**Big Mart Sales Prediction**

A Mini Project work submitted in partial fulfillment of the requirement for the award of the degree of

**[BACHELOR](https://www.google.co.in/search?q=BACHELOR&source=univ&tbm=nws&tbo=u&sa=X&ei=nwkoU4L4MYjmrAeB7IHYDw&ved=0CDkQqAI&biw=1366&bih=605) OF TECHNOLOGY**

**in**

**ELECTRONICS & COMMUNICATION ENGINEERING**

**by**

**V.Vishweshwar Reddy**

**Regd.No: 17211A04P4**

**Y.Shashank Reddy**

**Regd.No: 17211A04P9**

**K.Rakha Chandre**

**Regd.No: 17211A04K7**

**V.Vamshidhar**

**Regd.No: 17211A04P7**

**Under the esteemed guidance of**

**Dr.Syed Abudhagir Umar**

**Associate Professor**

****

**B.V.Raju Institute of Technology**

**UGC- Autonomous**

**Department of Electronics and Communication Engineering**

**Vishnupur, Narsapur, Medak.(Dt)**

**(Affiliated to JNTU, Hyderabad)**

**2019-2020**

**Department of Electronics & Communication Engineering**



**CERTIFICATE**

This is to certify that the Mini Project work entitled **Big Mart Sales Prediction** is being

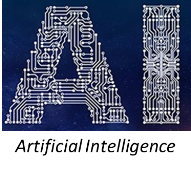
submitted by Mr.V.Vishweshwar Reddy**,** Mr.Y.Shashank Reddy, Mr.K.Rakha Chandre, Mr.V.Vamshidhar in partial fulfillment of the requirement for the award of the degree of **B.Tech. in Electronics & Communication Engineering**, by Jawaharlal Nehru Technological University Hyderabad is a record of bonafide work carried out by him under my guidance and supervision from 2020 to 2021.

The results presented in this project have been verified and are found to be satisfactory.

**INTERNAL GUIDE Dr.Sanjay Dubey., PhD**

**Professor & HOD, Dept. of ECE**

**EXTERNAL EXAMINER**

CENTER FOR EXCELLANCE

IN

ARTIFICIAL INTELLIGENCE AND MACHINE LARNING

**CERTIFICATE**

This is to certify that **V.Vishweshwar Reddy, Y.Shashank Reddy, K.Rakha Chandre, V.Vamshidhar** bearing Roll No: **17211A04P4, 17211A04P9, 17211A04K7,17211A04P7.** respectivelyhas successfully completed training on AI & Machine Learning and implemented a project tilted “**BIG MART SALES PREDICTION”** in Centre for Excellence In Artificial Intelligence and Machine Learning, Department of ECE, B.V.Raju Institute of Technology, Narsapur, during the period from 2020 to 2021.

|  |  |
| --- | --- |
| Coordinator | HOD |
| Dr. Syed Abudhagir Umar, Ph.D | Dr.Sanjay Dubey, Ph.D |

## ACKNOWLEDGEMENTS

First and foremost, we would like to express my gratitude to our advisor Mr.Syed Abudhagir Umar,Associate Professor,Ece Dept who has guided us throughout this thesis with his knowledge and experience.

Besides my advisor, we would like to thank our special lab AI & ML for the support,

knowledge and constant encouragement we have gotten over the years.

Name

V.Vishweswar Reddy 17211A04P4

Y.Shashank reddy 17211A04P9

K.Rakha Chandre 17211A04K7

V.Vamshidhar 17211A04P7

**ABSTRACT**

Nowadays shopping malls and Big Marts keep the track of their sales data of each and every individual item for predicting future demand of the customer and update the inventory management as well. These data stores basically contain a large number of customer data and individual item attributes in a data warehouse. Further, anomalies and frequent patterns are detected by mining the data store from the data warehouse. The resultant data can be used for predicting future sales volume with the help of different machine learning techniques for the retailers like Big Mart. In this paper, we propose a predictive models like Linear Regression,Random Forest,Decision Tree and KNN for predicting the sales of a company like Big Mart and found that the model produces better performance as compared to existing models. A comparative analysis of the model with others in terms performance metrics is also explained in details.

**CONTENTS**

CERTIFICATE I

ACKNOWLEDGEMENT II

ABSTRACT III

CONTENTS IV

LIST OF FIGURES & TABLES V

1. INTRODUCTION 1

1.1 Motivation 10

1.2 Objective 10 1.3 Scope 10

2. LITERATURE SURVEY 11

3. ANALYSIS & DESIGN 12

4. IMPLEMENTATION 34

Python Code 35

5. RESULTS 46

6. CONCLUSION AND FUTURE WORK 48

6.1 Conclusion 48

6.2 Future Work 48

REFERENCES 49

## LIST OF FIGURES

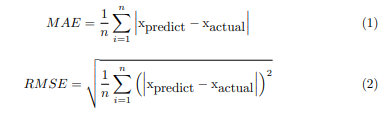
Figure 1 : Working Procedure of Proposed Model.

Figure 2: Univariet Distribution of target Variable Item Outlet Sales.

