**Data Warehouse:**

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.

1. **Subject-oriented:**   
    DW can be used to analyze any subject.
2. **Integrated:**   
    DW integrates current and historical data from different sources.
3. **Time-variant:**   
    DW keeps historical data of
4. **Non-volatile collection of data:**  
    Contents of DW should not be changed. It is historical data.

**Data Warehouse Application on Data mining:**

Our main focus in this research is to study different mining algorithms. But why do we need Data Warehouse? To apply data mining algorithms, we have to preprocess the data. Data warehouse helps us pre-processing the data. 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.

**Schemas for Multidimensional Data:**

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**. A little review on these schemas are given below:

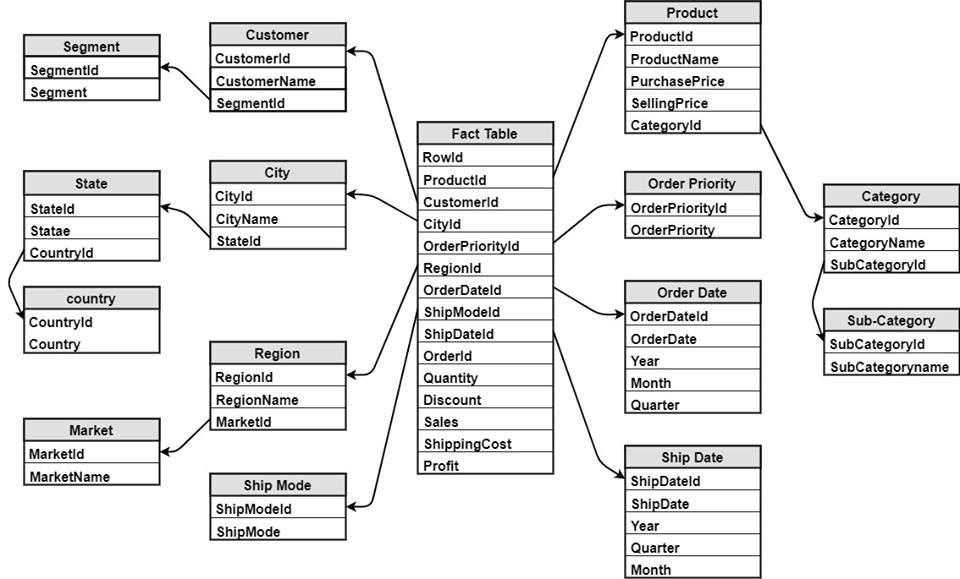
1. **Star Schema:**  
   The star schema architecture is the simplest data warehouse schema. It is called a star schema because the diagram resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables.  
     
   **Fact Table:**  
    The fact table contains business facts (or measures), and foreign keys which refer to candidate keys (normally primary keys) in the dimension tables. It also contains some measures.  
     
   **Dimension Table:**  
    Dimension tables contain descriptive attributes (or fields) that are typically textual fields (or discrete numbers that behave like text).
2. **Snow-flake Schema:**  The fact table has the same dimensions as it does in the star schema example. The most important difference is that the dimension tables in the snowflake schema are normalized. Interestingly, the process of normalizing dimension tables is called snowflaking.
3. **Fact constellation:** Fact constellation is a measure of online analytical processing, which is a collection of multiple fact tables sharing dimension tables, viewed as a collection of stars. This is an improvement over Star schema.  This schema is viewed as collection of stars hence called galaxy schema or fact constellation. Sophisticated application requires such schema.

