1. Abstract:

The aim of our work is to study different types of data mining algorithms by applying them on a proposed dataset to compare and analyze their performances. We've shown different data mining techniques and designed a basic data warehouse to simplify the dataset. Then we've applied the algorithms on the warehouse data and discussed different application issues. We've concluded our work by visualizing the comparative performance of the algorithms and made a discussion on further works in this field.

2. Introduction:

From the very beginning of human evolution, people started to observe and learn the environment. Human brain saved all the observations into its memory system. With the passage of time they adapted themselves to the environment based on their previous observations. Each distinct observation can be analogized as data. From then to now the increasing exponentially of data generation has been continued. After the industrial revolution, all the industrial operations started to generate data at high scales. These huge amounts of unstructured data became valuable to the authorities. From then data management become a part of study. With the help of increased computational ability, different complex data management and maintenance techniques were invented and became popular to the computer science community. During 90s internet became popular and internet based applications & users started to generate huge data than before. The importance of modern data management and analysis have become more essential than before. Modern business processes make billions of transactions per seconds and create enormous amount of data which are important to understand customer mode, business growth, sustainability etc. To deal with such data and perform efficient analysis, a new branch of computation is created which is called data mining.

2.1 Introduction to our work:

We have aimed to study on different data mining algorithms and apply them to understand their performance. For this we've selected a dataset from Tableau Community[13]. The dataset is about global superstore transactions over the period of 2011 to 2015. The dataset contains 26 attributes and more than fifty-one thousands of record. We have planned to predict different measures based on the interrelated data and perform the actions via different algorithms. For this we've selected **Python** as programming language and **Scikit Learn,** a python based machine learning library and **Pandas**, data manipulation library and different editor and IDEs.

2.2 Problem Definition:

The first and foremost step of our work is the problem definition. There are some interesting quotes regarding this step which will give us a clear view why this step is important.

***"There are no right answers to wrong questions."* - Ursula K. Le Guin.**

So, at first, we need to state what we really want to do. But that is not enough. More precisely, we must ask the right question with proper details. Then we can find a solution.

In our work, initially we've to analyze the dataset and create a structured format of it. To perform this, we've to design a data warehouse to store organized data. Then we can apply mining algorithm on the data warehouse. But there are different variance of each algorithm and performance of a particular algorithms varies in accordance to its application. So we will test the performance of the algorithms by finding the accuracy rate on different applications and then show a comparative performance chart.

2.3 Process of solution:

We've segmented our solutions into different processes which are given below:

i. Data Collection
ii. Data pre-processing
iii. Algorithm Selection
iv. Important Feature selection for each algorithm
v. Model Creation
vi. Performance Testing
vii. Comparison and visualization of algorithms.

Data Collection:

First and foremost step of our solution is to collect the data to perform all the operation. For this, we've collected a dataset. An overview of data collection is given on the introduction section of this report.

Data Pre-processing:

When the data collection is completed the second step is to process the dataset to perform further operations. We've designed a basic data warehouse to process the dataset where the dataset is normalized and segmented into different dimension tables and a fact table. Details description of this step will be discussed on the data warehousing section of this report.

Algorithm Selection:

Most important process of our work is to select appropriate algorithm to perform operations on the dataset. We've selected different algorithms for different operations.

Important Feature selection:

For a specific algorithm we've to select relevant features to perform the operation. There are many approaches for selecting best features. Some of the techniques are discussed later in this report.

Model Creation:

When the features/ parameters are ready, the next step is to make the mathematical model for the algorithm. In this step, the mathematical expression of the algorithms are developed.

Performance Testing:

After the mathematical model is created, we've test the model whether it works or not and provides desired solution.

Comparison and visualization of algorithms:

Finally, we've to compare the results of the algorithms for a specific operation and determine which algorithm works better for that specific operation and visualize the comparative diagram.

2.4 Our goal:

The goal of our work is simple. We will study on few data mining algorithms and understand their scopes and limitations. For this purpose, we've designed data warehouse and performed other operations which will be discussed later in this report.

2.5 Outline of the report:

This section gives an overview of the contents of our report. In chapter One, Abstract of our work has been given.

In Section two, we've provided a brief introduction to our work, goals and problem definition.

In Section three, we put the literature review of our work where basic concepts and historical background of data mining and warehousing were discussed.

In Section four, an overview of our targeted algorithm were enlisted. We've also put the necessary mathematical expressions of the algorithms.

Section five is about our design and implementation of a data warehouse. A snowflake schema and data dictionary of our data warehouse are also described there.

In Section six, we started with the implementation methodologies of our algorithms and then we described our total implementation techniques.

Section seven is about the comparative result analysis of our clustering and regression algorithms.

We've discussed our future plan and scope of further work in Section eight.

Finally, we've concluded our report by a formal conclusion. References were given at the end of this report.

3. Literature Review:

To understand and gain knowledge on any topic, literature is the best reliable source to start with. It gives an in-depth overview on any topic. There are many researchers who have worked on data mining before and their works are recognized globally. To start and continue our work we've to study few books and references of such researchers.

3.1 Historical background of Data Warehouse & Mining:

According to ***Data Mining: Concepts and Techniques, 3rd ed by Jiawei Han, Micheline Kamber and Jian Pei***[1] Since 1960, systematic database and information technology has been evolved from the primitive file processing system to sophisticated database system. In 1970 the database system progressed from hierarchical and network database system to relational database. From then users gained convenient flexible data access through query languages and user interface. After the establishment of database management systems, database technology moved toward the development of advanced database systems, data warehousing, and data mining for advanced data analysis and web-based databases during mid-1980. Advance data analysis started from 1980. Data warehousing techniques were introduced and from 1990, when internet started, huge volumes of data have been accumulated beyond databases and data warehouses and the trend of data mining started to popular to the community.

3.2 Data Mining:

According to '***Data Mining with Predictive Analytics for Financial Applications' by S.B. Soumya1, N. Deepika***[2] *'Data mining, also popularly referred to as knowledge discovery from data (KDD), is the extraction of patterns representing knowledge implicitly stored or captured in large databases, data warehouses, the Web, other massive information repositories or data streams.'*

From other Two references we can define data mining as: '**Data mining** is the computing process of discovering patterns in large data set involving methods at the intersection of machine learning, statistics and database system [3]. It is an essential process where intelligent methods are applied to extract data patterns. [1][3]

It exists, however, in many variations on this theme, such as the Cross Industry Standard Process for Data Mining (CRISP-DM) which defines six phases[2]:

(1) Business Understanding
(2) Data Understanding
(3) Data Preparation
(4) Modeling
(5) Evaluation
(6) Deployment

1. **Business Understanding:**

   Understand the project objectives and requirements from a business perspective, and then convert this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

2. **Data Understanding:**

   This is one of the important steps in data mining. Start by collecting data, then get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses about hidden information. Understanding these things will give us a clear view of the nature of data.

3. **Data Preparation:**

   Includes all activities required to construct the final data set. We have to design our data set in such a way that will be fed into the modeling tool from the initial raw data. Tasks include table, case, and attribute selection as well as transformation and data cleaning, deal with missing values, noisy data and correlated columns.

4. **Modeling:**

   Modeling is an important part of our research. Select and apply a variety of modeling techniques, and calibrate tool parameters to optimal values are the main activities of this phase. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase is often needed.

5. **Evaluation:**

   Thoroughly evaluate the model, and review the steps executed to construct the model, to be certain it properly achieves the business objectives. Determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results is reached.

6. **Deployment:**

   Organize and present the results of data mining. Deployment can be as simple as generating a report or as complex as implementing a repeatable data mining process.

3.3 Data Warehouse for Data Mining:

A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources [1]. According to William H. Inmon, a leading architect in the construction of Data Warehouse systems, “A data warehouse is a subject-oriented, integrated, time-variant and nonvolatile collection of data in support of management’s decision-making process”. In this definition we get four keywords: ***subject-oriented, integrated, time-variant*****, and *****Nonvolatile*** which distinguish the Data warehouse from other Data repository systems. Subject-oriented

means Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a warehouse that concentrates on sales. Using this warehouse, you can answer questions like "Who was our best customer for this item last year?" This ability to define a data warehouse by subject matter, sales in this case, makes the data warehouse subject oriented. By the word Integrated means DW integrates current and historical data from different sources. Time-variant specifies in order to discover trends in business, analysts need large amounts of data. This is very much in contrast to online transaction processing (OLTP) systems, where performance requirements demand that historical data be moved to an archive. A data warehouse's focus on change over time is what is meant by the term time variant. Non-volatile collection of data Nonvolatile means that, once entered into the warehouse, data should not change. This is logical because the purpose of a warehouse is to enable you to analyze what has occurred.