CHAPTER 1

## INTRODUCTION

Day by day competition among different shopping malls as well as big marts is getting more serious and aggressive only due to the rapid growth of the global malls and on-line shopping. Every mall or mart is trying to provide personalized and short-time offers for attracting more customers depending upon the day, such that the volume of sales for each item can be predicted for inventory management of the organization, logistics and transport service, etc. Present machine learning algorithm are very sophisticated and provide techniques to predict or forecast the future demand of sales for an organization, which also helps in overcoming the cheap availability of computing and storage systems. In this paper, we are addressing the problem of big mart sales prediction or forecasting of an item on customer’s future demand in different big mart stores across various locations and products based on the previous record. Different machine learning algorithms like linear regression analysis, random forest, etc are used for prediction or forecasting of sales volume. As good sales are the life of every organization so the forecasting of sales plays an important role in any shopping complex. Always a better prediction is helpful, to develop as well as to enhance the strategies of business about the marketplace which is also helpful to improve the knowledge of marketplace. A standard sales prediction study can help in deeply analyzing the situations or the conditions previously occurred and then, the inference can be applied about customer acquisition, funds inadequacy and strengths before setting a budget and marketing plans for the upcoming year. In other words, sales prediction is based on the available resources from the past. In depth knowledge of past is required for enhancing and improving the likelihood of marketplace irrespective of any circumstances especially the external circumstance, which allows to prepare the upcoming needs for the business. Extensive research is going on in retailers domain for forecasting the future sales demand. The basic and foremost technique used in predicting sale is the statistical methods, which is also known as the traditional method, but these methods take much more time for predicting a sales also these methods could not handle non linear data so to over these problems in traditional methods machine learning techniques are deployed. Machine learning techniques can not only handle non-linear data but also huge data-set efficiently. To measure the performance of the models, Root Mean Square Error (RMSE) [15] and Mean Absolute Error (MAE) [4] are used as an evaluation metric as mentioned in the Equation 1 and 2 respectively. Here Both metrics are used as the parameter for accuracy measure of a continuous variable.



where n: total number of error and | xpredict – xactual | : Absolute error. The remaining part of this article is arranged as following: Section 1 briefly describes introduction of sales prediction of Big Mart and also elaborate about the evaluation metric used in the model. Previous related work has been pointed in Section 2. The detailed description and analysis of proposed model is given in Section 3. Where as implementations and results are demonstrated in Section 4 and the paper concludes with a conclusion in the last section.

1.1.MOTIVATION:

Estimating future sales is an important aspect of any business. Accurate prediction of future sales help companies to develop and improve business strategies as well as to gain proper market knowledge. Standard sales forecast helps companies to analyze the situation which has occurred before and then apply customer purchase inferences to identify inadequacies and weaknesses before budgeting as well as to prepare good plan for the next year. A detailed knowledge of past opportunities permits one to plan for future market needs and increase the possibility of success Regardless of external factors, firms which see sales modeling as its first step towards improved performance compared to those who don't.

1.2.OBJECTIVE:

The aim is to build a predictive model and find out the sales of each product at a particular store.

Using this model, BigMart will try to understand the properties of products and stores which play a key role in increasing sales.

1.3.SCOPE:

The data scientists at BigMart have collected 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a particular store.

Using this model, BigMart will try to understand the properties of products and stores which play a key role in increasing sales.

CHAPTER 2

LITERATURE SURVEY

Machine Learning is defined as the computer program which learns by itself from its experience without any human interference. Research on sales prediction has been done and some of them has been discussed below: In paper[1], general linear approach, decision tree approach and good gradient approach were used to predict sales. The initial data set considered included many entries, but the final data set which is used for analyzing was much smaller than the original as it consists of non-usable data, redundant entries and insignificant sales data. In paper[2], linear regression method has been organized into structured data. Then it involves modeling data for predictions using machine learning techniques where the expected accuracy was 84%. In paper[3], they used linear regression and XG booster algorithm to forecast sales that included data collection and translation into processed data. Ultimately, they predicted which model would produce the better outcome. In paper[4],sales were predicted using three modules that are hive, R programming and tableau. By analysing the stores history which helps get an understanding of the store's revenue to make some improvements to the target so it can be more successful. Within the diagram, key values are obtained to reduce all intermediate values by reducing the intermediate key feature to obtain the results. Mohit Gurnani in his research proves that composite models achieve good results in comparison to individual models. He also stated that decomposition mechanisms are far better than hybrid mechanisms [5]. J. Scott Armstrong in his research discussed about predicting solutions to interesting and difficult sales forecasting problems [6]. Samaneh Beheshti-Kashi in his research reviewed different Various approaches on the predictive potential of consumer-generated content and search queries [7]. Gopal Behera has done effective study on Big mart sales prediction and has given prediction metrics for various existing models [8]. In this paper, we use random forest and XG booster methodology in which raw data obtained at large mart will be pre-processed for missing data, anomalies and outliers. Then an algorithm will be used to predict the final results. ETL stands for Extract, Transform and Load and finally we compare all the models and predict which model gives accurate result.