**Data Warehouse Design:**

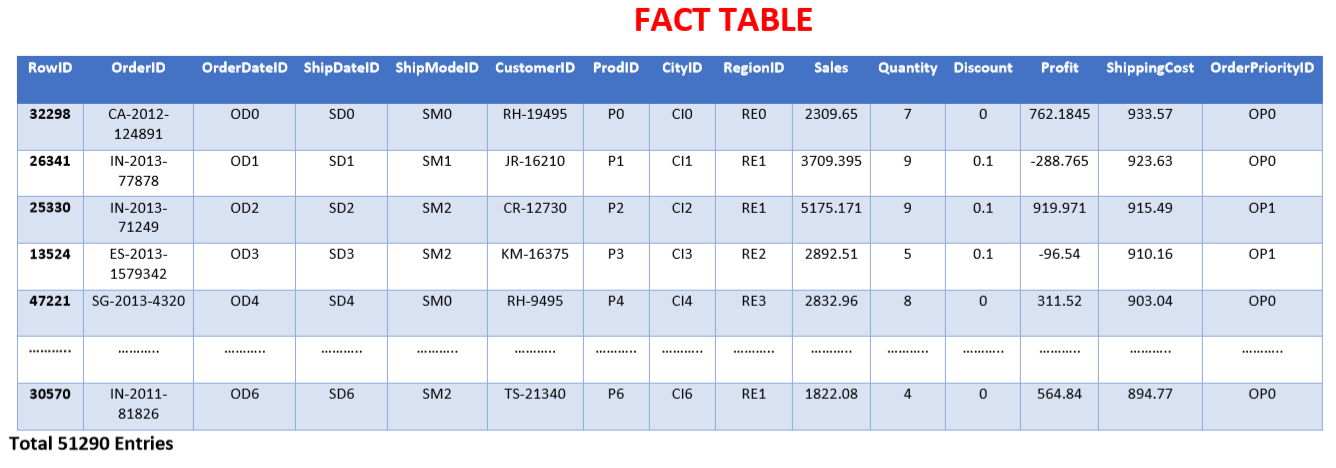
As we proposed earlier presentation, we have collected a dataset which contains a global superstores transactional data. We design our Data Warehouse using snow flake schema. The model, we have designed has fourteen dimension tables and one fact table. The fact table contains 15 attributes in which one 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.

Diagram of Data Warehouse:



**Fact Table:**

****

**Data Mining Process:**

Data mining is not an easy process. There are some very complicated steps required to complete the mining process. The whole process of data mining cannot be completed in a single step. In other words, we cannot get the required information from the large volumes of data as simple as that. It is a very complex process than we think involving a number of processes.   
It exists, however, in many variations on this theme, such as the [Cross Industry Standard Process for Data Mining](https://en.wikipedia.org/wiki/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. As we are working with transactional data of a global superstore, we need to understand what kind of business this superstore conducts and then process according to that.
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, correlated columns. In our research we have built a data warehouse which is basically data preprocessing.
4. **Modeling:**   
    Modeling is an important part of our research. Select and apply a variety of modelling 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. Modeling techniques we will use in our research are regression and classification.
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.

**Machine Learning:**

Machine learning is an orchestrated and repeatable pattern which systematically transforms and processes information to create prediction solutions. Here “an orchestrated and repeatable pattern” means using the same workflow it will not only define the problem but also build the solution. If we want to implement machine learning algorithm in our research we have to maintain a workflow. The steps of machine learning workflow are:

1. **Problem Definition:** The first and foremost step is 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 predict. But that is not enough. More precisely, we have to ask the right question with proper details. Then we can find a solution. For example, we want to predict the order priority of a product from our data set. First of all, we have to ensure that the dataset has enough information by which we can predict the order priority. In order to do that, we need to find the factors which are important to our problem. Then we need to define a goal. By goal here we mean we have to obtain more that 50% accuracy. If we have 50% accuracy that means the probability that our prediction is correct is 50 % which is not seems good. Half of our prediction will be wrong. So our goal will be to gain more than 70 % accuracy. So now we have the complete definition of our problem. Now we can say “We want to predict a product’s Order priority using our dataset at more than 70% accuracy”
2. **Making the dataset:**  
    After defining the problem completely now our main task is to prepare the dataset according to our problem. As we have mentioned earlier we have built a Data warehouse so half our preparing dataset is done. Now according to our goal we need to select the important features. Our final Dataset has 26 attributes. But not all our attributes are important to get the solution. We need to find out important factors which are important. Useless attributes will confuse the learning algorithm as a result the accuracy will be poor. There are many process which can be used to detect good features for the solution. We are using python scikit learn library so we need to prepare the data according to that. Machine learning algorithms only accepts numerical value. We need to convert non-numeric data into numeric data.
3. **Algorithm selection:**  
    The third step is to select algorithm according to our problem definition. There are three types of learning algorithm: supervised, unsupervised and semi-supervised.  
   In our research we will implement some supervised learning algorithms.  
    In supervised learning the computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that [maps](https://en.wikipedia.org/wiki/Map_(mathematics)) inputs to outputs. Input data is called training data and has a known label or result such as spam/not-spam or a stock price at a time. A model is prepared through a training process in which it is required to make predictions and is corrected when those predictions are wrong. The training process continues until the model achieves a desired level of accuracy on the training data. Example problems are classification and regression.  
    **Classification:**  
    When the target variable can take on only a small number of discrete values we call it a classification problem. From our dataset if we want to predict a product’s order priority then it is called classification problem. Orderpriority has only four discreate values : Low, high, medium and critical.  
    **Regression:**  
    When the target variable that we’re trying to predict is continuous, we call the learning problem a regression problem. For example, if we want to predict the profit earned by selling a particular product then it will be regression problem. Because we see that profit has continuous value.   
     
