie Thet Thwin, Rector, the University of Computer Studies, Yangon, for allowing me to develop this thesis and giving me general guidance during the period of my study.

I would like to acknowledge and special thanks to Dr. Kyaw May Oo, Professor, for her valuable advices and suggestions that help to improve this thesis.

I would also like to express my respectful gratitude to Dr. Khine Moe Nwe, Professor, and Course-coordinator of the Ph.D. 9th Batch, the University of Computer Studies, Yangon, for her excellent guidance, caring, and providing me during the Ph.D. study.

I would like to express my deepest gratitude to my supervisor, Dr. Thi Thi Soe Nyunt, Professor, the University of Computer Studies, Yangon, for her patient supervision, tenderness, encouragement and providing me with excellent ideas and guidance throughout the study of this thesis. I will always remember her for being a mentor to me.

I am very grateful to Daw Aye Aye Khine, Associate Professor, Head of English Department, for her valuable support and editing my thesis from the language point of view.

In addition, I would like to extend my appreciation and thanks to the board of examiners for valuable comments and corrections to my thesis.

I also thank all of my friends from the Ph.D.9th Batch for their co-operation and encouragement during the Ph.D. student life.

Last but not least, I am deeply indebted to my family for always believing in me, for their endless love, support and encouragement. They are always supportive of me during my period of studies, especially for this Doctorate Course. Finally, I am extremely grateful to my lovely mother, Daw Myint Myint Khine, for her unconditional love and care which mean everything to me.

ABSTRACT

Nowadays, data is extremely growing very fast to become “BIG DATA”, any voluminous amount of structured, semi-structured and unstructured data, which has high potential to be mined for valuable information in decision making process. Analyzing on big data using traditional data analysis methods has become the key challenge in data analytics research. In addition, high-dimensional data analytics has been a great attention in big data era because the dimensions of datasets are continuously growing in size. It creates a critical issue to reduce efficiently a subset of dimensions from all diverse and raw data dimensions which will fulfill valuable information in decision making process. With increasing volumes of data, classical dimensionality reduction algorithms which are designed to work well with small-scale data usually face scalability bottleneck. Although Principal Component Analysis (PCA) could be applied as a dimensionality reduction algorithm in high-dimensional data, it is absolutely required to transform as scalable PCA (sPCA) for high-dimensional big data.

With the purpose of constructing efficient prediction model, Multiple Linear Regression (MLR), the redundant and irrelevant features or data dimensions are highly potential to increase noises and biases which can hinder the prediction process of the model. In this research, two-stage dimension reduction approach is proposed for the MLR model. Firstly, scalable Principal Component Analysis (sPCA) is proposed to solve the storage and computational problems of PCA by reducing the number of redundant dimensions without much loss of information. To examine the reduced feature subset resulted from sPCA stage whether correlated or not with the output variable of MLR model, Pearson Correlation Coefficient (PCC) is also applied to reduce the number of irrelevant dimensions. Although the high dimensions of input voluminous data matrix have been reduced, it is still a big issue to solve how to split or decompose this voluminous matrix containing large amount of observations or data records. Therefore, “QR Decomposition” is proposed to decompose large-scale matrix X into a Q and R product of an orthogonal matrix Q and an upper triangular matrix R for MLR model.

In this research, the high-dimensional data reduction providing predictive big data analytics is implemented on distributed big data analytics platform, “Cloudera

Distribution Hadoop (CDH)” using Multi Node Cloudera Cluster using three computing nodes or VMs which all are interconnected with Cloudera Manager. Three diverse high-dimensional big data sources are applied not only evaluating the proposed approaches but also achieving predictive analysis results from the system. Firstly, geospatial big data, OpenStreetMap in XML format (OSM XML) is used to obtain “One-way Roads” prediction. Then, high-resolution or high-dimensional representation of images from MS-Celeb-A, a large-scale face attributes dataset are utilized to predict “Number of Faces” in these images. Finally, the raw, unstructured text data via “DeliciousMIL” dataset from UCI is applied as input text documents to obtain “Number of Documents (Education, Science && Technology, Culture && History)” prediction results.

According to the evaluation analysis, the proposed sPCA can efficiently perform dimension reduction process with increasing size or number of data dimensions for diverse data types. It also shows the good scalability performance while the traditional PCA offers “Out of Memory” results. Applying the proposed two-stage approach (sPCA and PCC) achieves the victory of accuracy in 99 percent (%) for “One-way Roads” prediction. Furthermore, QR Decomposition approach providing MLR model offers faster execution time for the system. Therefore, the proposed system provides better scalability, prediction accuracy, and faster execution time in predictive analytics on high-dimensional big data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

Cloud computing is a style of computing in which dynamically scalable, virtualized resources are provided as a service over the Internet. The rise of cloud computing and cloud data stores have been a precursor and facilitator to the emergence of “Big Data”. Big data is an evolving term that describes any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information. There are many different definitions about big data. It can be defined a collection of data with enormous volume which cannot be managed by traditional database system. Big data consists of not only tradition data types but also other types such as text, audio, and video. Thus, the format of big data can be assumed as irregular structure. According to the literatures, there are typically four areas in big data such as infrastructure, management, analysis, and decision support. For infrastructure, storing and management technologies of big data may beyond traditional data structure and database technologies.

There has been emerged as “Scientific Discovery of Data” by using the voluminous amount of data in which very meaningful and useful values are hidden to be extracted in systematic analysis. In other words, it can be known as “Big Data Analytics”. It can also be defined as the combination of traditional data analytics and data mining techniques together with a large volume of data creating a fundamental platform to analyze, model and predict the behavior of customers, markets, products, services and the competition and so on. It is the process of examining big data to uncover hidden patterns, unknown correlations and other useful information that can be used to make better decision. Moreover, the process of big data analysis can actually provide taking intelligent decisions for organizations. Many organizations used to apply predictive big data analytics in different ways ranging from predictive marketing and data mining algorithms to machine learning and artificial intelligence algorithms to explore new statistical patterns for business processes.

In general, predictive analytics can be expressed as recognizing the patterns in data to project probability. The core approach is regression analysis in predicting the related values of multiple and correlated variables which is based on proving a particular assumption. Predictive data analytics typically replies the question: “What could happen in future?" applying statistical models like regression and forecasts to understand the future. It comprises a variety of techniques that can predict future outcomes based on historical and current data. In addition, a number of limitations have emerged from managing and processing big data according to its increasing data volume, unstructured and complex structure or pattern, increasing data speed etc. Thus, extracting valuable information from extremely large collection of data has become the most important and complex challenges in data analysis research. The challenges of limiting memory usage, computational hurdles and slower response time are also the main contributing factors to consider traditional data analysis on big data. Traditional data analysis methods absolutely need to adapt in high-performance analytical systems running on distributed environment which provide scalability and flexibility.

With the rapid development in technologies, increasing in computational power and decreasing in the cost of data collection and processing, the dimension of datasets is continuously growing in size. In these datasets, the dimensions or feature variables "n" can be as high as in size or much higher than the observation size "m". The dimension of the data is the number of variables that are measured on each observation among thousands of dimensions or feature variables, only a small number or subset of them are possible to extract value or insight in data analysis. Therefore, it makes a critical situation to identify correctly and to reduce efficiently them. And, finding interesting features in datasets can fulfill valuable information to support decision making. High-dimensional data analytics has been a great attention in big data era. The complexity of big data often makes dimension reduction techniques necessary before conducting statistical inference.

The main purpose of dimension reduction is to find out how many dimensions can be reduced from all diverse and raw data dimensions. In fact, a single observation has dimension in number of the thousands or millions or billions. Besides, classical data analysis methods are not designed to cope with this kind of growth of dimensionality of the observation vector. In general, statistics is a good analytical tool for big data to be analyzed. Regression is a statistical empirical technique which is widely used in business, the social and behavioral sciences, the biological sciences, climate prediction, and many other areas. Multiple linear regression is an approach to find out the relationship between a dependent variable Y and one or more independent variables Xs. In regression, variables or predictors are used to construct a regression model in predicting a response for future. Thus, selecting the relevant predictors or dropping irrelevant predictors for the model is absolutely essential.

In the proposed system, Multiple Linear Regression (MLR) model based on "QR Decomposition" and "Ordinary Least Squares" method is proposed in decomposing big matrix data to extract MLR model's coefficient "β". It is suitable for parallel and distributed processing with the purpose of predictive analytics on massive datasets. The implementation of a parallel and distributed version for multiple linear regression with QR Decomposition enables to extract its coefficients with massive data processing. Then, the central idea of Principal Component Analysis (PCA) is to reduce the dimensionality of a dataset consisting of a large number of interrelated variables without losing information and "PCA based MLR analysis" is also applied removing redundant variables. Therefore, the most important predictors or variables are chosen with the purpose of not including irrelevant and redundant variables for the proposed regression model.

In selecting the most important predictors or features for the proposed regression model, two-stage dimensional reduction approach is proposed by applying Principal Component Analysis (PCA) and correlation-based measure, Pearson Correlation Coefficient (PCC) for selecting the most important and correlated variables or features to construct a regression model that predicts well when dealing with high-dimensional big data. The main objective of this approach is to improve the predictive power of statistical model, especially multiple linear regression model applying two-stage dimension reduction approach.

1.1 Problem Definition

The world today is built on the foundation of data. The rise of cloud computing technology and cloud-based data storage have been a precursor and facilitator to the development of big data and big data analytics. Big data is revolutionizing traditional operations and analysis everywhere such as researches in artificial intelligence, finance, biology, genetics, engineering and astronomy and so on. Storing huge amount of data available in various formats which is increasing with high speed to gain value out it is a great opportunity to emerge big data. In recent studies, many researchers have discovered the importance of big data and the vitality of big data analysis because data is growing in unpredicted volume and complexity. With the rapid development in technologies, increasing in computational power and decreasing in the cost of data collection and processing, the dimension of datasets is continuously growing in size.

High-dimensional datasets have increased the complexity of data analysis and it requires the sophisticated techniques to process these datasets. In general, high-dimensional data means a single observation which has dimensions in the thousands or millions or billions while only tens or hundreds of observations for study. As the number of dimensions of data increases, it becomes more and more difficult to process it. Classical algorithms and methods are not designed to sort out with this growth of dimensionality and the two main impacts are arising:

- Impact on Computing Infrastructure: Massive sample size of BIG DATA makes challenges upon the traditional computing infrastructure.
- Impact on Computing Methods: It is often computationally infeasible to directly make inferences based on the raw data.

The challenges of limiting storage size, computational difficulties and slower response time are the main consideration factors to decide traditional data analysis on huge collection of wide (dimensions or features) and tall (samples or observations or records) data nature. High-dimensional big datasets have experienced many issues and problems in extracting useful insights from it. Therefore, dimension reduction scheme is absolutely required to facilitate high dimensions of big data. With the volume and variety of data increasing, addressing high-dimensionality in big datasets has become very important in building effective statistical models. Moreover, classical statistical, data mining and machine learning algorithms often encounter scalability bottleneck when they are applied to big data. Such algorithms were designed to work with small-scale data that is assumed to fit in the primary memory of a single machine.

Classical dimension reduction algorithms are absolutely needed to transform distributed and scalable version running on distributed platforms, and classical PCA approaches cannot be applied to big data because of memory and computational barriers. Distributed machine learning algorithms introduce a new set of challenges. During the distributed execution of a machine learning algorithm, processing nodes may need to exchange data among each other, which means intermediate data can slow down the distributed execution of PCA algorithm. If not carefully managed, it may actually become the main bottleneck for scaling machine learning algorithms, regardless of the available number of computing nodes.

In the infrastructure and management for big data, one of the important issues in predictive big data analytics is how to apply statistical analysis like regression to the massive amount of data at once. The independent variables for regression analysis can be assumed as predictors or features for dependent variable to make predictions. The regression model can face difficulties when multicollinearity is present among the independent variables. “Multicollinearity" refers to variables that are correlated with other variables not only dependent variable but also other independent variables. It increases the standard errors of the coefficient “β”, the estimation coefficient of the regression model. Furthermore, a large number of variables in regression analysis can cause severe problems in estimation and prediction process.

Regression analysis may not be straightforward for massive datasets. Multiple linear regression, a classical statistical data analysis approach, also proves unsuitable to the distributed environment with the increasing volumes of data. Adaptation of classical data analysis algorithms generally and predictive algorithms specifically for multiple linear regression provide a response to the phenomenon of big data. Although there has been a solution to reduce the dimensionality of input large-scale matrix, it still remains a big problem or issue to solve how to split or decompose the voluminous matrix containing increasing size of observations or data records in computing the regression model coefficient “β”. In resolving the values of “β”, it can be hardly possible to process the entire huge input matrix at once. Matrix decomposition for the regression model is needed to overcome the limitations of multiple linear regression in huge amount of data. Besides, the computation of the coefficients “β_i[]” of multiple linear regression using QR Decomposition on distributed environments facilitates parallel and distributed processing of regression model in efficient manner.

The notable issues or problems which are related to the research work are:

- Issue 1. “Curse of Dimensionality” in High-Dimensional big data
- Issue 2. Scalability in traditional dimension reduction algorithms especially Principal Component Analysis (PCA)
- Issue 3. Intermediate data bottleneck and large-scale matrix multiplication in distributed processing of traditional PCA

- Issue 4. “Multicollinearity” problem and “Tall & Skinny” matrix form requirement in Multiple Linear Regression (MLR) model
- Issue 5. Regression analysis on big data and computing regression model’s coefficient “β”

1.2 Terminology

In order to achieve predictive analysis results from the proposed system which is implemented on distributed platform, the general terminologies are required:

Principal Component Analysis (PCA): Principal Component Analysis (PCA) is used to reduce the high dimensions of a dataset consisting of a large number of irrelevant and redundant variables [35, 36]. The eigenvectors with the highest eigenvalues are the “principal components” of original data. When a matrix Y composed of N ×D (N rows and D columns), PCA obtains d principal components (d ≤ D) that explain the most variance of that data in matrix Y [15].

Multiple Linear Regression (MLR): Multiple Linear Regression (MLR) is a statistical model used to describe a linear relationship between a dependent variable “Y” and one or more explanatory variables (or independent variables) “X”, and it is widely used for prediction and forecasting [1]. A primary goal of a regression model is to estimate the relationship between the variables or equivalently, to estimate the unknown parameter “β”. The estimates for “β” values are calculated with the purpose of minimizing the sum of squares of differences between the predicted values and observed values [17, 18].

QR Decomposition: QR Decomposition or QR Factorization is one of the common decomposition approaches for large-scale matrices to solve the problem of ordinary least squares problem. It is one of the processing steps for the resolution of multiple linear regression model in massive data processing [1]. To facilitate parallel and distributed processing of multiple linear regression in efficient manner, the values of coefficient “β” is computed by QR decomposition. The given matrix A is decomposed into a product A= QR of an orthogonal matrix Q and an upper triangular matrix R.

Pearson Correlation Coefficient (PCC): For a pair of variables (X, Y), the linear correlation coefficient “r” is used to measure the strength and the direction of a linear relationship between two variables. PCC is utilized to choose the most important and relevant variables for MLR model. The value of r is such that $-1 < r < +1$. If X and Y have a strong positive linear correlation, r is close to +1 is assumed as "Positive Correlation". If X and Y have a strong negative linear correlation, r is close to -1 is assumed as "Negative Correlation". If there is no linear correlation, r is 0 or a weak linear correlation, r is close to 0 is assumed as "No Correlation".

Distributed Platform: Apache Spark, a distributed computing framework, is applied on large-scale high-dimensional datasets. The programming model of Apache Spark is typically based on processing resilient distributed datasets (RDDs). It offers the ability to easily split the computation tasks that can be normally operated in a single thread way over multiple cluster nodes [93]. To achieve scalability benefit, data should be placed in distributed file system called Hadoop Distributed File System (HDFS).

1.3 Motivation of the Thesis

Data is extremely growing very fast to emerge "BIG DATA" and analyzing on big data to extract valuable insight has become important and complex challenge in data analytics research. The voluminous amount of data (Big Data) to be processed absolutely requires the use of high-performance analytical systems running on distributed environments which provide scalability and flexibility. Moreover, it has also become a challenge to statistical community because many statistics are difficult to compute by traditional algorithms when the input dataset contains millions of training samples with a large number of attributes. Thus, successful inference of useful information from increasing volume of data makes the computational complexity for traditional statistical, data mining and machine learning algorithms. In addition, the transition to these algorithms in distributed environment is hardly possible to implement. According to unpredictable volume of data with a variety of formats, addressing high-dimensionality is essential in building effective statistical models. Handling big data from both statistical and computational perspectives, the idea of dimension reduction is an important step before start processing of big data. Classical dimension reduction machine learning algorithms, however, are needed to transform distributed and scalable version running on distributed platform. Several optimizations are crucial for scaling various dimension reduction algorithms in distributed settings. Multiple linear regression, a classical statistical data analysis model, applied in predicting future events also proves unsuitable to large-scale high-dimensional data. Therefore, adaptation of classical prediction models specifically multiple linear regression model fulfills a response to the phenomenon of big data.

1.4 Objectives of the Thesis

The main objective of the thesis is to develop a scalable and efficient predictive analytics system applying on high-dimensional big data. The other objectives are as follows:

- To apply dimension reduction approach as an essential data pre-processing step for the system
- To show the proposed scalable PCA (sPCA) can overcome the computational and scalability problems of traditional PCA in high-dimensional big data
- To show the combined approach of sPCA and PCC can efficiently perform dimension reduction process on diverse data sources improving the accuracy
- To prove "QR Decomposition" can efficiently minimize the computational burden of traditional MLR model

1.5 Contributions of the Thesis

This thesis mainly proposes the efficient and effective dimension reduction and decomposition approach for the Multiple Linear Regression (MLR) model in distributed platform. The main contributions of the thesis are as follows:

- scalable PCA (sPCA) on distributed platform is developed to solve storage and computational burden of traditional PCA.
- The proposed sPCA is utilized on three diverse data sources including OpenStreetMap to prove it enables scalability performance on any high-dimensional big data sources with different data sizes.
- Two-stage dimension reduction approach is proposed to provide not only redundant but also irrelevant data removal from high-dimensional big data.
- For predictive big data analytics, parallel and distributed MLR model is implemented to support large-scale matrix processing.
- In computing "β" values, "QR" Decomposition is applied to overcome the limitations of traditional MLR model in big matrix operations.

- The state of original “Tall & Fat” data form is transformed into compatible state of “Tall & Skinny” data form in order to be efficiently processed by the system.
- In comparing images using Pearson Correlation Coefficient (PCC), region defining stage is applied to support face detection process.

1.6 Organization of the Thesis

This thesis is organized with seven chapters. They are:

Chapter 1 introduces big data and big data analytics, motivation, and also contributions of the research work. It also provides the objectives of the thesis.

Chapter 2 deals with literature review of the related research works. Many existing works to represent predictive big data analytics in high-dimensional big data are reviewed. This chapter also explains current research approaches on dimension reduction for high-dimensional big data.

Chapter 3 describes the detailed explanations of dimension reduction on high-dimensional data using Principal Component Analysis (PCA). In addition, the importance of correlation measures by Pearson Correlation Coefficient (PCC) and three different data sources utilized for the system are also expressed in this chapter.

In Chapter 4, the design of the system including all proposed approaches are discussed in detail. And, the applied techniques, theories, and the proposed algorithms are also described in this chapter.

The implementation for the proposed system including experimental environment and procedures with different approaches are presented in Chapter 5.

In Chapter 6, the experimental results, the comparative studies, the performance evaluations for the proposed system are all discussed.

Conclusions is drawn in Chapter 7. A summary of thesis, limitations, conclusion and points to directions for the future research on the topics are discussed in this chapter.

CHAPTER 2

LITERTATURE REVIEW AND RELATED WORK

Nowadays, the Internet represents a big storage where great amount of information is generated every second. The IBM Big Data Flood Infographic describes 2.7 Zettabytes of data exist in the today digital universe. Moreover, according to this study from Facebook, there are 100 Terabytes updated daily and an estimate of 35 Zettabytes of data generated leading to a lot of activities on social networks annually by 2020. The rise of cloud computing technology and cloud-based data storage have been a precursor and facilitator to the development of big data and big data analytics. Storing huge volume of data available in various formats increasing with high velocity to gain values out of it is itself a big deal. Big data can be described as large volumes of data in complex structures increasing with high velocity which requires advanced technologies, methods and algorithms to acquire, process and store efficiently. Amir Gandomi and Murtaza Haider [49] expressed that big data moves around 7 Vs: volume, velocity, variety, value, veracity, variability, and visibility as follow:

- Volume: The most visible aspect of big data referring to the fact that the amount of generated data has increased tremendously the past years.
- Velocity: Capturing the growing data production rates. More and more data are produced and must be collected in shorter time frames.
- Variety: The multiplication of data sources where comes the explosion of data formats, ranging from structured text to free text.
- Value: The importance or usefulness of the data to those consuming it – is probably the most relevant to organizations.
- Veracity: The need to deal with imprecise and uncertain data is another facet of big data.
- Variability: There are changes in the structure of the data and how users want to interpret that data.
- Visibility: The state of being able to see or be seen is implied. Data from disparate sources need to be stitched together where they are visible to the technology stack making up Big Data.

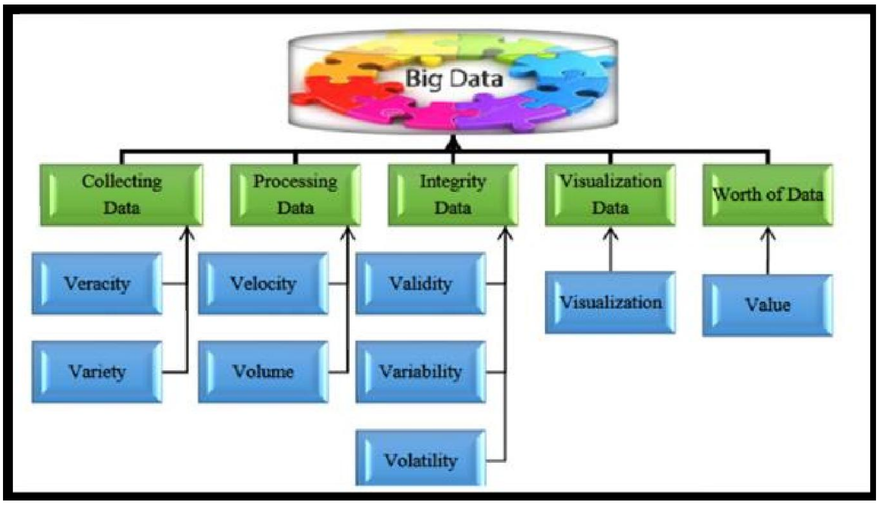


Figure 2.1 Categorization of Big Data's V

In recent studies, many researchers have discovered the importance of big data and the vitality of big data analytics because data is growing in unpredicted volume and complexity. Modern organizations collect massive amount of data to be analyzed for extracting insights which can be used to make better decisions.

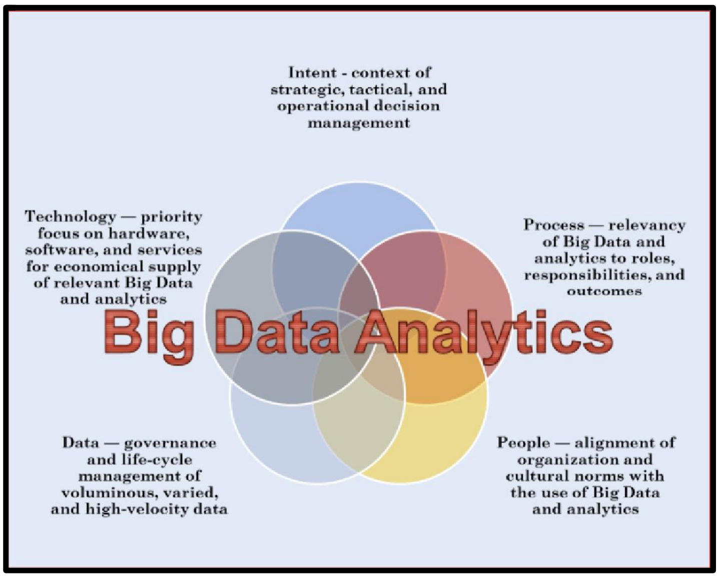

Intent - context of strategic, tactical, and operational decision management
Technology — priority focus on hardware, software, and services for economical supply of relevant Big Data and analytics
Process — relevancy of Big Data and analytics to roles, responsibilities, and outcomes
Big Data Analytics
Data — governance and life-cycle management of voluminous, varied, and high-velocity data
People — alignment of organization and cultural norms with the use of Big Data and analytics

Figure 2.2 Big Data Analytics

2.1 Predictive Big Data Analytics

Generally, data analytics is the primary use of data which can be processed by applying statistical analysis, data mining, machine learning, and mathematical models to obtain better insights and decisions. Thus, a process of transforming data into insights through analysis for decision making and problem-solving can be referred to as data analytics. Furthermore, big data analytics can be defined as the combination of traditional data analytics and data mining techniques together with any large voluminous amount of structured, semi-structured and unstructured data to create a fundamental platform to analyze, model and predict the behavior of customers, markets, products, services and so on. “Hadoop” has been widely embraced for its ability to economically store and analyze big datasets. Using parallel processing paradigm like MapReduce, Hadoop can minimize long processing times to hours or minutes.

2.1.1 Types of Data Analytics

There are mainly four types of data analytics:

- Descriptive Analytics: It is used to transform data into relevant information. On the other hand, it enables to describe what has happened in the system, however, it will not be applied for forecasting.
- Diagnostic Analytics: It is a kind of advanced analytics which examines data for the question “Why did it happen?”
- Predictive Analytics: It involves a number of advanced techniques such as statistical, mathematical, data mining and machine learning to explore data and make predictions for data analysts. Predictive analytics can be applied to forecast what will happen in future.
- Prescriptive Analytics: With this type of analytics, we enable to predict the potential consequences based on different choices for a possible action which can find out the best course of action for any pre-specified outcome.

Therefore, the descriptive analytics which answer the question: “What has happened?”, by using data aggregation and data mining techniques to provide insights into the past, predictive analytics which also replies like this “What could happen in future?” applying statistical models like regression and forecasts to understand the future. It comprises a variety of techniques that can predict future outcomes based on historical and current data and the last one, prescriptive analytics for optimization and simulation algorithms to advice on possible outcomes for the question: “What should we do to happen in future?” [6, 20, 88].

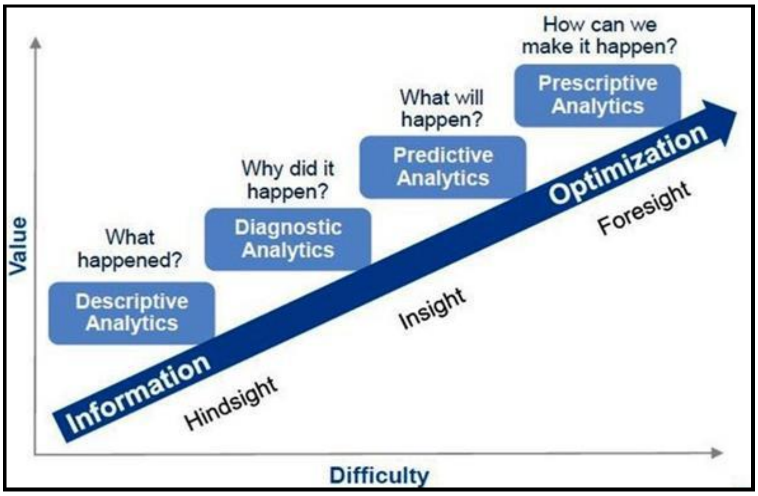


Figure 2.3 Types of Data Analytics

2.1.2 Predictive Data Analytics

Predictive data analytics is the process of extracting information from existing datasets to predict future outcomes and trends with the purpose of forecasting what will happen in the future. It transforms data into valuable and actionable information with an acceptable level of reliability to determine the probable future outcome of an event. Furthermore, it comprises a number of statistical techniques from modeling, data mining, and machine learning and so on. Historical and current data are intended to make predictions with statistical or other analytical models and methods [16]. For example, the predictive models describe patterns that are discovered from historical and transactional data in defining risks and opportunities for business.

There are several ways to analyze big data, however, the underlying problem may be a statistical problem. Typically, learning methods for big data are statistical methods which can generalize and modify the classical ones with new datasets. With big data, there is more emphasis on data quality, dimensionality reduction, predictive capability and reliability of the learning models. The analysis of big data requires the ability of assessing the generalizability and regularization of models with the language of machine learning. The model with an extreme number of variables or observations needs a dimensionality reduction to avoid a large amount of noises that can affect the estimation of that model.

2.2 Statistical Data Analysis for Big Data

The statistics can be defined as "the art of learning from data". The optimal decision is made learning from data by statistics. Analyzing data by statistical methods is typically the primary process of learning from data. In big data era, efficient statistical methods are required because statistics is a good analytical tool in analyzing big data. On the other hand, big data is only data which is considered in a part of data science called statistics. There are actually four parts in data science which studies all about data including big data. They are

- Structure of data
- Collecting of data
- Analysis of data
- Storage of data

Statistics can provide big data analytics as a key performance. It is the science of collecting, analyzing and understanding the data in diverse fields such as health, economics, and education and so on. The analysis of big data, a huge collection of data, which is complex in terms of its volume, variety and velocity at which it is collected. In fact, statistics is a fundamental thing to ensure the meaningful and precise information to be extracted from "Big Data". The following issues or problems in statistics which are emerged from big data:

- The quality of data and missing data
- The observational nature of data
- The quantification of uncertainty of predictions, forecasts and models

The main valuable insight of big data does not derive from the data in its raw condition, it actually comes from consolidation of data, and products and services which can emerge from the analysis. The changes that are emerging not only in data management technologies but also in analytical techniques should be harmonized with decision support process. It is a great opportunity to obtain many advantages from computational and statistical methods to transform raw big data into valuable insights in several areas such as medicine, education, and business intelligence and so on. By using big data in statistical data analysis, the issue of high dimensionality introduces new challenges as follows:

- There cannot have sampling errors, if there is no sampling. It needs to evaluate the model bias or analyze the quality of data.

- In some cases, big data have much more features than observations.
- The number of untrusted correlations is usually increasing with the number of features. It may cause wrong statistical results.
- When testing sequentially has many hypotheses, it needs to correct for multiple testing.
- The traditional statistical modelling assumptions are hardly satisfied, which implies biased model parameters and inaccurate statistical tests.
- With big data, common problems such as sampling bias, missing or incomplete data and sparsity must also be addressed.

According to statistics which is an important part of big data, the big or massive datasets which are applied by statistical modeling can express the relationship among two variables or several variables. The variable which is predictable in a prediction process is called dependent or response variable. Moreover, the variable that is used for predicting the cause of dependent variable is called independent or predictor variable. The statistical dependences between the variables offer the magnitude of correlation among them. Although finding the valuable knowledge from big data may be serious problem big data analytics, however, there are also some important problems in it. One of them is how to apply the whole large amount of data to statistical analysis at once. These datasets are actually too huge to perform for standard statistical analysis on a typical single computer. From a statistical point of view, the large-scale data could emerge either huge numbers of predictors or huge size of samples, or sometimes both conditions [23]. Typically, the statistical methods are used to encounter computational limitation to manipulate extraordinary data size. Thus, it may take time and it needs many efforts to analyze all data at once.

2.3 Regression Analysis and Big Data

Nowadays, statistics takes the important role in big data because many statistical methods are used for big data analysis. Statistical techniques also provide rich functionality for data analysis and modeling, but it can handle only limited amount of data. Regression which can be seen in many areas widely used such as business, the social, behavioral and biological sciences, climate prediction, and so on. Regression analysis is applied in statistical big data analysis because regression model itself is popular in data analysis. However, the standard statistical models applied to big data may face two computational hurdles: data can be too big to hold in a single computer's memory; and the computing job can take too long for the results. These difficulties are demanding for the newly upgraded statistical methods and computational techniques. To settle the computational burden of statistics for big data, in this research, regression analysis is applied for big data on distributed platform to reduce the computational burden of statistical regression model [75].

2.3.1 Challenges of Regression in Big Data Analysis

There are two approaches for big data analysis using statistical methods like regression. First, we extract sample from the big data, and then we analyze this sample using statistical methods. This is actually the traditional approach of statistical analysis. In this process, big data is considered as a population. Sunghae Jun and et al. [38] already expressed that in statistics, a population is defined as a collection of total elements in the subject of study, and the whole population cannot be analyzed because of its analyzing time and other factors. They also discussed the approach to overcome the difficulty in computing memory by using the sub-sampling technique and this approach is also useful for preliminary regression analysis. In fact, a big dataset can be analyzed which close to a population due to the increasing development of large-scale data computation and decreasing the price of data storage. However, the computing burden of big data analysis remains the same because the traditional data analysis using statistical methods have to face a limitation for analyzing big data.

Moreover, splitting the whole dataset into several data blocks, in which applying the classical regression technique for all data in each block. Then, these regression outputs resulted from all blocks are combined as final output [17]. Input data can be sequentially read and then store in primary memory block by block, and analyze data in each block separately. As long as the size of block is small enough, one can easily implement this estimation procedure within each block under various computing environments. However, how to replace sequential processing of these several data blocks still remain as an issue for big data processing according to increasing volume of data [68]. The two challenges are emerged for massive data in supervised learning which is explained by Moufida Rehab Adjout and Faouzi Boufares [1]. First, the massive datasets will face two severe situations such as limiting memory usage and computational hurdles for the most complicated supervised learning systems. Therefore, loading this massive data in primary memory cannot be possible in reality. Second, analyzing the voluminous data may take unpredictable time to response in targeted analytical results.

2.3.2 Multiple Linear Regression

A continuous dependent variable can be predicted from a number of independent variables by using regression analysis. The regression model, a statistical procedure, allows us to forest the linear relationship which associates two or more variables. This linear relationship can explore the amount of change in one variable which is related to change in another variable or variables. And, the regression model can also be applied to examine the statistical significance, to check whether the observed linear relationship could come out by chance or not. The independent variables in regression model can be assumed as causing changes in the dependent variable. Furthermore, regression analysis, a statistical empirical technique, is not only a statistical process for estimating the relationships between variables, but it is also widely used for prediction and forecasting. Linear or simple regression is an approach for modeling the linear relationship between one dependent variable denoted by "Y" and one independent variable denoted by "X". Multiple linear regression is an approach for modeling the linear relationship between one dependent variable and a set of independent or predictor variables.

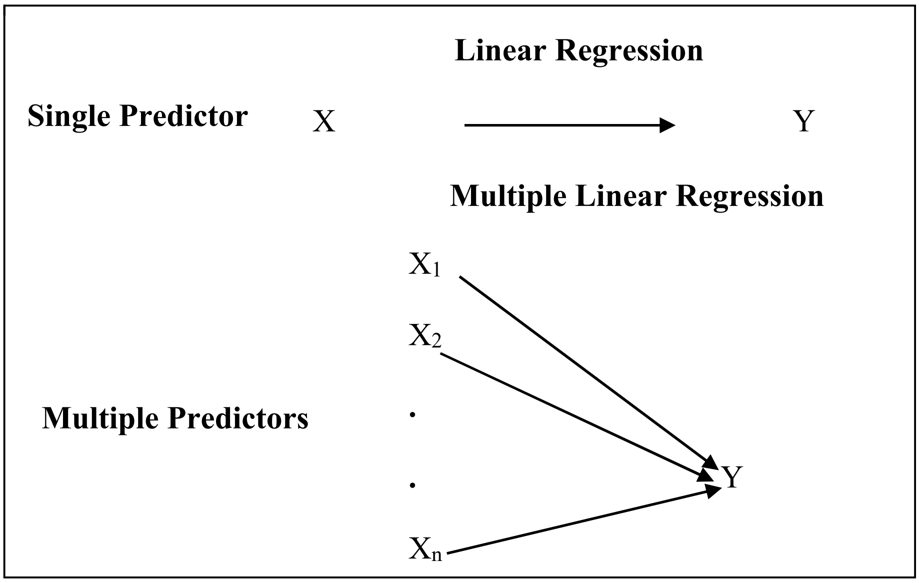


Figure 2.4 Linear Regression vs. Multiple Linear Regression

The simplest form of regression, linear regression, uses the formula of a straight line ($Y_i = \beta_i X_i$ + ƹ) and it determines the appropriate value for β and ƹ to predict the value of Y based on the input's parameters, X.

For simple linear regression, meaning only one predictor, the model can be expressed by Equation 2.1.

$$Y=\beta_0+\beta_1 X_1+\varepsilon \tag{2.1}$$

This model includes the assumption that the ε is a sample from a population with mean zero and standard deviation σ. Multiple linear regression, meaning more than one predictor is represented by Equation 2.2.

$$Y=\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n+\varepsilon \tag{2.2}$$

where Y is the dependent variable; X_1, X_2,…., X_n are the independent variables measured without error; β_0, β_1,….., β_n are the coefficients or parameters of the model. This equation defines how the dependent variable Y is connected to the independent variables X. The primary goal of multiple linear regression analysis is to find β_0, β_1,……., β_n so that the sum of squared errors is the smallest (minimum). When the dataset consists of "n" rows of observations which offers Y_i, X_{i1}, Xi_2,….., X_{ip}; i = 1, 2,…., n. Furthermore, multiple linear regression model enables to specify how much dependency or connection exist between "Y" and one or more "Xs". The matrix formulation of the multiple linear regression in initial version is shown in Figure 2.6 and the complete version is also shown in Figure 2.7.