Data Warehouse Application on Data mining[1]:

Data warehousing is particularly important for data mining for the following reasons:

1. **High quality of data in data warehouses:**
   Most data mining tools need to work on integrated, consistent, and cleaned data, which requires costly data cleaning, data integration, and data transformation as preprocessing steps. A data warehouse constructed by such preprocessing serves as a valuable source of high-quality data for OLAP as well as for data mining.

2. **Exploration of multidimensional data:**
   Effective data mining needs exploratory data analysis. A user will often want to traverse through a database, select portions of relevant data, analyze them at different granularities, and present knowledge results in different forms.

4. Analysis of the Selected Algorithms:

4.1 Reason behind Selecting Particular Algorithms:

We have selected four Data Mining algorithms to study. They are:

1. Linear Regression,
2. Polynomial Regression,
3. Logistic Regression &
4. K-Nearest Neighbor.

Here linear & polynomial regression are regression algorithm and logistic regression, K-Nearest Neighbor are classification algorithm. We are working on transactional data. Most of the attributes are numeric. For numeric data, these four algorithms gives better performance. Beside these are the fundamental Data mining algorithms. Details of these algorithms and implementation are discussed in the following section.

4.2 Overview of selected algorithms:

4.2.1 Linear regression:

Regression problems are supervised learning problems in which the response is continuous**. Linear regression** is a technique that is useful for regression problems. Simple linear regression is an approach for predicting a **quantitative response** using a **single feature** (or "predictor" or "input variable"). It takes the following form:

$$\boldsymbol{y = \beta_0 + x\beta_1}$$

Here y is the response, x is the feature, $\boldsymbol{\beta_0}$ is the intercept, $\boldsymbol{\beta_1}$ is the coefficient for x. Together, $\boldsymbol{\beta_0}$ and $\boldsymbol{\beta_1}$ are called the **model coefficients.**

Generally speaking, coefficients are estimated using the **least squares criterion**, which means we are find the line (mathematically) which minimizes the **sum of squared residuals** (or "sum of squared errors"):

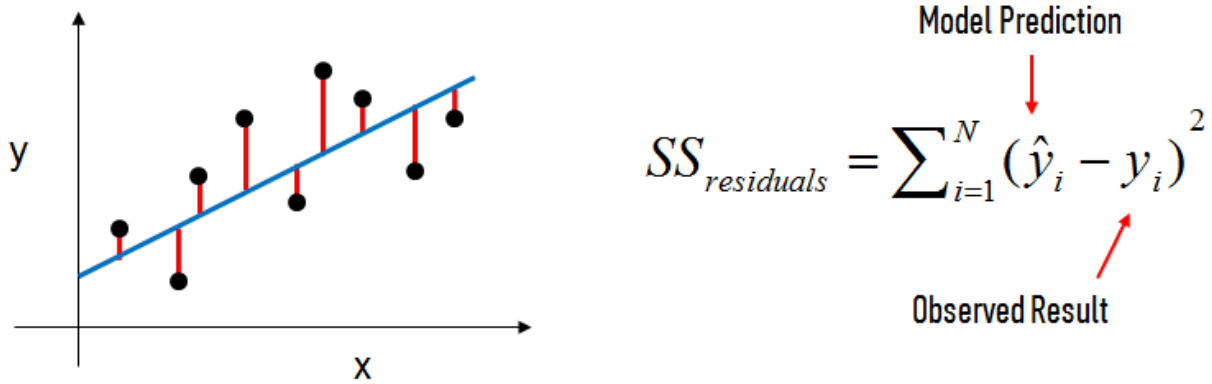

Figure 1: Linear regression

In this figure, the black dots are the observed values of x and y. The blue line is our least squares line. The red lines are the residuals, which are the distances between the observed values and the least squares line.

The model coefficients relate to the least squares line $\boldsymbol{\beta_0}$ is the intercept (the value of y when x=0) and $\boldsymbol{\beta_1}$ is the slope (the change in y divided by change in x).Here is a graphical representation of those calculations:

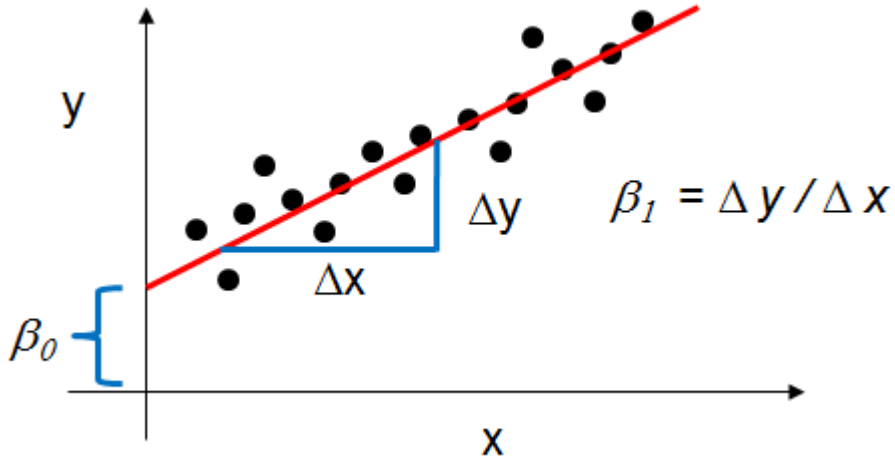

Figure 2: Co-efficient calculation of linear regression

Multiple Linear Regression:

Simple linear regression can easily be extended to include multiple features. This is called **multiple linear regression**:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Each $\boldsymbol{x}$ represents a different feature, and each feature has its own coefficient. From our dataset if we want to predict the profit a product and if we use Sales, Quantity and Discount as a feature then the equation will be looked like this:

$$Profit = \beta_0 + \beta_1 \times Sales + \beta_2 \times Quantity + \beta_3 \times Discount$$

Model Evaluation Metrics for Regression:

The metrics that are popular for regression problems are Mean Absolute Error, Mean Squared Error and Root Mean Squared Error.

Mean Absolute Error (MAE) is the mean of the absolute value of the errors:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - py_i|$$

Mean Squared Error (MSE) is the mean of the squared errors:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - py_i)^2$$

Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - py_i)^2}$$

Here, y_i is the original value of target variable and py_i is the predicted value of target variable and n is the number of samples. MSE is more popular than MAE because MSE "punishes" larger errors. But, RMSE is even more popular than MSE because RMSE is interpretable in the "y" units. We will use the RMSE metrics in our implementation.

4.2.2 Polynomial Regression:

Polynomial regression is a form of regression analysis in which the relationship between the independent variable *x* and the dependent variable *y* is modeled as an n-th degree polynomial in *x*. Polynomial regression fits a nonlinear relationship between the value of *x* and the corresponding conditional mean of y .

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_m x_i^m (i = 1,2,3 \ldots., n)$$

The goal of regression analysis is to model the expected value of a dependent variable *y* in terms of the value of an independent variable *x*. A second order (m=2) polynomial forms a quadratic expression (parabolic curve), a third order (m=3) polynomial forms a cubic expression and a fourth order (m=4) polynomial forms a quartic expression. In many settings, such a linear relationship may not hold. That's where the polynomial regression become popular. Some important point regarding polynomial regression is:

- The fitted model is more reliable when it is built on large numbers of observations.
- Do not extrapolate beyond the limits of observed values.

- Choose values for the predictor (x) that are not too large as they will cause overflow with higher degree polynomials; scale x down if necessary.

4.2.3 Logistic Regression:

It is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a *logit transformation* of the probability of presence of the characteristic of interest.

Let assign 1 if an email is spam and 0 if it's not. So, our prediction choices can be written as:

$P(Y=1|x; \vartheta)$ *and* $P(Y=0|x; \vartheta)$

Then we'll chose a threshold value for our prediction function $h_\theta(x)$ is 0.5

If $h_\vartheta(x) \geq 0.5$ then Y = 1 and if $h_\vartheta(x) < 0.5$ then Y = 0;

Now, if we have the training set $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots\ldots, (x^{(m)}, y^{(m)}),\}$

For m examples $x = \begin{pmatrix} x0 \\ x1 \\ x2 \\ \ldots. \\ xn \end{pmatrix}$ where $x_0 = 1$ and Y € {0,1}

we can write our hypothesis of the logistic function as $h_\vartheta(x) = 1 / 1 + e^{-\theta x}$. This function is called **Sigmoid function** or logistic function.

Now we've to choose the parameter ϑ to fit the function.

Cost Function: we can denote the cost function as ***j,***

So, $j(\vartheta) = \frac{1}{m} \sum_{i=1}^{m} cost(h_\vartheta(x^{(i)}), y^{(i)})$

$= \frac{1}{m} [\sum_{i=1}^{m} y^{(i)} \log h_\vartheta(x^{(i)}) + (1- y^{(i)}) \log(1- h_\vartheta(x^{(i)}))]$, m is the number of total examples.

To fit the parameter ϑ we've to take the minimum of $j(\vartheta)$.