    One last important criteria for selecting a algorithm is that to find out it’s complexity. We will always try to take the less complexity algorithm.

### Model training: The purpose of this phase is to let specific data teach a Data mining algorithm to create specific prediction model. We will divide our dataset into two portions. One will be used to train the machine learning algorithm and the remaining part will be used for testing the model. There are some problems if we don’t divide our dataset in two portions. Problems with training and testing on the same data : a) Goal is to estimate likely performance of a model on out-of-sample data But, maximizing training accuracy rewards overly complex models that won't necessarily generalize

**b)** Unnecessarily complex models overfit the training data

The process of splitting the dataset into two portions is called Train-test split. The main purpose of this process is to split the dataset into two pieces: a **training set** and a **testing set**. We will train the model with training dataset and test the model with testing dataset. There are some disadvantages of train-test split process. It provides high variance estimate of out-of-sample data**. K-fold validation** overcomes this limitation. But train-test is still useful for its flexibility and speed. Details of K-fold will be discussed in the following section.

1. **Result analysis:** After training the model, we will analyze the result and decide which model satisfies our requirement.

**K-fold validation:**

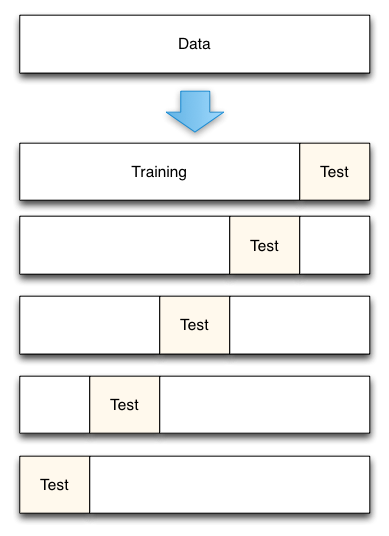
In train-test split, We split the dataset into two pieces, so that the model can be trained and tested on different data.Testing accuracy is a better estimate than training accuracy of out-of-sample performance

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 essense 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.

Diagram of **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)

**Implementation:**

We have implemented four machine learning algorithms. Name of these algorithms are given below:

1. Linear Regression,

2. Polynomial Regression,

3. Logistic Regression &

4. K-Nearest Neighbor.

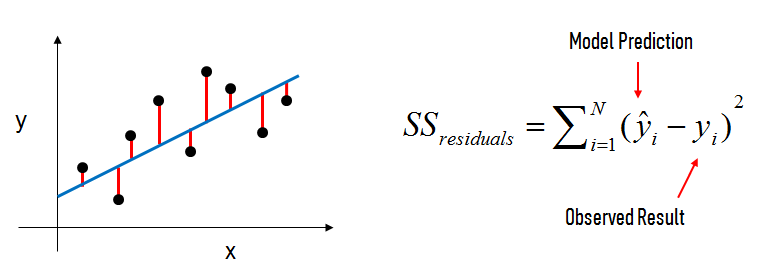
Here Linear regression, polynomial regression are regression algorithm and logistic regression, K-Nearest Neighbor are classification algorithm. Details of these algorithms and implementation are discussed in the following section.