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}_{N\times 1}, \quad X = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,p-1} \\ 1 & x_{2,1} & \cdots & x_{2,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & \cdots & x_{N,p-1} \end{bmatrix}_{N\times p}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}_{p\times 1}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}_{N\times 1}.$$

Figure 2.5 Matrix formulation of the multiple linear regression in initial version

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \beta_3 x_{13} + \cdots + \beta_K x_{1K} + \varepsilon_1 \\ \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \beta_3 x_{23} + \cdots + \beta_K x_{2K} + \varepsilon_2 \\ \beta_0 + \beta_1 x_{31} + \beta_2 x_{32} + \beta_3 x_{33} + \cdots + \beta_K x_{3K} + \varepsilon_3 \\ \vdots \\ \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \beta_3 x_{n3} + \cdots + \beta_K x_{nK} + \varepsilon_n \end{pmatrix}$$

Figure 2.6 Matrix formulation of the multiple linear regression in complete version

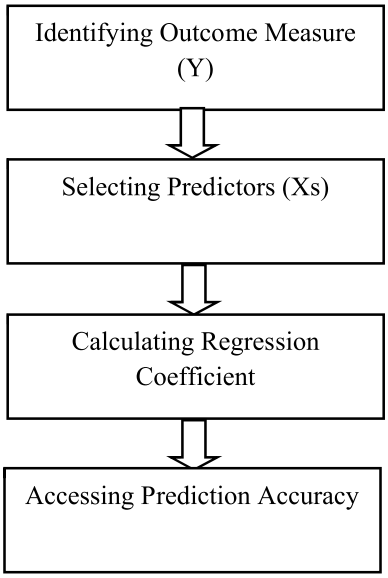

Identifying Outcome Measure (Y)
Selecting Predictors (Xs)
Calculating Regression Coefficient
Accessing Prediction Accuracy

Figure 2.7 Predictive Analysis using Regression Model

2.4 Dimensionality Reduction

Nowadays, in big data era, the terms “Scalable” and “Large-Scale Data Analytics” have been popular in statistics, data mining and machine learning environments. Certain problems have been arisen due to the increasing volumes of data, for example in bioinformatics, or in dealing with large number of text documents. Generally, classical data analysis techniques always give a question about how to deal with extremely large datasets. While the dimensions in data increase, “Curse of Dimensionality” has become a big issue to address. In high-dimensional datasets, the observations which are gathered on individual instances so that a single observation has dimensions in thousands or billions and more. Classical algorithms and methods are typically not designed to solve with this growth of dimensionality of the observation vector [11]. The selection or reduction of features is often applied as a pre-processing step to statistical, data mining and machine learning methods and models. It is, typically, a process of filtering a subset of original features to be optimally reduced to reach a certain evaluation criterion.

Feature reduction or selection is a well-developed research area, however, there is an ongoing issue to make classification and prediction tasks to be more efficient. It identifies the most important features for a learning algorithm to be useful in analysis and future prediction. It enables to enhance knowledge discovery for learning by extracting useful information from high-dimensional data nature [31, 61]. Tsai-Hung Fan and Kuang-Fu [7] Cheng developed a new variables selection criterion collaborating with traditional stepwise procedure in selecting the most important independent variables. They also proposed a two-stage block hypothesis testing method to avoid problems in classical tests for massive datasets. In addition, the most frequent issue of data mining and machine learning for regression model is that how to predict the outcome of a dependent variable when there are a large number of independent variables in the model. With the advanced technologies and modern algorithms for regression model, it is a difficult situation to handle all variables at once for the model. In current time, large-scale high-dimensional datasets are increasingly common and widespread in many areas and applications. Identifying patterns in data and expressing the data to analyze similarities and differences between them can be very hard to find in high-dimensional data.

Dimensionality reduction has several advantages, the most important of which is the fact that with dimensionality reduction, we could drastically speed up the execution of an algorithm whose runtime depends exponentially on the dimensions of the working space. At the same time, the solution found by working in the low dimensional space is a good approximation to the solution in the original high dimensional space. Principal Component Analysis (PCA) can explain a large number of input variables by a small number of principal component results with the purpose of interpreting and understanding easily. PCA also transforms a set of uncorrelated linear combinations, called principal components from the observations of variables. The total number of principal components is less than or equal to the number of input variables in datasets [80]. Tonglin Zhang and Baijian Yang [80] described about principal component analysis, a dimension reduction technique has many issues and challenges in solving high-dimensional big data. They also presented that large-scale standardized matrix computation in PCA. Chaman Lal Sabharwal and Bushra Anjum [50] presented an adaptive hybrid approach by applying PCA to traditional regression algorithms to reduce the dimensionality of a dataset as identifying pattern in data of high dimension can be very hard in data analysis applications.

The central idea of using PCA is to fulfill an advantage of lossless data reduction. Improving the predictive power of traditional multiple linear regression model using PCA is studied by Ahmad Zia UI-Saufie, Ahmad Shukri Yahya and Nor Ramli [70] to predict PM10 concentration for next day. Application of PCA in regression models is intended to avoid multicollinearity problem and to ensure that principal components selected have maximum variance. According to experimental studies, they proved that the principal components as input to regression process offer a more accurate result than original data input to regression process because of reduced number of inputs. Therefore, applying PCA based regression models can be considered as more efficient and decreased complexity models.

In the infrastructure and management for big data, one of important issues in predictive big data analysis is how to apply statistical analysis like regression to the huge data at once. Regression analysis, a statistical process, is widely used for prediction and forecasting by estimating relationships between variables. Moreover, it may not be straightforward for massive datasets [18]. Multiple Linear Regression, a classical statistical data analysis method, also proves unsuitable to the distributed environment with the increasing volumes of data. The most powerful and mathematically mature data analysis method, multiple linear regression is focused on a central approach traditionally where the computation is only done on a set of data stored in a single machine. With an increasing volume of data, the transition to the algorithm in distributed environment is hardly possible to implement. Multiple linear regression also proves unsuitable to facilitate the scalability of the data processed in the distributed environment due to computing memory and response time. In big data era, it is an essential requirement to solve the transition to the scalability of the algorithms for parallel and distributed massive data processing with the use of MapReduce and Spark paradigm seems like a natural solution to this problem.

Young Kyung Lee and et al. [44] expressed that applying the standard and classical principal component analysis technique may become a big problem when it fails to offer consistent results in very high-dimensional setting. They proposed some modifications for standard PCA that can be worked well in high-dimensional data. Sai Kiranmayee Samudrala, Jaroslaw Zola and et al. [4] explained existing traditional dimensionality reduction methods and techniques have encountered scalability issue because they don’t scale well in datasets containing thousands of data points with millions of dimensions for example. It is always better to make prediction models that do not include irrelevant variables. The independent variables in training dataset for regression model can be assumed as the features for dependent variable or predicted output to make predictions. Therefore, the goal is to select the most important predictors for the Multiple Linear Regression Model. In this way, irrelevant features or independent variables can also be dropped for the model.

2.5 Predictive Analytics on Three Different High-Dimensional Data Sources

For predictive analysis on high-dimensional data, there are three different data sources to apply in this research.

2.5.1 OpenStreetMap (OSM) Data

OpenStreetMap (OSM) is an open source data resource for geographic information all over the world. OSM which is maintained by the OpenStreetMap Community has motivation to offer a restriction free mapping solution for commercial and non-commercial applications [63]. There are two main mapping environments to find simply a particular place on a map:

- Google Maps
- OpenStreetMap

These two mapping environments are typically based on different fundamental; however, they solve the same basic requirement to know “WHERE”. The key difference between them is “Open” and “Closed” approach. Every change that make to OSM is owned by individual and the community but every edit which make to Google Maps is owned by Google. Over 2.2 million registered users have applied OSM for the ease of use in open map data sources but OSM community has also managed to achieve the higher data granularity than other map sources [90].

Stefan Funke, Robin Schirrmeister and Sabine Storandt [35] introduced that how to apply methods in detection of gaps in the road network automatically and then discovery of missing street names by using OSM road network data. They showed that data mining and machine learning methods are very useful to detect missing road network data in OSM. Growing rapidly in volume and popularity of geospatial data, Geographical Information System (GIS) applications are demanding to data mining and machine learning approaches integrated with spatial big data. Hemlata Goyal, Chilka Sharma and Nisheeth Joshi [36] presented issues, challenges, tools and algorithms for spatial data mining collaborated with big spatial data. The availability of metadata concerning road information may be one of the primary advantages of OpenStreetMap. The road information consists of the capacity and category of road such as primary, secondary and tertiary. Luda Zhao and Paula Kusumaputri [87] are intended to extract accurate and reliable measures of poverty metrics from publicly available large-scale OSM data leading to efficient policy-making decisions. Principal component analysis can be applied to extract and interpret features, and then the different specific features which are important in predicting poverty measures are also analyzed. These features can characterize the road access in remote regions across two countries in sub-Saharan Africa in effectively predicting key poverty metrics.

Frank F. Xu, Bill Y. Lin and et al. [42] presented an approach for developing a system to learn a prediction model from graphical traffic condition data which is provided by Baidu Map, a map provider in China. Then, the prediction model is applied on OpenStreetMap data with the purpose of predicting the traffic conditions on any roads given a map formatted with OSM with nearly 90% accuracy. Amerah Alghanim and Michela Bertolotto [5] expressed that how to apply spatial data mining techniques in OSM's feature extraction for diverse types of streets. Furthermore, they discussed about how to automatically evaluate the extracted data for predicting the street type by using machine learning techniques. Thus, the key idea of the paper is analyzing and evaluating the quality of street network tags of OSM which based on not only specific features of streets but also their context and user trustworthiness.

Increasing enormous amount of geospatial data, the capability of high-performance computing has been an essential requirement to fully utilize huge collection of geospatial big data with high-velocity in demanding applications. The distributed and parallel computing on a cluster of commodity computers for big data analysis such as Hadoop and Spark have become popular in current time. It can provide geospatial big data analytics easily implemented on big data platforms.

2.5.2 High-Dimensional Image Data

Nowadays, the digital multimedia is so popular and 26,396 Giga Bytes of data is transferred over the world nearly every minute. Among them, 1,873 Giga Bytes of data are only images which requires a large amount of storage space and the demand of resolution is absolutely much higher than before. Thus, data reduction is performed commonly in almost all image processing tasks [50, 65]. In general, the high-resolution images can be assumed as high-dimensional images. These images can be represented as two-dimensional data matrix containing pixel values. Each and every pixel consists of the bits value for RGB which represent this pixel's color. High-dimensional or high-resolution image data has become a big challenge in transmission over Internet as large volumes of images are transferred every day taking a great time. Besides, it may consume a great storage. To solve the problems of taking time and consuming storage in large-scale images, principal component analysis is typically applied for reducing the dimensionality of images. Preservation of images as much as possible is essential for dimensionality reduction technique according to two factors; conserving the memory and increasing the speed of execution of any algorithms which use these images [53]. S.C Ng [49] evaluated principal component analysis on high-resolution image's feature reduction comparing with feature reduced images' quality in different variance values. Their experimental studies show that the images' dimensionality reduction approach can contribute not only significant saving of time in storage and transmission but also still maintaining the integrity of the image.

Shereena V. B and Julie M. David [91] presented the comparative study between two dimensionality reduction methods; principal component analysis technique and linear discriminant analysis technique. These methods are typically used to extract the best features from the images. According to Precision and Recall rates calculated in the paper, principal component analysis outperforms linear discriminant analysis in almost all cases. Therefore, PCA can be denoted as an effective tool for dimension reduction. Basically, PCA can be considered as a powerful data analysis and pattern recognition tool for image processing. It is also a classical, statistical linear transformation method used in data compression. DR. RM. Vidhyavathi [59] discussed about the steps for PCA implementing in medical image processing. The main purpose of this research work is that the various applications of PCA are discovered in the field of medical image processing with a low level of loss. In this work, it is observed that PCA can be used for a variety of image applications such as image compression, image segmentation, image fusion, image registration, and so on. The human face is a vital medium of communication in every day conversation. It offers a lot of information including gender, age, emotional expressions, identity, and so on. Jungseock Joo1, Francis F. Steen, and Song-Chun Zhu [37] presented that the traditional computational methods have many challenges in massive datasets and a large number of feature dimensions that are intractable with traditional methods.

2.5.3 High-Dimensional Text Data

The voluminous amount of text data has been grown rapidly approximately in every day. The requirement of well-designed system for analyzing and classifying upon these unpredicted amounts of data has make many developers, researchers, explorers and others to pay attention this kind of data so-called unstructured big data (big text data). Moreover, classification of text documents can be expressed as the process of assigning one or more suitable categories which are typically based on their content to the text documents by assuming a set of predefined labeled documents as training set. In fact, there have been many issues in classification of text as the increasing size of datasets presents a big issue called "High Dimensionality", a large number of dimensions which create the poor classifier performance problem. This high dimensionality problem can be solved from the feature selection phase as a lot of attributes are redundant and irrelevant. Besides, other issues for high-dimensional text data are representing the features of documents and reducing unnecessary features from these documents.

Typically, text data must be pre-processed before any operation going to perform. In English language, the words may exist in many forms. Therefore, these words should be reduced to their roots. Text data can be transformed into vector space model by applying stop word elimination and stemming methods with the purpose of handling data in terms of numeric values. Dimension reduction, an important processing step, enables text mining procedures by reducing dimensions to be processed with a reduced number of terms retaining important characteristics of the original data. According to advances in data collection and storage capabilities in recent time, the traditional datasets have encountered new challenges in data analysis. Furthermore, the increase in number of observations and also number of variables related with each observation make traditional statistical methods more and more difficult to process. One of the problems of high-dimensional datasets is that all variables are not important and essential for analysis.

To construct predictive models with high accuracy, it is absolutely required to reduce the dimensions of the original data before any modelling of that data. The dimensionality reduction can mitigate the problem of inefficient computation and relationships among terms for detecting and exploiting [67]. Analyzing huge amount of textual data, there are many challenges due to not only in unpredictable size but also inherent noise of data. One approach to sort out is reducing the text data for representing each document together with a few dimensions. Although "Dimensionality Reduction" techniques have been well-studied for images and numerical data, it can be applied to text documents' dimension reduction. In fact, the content of a text document can be assumed as a vector consisting a lot of dimensions, one for each document represents the useful word in the corpus. Document classification is important and useful task in text analysis. David G. Underhill, Luke K. McDowell and et al. [72] has examined that pre-processing with dimensionality reduction techniques could improve text analysis indeed. They also indicated that how well each dimensionality reduction technique has retained the interesting inter-document relationships which help in prediction performance.

When the dimensionality of data becomes extremely high, it can be difficult in learning a classifier called the "Curse of Dimensionality". Dimension reduction as a pre-processing step can greatly promote the classification and prediction accuracy in high-dimensional data [66]. By extracting the most optimal features in small number, the original high-dimensional data space is transformed into a low dimensional space to be more efficiently performed in learning process. Cheong Hee Park [65] showed that the results from multiple linear methods can be applied to compose new features for low dimensional representation in multi-labeled text categorization. The problem in classification automatically a collection of documents either classes or topics can be referred to as categorization of documents.

Typically, data has to be transformed into the suitable form in accordance with the standard data mining and machine learning algorithms. Thus, it may be possible to reduce the dimension of data while retaining the core information in original data as much as it can. Principal Component Analysis, a standard algorithm for data's dimension reduction, enable to transform a set of data points onto a smaller dimensional subspace of something called "best fit". Benjamin Fayyazuddin Ljungberg [79] made an experiment using a collection of texts which is stored as "bags of words" for reducing the dimension of the data. This is implemented by applying Principal Component Analysis (PCA) and Latent Semantic Analysis (LSA) in reducing the dimension of that data. In this paper, the author discussed that the features which are resulted from PCA much better than those resulted from LSA by indicating that PCA can perform much more efficient way to reduce dimensionality of data. In text mining, text categorization exists an important sector to apply in automatically classification of documents into predefined categories based on their content. In fact, the first approaches to text categorization have been based on expert knowledge which may exist sets of rules format to assign documents with given predefined classes [62].

In general, a vector model representation of text documents is used to categorize the classes or targets. The vector can represent the document as a set of document terms and the weight values are also needed to assign to each term [83]. Recently, the classification of text documents automatically has become an essential role in text mining field as the availability of text documents are increasingly growing day to day. Then, the requirement of effectively managing on these large collection of text documents has also demanded in large-scale text data analysis. For text categorization, the traditional approaches were typically based on expert knowledge existing in the form of sets of rules. These rules were used to assign text documents with given predefined classes. Distributed processing on large-scale text data can be performed using Apache Spark distributed big data computing framework. Piotr Semberecki and Henryk Maciejewski [62] presented the implementation of the system which classifies text documents based on the Apache Spark distributed framework intended for distributed processing of big text data. They illustrated a sample classifier which enables to predict the subject category of a text document in English language Wikipedia.

2.6 Big Data Analytics Platforms

Apache Hadoop has become popular due to enormous demand for Big Data in various applications. Commercial Hadoop Distributions are usually packed with features which are designed to streamline the deployment of Hadoop. Furthermore, Cloudera Hadoop Distribution provides a scalable, flexible and integrated platform that can easily manage increasing volumes and varieties of data called Big Data.

There are primarily three types of Hadoop Distribution which have its own set of functionalities and features and are built under the base HDFS. They are:

- MapR Distribution
- HortonWorks Distribution
- Cloudera Hadoop Distribution

2.6.1 Three Types of Hadoop Distribution

Firstly, MapR is a platform-focused Hadoop solutions provider and integrating its own database system called MapR-DB. MapR-DB is proudly declared to be four to seven times faster than the Hadoop database's HBase. MapR is the only Hadoop distribution including Pig, Hive, and Sqoop without any Java dependencies because it relies on MapR-File System. It is the most production ready Hadoop distribution with many enhancements that make it more user-friendly, faster and dependable. Secondly, the HortonWorks Data Platform (HDP) is totally an open source platform designated for data from diverse sources and formats. The platform consists of a variety of Hadoop tools such as the Hadoop Distributed File System (HDFS), MapReduce, Hive, HBase, Zookeeper, Pig and additional components. Finally, Cloudera is the first one to release commercial Hadoop Distribution by offering consulting services to bridge the gap between – "what does Apache Hadoop provides" and "what organizations need". Cloudera Distribution is:

- Fast for business: From analytics to data science and everything in between, Cloudera delivers the performance you need to unlock the potential of unlimited data.
- Makes Hadoop easy to manage: With Cloudera Manager, automated wizards let you quickly deploy your cluster, irrespective of the scale or deployment environment.
- Secure without compromise: Meets stringent data security and compliance needs without sacrificing business agility. Cloudera provides an integrated approach to data security and governance.

Table 2.1 Comparison between Three Hadoop Distribution

No.	Hadoop Distribution	Advantages	Disadvantages
1.	Cloudera Hadoop Distribution	User friendly interface with many features and useful tools	Slower than MapR Hadoop Distribution
2.	MapR Distribution	One of the fastest hadoop distribution with multi node direct access	It does not have a good interface console as Cloudera Hadoop Distribution
3.	HortonWorks Distribution	The only Hadoop Distribution that supports Windows platform	It does not have many rich features in interface

2.6.2 MapReduce

Apache Hadoop is capable of running MapReduce, a programming framework, is suitable for processing extremely large amount of data. It enables large-scale datasets for parallel and distributed processing. MapReduce programs are very useful in performing large-scale data analysis which is implemented on a number of machines in the cluster. These programs work in two distinct tasks:

- Map
- Reduce

In “Map” task, input data is read and processed and then it produces key-value pairs as intermediate output. The intermediate output of map task is then passed as input to the “Reduce” task. It receives the key-value pairs from a number of maps. Then, it aggregates or merges these intermediate values into a smaller set of tuples or key-value pairs which is the final outcome. In the following figure, the fundamental processing of “Map” and “Reduce” tasks on input data. Zhu et al. [85] have discussed about infrastructure, data flow, and processing of MapReduce Framework.

MapReduce, a programming platform cooperating with HDFS in Hadoop, which is popular in analyzing huge amount of data.

There are two kinds of computational nodes in MapReduce Framework: one master node ("Name Node") and several slave nodes ("Data Node"). This can be known as master-slave architecture and all the computational nodes and their respective operations are in the form of massively parallel and distributed data processing. The master node serves the duty of entire file system and each slave node serves as a worker node. Actually, each slave node performs the two main phases or processes called Map () and Reduce (). The data structure for these both phases exist in the form of <Key, Value> pairs. In the Map phase, each worker node initially organizes <Key, Value> pairs with same key nature and then produces a list of intermediate <Key, Value> pairs as intermediate Map results. Moreover, MapReduce system can also perform another shuffling process in which intermediate results produced from all Map operations by lists of same-key pairs with an implicit set of functions such as sort, copy and merge steps. Then, the shuffled lists of pairs with the specific keys are combined and finally passed down to the Reduce phase. In the Reduce phase, it takes lists of <Key, Value> pairs that are resulted from previous process to compute the desirable final output in <Key, Value>pairs.

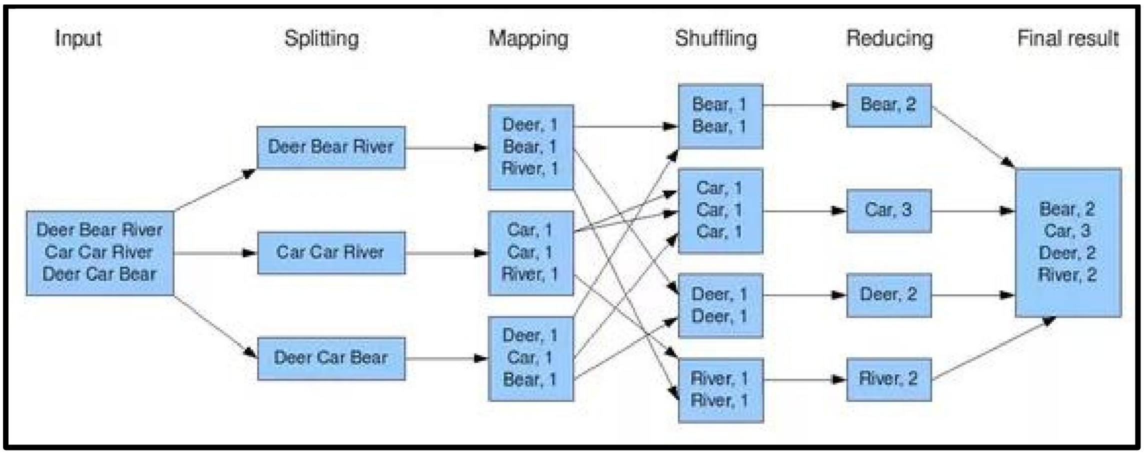


Figure 2.8 Processing Flow of MapReduce with Example

2.6.3 Architecture of Apache Spark

In fact, Apache Spark has emerged to overcome the problems of Apache Hadoop, a distributed data processing framework, which was developed by Google.

The significant drawback of Hadoop is that it always performs read/write operations to and from hard drive vice versa for each MapReduce job. On the other hand, Apache Spark enable to cache the data in memory and it can serialize RDD structures to store them on the hard drive. Apache Hadoop provides only batch processing of data; however, Apache Spark has the ability to run applications interactively. Moreover, the distributed big data processing like MapReduce have many successful milestones in large-scale matrix computations. But it makes a problem of inefficient iterative computations. Apache Spark, a distributed data processing framework, can be utilized to solve the problem of iterative processing by providing efficient computations on Resilient Distributed Datasets (RDDs).

In fact, the step by step operations on sets of RDD sets are transformed into direct acyclic graph which can be scheduled and processed in parallel by the worker nodes in the clusters. The evaluation will be under control by the main or driver program. Apache Spark has many benefits in matrix computations as follows:

- Resilient Distributed Datasets (RDDs), storage abstraction of Spark, is a distributed fault-tolerant vector for developers who can perform a subset of operations from a regular local vector.
- RDDs allow not only user-defined data partitioning but also execution engine to co-partition RDDs and co-schedule tasks for movement of data.
- To build an RDD, spark usually logs the lineage of operations to be capable automatic reconstruction of lost partitions upon failures without concern for performance. Moreover, spark fulfill a benefit of reducing the amount of re-computation on failure with optional in-memory distributed replication.
- Spark also supports a high-level API in Scala which can help in creating a coherent API for matrix computation.

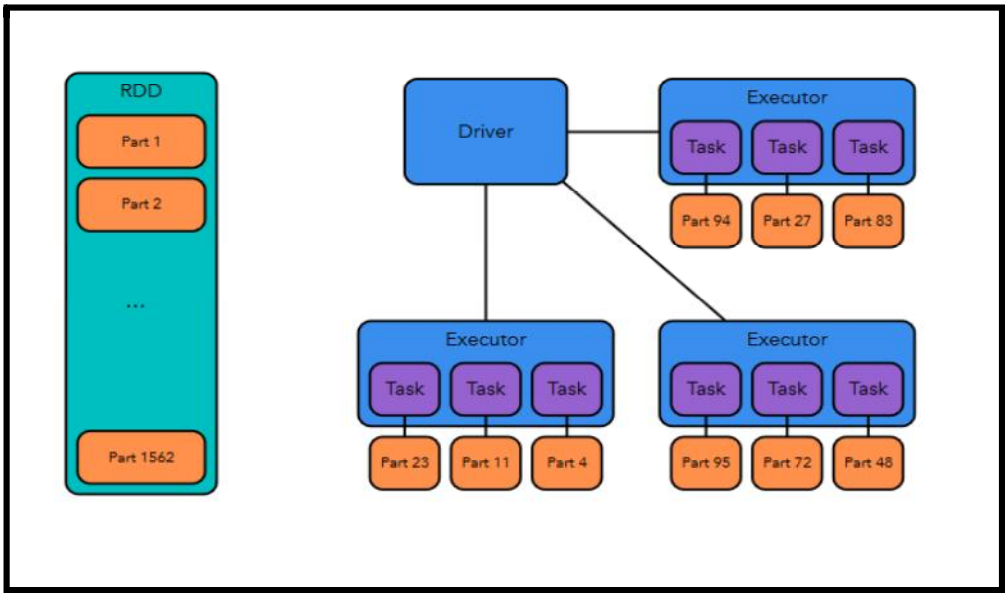


Figure 2.9 Apache Spark

The initial phase of data processing in Apache Spark creates RDD objects from loaded data. The order of the operations on RDD sets is transformed into direct acyclic graph, scheduled and processed in parallel by the cluster worker nodes. The main program - driver controls the evaluation. Apache Spark has native libraries that support common Machine Learning algorithms for Clustering, Classification and Data Dimensionality Reduction, which are optimized for RDD datasets processing. Apache Spark nowadays is quite popular to scale up any data processing application. For Machine Learning also, it provides a library called "MLlib". It is a distributed programming approach to solve ML problems.

2.7 Chapter Summary

In this chapter, there are many sections to categorize for shaping the system to be implemented for this research. The main purpose of the chapter is to discuss the reviews of related research works to motivate in construction of the proposed system. The discussions of current trends in big data, big data analytics, and types of data analytics including predictive data analytics are presented to understand. The statistical regression analysis for big data using multiple linear regression model is also discussed to clearly understand it is intended to implement the proposed system for prediction purposes. The importance of dimension reduction in high-dimensional data analytics and then the three high-dimensional data sources utilized in this system are introduced in above sections. In addition, the big data analytics platforms for parallel and distributed data processing, types of Hadoop Distribution including Cloudera Hadoop Distribution, and two popular processing frameworks for big data namely, MapReduce and Spark are completely discussed.

CHAPTER 3

DIMENSIONALITY REDUCTION ON HIGH-DIMESIONAL BIG DATA

Storing huge amount of data available in various formats which is increasing with high speed to gain value out it is a great opportunity to emerge big data. Big Data can be defined as the amount of data which beyond traditional technologies and processing capabilities to store, manage and process efficiently. The rise of cloud computing technology and cloud-based data storage have been a precursor and facilitator to the development of big data and big data analytics. Big data is also revolutionizing traditional operations and data analysis everywhere such as researches in artificial intelligence, finance, biology, genetics, engineering and astronomy and so on. In recent studies, many researchers have discovered the importance of big data and the vitality of big data analytics because data is growing in unpredicted volume and complexity. Owning to rapid development in studies and technologies, the growing number of research fields encounter data with unprecedented size and complexity such as researches in artificial intelligence, economy, finance, biology, genetics, engineering and astronomy. The importance of data and the vitality of data analysis cannot be downplayed in contemporary science. Data analysis, also known as analysis of data or data analytics, is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, suggesting conclusions, and supporting decision-making.

3.1 High-Dimensional Data Analysis

As computational power increases and the expense of data collection and processing decrease significantly, the dimensions of datasets is continuously growing in size. Among thousands of dimensions or feature variables, only a small number or subset of them are possible to extract value or insight in data analysis. This makes a critical situation to identify correctly and to reduce efficiently them. And, finding interesting features in datasets will fulfill valuable information to support decision making. As the number of dimensions of data increases, the processing on these dimensions may become more and more complex indeed. The sophisticated techniques are being required to minimize the complexity of analysis in high-dimensional large-scale datasets [35]. Therefore, dimensionality reduction scheme is absolutely required to facilitate high dimensions of big data. High-dimensional big datasets have experienced many issues and challenges in extracting useful insights from it. Therefore, high-dimensional data analysis has been a great attention in big data era. The main purpose of dimensionality reduction is to find out how many dimensions can be reduced from all diverse and raw data dimensions. In current time, dimensionality reduction has been playing a key role in high-dimensional voluminous amount of data as the complexity of big data often makes dimension reduction techniques necessary before conducting statistical inference. In high-dimensional datasets, the dimension of predictor variables “p” can be as large as or much larger than the sample size n, but very often, among thousands of available predictor variables only a small number of them are informative and it is critically important to identify them correctly.

3.1.1 Curse of Dimensionality

High-dimensional data analysis has received a tremendous of attention recently. Moreover, high-dimensional big datasets are increasingly common and widespread in many areas and applications. Identifying patterns in data and expressing the data to analyze similarities and differences between them can be very hard to find in high-dimensional big data. When a dataset becomes very large, it is needed to consider how to manipulate these large-scale data with the convenient form of computing. We call “High-Dimensional” data in which one observation or record contains dimension in thousands or millions or more while only tens or hundreds of observations. While exponential increase in the size of data caused by a large number of dimensions in big data make a big problem in data analysis. This is “Curse of Dimensionality” in high-dimensional big data analytics. Curse of Dimensionality, a big issue to address, refers to extremely increasing in the size of data that is caused by single observation comprising dimension in thousands or millions or billions, we called “High-Dimensional” data. In real world applications involving text, image, audio, video and many applications involving biomedical, relational or network data also deliver datasets with high dimensions.

According to higher dimensions, the processing on these dimensions may become more and more complex indeed. The sophisticated techniques are being required to minimize the complexity of analysis in high-dimensional big datasets [10].

In fact, the curse of dimensionality can be expressed as an appearance from high-dimensional data, and it often creates severe consequences on the behavior and performance of learning algorithms [24]. When we encounter the difficulties which are resulting from the problem of high-dimensional data space, there will be only a solution to reduce these dimensions without losing relevant information in the original data. Therefore, dimensionality reduction has become a solution to curse of dimensionality for high-dimensional datasets. It is used as preprocessing step before applying data analysis models with a lower dimension [73]. Moreover, there are many diverse data sources that can be viewed as a large-scale matrix form. In many matrix operations in diverse applications, the matrix can be summarized by finding "narrower" matrices which have only a small number of columns or dimensions. These matrices can be used to apply more efficiently than the original large-scale matrices and how to transform the narrow matrices can also be called dimensionality reduction. High dimensionality in large-scale datasets causes modeling and processing challenges. Principal Component Analysis (PCA), a fundamental dimensionality reduction model in statistics, data mining and machine learning with the advantage of being usable to any dataset with high dimensions.

3.1.2 Multicollinearity in Regression Model

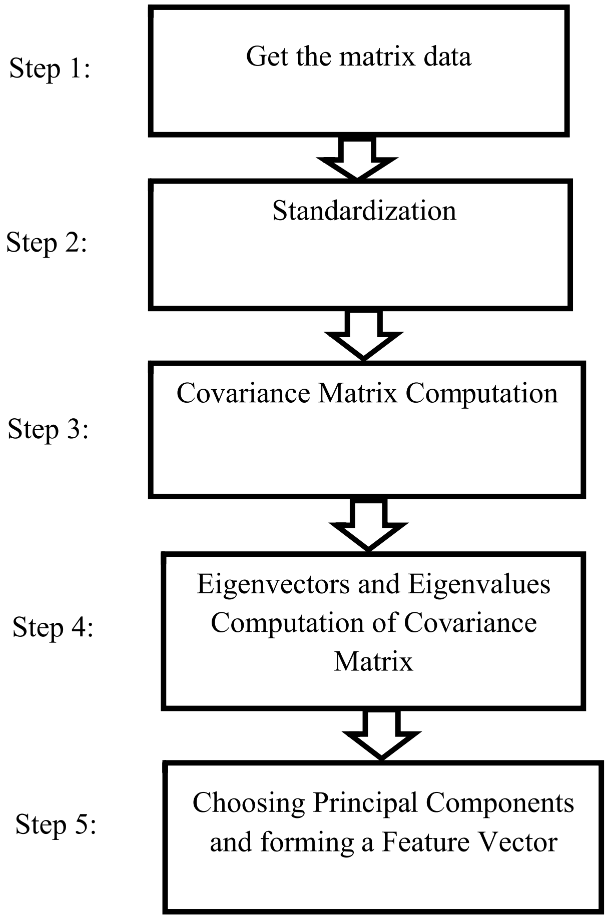
Typically, regression analysis can face serious difficulties when "Multicollinearity" exists among the independent variables. There is a linearly dependence among the independent variables in regression analysis arises a problem, "Multicollinearity". It is a statistical phenomenon which causes many difficulties in linear regression models such as insignificant variable selection and biased variances of regression models [3]. In multiple linear regression models, regressors are sometimes highly correlated with one another. Consequently, it can affect harmfully on the estimation of regression parameters to be inaccurate and also on variables selection techniques. When two or more regressors are correlated with each other, it may also affect the variance of coefficient estimates to be increased. Increasing the variance of regression estimates not only causes wrong sign regression coefficients but also makes more difficult to specify the correct model more difficult. In addition, a dataset consists of extremely high dimensions (multiple variables) for multiple linear regression analysis may degrade the predictive power of the model. To get better prediction outcomes, constructing the better model is an important task of data analysis process.

In present time, the terms “Big Data” and “Predictive Data Analytics” have been popular in statistics, data mining and machine learning environments. Certain problems have been arisen due to the increasing volumes of data, for example in bioinformatics, or in dealing with large number of text documents. Generally, classical data analysis techniques always shoot a question about how to deal with extremely large datasets. Analyzing high-dimensional data and then predicting some insights from it become more and more difficult in big data era. As a predictive analysis, the multiple linear regression model is used in this system to explain the linear relationship between one dependent variable and two or more independent variables. Although multiple linear regression, the most powerful and mathematically mature data analysis method, has succeeded in many application areas, it will encounter the two severe conditions such as “Curse of Dimensionality” and “Multicollinearity” because of high-dimensional data input in regression model.

3.2 Principal Component Analysis (PCA)

Predictive big data analytics with multiple linear regression, voluminous numerical data often can be modeled as a number of input independent variables or regressors along with one output dependent variable. Regarding the problem of “Multicollinearity” in regression analysis and the accuracy of predicting values, we can use a process called “Principal Component Analysis”. It is a dimension reduction approach which is used to reduce a large set of correlated predictor variables into a smaller, less correlated set, called principal components. These components still contain most of the information in the larger set. For instance, the first principal component contains as much of the variability in the data as possible. Thus, principal component analysis, a dimension reduction technique, has also become a remedy to “Curse of Dimensionality” in high-dimensional data analysis.

Principal Component Analysis (PCA) performs dimensionality reduction by extracting the principal components (PCs) of high-dimensional data. PCA can explain a large number of input variables by a small number of principal components (PCs) with the purpose of interpreting and understanding easily. It provides a roadmap for how to reduce a complex dataset to a lower dimension with the advantage of being applicable to any dataset with numeric dimensions [26]. It can be applied alone and sometimes it can be a starting solution for other dimension reduction methods. Therefore, PCA is often applied as the first step of data analysis, which may be followed by linear regression, multiple linear regression, cluster analysis, image analysis, and many others. PCA also transforms a set of uncorrelated linear combinations, called principal components from the observations of variables. The total number of principal components is less than or equal to the number of input variables in respective datasets. In general, datasets can be represented as matrices and vectors with a lot of features. For a matrix, each column refers to a conceptual attribute of all the data. Reducing big original data matrix into smaller one but retaining the same information of original data matrix to gain value or insight from this. Computing PCA of a matrix Y of size N ×D (N rows and D columns), it can be obtained "d" principal components ($d \leq D$) that explains the most variance (information) of the data in matrix Y [14, 15, 58, 70, 82]. The fundamental processing steps of PCA is shown in Figure 3.1.


Step 1: Get the matrix data

Step 2: Standardization

Step 3: Covariance Matrix Computation

Step 4: Eigenvectors and Eigenvalues Computation of Covariance Matrix

Step 5: Choosing Principal Components and forming a Feature Vector

Figure 3.1 Fundamental Processing Steps of PCA

The input for PCA is mainly numerical form. If the data is other form, for example, categorical or logical, it must be converted into numeric first. And then, eigenvalues and eigenvectors are computed to transform original high-dimensional data matrix into lower dimensional one. PCA decomposition for a data matrix A which is square and symmetric is A = UDU^{T} where U is matrix of eigenvectors and D is diagonal matrix of eigenvalues of A. PCA also arranges eigenvalues by ordering in descending magnitude. In data mining, each observation is a vector with “n” components in an “m x n” data matrix. The largest possible variance can be found in the first principal component (PC) which is extracted from PCA process and it will be a maximum amount of variance in the observed data variables. The second principal component or second PC will be uncorrelated with the first PC and the remaining PCs computed from PCA possess the same characteristics [15, 30, 36, 40, 74].