Gradient Descent: from the previous section, we know the cost function $j(\vartheta)$*:*

$J(\vartheta) = \frac{1}{m} [\sum_{i=1}^{m} y^{(i)} \log h_\vartheta(x^{(i)}) + (1- y^{(i)}) \log(1- h_\vartheta(x^{(i)}))]$

Now we will minimize the cost function by repeating the following formula:

$\vartheta_j = \vartheta_j - \alpha \frac{\partial}{\partial \theta} j(\vartheta)$

$\vartheta_j = \vartheta_j - \alpha \sum_{-=1}^{m} (h_\vartheta (x^{(i)}) - y^{(i)}) x_j^{(i)}$

where α is the learning rate of the equation.

Finally if we update the value of minimized ϑ_j to the hypothesis we will get the logistic regression equation.

4.2.4 K-Nearest Neighbor:

K-Nearest Neighbor (in short KNN) is a supervised learning algorithm. This means a labelled dataset is given consisting of training observations (x,y) and we would like to capture the relationship between x and y. More formally, our goal is to learn a function h:X→Y so that given an unseen observation x, h(x) can confidently predict the corresponding output y. KNN is a non-parametric and instance based algorithm.

Non-Parametric means it makes no explicit assumptions about the functional form of h, avoiding the dangers of miss-modeling the underlying distribution of the data. For example, suppose our data is highly non-Gaussian but the learning model we choose assumes a Gaussian form. In that case, our algorithm would make extremely poor predictions.

Instance-Based learning means that our algorithm doesn't explicitly learn a model. Instead, it chooses to memorize the training instances which are subsequently used as "knowledge" for the prediction phase.

The K-nearest-neighbor (KNN) algorithm measures the distance between a query scenario and a set of scenarios in the data set. We can compute the distance between two scenarios using some distance function d(x, y), where x, y are scenarios composed of K features, such that :

$X = \{x_1, ..., x_N\}$, $Y = \{Y_1, ..., Y_N\}$.

Two distance measuring techniques are given bellow:

Euclidian Distance: $d_E (X,Y) = \sum_{i=1}^{K} \sqrt{xi2 - yi2}$

Manhattan Distance: $d_A (X,Y) = \sum_{i=1}^{K} |X1 - Y1|$

Now that we have established a measure in which to determine the distance between two scenarios, we can simply pass through the data set, one scenario at a time, and compare it to the query scenario.

We can represent our data set as a matrix M = N x P , containing P data points D^1,……,D^P , where each datapoint contains N features D^i = { D^i_1,……, $_{Dn}{}^p$ }. A vector ***v*** with length P of output values $\mathbf{v = \{v^0,\ldots\ldots v^p\}}$ accompanies this matrix, listing the output value $\mathbf{v^i}$ for each data point D^i .

It should be noted that the vector ***v*** can also be a column matrix; if multiple output values are desired, the width of the matrix may be expanded.

5. Designing of Data Warehouse for Mining in Sales Data:

5.1 Initial transactional Data:

The dataset we decided to work on is an Open source Dataset and collected from Tableau Community [13]. Our Dataset have 24 Columns, 51290 transactional tuples, across 147 countries. There are 10768 products in our dataset which divided into three categories and 17 Sub categories.

5.2 Schema of Data Warehouse:

The most popular data model for a data warehouse is a multidimensional model, which can exist in the form of a star schema, a snowflake schema, or a fact constellation schema.

We have designed our Data Warehouse using snow flake schema. The model, we have designed has fourteen dimension tables and one fact table. The fact table contains 15 attributes in which one is used as a primary key, 8 attributes is used as a foreign key for the dimension tables and rest of the attributes are different measures of a single tuple.

There are some dimension tables which contain some new attributes out of our data set. The Order Date and Ship Date both dimension tables have details information about time. In our dataset we only have a date. For mining purpose, we now break this date into day, month, year and quarter. Similarly, for any particular product we did not have the actual price and selling price. We also find out that from our existing data. These new attributes will give us a better understanding of our data. The design of data warehouse is shown on fig 5.1:

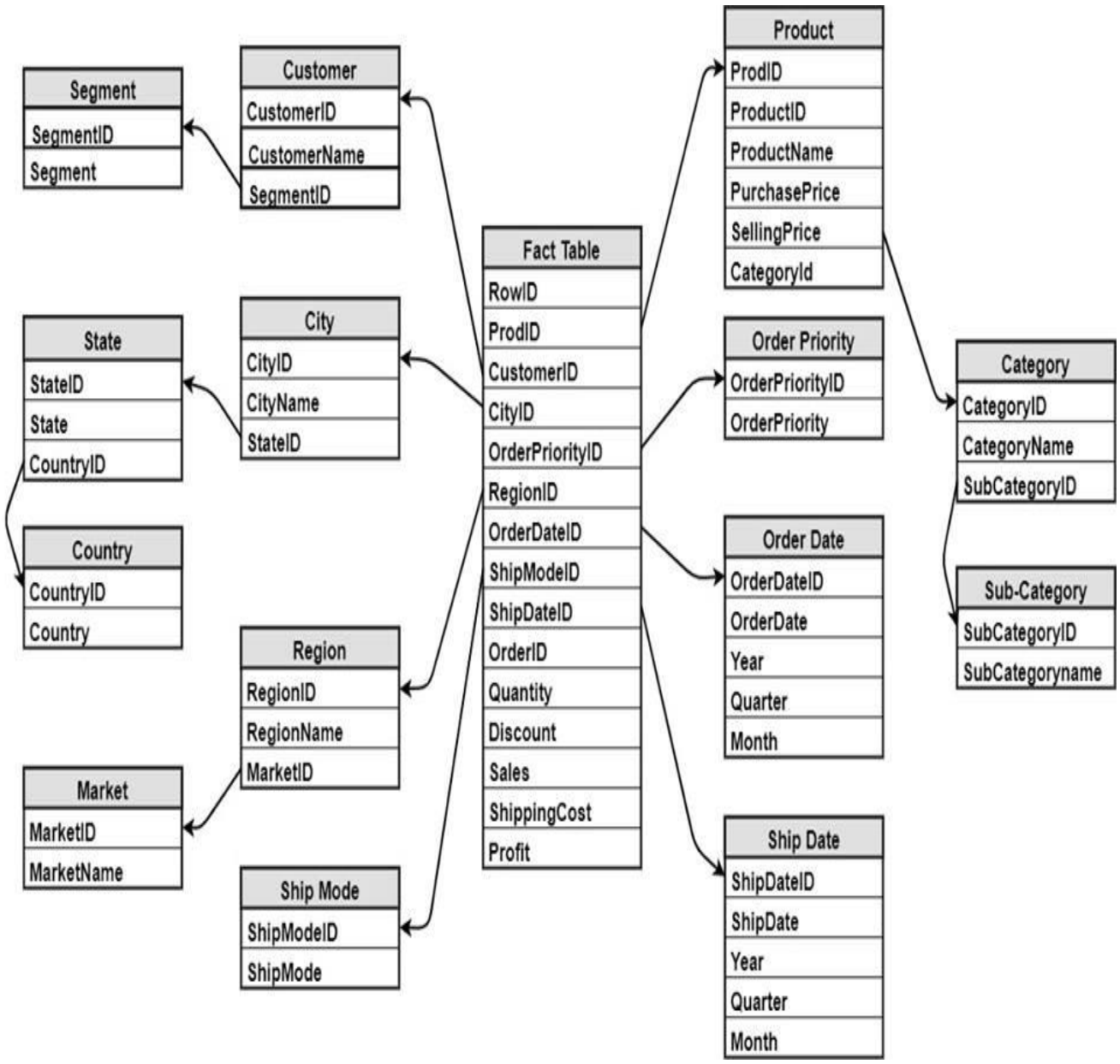

Figure: 3: Snow-flake schema of our Data Warehouse

5.3. Data Dictionary of Data Warehouse:

A data dictionary is a list of definitions used in the system. Each definition is about usually about 10 to 50 words long. And there are about 50 to 200 definitions. What being defined is mostly technical terms, such as the meaning of each field in the database. The data dictionary of our data warehouse is given below:

FACT-TABLE:

Column Name	DATA TYPE	Constraints	Description
RowID	VARCHAR(50 BYTE)	Primary Key	Fact Table Row ID
ProdID	VARCHAR(50 BYTE)	Foreign Key	ProdID from Product dimension table
CustomerID	VARCHAR(50 BYTE)	Foreign Key	CustomerID from Customer dimension table
CityID	VARCHAR(50 BYTE)	Foreign Key	City ID from City dimension table
OrderPriorityID	VARCHAR(50 BYTE)	Foreign Key	OrderPriorityID from OrderPriority dimension table
RegionID	VARCHAR(50 BYTE)	Foreign Key	RegionID from region dimension table
OrderDateID	VARCHAR(50 BYTE)	Foreign Key	OrderDateID from OrderDate dimension table
ShipModeID	VARCHAR(50 BYTE)	Foreign Key	ShipModeID from ShipMode dimension table
ShipDateID	VARCHAR(50 BYTE)	Foreign Key	ShipDateID from Shipdate dimension table
OrderID	VARCHAR(50 BYTE)	-	OrderID of products
Quantity	NUMBER(7,0)	-	Quantity of products
Sales	NUMBER(7,4)	-	Transaction amount
Discount	NUMBER(7,4)	-	Discount on products
ShippingCost	NUMBER(7,4)	-	Cost of shipping
Profit	NUMBER(7,4)	-	Benefits from selling