CHAPTER 3

## ANALYSIS AND DESIGN

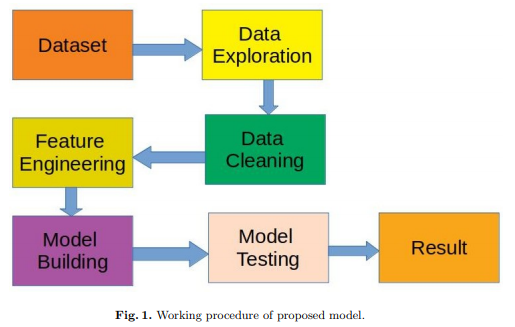
Sales forecasting as well as analysis of sale forecasting has been conducted by many authors as summarized: The statistical and computational methods are studied in [2] also this paper elaborates the automated process of knowledge acquisition. Machine learning [6] is the process where a machine will learn from data in the form of statistically or computationally method and process knowledge acquisition from experiences. Various machine learning (ML) techniques with their applications in different sectors has been presented in [2]. Pat Langley and Herbert A [7] pointed out most widely used data mining technique in A Comparative Study of Big Mart Sales Prediction 3 the field of business is the Rule Induction (RI) technique as compared to other data mining techniques. Where as sale prediction of a pharmaceutical distribution company has been described in [12,10]. Also this paper focuses on two issues: (i) stock state should not undergo out of stock, and (ii) it avoids the customer dissatisfaction by predicting the sales that manages the stock level of medicines. Handling of footwear sale fluctuation in a period of time has been addressed in [5]. Also this paper focuses on using neural network for predicting of weekly retail sales, which decrease the uncertainty present in the short term planning of sales. Linear and non-linear [3] a comparative analysis model for sales forecasting is proposed for the retailing sector. Beheshti-Kashi and Samaneh [1] performed sales prediction in fashion market. A two level statistical method [11] is elaborated for forecasting the big mart sales prediction. Xia and Wong [17] proposed the differences between classical methods (based on mathematical and statistical models) and modern heuristic methods and also named exponential smoothing, regression, auto regressive integrated moving average (ARIMA), generalized auto regressive conditionally heteroskedastic (GARCH) methods. Most of these models are linear and are not able to deal with the asymmetric behavior in most real-world sales data [9]. Some of the challenging factors like lack of historical data, consumer-oriented markets face uncertain demands, and short life cycles of prediction methods results in inaccurate forecast.

**Proposed System:**

For building a model to predict accurate results the dataset of Big Mart sales undergoes several sequence of steps as mentioned in Figure 1 and in this work we propose a model using Xgboost technique. Every step plays a vital role for building the proposed model. In our model we have used 2013 Big mart dataset [13]. After preprocessing and filling missing values, we used ensemble classifier using Decision trees, Linear regression, Ridge regression, Random forest and Xgboost. Both MAE and RSME are used as accuracy metrics for predicting the sales in Big Mart. From the accuracy metrics it was found that the model will predict best using minimum MAE and RSME. The details of the proposed method is explained in the following section.

**Dataset Description of Big Mart:**

In our work we have used 2013 Sales data of Big Mart as the dataset. Where the dataset consists of 12 attributes like Item Fat, Item Type, Item MRP, Outlet Type, Item Visibility, Item Weight, Outlet Identifier, Outlet Size, Outlet Establishment Year, Outlet Location Type, Item Identifier and Item Outlet Sales. Out of these attributes response variable is the Item Outlet Sales attribute and remaining attributes are used as the predictor variables. The data-set consists of 8523 products across different cities and locations. The data-set is also based on hypotheses of store level and product level. Where store level involves attributes like: city, population density, store capacity, location, etc and the product level



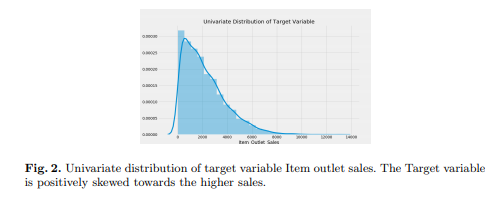
hypotheses involves attributes like: brand, advertisement, promotional offer, etc. After considering all, a dataset is formed and finally the data-set was divided into two parts, training set and test set in the ratio 80 : 20.

**Data Exploration:**

In this phase useful information about the data has been extracted from the dataset. That is trying to identify the information from hypotheses vs available data. Which shows that the attributes Outlet size and Item weight face the problem of missing values, also the minimum value of Item Visibility is zero which is not actually practically possible. Establishment year of Outlet varies from 1985 to 2009. These values may not be appropriate in this form. So, we need to convert them into how old a particular outlet is. There are 1559 unique products, as well as 10 unique outlets, present in the dataset. The attribute Item type contains 16 unique values. Where as two types of Item Fat Content are there but some of them are misspelled as regular instead of ’Regular’ and low fat, LF instead of Low Fat. From Figure 2. It was found that the response variable i.e. Item Outlet Sales was positively skewed. So, to remove the skewness of response variable a log operation was performed on Item Outlet Sales.

**Data Cleaning:**

It was observed from the previous section that the attributes Outlet Size and Item Weight has missing values. In our work in case of Outlet Size missing

****

value we replace it by the mode of that attribute and for the Item Weight missing values we replace by mean of that particular attribute. The missing attributes are numerical where the replacement by mean and mode diminishes the correlation among imputed attributes. For our model we are assuming that there is no relationship between the measured attribute and imputed attribute.