According to the Figure 3.1, firstly, the input high-dimensional data matrix is applied to be processed. To perform standardization prior to PCA, the input large-scale matrix can be done by subtracting the mean and dividing by the standard deviation for each value of each variable by the Equation 3.1.

$$\mathrm{z} = \frac{\text{value}-\text{mean}}{\text{standard deviation}} \tag{3.1}$$

The aim of covariance matrix computation step is to understand how the variables of the input data matrix are varying from the mean with respect to each other. Sometimes, variables which contain redundant information are highly correlated. In order to identify these correlations, the covariance matrix is calculated. The covariance matrix is a $p \times p$ symmetric matrix where p is the number of dimensions. For example, for a three-dimensional dataset with three variables x, y, and z, then the covariance matrix is a “3×3” matrix of this from:

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

And then, eigenvectors and eigenvalues are computed from the covariance matrix in order to identify the principal components of the data. By ranking the eigenvectors in order of their eigenvalues from the highest to the lowest, principal components in order of significance are obtained. Principal components are typically new variables which are constructed as uncorrelated linear combinations of the original input variables. Finally, a feature vector or matrix of vectors are constructed by taking the eigenvectors from the list of eigenvectors and forming an output matrix together with these eigenvectors in the columns.

$$\text{FeatureVector} = (\text{eig}_1, \text{eig}_2, \ldots\ldots, \text{eig}_n) \tag{3.2}$$

PCA actually attempts to put the maximum possible information in the first principal component, then second maximum possible information in the second component and so on, as shown in Figure 3.2.

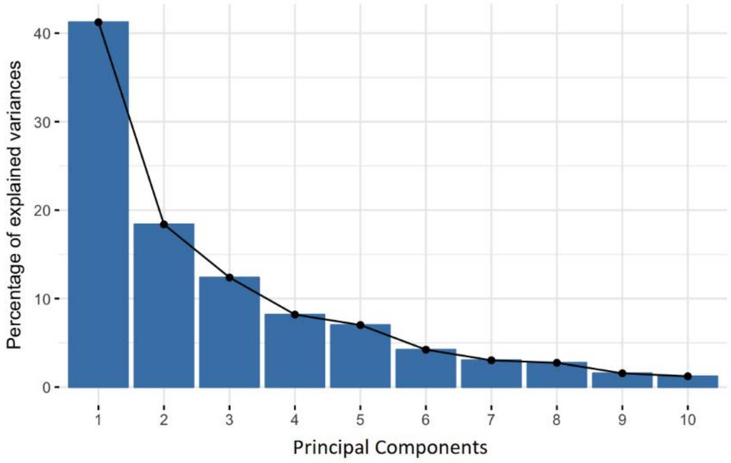


Figure 3.2 Percentage of Variance (Information) for each PC

There are four different ways of Computing PCA:

- Using Traditional way
- Using SVD
- Using Stochastic SVD (SSVD)
- Using Probabilistic Theory

In this research, we would like to prove that PCA which is mostly applied in dimension reduction can also effectively reduce insignificant and sometimes, noisy predictors or independent variables of multiple linear regression model. There are a number of reasons why variable selection becomes an essential role in constructing the optimal regression model. A large number of variables will also cause a problem called “Multicollinearity”. It is a statistical phenomenon of existing a perfect or exact relationship between predictors which will cause incorrectness about the relationship between predictors and outcome variable of that regression model [2]. Moreover, too many variables to a multiple linear regression model increase the amount of explained variance in the dependent variable which may result in an over-fit model. In fact, feature selection or data dimension reduction in predictive analytics refers to the process of identifying the most important features or variables or parameters which can improve the quality of predictive outcome. Dimension reduction tends to raise the statistical significance of predictors or variables of the model. The primary goals of PCA are not only reducing a large number of variables to the smaller number of principal components and but also providing multiple linear regression model by reducing a large set of correlated variables into a smaller, uncorrelated set, called principal components, that still contains most of the information in the larger set.

In recent studies, feature subset selection is intended to identify and then remove as many irrelevant and redundant features as possible. As irrelevant features are features which can even make negative effects to the predictive accuracy. The redundant features are features which cannot help for getting a better predictor even if they are being relevant to the target feature [60, 76]. In general, if the correlation measure between a feature and response variable is high enough, this feature will be relevant to include as a predictor in the prediction process. Pearson Correlation Coefficient (PCC), sometimes it is referred to Pearson Product Moment Correlation Coefficient is applied between a pair of variables (X, Y) to measure the strength and the direction of a linear relationship between two variables. In this research, two-stage dimension reduction approach is proposed by applying Principal Component Analysis (PCA) and correlation-based measure, Pearson Correlation Coefficient (PCC) for selecting the most important and correlated features or variables. It is intended to improve the predictive power of statistical model, especially multiple linear regression model applying the proposed dimension reduction approach.

3.3 Correlation-based Measures

Correlation, a statistical measure, indicates the direction of a linear relationship between two random variables. If two variables X and Y are assumed as correlated when the variations in X may be accompanied by the variations in Y. The variations may be either the same or opposite direction. For the same direction, both X and Y increases or decreases together, the correlation is said to be positive. On the other hand, either X increases and Y decreases or X decreases and Y increases, the correlation is said to be negative. When Y is unchanged by any variations in X, then the correlation between X and Y is said to be uncorrelated [8].

In general, a feature may be good candidate for optimal feature subset if it is relevant to the target (output) feature or features but it may not redundant to any other relevant features. We assume that correlation between two features may be a good measure when a feature is highly correlated to the target but not highly correlated to other features. In other words, the correlation between a feature and the target class is high enough to make it relevant to (or predictive of) this class [30]. There exist broadly two approaches to measure the correlation between two random variables. They are:

- Classical Linear Correlation
- Information Theory

3.3.1 Pearson Correlation Coefficient (PCC)

The most well-known correlation-based measure is "Linear Correlation Coefficient". The linear correlation coefficient can be referred to as the "Pearson Correlation coefficient" or "Pearson Product Moment Correlation Coefficient". For a pair of variables (X, Y), the Linear Correlation Coefficient "r" is given by the formula which measures the strength and the direction of a linear relationship between two variables [21]. The Pearson Correlation Coefficient (PCC) denoted by "r" is commonly utilized in statistical analysis, image processing and pattern recognition. It is a statistical measurement for the strength of a linear relationship between the paired data. It ranges between -1 and +1 and quantifies the direction and strength of the linear association between the two variables. The correlation between two variables can be positive or negative. The sign of the correlation coefficient indicates the direction of the association. The magnitude of the correlation coefficient indicates the strength of the association.

$$r = \frac{\sum_i (x_i - x_m)(y_i - y_m)}{\sqrt{\sum_i (x_i - x_m)^2}\sqrt{\sum_i (y_i - y_m)^2}} \tag{3.3}$$

where x_m is the mean of X, and y_m is the mean of Y. The value of r is such that $-1 < r < +1$. The + and – signs are used for positive linear correlations and negative linear correlations, respectively.

3.3.2 Three Forms of Correlation

There are typically three forms of correlation as follow:

- Positive Correlation: If X and Y have a strong positive linear correlation, r is close to +1. An r value of exactly +1 indicates a perfect positive fit. Positive values indicate a relationship between X and Y variables such that as values for X increases, values for Y also increase.
- Negative Correlation: If X and Y have a strong negative linear correlation, r is close to -1. An r value of exactly -1 indicates a perfect negative fit. Negative values indicate a relationship between X and Y such that as values for X increases, values for Y decreases.
- No Correlation: If there is no linear correlation, r is 0 or a weak linear correlation, r is close to 0.

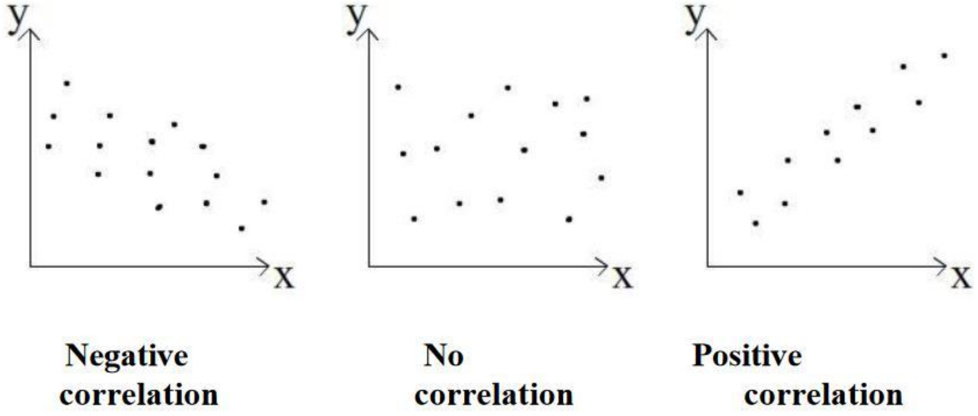


Figure 3.3 Three Forms of Correlation

In this research, there are three diverse high-dimensional big data sources which are applied to implement the proposed system.

3.4 Geospatial or Spatial Big Data

Nowadays, it can be estimated that 2.5 quintillion bytes of data is being generated every day and a large data portion among them is location-aware. Geographical location-aware data which is usually stored as coordinates and topology for mapping can be referred to as geospatial or spatial data. Geospatial big data or spatial big data is deserved to pay attention in analyzing large-scale spatial datasets which exceed traditional computing systems [26, 36, 43]. Geospatial big data cannot be assumed as new issue or problem in data analytics era. Due to not only exponential increase in data production but also in data production rate (velocity). In EOSDIS, 4TB of remote sensing data archives are growing in every day. This data flow means more than 630 million data files, nearly 20 TB can be delivered to users all over the world. The observation data of NASA in each unit time can be collected from approximately 100 active missions which would be about 1.73GB. The volume of OSM datasets typically increases in every year as it is a large collection of geospatial information [22, 57].

In current time, huge amount of geospatial data can be generated from hundreds of millions of mobile phones, sensors, satellites and other resources. The enormous increase of geospatial data, the capability of high-performance computing has been an essential requirement to fully utilize huge collection of geospatial big data with high-velocity in demanding applications. High performance computing or cloud computing platforms are absolutely required in analyzing large-scale geospatially enabled contents [2]. The distributed, parallel computing on a cluster of commodity computers for big data such as Hadoop and Spark have become popular in current time. It can provide geospatial big data analytics easily implemented on big data platforms. By analyzing geospatial data, we can make innovative activities in our daily life and business. In general, we can classify geospatial data into three categories such as raster data, vector data, and graph data.

- Raster Data: It consists of geo-images taken by digital cameras, satellite etc. and it can be utilized by digital map services, for example, Google Earth.
- Vector Data: The map data belongs to vector data category which includes points, lines, and polygons, for example, OpenStreetMap.
- Graph Data: This kind of geospatial data appears in the form of city maps including roads and landmark. In road networks, an edge can be represented as a road segment, and a node as an intersection or a landmark.

3.4.1 OpenStreetMap (OSM)

In 2004, a community of mappers contributed and maintained data about roads, trails, railway stations, cafés, and many others, around the world to build OpenStreetMap (OSM). Contributors verify that how OSM is accurate and up to date not only emphasizing local knowledge but also using aerial imagery, GPS devices, and maps in low-tech field. Contributors of OpenStreetMap's community consists of GIS professionals, engineers running the OSM servers, enthusiast mappers, humanitarians mapping disaster-affected areas, and many more. OSM which is diverse and growing with high speed has become one of the most important data sources for consideration as it is open data: ones are free of charge to apply for any purpose [28]. Vast amounts of raw, unstructured and spatially attributed data are continuously generated and available from OpenStreetMap. The entire OSM data typically in the form of XML and PBF can be publicly accessed on http://planet.openstreetmap.org/.

An increasing number of developers, researchers and other end users are using OSM data in their applications. Many well-known applications and services collaborating with some kinds of geolocation or map-based component using OSM data are as follows: OpenStreetMap-based map for iPhoto for iOS and it has been cited a lot of sources for Apple's custom maps in iOS 6. Interactive data visualization products by Tableau Software Company has integrated OSM for all their mapping requirements. The professional robot simulator widely used for educational purposes, Webots applies OSM data to create virtual environment for autonomous vehicle simulations. The free web-based travel guide for travel destinations and travel topics, Wikivoyage uses OpenStreetMap as a locator map for cities and travel points of interest. Mapworks incorporated the OSM dataset for rendering under a vector publication method and many other applications and services widely apply OSM data resource around the world.

OSM uses three types of files:

- Data files

  The files containing the OSM data is commonly used for a specific point in time. It is intended for every object (node, way, or relation) and the usual suffix is ". osm".

- History files

These files are intended not only the current version but also their history of an object. The usual suffix is “. osm” or “. osh”.

- Change files

  These files are sometimes called “diff” files or “replication diffs”. Changes between one state into another state of the OSM database and various versions of an object. The usual suffix is “. osc”.

3.4.2 Structures of OSM Data

All OSM data can be extracted for a region, or specific map features for a region, or specific map features for a user-defined area. Furthermore, OSM is an open source data resource for geographic information all over the world. The raw, unstructured large-scale OSM XML files are available for developers to create free, editable map of the world. It uses a topological data format with four main elements which also known as data primitives:

- Nodes are typically pointing with a geographic position which are stored as pairs of a latitude and a longitude called coordinates.
- Ways are ordered lists of nodes which can represent a polyline or a polygon if they form a closed loop.
- Relations are ordered lists of nodes, ways and relations
- Tags are key-value pairs and they are used not only for storing metadata about the map objects but also for describing attributes of the features being shown by the geometries.

A node can represent a specific point on the earth’s surface which is defined by its latitude and longitude [79]. Thus, it includes at least an id number and a pair of coordinates. For example, a node can represent a park bench or a water well. Besides, nodes are also used to define the shape of a way. When points are along ways, nodes usually have no tags. A node can be a member of relation and the relation also may indicate the member’s role. A way is typically an ordered list of between 2 and 2,000 nodes which define a polyline. Ways are used to represent linear features such as rivers and roads. Moreover, ways can also represent the boundaries of areas (solid polygons) such as buildings or forests. In this case, the way’s first and last node may be the same. This is called a “closed way”. In fact, closed ways occasionally represent loops, such as roundabouts on highways, rather than solid areas.

The way's tags must be examined to find out which it is. Areas with holes, or with boundaries of more than 2,000 nodes, cannot be represented by a single way. The feature requires a more complex multi-polygon relation data structure. A relation is a multi-purpose data structure that documents a relationship between two or more data elements (nodes, ways, and/or other relations). For example, a route relation, which lists the ways that form a major (numbered) highway, a cycle route, or a bus route. A multi-polygon that describes an area with holes. Therefore, relations can have different meanings which can be defined by its tags. Typically, the relation has a 'type' tag. The relation's other tags need to be interpreted in light of the type tag. The relation is primarily an ordered list of nodes, ways, or other relations. These objects are known as the relation's members. Each element can optionally have a role within the relation.

A single element such as a particular way may appear multiple times in a relation. Besides the basic nodes and ways, our information also requires from another important part of the data format, other than just longitude and latitude, and that is the tags of those basic elements. All types of data element (nodes, ways and relations) can have tags. Tags describe the meaning of the particular element to which they are attached. A tag consists of two free format text fields; a "key" and a "value" [27]. Each of these are Unicode strings of up to 255 characters. For example, "highway=residential" defines the way as a road whose main function is to give access to people's homes. There is no fixed dictionary of tags, but there are many conventions documented on the OpenStreetMap wiki. If there is more than one way to tag a given feature, it's probably best to use the most common approach.

The raw, unstructured XML format large-scale OSM spatial dataset which can provide as reality fulfillment to GIS market and spatial world is applied in this system. Firstly, it is actually XML, a Meta format, which enable to provide human readable data interexchange formats and a variety of file formats can use this type of tree structure data to embed their data. The XML format with a following XML schema definition is mostly applied by major tools of OSM universe. It is generally the instances of data primitives such as nodes, ways, and relations. There is an issue to generate geospatial data and then pre-process for further applying in diverse domains [24]. In general, OSM data exists in the form of data structures such as nodes, ways and relations. It is essential to transform the raw, unstructured OSM XML format data into suitable format compatible with big data analytics platforms such as MapReduce and Spark.

3.5 High-Dimensional or High-Resolution Images

In current time, the enormous growth in digital information has motivated many researches related with images. When images are increasing in today's multimedia age, the most significant visual features are at high-dimensional data space. An image can typically be assumed as a two-dimensional matrix which can be specifically arranged in rows and columns. Every element of this matrix is called image element or pixel. The fundamental processing of an image consists of image acquisition, pre-processing, feature extraction, segmentation, representation, description, recognition and interpretation. In feature extraction, visual information is extracted from the images and saves them as feature vectors. One of the major problems is that the number of features extracted from images which exceed traditional computer system's memory status and computation power. Therefore, traditional computer systems have encountered issues and challenges to represent and analyze tremendous amount of image data. Moreover, extracting efficient features from images become essential as processing all features directly can face a problem called Curse of Dimensionality. It has also introduced many issues and challenges including computational complexity, feature sparsity and redundancy. To solve the problem, there is a need to extract features which can express the data with efficient accuracy.

The use of dimension reduction algorithms, a solution for high-dimensionality, which extract only essential features from the feature vectors of images. The results of feature extraction stage offer the reduced number of features which best express the image. It brings several advantages such as speeding up the computing algorithms' execution time which depends on the dimensions of working space and better approximation to the solution in original high dimensional space than low dimensional space. Large-scale high-resolution images have encountered many difficulties to deal with as the higher resolution of the image is, the larger its data volume is. It is observed that images of larger sizes would require a large number of computations. As a consequence, compression of images has become an essential requirement in various applications related with images. Images, in general, will consist of redundant data which means duplication of data in these images. It may be either repeating pixel across the image or pattern, which is more frequently repeated in image. Reduction of redundant data in images help to achieve saving in storage space of that images. Image compression occurs when one or more of redundant pixels in image are reduced or eliminated. The quantity of pixels used in image representation can also be reduced without extremely change visualization of image.

Image compression is typically an application of data compression which encodes the original image with reduced bits. Moreover, the main objective of image compression is to reduce redundancy pixels of images for enhancing image sharing, transmitting and storing. Actually, an image is removed or group together its certain parts in order to reduce its size as sending and uploading an uncompressed image can take a long time indeed. There are typically two kinds of image compression methods: lossless and lossy. In lossy compression, it compresses an image by encoding all information from the original one. If the image is decompressed again, it is not exact to the original one as a little or more information related to the image has been lost within the compression process. Compression of JPEG is a very distinct of example of the lossy technique. For lossless compression, is a method used to reduce the size of a file while maintaining the same quality as before it was compressed. The compressed image is same as to original image without being compressed. Compression of PNG and GIF are examples of the lossless technique. Lossy image compression has become essential as the compression ratio obtained by lossless image compression is sometimes unsatisfactory.

3.5.1 Human Face Detection using High-Resolution Images

Recently, a large collection of images and videos creates unstructured big data in almost 80 percent of public and business corporations. Collecting, transferring and processing or interpreting images for analytical systems are not as easy as structured data such as text and numbers. Generally, images may be either raster or vector images or each image data is organized into two-dimensional matrix pixel values. As each pixel consists of its respective RGB bits value, high-resolution image can also be referred as high-dimensional data space. The main issue or challenge is that how to process high-resolution or high-dimensional representation of image data in predictive big data analytics because these images require greater bits for storage and transmission. The fundamental of image compression is reducing the amount of data (bits) by removing redundant data from the contents of this image. The formal image compression method lossy removes redundant data from image content which will cause the information loss and visual quality degradation. However, another image compression method lossless maintains the quality of the original image [86].

Principal Component Analysis (PCA) can be applied as a digital image compression preserving with as much as possible of the original information content. Moreover, PCA which used in reducing of dimensions allows the identification of standards in data and their expression. When patterns are found, they can be compressed, therefore, their dimensions can be reduced without much loss of information. According to PCA can represent high-dimensional data into a lower dimensional form, it can be applied to compress an image from original high-dimensional representation (RGB values for each pixel) to lower dimensional representation. The benefit of applying PCA is that it can perform lossless information not only finding patterns in data but also compressing data by reducing the number of dimensions. Most of the images which are belonging to same category can differ in lighting conditions, noises and so on. Although there are some differences between them, they may present some patterns which can be referred as principal components extracted from original image data. These principal components can also be referred as eigenimages. An essential property of PCA for images is that any original high-dimensional images can be reconstructed by combining the eigenimages.

Reducing high dimensionality of the feature space to the smaller dimensionality of feature space is the main idea of applying PCA for face recognition and prediction. Eigenspace projection can be performed by constructing large-scale 1-D vector of pixels from 2-D image to get principal components (PCs) of the feature space. This eigenspace is computed by identifying eigenvectors of the covariance matrix which is derived from a large collection of images (vectors). Eigenvectors which are applied in face recognition and prediction can be defined as eigenfaces. Individual eigenface can also be assumed as a candidate feature in prediction process. In addition, PCA aims to catch the total variation in the training sets of faces, and to explain this variation by a few variables. In fact, observation represented by a few variables as much as possible is easier to understand than it is represented by a large number of variables. The dimensionality reduction is very essential step when many faces are needed to recognize in various applications.

Generally, the steps which are followed in a PCA of a digital image are as follows:

- Correcting Image by the Mean

In a digital image, the variables X_1, X_2, X_p normally are the columns of that image. The PCA process is initiated by correcting the image to its columns have zero means and unitary variances with the following Equation 3.4,

$$\text{image corrected by the mean} = \text{image} - \text{mean of the image} \qquad (3.4)$$

- Computing Covariance Matrix

The covariance matrix C is computed with the following Equation 3.5,

$$\text{covImage} = \text{image corrected by the mean} \times (\text{image corrected by the mean})^{T} \qquad (3.5)$$

- Calculating Eigenvalues and Eigenvectors

The eigenvalues and the corresponding eigenvectors are calculated.

- Finding Principal Components

A matrix with vectors containing the list of eigenvectors is obtained. These vectors are columns of the covariance matrix. A covariance matrix which consists of all eigenvectors or components can be expressed as final data matrix.

- Reconstructing Original Image

Finally, the original image can be obtained from final data matrix where any components explain only a small portion of variation in data for image are removed. The removals can have the effect of reducing the quantity and characteristics of eigenvectors which can result final image data with a smaller or lower dimension.

3.5.2 Generation of the Eigefaces Process

The face is deserved as an essential focus of attention in social science. According to studies and research, automatic face detection is a complex problem in image processing. The detection of face in an image performs as preliminary step for face recognition. We can realize that how to locate or determine the position of face in image by detection process. In general, every face image can be seen as a vector. If the width and height of an image are "w" and "h" pixels respectively, the number of the components of this vector is w * h. Therefore, the dimension of the whole image or feature space is w * h. However, the dimension of the face portion is less than the dimension of that whole image as all the pixels of the face are not relevant, and each pixel depends on its neighbors.

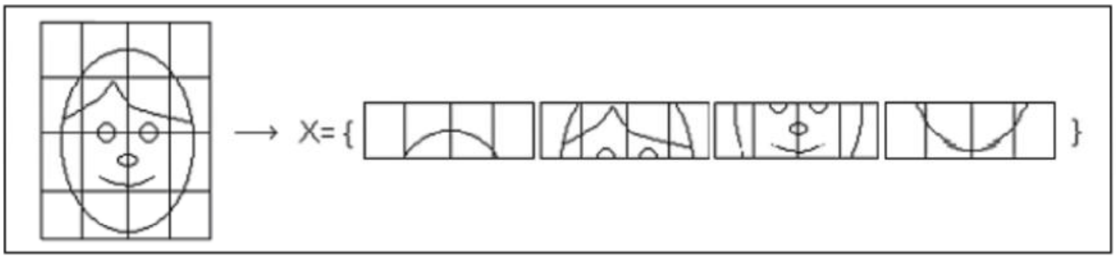


Figure 3.4 Formation of the face's vector from the face's image

Image space is typically a space of all images whose dimension is w by h pixels. The fundamental image space is composed of the following vectors in Figure 3.5.

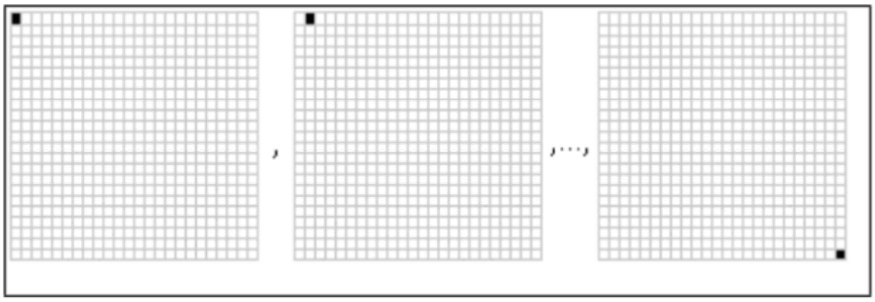


Figure 3.5 Fundamental image space

Generally, all faces which belong to two eyes, a mouth, a nose, etc. are located at the same place. Thus, image space with full dimensions is not absolutely optimal space for face representation. Building the vectors of a face space, a better representation of the faces, using PCA are called principle components or eigenfaces. The number of eigenfaces is equal to number of training face images, however, faces can also be estimated choosing the best eigenfaces which have the largest eigenvalues. The largest eigenvalues represent the most variance values among the set of face images. According to this computational efficiency, faces can represent each face image set in terms of eigenfaces. The eigenface approach is normally based on information theory and then face recognition also based on small set of image features that best approximate the set of known face image.

Decomposing all set of face images into a small set of characteristics called the principle components or "eigenfaces". Finding principle components or eigenvectors of the covariance matrix of the set of face images can be assumed as the set of features which characterize the variation between face images. Of course, it is a process of extracting relevant information in a face image, catching up the variations in a collection of the face images. Applying PCA to reduce the dimensions or space of images can describe the typical models with new set of dimensions or space. For example, a model with a set of training faces, there are components for this face space will be uncorrelated and maximized the variance of that original space. Principle component analysis which intends to explain the total variation in training faces set by a few principal components or variables. In fact, an observation which composed of a few principal components or variables is easier to understand than it is represented by a large number of variables.

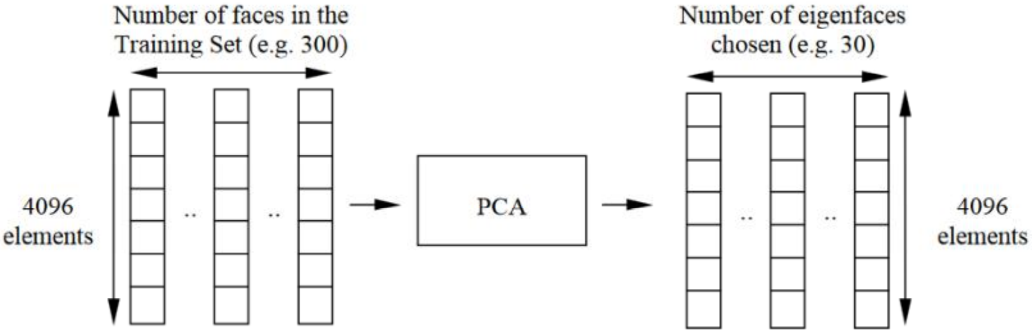

Number of faces in the Training Set (e.g. 300)
Number of eigenfaces chosen (e.g. 30)
4096 elements
PCA
4096 elements

Figure 3.6 Eigenfaces generation process

3.6 Unstructured, Large-Scale Text Documents

Nowadays, the extremely large amount of data is available for analysis and management in every day according to advances in data collection and storage capabilities. While more and more text documents are collected, it becomes a big challenge to understand hidden patterns in the data. In addition, processing unstructured massive amount of text data has become a challenging task to emphasize the most appropriate insight due to high-dimensionality. To analyze with statistical methods and algorithms, it also become a problem as text data is not in numerical format. Typically, text mining has involved the datasets consisting of a large number of terms to find out interesting patterns in text data. The redundant and infrequent terms could affect not only the accuracy of analysis results but also the complexity of the analysis. In general, the number of features or attributes or variables which are associated with each observation can be regarded as dimension of data. For large-scale text data, dimension reduction is applied in reducing the frequent terms or features from the text dataset.

3.6.1 Dimensionality Reduction in Text Documents

In current time, high-dimensional text documents are absolutely required to utilize feature reduction to transform high-dimensional into low dimensional representation in feature space. It is a process of finding an optimal transformation matrix corresponding the input high-dimensional document feature matrix. It is also a preliminary process for text mining applications by eliminating less significant features or terms. Text document or document dimensionality can be defined as the number of features or terms exist in that document. When the dimensionality of data becomes high, processing in a high-dimensional space cannot be straightforward for further analytical tasks. It creates a problem called "Curse of Dimensionality". The degree of complexity in analyzing documents is risen by higher dimensionality of text documents. Many researchers have been explored high-dimensional datasets which present new issues and challenges in statistical data analysis.

Moreover, traditional statistical methods also present issues and challenges as not only increasing in the number of observations but also increasing in the number of variables associated with each observation. Although, high-dimensional data have many challenges and issues in data analytics era, it offers some opportunities to emerge new theoretical developments. To construct optimal predictive models with better accuracy, it is essential to reduce the dimension of the original high-dimensional data prior to any modelling of that data. In addition, reduction of high dimensionality addresses not only the problem of inefficient computation but also the difficulty of detecting and exploiting the relationships between the terms. Having a lot of features or terms will face many difficulties such as over fitting, accuracy degradation and time-consuming. Therefore, dimensionality or feature reduction totally reduce the feature matrix size which can improve efficiency and accuracy of the corresponding classifiers and predictors. The main idea of extracting the relevant features is to reduce the features by making new combinations of features or terms (words).

3.6.2 Text Documents' Dimensionality Reduction Techniques

There are different dimensionality reduction techniques for unstructured text documents. They are:

- Document Frequency (DF)

  DF method reduces the size of vocabulary with a term selection which is based on assigning weights to terms. And, DF provides that an important term which appears frequently in documents under the same category.

- Category Frequency - Document Frequency (CF-DF):

  In general, terms which appear in most of the documents have the high frequency outside-group in DF method. Thus, CF-DF solves this problem by introducing a new quantity called Category Frequency (CF). To determine the Category Frequency of a term, learning documents are grouped according to categorization information.

- TF*IDF

  The main purpose of minimizing the loss of information for finding the best measure is to select discriminatory terms. And, the best measure of term importance in a document corpus is: the product of Term Frequency (TF) and the inverse of number of documents including this term (IDF: Inverse of Document Frequency). The IDF factor is computed by the following function

  $$\mathrm{IDF_i} = \log\frac{\mathrm{N}}{\mathrm{n}} \tag{3.6}$$

  where N is the total number of documents, n is the number of documents including the term i. In fact, a term that appears in a low appearance frequency has a high IDF factor. Besides, a term with a high has a high TF factor. Therefore, the terms which are concentrated in a low number of documents have a high TF∗IDF weights. To reduce the dimension of document corpus, a threshold is defined "t" lower than the total number of terms, and only the t terms having the highest TF∗IDF weights are chosen.

- Latent Semantic Analysis (LSA)

  Finding a lower rank matrix which is initiated from a matrix including the occurrences of terms in documents is the main purpose of LSA. Reducing the rank of occurrences of matrix has the issue of combining some irrelevant dimensions. It can be done by merging terms with similar meaning. This method consists of three main steps and the corpus is steadily converted into a vector space of a large number of dimensions. In fact, the initial size of the occurrences of matrix is (t∗d), where the columns “d” refers to textual units (paragraphs, sentences, texts, etc.) and the lines “t” refers to lexical units (words). In second step, singular value decomposition (SVD) is applied to this matrix. In decomposing the matrix X (t∗d) into a product of three matrices U, V and D:

$$X = UVD^T \tag{3.7}$$

where U (t∗r) is an orthogonal matrix, V (r∗d) is an orthogonal matrix, and D (r∗r) is a diagonal matrix and the values in this matrix represent dimensions in the vector space sorting in ascending order. The main purpose of LSA is to remove the very low singular values from the diagonal matrix, in order to transform the vector space from r dimensions to k dimensions the most representative occurrences, where (k << r).

- Principal Component Analysis (PCA)

  PCA, a statistical approach of dimension reduction, is applicable for a large varieties of data reduction. It is targeted to reduce the loss into the variance of original data. To reduce the dimension of an input data matrix by using PCA, it can be represented by a matrix M (d ∗ t) including the weights of terms in documents. Then, PCA is applied on the matrix M, by setting “s” the number of output vectors (s is less than the number of departure terms t). These “s” vectors represent the principal components of the original data where each vector consists of respective values.

Processing unstructured massive amount of text data has become a challenging task to emphasize the most appropriate insight from it. Many pre-processing methods and algorithms are required to extract useful features from huge amount of data. Feature reduction method using PCA is conducted as an initial step towards reducing the number of attributes without losing the main purpose and objective information from the original data.

3.7 Chapter Summary

In this chapter, the theoretical background for the proposed system is presented with detailed explanations. The goal of the system is the reduction of extremely high-dimensions of large-scale data to increase the prediction accuracy of the regression model. The explanation of high-dimensional data analytics using Principal Component Analysis (PCA) and correlation-based measures using Pearson Correlation Coefficient (PCC) is described section by section. Moreover, the information of applied data sources such as geospatial OpenStreetMap data, high-resolution or high-dimensional images and large-scale text documents are presented in detail.

CHAPTER 4

THE PROPOSED SYSTEM ARCHITECTURE

This chapter presents a whole predictive analytics system on high-dimensional big data using three different case studies with the proposed methods and theories. The primary flow of the proposed system can be seen in Figure 4.1.

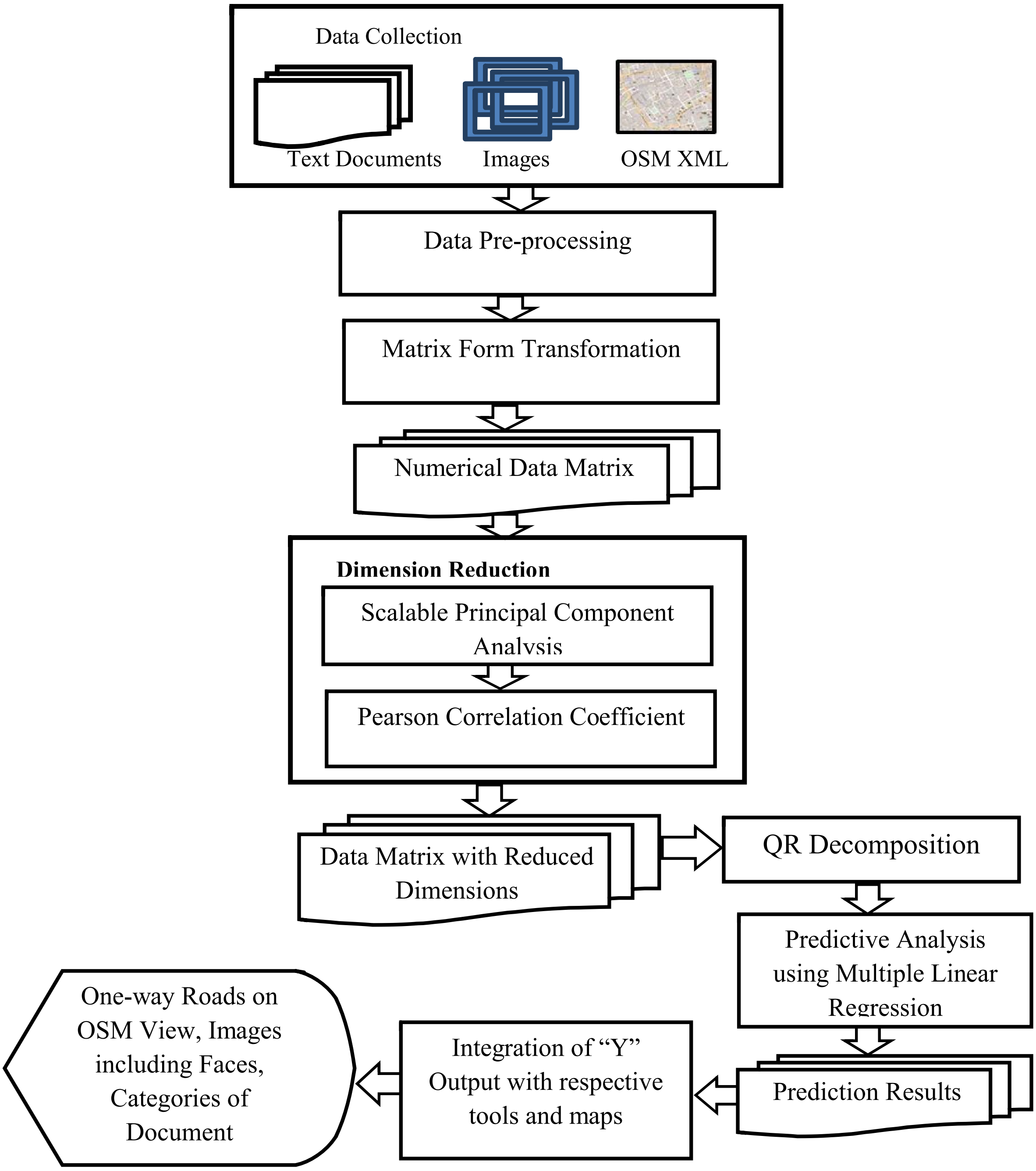

Data Collection
Text Documents
Images
OSM XML
Data Pre-processing
Matrix Form Transformation
Numerical Data Matrix
Dimension Reduction
Scalable Principal Component Analysis
Pearson Correlation Coefficient
Data Matrix with Reduced Dimensions
QR Decomposition
Predictive Analysis using Multiple Linear Regression
Prediction Results
Integration of "Y" Output with respective tools and maps
One-way Roads on OSM View, Images including Faces, Categories of Document

Figure 4.1 System Flow of the Proposed System

There are mainly five portions to explain the system are as follows:

- Data collection of three diverse high-dimensional data sources
- Data pre-processing and transforming into matrix data for each data source
- Dimension reduction using Two-Stage Approach
- Applying “QR” Decomposition to provide the efficient processing of MLR model
- Evaluation performance of the system with different case studies

4.1 Data Collection

There are three different sources of data for collecting high-dimensional data in this system. The first source of data is geodata or geospatial data with “XML” format which is called “OSM XML” data extracted from “OpenStreetMap (OSM)” for location “Myanmar”. The second data source is high-resolution or high-dimensional images obtained from “MS-Celeb-A”. The last data source is unstructured text documents which are collected from “UCI-Machine Learning Respiratory”.