Dimension Table: Customer

Column Name	DATA TYPE	Constraints	Description
CustomerID	VARCHAR(50 BYTE)	Primary Key	Customer ID
CustomerName	VARCHAR(50 BYTE)	-	Name of the customer
SegmentID	VARCHAR(50 BYTE)	Foreign Key	Segment ID

Dimension Table: City

Column Name	DATA TYPE	Constraints	Description
CityID	VARCHAR(50 BYTE)	Primary Key	City ID
CityName	VARCHAR(50 BYTE)	-	City Name
StateId	VARCHAR(50 BYTE)	Foreign Key	State ID

Dimension Table: Region

Column Name	DATA TYPE	Constraints	Description
RegionID	VARCHAR(50 BYTE)	Primary Key	Region ID
RegionName	VARCHAR(50 BYTE)	-	Region Name
MarketID	VARCHAR(50 BYTE)	Foreign Key	Market ID

Dimension Table: OrderPriority

Column Name	DATA TYPE	Constraints	Description
OrderPriorityID	VARCHAR(50 BYTE)	Primary Key	Unique Order Priority ID
OrderPriority	VARCHAR(50 BYTE)	-	Priority of ordered product

Dimension Table: Product

Column Name	DATA TYPE	Constraints	Description
ProdID	VARCHAR(50 BYTE)	Primary Key	Unique key defining unique product id and name
ProductID	VARCHAR(50 BYTE)	-	Product ID
ProductName	VARCHAR(50 BYTE)	-	Product Name
PurchasePriceUnit	NUMBER(7,4)	-	Price of purchasing a product
SellingPricePerUnit	NUMBER(7,4)	-	Price of selling a product
CategoryID	VARCHAR(50 BYTE)	Foreign Key	Category ID of a product

Dimension Table: OrderDate

Column Name	DATA TYPE	Constraints	Description
OrderDateID	VARCHAR(50 BYTE)	Primary Key	Unique Order Date ID
OrderDate	DATE	-	Date of order
Year	NUMBER(7,0)	-	Year portion of order date
Quarter	VARCHAR(50 BYTE)	-	Quarter of the year
Month	VARCHAR(50 BYTE)	-	Month of order date

Dimension Table: ShipMode

Column Name	DATA TYPE	Constraints	Description
ShipModeID	VARCHAR(50 BYTE)	Primary Key	Unique Ship Mode ID
ShipMode	VARCHAR(50 BYTE)	-	Mode of shipping

Dimension Table: ShipDate

Column Name	DATA TYPE	Constraints	Description
ShipDateId	VARCHAR(50 BYTE)	Primary Key	Unique Ship Date ID
ShipDate	DATE	-	Date of shipping
Year	NUMBER(7,0)	-	Year portion of shipping date
Quarter	VARCHAR(50 BYTE)	-	Quarter of shipping date
Month	VARCHAR(50 BYTE)	-	Month of shipping date

Dimension Table: Segment

Column Name	DATA TYPE	Constraints	Description
SegmentID	VARCHAR(50 BYTE)	Primary Key	Unique Segment ID
Segment	VARCHAR(50 BYTE)	-	Name of the segment

Dimension Table: State

Column Name	DATA TYPE	Constraints	Description
StatetID	VARCHAR(50 BYTE)	Primary Key	Unique State ID
State	VARCHAR(50 BYTE)	-	Name of the state
CountryID	VARCHAR(50 BYTE)	Foreign Key	Country ID from country dimension table

Dimension Table: Market

Column Name	DATA TYPE	Constraints	Description
MarketID	VARCHAR(50 BYTE)	Primary Key	Unique Market ID
MarketName	VARCHAR(50 BYTE)	-	Name of the market

Dimension Table: Category

Column Name	DATA TYPE	Constraints	Description
CategoryID	VARCHAR(50 BYTE)	Primary Key	Unique Category ID
CategoryName	VARCHAR(50 BYTE)	-	Name of the category
SubCategoryID	VARCHAR(50 BYTE)	Foreign Key	SubCategory ID from SubCategory Dimension table

Dimension Table: Country

Column Name	DATA TYPE	Constraints	Description
CountrytID	VARCHAR(50 BYTE)	Primary Key	Unique Country ID
Country	VARCHAR(50 BYTE)	-	Name of the country

Dimension Table: SubCategory

Column Name	DATA TYPE	Constraints	Description
SubCategorytID	VARCHAR(50 BYTE)	Primary Key	Unique Sub-Category ID
SubCategory	VARCHAR(50 BYTE)	-	Name of the sub-category

6. Implementation of Data Mining Algorithms:

6.1 Methodology:

6.1.1 Important Feature selection for each algorithm:

Feature selection is useful as a preprocessing step to improve computational efficiency in predictive modeling. Oracle Data Mining implements feature selection for optimization. We use **Pearson Correlation** in our research for feature selection for regression problem and **RFECV** for classification problem.

Pearson Correlation:

Correlation between sets of data is a measure of how well they are related. The most common measure of correlation in stats is the Pearson Correlation. It shows the linear relationship between two sets of data. In simple terms, it answers the question, Can I draw a line graph to represent the data?

We can categorize the type of correlation by considering as one variable increases what happens to the other variable:

- Positive correlation – the other variable has a tendency to also increase;
- Negative correlation – the other variable has a tendency to decrease;
- No correlation – the other variable does not tend to either increase or decrease.

The starting point of any such analysis should thus be the construction and subsequent examination of a *scatterplot*. Examples of negative, no and positive correlation are as follows.

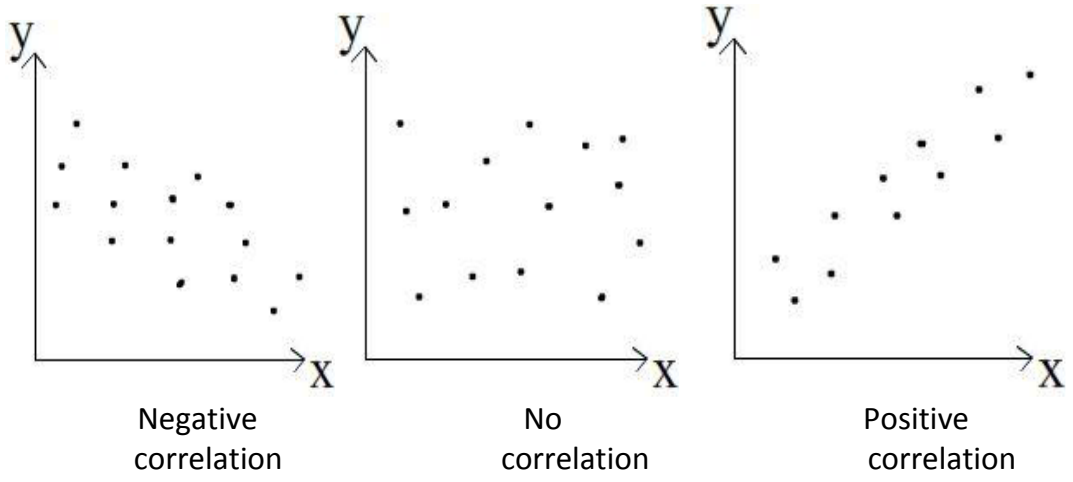

Figure 4 : Correlation

Pearson's correlation coefficient is a statistical measure of the strength of a linear relationship between paired data. In a sample it is denoted by *r* and is by design constrained as follows:

- Positive values denote positive linear correlation;
- Negative values denote negative linear correlation;
- A value of 0 denotes no linear correlation;
- The closer the value is to 1 or –1, the stronger the linear correlation.

The general formula of Pearson's co-efficient is:

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\, n\Sigma x^2 - (\Sigma x)^2 \,][\, n\Sigma y^2 - (\Sigma y)^2 \,]}}$$

Where x is the feature and y is the dependent variable or target variable and n is the number of samples. In Python, we have a built in library to calculate Pearson co-efficient.

Recursive feature elimination with cross-validation:

For classification with small training samples and high dimensionality, feature selection plays an important role in avoiding over fitting problems and improving classification performance. One of the commonly used feature selection methods for small samples problems is recursive feature elimination (RFE) method. RFE method utilizes the generalization capability embedded in support vector machines and is thus suitable for small samples problems. Despite its good performance, RFE tends to discard "weak" features, which may provide a significant improvement of performance when combined with other features. We initially start with all the features. For every step or iteration, the worst x number of features are eliminated using the "step" parameter till "n-features" are left. If you notice, you need to provide the n-features parameter in the constructor.