**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:

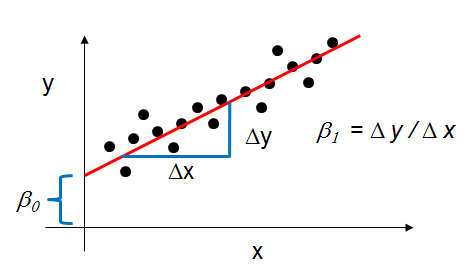
Here y is the response, x is the feature,  is the intercept,  is the coefficient for x. Together,   and   are called the **model coefficients.**

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"):

****

In this figure, the black dots are the observed values of x and y. The blue line is our least squaresline. 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 is the intercept (the value of  y when x=0) and  is the slope (the change in y divided by change in x ).Here is a graphical representation of those calculations:



## **Multiple Linear Regression:**

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

Each  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:

## **Model Evaluation Metrics for Regression:**

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

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

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

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

Here, is the original value of target variable and 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.

**Linear Regression Implementation:**

We have already discussed machine learning workflow. First of all, we need to specify what we really want to predict from our dataset, then prepare the dataset according that. After preparing the data we need select to an algorithm. Here we are implementing linear regression. So, we will build our model and check the result.

**Problem Definition:**  
 The first problem we want to solve is predicting the profit of a single product. After analyzing our dataset, we see 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.

**Preparing the dataset:**

First of all, we select the “Binney & Smith Sketch Pad, Blue” product from our dataset. We have mentioned earlier that machine learning algorithms only accept numeric value. So, we converted all our data accordingly and save that dataset separately. This data contains 21 attributes. But training the model we don’t need all 21 attributes.so we have to select the best features among them. There are many ways for feature selection for a regression problem like **Pearson Correlation**, chi-square etc. In our research we will use **Pearson Correlation**. It is easy to understand and implement. We can also see the relationship among the target variable and feature by plotting them in graph. This will give us a very clear view on which feature or attribute is important to us.

According to **Pearson Correlation we will select our features. We will store them in variable X. Our target variable is profit. So we will this attribute to another data-frame named Y.**

**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. The full name is the **Pearson Product Moment Correlation (PPMC)**. It shows the [linear relationship](http://www.statisticshowto.com/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:

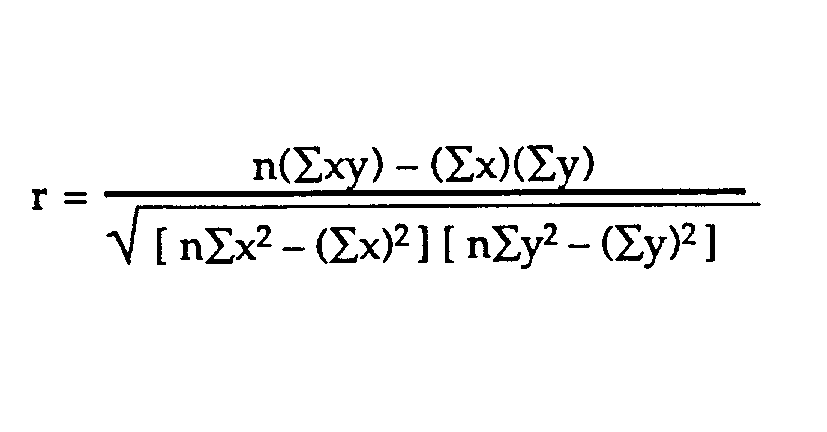
* Positive correlation– the other variable has a tendency to also increase;
* Negative correlation– the other variable has a tendency to decrease;
* No correlation – the other variable does not tend to either increase or decrease.

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.

   
 Negative No Positive

correlation correlation correlation  
  
  
 Pearson’s correlation coefficient is a statistical measure of the strength of a linearrelationship 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:   
 

### where x is the feature and y is the dependent variable or target variable and n is the number of samples.

**Algorithm selection:**

Linear regression will be used as the target variable’s value is continuous.

**Model training:**

We will use cross-validation to select and get the accuracy. RMSE metrics has been used. We don’t have much data so we will use 3-fold cross validation then calculated the average RMSE value. We will also use train-test split and what is the difference between cross validation and train-test split.