Table 4.1 Information for Applied Data Sources

No.	Name of Dataset	Data Size	Data Sources	Download Links
1.	OSM XML	78.8 GB and above	OpenStreetMap	http://www.download.geofabrik.de
2.	Images	1.41 GB (202599 images) and above (MS-Celeb-1M)	MS-Celeb-1M MS-Celeb-A	https://www.megapixels.cc/datasets/msceleb/ http://www.mmlab.ie.cuhk.edu.hk
3.	DeliciousMIL	30 GB and above	UCI Machine Learning Respiratory	https://archive.ics.uci.edu

4.2 Data Pre-processing and Matrix Form Transformation

For the proposed approach, high-dimensional big datasets are input datasets for predictive data analysis of the system. With the purpose of reducing high dimensionality of these datasets, firstly, the original raw data needs transforming into suitable matrix form for subsequent processing of the system. In this system, the input raw data from three different data sources are applied to show that these three kinds of data can be reduced high dimensionality by using the proposed approach. Therefore, matrix form pre-processing is a key step and there are also three ways of pre-processing for three diverse data nature. The first type of data is "OSM XML" data collected from OpenStreetMap (OSM) and it is actually intended to utilize "One-way Roads Prediction" for Myanmar. High-resolution or high-dimensional images, the second type of data, is applied for "Number of Faces Prediction" to detect the number of faces existing in images. Finally, unstructured text documents are used in "Number of Documents Prediction". To be efficiently applied in proposed dimensionality reduction approach, these all three sorts of datasets are converted as numerical form in accordance with Principal Component Analysis (PCA), a key method for dimension reduction in this system.

4.3 Parallel, Distributed and Scalable PCA for Large-Scale Data Matrix

The scalability bottlenecks are the main difficulty and obstacle for traditional machine learning algorithms when they are applied to large-scale data analysis. Current data mining and machine learning algorithms were designed for centralized processing working with small-scale data which can fit in the memory of a single machine. In big data era, there is a need to re-design traditional data mining and machine learning algorithms into distributed algorithms which can work well with the increasing volumes of data not only in size of samples but also in number of dimensions. In large-scale data analytics, we have realized that traditional statistical, data mining and machine learning techniques are responsible to fulfil two essential properties such as scalability and flexibility adaptable with parallel and distributed processing. Furthermore, traditional dimension reduction algorithms were also designed to work well with small-scale data and such algorithms are needed to transform into parallel, distributed and scalable version adaptable on distributed platforms. We have also investigated that many traditional PCA approaches are developed on small and moderate data or datasets. Besides, traditional PCA algorithms were designed to work in centralized architecture where the entire dataset can fit in the memory of one computing node. They have many issues and challenges to apply in large-scale data according to storage and computational barriers. PCA actually faces scalability problem when it is applied to large-scale data and it is not suitable for centralized processing as computing covariance matrix is very difficult and time-consuming procedure of the PCA algorithm. Therefore, scalable PCA (sPCA) on Memory-based Spark, a popular distributed platform, is developed in this system. In designing sPCA on distributed platforms, the following optimizations are considered. They are:

- Distributed Operations or Functions

  Operations which use large-scale matrices that cannot fit in the memory are computed in distributed manner. For example: X === >X_m (Loading the entire matrix X in computing mean-centered matrix) X_c === > Transpose (X_m) * X_m (Multiplying two matrices in computing Covariance Matrix)
- Minimizing Size of Intermediate Data

  Intermediate Data can slow down the distributed execution of any PCA algorithm, because it needs to be transferred to other nodes for processing to continue. Minimizing the size of the intermediate data by two operations:
  - Redundant Computation
  - Distributed Job Consolidation
- Efficient large-scale matrix multiplication

  Although parallel and distributed computing paradigm has facilitated the computational requirements, matrix-by-matrix multiplication may be very expensive operation in distributed setting. To overcome the inefficiency of matrix multiplication "A∗B", the computation of "A′ ∗ B" and $(A * B)_i = A_i *$ B if matrix B can entirely fit in memory are applied instead.

4.4 The Proposed Two-Stage Dimension Reduction Approach

Feature reduction or selection technique is that data contains some features that are either redundant or irrelevant, and can be removed without incurring much loss of information. It is used for four reasons:

- simplification of models
- shorter training time,
- avoiding the curse of dimensionality, and

- enhanced generalization by reducing over fitting

4.4.1 Traditional Feature Selection in Multiple Linear Regression Model

The most frequent issue in data mining and machine learning for regression model is that how to predict the outcome of a dependent variable when there are a large number of independent variables in the model. With the advanced technologies and modern algorithms for regression model, it is a difficult situation to handle all variables at once for the model. There are many reasons to choose the subset of features or independent variables:

(a) As for prediction, there may be very expensive and time-consuming job to collect all features which may be not sure to be included or not in the model.

(b) Fewer variables may enable more precise and accurate prediction in most of the applications.

(c) In data analytics, if the model has a few input parameters, there may be more chance to obtain insights from the data.

(d) In regression model, the important one is to get the stable state of estimates of regression coefficients because many variables with model may cause multicollinearity problem.

(e) Independent variables which are uncorrelated with the dependent variable will increase the variance of the predictions.

Multiple Linear Regression (MLR) is a statistical model which is intended for estimating the relationship between a dependent variable “Y” and one or more explanatory variables (or independent variables) “X” to estimate the unknown regression model’s parameter “β”. The dataset consists of “n” rows of observations which offers Y_i, X_{i1}, X_{i2},…..,X_{ip}; i = 1,2,….,n. The estimates for “β” values are calculated with the purpose of minimizing the sum of squares of differences between the predicted values and observed values. Moreover, MLR specifies how much dependency or connection exist between “Y” and one or more “Xs”. In general, several independent variables or “Xs” for dependent variable “Y” can be some bias which is very likely to reduce RMSE, a performance indicator of MLR. Therefore, independent variables or “Xs” which may affect MLR’s predictive power should be dropped or removed from the model in the analysis.

In predictive big data analytics, selecting subset of features or attributes or dimensions from high-dimensional massive datasets has become a big issue to improve model's predictive power because it is a difficult computational problem to deal with extremely high-dimensions. In general, feature selection is to choose the best subset of features or features for the intended model. For multiple linear regression model, predictors or independent variables can be assumed as features for the model. There are a number of reasons why feature selection becomes an essential role in constructing the optimal regression model. Firstly, redundant predictors can hinder the regression analysis while we are trying to explain data in the simplest way. And, insignificant predictors are highly potential to increase noises and biases for the model. In addition, a large number of predictors will also cause a problem called "Multicollinearity". Finally, if we apply the model with redundant predictors for prediction purpose, it will be time-consuming and high expensive job indeed.

There are three common procedures in traditional feature selection of multiple linear regression model:

- Forward Selection
- Backward Elimination
- Step-wise Regression

In forward selection, firstly starting the model with no feature or predictor state. Checking the corresponding threshold value of individual feature or predictor to decide whether or not it can be added to the model. The selection process will continue until no new feature or predictor can be added for the model. Adding the predictors one at a time to construct a model may be a reasonable procedure for small and moderate features or predictors in datasets, however, for high-dimensional features or predictors it will be complicated and time-consuming procedure. Backward elimination, the simplest selection procedure, can be easily implemented for feature selection. Starting with all features or predictors to the model and then checking the corresponding threshold value of individual feature or predictor to decide whether or not it can be removed from the model. The removing process will continue until no new feature or predictor can be removed from the model. Although removing the predictors one at a time is a good procedure for the regression model, loading all features or predictors at once to the model is not convenient in high-dimensional datasets. The combination of two procedures backward elimination and forward selection called stepwise regression solve the problem of adding or removing predictors early in the process. However, it gives some drawbacks as there are several variations in the regression process.

Adding or removing predictor one-at-a-time may be less opportunity to become an optimal regression model. Sometimes, the removal of insignificant predictors affects the remaining predictors to overstate the most significant ones for the model. As a result, regression model may fit well in training data, but it may not fit in out-of-training data which lead to over fitting and create a false confidence in the final model. Therefore, many big data researchers point out the larger number of predictors is useful in selection process of stepwise regression, but the reality is that it does not solve the problem of too many predictors in massive datasets.

4.4.2 The Proposed Two-Stage Dimension Reduction Approach

In the era of big data, these three feature selection procedures no longer work when dealing with high-dimensional voluminous data nature. In this research, two-stage feature or dimension reduction approach is proposed for Multiple Linear Regression (MLR) model. Firstly, sPCA is applied to features (dimensions) comprising in matrix form data by reducing dimensions to avoid multicollinearity problem in regression model. Then, the correlation between a feature and the target class (predicted output) is also considered for the MLR model using Pearson Correlation Coefficient (PCC). In general, if the correlation measure between a feature and response variable is high enough, this feature will be relevant to include as a predictor in the prediction process.

In fact, dimension reduction or feature selection tends to raise the statistical significance of predictors or variables of the MLR model. The main purpose of the proposed two-stage dimension reduction approach is to prove that applying sPCA for high-dimensions of big datasets can reduce less significant or sometimes noisy features for the MLR model, however, there still remain an issue to find out the reduced feature subset or features resulted from sPCA stage are also correlated or not with the dependent or output variable of MLR model. After applying sPCA, the first dimensionality reduction stage of the system, the dataset with reduced dimensions or features will be input into next dimension reduction stage to sort out how many features remain for the prediction process of the model. The ultimate goal of MLR model is to filter efficiently the most important and correlated features with dependent variable of the model to provide better prediction accuracy indeed. By applying the proposed two-stage approach, the most important predictors or variables are chosen with the purpose of not including irrelevant and redundant variables for the MLR model.

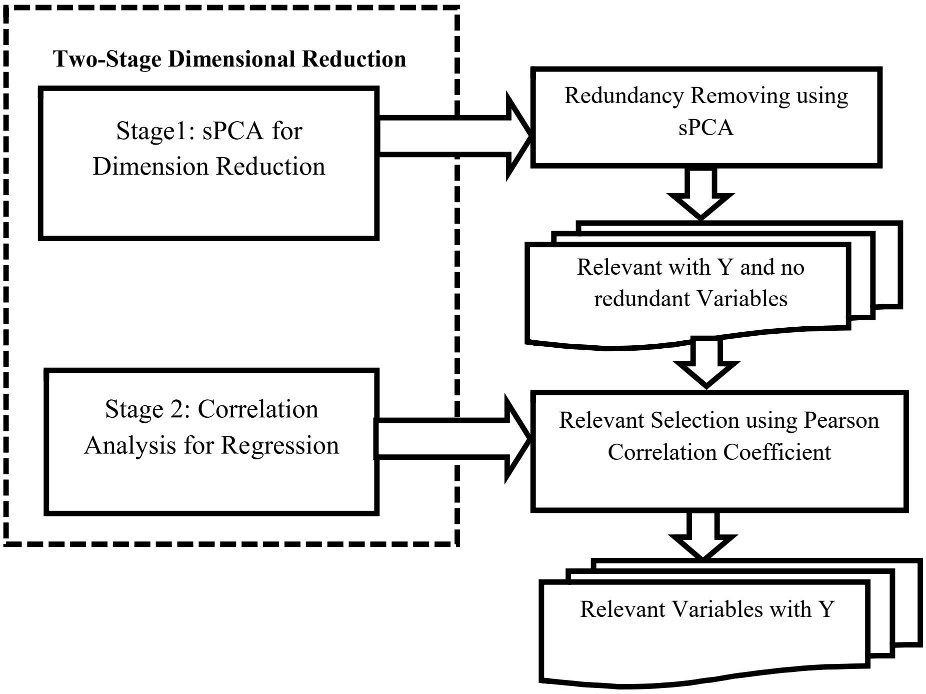

Two-Stage Dimensional Reduction
Stage1: sPCA for Dimension Reduction
Stage 2: Correlation Analysis for Regression
Redundancy Removing using sPCA
Relevant with Y and no redundant Variables
Relevant Selection using Pearson Correlation Coefficient
Relevant Variables with Y

Figure 4.2 The Proposed Two-Stage Dimension Reduction Approach

According to the procedure as shown in Table 4.2, there are several input dimensions or independent variables “Xs” for traditional MLR model. Adding all independent variables “Xs” at once to construct a model may be reasonable for small and moderate dimensions in datasets, however, it will be complicated and time-consuming procedure for high-dimensional data nature [4]. In general, several independent variables “Xs” for dependent variable “Y” can be some bias which is very likely to reduce RMSE, a performance indicator of MLR. Therefore, independent variables “Xs” which may affect MLR’s predictive power should be dropped or removed from the model in the analysis.

Table 4.2 Procedure for Traditional Multiple Linear Regression

Procedure for Traditional Multiple Linear Regression

Input: m x n data matrix “X”

Output: Coefficient “β” values of Regression Model

Steps

1. Define dependent variable “Y” and independent variables “X” for the model
2. Find “β” values from the equation
3. $Y= \beta_0+\beta_1X_1+\ldots\ldots\ldots\ldots+\beta_nX_n$
4. Calculate Residual, R2 and RMSE for model performance

The procedure for improved version of MLR model using two-stage approach is shown in Table 4.3. According to the procedure, the high-dimensional large-scale input data matrix “X” with original “n” dimensions is processed by the proposed sPCA. The reconstructed matrix “X” with reduced number of dimensions “nk” is obtained from sPCA stage. Then, the correlation procedures by PCC is applied on this reconstructed matrix to divide mainly three categories of correlation such as positive, negative and no correlation. Dropping the irrelevant features according to the PCC’s procedures and then reconstruct the matrix “X” with more reduced number of dimensions “np” which are resulted from PCC stage. Finally, the reconstructed matrix “X” with reduced number of dimensions “np” is used to find out the “β” values of the MLR model.

Table 4.3 Procedure for Improved Multiple Linear Regression

Procedure for Improved Multiple Linear Regression

Input: m x n data matrix “X” (“n” dimensions)

Output: Coefficient “β” values of Regression Model

Steps

1. Define dependent variable “Y” and independent variables “X” for the model
2. Apply sPCA on raw, high-dimensional matrix X_{mxn}
3. Compute eigenvalues and eigenvectors of X_{mxn}
4. Choose top “k” PCs by ranking the eigenvalues from eigenvectors in descending order
5. Construct the matrix X_{mxn} using “k” eigenvectors into X_{mxnk}
6. Reconstruct the matrix X_{mxnk} into original input matrix form with reduced “n” to “nk” dimensions
7. Apply Correlation Procedures on transformed matrix X_{mxnk}
8. Divide the features or attributes x_i in matrix X_{mxnk} into positive, perfect positive, negative, perfect negative, no correlation categories
9. Rank the features x_i in descending order to select top “p” x_i
10. Remove or drop no correlation and lower ranked x_i
11. Reconstruct the matrix X_{mxnp} into original input matrix form with reduced “nk” to “np” dimensions
12. Find “β” values from the equation by using matrix X_{mxnp}
13. $Y= \beta_0+\beta_1X_1+\ldots\ldots\ldots\ldots+\beta_nX_n$
14. Calculate Residual, R^2 and RMSE for model performance

The detailed procedure for finding correlation measures between a pair of variables (X, Y) by PCC is shown in Table 4.4.

Table 4.4 Procedure for finding correlation measures between a pair of variables (X, Y)

Procedure for Finding Correlation Measures between a pair of Variables (X, Y)

```
Input: $X_1$, $X_2$, ……, $X_n$ and $Y_i$ // Independent variables and dependent
variable resulted    from the first stage, sPCA
Output: SList // List for Selected Independent Variables
Steps
 Begin
    for i =1 to n do
        Begin
        Calculate rXi,Y for each Xi
        if (rXi,Y  = =0)
        Append Xi to Listno_corre; // Drop for the model
        if (rXi,Y  = = -1 || rXi,Y  = = 1)
        Append Xi to Listper_corre;
        if (rXi,Y  > -1 && rXi,Y  < 0)
        Append Xi to Listneg_corre;
        if (rXi,Y  < +1 && rXi,Y  > 0)
        Append Xi to Listpos_corre;
        End
        Order Listneg_corre in descending rXi,Y  value;
        Order Listpos_corre in descending rXi,Y  value;
        Merge Listper_corre, Listneg_corre, Listpos_corre as SList;
 End
```

In this research, there are mainly three case studies which are developed and experimented on the proposed system. They are:

- One-way Roads Prediction using OSM Data
- Number of Faces Prediction using Large-scale Images
- Number of Documents Prediction using Unstructured Text Documents

4.5 Case Study 1: One-way Roads Prediction using OSM Data

We prefer to use OSM maps instead of Google Maps, because the latter can only be downloaded as raster images. OSM data, however, can be accessed and manipulated in vector format, each object type comes with Meta data and identifiers that allow straight-forward filtering. One-way roads and streets are usually used in high volume situations which occur in downtown areas with closely-spaced intersections. In Yangon, the former capital and now business city of Myanmar, roads and streets are often congested and people lose much time stuck in traffic every day. Peak hours are 8:00 to 9:00 in the morning, 14:00 to 16:00 in the evening and after working hours. Sometimes, a ten-minute trip could take as long as 2 hours because of severe traffic situation during peak hours. Although one-way roads and streets can cause some disadvantages such as increased travel distance, wider pedestrian crossings, and driver confusion, it can offer some important advantages such as enhance traffic capacity and increase safety. Not only providing additional lanes and reducing number and severity of crashes by eliminating head-on crashes to be efficient in traffic control operation and increased safety. The main purpose of implementing this system will predict one-way roads in major business city, Yangon using OSM data as a way to facilitate the traffic problems. The four main types of roads and streets in Yangon City especially in busy and crowded downtown area are as follows:

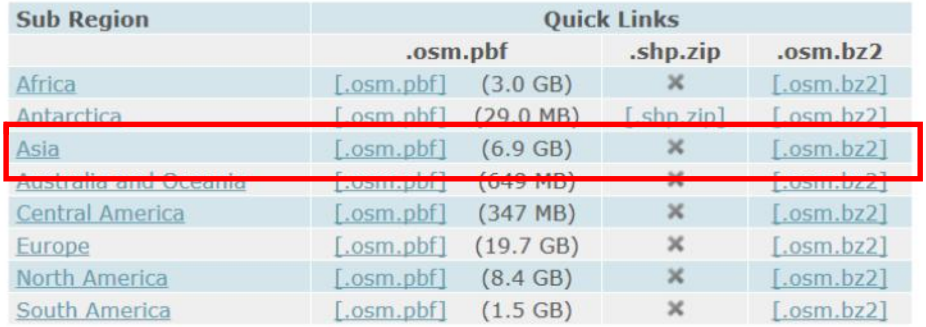
- The broad roads which are running west to east
- The broad roads which are running south to north
- The narrow streets which are running south to north
- The wide streets which are running south to north

Although one-way roads and streets can cause some disadvantages such as increase in travel distance, wider pedestrian crossings, and driver confusion, it can offer some important advantages such as enhance traffic capacity and increase safety.

Not only providing additional lanes and reducing number and severity of crashes by eliminating head-on crashes to be efficient in traffic control operation and increased safety. The main purpose of implementing this system will predict one-way roads in major business city, Yangon using OSM data as a way to facilitate the traffic problems.

4.5.1 OpenStreetMap (OSM) Data Extract

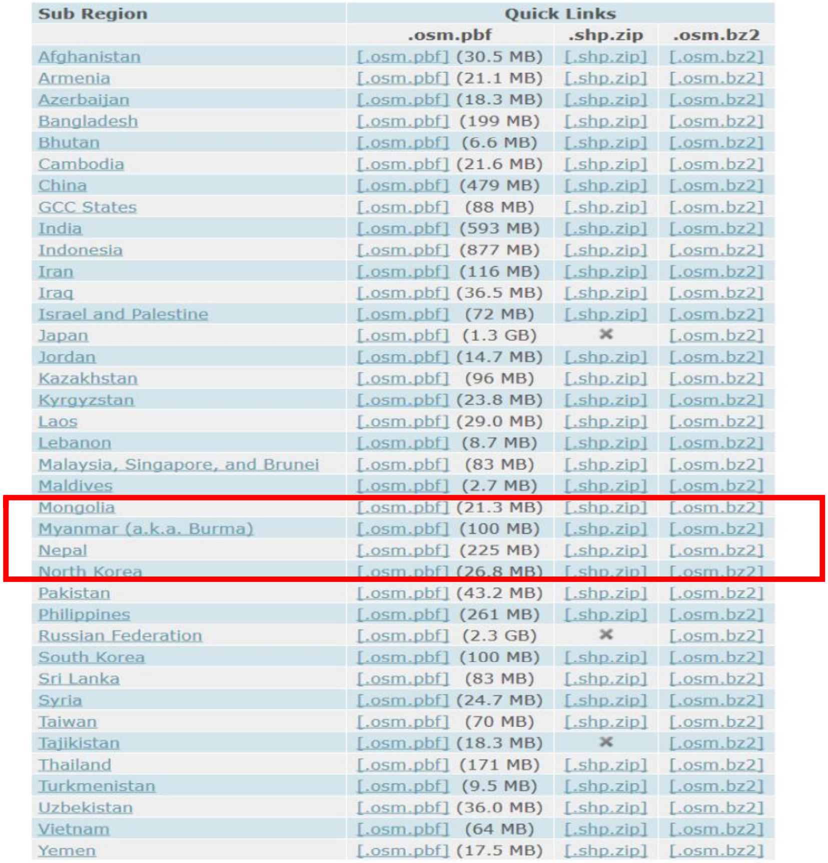
To extract “OSM XML” data from raw data (OSM source data), it is downloaded from Geofabrik's free download server which has data extracts from the OpenStreetMap project updated every day manner. The files which are ending in “. osm.bz2" are typically bzip2 compressed OSM raw data files. By selecting desire continent and more specifically, the country of interest from the links is available. Firstly, we can select the region name to download the one of the file extensions links for quick access.



Sub Region	Quick Links			
	.osm.pbf		.shp.zip	.osm.bz2
Africa	[.osm.pbf]	(3.0 GB)	✖	[.osm.bz2]
Antarctica	[.osm.pbf]	(29.0 MB)	[.shp.zip]	[.osm.bz2]
Asia	[.osm.pbf]	(6.9 GB)	✖	[.osm.bz2]
Australia and Oceania	[.osm.pbf]	(649 MB)	✖	[.osm.bz2]
Central America	[.osm.pbf]	(347 MB)	✖	[.osm.bz2]
Europe	[.osm.pbf]	(19.7 GB)	✖	[.osm.bz2]
North America	[.osm.pbf]	(8.4 GB)	✖	[.osm.bz2]
South America	[.osm.pbf]	(1.5 GB)	✖	[.osm.bz2]

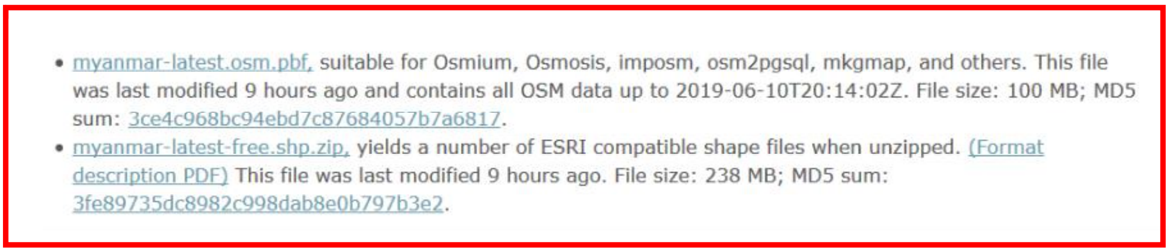
Figure 4.3 Choosing desire region for OSM data

The sub region name can be chosen to download the one of the file extensions links for quick access.



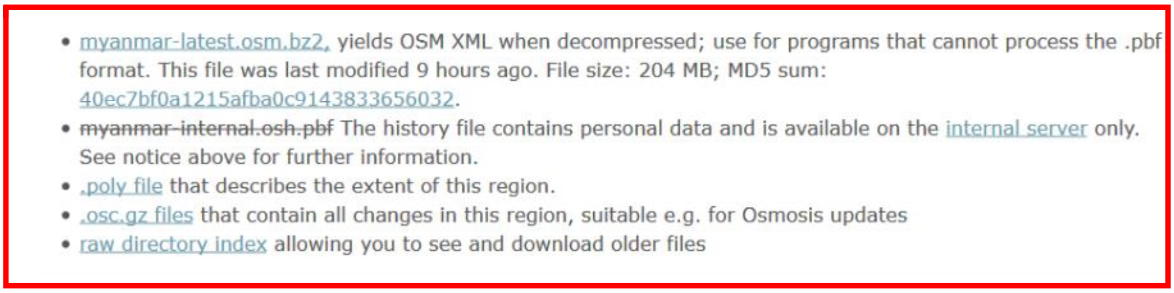
Sub Region	Quick Links			
	.osm.pbf		.shp.zip	.osm.bz2
Afghanistan	[.osm.pbf]	(30.5 MB)	[.shp.zip]	[.osm.bz2]
Armenia	[.osm.pbf]	(21.1 MB)	[.shp.zip]	[.osm.bz2]
Azerbaijan	[.osm.pbf]	(18.3 MB)	[.shp.zip]	[.osm.bz2]
Bangladesh	[.osm.pbf]	(199 MB)	[.shp.zip]	[.osm.bz2]
Bhutan	[.osm.pbf]	(6.6 MB)	[.shp.zip]	[.osm.bz2]
Cambodia	[.osm.pbf]	(21.6 MB)	[.shp.zip]	[.osm.bz2]
China	[.osm.pbf]	(479 MB)	[.shp.zip]	[.osm.bz2]
GCC States	[.osm.pbf]	(88 MB)	[.shp.zip]	[.osm.bz2]
India	[.osm.pbf]	(593 MB)	[.shp.zip]	[.osm.bz2]
Indonesia	[.osm.pbf]	(877 MB)	[.shp.zip]	[.osm.bz2]
Iran	[.osm.pbf]	(116 MB)	[.shp.zip]	[.osm.bz2]
Iraq	[.osm.pbf]	(36.5 MB)	[.shp.zip]	[.osm.bz2]
Israel and Palestine	[.osm.pbf]	(72 MB)	[.shp.zip]	[.osm.bz2]
Japan	[.osm.pbf]	(1.3 GB)	✖	[.osm.bz2]
Jordan	[.osm.pbf]	(14.7 MB)	[.shp.zip]	[.osm.bz2]
Kazakhstan	[.osm.pbf]	(96 MB)	[.shp.zip]	[.osm.bz2]
Kyrgyzstan	[.osm.pbf]	(23.8 MB)	[.shp.zip]	[.osm.bz2]
Laos	[.osm.pbf]	(29.0 MB)	[.shp.zip]	[.osm.bz2]
Lebanon	[.osm.pbf]	(8.7 MB)	[.shp.zip]	[.osm.bz2]
Malaysia, Singapore, and Brunei	[.osm.pbf]	(83 MB)	[.shp.zip]	[.osm.bz2]
Maldives	[.osm.pbf]	(2.7 MB)	[.shp.zip]	[.osm.bz2]
Mongolia	[.osm.pbf]	(21.3 MB)	[.shp.zip]	[.osm.bz2]
Myanmar (a.k.a. Burma)	[.osm.pbf]	(100 MB)	[.shp.zip]	[.osm.bz2]
Nepal	[.osm.pbf]	(225 MB)	[.shp.zip]	[.osm.bz2]
North Korea	[.osm.pbf]	(26.8 MB)	[.shp.zip]	[.osm.bz2]
Pakistan	[.osm.pbf]	(43.2 MB)	[.shp.zip]	[.osm.bz2]
Philippines	[.osm.pbf]	(261 MB)	[.shp.zip]	[.osm.bz2]
Russian Federation	[.osm.pbf]	(2.3 GB)	✖	[.osm.bz2]
South Korea	[.osm.pbf]	(100 MB)	[.shp.zip]	[.osm.bz2]
Sri Lanka	[.osm.pbf]	(83 MB)	[.shp.zip]	[.osm.bz2]
Syria	[.osm.pbf]	(24.7 MB)	[.shp.zip]	[.osm.bz2]
Taiwan	[.osm.pbf]	(70 MB)	[.shp.zip]	[.osm.bz2]
Tajikistan	[.osm.pbf]	(18.3 MB)	✖	[.osm.bz2]
Thailand	[.osm.pbf]	(171 MB)	[.shp.zip]	[.osm.bz2]
Turkmenistan	[.osm.pbf]	(9.5 MB)	[.shp.zip]	[.osm.bz2]
Uzbekistan	[.osm.pbf]	(36.0 MB)	[.shp.zip]	[.osm.bz2]
Vietnam	[.osm.pbf]	(64 MB)	[.shp.zip]	[.osm.bz2]
Yemen	[.osm.pbf]	(17.5 MB)	[.shp.zip]	[.osm.bz2]

Figure 4.4 Choosing desire sub region for OSM data



- myanmar-latest.osm.pbf, suitable for Osmium, Osmosis, imposm, osm2pgsql, mkgmap, and others. This file was last modified 9 hours ago and contains all OSM data up to 2019-06-10T20:14:02Z. File size: 100 MB; MD5 sum: 3ce4c968bc94ebd7c87684057b7a6817.
- myanmar-latest-free.shp.zip, yields a number of ESRI compatible shape files when unzipped. (Format description PDF) This file was last modified 9 hours ago. File size: 238 MB; MD5 sum: 3fe89735dc8982c998dab8e0b797b3e2.

Figure 4.5 Downloadable commonly used formats for “Myanmar”



- myanmar-latest.osm.bz2, yields OSM XML when decompressed; use for programs that cannot process the .pbf format. This file was last modified 9 hours ago. File size: 204 MB; MD5 sum: 40ec7bf0a1215afba0c9143833656032.
- ~~myanmar-internal.osh.pbf~~ The history file contains personal data and is available on the internal server only. See notice above for further information.
- .poly file that describes the extent of this region.
- .osc.gz files that contain all changes in this region, suitable e.g. for Osmosis updates
- raw directory index allowing you to see and download older files

Figure 4.6 Downloadable other formats and auxiliary files for “Myanmar”

4.5.2 Pre-Processing of OSM XML

The step-by-step pre-processing of OSM XML data is shown in Figure 4.7. In general, OSM data exists in the form of data structures such as nodes, ways and relations. It is essential to transform raw, unstructured OSM XML data into suitable matrix format compatible with dimension reduction approach. OSM data (OSM XML) is firstly converted into GeoJSON files by using Osmosis. Osmosis is an actually powerful command-line tool for manipulating and processing raw OSM data or .osm files. It can be applied to process large-scale data files in splitting into manageable pieces. There are a variety of functions available with Osmosis. But, many of them are very complex and not easy to understand. Furthermore, GeoJSON, which can represent geodata as JSON, is intended to apply in encoding of various geographic data structures. In fact, a GeoJSON object can typically represent a region of space (a Geometry), a spatially bounded entity (a Feature), or a list of Features (a FeatureCollection). Features in GeoJSON includes a Geometry object, additional properties, and a FeatureCollection which consists of a list of Features. For geospatial data analysis in big data platforms, GeoJSON files are then converted into CSV files by using QGIS (Quantum GIS) which allows users to view, edit and analyze spatial information.

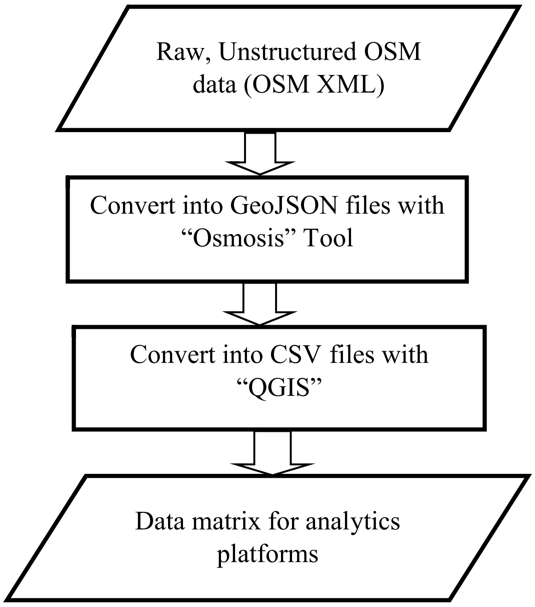

Raw, Unstructured OSM data (OSM XML)
Convert into GeoJSON files with "Osmosis" Tool
Convert into CSV files with "QGIS"
Data matrix for analytics platforms

Figure 4.7 OSM Data Pre-processing Steps

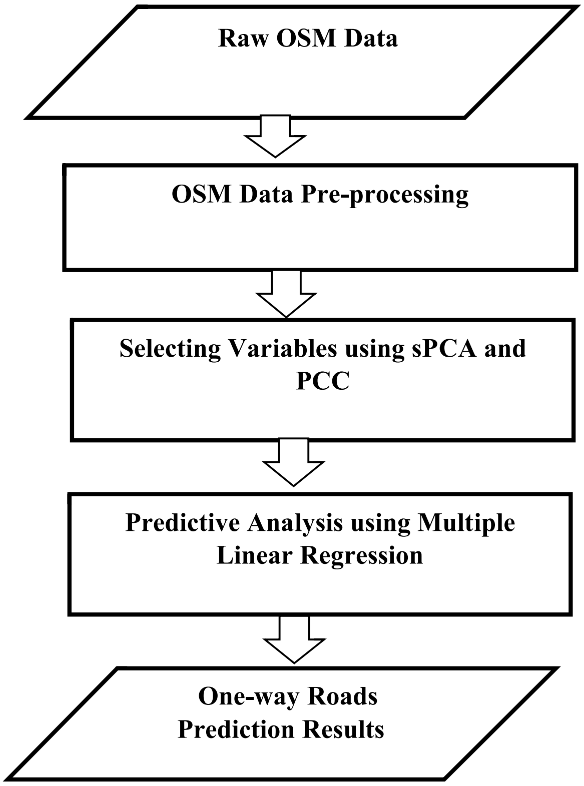

Raw OSM Data
OSM Data Pre-processing
Selecting Variables using sPCA and PCC
Predictive Analysis using Multiple Linear Regression
One-way Roads Prediction Results

Figure 4.8 One-way Roads Prediction Process

4.6 Case Study 2: Number of Faces Prediction using Large-Scale Images

The large volumes of images would require a large number of computations and compression of these images has become an essential requirement in various applications related with images. In this case study, we utilized images from MS-Celeb-A dataset to predict "Number of Faces" containing in these images. MS-Celeb-A dataset is a large-scale face attributes dataset which consists of more than 200K celebrity images and 202,599 number of face images.

4.6.1 Face Detection using Haar Feature-Based Cascade Classifier

Face detection can perform as a first and essential step for face recognition. It can be realized that it is the process of locating and determining the position of faces or objects in images by detection process. Face detection typically uses classifiers which are algorithms to detect whether it is a face or not in an image. For detecting faces, these classifiers have to be trained by using thousands to millions of images in order to obtain better accuracy. Generally, OpenCV uses two kinds of classifiers:

- LBP (Local Binary Pattern)

- Haar Cascades

In fact, Haar feature-based Cascade Classifier is based on Haar Wavelet technique to analyze pixels in the image into squares by function. It commonly uses machine learning models to obtain good accuracy from training data. By using training data, the classifier identifies features which can be considered as a face. Besides, it utilizes integral image concepts for computing features detected. Haar Cascades also apply the Adaboost machine learning algorithm which selects a small number of important features from a large set to offer an efficient detection result of classifiers. Firstly, the algorithm requires a large number of images with faces called "positive images" and images without faces called "negative images" to train the classifier. These images are then needed to extract features from them.

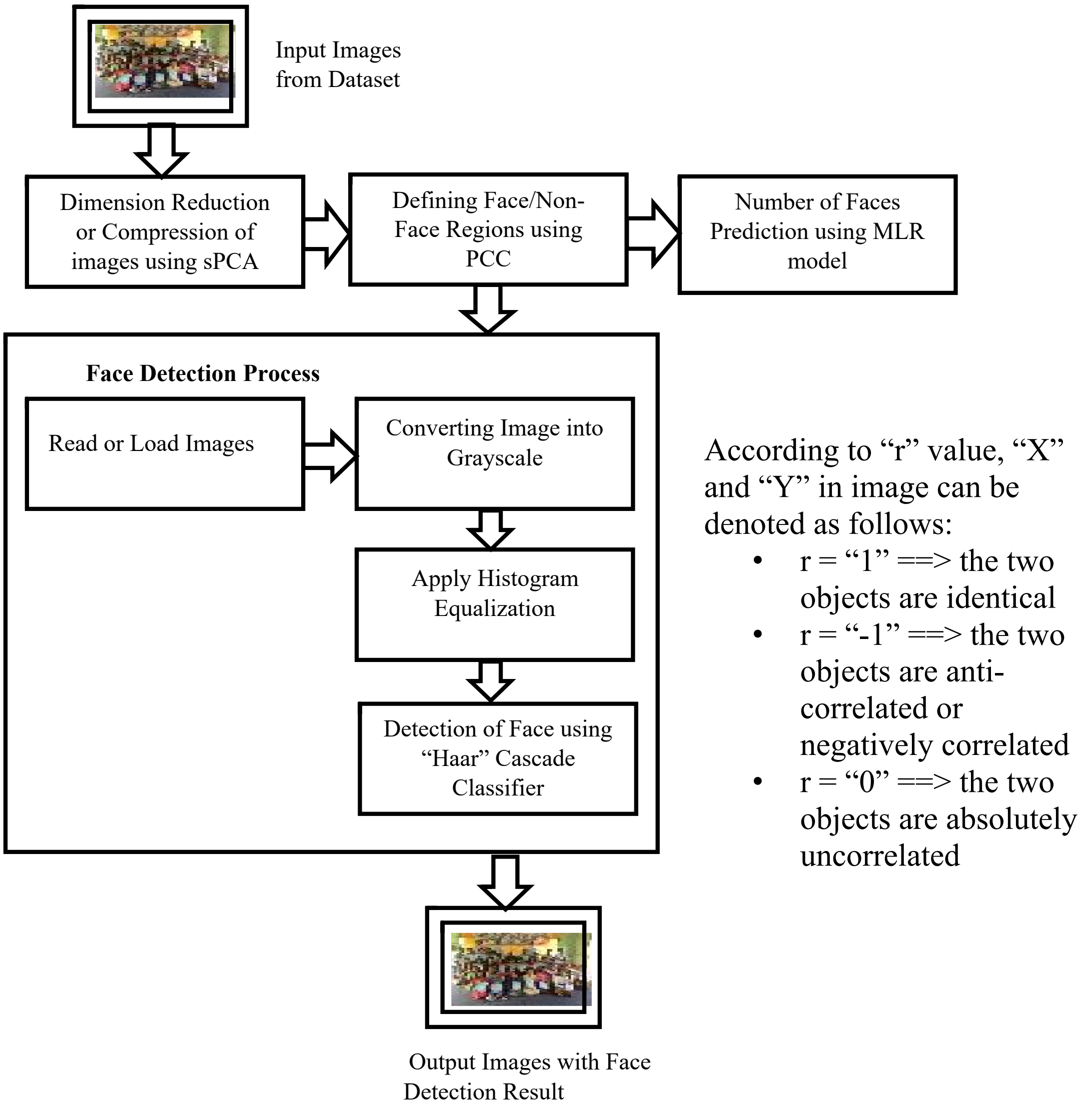
4.6.2 Number of Faces Prediction Process

Number of faces prediction process of the proposed system is shown in Figure 4.9. Firstly, large-scale input images are collected to be processed. sPCA is applied to compress large-scale images from original high-dimensional (RGB values for each pixel) to lower dimensional representation. And then, step by- step procedures of sPCA are applied for input image with the purpose of reducing high-dimensionality in that image. The compressed image which is resulted from sPCA procedures is also taken as input image to face detection process. Loading the compressed image and transforming it into grayscale are performed. Then, histogram equalization is utilized on that image. To accurate region defining process for faces, Pearson Correlation Coefficient (PCC) is applied to clearly identify face and non-face parts in image.