The RFECV object helps to tune or find this n_features parameter using cross-validation. For every step where "step" number of features are eliminated, it calculates the score on the validation data. The number of features left at the step which gives the maximum score on the validation data, is considered to be "the best n_features" of data. To use RFECV we need to import RFECV library from sklearn.feature_selection . The RFECV function has some parameters.

class sklearn.feature_selection.**RFECV**(*estimator*, *step=1*, *cv=None*, *scoring=None*, *verbose=0*, *n_jobs=1*)

Here, **Estimator:** a supervised learning estimator, **Step** is the number of feature eliminate at each iteration, **CV** Determines the cross-validation splitting strategy and it optional, **Scoring** is A string or a scorer callable object / function with signature(estimator, X, y).Rest of the parameters are not important.

After creating an instance of RFECV we call Fit function with our feature vector and target variable. The RFE model and automatically tune the number of selected features. RFECV has two important attribute named **n_features_ & ranking_** . **n_features_ returns** The number of selected features with cross-validation. **ranking_ is an array that returns t**he feature ranking, such that *ranking_[i]* corresponds to the ranking position of the i-th feature. Ranking with 1 indicates the best feature.

6.1.2 K-fold Cross Validation:

In train-test split, We split the dataset into two pieces, so that the model can be trained and tested on different data. Testing accuracy is a better estimate than training accuracy of out-of-sample performance. But, it provides a high variance estimate since changing which observations happen to be in the testing set can significantly change testing accuracy. What if we created a bunch of train/test splits, calculated the testing accuracy for each, and averaged the results together? That's the essence of cross-validation.

Steps for K-fold cross-validation:

1. Split the dataset into K **equal** partitions (or "folds").
2. Use fold 1 as the **testing set** and the union of the other folds as the **training set.**
3. Calculate **testing accuracy**.
4. Repeat steps 2 and 3 K times, using a **different fold** as the testing set each time.
5. Use the **average testing accuracy** as the estimate of out-of-sample accuracy.

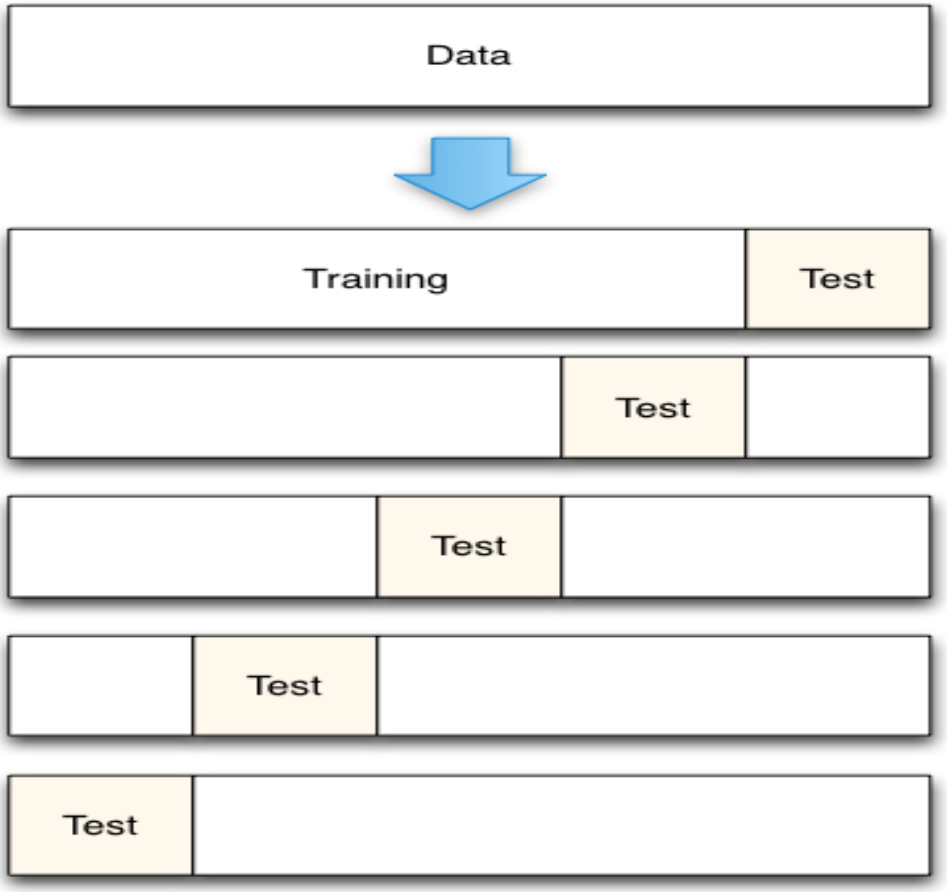

Figure 5: 5-fold cross-validation

Comparing cross-validation to train/test split

Advantages of **cross-validation:**

- More accurate estimate of out-of-sample accuracy
- More "efficient" use of data (every observation is used for both training and testing)

6.2 Implementation of Algorithms:

6.2.1 Linear regression Implementation:

The first problem we want to solve is predicting the profit of a particular product using linear regression algorithm. After analyzing our dataset, we have seen that the highest selling product is “Binney & Smith Sketch Pad, Blue” which product id is “OFF-BIN-10002061”. This product has four variances but the “OFF-BIN-10002061” has been sold 23 times. Our goal is to predict the profit of this product from our dataset.

According to Pearson Correlation we will select our features. We have seen that sales, Discount and quantity has linear relationship with our target variable Profit. We will select Sales, Discount and Quantity as our features and store them in a data frame named X. Our target variable is profit. So we will take this attribute to another data-frame named Y.

We will use cross-validation to evaluate the predictive model. RMSE metrics has been used. Since we have little amount of data for a single product, we will use 3-fold cross validation then calculated the average RMSE value. We will also use train-test split and observe the difference between cross validation and train-test split. We get a RMSE value of 5.96993034214e-14 which is near to zero. In train-test split we got 6.29793031992e-14.

Now we want to predict the profit of any product using linear regression. After analyzing our dataset, we see that we have nearly 10768 products. Now we will try to build a model that can predict any product’s profit .Here we will use full dataset that means all 51290 tuples to build our model.

According to previous problem, we will first select some important features using Pearson correlation. But here we see that there is no strong relationship between any features with the target variable. All Pearson co-efficient value is between -.316 to .485**.** So, it is clear to us that there is no linear relationship exist and it will give us high RMSE value if we want to apply linear regression. Now we pick some features that are not very close to zero and use them as our input features. We will select SellingPriceperunit, PurchasingPriceperUnit, Sales, Quantity & Discount as our feature vector X and Profit as our target variable y.

Here we have plenty of data so we have used 10-fold cross validation. We got the RMSE value of 98.76.

We also have another attribute named shipping cost which indicates the cost required to ship a product. Using linear regression we can also predict shipping cost. The working procedure of linear regression is known to us now. Like before, first we try to build a model which predict the shipping cost of a particular product. Then we will attempt to build

a model that can predict shipping cost for any product. One thing should be mentioned that previously we have used Pearson co-relation to take features. It gives us better understanding of which attribute has linear relation with target variable but it also has limitation. Combination of non-linear features can be used as a good predictor for the model.

"Binney & Smith Sketch Pad, Blue" is the selected product for which we want to predict the shipping cost. So we separate this product data from our dataset and train our model using this data. The input features are Sales, Quantity, OrderpriorityID, ShipmodeID & RegionID. Target variable is Shipping Cost. Here we have little amount of data so we have used 3-fold cross validation. We got the RMSE value of 3.31. We did not take the features accordingly Pearson correlation. If we take features according to Pearson Co-efficient then we get a higher RMSE value. In the first case we took StateID as our input feature which has no linear relationship with the target variable but still gave us lower RMSE. But when we take the feature that have strong linear relation with the target variable we have higher RMSE.

We have got reasonable RMSE while predicting shipping cost of a particular product. Now we try to predict the shipping cost of all products. After observing the values of Pearson Correlation, we see that there is no strong relationship with Shipping cost with any attribute. Only Sales have slightly strong relationship with shipping cost. So, we took only Sales as our input feature. The RMSE value we get after evaluating the model is 29.26.

6.2.2 Polynomial regression Implementation:

Previously, we did not get a good model for predicting the profit of any product using linear regression. In this section, we have tried to predict the profit and shipping cost using polynomial regression. We will select SellingPriceperunit, PurchasingPriceperUnit, Sales, Quantity & Discount as our feature vector X and Profit as our target variable y. We will use cross-validation to evaluate the model. RMSE metrics has been used. Here we have plenty of data so we have used 10-fold cross validation then calculated the average RMSE value. We run polynomial regression using degree 1 to 4 . We get 0.34 RMSE value which is near to zero using polynomial with Degree 3.