**Result Analysis:**

We get a RMSE value of 5.44440115183e-14 which is very low. But in train-test split we get 6.19861155074e-14. But if we select different portion of our data again 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 0.99999999999999933,'Quantity' is -34.29000000000002, and 'Discount' is 7.1054273576010019e-15. The intercept is 1.13686837722e-13.

If we see the co-efficient of the feature attributes then we see that Discount’s co-efficient is zero. Basically, this attribute has no impact on the final result. If we remove this attribute as our feature and check our score again then we see that our RMSE value has been decreased. More precisely, Discount attribute hampers our training. By removing it we get a better model.

**Polynomial Regression:**

Polynomial regression is a form of [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis) in which the relationship between the [independent variable](https://en.wikipedia.org/wiki/Independent_variable)  *x*  and the [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) *y*  is modelled as an n-th degree [polynomial](https://en.wikipedia.org/wiki/Polynomial) in *x*. Polynomial regression fits a nonlinear relationship between the value of *x* and the corresponding [conditional mean](https://en.wikipedia.org/wiki/Conditional_expectation) of y *.*

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. In our dataset we previous predicted a single product Profit from some attribute. But when we try to calculate the profit of every product then we can’t find a good linear model. When we use polynomial regression then we find a model that gives us least error. 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.

**Problem Definition:**  
 We want to predict the profit of any product. 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 using necessary factor. Here we will use full dataset that means all 51290 tuples to build our model.

**Preparing the dataset:**

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 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.**

**Algorithm selection:**

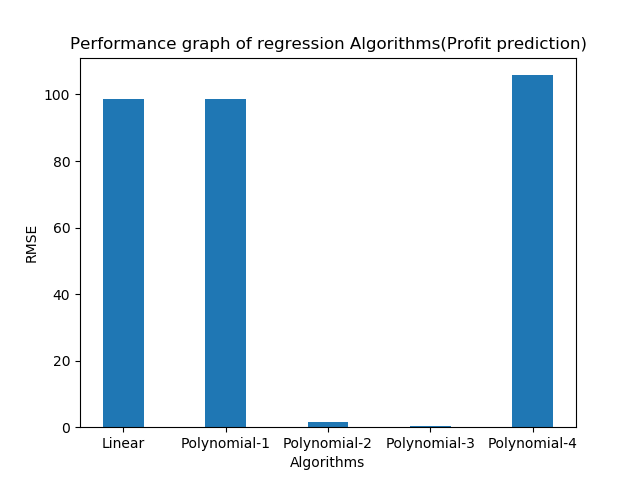
Polynomial regression will be used because the target variable is continuous and there is no strong linear relationship between feature vector and target variable.

**Model training:**

We will use cross-validation to select and get the accuracy. RMSE metrics has been used. Here we have plenty of data so we will use 10-fold cross validation then calculated the average RMSE value. We will run polynomial regression using degree 1 to 4 .

**Result Analysis:**

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 which is logical. Polynomial regression with degree 1 is nothing but simple linear regression. As we have already seen that there is no strong relationship here that why the RMSE is high. We get lowest value at degree 3. So, our equation will look like a Cubic expression.



**Some Other Prediction Using Regression Algorithms:**

The working procedure of linear and polynomial regression is known to us now. Let’s try to predict the shipping cost. Like before, first we try to build a model which predict the shipping cost of a single product. Then we will attempt to build 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.

**Problem Definition:**

Our goal is a build a model that can predict the shipping with least RMSE.

**Preparing Dataset:**

“Binney & Smith Sketch Pad, Blue” is the product that has been sold twenty-three times. 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.

**Algorithm Selection:**

Polynomial regression with degree 1 to 4 has been used. Since polynomial regression with degree 1 is Linear regression, we don’t need to implement linear regression separately.

**Model Building:**

We will use cross-validation to select and get the accuracy. RMSE metrics has been used. We don’t have much data so we will use 3-fold cross validation then calculated the average RMSE value.