Pearson Correlation Coefficient (PCC) mostly applied to discover the linear relationship of correlation and dependence between input and output variables of the learning model. In this system, the correlation between target or output "face" and a variety of objects (face/non-face and others) existing in images is discovered by using PCC. In order to categorize the regions in images, we have to prepare training data which consists of a lot of samples of faces and non-faces and, other procedures for locating them. For defining regions or parts, there are three different categories:

- the "truly matched" parts
- the "anti-correlated" parts
- the "no correlation" parts

The face regions are assumed as "Positive" because these regions are truly matched parts of the target face. On the other hand, the other regions of the human body can also be assumed as "Negative" because these regions are not correlated or anti-correlated with the target face. Finally, the objects in images which are not human or persons are precisely assumed as "No Correlation" because these regions are not required to consider in face detection process. Therefore, the correlation procedure by PCC is contributed before detecting faces in images in order to increase correctness and preciseness in face detection. By applying the output of from PCC, OpenCV's Haar Cascade Classifier finally detects the faces in images for final detection results.


Input Images
from Dataset
Dimension Reduction or Compression of images using sPCA
Defining Face/Non-Face Regions using PCC
Number of Faces Prediction using MLR model
Face Detection Process
Read or Load Images
Converting Image into Grayscale
Apply Histogram Equalization
Detection of Face using "Haar" Cascade Classifier
Output Images with Face Detection Result

According to "r" value, "X" and "Y" in image can be denoted as follows:

- r = "1" ==> the two objects are identical
- r = "-1" ==> the two objects are anti-correlated or negatively correlated
- r = "0" ==> the two objects are absolutely uncorrelated

Figure 4.9 Number of Faces Prediction Process

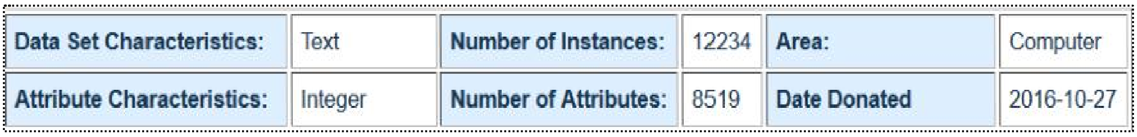
4.7 Case Study 3: Number of Documents Prediction using Unstructured Text Documents

The raw, unstructured text documents from “DeliciousMIL” dataset is applied as input text documents in this case study. It is intended to obtain “Number of Documents (Education, Science && Technology, Culture && History)” prediction results from the system.

4.7.1 DeliciousMIL Dataset

DeliciousMIL [92] dataset consists of a subset of tagged web pages from the social bookmarking site “delicious.com”. The original web pages were obtained from DeliciousT140 dataset which was collected from the delicious.com in June 2008. Users of the website delicious.com bookmarked each page with word tags. From this dataset, the text parts of each web page are extracted and then 20 common tags are selected as class labels or targets. These target classes are as follows: reference, design, programming, internet, computer, web, java, writing, English, grammar, style, language, books, education, philosophy, politics, religion, science, history, and culture. This dataset includes 12234 instances or text documents and 8519 attributes or terms in these text documents. It also provides ground-truth class labels to evaluate the performance of text document classification. In the following Table 4.5, the brief information about DeliciousMIL dataset is described.

Table 4.5 Information of DeliciousMIL Dataset



Data Set Characteristics:	Text	Number of Instances:	12234	Area:	Computer
Attribute Characteristics:	Integer	Number of Attributes:	8519	Date Donated	2016-10-27

4.7.2 Text Documents’ Categorization Process

In this approach, the raw unstructured text data from DeliciousMIL dataset is applied as input text documents. According to the structure and characteristics of dataset, the rows or observations can be assumed as text documents, and the attributes or features or dimensions can also be denoted as terms or many sets of characters which will form individual document. In fact, the initial raw state of text data may be unsuitable to be further processed or analyzed. Thus, the pre-processing procedures for text data is performed as follows:

- “HTML” Tags Removal
- Stopword Removal
- Stemming
- termtoNumerical Conversion

The unnecessary “HTML” codes or tags are removed to achieve the text parts from unstructured raw data. Then, the standard stopword removal and stemming procedures are applied. In order to use Principal Component Analysis (PCA), a dimension reduction approach, “termtoNumerical Conversion” procedure is inserted to transform the text value of each term into suitable numeric value. After applying the text pre-processing procedures, the text documents including terms with numeric values are used to reduce the dimensions or the number of terms which may be irrelevant and redundant. Although PCA enables to mitigate the existing a large number of dimensions or terms in dataset, there is a need to reduce the uncorrelated or unrelated terms for the predefined targets or classes. To achieve term and target correlation, in this approach, Pearson Correlation Coefficient (PCC) is utilized between two variables; one for “term” and other for “target”.

Furthermore, in order to find the pair-wise correlation between the terms and target in each document, “weight value assigning to each term” using TF*IDF is preliminary contributed to correlation approach. TF*IDF, an information retrieval technique, can be applied to weigh a term’s frequency (TF) and its inverse document frequency (IDF). In TF-IDF terms weighting, the text documents are modeled as transactions. In fact, TF*IDF algorithm is utilized with the purpose of weighing a term or word in any text content to assign the importance to that term or word which is based on the fact that the number of times it appears in the document. This process is known as term weighting and the product of the TF and IDF scores of a term is called the TF*IDF weight of that term. For a term t in a document d, the weight Wt,d of term t in document d is given by:

$$W_{t,d} = TF_{t,d} \log (N/DF_t) \tag{4.1}$$

TFt,d is the number of occurrences of t in document d

DFt is the number of documents containing the term t

N is the total number of documents in the corpus

After computing all weight values of terms in each document, then, we can find the correlation between these terms and three predefined classes or targets; “Science & Technology”, “Education”, and “History & Culture”. In finding correlation between term and target, for example, we would like to know the terms composing in Document 1 is correlated or relevant the most with one of three predefined classes. When we calculate the pairs of term and three predefined classes’ combinations, we will obtain all correlation values from all combinations. For instance, if there are three terms in each text document, we will get the possible pairs like this; (term1, classs1), (term1, class2), (term1, class3), (term2, class1), (term2, class2), (term2, class3), (term3, class1), (term3, class2), (term3, class3). In this way, the number of correlations or associations between each term and three predefined classes can be known. And then, the largest correlation value which possess the class will be the most relevant of that term. In this approach, after computing the correlation values between all terms and classes, we enable to assign the most relevant or correlated class for each document in the dataset.

Table 4.6 Example of Computing Terms/Targets and Weights

Terms/Targets	T1	T2	T3	C1	C2	C3
Weights	W1	W2	W3	W4	W5	W6

Table 4.7 Example of Correlation Computing between Term and Target

Terms/Features	Target/Class 1	Target/Class 2	Target/Class 3
T1	Corr (T1, C1)	Corr (T1, C2)	Corr (T1, C3)
T2	Corr (T2, C1)	Corr (T2, C2)	Corr (T2, C3)
T3	Corr (T3, C1)	Corr (T3, C2)	Corr (T3, C3)

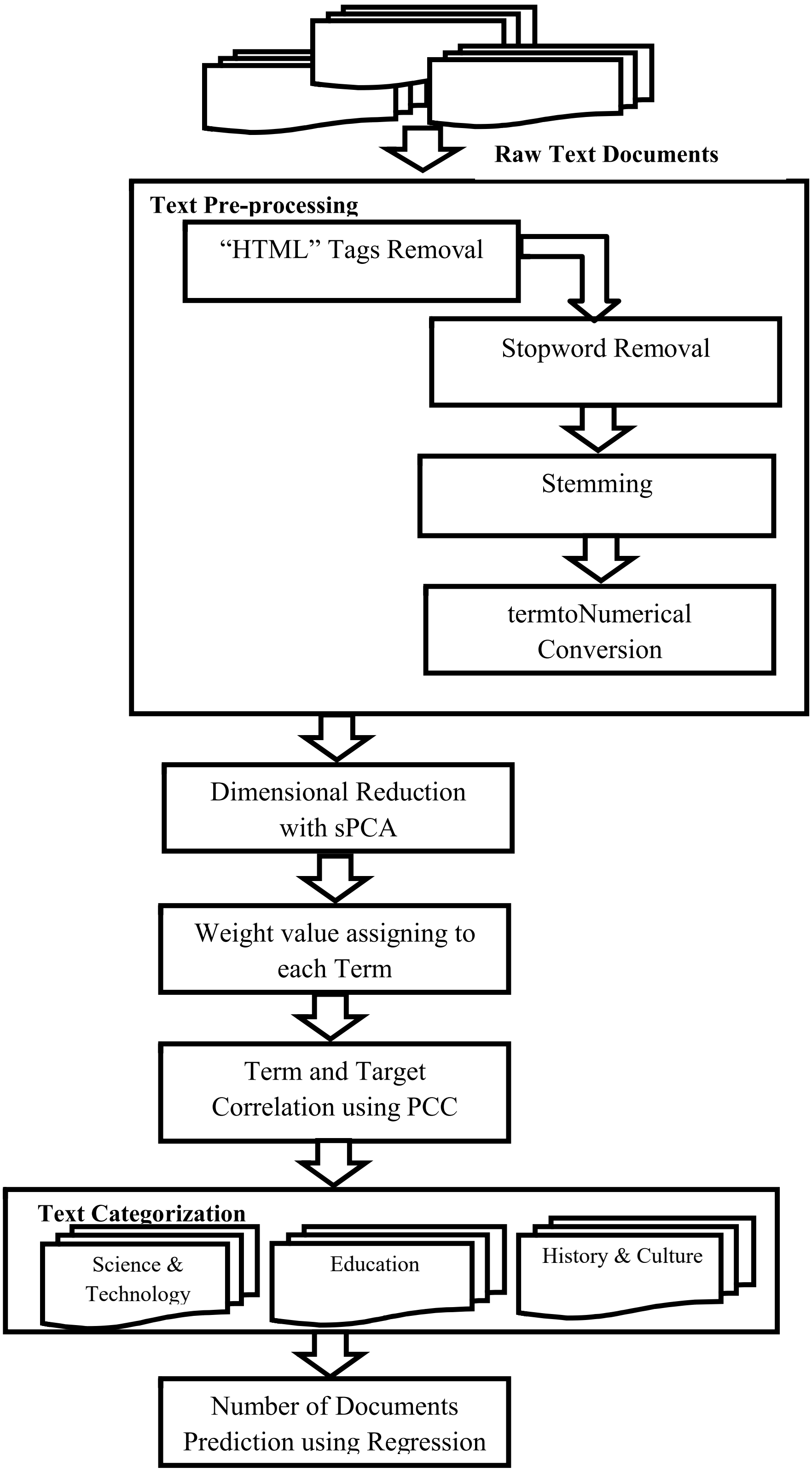

Raw Text Documents
Text Pre-processing
"HTML" Tags Removal
Stopword Removal
Stemming
termtoNumerical Conversion
Dimensional Reduction with sPCA
Weight value assigning to each Term
Term and Target Correlation using PCC
Text Categorization
Science & Technology
Education
History & Culture
Number of Documents Prediction using Regression

Figure 4.10 Number of Documents Prediction Process

Finally, the regression process is performed by taking large-scale text matrix including categorized text documents. It is intended to offer the number of documents as prediction outcome. For example, there are three predefined targets/classes for the proposed approach; the predicted results will be the number of Education related documents = “X”, the no. of Science & Technology related documents = “Y”, the number of History and Culture related documents = “Z”.

4.8 Multiple Linear Regression (MLR) Model with QR Decomposition

Although there has been a solution to reduce the dimensionality of input large-scale data matrix, it is still a big problem or issue to solve how to split or decompose the voluminous matrix containing increasing size of observations or data records in computing the regression model parameter “β”. In resolving the values of “β”, it can be hardly possible to process the entire huge input matrix at once. The most powerful and mathematically mature data analysis method, multiple linear regression is focused on a central approach traditionally where the computation is only done on a set of data stored in a single machine. With an increasing volume of data, the transition to the algorithm in distributed environment is hardly possible to implement. Multiple Linear Regression (MLR), a traditional statistical data analysis method, also proves unsuitable to facilitate the scalability of the data processed in the distributed environment due to computing memory and response time. The implementation of a parallel and distributed version for MLR model with QR Decomposition is applied to extract model’s coefficients with massive data processing.

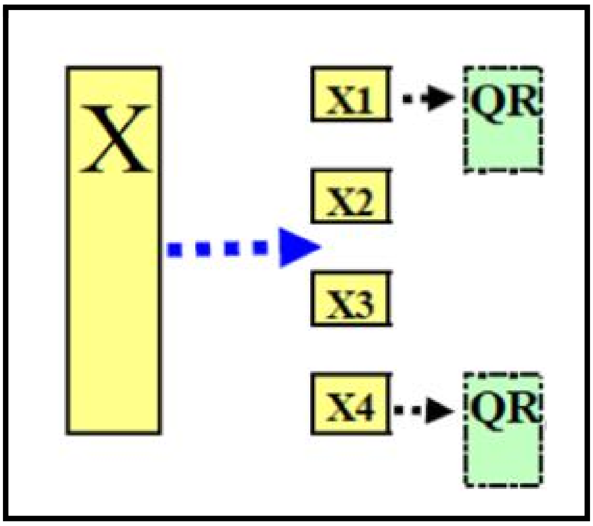


Figure 4.11 Decomposition of Matrix “X” into “QR”

4.8.1 The Proposed MLR Model with QR Decomposition on MapReduce

In big data era, it is an essential requirement to solve the transition to the scalability of the algorithms for parallel and distributed massive data processing with the use of MapReduce paradigm seems like a natural solution to this problem. Hadoop is an open framework used for big data analytics and its main processing engine is MapReduce, which is one of the most popular big data processing frameworks available. Algorithms that need to be highly parallelizable and distributable across huge datasets can also be executable on MapReduce using a large number of commodity computers. The focus is particularly on the adaptation of MLR in distributed massive data processing. However, there is still a big problem to solve how to split or decompose the large input matrix in computing the regression model parameter "β. In resolving the values of "β", it is actually needed to load the transpose of the input matrix and multiplication with its original matrix and then other subsequent complex matrix operations. It is impossible to process the entire huge input matrix at once. Therefore, matrix decomposition for the MLR model is contributed to overcome the limitations and the challenges of MLR model in big data. The proposed MLR model with QR Decomposition, a new computational approach, is presented to provide computing on the decomposed or factorized matrix with scalability advantage that is much faster than computing on the original matrix immediately without any decomposition.

When scaling large matrices, it is important to design efficient parallel algorithms for matrix operations, and using MapReduce is one way to achieve this goal. For example, in computing "β" values from the Equation 4.2, the inversion of matrix "R" must be calculated. Matrix inversion is difficult to implement in MapReduce because each element in the inverse of a matrix depends on multiple elements in the input matrix, so the computation is not easily splitting as required by the MapReduce programming model. QR Decomposition (also called a QR Factorization) of a given matrix A is a decomposition of matrix X into a product X = QR of an orthogonal matrix Q if $\boldsymbol{Q^T} = \boldsymbol{Q^{-1}}$ or $\boldsymbol{Q^T}$ Q = I and an upper triangular matrix R. It is used to solve the ordinary least squares problem in multiple linear regression and also the standard method for computing QR Factorization of a matrix which has many rows than columns (m > n) causing a common problem arisen in many real-world applications.

As it has been already known that data in a MapReduce processing is represented by a collection of Key-Value pairs. When applying MapReduce to analyze matrix-form data, a key represents the identity of a row and a value represents the elements in that row. Therefore, the matrix is also a collection of Key-Value pairs assuming that each row has a distinct key for simplicity although sometimes each key may represent a set of rows. To determine multiple linear regression model's coefficient, "b" the computational approach QR Decomposition is to simplify the calculation by decomposing the data matrix X into two matrices "Q" and "R" as follows:

$$\beta = (X^{T}X)^{-1}X^{T}Y \tag{4.2}$$

By Substituting X=QR,

$$\beta = (R)^{-1}Q^{T}Y \tag{4.3}$$

The procedure for the proposed MLR model with QR Decomposition on MapReduce takes block numbers as parameters to divide the large training input matrix 'X' and distribute it on several tasks of "Map" functions. The 'Map' function of the first stage takes 'Xi' sub-matrices decomposed from big training data 'X' matrix for all 'noBlock'. The two result matrices 'Qi' and 'Ri' with the respective 'Keyi' are produced for the 'Reduce' function. The main idea here is to highlight that each 'Map' process loads into memory at the maximum size of a matrix (BlockSize, n) which significantly overcomes the problem of "out of memory" with big matrix training data. Likewise, the 'Reduce' process will also receive a maximum array with size (n*noBlock, n). Therefore, choosing the number of blocks should be considered according to the size or number of machines in the cluster we applied. The more increasing computing power may be simply adding new machines into the cluster for the purpose of improving the 'MapReduce Framework Parallelism'.

The second stage receives input from the result of first stage and the y_i of for all blocks 'i'. In the 'Map' function, the vector y is decomposed into several vector y_i (number of blocks) and then sent to 'Reduce' function with associated key 'Key_i'. The third or final stage uses the input from second stage including set of vectors of

'V_i' and 'R_{final}'. The 'Map' function constructs a list 'ListRV' combining with all sets of 'Vi' from all blocks 'i' and 'R_{final}' with the associated key 'Key_{final}'. The 'Reduce' function takes the list 'ListRV' and adding the values of all 'V_i' vectors together to get the final vector V. Moreover, 'R_{final}' is applied as inverse matrix and finally multiplying with 'V' to obtain the 'bi [ ]' as final output for the proposed model. The detailed implementation procedure of the model is shown in Table 4.8, 4.9, 4.10, and 4.11 respectively.

Table 4.8 Main or Driver Function for the Proposed Model

Procedure for Driver Function
Input: Matrix X and Vector y from Input Dataset Start noBlock = Size (X) / BlockSize n (No of Observations for each Block 'i'): = X /noBlock X_i: = n For all row of y y_i: = y / noBlock End Output: Xi; with i: =1...noBlock

Table 4.9 First Stage Map/Reduce Function for the Proposed Model

Procedure for Map Function *(Mapper 1)*	Procedure for Reduce Function *(Reducer 1)*
Input: X_i of X for all blocks Output: Q_i, R_i Start For all block X_i of X (Q_i, R_i)= Map1 (X_i) Produce (Key_i, Q_i) Produce (Key_i, R_i) End for Function Map1 (X_i) Input: X_i of Matrix 'X' Start (Q_i,R_i):=QRDecompose (X_i) Output: Q_i, R_i End Function QRDecompose (X_i) Start	Input: <key, value> pairs for 'R_i' Matrix from Mapper1. Output: Q_i' for all block 'i', R_{final} for intermediate result in 'β' calculation(Q_i', R_{final})= Reduce1 ((Key_r, [R_1,R_2,.....,R_i]) For all blocks Q_i' of Q' Produce (Key_i, Q_i') End for Produce (Key_{final},R_{final}) Function Reduce1 (Key_r,[R_1,R_2,.....,R_i]) Input: R_{temp}=Matrix[R_1,R_2,.....,R_i] Start (Q_i',R_{final}):=QRDecompose (R_{temp}) For all row of Q' Q_i':= Decompose (Q', BlockSize)

Factorizing X_i into Q_i and R_i for all respective blocks ‘i’ End End	End For Function QRDecompose (R_{temp}) Start Factorizing R_{temp} into Q’ and R_{final} for subsequent decomposition of Matrix ‘R_{temp}’ End Function Decompose (Q’, BlockSize) Start Decomposition of Q’ into $Q_i^{’}$ for all blocks ‘i’ applying Q’/ BlockSize End End

Table 4.10 Second Stage Map/Reduce Function for the Proposed Model

Procedure for Map Function (Mapper 2)	Procedure for Reduce Function (Reducer 1)
Input: ListQ_i: = List [Key_i,(Q_i, Q’)] y_i of Vector y for all blocks ‘i’ Output: (Key_i, y_i) for all blocks ‘i’ Start y_i = Map2(y_i) Function Map2 (y_i) Input: y_i Start Produce (Key_i, y_i) End End	Input: List of Q_i, $Q_i^{’}$, y_i with the same key ‘Key_i’ Output: V_i[] for intermediate result in ‘β’ calculation For all block ‘i’ V_i[ ]= Reduce2(Key_i, List[Q_i, $Q_i^{’}$, y_i]) End for Function Reduce2 (Key_i, List [Q_i, $Q_i^{’}$, y_i]) Input: List [Q_i, $Q_i^{’}$, y_i] Start Q: = Multiply1 (Q_i, $Q_i^{’}$) Q^T: = Transpose (Q) $V_{i: =}$ Multiply2 (Q^T, y_i) End Function Multiply1 (Q_i, $Q_i^{’}$) Start Matrix multiplication of two matrices Q_i and $Q_i^{’}$ producing result matrix ‘Q’ End Function Multiply2 (Q^T, y_i) Start Matrix multiplication of transpose matrix ‘Q^T’ and y_i resulting ‘V_i’ arrays for

	all blocks 'i' End Function Transpose (Q) Start Transpose matrix operation of 'Q' End

Table 4.11 Third Stage Map/Reduce Function for the Proposed Model

Procedure for Map Function (*Mapper 3*)	Procedure for Reduce Function (*Reducer 3*)
29 Input: V_i[ ] for all blocks 'i' , <key, value> pairs for R_{final} with ($Key_{final,}$ R_{final}) 30 Output: ListRV ([R_{final}, $V_{1,....,}$ V_i]) 31 Start 32 ListRV: = Map3 (V_i[ ],($Key_{final,}R_{final}$)) 33 Function Map3 (V_i[ ],($Key_{final,}R_{final}$)) 34 Input: V_i[],($Key_{final,}R_{final}$) 35 Start 36 Mapping Key_{final} with all of V_i[] from all blocks 'i' (Key_{final}, List[R_{final}, $V_{1,....,}$ V_i]) 37 Produce ListRV ([R_{final}, $V_{1,....,}$ V_i]) 38 End 39 End	40 Input: ListRV ([R_{final}, $V_{1,....,}V_i$]) 41 Output: β_i[] coefficients of proposed model and final output of three stage MapReduce processing 42 Start 43 β_i[] := Reduce3(ListRV ([R_{final}, $V_{1,....,}V_i$])) Function Reduce3 (ListRV ([R_{final}, $V_{1,....,}V_i$])) Start InvR_{final} := Inverse (R_{final}) SumV := Sum (V_i) β_i[] := Multiply (Inv$R_{final,}$ SumV) End Function Inverse (R_{final}) Start Inverse matrix operation of 'R_{final}' End Function Sum (V_i) Start $\sum V_i$ for all blocks 'i' End Function Multiply (Inv$R_{final,}$ SumV) Start Multiplication of inverse matrix InvR_{final} and the values of SumV for all summation from V_i values of blocks 'i' End End

4.8.2 The Proposed MLR Model with QR Decomposition on Apache Spark

Nowadays, the modern datasets in the form of matrices are increasingly growing in size. Thus, it is a must thing to handle these large-scale matrices which spread across many machines with the same conditions available for single machine analysis. Batch processing-based MapReduce and real-time processing-based Apache Spark has been intended for large-scale data processing. In particular, Apache Spark has been emerged as a widely used open-source engine. It is a fault-tolerant and general-purpose cluster computing system providing APIs in Java, Scala, Python, and R, along with an optimized engine that supports general execution graphs.

Table 4.12 Procedure for the Proposed Model on Apache Spark

Procedure for the Proposed MLR Model on Apache Spark

```
Input: m x n real matrix type "M"
Output: RowMatrix "RM" of Spark
Steps
Converting matrix to Distributed Spark Matrix (RowMatrix)
Begin
  RowMatrix RM = convertMatrixtoRowMatrix (JavaSparkContext sc, RealMatrix M, int numSlices) {
     final double [ ][ ] dataArray = M. getData ();
     final LinkedList<Vector> rowsList = new LinkedList< > ();
     for (final double [ ] i: dataArray) {
          final Vector currentRow = Vectors.dense( i );
          rowsList.add(currentRow);
          }
final JavaRDD<Vector> rows = sc. parallelize (rowsList, numSlices);
Create a RowMatrix from JavaRDD<Vector>
```

```
        Begin
            final RowMatrix mat = new RowMatrix (rows.rdd ());
            return mat;
        End
        Getting size of the RowMatrix
        Begin
          long m = mat. numRows ();
          long n = mat.numCols();
        End
        Applying "QR" Decomposition on RowMatrix
        Begin
          QRDecomposition<RowMatrix, Matrix> result = mat.tallSkinnyQR(true);
        End
End
```

4.9 Chapter Summary

The detailed explanation of the proposed system is described in this chapter. The processing of each stage in main architecture of the system is explained with applied algorithms, methods, tools and libraries respectively. Furthermore, the explanation of different case studies for the system are mentioned to provide the complete structure of the system. In order to promote the system's performance, the proposed approaches which solve the issues and problems of the existing system are discussed in detail.

CHAPTER 5

IMPLEMENTATION OF THE PROPOSED SYSTEM

The main purpose of the chapter is to describe the experimental environment and implementation procedures of the proposed system. The performance of the proposed system has been evaluated on big data analytics platform using "Cloudera Distribution Hadoop (CDH)".

5.1 Experimental Environment

The proposed system is implemented on "Multi Node Cloudera Cluster using three computing nodes or VMs which all are interconnected with Cloudera Manager. These VMs including a VM which is already configured with Cloudera Manager are interconnected through a 1-gigabit Ethernet. To setup the cluster, single Cloudera VM on Oracle Virtual Box is needed to install.

The specification of host machine can be expressed as follows:

- Intel Core i7-2.90GHz Processor
- 8GB Physical RAM
- 950-GB Disk

The software components for each VM are as follows:

- Hadoop 2.6.0
- Spark 2.4.0
- Mahout Machine Learning Library
- MLlib Machine Learning Library

The installation and configuration of the experimental environment can be seen in Appendix. The proposed system is implemented on Cloudera Distribution Hadoop (CDH) big data analytics platform which offers good opportunities to analyze data for gaining a better understanding of the data. The proposed system using Apache Spark Architecture on Cloudera VM of CDH platform is illustrated in Figure 5.1. There are four main parts in the Figure 5.1:

- Data Collection
- Data Analytics

- Data Processing
- Data Storage

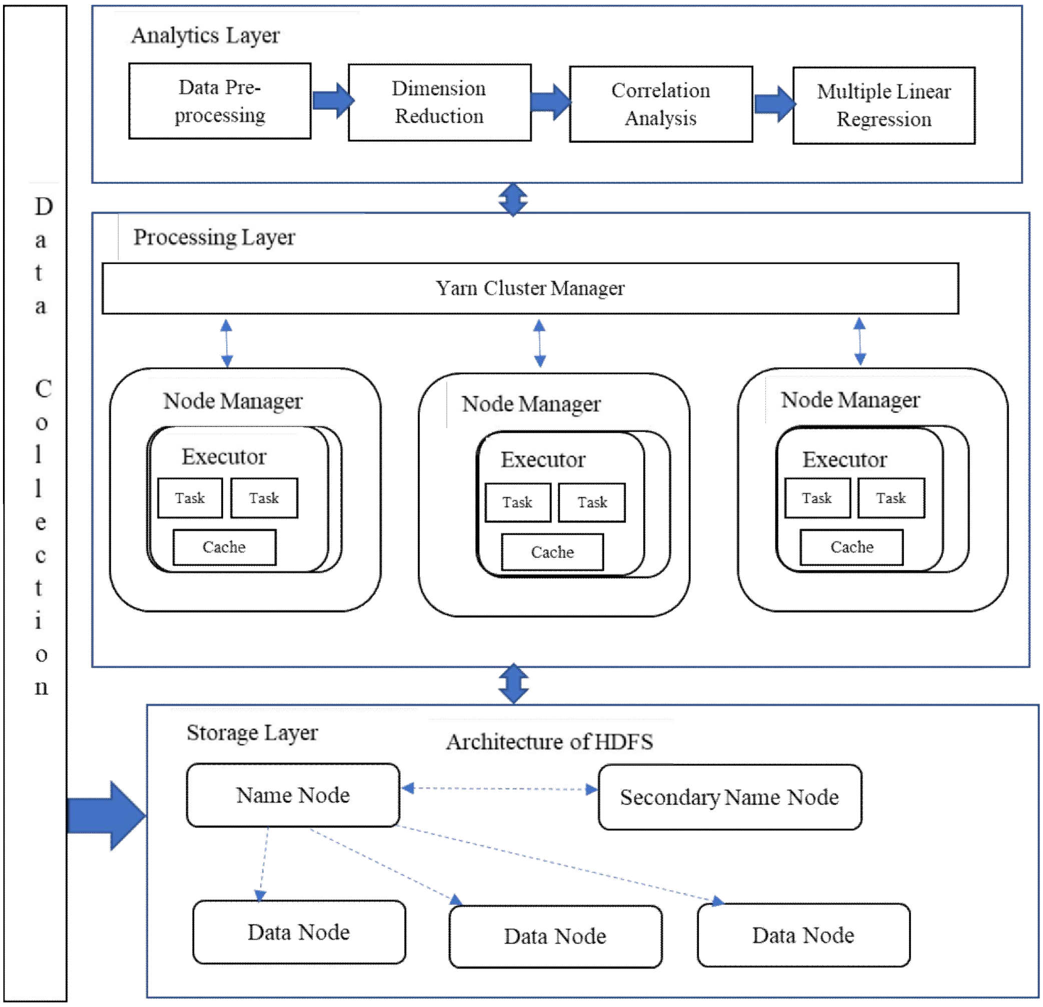


Figure 5.1 The Proposed System using Apache Spark Architecture of Cloudera VM

5.2 Implementation of the System without using the Proposed Approaches

For this section, the implementation of the system is shown in Figure 5.2. According to this figure, principal component analysis (PCA) is applied to reduce high-dimension of input data matrix. Then, the data matrix with reduced dimensions are given to the multiple linear regression model for prediction purpose.

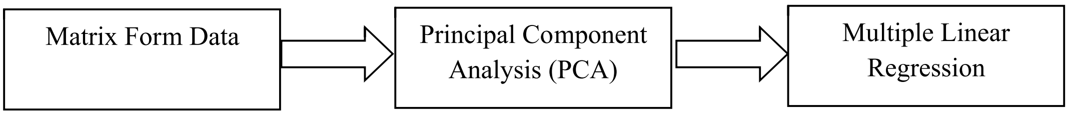

Matrix Form Data
Principal Component Analysis (PCA)
Multiple Linear Regression

Figure 5.2 Implementation of the system with traditional PCA

According to issues and problems of traditional PCA, the scalable Principal Component Analysis (sPCA) is proposed in this system to facilitate the implementation of classical PCA on high-dimensional big data.

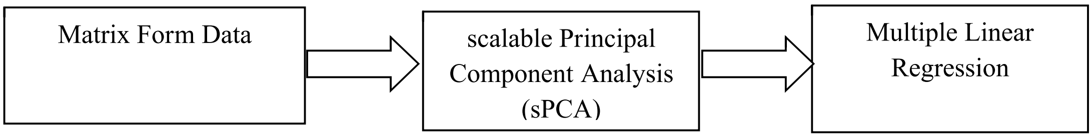

Matrix Form Data
scalable Principal Component Analysis (sPCA)
Multiple Linear Regression

Figure 5.3 Implementation of the system with sPCA

5.3 Implementation of the System using Two-Stage Dimension Reduction Approach

In order to increase the prediction accuracy of the multiple linear regression model, the features are needed to filter out for the model as the redundant and irrelevant features are a big problem in prediction process. The proposed two-stage dimension reduction approach is contributed to filter or select the most important and relevant features for multiple linear regression model. Firstly, PCA is applied to features (dimensions) comprising in matrix form data by reducing the dimensions to avoid multicollinearity problem in regression model. And then, the correlation between a feature and the target class (predicted output) is also considered for the model. The correlation process is carried out by Pearson Correlation Coefficient (PCC). Applying PCA for high-dimensions of big data can reduce less significant or sometimes noisy features, however, it needs to find out the most correlated features with the predicted output. Therefore, the implementation of the system is shown in Figure 5.4 where the dimension reduction is experimented by the proposed two-stage approach. Although PCA, the first stage of the approach, can efficiently reduce the dimensions of data, correlation process by PCC, the second stage of the approach, offers more reduced number of dimensions. Filtering by PCC removes the uncorrelated or irrelevant number of features or dimensions with the target predicted output. The main purpose of this approach is to obtain the most important and irrelevant features for the regression model. Thus, the first stage of the model is intended to remove the redundant features and the second stage is also intended to minimize the irrelevant features for the model.

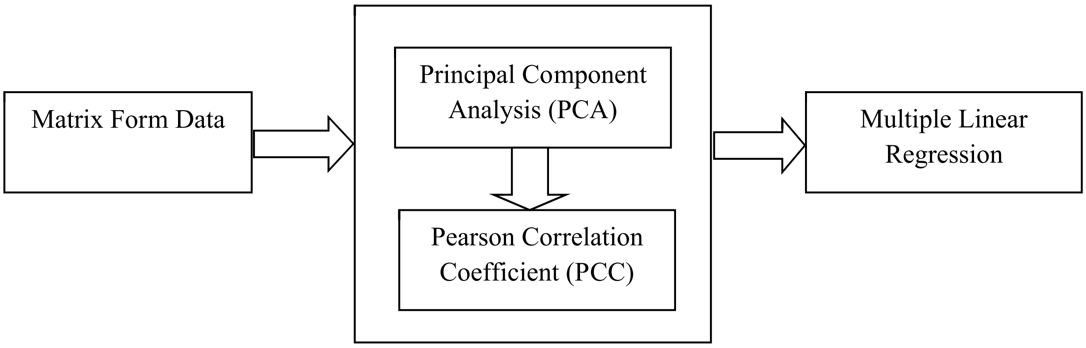

Matrix Form Data
Principal Component Analysis (PCA)
Pearson Correlation Coefficient (PCC)
Multiple Linear Regression

Figure 5.4 Implementation of the system with two-stage approach (PCA and PCC)

The proposed two-stage approach using classical PCA in first stage also encounters the problem of "Out of Memory" in the system. Therefore, sPCA is also applied as the first stage of two-stage approach replacement of PCA. The implementation for the system is shown in Figure 5.5.

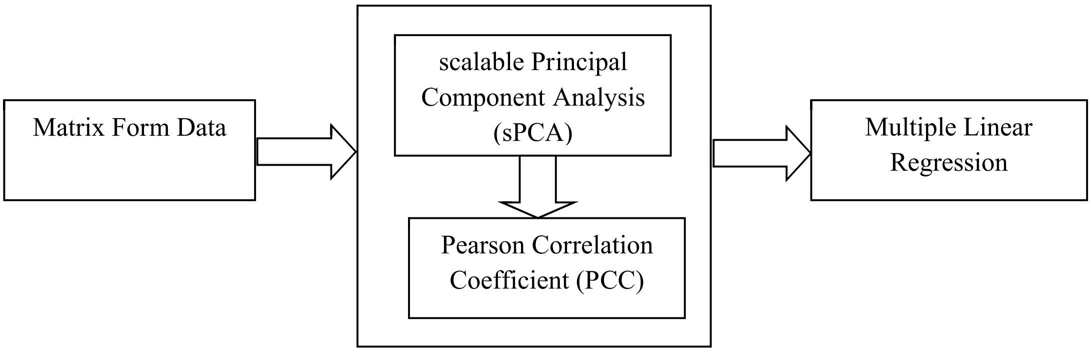

Matrix Form Data
scalable Principal Component Analysis (sPCA)
Pearson Correlation Coefficient (PCC)
Multiple Linear Regression

Figure 5.5 Implementation of the system with two-stage approach (sPCA and PCC)

5.4 Implementation of the System using QR Decomposition

After applying the proposed two-stage dimension reduction approach, the high-dimensions of respective data matrix are reduced as much as it is possible. Then, the reduced number of dimensions are given to the multiple linear regression model for prediction process. The reduction of high-dimensions can be defined as the transformation of tall-and-fat into tall-and-skinny data form. Although the dimension reduction approach can solve the issues of redundant and irrelevant features and also the problem of multicollinearity in regression model, the system still remains a challenge of voluminous amount of records or observations in respective data matrices. The system is implemented on distributed big data processing platform; however, it is a big problem to process large-scale matrices consisting voluminous observations. It is hardly possible to process big matrix data at once for the prediction model as the distributed processing of matrix data is not easy to process. Therefore, matrix decomposition is absolutely needed in multiple linear regression model to process input large-scale data matrices.