Using linear regression, we also get higher RMSE value in predicting shipping cost. Now we will to use polynomial regression and see our model's performance improves or not. "Binney & Smith Sketch Pad, Blue" is the selected product for which we want to predict the shipping cost. So we separate this product's data from our dataset and train our model using these data. The input features are Sales, Quantity, OrderpriorityID, ShipmodeID & RegionID. Target variable is Shipping Cost. Here we have little amount of data so we have used 3-fold cross validation. We got the RMSE value of 3.31 when we use polynomial with degree 1. We got higher RMSE value 23.18, 261.58 for using polynomial regression with degree 2, 3 respectively.

When we take all products data and try to predict shipping cost using polynomial regression using degree 1 to 5 we get RMSE value of 29.26, 26.27, 26.74, 30.32, 89.53 respectively.

6.2.3 Logistic regression Implementation:

In this section, our goal is to detect the order priority of any product using Logistic regression. Since Order Priority has discrete value, we are using classification algorithm. To prepare the dataset, first we need to identify the important features. In case of classification problem, we will use Recursive feature elimination with cross-validation (RFECV). For predicting OrderPriorityID , we have checked the ranking of all attributes using RFECV. Then we have selected the feature which has rank 1. After running RFECV we have got 3 features ranking 1. They are DateDif, Discount & CategoryID. So, we will use these three features as our input feature X. Our target variable y, is Order Priority. We will use 10 fold cross-validation to evaluate the model. Accuracy has been used as a scoring parameter. Despite of selecting the best features we got 62.73% accuracy.

We also have another attribute called Ship mode which also contains discrete values and an important feature in our dataset. To predict the ship mode of any product first we need to identify the important features. For predicting The Shipping mode , we have checked the ranking of all attributes using RFECV. Then we have selected the feature which has rank 1. After running RFECV we get one features as ranking one which is DateDif. So, we take this as a input feature and evaluate our model using 10 fold cross validation. It gives us 78.11% accuracy .

6.2.4 K-Nearest Neighbor Implementation:

In this section, our goal is to predict the order priority and ship mode using K-nearest neighbor. Like logistic regression, to prepare the dataset first we need to identify the important features. For predicting OrderPriorityID , we have selected DateDif, Discount & CategoryID as our input features . We will run KNN for k= 1 to 700 and select the best k to build our model.

Using same techniques we have tried to predict the shipping mode of any product. This time we have selected DateDif as our input feature and run KNN for k=1 to 60. The reason of selecting the range of k differently for two will be discussed in result analysis section.

7. Result Analysis:

In this section we will analysis the results we have got from the implementation of section 6.2.1-6.2.4. First we will see the performance of regression algorithms on our dataset then we will check the performance and comparison of classification algorithms.

7.1 Result Analysis of Regression Algorithms:

In section 6.2.1, we first implemented linear regression for a single particular product only. We get a RMSE value of 5.96993034214e-14 in K-fold Validation. In train-test split we got 6.29793031992e-14. But if we select different portion of our data again in train-test split then we will get different RMSE value. The K-fold validation gives us the best estimation. After fitting the data we get the co-efficient of 'Sales' is 1.00000000e+00,'Quantity' is -3.42900000e+01, and 'Discount' is 1.95399252e-14. The intercept is 1.09690034833e-13. If we put these values in our equation the equation looks like:

$$Profit = 1.097e - 13 + 1.00 \times Sales - (3.429\mathrm{e} + 01) \times Quantity +$$

$$(1.95\mathrm{e} - 15) \times Discount$$

After observing the co-efficient of feature attributes, we can say that Discount's co-efficient is near to zero. That means this attribute has no impact on the final result. If we remove this attribute as our feature and check our score again then we got RMSE value of 5.42633456253e-14 which is slightly decreased from previous model. More precisely, Discount attribute hampers our training. By removing it we get a better model.

We have built a model for predicting the profit for all product in the second part of linear regression. . We got the RMSE value of 98.76. The reason behind getting higher RMSE is there is no strong relationship between feature attributes and target variable. In section 6.2.2 we also use polynomial regression using degree 1 to 4 for predicting the profit. After running the polynomial regression using degree 1 to 4 we see that when we are using degree 1 then it gives us a higher RMSE value and it is equal to linear regression RMSE. Polynomial regression with degree 1 is nothing but simple linear regression. As we have already seen that there is no strong relationship here that's why the RMSE is high.

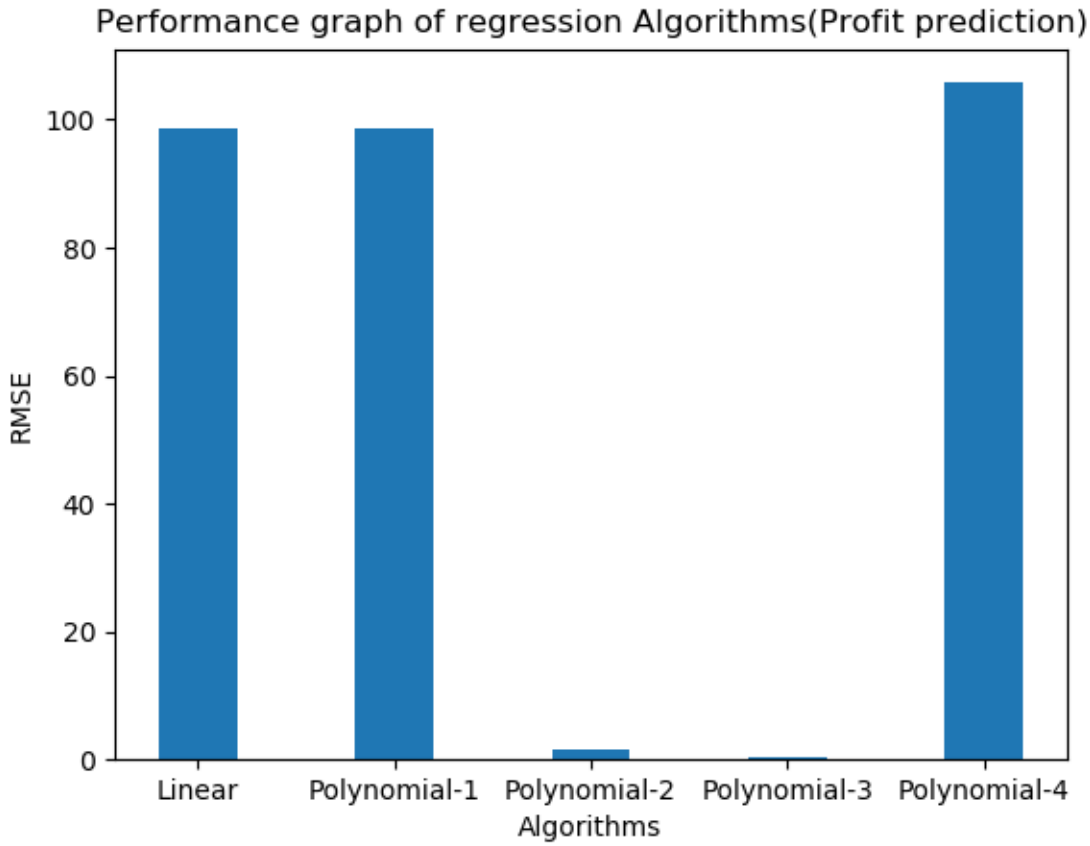

Figure 6: Performance graph for regression algorithms

(Profit prediction for any product)

From the graph we see that, the lowest value we get is 0.34 using polynomial with degree 3. Then the RMSE increases again. We have already known from section 4.3 that if we have plenty of observation and there is no linear relationship in our dataset using polynomial regression will give us a better model. It will try to fit a curve that covers maximum of our observed data. Here we did not have any linear relationship between our target variable and input features. That's why using polynomial regression gives us better model than linear regression. Since the degree is 3 we have a cubic expression.

We have also predicted shipping cost using polynomial regression. We got the RMSE value of 3.31 when we use polynomial with degree 1. We got higher RMSE value 23.18, 261.58 for using polynomial regression with degree 2, 3 respectively. Since we have little amount of data polynomial regression with higher degree gives us higher RMSE. One thing should be mentioned that we did not take the features accordingly Pearson correlation. If we take features according to Pearson Co-efficient then we get a RMSE value of 5.7. In the first case we took StateID as our input feature which has no linear relationship with the target

variable but still gave us lower RMSE. But when we take the feature that slightly have linear relation with the target variable we get higher RMSE. From here we can conclude that Pearson correlation doesn't always give us best features. We can combine non-linear features and can gain better performance.