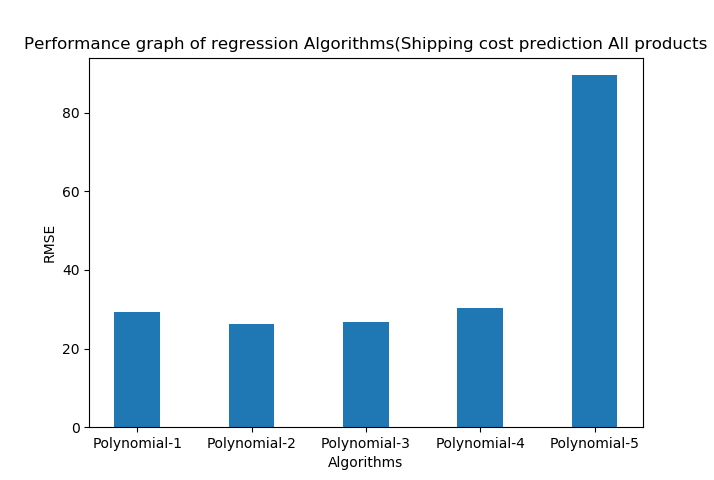
**Result Analysis:**

The average RMSE value is 2.50 obtained from linear regression. Since we have little amount of data polynomial regression gives us higher RMSE. 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 2.70. In the first case we took StateID as our input feature which has no linear relation ship 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. So not always Pearson correlation gave us best features selection.

**Shipping Cost Prediction Of all products:**

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. From previous study we can say that it will give higher RMSE value in linear regression.

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.



**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; θ) and P(Y=0|x; θ)*

Then we’ll chose a threshold value for our prediction function ***h*θ*(x)*** is 0.5

If ***hθ(x) ≥ 0.5*** then Y = 1 and if ***hθ(x) < 0.5*** then Y = 0;

Now, if we have the training set {(x(1) , y(1) ), (x(2) , y(2) ), ……., (x(m) , y(m) ),}

For m examples x = where x0 = 1 and Y € {0,1}

we can write our hypothesis of the logistic function as ***hθ(x) =* 1 /1+ e- θx .** 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(θ) = (hθ (x (i) ), y (i) )*

***=*** *(i) log hθ (x (i) ) + (1- y (i)) log(1- hθ (x (i) ))] ,* m is the number of total examples.

To fit the parameter *θ* we’ve to take the minimum of j(*θ*).

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

*J(θ) = (i) log hθ (x (i) ) + (1- y (i)) log(1- hθ (x (i) ))]*

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

*θj = θj – α j(θ)*

*θj = θj – α hθ (x (i) ) - y (i)) xj (i)*

where *α* is the learning rate of the equation.

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

**Logistic regression Implementation:**

**Problem Definition:**

Our goal is to detect the order priority of any product using classification algorithm. Since Order Priority has discrete value , we are using classification algorithm. We will try to obtain accuracy above 50 % .

**Preparing the dataset:**

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). Details of RFECV will be discussed in the following section. According to RFECV we select the best features from our dataset and then use them as a feature vector X and set OrderPriorityID as our target variable y.

**Recursive feature elimination with cross-validation:**

For classification with small training samples and high dimensionality, feature selection plays an important role in avoiding overfitting 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.

For predicting OrderPriorityID , we check the ranking of all attributes. Then select the feature which has rank 1. After running RFECV we get 3 features ranking 1. They are DateDif, Discount & CategoryID.

**Algorithm selection:**

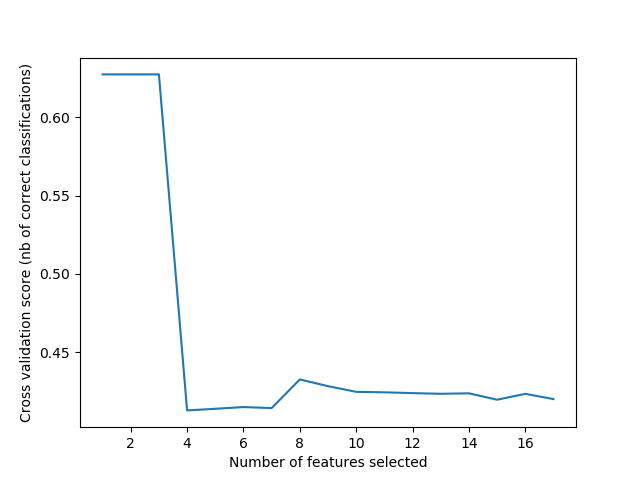
Logistic regression will be used because the target variable is continuous.