**Feature Engineering:**

Some nuances were observed in the data-set during data exploration phase. So this phase is used in resolving all nuances found from the dataset and make them ready for building the appropriate model. During this phase it was noticed that the Item visibility attribute had a zero value, practically which has no sense. So the mean value item visibility of that product will be used for zero values attribute. This makes all products likely to sell. All categorical attributes discrepancies are resolved by modifying all categorical attributes into appropriate ones. In some cases, it was noticed that non-consumables and fat content property are not specified. To avoid this we create a third category of Item fat content i.e. none. In the Item Identifier attribute, it was found that the unique ID starts with either DR or FD or NC. So, we create a new attribute Item Type New with three categories like Foods, Drinks and Non-consumables. Finally, for determining how old a particular outlet is, we add an additional attribute Year to the dataset.

**Model Building:**

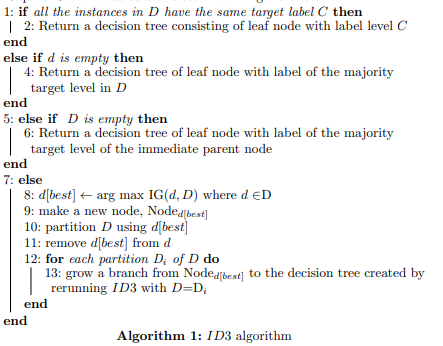
After completing the previous phases, the dataset is now ready to build proposed model. Once the model is build it is used as predictive model to forecast sales of Big Mart. In our work, we propose a model using Xgboost algorithm and compare it with other machine learning techniques like Linear regression, Ridge regression [14], Decision tree [8,16] etc. Decision Tree: A decision tree classification is used in binary classification problem and it uses entropy [8] and information gain [16] as metric and is defined in Equation 3 and Equation 4 respectively for classifying an attribute which picks the highest information gain attribute to split the data set.

****

where H(S): Entropy, C: Class Label, P:Probability of class c.

****

where S: Set of attribute or dataset, H(S): Entropy of set S, T: Subset created from splitting of S by attribute A. p(t): Proportion of the number of elements in t to number of element in the set S. H(t): Entropy of subset t. The decision tree algorithm is depicted in Algorithm 1. Require: Set of features d and set of training instances D



The Algorithms used in this Framework are:

**Linear Regression:**

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

Application

Description automatically generated with medium confidence

**Here,**

Y= Dependent Variable (Target Variable)  
X= Independent Variable (predictor Variable)  
a0= intercept of the line (Gives an additional degree of freedom)  
a1 = Linear regression coefficient (scale factor to each input value).  
ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

## Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

**Simple Linear Regression:**  
If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

**Multiple Linear regression:**  
If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

## Linear Regression Line:

A linear line showing the relationship between the dependent and independent variables is called a **regression line**. A regression line can show two types of relationship:

* **Positive Linear Relationship:**  
  If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.

Diagram

Description automatically generated

* **Negative Linear Relationship:**  
  If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.

Diagram

Description automatically generated

## Finding the best fit line:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

The different values for weights or the coefficient of lines (a0, a1) gives a different line of regression, so we need to calculate the best values for a0 and a1 to find the best fit line, so to calculate this we use cost function.

### **Cost function-**

* The different values for weights or coefficient of lines (a0, a1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.
* Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.
* We can use the cost function to find the accuracy of the **mapping function**, which maps the input variable to the output variable. This mapping function is also known as **Hypothesis function**.

For Linear Regression, we use the **Mean Squared Error (MSE)** cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

For the above linear equation, MSE can be calculated as:

Text

Description automatically generated

**Where,**

N=Total number of observation  
Yi = Actual value  
(a1xi+a0)= Predicted value.

**Residuals:** The distance between the actual value and predicted values is called residual. If the observed points are far from the regression line, then the residual will be high, and so cost function will high. If the scatter points are close to the regression line, then the residual will be small and hence the cost function.

### **Gradient Descent:**

* Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.
* A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.
* It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

## Model Performance:

The Goodness of fit determines how the line of regression fits the set of observations. The process of finding the best model out of various models is called **optimization**. It can be achieved by below method:

**1. R-squared method:**

* R-squared is a statistical method that determines the goodness of fit.
* It measures the strength of the relationship between the dependent and independent variables on a scale of 0-100%.
* The high value of R-square determines the less difference between the predicted values and actual values and hence represents a good model.
* It is also called a **coefficient of determination,** or **coefficient of multiple determination** for multiple regression.
* It can be calculated from the below formula:

Text

Description automatically generated

## Assumptions of Linear Regression

Below are some important assumptions of Linear Regression. These are some formal checks while building a Linear Regression model, which ensures to get the best possible result from the given dataset.

* **Linear relationship between the features and target:**  
  Linear regression assumes the linear relationship between the dependent and independent variables.
* **Small or no multicollinearity between the features:**  
  Multicollinearity means high-correlation between the independent variables. Due to multicollinearity, it may difficult to find the true relationship between the predictors and target variables. Or we can say, it is difficult to determine which predictor variable is affecting the target variable and which is not. So, the model assumes either little or no multicollinearity between the features or independent variables.
* **Homoscedasticity Assumption:**  
  Homoscedasticity is a situation when the error term is the same for all the values of independent variables. With homoscedasticity, there should be no clear pattern distribution of data in the scatter plot.
* **Normal distribution of error terms:**  
  Linear regression assumes that the error term should follow the normal distribution pattern. If error terms are not normally distributed, then confidence intervals will become either too wide or too narrow, which may cause difficulties in finding coefficients.  
  It can be checked using the **q-q plot**. If the plot shows a straight line without any deviation, which means the error is normally distributed.
* **No autocorrelations:**  
  The linear regression model assumes no autocorrelation in error terms. If there will be any correlation in the error term, then it will drastically reduce the accuracy of the model. Autocorrelation usually occurs if there is a dependency between residual errors.