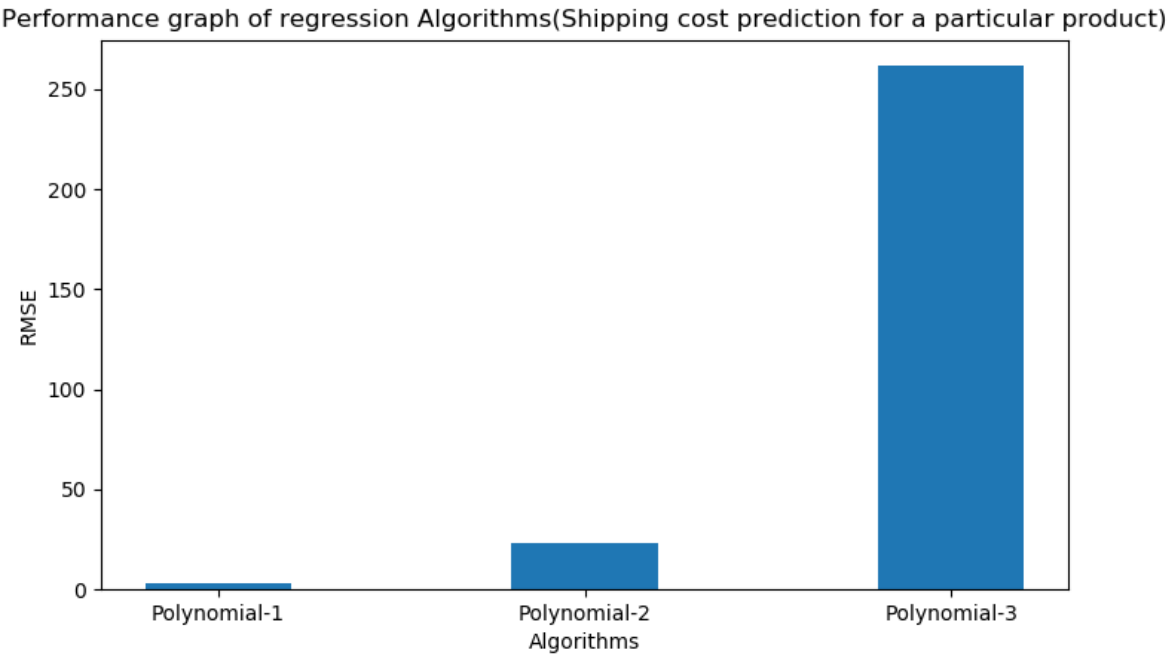

Figure 7: Performance graph for regression algorithms

(Shipping cost prediction for a particular product)

Finally we have built a model using polynomial regression that can predict the shipping cost of any product. We got reasonable RMSE while predicting shipping cost of a particular product. Now we try to predict the shipping cost of all products. After observing the values of Pearson Correlation, we see that there is no strong relationship with shipping cost. Only Sales have slightly strong relationship with shipping cost. We took only Sales as our input feature.

After building the model we see that it gives us lower RMSE value at polynomial regression with degree 2. The lowest RMSE value is 26.27. It seems that the error is high but this is the best we got. Using other features along with Sales doesn't improve the performance. If we select the features we have used for predicting the shipping cost of a particular product, then the RMSE rises again.

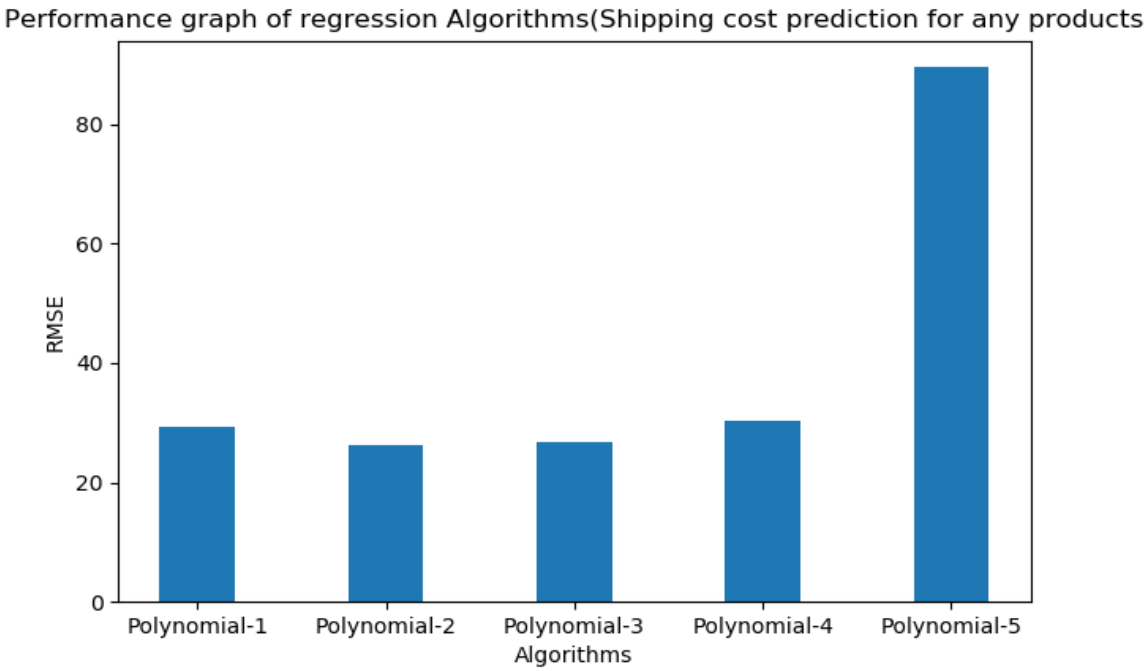

Figure 8: Performance graph for regression algorithms

(Shipping cost prediction for any product)

7.2 Result Analysis of Classification Algorithms:

In section 6.2.3 & 6.2.4 we have implemented KNN and logistic regression for predicting the order priority and shipping mode of any product from our dataset. We first select important features using RFECV. While predicting the order priority, we can see how the accuracy of logistic regression change from selecting the different number of features from the figure 9:

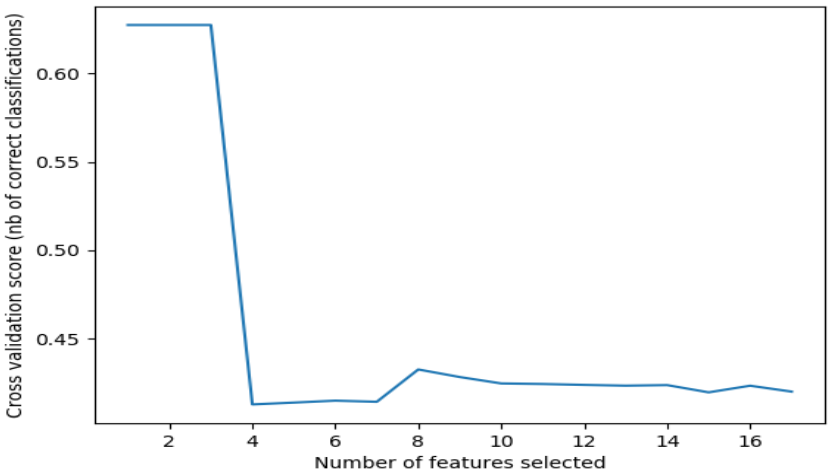

Figure 9: Accuracy vs. Number of features selected (Order priority prediction)

While predicting the order priority using KNN we run the algorithm from 1-700 and select the best k for which we get the maximum accuracy and build the model according to that. The accuracy vs. Value of K ranging from 500 to 600 is shown in figure 10.

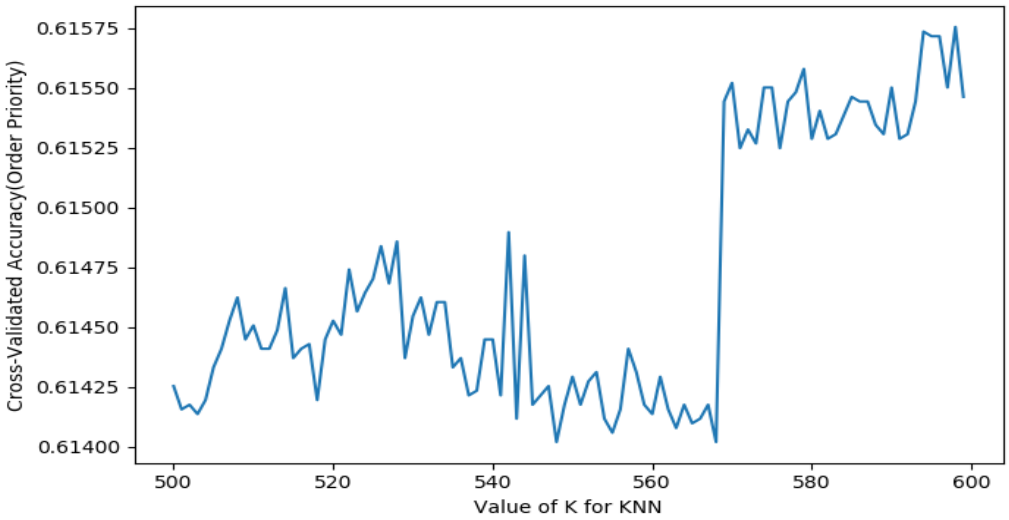

Figure 10: Accuracy vs. Value of K for KNN (Order priority prediction)

After running the model for k=1 to 700, we finally get our desired k value 595. Accuracy achieved using k is 61.57 %. We took the highest value of K. For KNN models, complexity is determined by the value of K .Lower values of K means more complexity. So we take the highest value of K that gives us the best accuracy. Logistic regression gives us 62.73 % accuracy.