**Model training:**

We will use cross-validation to select and get the accuracy. Accuracy has been used as a scoring parameter. Here we have plenty of data so we will use 10-fold cross validation then calculated the average Accuracy.

**Result Analysis:**

Despite of selecting the best features we got 62.73% accuracy. Since we have product up to ten thousand and some of them only sold one time, it is difficult to find any pattern from the dataset we have.



**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 mismodeling 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 = {x1, …., xN}, Y = {Y1, ……, YN}.

Two distance measuring techniques are given bellow:

**Euclidian Distance:** ***dE (X,Y ) =***

***Manhattan Distance: dA (X,Y ) =***

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 datapoints D1,……,DP , where each datapoint contains N features Di = { Di1,……,Dnp }. A vector ***v*** with length P of output values **v = {v0,……vp}** accompanies this matrix, listing the output value **vi** for each datapoint Di .

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.

**Problem Definition:**

We will try to predict the order priority ID once again but this time we will use KNN. Lets see if it gives us better accuracy than logistic regression.

**Preparing the dataset:**

Using RFECV we will select the best ranking attributes as our feature vector X . The target variable y will be OrderPriorityID.

**Algorithm selection:**

K-nearest neighbor will be used because the target variable is continuous.

**Model training:**

We will use cross-validation as our model evaluation technique. Accuracy has been used as a scoring parameter. Here we have plenty of data so we will use 10-fold cross validation then calculated the average accuracy. We will check for which value of k we get the highest accuracy. Using the best k value we will build our model.

**Result Analysis:**

After running the model for k=1 to 700, we finally get our desired k value 550. Accuracy achieved using k is 61.50 %. 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.

**Some Other Prediction Using Classification Algorithms:**

In this part, we will try to predict the Ship Mode of any product using Classification algorithms and compare the result of two algorithms.

**Problem Definition:**

Predict the shipment mode using necessary factor from our dataset.

**Preparing Dataset:**

We have selected only one feature to our input variable. The target variable is ShipModeID.

**Algorithm Selection:**

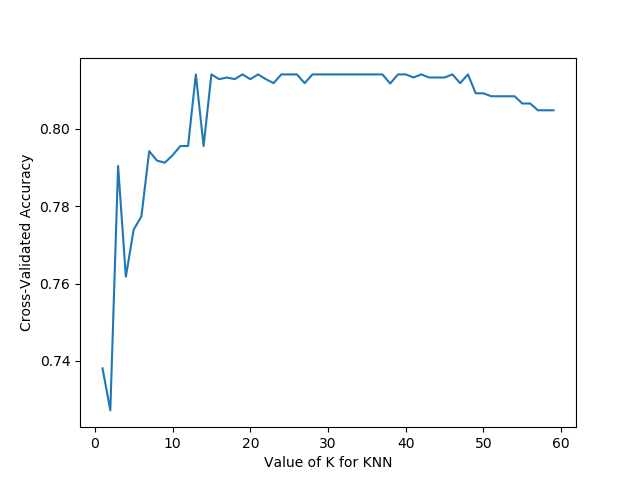
Since our target variable is discrete and we are studying four algorithms among which logistic and K-nearest neighbor are classification algorithm, we have selected them.

**Model Training:**

For KNN we will use neighbors value from 1 to 60 and select the best neighbor value.

For logistic regression simply train with input and output variable.

In both cases, 10-fold cross validation has been used.



**Result Analysis:**

Logistic regression gives us 78.11% accuracy. In case of KNN if we use 35 neighbors then we have accuracy 81.40 % . Both of the classification algorithms are giving us good accuracy. While we were studying our dataset we have seen that when the Shipment Date and Order Date are same then it belongs to Shipment mode ‘Same Day’. So here we have found a pattern.

**Analyzing the nature of classification Algorithms:**

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 a little bit more time than logistic regression since it was using 550 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 machine learning 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.

Reference:

1.https://docs.oracle.com/cd/B10501\_01/server.920/a96520/concept.htm

2. https://docs.oracle.com/cd/B19306\_01/datamine.102/b14339/5dmtasks.htm