# Decision Tree Classification Algorithm

* Decision Tree is a **Supervised learning technique**that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where**internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
* In a Decision tree, there are two nodes, which are the **Decision Node** and**Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* The decisions or the test are performed on the basis of features of the given dataset.
* **It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.**
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
* A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
* Below diagram explains the general structure of a decision tree:

Diagram

Description automatically generated

## Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

* Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
* The logic behind the decision tree can be easily understood because it shows a tree-like structure.

## Decision Tree Terminologies

 **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

 **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

 **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

 **Branch/Sub Tree:** A tree formed by splitting the tree.

 **Pruning:** Pruning is the process of removing the unwanted branches from the tree.

 **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

**How does the Decision Tree algorithm Work?**

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

* **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
* **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
* **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
* **Step-4:** Generate the decision tree node, which contains the best attribute.
* **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

## Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.**By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

* **Information Gain**
* **Gini Index**

### **1. Information Gain:**

* Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
* It calculates how much information a feature provides us about a class.
* According to the value of information gain, we split the node and build the decision tree.
* A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

1. Information Gain= Entropy(S)- [(Weighted Avg) \*Entropy(each feature)

**Entropy:** Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

**Where,**

* **S= Total number of samples**
* **P(yes)= probability of yes**
* **P(no)= probability of no**

### **2. Gini Index:**

* Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
* An attribute with the low Gini index should be preferred as compared to the high Gini index.
* It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
* Gini index can be calculated using the below formula:

Gini Index= 1- ∑jPj2

## Pruning: Getting an Optimal Decision tree

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree.

A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning. There are mainly two types of tree **pruning**technology used:

* **Cost Complexity Pruning**
* **Reduced Error Pruning.**

## Advantages of the Decision Tree

* It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
* It can be very useful for solving decision-related problems.
* It helps to think about all the possible outcomes for a problem.
* There is less requirement of data cleaning compared to other algorithms.

## Disadvantages of the Decision Tree

* The decision tree contains lots of layers, which makes it complex.
* It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
* For more class labels, the computational complexity of the decision tree may increase.

**[Next →](https://www.javatpoint.com/machine-learning-interview-questions)[← Prev](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm)**

# Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

**The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

Diagram

Description automatically generated

## Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

* There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
* The predictions from each tree must have very low correlations.

## Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

## How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

## Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

## Advantages of Random Forest

* Random Forest is capable of performing both Classification and Regression tasks.
* It is capable of handling large datasets with high dimensionality.
* It enhances the accuracy of the model and prevents the overfitting issue.

## Disadvantages of Random Forest

* Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

# K-Nearest Neighbor(KNN) Algorithm

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Graphical user interface, application

Description automatically generated

## How does K-NN work?

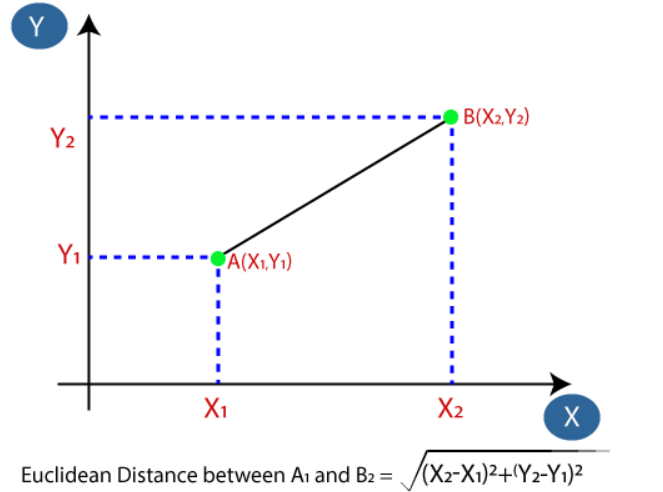
The K-NN working can be explained on the basis of the below algorithm:

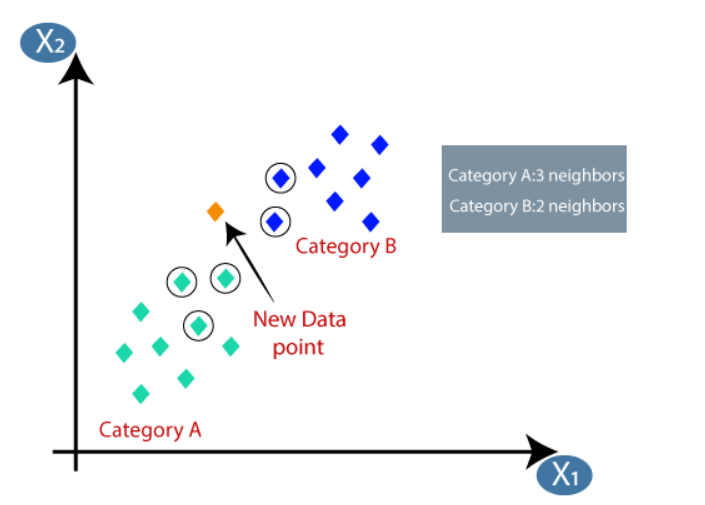
* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