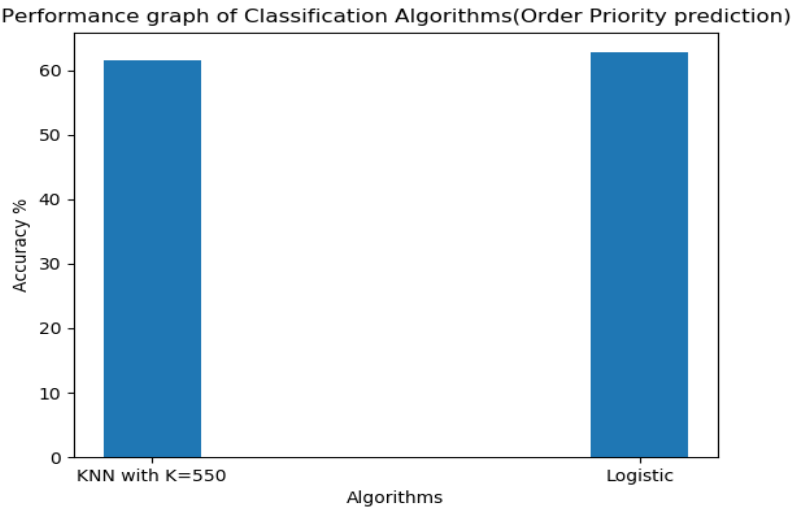

Figure 11: Performance graph of classification algorithms (Order Priority Prediction)

While predicting the shipping mode using KNN we run the algorithm from 1-60 and select the best k for which we get the maximum accuracy and build the model according to that. The accuracy vs. Value of K ranging from 1 to 60 is shown in figure 12:

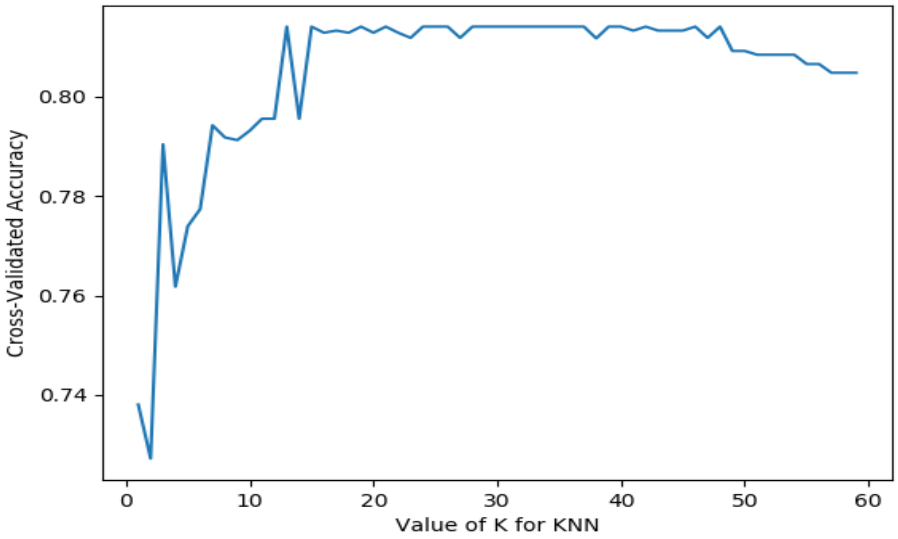

Figure 12: Accuracy vs. value of K for KNN (Shipping Mode Prediction)

After running the model for k=1 to 60, we have selected 35 as our k value. Accuracy achieved using k=35 is 81.40 %. Logistic regression gives us 78.12 % accuracy.

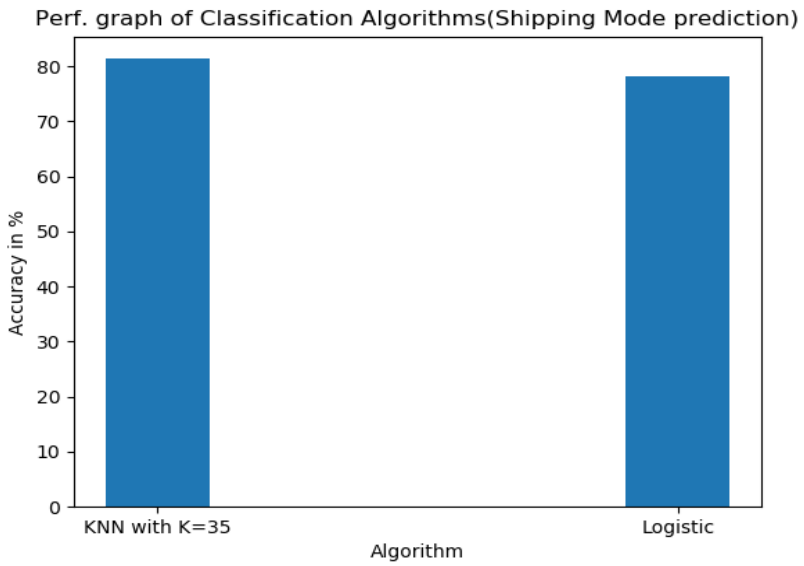

Figure 13: Performance graph of classification algorithms (Shipping Mode Prediction)

K-nearest neighbor and logistic regression both are excellent for classification problem. But can we make a conclusion that any of them is better than another? The answer is no. When we were predicting Order priority we had seen that Logistic regression gave us better accuracy. It also takes little time to build the model. But for KNN it requires more time than logistic regression since it was using 595 neighbors. Using more time, KNN did not give us better accuracy.

In case of predicting the shipment mode, we saw the reverse. KNN gave us better accuracy than logistic regression. The value of K is also reasonable so it took tolerable time.

The performance of these classification algorithms depends on the nature of data. Some algorithms are able to discover more insights in data and can improve their accuracy. Some algorithms work best on certain type of prediction. We are using the same data and features for prediction but the results are disparate. Same data affects different algorithms different way. In spite of using the same data & features, we get different accuracy.

8. Discussion and Future plan:

We have studied and implemented the two basic regression and classification algorithms and analyze their performance. In future, we will try to implement some advanced data mining algorithms like support vector machine. Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outlier's detection. We've also planned to apply Apriori algorithm, an important algorithm for transactional database. Then we will perform outlier analysis to reduce noise of data to improve our implemented algorithms performance. And finally, we will visualize our predictions and performances to represent the visual comparison of results.

9. Conclusion:

Data mining is an emerging field. With the growth of data production and manipulation fields, knowledge of data analysis and mining is becoming one of the most valuable assets day by day. In our work, we've tried to understand an in-depth overview of recent data mining techniques and algorithms. To achieve the goals, we've implemented the algorithms and studied their performances. From our study we've seen that the performance of any specific algorithm mostly depends on the pattern of the data. Same algorithm may provide different result on different data. So it is not possible to make any generalization to prioritize any specific algorithm over another.

References

1. Jiawei Han, M. K. (2011). *Data Mining: Concepts And Techniques* (Vol. III). USA: Morgan Kaufmann.
2. Murphy, K. P. (2012). *Machine Learning a Probabilistic Approach.* Cambridge: The MIT Press.
3. S.B. Soumya, N. D. (2016). Data Mining With Predictive Analytics for Financial Applications . *International Journal of Scientific Engineering and Applied Science, 2*(1), 8.
4. Usama Fayyad, G. P.-S. (1996). KDD-96: Proceedings : Second International Conference on Knowledge Discovery & Data Mining. *ASSOCIATION FOR THE ADVANCEMENT OF ARTIFICIAL INTELLIGENCE*, 391.
5. http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html **,** last accessed on 01.11.17
6. http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html, last accessed on 03.11.17
7. http://scikit-learn.org/stable/modules/neighbors.html, last accessed on 05.11.17
8. http://scikit-learn.org/stable/auto_examples/linear_model/plot_polynomial_interpolation.html, last accessed on 06.11.17
9. https://en.wikipedia.org/wiki/Cross-validation_(statistics) , last accessed on 28.10.17
10. https://docs.oracle.com/cd/B10501_01/server.920/a96520/concept.htm, last accessed 28.10.17
11. https://docs.oracle.com/cd/B10501_01/server.920/a96520/concept.htm
12. https://docs.oracle.com/cd/B19306_01/datamine.102/b14339/5dmtasks.htm
13. https://www.tableau.com/sites/default/files/training/global_superstore.zip