In this system, the matrix decomposition approach called "QR Decomposition" is proposed in decomposing data matrix into "Q" and "R" matrices with the purpose of finding "β" values for regression model. The implementation for the system is shown in Figure 5.6.

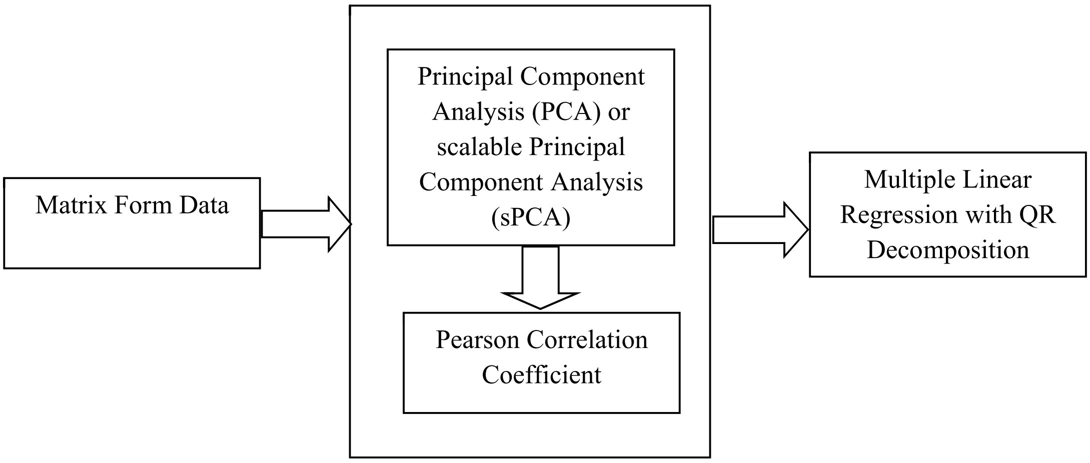

Matrix Form Data
Principal Component Analysis (PCA) or scalable Principal Component Analysis (sPCA)
Pearson Correlation Coefficient
Multiple Linear Regression with QR Decomposition

Figure 5.6 Implementation of the system with "QR" Decomposition

5.5 Chapter Summary

In this chapter, the implementation design of the proposed system is presented on distributed platform called “Cloudera Distribution Hadoop (CDH)”. Moreover, the implementation procedures of the system are discussed in detail with and without using the proposed approaches.

CHAPTER 6

EXPERIMENTAL RESULTS EVALUATION

In this chapter, the main purpose is to discuss the experimental results of the proposed system using three different case studies with a number of experiments. For experimentation, three diverse high-dimensional big data sources such as large-scale text documents from “DeliciousMIL”, images from “MS-Celeb-A” and OSM XML from “OpenStreetMap” are utilized in the system. According to different data sources, raw data from these sources is pre-processed into matrix form data by respective pre-processing procedures. The pre-processed data matrices are then processed by step-by-step procedures to achieve prediction outcomes from the system. In addition, the prediction outcomes resulted from the system are evaluated according to scalability, execution time and prediction accuracy. These performance evaluation factors are mainly based on scalable Principal Component Analysis (sPCA), two-stage dimension reduction approach, and QR Decomposition with Multiple Linear Regression (MLR) model for the proposed system.

6.1 Dimension Reduction Results using Traditional PCA

The results of dimension reduction which come from traditional PCA are described in Table 6.1, 6.2 and 6.3 respectively. In these tables, it can be seen that the input size or number of dimensions for text, image and osm data matrix are different according to diverse data sources, sizes and structures. For image data matrix as shown in Table 6.2, there exists the higher number of dimensions than osm and text data matrices. It is intended to prove that the proposed dimension reduction approach can efficiently perform in different sizes of data dimensions which come from diverse data sources. In addition, it also creates a good opportunity to obviously see the scalability performance of the proposed sPCA by using different sizes of data dimensions.

Table 6.1 Dimension Reduction using Traditional PCA on Text Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using Traditional PCA
1	20000	12500
2	21000	13200
3	22000	14510
4	23000	15400
5	24000	16900
6	25000	Out of Memory

Table 6.2 Dimension Reduction using Traditional PCA on Image Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using Traditional PCA
7	4000000	86000
8	5000000	100000
9	6000000	122000
10	7000000	164000
11	8000000	188000
12	10000000	Out of Memory

Table 6.3 Dimension Reduction using Traditional PCA on OSM Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using Traditional PCA
13	1000	210
14	1200	252
15	1400	310
16	1600	336
17	1800	378
18	2000	Out of Memory

6.2 Dimension Reduction Results using the Proposed sPCA

According to the results from Table 6.1, 6.2 and 6.3, it can be clearly seen that "Out of Memory" condition has emerged when input dimensions are exceeded to process by Traditional PCA. Therefore, the evaluation procedure is replaced with sPCA instead of PCA. sPCA can efficiently perform dimension reduction process with the higher number of dimensions and these results are presented in Table 6.4, 6.5 and 6.6 respectively.

Table 6.4 Dimension Reduction using sPCA on Text Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using sPCA
19	25000	17500
20	26000	18712
21	27000	19000
22	28000	22700
23	29000	24512
24	30000	26890

Table 6.5 Dimension Reduction using sPCA on Image Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using sPCA
25	4000000	86000
26	5000000	100000
27	6000000	122000
28	7000000	164000
29	8000000	188000
30	10000000	1980000

Table 6.6 Dimension Reduction using sPCA on OSM Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using sPCA
31	2000	914
32	2200	1324
33	2400	1569
34	2600	1678
35	2800	1731
36	3000	2260

The comparison results of dimension reduction between Traditional PCA and the proposed sPCA for text data matrix is shown in Table 6.7. According to the Table 6.7, it can be analyzed that sPCA can efficiently perform dimension reduction with the size or number of dimensions being increased. The proposed sPCA also shows the good scalability performance which traditional PCA cannot facilitate in high-dimensional big data. While the traditional PCA offers "Out of Memory" results, sPCA enables to continue the dimension reduction process for the system.

Table 6.7 Comparison of Dimension Reduction between Traditional PCA and sPCA using Text Data

No.	Number of Input Dimensions	Number of Reduced Dimensions using Traditional PCA	Number of Reduced Dimensions using sPCA
1.	20000	12500	12500
2.	21000	13200	13200
3.	22000	14510	14510
4.	23000	15400	15400
5.	24000	16900	16900
6.	25000	Out of Memory	17500

7.	26000	Out of Memory	18712
8.	27000	Out of Memory	19000
9.	28000	Out of Memory	22700
10.	29000	Out of Memory	24512
11.	30000	Out of Memory	26890

For image data matrix, the comparison of dimension reduction results between traditional PCA and the proposed sPCA is described in Table 6.8. By analyzing the Table 6.8, it can be seen that sPCA can efficiently perform dimension reduction with extremely large number of dimensions. Although the number of dimensions for image data are significantly higher than other data types, the proposed sPCA overcomes the scalability problem of Traditional PCA. The comparison results of dimension reduction between Traditional PCA and the proposed sPCA for OSM data is shown in Table 6.9.

Table 6.8 Comparison of Dimension Reduction between Traditional PCA and sPCA using Image Data

No.	Number of Input Dimensions	Number of Reduced Dimensions using Traditional PCA	Number of Reduced Dimensions using sPCA
1.	4000000	86000	86000
2.	5000000	100000	100000
3.	6000000	122000	122000
4.	7000000	164000	164000
5.	8000000	188000	188000
6.	10000000	Out of Memory	1980000
7.	11000000	Out of Memory	2130000
8.	12000000	Out of Memory	2400000

9.	13000000	Out of Memory	2680000
10.	14000000	Out of Memory	3140000
11.	15000000	Out of Memory	3240000

Table 6.9 Comparison of Dimension Reduction between Traditional PCA and sPCA using OSM Data

No.	Number of Input Dimensions	Number of Reduced Dimensions using Traditional PCA	Number of Reduced Dimensions using sPCA
1.	1000	210	210
2.	1200	252	252
3.	1400	310	310
4.	1600	336	336
5.	1800	378	378
6.	2000	Out of Memory	914
7.	2200	Out of Memory	1324
8.	2400	Out of Memory	1569
9.	2600	Out of Memory	1678
10.	2800	Out of Memory	1731
11.	3000	Out of Memory	2260

6.3 Prediction Results from MLR Model using PCA/sPCA

After applying only PCA or sPCA for dimension reduction process, the reduced dimensions are given as input variables “X” to the MLR model. Prediction output “Y” for the model is denoted according to the incoming type of data matrix. For “OSM” data matrix, the intended prediction output “Y” is “One-way Roads”. “Number of Faces” is denoted as prediction output “Y” for “Image” data matrix.

Document types such as “Education”, “Culture and History”, and “Science and Technology” are also prediction output “Y” for “Text” data matrix. These input and output variables are utilized to the MLR model as training data for finding out the regression model’s coefficient “β”. After applying the training data and “β”, the prediction results are obtained as new prediction output “Y”.

In this research, one of the goals of the system is to show that applying dimension reduction approach for regression model offers better prediction accuracy in diverse high-dimensional data sources. Therefore, “One-way Roads Prediction” for OSM data, “Number of Faces Prediction” for image data, and “Number of Documents (Education, Culture && History, Science && Technology) Prediction” for text data using one-stage dimension reduction (PCA or sPCA) are presented in detail from Table 6.10 to 6.14 respectively. In these tables, the comparative studies between total and predicted number of outputs which are resulted from the MLR model are conducted. Prediction outcomes between types of documents such as Education, Science && Technology, and Culture && History are shown in Table 6.10, 6.11 and 6.12 respectively. It can be analyzed that prediction accuracy for document type “Education” is the best among them.

The accuracy for “Culture && History”, however, is not as good as “Science && Technology” which stands at an acceptable level. For number of faces prediction, as shown in Table 6.13, the results are satisfying for predicted number of faces compared with total number of faces which exist in original image dataset. Finally, the number of one-way roads prediction as shown in Table 6.14 can be assumed as the optimal prediction accuracy among other types of prediction. The overall comparison of prediction accuracy between three different kinds of document predictions can be seen in Figure 6.8. According to this figure, it can also be analyzed that the accuracy of “Education” document type prediction is the best in all conditions. The prediction accuracy of “Science&&Technology” document type is sometimes equal as and mostly lower than that of “Education” document type. The worst accuracy among them is “Culture&&History” document type prediction in all conditions.

Table 6.10 Number of "Education" Documents Prediction Results

Test Case No.	1	2	3	4	5
Total Number of Education Documents	1422	2885	4302	5831	7161
Predicted Number of Education Documents	1023	2381	3001	3862	5074

Table 6.11 Number of "Science && Technology" Documents Prediction Results

Test Case No.	1	2	3	4	5
Total Number of Science && Technology Documents	1061	2173	3165	4155	5211
Predicted Number of Science && Technology Documents	838	435	1393	1870	2345

Table 6.12 Number of "Culture && History" Documents Prediction Results

Test Case No.	1	2	3	4	5
Total Number of Culture && History Documents	626	1218	1855	2463	3094
Predicted Number of Culture && History Documents	473	748	810	941	1052

Table 6.13 Number of “Faces” Prediction Results

Test Case No.	1	2	3	4
Total Number of Faces	100	110	143	180
Predicted Number of Faces	74	98	107	137

Table 6.14 Number of “One-way Roads” Prediction Results

Test Case No.	1	2	3	4
Total Number of One-way Roads	133	200	267	300
Predicted Number of One-way Roads	117	176	221	237

6.4 Dimension Reduction Results using Traditional PCA and PCC

The results of dimension reduction which come from Traditional PCA and correlation process PCC combined approach are described in Table 6.15, 6.16 and 6.17 respectively.

Table 6.15 Dimension Reduction using PCA and PCC on Text Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using Traditional PCA	Number of Reduced Dimensions using Traditional PCA and PCC
1	20000	12500	6400
2	21000	13200	8300
3	22000	14510	9100
4	23000	15400	11000
5	24000	16900	13800
6	25000	Out of Memory	-

Table 6.16 Dimension Reduction using PCA and PCC on Image Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions after Traditional PCA	Number of Reduced Dimensions using Traditional PCA and PCC
7	4000000	86000	16600
8	5000000	100000	20000
9	6000000	122000	22500
10	7000000	164000	33300
11	80000000	188000	38000
12	10000000	Out of Memory	-

Table 6.17 Dimension Reduction using PCA and PCC on OSM Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions after Traditional PCA	Number of Reduced Dimensions using Traditional PCA and PCC
13	1000	210	147
14	1200	252	176
15	1400	310	197
16	1600	336	235
17	1800	378	264
18	2000	Out of Memory	-

6.5 Dimension Reduction Results using sPCA and PCC

The increased number of dimensions are also efficiently reduced by sPCA for respective types of data matrix. Dimension reduction results by sPCA and subsequent correlation process PCC are then presented in Table 6.18, 6.19 and 6.20 respectively.

Table 6.18 Dimension Reduction using sPCA and PCC on Text Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using sPCA	Number of Reduced Dimensions using sPCA and PCC
19	25000	17500	15700
20	26000	18712	16200
21	27000	19000	18400
22	28000	22700	20500
23	29000	24512	21300
24	30000	26890	22900

Table 6.19 Dimension Reduction using sPCA and PCC on Image Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using sPCA	Number of Reduced Dimensions using sPCA and PCC
25	10000000	1980000	380000
26	11000000	2130000	430000
27	12000000	2400000	480000
28	13000000	2680000	540000
29	14000000	3140000	600000
30	15000000	3240000	630000

Table 6.20 Dimension Reduction using sPCA and PCC on OSM Data

Test Case No.	Number of Input Dimensions	Number of Reduced Dimensions using sPCA	Number of Reduced Dimensions using sPCA and PCC
31	2000	914	653
32	2200	1324	962

33	2400	1569	1024
34	2600	1678	1113
35	2800	1731	1298
36	3000	2260	1684

In addition, the comparison of dimension reduction between PCA/sPCA only and PCA/sPCA and PCC combined approach for three types of data matrix are shown in Table 6.21, 6.22 and 6.23 respectively. By analyzing the results from these tables, it can be clearly seen that the number of dimensions resulted from PCA/sPCA and PCC combined approach offers more reduced number of dimensions for the subsequent prediction process of MLR model.

Table 6.21 Comparison of Dimension Reduction between PCA/sPCA only and PCA/sPCA and PCC using Text Data

No.	Number of Input Dimensions	Number of Reduced Dimensions using PCA/sPCA		Number of Reduced Dimensions using PCA/sPCA and PCC
1.	20000	12500	PCA	6400
2.	21000	13200	PCA	8300
3.	22000	14510	PCA	9100
4.	23000	15400	PCA	11000
5.	24000	16900	PCA	13800
6.	25000	Out of Memory	PCA	-
7.	25000	17500	sPCA	15700
8.	26000	18712	sPCA	16200
9.	27000	19000	sPCA	18400
10.	28000	22700	sPCA	20500

11.	29000	24512	sPCA	21300
12.	30000	26890	sPCA	22900

Table 6.22 Comparison of Dimension Reduction between PCA/sPCA only and PCA/sPCA and PCC using Image Data

No.	Number of Input Dimensions	Number of Reduced Dimensions using PCA/sPCA		Number of Reduced Dimensions by PCA/sPCA and PCC
1.	4000000	86000	PCA	16600
2.	5000000	100000	PCA	20000
3.	6000000	122000	PCA	22500
4.	7000000	164000	PCA	33300
5.	80000000	188000	PCA	38000
6.	10000000	Out of Memory	PCA	-
7.	10000000	1980000	sPCA	380000
8.	11000000	2130000	sPCA	430000
9.	12000000	2400000	sPCA	480000
10.	13000000	2680000	sPCA	540000
11.	14000000	3140000	sPCA	600000
12.	15000000	3240000	sPCA	630000

Table 6.23 Comparison of Dimension Reduction between PCA/sPCA only and PCA/sPCA and PCC using OSM Data

No.	Number of Input Dimensions	Number of Reduced Dimensions using PCA/sPCA		Number of Reduced Dimensions using PCA/sPCA and PCC
1.	1000	210	PCA	147
2.	1200	252	PCA	176
3.	1400	310	PCA	197
4.	1600	336	PCA	235
5.	1800	378	PCA	264
6.	2000	Out of Memory	PCA	-
7.	2000	914	sPCA	653
8.	2200	1324	sPCA	962
9.	2400	1569	sPCA	1024
10.	2600	1678	sPCA	1113
11.	2800	1731	sPCA	1298
12.	3000	2260	sPCA	1684

6.6 Prediction Results from MLR Model using PCA/sPCA and PCC

"One-way Roads Prediction", "Number of Faces Prediction" and "Number of Documents (Education, Science&&Technology, Culture&&History) Prediction" using the proposed two-stage approach (PCA/sPCA and PCC) are described in detail from Table 6.24 to 6.28. After applying not only PCA/sPCA but also PCC for three different types of data matrices, the predicted number of outputs are achieved for respective types of predictions. These tables also show the comparative studies between total number of outputs and predicted number of outputs which are resulted from the MLR model. According to the Tables 6.24, 6.25 and 6.26, it can be analyzed that the predicted number of documents such as Education, Science && Technology, Culture && History are not quite different with total number of documents. Moreover, the accuracy for number of faces and number of one-way roads prediction can be seen as satisfying conditions in Table 6.27 and 6.28 because the total and predicted number of outputs are approximately same in number.

In these tables, it can also be analyzed that when total number of faces and one-way roads are increased in number, the predicted number of faces and one-way roads are also increased and nearly as same as in number. The more detailed analysis results of prediction accuracy comparison for three types of documents (Education, Science && Technology, Culture && History) after applying PCA/sPCA and PCC are illustrated in Figure 6.1 to 6.3. According to the results as shown in these figures, the percentage of accuracies after applying PCA/sPCA and PCC are significantly higher than applying PCA/sPCA only. It can be distinctly seen that the percentage of accuracies for Science && Technology and Culture && History document types between two approaches of dimension reduction are quite different in Figure 6.2 and 6.3. Furthermore, the prediction accuracy comparison between three types of documents after applying PCA/sPCA is shown in Figure 6.4. It can be analyzed that the percentage of accuracies for Science && Technology and Culture && History document types are significantly lower than Education type.

However, the percentage of accuracies for three types of documents after applying PCA/sPCA and PCC are distinctly increased than applying PCA/sPCA only and sometimes they reach the same level especially "Dimension Number 29000" and "Dimension Number 33000" in Figure 6.5. According to Figure 6.5, it can be claimed that applying the proposed two-stage dimension reduction approach (PCA/sPCA and PCC) provides the MLR model with better prediction accuracy. The analysis results of prediction accuracy comparison for "No. of Faces Prediction" and "One-way Roads Prediction" are shown in Figure 6.6 and 6.7 respectively. From these figures, it can be also analyzed that the predicted number of outputs resulted from MLR model after applying PCA/sPCA and PCC are close to total number of respective types than applying PCA/sPCA only.

According to Figure 6.6, the prediction accuracies applying PCA/sPCA and PCC can be claimed that "Number of One-way Roads Prediction" achieves the best prediction accuracy than other types of prediction. "Number of Documents Prediction" obtain good prediction accuracies up to 86 percent (%). Moreover, the prediction accuracies for "Number of Faces Prediction" as shown in Figure 6.7 obtain better prediction accuracies up to 97 percent (%). According to the Figure 6.6 and 6.7, it can also be claimed that "Number of Faces Prediction" and "Number of One-way Roads Prediction" achieve the satisfactory accuracies over 90 percent (%), and Number of One-way Roads Prediction reaches the victory of accuracy in 99 percent (%). It can also be proved that applying the proposed two-stage approach (PCA/sPCA and PCC) offers the better prediction accuracy by reducing the redundant and irrelevant dimensions or features from the input data matrix before MLR model.

Table 6.24 Number of "Education" Documents Prediction Results (PCA/sPCA and PCC)

Test Case No.	1	2	3	4	5
Total Number of Education Documents	1422	2885	4302	5831	7161
Predicted Number of Education Documents	1123	2482	3012	3966	5085

Table 6.25 Number of "Science && Technology" Documents Prediction Results (PCA/sPCA and PCC)

Test Case No.	1	2	3	4	5
Total Number of Science && Technology Documents	1061	2173	3165	4155	5211
Predicted Number of Science && Technology Documents	848	1868	2215	2825	3699

Table 6.26 Number of "Culture && History" Documents Prediction Results (PCA/sPCA and PCC)

Test Case No.	1	2	3	4	5
Total Number of Culture && History Documents	626	1218	1855	2463	3094
Predicted Number of Culture && History Documents	494	1047	1298	1674	2196

Table 6.27 Number of "Faces" Prediction Results (PCA/sPCA and PCC)

Test Case No.	1	2	3	4
Total Number of Faces	100	110	143	180
Predicted Number of Faces	95	105	136	174

Table 6.28 Number of "One-way Roads" Prediction Results (PCA/sPCA and PCC)

Test Case No.	1	2	3	4
Total Number of One-way Roads	133	200	267	300
Predicted Number of One-way Roads	132	198	263	288

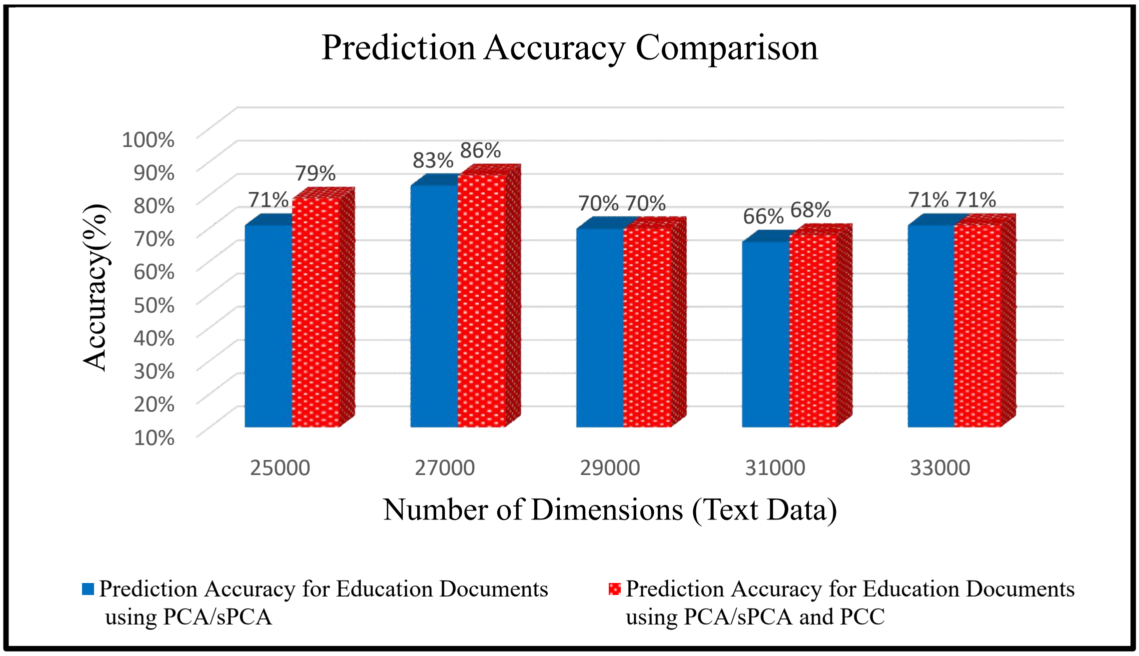


Figure 6.1 Comparison of Prediction Accuracy for Text Data ("Education" Type)

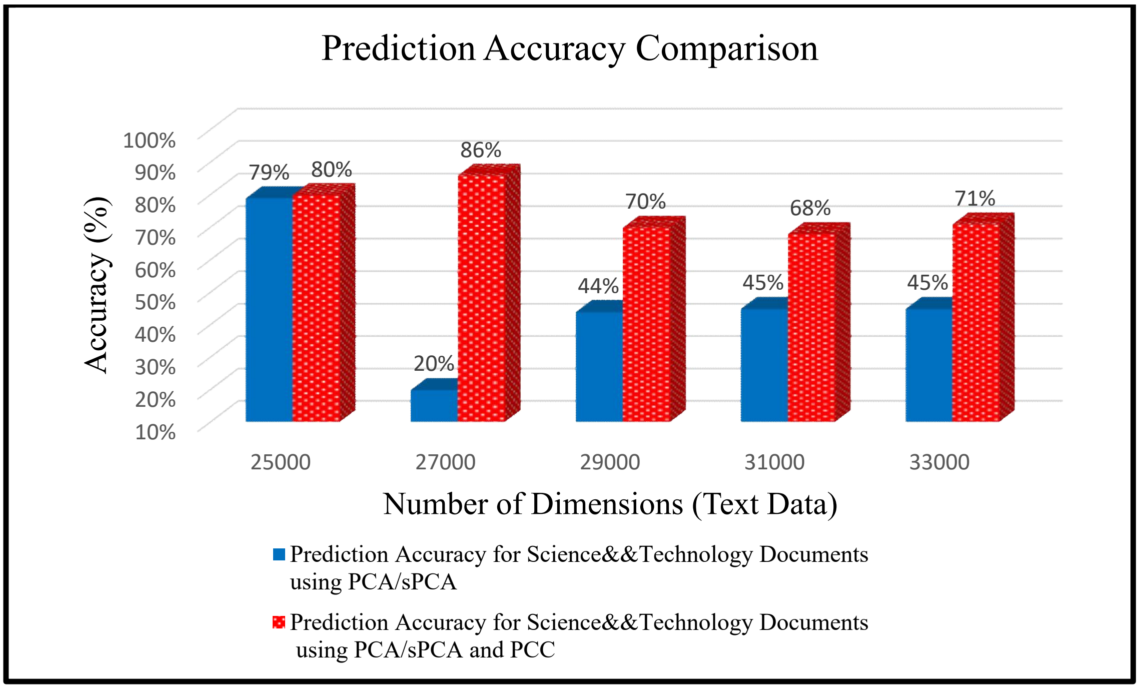


Figure 6.2 Comparison of Prediction Accuracy for Text Data ("Science&&Technology" Type)

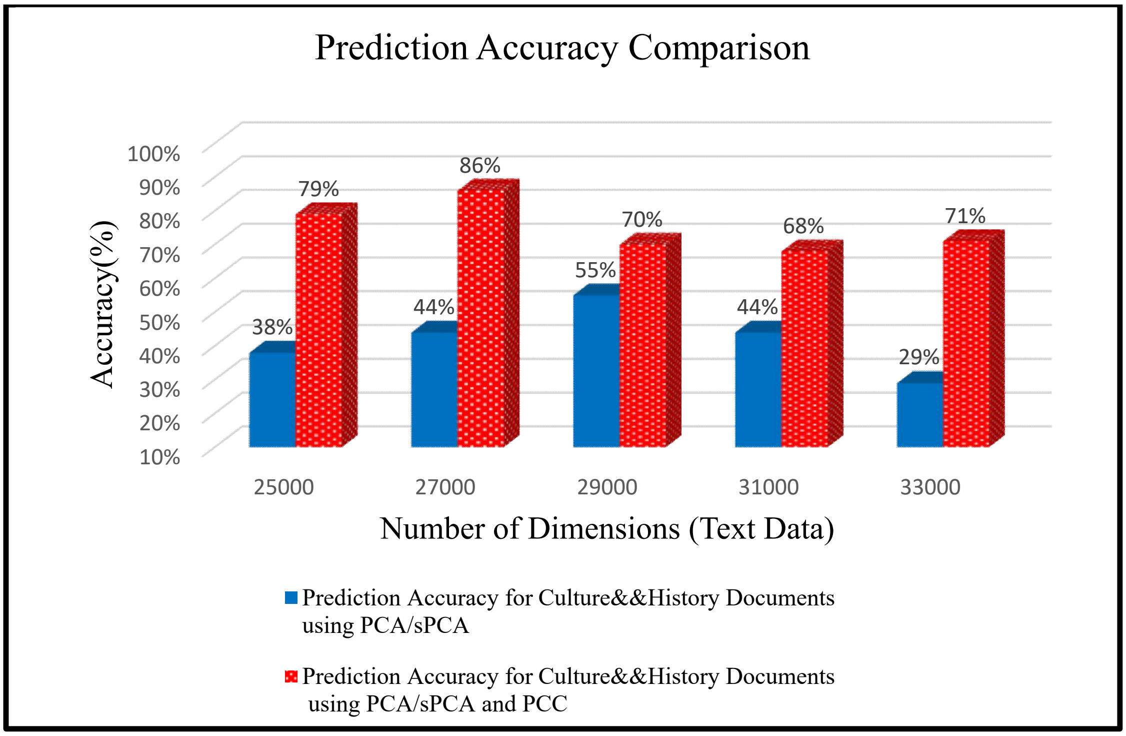


Figure 6.3 Comparison of Prediction Accuracy for Text Data ("Culture&&History" Type)

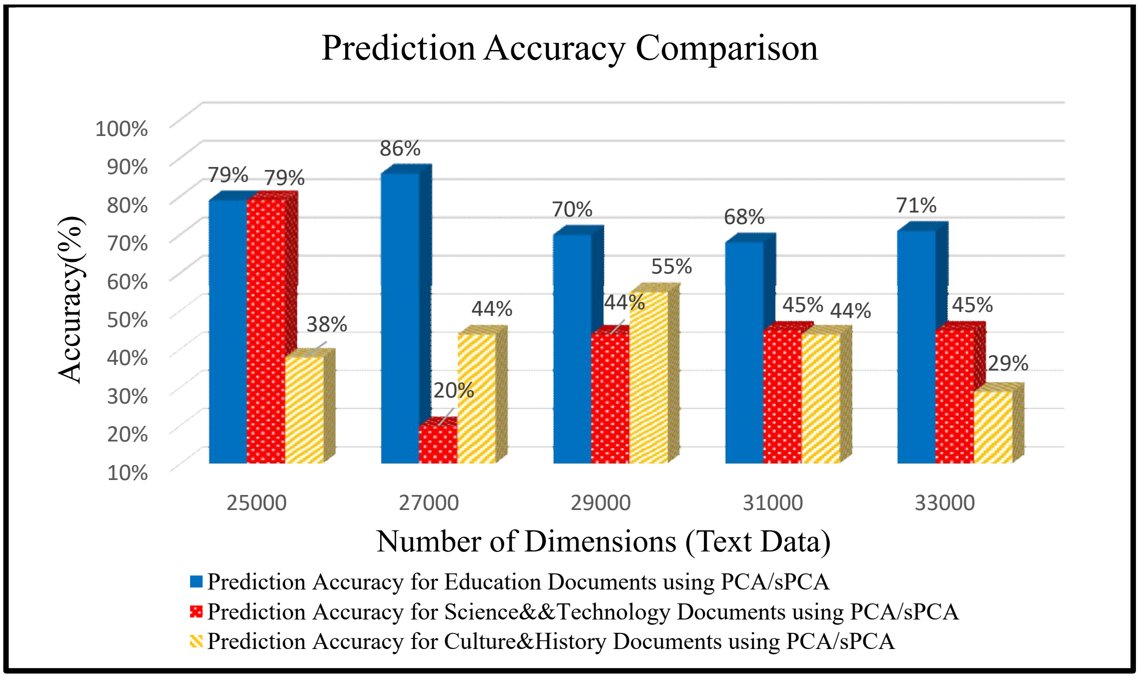


Figure 6.4 Comparison of Prediction Accuracy for Text Data using PCA/sPCA (Three Document Types)

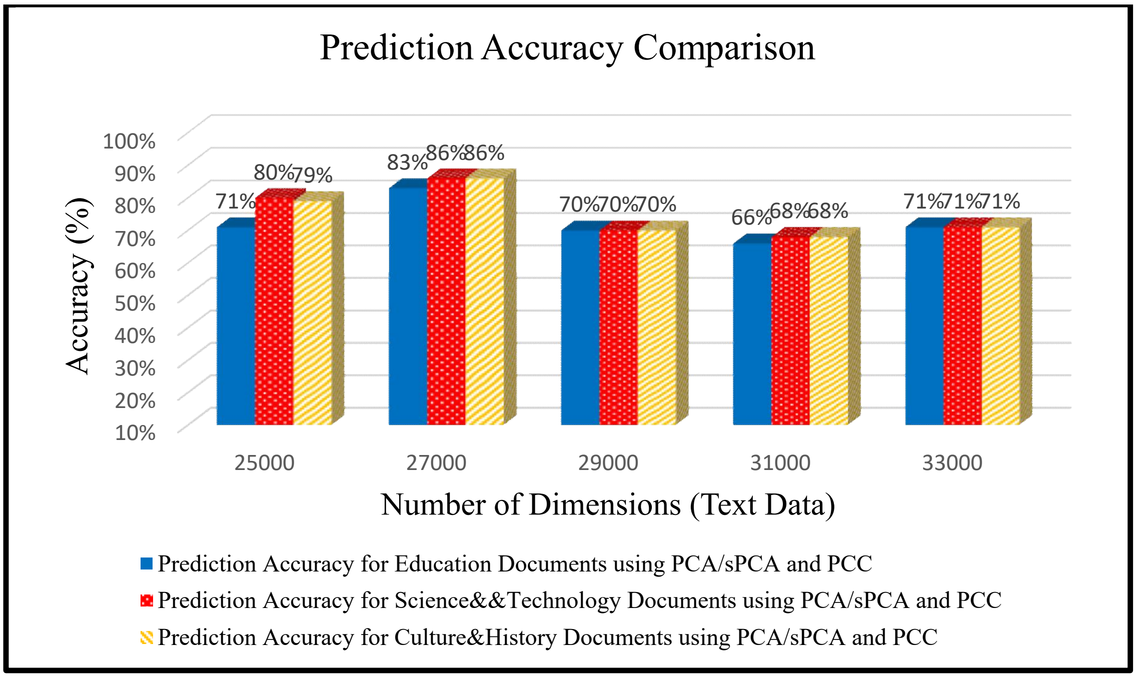


Figure 6.5 Comparison of Prediction Accuracy for Text Data using PCA/sPCA and PCC (Three Document Types)

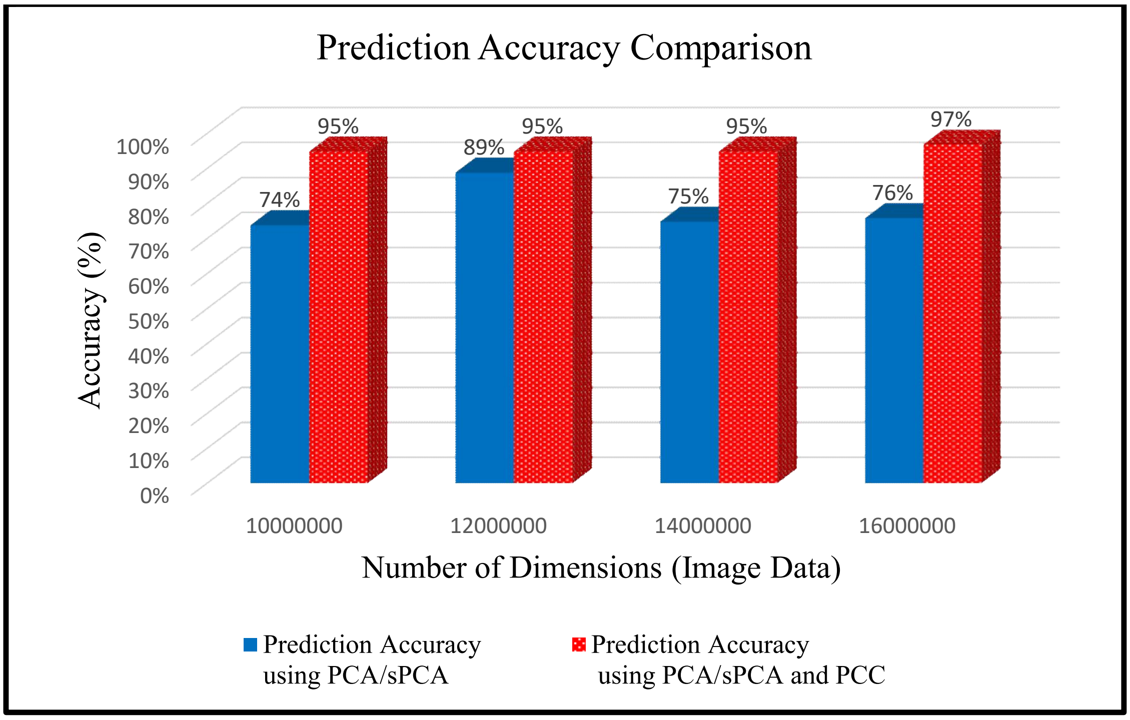


Figure 6.6 Comparison of Prediction Accuracy for Image Data

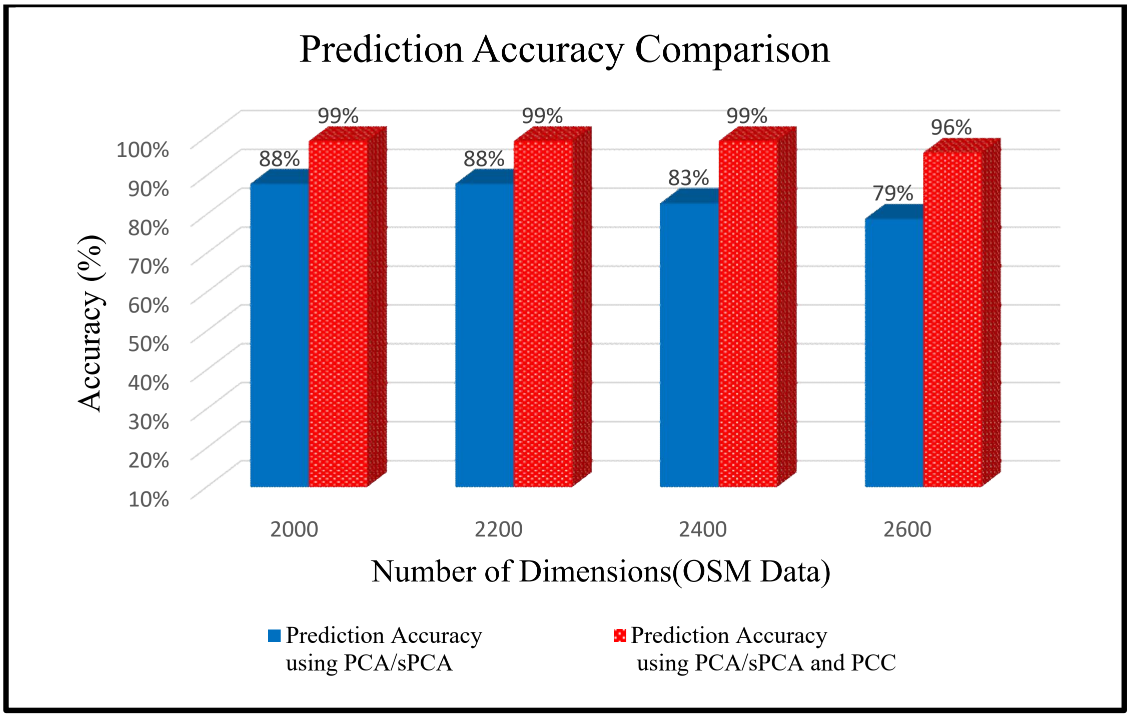


Figure 6.7 Comparison of Prediction Accuracy for OSM Data

6.7 Execution Time Comparison for Dimension Reduction Approaches

The comparative studies of execution time between dimension reduction using PCA and two-stage approach (PCA and PCC) are described in Table 6.29 and Figure 6.8 respectively. According to these table and figure, the execution time (in seconds) taken for only PCA process is longer than that of PCA and PCC combined approach. Although the two-stage approach has the challenge of more processing stage than PCA only, it can minimize the execution time as much as it can. Moreover, PCC has the opportunity to obtain the reduced number of dimensions from PCA to be processed. Therefore, it can perform dimension reduction process with faster execution time. It can also be analyzed that the difference of execution time between two reduction processes is no significant difference in test case 2 and 5 of Table 6.29.