Chart, scatter chart

Description automatically generated

* Firstly, we will choose the number of neighbors, so we will choose the k=5.
* Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as: It can be calculated as: 
* By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



* As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

## How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

## Advantages of KNN Algorithm:

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

## Disadvantages of KNN Algorithm:

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

Chapter 4

**IMPLEMENTATION**

**HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system does and not how it should be implemented.

* PROCESSOR : DUAL CORE 2 DUOS.
* RAM : 4GB DD RAM
* HARD DISK : 250 GB

**SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

**SOFTWARE REQUIREMENTS**

* Operating System : Windows 7/8/10
* Platform : Jupyter Lab
* Programming Language : Python
* Front End : Jupyter Lab

**FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly; the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

**NON-FUNCTIONAL REQUIREMENTS**

**EFFICIENCY**

Our multi-modal event tracking and evolution framework is suitable for multimedia documents from various social media platforms, which can not only effectively capture their multi-modal topics, but also obtain the evolutionary trends of social events and generate effective event summary details over time. Our proposed mmETM model can exploit the multi-modal property of social event, which can effectively model social media documents including long text with related images and learn the correlations between textual and visual modalities to separate the visual-representative topics and non-visual-representative topics.

PYTHON CODE FOR BIG MART SALES PREDICTION

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

data\_train=pd.read\_csv(r'C:\Python3\projects\BigMart Sales Data\train.csv')

data\_test=pd.read\_csv(r'C:\Python3\projects\BigMart Sales Data\test.csv')

data\_train

| **Item\_Identifier** | **Item\_Weight** | **Item\_Fat\_Content** | **Item\_Visibility** | **Item\_Type** | **Item\_MRP** | **Outlet\_Identifier** | **Outlet\_Establishment\_Year** | **Outlet\_Size** | **Outlet\_Location\_Type** | **Outlet\_Type** | **Item\_Outlet\_Sales** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | FDA15 | 9.300 | Low Fat | 0.016047 | Dairy | 249.8092 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 | 3735.1380 |
| **1** | DRC01 | 5.920 | Regular | 0.019278 | Soft Drinks | 48.2692 | OUT018 | 2009 | Medium | Tier 3 | Supermarket Type2 | 443.4228 |
| **2** | FDN15 | 17.500 | Low Fat | 0.016760 | Meat | 141.6180 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 | 2097.2700 |
| **3** | FDX07 | 19.200 | Regular | 0.000000 | Fruits and Vegetables | 182.0950 | OUT010 | 1998 | NaN | Tier 3 | Grocery Store | 732.3800 |

data\_test

| **Item\_Identifier** | **Item\_Weight** | **Item\_Fat\_Content** | **Item\_Visibility** | **Item\_Type** | **Item\_MRP** | **Outlet\_Identifier** | **Outlet\_Establishment\_Year** | **Outlet\_Size** | **Outlet\_Location\_Type** | **Outlet\_Type** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | FDW58 | 20.750 | Low Fat | 0.007565 | Snack Foods | 107.8622 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 |
| **1** | FDW14 | 8.300 | reg | 0.038428 | Dairy | 87.3198 | OUT017 | 2007 | NaN | Tier 2 | Supermarket Type1 |
| **2** | NCN55 | 14.600 | Low Fat | 0.099575 | Others | 241.7538 | OUT010 | 1998 | NaN | Tier 3 | Grocery Store |
| **3** | FDQ58 | 7.315 | Low Fat | 0.015388 | Snack Foods | 155.0340 | OUT017 | 2007 | NaN | Tier 2 | Supermarket Type1 |
| **4** | FDY38 | NaN | Regular | 0.118599 | Dairy | 234.2300 | OUT027 | 1985 | Medium | Tier 3 | Supermarket Type3 |
| **5** | FDH56 | 9.800 | Regular | 0.063817 | Fruits and Vegetables | 117.1492 | OUT046 | 1997 | Small | Tier 1 | Supermarket Type1 |
|  |  |  |  |  |  |  |  |  |  |  |  |

data\_train.describe()

|  | **Item\_Weight** | **Item\_Visibility** | **Item\_MRP** | **Outlet\_Establishment\_Year** | **Item\_Outlet\_Sales** |
| --- | --- | --- | --- | --- | --- |
| **count** | 7060.000000 | 8523.000000 | 8523.000000 | 8523.000000 | 8523.000000 |
| **mean** | 12.857645 | 0.066132 | 140.992782 | 1997.831867 | 2181.288914 |
| **std** | 4.643456 | 0.051598 | 62.275067 | 8.371760 | 1706.499616 |
| **min** | 4.555000 | 0.000000 | 31.290000 | 1985.000000 | 33.290000 |
| **25%** | 8.773750 | 0.026989 | 93.826500 | 1987.000000 | 834.247400 |
| **50%** | 12.600000 | 0.053931 | 143.012800 | 1999.000000 | 1794.331000 |
| **75%** | 16.850000 | 0.094585 | 185.643700 | 2004.000000 | 3101.296400 |
| **max** | 21.350000 | 0.328391 | 266.888400 | 2009.000000 | 13086.964800 |

data\_train.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 8523 entries, 0 to 8522

Data columns (total 12 columns):