Table 6.29 Execution Time Comparison for Dimension Reduction

Type of Execution	Execution Time (Seconds)				
	Test Case No.				
	1	2	3	4	5
Dimension Reduction with PCA only	10639	15964	17483	20103	20992
Dimension Reduction with PCA and PCC	9381	15361	15953	17479	19928

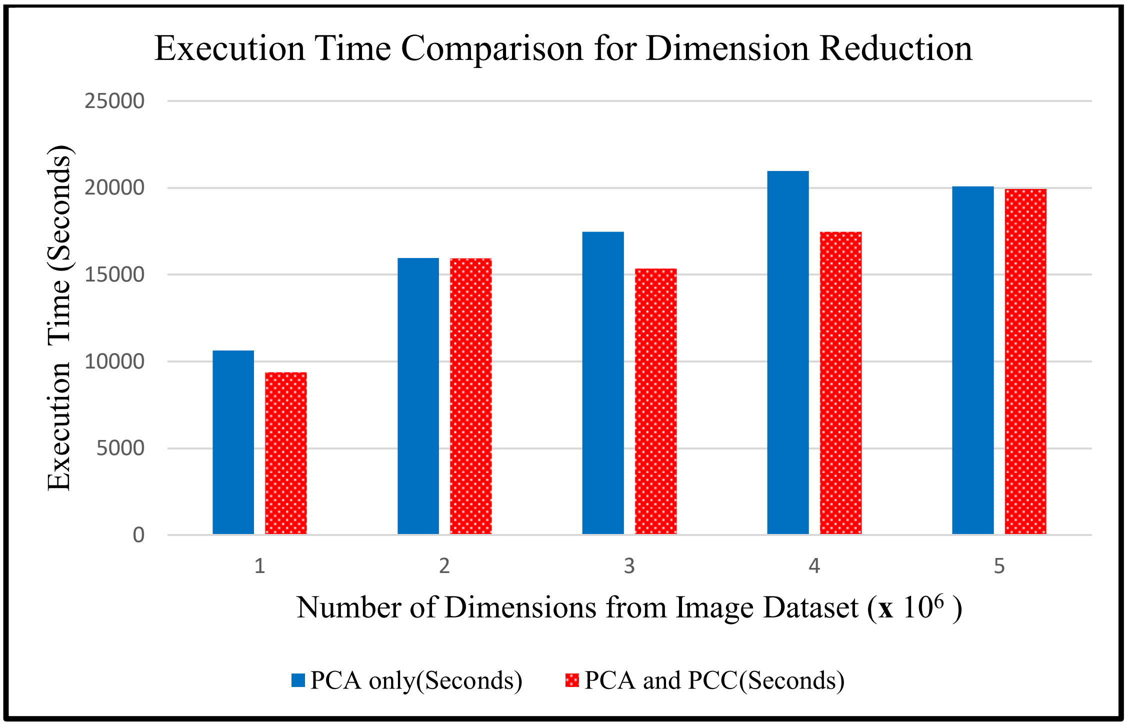


Figure 6.8 Execution time comparison between PCA only, and PCA and PCC

The comparison of execution time between dimension reduction by sPCA only and dimension reduction by two-stage approach (sPCA and PCC) is also described in Table 6.30 and Figure 6.9 respectively. By analyzing the results, the execution time (in seconds) for sPCA and PCC combined approach is faster than that of sPCA only process.

Table 6.30 Execution Time Comparison for Dimension Reduction with sPCA

Type of Execution	Execution Time (Seconds)				
	Test Case No.				
	1	2	3	4	5
Dimension Reduction with sPCA only	11300	11784	13285	16248	19026
Dimension Reduction with sPCA and PCC	7967	8448	10134	12398	15953

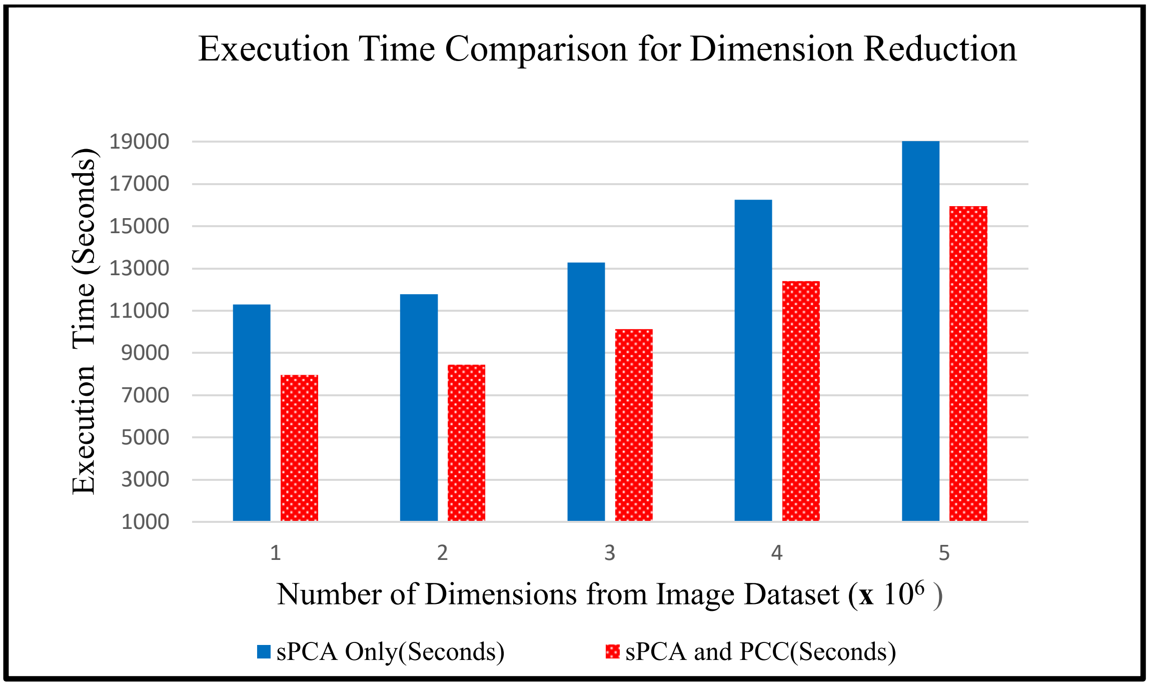


Figure 6.9 Execution time comparison between sPCA only, and sPCA and PCC

6.8 Execution Time Comparison for Matrix Decomposition

The comparative studies of execution time between the proposed system with “QR Decomposition” and the proposed system without “QR Decomposition” for the MLR model are described in Table 6.31 and Figure 6.10 respectively. The execution time (in seconds) taken for without “QR Decomposition” approach is longer than that of with “QR Decomposition” approach. Moreover, it can also be analyzed significantly that execution time difference between two approaches is approximately double in test case 3 and 4 of Table 6.31.

Table 6.31 Execution Time Comparison for Matrix Decomposition

Type of Execution	Execution Time (Seconds)				
	Test Case No.				
	1	2	3	4	5
The system without QR Decomposition	7673	8736	11180	13940	14238
The system with QR Decomposition	5411	5946	6066	6223	12440

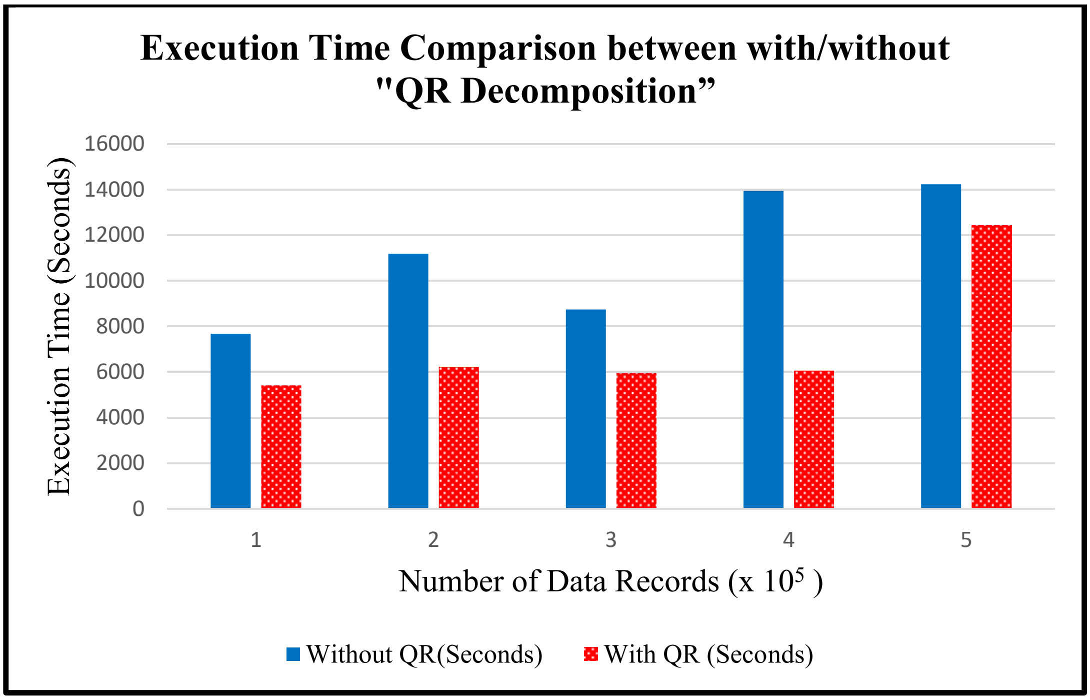


Figure 6.10 Execution time comparison between with and without “QR” Decomposition

6.9 Findings and Discussion

In this section, the comparisons are made between the existing or previous works and the research works upon three main factors. They are:

- PCA in Distributed Settings
  Although, PCA is considered on distributed platform, it is not contributed to design in scalable manner in existing works. However, in research works, designing scalable PCA (sPCA) in distributed platform is proposed by applying some optimizations for distributed processing.
- Feature Section in Regression Model
  In existing works, traditional feature selection methods are applied either irrelevant or redundant features removal for the MLR model. In research works, two-stage dimension reduction approach is proposed for not only irrelevant but also redundant features removal for the MLR model.
- Multiple Linear Regression in Massive Data Processing
  In existing works, it is claimed that traditional MLR model actually need to adapt in massive data processing solving extremely large matrix operations. "QR Decomposition" is applied for traditional MLR model on distributed platform in research works.

6.10 Chapter Summary

In this chapter, the results of dimension reduction from traditional PCA and the proposed sPCA are presented in detail. To analyze obviously the prediction accuracy of the MLR model, the comparisons are made between the actual and predicted number of respective outcomes using many tables and graphs. The comparisons of execution time are also made between one-stage dimension reduction using PCA/sPCA and two-stage dimension reduction using PCA/sPCA and PCC. In addition, the "QR Decomposition" for the MLR model is discussed to realize that the proposed system using this decomposition approach enables to split the voluminous matrix data with decreased execution time. Furthermore, some findings between existing and the proposed works are also discussed upon three main factors. According to experimental results of the system, the proposed two-stage dimension reduction approach can efficiently perform in different sizes of data dimensions which come from diverse data sources. Moreover, the prediction accuracy of the system resulted from the MLR model is actually increased after applying the proposed two-stage approach. Although there are two stages to perform in dimension reduction process in this research, the execution time can be reduced at an acceptable level. Therefore, the proposed system provides the satisfactory results in dimension reduction, and faster execution time not only in dimension reduction process but also in matrix decomposition process for MLR model.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Nowadays, data is extremely growing very fast to become “BIG DATA”, any voluminous amount of structured, semi-structured and unstructured data, which has high potential to be mined for valuable information in decision making process. Analyzing on big data using traditional data analysis methods in extracting valuable information has also become essential in data analytics research. These methods need to adapt in high-performance analytical systems running on distributed platforms which provide scalability and flexibility. According to unpredictable volume of data with a variety of formats, addressing high-dimensionality is absolutely essential in building effective statistical models. In constructing efficient multiple linear regression model, redundant and irrelevant features are highly potential to increase noises and biases which can hinder the prediction process of the model. This research work mainly focuses to develop predictive analysis system using diverse high-dimensional big data sources on distributed platform.

7.1 Summary of Thesis

Today, high-dimensional data analytics has been a great attention in big data era because dimensions of datasets are continuously growing in size. It creates a critical issue to reduce and identify efficiently into a subset of dimensions or features. This subset of dimensions is very potential to fulfill valuable information in decision making. The purposes of dimensionality reduction are to find out how many dimensions can be reduced from all diverse and raw data dimensions and to solve “Curse of Dimensionality”, a big problem in high-dimensional big data analytics caused by increasing number of dimensions. Moreover, traditional dimensionality reduction algorithms which are designed to work with small-scale data often face scalability bottleneck when they are applied to big data. These algorithms are needed to transform distributed and scalable version running on distributed platforms. Principal Component Analysis (PCA) could be applied as a dimensionality reduction machine learning algorithm when dealing with high-dimensional data. The key advantage of PCA is finding the patterns in the data and compressing the data by reducing the number of dimensions without much loss of information.

With increasing volumes of data, the traditional PCA approaches could not be applied to big data. It absolutely needs to transform as scalable PCA (sPCA) on distributed platform such as memory-based spark framework in solving storage burden and obtaining fast results. In designing sPCA on distributed platforms, some optimizations such as minimizing size of intermediate data and fixing large-scale matrix operations including matrix-by-matrix multiplication are considered. Multiple linear regression, a statistical model, describes a linear relationship between a dependent variable and a set of independent variables widely used for prediction and forecasting. The ultimate goal of regression model is that filtering the most important and correlated independent variables with dependent variable for better prediction accuracy. It faces serious problem when "Multicollinearity" exists among the independent variables. Applying sPCA for high-dimensions of big data can reduce less significant or sometimes noisy variables for the multiple regression model. However, it still remains an issue to examine the reduced feature subset resulted from sPCA stage whether correlated or not with the output variable of multiple regression model. In general, Pearson's Correlation Coefficient (PCC) is applied between a pair of variables (X, Y) to measure a linear relationship between these variables.

In this research, two-stage dimension reduction approach is proposed for the multiple linear regression model. Firstly, sPCA was applied to reduce features (dimensions) comprising in matrix form data avoiding multicollinearity problem in regression model. And then, the correlation applied by PCC between a feature and the target class (predicted output) is also considered for the model. Although the high dimensions of input voluminous matrix have been reduced, it is still a big issue to solve how to split or decompose this voluminous matrix still containing increasing size of observations or data records in computing the regression model parameter "β". In solving the values of "β", it can be hardly possible to process the entire huge input matrix at once. Therefore, "QR Decomposition" of a given large-scale matrix X is utilized to decompose X into a product X= QR of an orthogonal matrix Q and an upper triangular matrix R.

7.2 Scope and Limitations

In this research, the predictive analysis system using diverse high-dimensional big data sources is implemented on distributed platform. Firstly, geospatial big data, OpenStreetMap in XML format (OSM XML) is used to obtain "One-way Roads Prediction" results. Raw OSM XML data exists in the form of data structures such as nodes, ways and relations. It has to transform step-by-step to obtain a suitable format which is compatible with big data processing platforms such as MapReduce and Spark. Secondly, high-resolution or high-dimensional representation of images are utilized from MS-Celeb-A, a large-scale face attributes dataset for "Number of Faces Prediction" results. However, to obviously know the accuracy of the results, OpenCV's Haar Cascade Classifier is further applied in face detection process to get how many faces are obtained from input images. Moreover, the correlation between output "face" and a variety of objects (face/non-face and others) existing in images is discovered by using PCC. Therefore, it has to prepare training data which consists of a lot of samples of faces and non-faces and, other procedures for locating them.

Finally, the raw, unstructured text data via "DeliciousMIL" dataset from UCI is applied as input text documents. This dataset consists of a subset of tagged web pages from the social bookmarking site "delicious.com". Therefore, HTML" Tags Removal and other pre-processing procedures are required to obtain a suitable format which is compatible with big data processing platforms. For "Number of Documents (Education, Science&&Technology, Culture&&History) Prediction" results, PCC is also used between two variables; one for "term" and other for "target" to achieve term and target correlation. Moreover, to find the pair-wise correlation between the terms and the target in each document, TF*IDF process is applied for weight value assigning to each term in text documents. With the purpose of better accuracy for training data, in this research, a number of diverse pre-processing procedures are needed to perform as three input data sources are different in size, structure, and ways of data collection. There are many efforts to achieve the same structured numerical matrix form data as input data matrix for sPCA. Furthermore, the final prediction results from the model are visualized as maps, images with face detected regions, and pie charts with the purpose of achieving obvious prediction results for the system. In this research, the proposed system is intended to show that it can efficiently perform on diverse high-dimensional big data sources on distributed platform. However, there are some limitations to fix as follows:

- Different pre-processing procedures for respective data sources are required to perform
- Unique numerical, structured data format is needed to transform for all input data sources
- Suitable tools and libraries for the system such as “OpenCV” in face detection process and “Osmosis”, “QGIS” in roads prediction are needed to perform

7.3 Results and Conclusion

According to the experimental results, it can be analyzed that the proposed sPCA can efficiently perform dimension reduction process with increasing size or number of dimensions for diverse data sources. It also shows the good scalability performance while the traditional PCA offers “Out of Memory” results. After applying not only sPCA but also PCC for three different data matrices, the predicted number of outputs for respective types of predictions are resulted from the MLR model. The percentage of prediction accuracies after applying sPCA and PCC are significantly higher than applying sPCA only. It can be claimed that “Number of One-way Roads Prediction” using osm data achieves the best prediction accuracy than other types of prediction. “Number of Documents Prediction” using text data obtain good prediction accuracies up to 86 percent (%). As for “Number of Faces Prediction” using images, it obtains better prediction accuracies up to 97 percent (%). Moreover, “Number of One-way Roads Prediction” reaches the victory of accuracy in 99 percent (%). Therefore, it can also be proved that applying the proposed two-stage approach (sPCA and PCC) offers the optimal prediction accuracy by reducing the redundant and irrelevant dimensions or features from the input data matrix before MLR model.

Although the proposed two-stage approach has the challenge of more processing stage than sPCA only, it can minimize the execution time as much as it can. In addition, the second stage, PCC has a good opportunity to receive the reduced number of dimensions from sPCA to be processed. Therefore, it can perform dimension reduction process with faster execution time. Furthermore, the execution time taken for the proposed system without applying “QR Decomposition” is significantly longer than that of with “QR Decomposition”. By using “QR

Decomposition" approach providing traditional MLR model for the proposed system, it offers reduced execution time indeed. According to the evaluation analysis, it can be concluded that the proposed system provides good scalability, prediction accuracy, and execution time in predictive analytics on high-dimensional big data.

7.4 Future Work

The research works in this thesis are mainly considered on dimension reduction in association with correlation approach for the MLR model. For future works, without applying sPCA and PCC, other approaches can be considered to deal with high-dimensional big data. The proposed system is implemented on Multi Node Cloudera Cluster including Cloudera Manager as a distributed big data processing platform. Therefore, other high-performance processing platforms can be replaced to implement the system to perform the comparative studies between execution time, scalability, and so on. In this research, the validity of the proposed two-stage dimension reduction approach can be checked by prediction outcomes from the MLR model. As future work, the validity checking of the proposed approach can be considered early state of the system before the prediction process of MLR model. Moreover, the MLR model is utilized as prediction model for the proposed system in this research. By using other prediction models in collaboration with the dimension reduction approach, it can also be examined whether a good solution compared with the existing works.

AUTHOR'S PUBLICATIONS

[P1] K. L. L. Khine, T. T. S. Nyunt, “Big Data Analytics for Price Prediction”, In 14th International Conference on Computer Applications (ICCA, 2016), Yangon, MYANMAR, pp. 279-283, February 2016.

[P2] K. L. L. Khine, T. T. S. Nyunt, “Big Data Analytics for Rainfall Prediction Using MapReduce Based Regression Approach”, In 15th International Conference on Computer Applications (ICCA, 2017), Yangon, MYANMAR, pp. 48-54, February 2017.

[P3] K. L. L. Khine, T. T. S. Nyunt, “Statistical Data Analysis for Rainfall Prediction Using Multiple Linear Regression Approach”, In Asean Science Technology and Innovation Conference 2017, Naypyidaw, MYANMAR, October 2017.

[P4] K. L. L. Khine, T. T. S. Nyunt, “Predictive Big Data Analytics using Multiple Linear Regression Model", In 1st International Conference on Big Data Analysis and Deep Learning Applications (ICBDL), Springer (Scopus Index), Miyazaki, JAPAN, May 2018.

[P5] K. L. L. Khine, T. T. S. Nyunt, “Predictive Analytics on High-Dimensional Big Data using Principal Component Regression (PCR)”, In 11th International Conference on Future Computer and Communication (ICFCC), Scopus Index, Yangon, MYANMAR, pp. 148-153, February 2019.

[P6] K. L. L. Khine, T. T. S. Nyunt, “Predictive Geospatial Analytics using Principal Component Regression (PCR)”, In International Journal of Electrical and Computer Engineering (IJECE), Scopus Index - Q2, INDONESIA, Vol. 10, No. 3 (Part I), June 2020.

[P7] K. L. L. Khine, T. T. S. Nyunt, “Predictive Big Data Analytics using Multiple Linear Regression Model”, In Advances in Intelligent Systems and Computing (Book Series Volume: 744), Springer (Scopus Index – Q3), pp. 9-19, 2019.

BIBLIOGRAPHY

[1] M.R.Adjout, B.F, "A massively parallel processing for the multiple linear regression", In Proceedings of 10th International Conference on Signal-Image Technology & Internet-Based Systems, November 2014, pp. 666-671.

[2] E.Ahmed, M.F.Mohamed, "Spatialhadoop: A mapreduce framework for spatial data", In Proceedings of 31st International Conference on Data Engineering, IEEE, April 2015, pp. 1352–136.

[3] M.C.Alibuhtto, T.S. Peiris, "Principal component regression for solving multicollinearity problem", 2015.

[4] S.Aluru, B.Ganapathysubramanian, J.Zola, and S.K.Samudrala, "Parallel Framework for Dimensionality Reduction of Large-Scale Datasets, Mechanical Engineering Publications", January 2015.

[5] A.Amerah, B.Michela, "Assessing OSM street semantics quality using context and user trustworthiness", 2017.

[6] S.Banumathi, A.Aloysius, "PREDICTIVE ANALYTICS CONCEPTS IN BIG DATA-A SURVEY", In International Journal of Advanced Research in Computer Science, Vol. 8, No. 8, September 2017.

[7] P.Belhumeur, J.Hespanha, and D.Kriegman, "Eigenfaces v.s. fisherfaces: Recognition using class specific linear projection", IEEE transactions on pattern analysis and machine learning, Vol. 19, No. 7, July 1997, pp. 711–720.

[8] C.Biswanath, M.Anirban, B.Siddhartha, and C.Manojit, "Towards reliable clustering of english text documents using correlation coefficient", In Proceedings of International Conference on Computational Intelligence and Communication Networks, IEEE, November 2014, pp. 530-535.

[9] M.Brovelli, G.Zamboni, "A new method for the assessment of spatial accuracy and completeness of OpenStreetMap building footprints", In ISPRS International Journal of Geo-Information, Vol. 7, No. 8, August 2018.

[10] F.A.Castro, M.Sester, and S.Winter, "Geospatial big data handling theory and methods: A review and research challenges", In ISPRS journal of

Photogrammetry and Remote Sensing, Vol. 115, May 2016, pp. 119-133.

[11] L.D.David, "High-dimensional data analysis: The curses and blessings of dimensionality", AMS math challenges lecture 1, 2000.

[12] S.Dray, J.Josse, “Principal component analysis with missing values: a comparative survey of methods. Plant Ecology”, Vol. 216, No. 5, May 2015, pp. 657-667.

[13] A.Dutta, R.Veldhuis, and L.Spreeuwers, “Predicting face recognition performance using image quality”, 2015.

[14] T.Elgamal, H.Mohamed, "Analysis of PCA algorithms in distributed environments”, 2015.

[15] T.Elgamal, Y.Maysam, M.Waleed, and H.Mohamed, "sPCA: Scalable principal component analysis for big data on distributed platforms", In Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM, May 2015, pp. 79-91.

[16] T.Eswari, P.Sampath, and S.Lavanya, "Predictive methodology for diabetic data analysis in big data", Procedia Computer Science, 2015, pp. 203-208.

[17] T.H.Fan, K.F.Cheng, “Regression analysis for massive datasets”, In Proceedings of Data Knowl. Eng, Vol. 61, 2007, pp. 554–562.

[18] T.H.Fan, K.F.Cheng, “Tests and variables selection on regression analysis for massive datasets”, In Proceedings of Data & Knowledge Engineering, Vol. 63, No. 3, December 2007, pp. 811-819.

[19] U.Feuerhake, O.Wage, M.Sester, N.Tempelmeier, W.Nejdl, and E. Demidova, "Identification of similarities and prediction of unknown features in an urban street network", In International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives 42, 2018, pp. 261-266.

[20] C.Florina, G.Elena, “Perspectives on Big Data and Big Data Analytics”, 2013.

[21] D.P.Foster, M.Liberman, and R.A.Stine, “Featurizing text: Converting text into predictors for regression analysis”, The Wharton School of the University of Pennsylvania, Philadelphia, PA, 2013.

[22] S.Funke, S.Robin, and S.Sabine, "Automatic extrapolation of missing road network data in OpenStreetMap", In Proceedings of the 2nd International

Conference on Mining Urban Data, Vol. 1392, July 2015, pp. 27-35.

[23] A.Gandomi, H.Murtaza, "Beyond the hype: Big data concepts, methods, and analytics", In International journal of information management, 2015, pp. 137-144.

[24] A.Golubev, C.Ilya, P.Danila, S.Alexander, and S.Maxim, “Geospatial data generation and preprocessing tools for urban computing system development”, Procedia Computer Science 101, 2016, pp. 217-226.

[25] J.C.Gomez, M.F.Moens, “PCA document reconstruction for email classification. Computational Statistics & Data Analysis”, Vol. 56, No. 3, March 2012, pp. 741-751.

[26] H.Goyal, S.Chilka, and J.Nisheeth, "An integrated approach of GIS and spatial data Mining in Big Data", In International Journal of Computer Application, Vol. 169, No. 11, July 2017, pp. 1-6.

[27] M.Hacar, K.Batuhan, and Ş.Kadir, "Analyzing openstreetmap road data and characterizing the behavior of contributors in Ankara, Turkey" In ISPRS International Journal of Geo-Information, Vol. 7, No. 10, October 2018.

[28] M.Haklay and P.Weber, “OpenStreetMap: User-Generated Street Maps,” IEEE Pervasive Computing, Vol. 7, No. 4, October 2008, pp. 12-18.

[29] M.Helbich, A.Chritoph, and Z.Alexander, "Comparative spatial analysis of positional accuracy of OpenStreetMap and proprietary geodata", In Proceedings of GI_Forum, July 2012, pp. 24-33.

[30] H.Hotelling, “Analysis of a complex of statistical variables into principal components”, In Journal of educational psychology, Vol. 24, No. 6, 1993.

[31] J.L.Hou, C.A.Chuan, "A Document Content Extraction Model Using Keyword Correlation Analysis", IJEBM 1, Vol. 1, 2003, pp. 54-62.

[32] M.Hubert, S.Verboven, “A robust PCR method for high-dimensional regressors”, In Journal of Chemometrics: A Journal of the Chemometrics Society, Vol. 17, No. 8-9, August 2003, pp. 438-452.

[33] R.A.Jandarov, L.A.Sheppard, and P.D.Sampson, “A novel principal component analysis for spatially misaligned multivariate air pollution data”, In Journal of the Royal Statistical Society: Series C (Applied Statistics), Vol. 66, No. 1, January 2017, pp. 3-28.

[34] C.Jingdong, H.Yiteng, “On the importance of the Pearson correlation

coefficient in noise reduction", IEEE Transactions on Audio, Speech, and Language Processing, Vol. 16, No. 4, April 2008, pp. 757-765.

[35] I.Jolliffe, "Principal component analysis. International Encyclopedia of Statistical Science", 2011, pp. 1094-1096.

[36] I.Jolliffe, J.Cadima, "Principal component analysis: a review and recent developments", Adaptive data analysis: theory and applications, January, 2016.

[37] J.Joo, S.F.Francis, and Z.C.Song, "Automated facial trait judgment and election outcome prediction: Social dimensions of face", In Proceedings of the IEEE international conference on computer vision, 2015, pp. 3712-3720.

[38] S.Jun, R.B.Jea, "A divided regression analysis for big data." In International Journal of Software Engineering and Its Applications, Vol. 9, No. 5, 2015, pp. 21-32.

[39] P.Kaiser, W.D.Jan, and S.Konrad, "Learning aerial image segmentation from online maps", IEEE Transactions on Geoscience and Remote Sensing, Vol. 55, No. 11, July 2017, pp. 6054-6068.

[40] S.Krishnamurthy, "Dealing with high-dimensionality in large data sets", 2011.

[41] T.Krištof, Š.Vanja, and F.Mario, "Textual Content and Engagement Correlation Analysis with Naive Bayes", In International Journal of Digital Technology & Economy, Vol. 2, No. 1, September 2017, pp. 1-12.

[42] A. Kumar, S. Chandrasekhar, "Text data pre-processing and dimensionality reduction techniques for document clustering", In International Journal of Engineering Research and Technology (IJERT), Vol. 1, July 2012.

[43] J.G.Lee, K.Minseo, "Geospatial big data: challenges and opportunities", Big Data Research 2, Vol. 2, No. 2, June 2015, pp. 74-81.

[44] Y.K.Lee, B.U.Park, "Principal Component Analysis In Very High-Dimensional Spaces", 2010.

[45] K.W.Lee, J.Jo, "High-performance geospatial big data processing system based on MapReduce", In ISPRS International Journal of Geo-Information, Vol. 7, No. 10, October 2018.

[46] R.K.Lenka, N.Gupta, and H.Dubey, "Comparative analysis of SpatialHadoop and GeoSpark for geospatial big data analytics", In

Proceedings of 2nd International Conference on Contemporary Computing and Informatics (IC3I), IEEE, December 2016, pp. 484-488.

[47] S.Lhazmir, M.E.Ismail, and K.Abdellatif, "Feature extraction based on principal component analysis for text categorization", In Proceedings of International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), IEEE, November 2017, pp. 1-6.

[48] H.Londögård, L.Hannah, "Improving the OpenStreetMap Data Set using Deep Learning", 2018.

[49] C.Meng, W.Ye, and M.Ping, "Effective statistical methods for big data analytics", In Handbook of Research on Applied Cybernetics and Systems Science, IGI Global, 2017, pp. 280-299.

[50] M.Mudrova, P.Aleš, "Principal component analysis in image processing", In Proceedings of the MATLAB Technical Computing Conference, Prague, November 2005.

[51] A.Mustapha, A.Abdu, "Application of principal component analysis & multiple regression models in surface water quality assessment", In Journal of environment and earth science, Vol. 2, No. 2, 2012, pp. 16-23.

[52] S.C.Ng, "Principal component analysis to reduce dimension on digital image." Procedia computer science 111, 2017, pp. 113-119.

[53] S.A.Nsang, B.Musa, and S.Hammed, "Image Reduction Using Assorted Dimensionality Reduction Techniques", MAICS, 2015, pp. 139-146.

[54] H.C.Park, "Dimension reduction using least squares regression in multi-labeled text categorization", In Proceedings of 8th IEEE International Conference on Computer and Information Technology, IEEE, July 2008, pp. 71-76.

[55] A.Pawar, M.Vijay, "Calculating the similarity between words and sentences using a lexical database and corpus statistics", 2018.

[56] V.D.Rojatkar, B.D.Nitesh, and N.R.Peddiwar, "Image Compression Techniques, Lossy and Lossless", In International Journal of Engineering Research and General Science, Vol. 3, No. 2, 2015.

[57] S. Shekhar, "Spatial big data challenges", Keynote at ARO/NSF Workshop on Big Data at Large: Applications and Algorithms, Durham, NC, 2012.

[58] L.C.Sabharwal, A.Bushra, "Data Reduction and Regression Using Principal

Component Analysis in Qualitative Spatial Reasoning and Health Informatics”, In Polibits, No. 53, June 2016, pp. 31-42.

[59] N.Y.Sandhya, A.Govardhan, and K.Anuradha, "Analysis of similarity measures for text clustering", In CSC Journals, 2008.

[60] Y.Saeys, I.Iñaki, and L. Pedro, "A review of feature selection techniques in bioinformatics”, Vol. 23, No. 19, October 2007, pp. 2507-2517.

[61] C.Sarkar, "Improving Predictive Modeling in High Dimensional, Heterogeneous and Sparse Health Care Data”, 2015.

[62] P.Semberecki, M. Henryk, "Distributed classification of text documents on Apache Spark platform", In Proceedings of International Conference on Artificial Intelligence and Soft Computing, Springer, 12 June 2016, pp. 621-630.

[63] V.B.Shereena, M.D.Julie, "Significance of dimensionality reduction in image processing", In Signal & Image Processing, An International journal (SIPIJ), Vol. 6, 2015.

[64] A.D.Singh, W.Wei, and K.Shonali, "Taxi trip time prediction using similar trips and road network data", In Proceedings of IEEE International Conference on Big Data (Big Data), IEEE, 1 October 2015, pp. 2892-2894.

[65] M.Singh, K.Sushil, and S.Manish, "Various Image Compression Techniques: Lossy and Lossless", In International Journal of Computer Applications, Vol. 142, No. 6, May 2016, pp. 0975-8887.

[66] H.Soleimani, D.J.Miller, “Semi-supervised multi-label topic models for document classification and sentence labeling”, In Proceedings of the 25th ACM international on conference on information and knowledge management, ACM, 24 October 2016, pp. 105-114.

[67] J.H.Stephen, H.T.Owen, et al., “Predicting Students' Academic Performance Using Multiple Linear Regression and Principal Component Analysis”, In Journal of Information Processing, Vol. 26, 2018, pp. 170-176.

[68] L.Tang, L.Zhou, “Method of divide-and-combine in regularized generalized linear models for big data”, 2016.

[69] J.Thomson, L.Hugh, “Predicting and optimizing image compression”, In Proceedings of the 24th ACM international conference on Multimedia, ACM, October 2016, pp. 665-669.

[70] A.Z.Ul-Saufie, A.S.Yahya, and N.A.Ramli, "Improving multiple linear regression model using principal component analysis for predicting PM10 concentration in Seberang Prai, Pulau Pinang", In International Journal of Environmental Sciences, Vol. 2, No. 2, 2011, pp. 403-409.

[71] N.Untari, Wisesty, et al., "Dimensionality Reduction using Principal Component Analysis for Cancer Detection based on Microarray Data Classification", In Journal of Computer Science, Vol. 14, No. 10, 2018, pp. 1521-1530.

[72] G.D.Underhill, K.McDowell, and L.Solka, "Enhancing text analysis via dimensionality reduction", In Proceedings of IEEE International Conference on Information Reuse and Integration, IEEE, August 2007, pp. 348-353.

[73] M.Verleysen, F.Damien, "The curse of dimensionality in data mining and time series prediction", In Proceedings of International Work-Conference on Artificial Neural Networks, Heidelberg, Berlin, Springer, June 2005, pp. 758-770.

[74] R.M.Vidhyavathi, "Principal component analysis (PCA) in medical image processing using digital imaging and communications in medicine (DICOM) medical images", 2017.

[75] C.Wang, C.Ming-Hui, and Y.Jun, "Statistical methods and computing for big data", Statistics and its interface, Vol. 9, No. 4, 2016.

[76] G.Wang, S.Qinbao, and Z.Yuming, "Selecting feature subset for high dimensional data via the propositional FOIL rules", Pattern Recognition, Vol. 46, No. 1, January 2013, pp. 199-214.

[77] S.Wang, H.Yuan, "Spatial data mining: a perspective of big data", In International Journal of Data Warehousing and Mining (IJDWM), Vol. 10, No. 4, October 2014, pp. 50-70.

[78] J.Weng, D.S.Young, "Some dimension reduction strategies for the analysis of survey data", In Journal of Big Data, Vol. 4, No. 1, December 2017.

[79] F.Xu, Y.L.Bill, H.Yifei, and Q.Z.Kenny, "Cross-region traffic prediction for china on openstreetmap", In Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science, 31 October 2016, pp. 37-42.