Item\_Identifier 8523 non-null object

Item\_Weight 7060 non-null float64

Item\_Fat\_Content 8523 non-null object

Item\_Visibility 8523 non-null float64

Item\_Type 8523 non-null object

Item\_MRP 8523 non-null float64

Outlet\_Identifier 8523 non-null object

Outlet\_Establishment\_Year 8523 non-null int64

Outlet\_Size 6113 non-null object

Outlet\_Location\_Type 8523 non-null object

Outlet\_Type 8523 non-null object

Item\_Outlet\_Sales 8523 non-null float64

dtypes: float64(4), int64(1), object(7)

memory usage: 566.0+ KB

data\_test.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 5681 entries, 0 to 5680

Data columns (total 11 columns):

Item\_Identifier 5681 non-null object

Item\_Weight 4705 non-null float64

Item\_Fat\_Content 5681 non-null object

Item\_Visibility 5681 non-null float64

Item\_Type 5681 non-null object

Item\_MRP 5681 non-null float64

Outlet\_Identifier 5681 non-null object

Outlet\_Establishment\_Year 5681 non-null int64

Outlet\_Size 4075 non-null object

Outlet\_Location\_Type 5681 non-null object

Outlet\_Type 5681 non-null object

dtypes: float64(3), int64(1), object(7)

memory usage: 332.9+ KB

data\_train.isnull().sum()

Item\_Identifier 0

Item\_Weight 1463

Item\_Fat\_Content 0

Item\_Visibility 0

Item\_Type 0

Item\_MRP 0

Outlet\_Identifier 0

Outlet\_Establishment\_Year 0

Outlet\_Size 2410

Outlet\_Location\_Type 0

Outlet\_Type 0

Item\_Outlet\_Sales 0

dtype: int64

data\_test.isnull().sum()

Item\_Identifier 0

Item\_Weight 976

Item\_Fat\_Content 0

Item\_Visibility 0

Item\_Type 0

Item\_MRP 0

Outlet\_Identifier 0

Outlet\_Establishment\_Year 0

Outlet\_Size 1606

Outlet\_Location\_Type 0

Outlet\_Type 0

dtype: int64

x1=data\_train['Item\_Weight'].mean()

y1=data\_train['Outlet\_Size'].mode()[0]

data\_train['Item\_Weight']=data\_train['Item\_Weight'].fillna(data\_train['Item\_Weight'].mean())

data\_train['Outlet\_Size']=data\_train['Outlet\_Size'].fillna(data\_train['Outlet\_Size'].mode()[0])

data\_train

| **Item\_Identifier** | **Item\_Weight** | **Item\_Fat\_Content** | **Item\_Visibility** | **Item\_Type** | **Item\_MRP** | **Outlet\_Identifier** | **Outlet\_Establishment\_Year** | **Outlet\_Size** | **Outlet\_Location\_Type** | **Outlet\_Type** | **Item\_Outlet\_Sales** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | FDA15 | 9.300000 | Low Fat | 0.016047 | Dairy | 249.8092 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 | 3735.1380 |
| **1** | DRC01 | 5.920000 | Regular | 0.019278 | Soft Drinks | 48.2692 | OUT018 | 2009 | Medium | Tier 3 | Supermarket Type2 | 443.4228 |
| **2** | FDN15 | 17.500000 | Low Fat | 0.016760 | Meat | 141.6180 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 | 2097.2700 |
| **3** | FDX07 | 19.200000 | Regular | 0.000000 | Fruits and Vegetables | 182.0950 | OUT010 | 1998 | Medium | Tier 3 | Grocery Store | 732.3800 |

x=data\_test['Item\_Weight'].mean()

y=data\_test['Outlet\_Size'].mode()[0]

print(x," ",y)

12.695633368756374 Medium

data\_test['Item\_Weight']=data\_test['Item\_Weight'].fillna(data\_test['Item\_Weight'].mean())

data\_test['Outlet\_Size']=data\_test['Outlet\_Size'].fillna(data\_test['Outlet\_Size'].mode()[0])

data\_test

| **Item\_Identifier** | **Item\_Weight** | **Item\_Fat\_Content** | **Item\_Visibility** | **Item\_Type** | **Item\_MRP** | **Outlet\_Identifier** | **Outlet\_Establishment\_Year** | **Outlet\_Size** | **Outlet\_Location\_Type** | **Outlet\_Type** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | FDW58 | 20.750000 | Low Fat | 0.007565 | Snack Foods | 107.8622 | OUT049 | 1999 | Medium | Tier 1 | Supermarket Type1 |
| **1** | FDW14 | 8.300000 | reg | 0.038428 | Dairy | 87.3198 | OUT017 | 2007 | Medium | Tier 2 | Supermarket Type1 |
| **2** | NCN55 | 14.600000 | Low Fat | 0.099575 | Others | 241.7538 | OUT010 | 1998 | Medium | Tier 3 | Grocery Store |
| **3** | FDQ58 | 7.315000 | Low Fat | 0.015388 | Snack Foods | 155.0340 | OUT017 | 2007 | Medium | Tier 2 | Supermarket Type1 |

data\_train.isnull().sum()

Item\_Identifier 0

Item\_Weight 0

Item\_Fat\_Content 0

Item\_Visibility 0

Item\_Type 0

Item\_MRP 0

Outlet\_Identifier 0

Outlet\_Establishment\_Year 0

Outlet\_Size 0

Outlet\_Location\_Type 0

Outlet\_Type 0

Item\_Outlet\_Sales 0

dtype: int64

data\_test.isnull().sum()