[80] B.Yang, T.Zhang, "Big Data Dimension Reduction using PCA", In

Proceedings of IEEE International Conference on Smart Cloud, IEEE, November 2016, pp. 152-157.

[81] M.M.Yagoub, “Assessment of OpenStreetMap (OSM) Data: The Case of Abu Dhabi City, United Arab Emirates”, In Journal of Map & Geography Libraries, Vol. 13, No. 3, September 2017, pp. 300-319.

[82] L.Yu, L.Huan, "Feature selection for high-dimensional data: A fast correlation-based filter solution", In Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003, pp. 856-863.

[83] R.O.Zaïane, A.Maria-Luiza, "Classifying text documents by associating terms with text categories", Australian computer Science communications, Australian Computer Society, Inc., Vol. 24, No. 2, 2002, pp. 215-222.

[84] D.Zhang, C.Songcan, and L.Jun, "Representing image matrices: eigenimages versus eigenvectors", In International Symposium on Neural Networks, Berlin, Heidelberg, Springer, May 2005, pp. 659-664.

[85] J.G.Zhu, Z.Song, “Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data”, IEEE Transactions on Industrial Informatics, Vol. 13, No. 4, January 2009, pp. 1877–1885.

[86] L.Zhao, Y.Yee-Hong, "An efficient algorithm to compute eigenimages in PCA-based vision systems", Pattern recognition, Vol. 32, No. 5, 1999, pp. 851-864.

[87] L.Zhao, K.Paula, "Openstreetmap road network analysis for poverty mapping", 2016.

[88] https://halobi.com/blog/descriptive-predictive-and-prescriptive-analytics-explained/

[89] https://www.dezyre.com/

[90] https://geoawesomeness.com

[91] https://agilestorelocator.com

[92] https://archive.ics.uci.edu

[93] https://www.apache.org

[94] https://www.cloudera.com

[95] https://www.openstreetmap.org

[96] https://www. Learnosm.org

ACRONYMS

API	Application Program Interface
CDH	Cloudera Distribution Hadoop
CF	Category Frequency
CF-DF	Category Frequency - Document Frequency
CSV	Comma Separated Value
1-D	One Dimensional
2-D	Two Dimensional
DF	Document Frequency
GIF	Graphics Interchange Format
GIS	Geographical Information System
GPS	Global Positioning System
HDFS	Hadoop Distributed File System
HDP	HortonWorks Data Platform
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IDF	Inverse of Document Frequency
IP	Internet Protocol
iOS	iPhone OS
JSON	JavaScript Object Notation
LBP	Local Binary Pattern
LSA	Latent Semantic Analysis
ML	Machine Learning
MLlib	Apache Spark's scalable machine learning library
MLR	Multiple Linear Regression
MS-Celeb-A	Microsoft Celebfaces Attributes Dataset (CelebA)
MS-Celeb-1M	Microsoft Celeb Dataset (10 million face images)
OpenCV	Open Source Computer Vision Library
OSM	OpenStreetMap
PBF	Protocolbuffer Binary
PC	Personal Computer

PCA	Principal Component Analysis
PCC	Pearson Correlation Coefficient
PCs	Principal Components
PNG	Portable Network Graphics
QGIS	Quantum Geographical Information System
QR	Orthogonal matrix “Q” and an upper triangular matrix “R”
RAM	Random Access Memory
RDDs	Resilient Distributed Datasets RDDs
RGB	Red Green Blue
RMSE	Root Mean Square Error
sPCA	scalable Principal Component Analysis
SSH	Secure Socket Shell
SSVD	Stochastic Singular Value Decomposition
SVD	Singular Value Decomposition
TB	Terabyte
TF	Term Frequency
VM	Virtual Machine
XML	Extensible Markup Language
YARN	Yet Another Resource Negotiator

Appendix

In this research, the experimentation of the system is evaluated on distributed big data analytics platform called "Cloudera Distribution Hadoop (CDH)". The configuration and installation of Cloudera VM, Cloudera Manager, Multi Node Cluster and additional software components for the implementation are presented in following sections.

Configuration of a Cloudera Cluster

To setup the cluster, firstly, the installation of single Cloudera VM on Oracle Virtual Box is needed to configure. In downloading Oracle Virtual Box, the correct version of Virtual Box with own operating system is required to choose.

- For PC users: Virtual Box 5.0.14 for Windows hosts
- For MAC users: Virtual Box 5.0.14 for OS X hosts

After downloading and installing .exe file for Oracle Virtual Box, Cloudera QuickStart VM installation zip archive file is required to download. To open Cloudera QuickStart VM in Virtual Box, the following procedures are needed to perform.

- Launch Virtual Box
- Click on Import appliance from File tab
- Choose the .ovf file from the location where Cloudera Quickstart VM files are existed
- Click Import to install and start the VM

Installing Cloudera Manager

Before installing Cloudera Manager Server, it is absolutely needed to make sure the following configuration in the host VM:

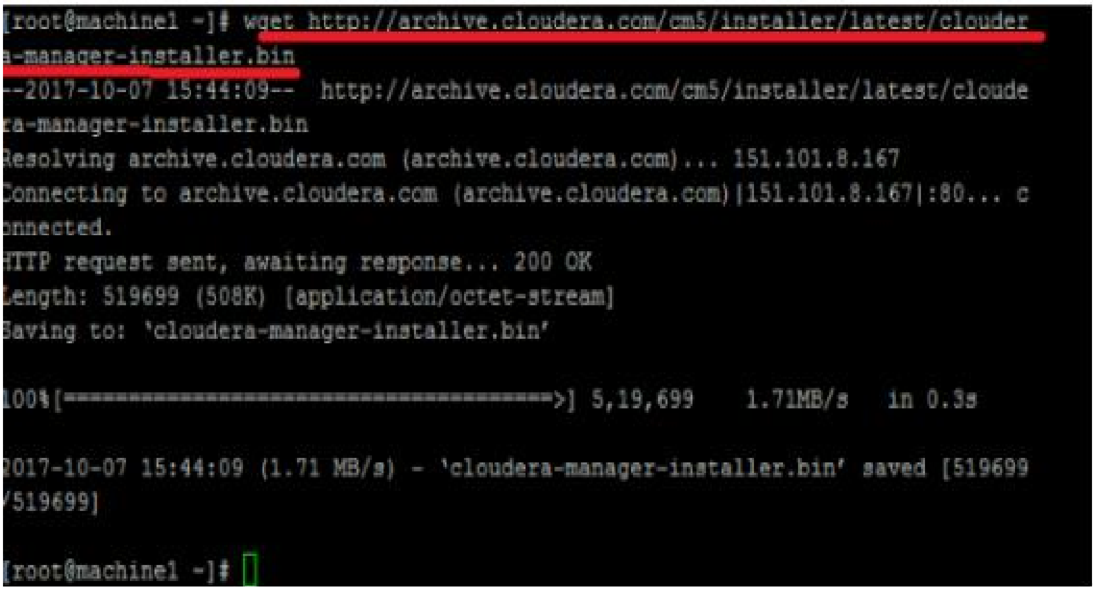
- "SELINUX=disabled" can be modified in /etc/selinux/config file.
- "chkconfig iptables off" is required to disable firewall.
- "vm.swappiness=0" can be modified in /etc/sysctl.conf file.
- "net.ipv6.conf.default.disable_ipv6=1",

"net.ipv6.conf.all.disable_ipv6 = 1" can be modified in /etc/sysctl.conf file.

- It needs to configure passwordless SSH for root user.

After downloading Cloudera Manager Server installer, it needs to login as root on host VM for all installations are under root user. Using the following commands and procedures, the installer is successfully finished to proceed.

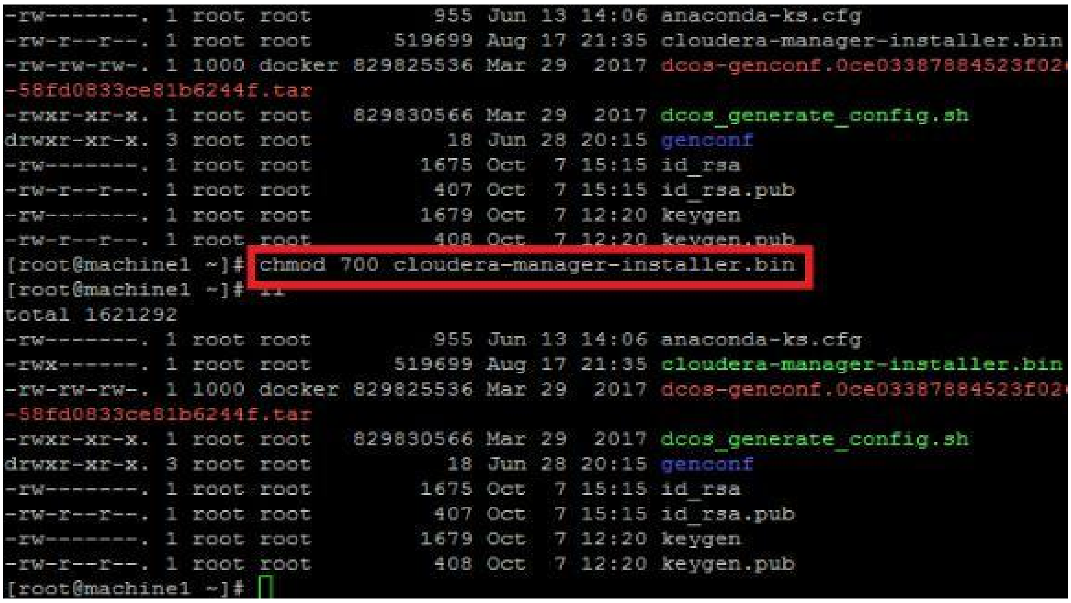
- yum install openssl python perl
- yum clean all
- yum repolist
- wget http://archive.cloudera.com/cm5/installer/latest/cloudera-manager-installer.bin



```
[root@machine1 ~]# wget http://archive.cloudera.com/cm5/installer/latest/clouder
a-manager-installer.bin
--2017-10-07 15:44:09--  http://archive.cloudera.com/cm5/installer/latest/cloude
ra-manager-installer.bin
Resolving archive.cloudera.com (archive.cloudera.com)... 151.101.8.167
Connecting to archive.cloudera.com (archive.cloudera.com)|151.101.8.167|:80... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: 519699 (508K) [application/octet-stream]
Saving to: 'cloudera-manager-installer.bin'

100%[=====================================>] 5,19,699    1.71MB/s   in 0.3s

2017-10-07 15:44:09 (1.71 MB/s) - 'cloudera-manager-installer.bin' saved [519699
/519699]

[root@machine1 ~]#
```

Cloudera Manager Server Installation Step 1

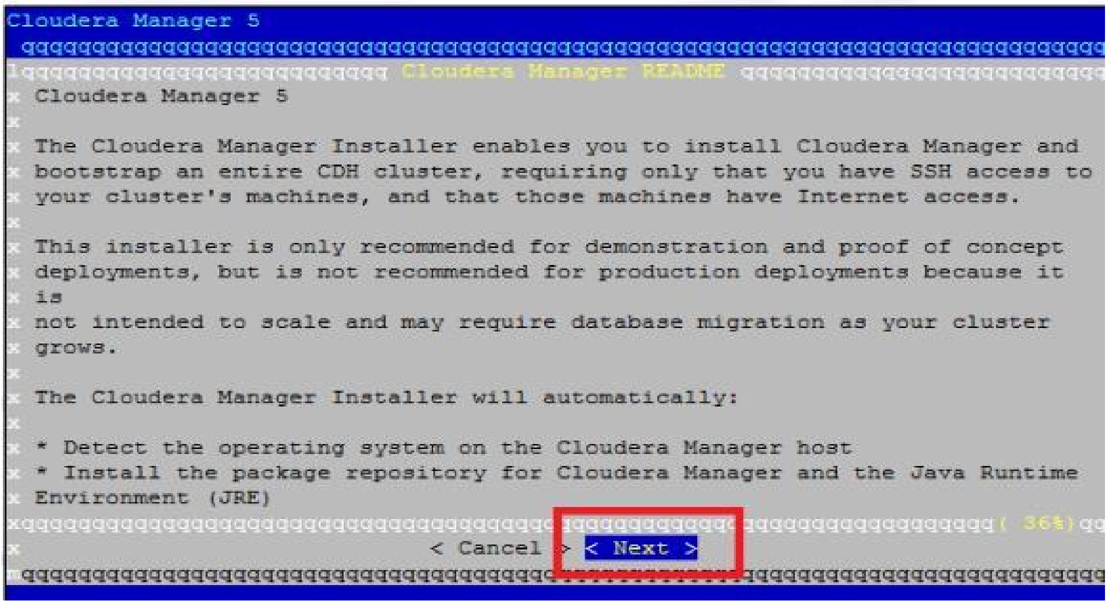
- chmod 700 cloudera-manager-installer.bin



```
-rw-------. 1 root root          955 Jun 13 14:06 anaconda-ks.cfg
-rw-r--r--. 1 root root       519699 Aug 17 21:35 cloudera-manager-installer.bin
-rw-rw-rw-. 1 1000 docker 829825536 Mar 29  2017 dcos-genconf.0ce03387884523f02
-58fd0833ce81b6244f.tar
-rwxr-xr-x. 1 root root    829830566 Mar 29  2017 dcos_generate_config.sh
drwxr-xr-x. 3 root root           18 Jun 28 20:15 genconf
-rw-------. 1 root root         1675 Oct  7 15:15 id_rsa
-rw-r--r--. 1 root root          407 Oct  7 15:15 id_rsa.pub
-rw-------. 1 root root         1679 Oct  7 12:20 keygen
-rw-r--r--. 1 root root          408 Oct  7 12:20 keygen.pub
[root@machine1 ~]# chmod 700 cloudera-manager-installer.bin
[root@machine1 ~]# ll
total 1621292
-rw-------. 1 root root          955 Jun 13 14:06 anaconda-ks.cfg
-rwx------. 1 root root       519699 Aug 17 21:35 cloudera-manager-installer.bin
-rw-rw-rw-. 1 1000 docker 829825536 Mar 29  2017 dcos-genconf.0ce03387884523f02
-58fd0833ce81b6244f.tar
-rwxr-xr-x. 1 root root    829830566 Mar 29  2017 dcos_generate_config.sh
drwxr-xr-x. 3 root root           18 Jun 28 20:15 genconf
-rw-------. 1 root root         1675 Oct  7 15:15 id_rsa
-rw-r--r--. 1 root root          407 Oct  7 15:15 id_rsa.pub
-rw-------. 1 root root         1679 Oct  7 12:20 keygen
-rw-r--r--. 1 root root          408 Oct  7 12:20 keygen.pub
[root@machine1 ~]#
```

Cloudera Manager Server Installation Step 2

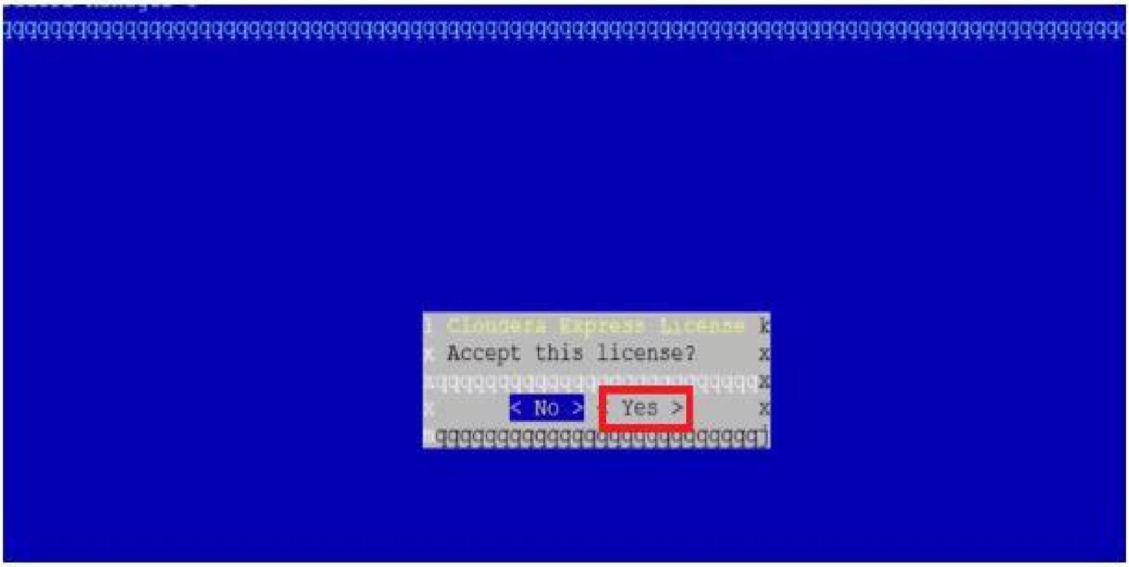
- ./cloudera-manager-insatller.bin



```
Cloudera Manager 5
                         Cloudera Manager README
 Cloudera Manager 5

 The Cloudera Manager Installer enables you to install Cloudera Manager and
 bootstrap an entire CDH cluster, requiring only that you have SSH access to
 your cluster's machines, and that those machines have Internet access.

 This installer is only recommended for demonstration and proof of concept
 deployments, but is not recommended for production deployments because it
 is
 not intended to scale and may require database migration as your cluster
 grows.

 The Cloudera Manager Installer will automatically:

 * Detect the operating system on the Cloudera Manager host
 * Install the package repository for Cloudera Manager and the Java Runtime
 Environment (JRE)
                                                                     ( 36%)
                           < Cancel >  < Next >
```

Cloudera Manager Server Installation Step 3

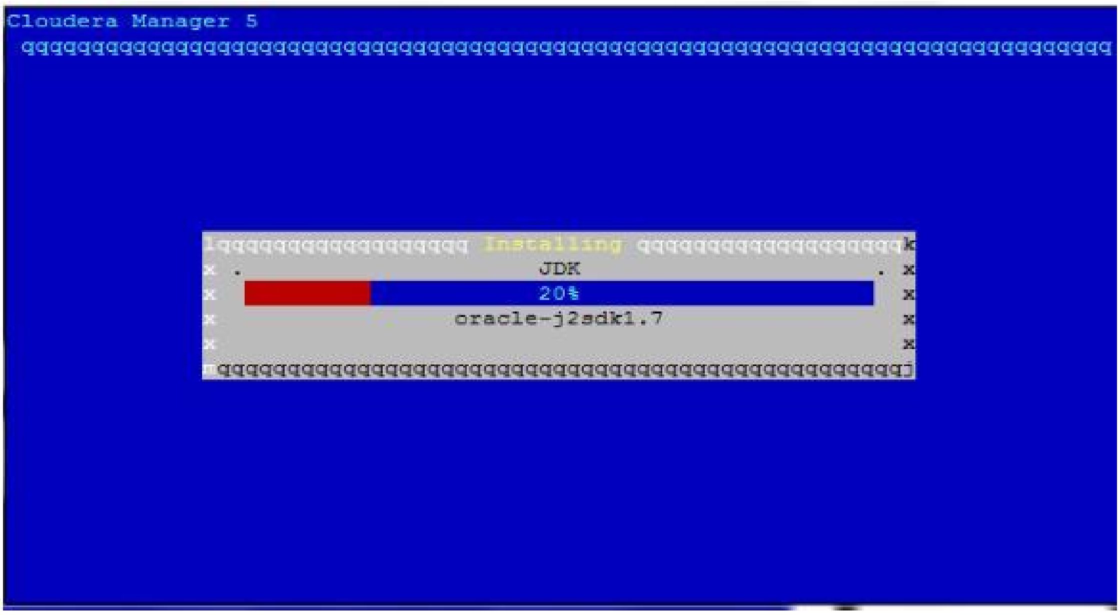
- Accepting License



```
Cloudera Express License
Accept this license?
```

Cloudera Manager Server Installation Step 4

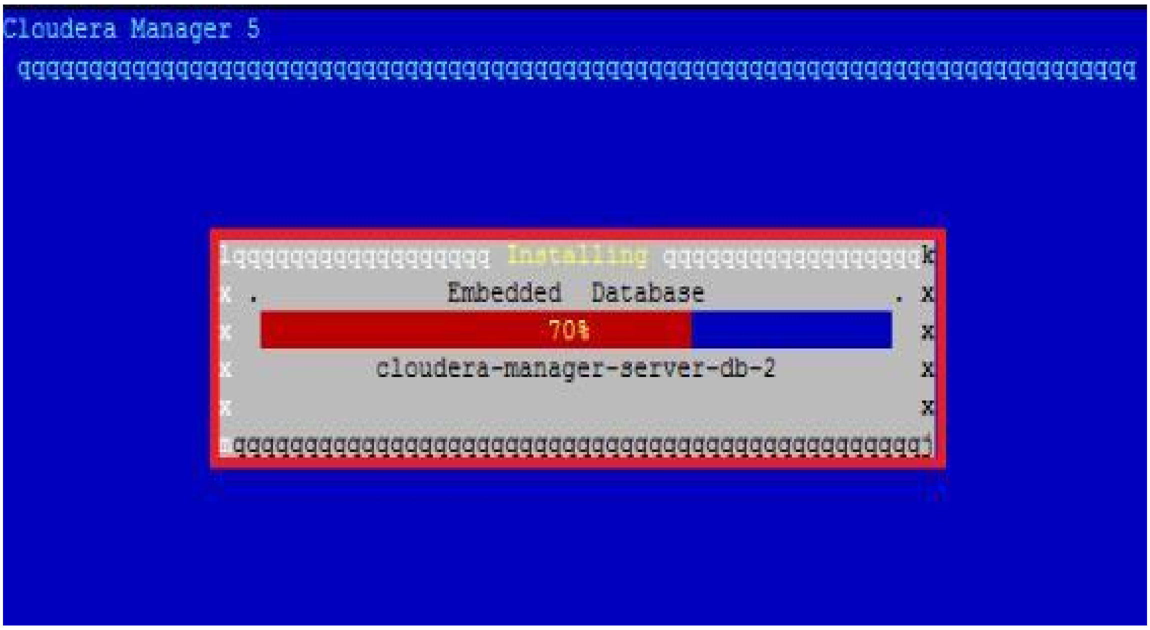
- Automatically installing JDK by the installation process



```
Cloudera Manager 5
Installing
JDK
20%
oracle-j2sdk1.7
```

Cloudera Manager Server Installation Step 5

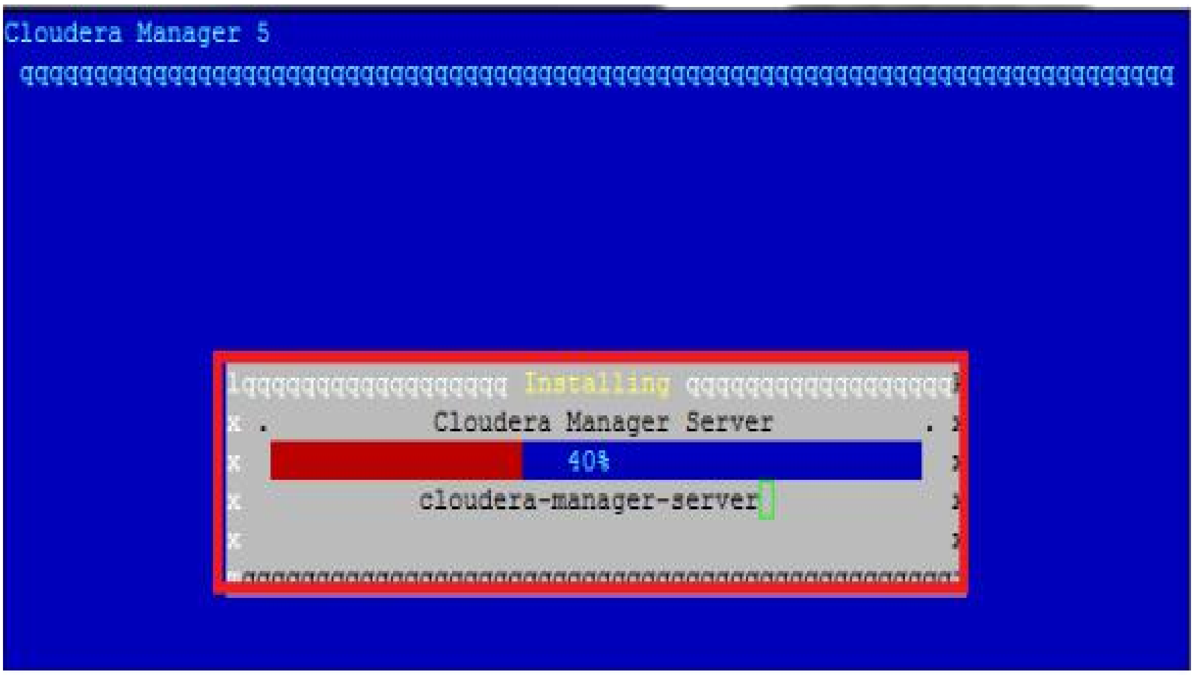
- Automatically installing Embedded Database by the installation process



```
Cloudera Manager 5
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq

lqqqqqqqqqqqqqqqqqqq Installing qqqqqqqqqqqqqqqqqqk
x .            Embedded  Database              . x
x                     70%                        x
x          cloudera-manager-server-db-2          x
x                                                x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Cloudera Manager Server Installation Step 6

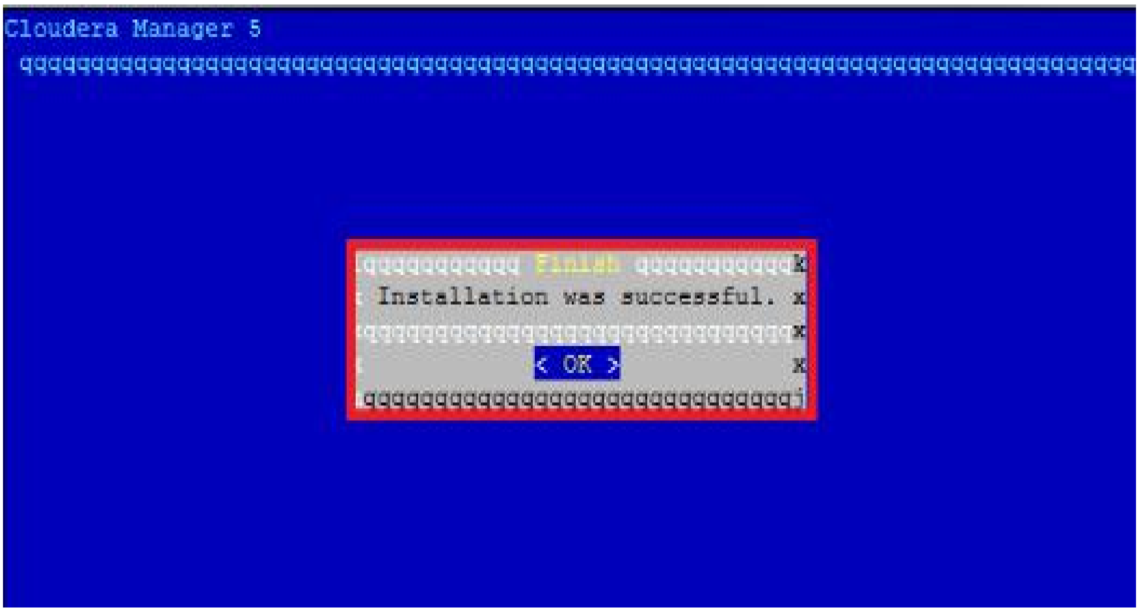
- Installing Cloudera Manager Server



```
Cloudera Manager 5
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq

lqqqqqqqqqqqqqqqqqqq Installing qqqqqqqqqqqqqqqqqqk
x .          Cloudera Manager Server           . x
x                     40%                        x
x            cloudera-manager-server             x
x                                                x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Cloudera Manager Server Installation Step 7

Finally, the installation is successfully completed with the message on screen like this:

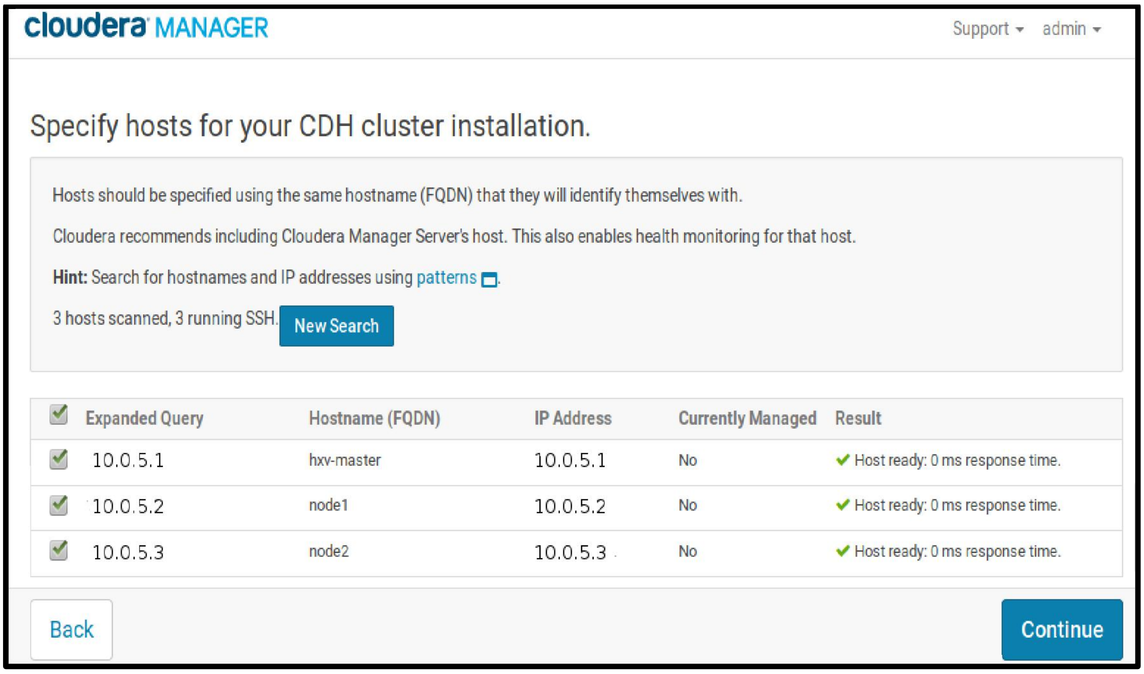


```
Cloudera Manager 5

Finish
Installation was successful.
< OK >
```

Cloudera Manager Server Installation Completion

Adding Hosts to Cloudera Manager

The hosts that are going to configure as computing nodes for the cluster are specified by using Cloudera Manager. The IP addresses for each host VM are already assigned to construct the Multi Node Cloudera Cluster.



Specify hosts for your CDH cluster installation.

Hosts should be specified using the same hostname (FQDN) that they will identify themselves with.

Cloudera recommends including Cloudera Manager Server's host. This also enables health monitoring for that host.

Hint: Search for hostnames and IP addresses using patterns.

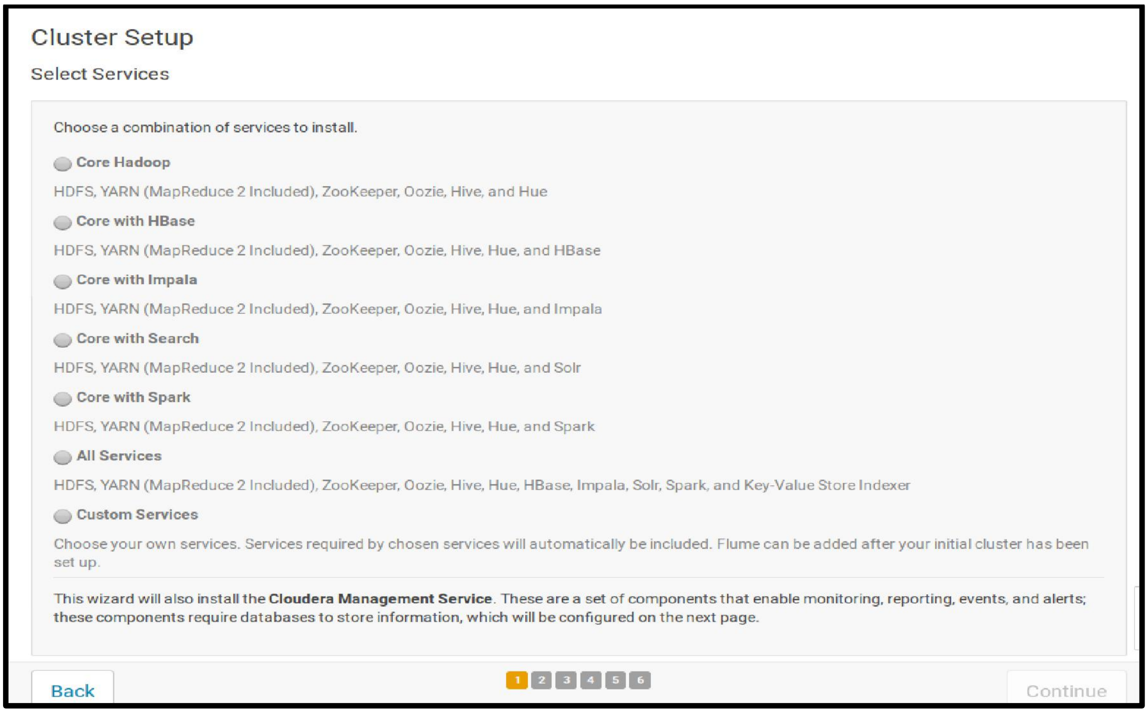
3 hosts scanned, 3 running SSH. New Search

☑	Expanded Query	Hostname (FQDN)	IP Address	Currently Managed	Result
☑	10.0.5.1	hxv-master	10.0.5.1	No	✓ Host ready: 0 ms response time.
☑	10.0.5.2	node1	10.0.5.2	No	✓ Host ready: 0 ms response time.
☑	10.0.5.3	node2	10.0.5.3	No	✓ Host ready: 0 ms response time.

Example for adding hosts to Cloudera Manager

Cluster Setup

It is necessary to choose the option that best suits the requirements. Some extra services can be selected starting from Core Hadoop.



Cluster Setup

Select Services

Choose a combination of services to install.

○ **Core Hadoop**

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, and Hue

○ **Core with HBase**

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and HBase

○ **Core with Impala**

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Impala

○ **Core with Search**

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Solr

○ **Core with Spark**

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Spark

○ **All Services**

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer

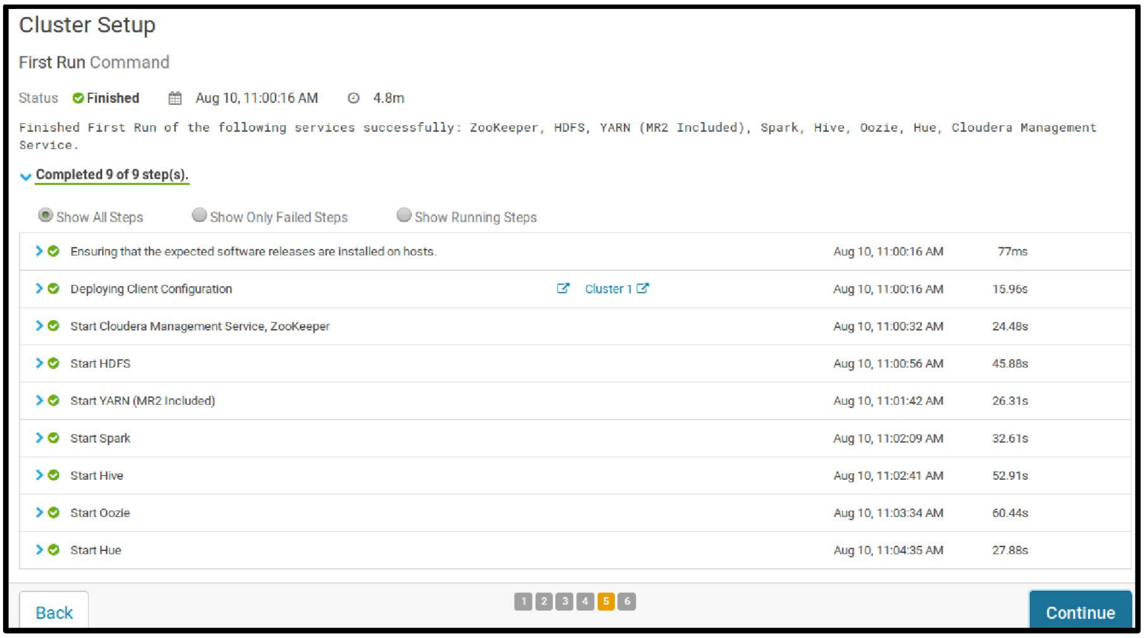
○ **Custom Services**

Choose your own services. Services required by chosen services will automatically be included. Flume can be added after your initial cluster has been set up.

This wizard will also install the **Cloudera Management Service**. These are a set of components that enable monitoring, reporting, events, and alerts; these components require databases to store information, which will be configured on the next page.

Selecting options for cluster setup

After doing many steps such as assigning roles, setting up desired database, and assigning customized block size for HDFS, the cluster is successfully set up to use.



Cluster Setup

First Run Command

Status ✓ **Finished** Aug 10, 11:00:16 AM 4.8m

```
Finished First Run of the following services successfully: ZooKeeper, HDFS, YARN (MR2 Included), Spark, Hive, Oozie, Hue, Cloudera Management
Service.
```

Completed 9 of 9 step(s).

● Show All Steps ○ Show Only Failed Steps ○ Show Running Steps

✓ Ensuring that the expected software releases are installed on hosts.		Aug 10, 11:00:16 AM	77ms
✓ Deploying Client Configuration	Cluster 1	Aug 10, 11:00:16 AM	15.96s
✓ Start Cloudera Management Service, ZooKeeper		Aug 10, 11:00:32 AM	24.48s
✓ Start HDFS		Aug 10, 11:00:56 AM	45.88s
✓ Start YARN (MR2 Included)		Aug 10, 11:01:42 AM	26.31s
✓ Start Spark		Aug 10, 11:02:09 AM	32.61s
✓ Start Hive		Aug 10, 11:02:41 AM	52.91s
✓ Start Oozie		Aug 10, 11:03:34 AM	60.44s
✓ Start Hue		Aug 10, 11:04:35 AM	27.88s

Completion stage of cluster setup