Item\_Identifier 0

Item\_Weight 0

Item\_Fat\_Content 0

Item\_Visibility 0

Item\_Type 0

Item\_MRP 0

Outlet\_Identifier 0

Outlet\_Establishment\_Year 0

Outlet\_Size 0

Outlet\_Location\_Type 0

Outlet\_Type 0

dtype: int64

data\_train.Item\_Fat\_Content.value\_counts()

Low Fat 5089

Regular 2889

LF 316

reg 117

low fat 112

Name: Item\_Fat\_Content, dtype: int64

#merging similar values

data\_train.Item\_Fat\_Content=data\_train.Item\_Fat\_Content.replace({'LF':'Low Fat','low fat':'Low Fat'})

data\_train.Item\_Fat\_Content=data\_train.Item\_Fat\_Content.replace({'reg':'Regular'})

data\_test.Item\_Fat\_Content=data\_test.Item\_Fat\_Content.replace({'LF':'Low Fat','low fat':'Low Fat'})

data\_test.Item\_Fat\_Content=data\_test.Item\_Fat\_Content.replace({'reg':'Regular'})

data\_train.Item\_Fat\_Content.value\_counts()

Low Fat 5517

Regular 3006

Name: Item\_Fat\_Content, dtype: int64

data\_train.Item\_Type.value\_counts()

Fruits and Vegetables 1232

Snack Foods 1200

Household 910

Frozen Foods 856

Dairy 682

Canned 649

Baking Goods 648

Health and Hygiene 520

Soft Drinks 445

Meat 425

Breads 251

Hard Drinks 214

Others 169

Starchy Foods 148

Breakfast 110

Seafood 64

Name: Item\_Type, dtype: int64

#grouping values based on Item\_Identifier

data\_train.Item\_Type\_Combined = data\_train.Item\_Identifier.apply(lambda x: x[0:2])

data\_train.Item\_Type\_Combined = data\_train.Item\_Type\_Combined.map({'FD':'Food',

'NC':'Non-Consumable',

'DR':'Drinks'})

data\_test.Item\_Type\_Combined = data\_test.Item\_Identifier.apply(lambda x: x[0:2])

data\_test.Item\_Type\_Combined = data\_test.Item\_Type\_Combined.map({'FD':'Food',

'NC':'Non-Consumable',

'DR':'Drinks'})

data\_train.Item\_Type\_Combined.value\_counts()

Food 6125

Non-Consumable 1599

Drinks 799

Name: Item\_Identifier, dtype: int64

data\_train.Outlet\_Size.value\_counts()

Medium 5203

Small 2388

High 932

Name: Outlet\_Size, dtype: int64

data\_train.Outlet\_Location\_Type.value\_counts()

Tier 3 3350

Tier 2 2785

Tier 1 2388

Name: Outlet\_Location\_Type, dtype: int64

data\_train.Outlet\_Type.value\_counts()

Supermarket Type1 5577

Grocery Store 1083

Supermarket Type3 935

Supermarket Type2 928

Name: Outlet\_Type, dtype: int64

new\_trained\_data=pd.get\_dummies(data\_train,columns=['Item\_Fat\_Content','Item\_Type','Outlet\_Size','Outlet\_Location\_Type','Outlet\_Type'])

new\_tested\_data=pd.get\_dummies(data\_test,columns=['Item\_Fat\_Content','Item\_Type','Outlet\_Size','Outlet\_Location\_Type','Outlet\_Type'])

new\_trained\_data

A picture containing table

Description automatically generated

#droping unusual data columns

new\_trained\_data.drop(['Item\_Identifier','Outlet\_Identifier','Outlet\_Establishment\_Year'],axis=1,inplace=True)

new\_tested\_data.drop(['Item\_Identifier','Outlet\_Identifier','Outlet\_Establishment\_Year'],axis=1,inplace=True)

new\_trained\_data

Graphical user interface

Description automatically generated

X=new\_trained\_data.drop(['Item\_Outlet\_Sales'],axis=1)

y=new\_trained\_data['Item\_Outlet\_Sales']

A=new\_tested\_data

A

Table

Description automatically generated

# linear Regression

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X,y)

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=1, normalize=False)

B\_pred=regressor.predict(A)

B\_pred

array([1773.94170445, 1476.2854121 , 1876.85043013, ..., 1942.57186375,

3531.78127405, 1404.25629418])

from sklearn.metrics import accuracy\_score,r2\_score,mean\_squared\_error

accuracy=round(regressor.score(X,y) \* 100,3)

accuracy

56.266

# DECISION\_TREE\_ALORITHM

# from sklearn.tree import DecisionTreeRegressor

# DT=DecisionTreeRegressor(max\_depth=15,min\_samples\_leaf=300)

# DT.fit(X,y)

DecisionTreeRegressor(criterion='mse', max\_depth=15, max\_features=None,

max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, min\_samples\_leaf=300,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

presort=False, random\_state=None, splitter='best')

B\_pred=regressor.predict(A)

B\_pred

array([1773.94170445, 1476.2854121 , 1876.85043013, ..., 1942.57186375,

3531.78127405, 1404.25629418])

#measuring Accuracy

from sklearn.metrics import accuracy\_score, r2\_score, mean\_squared\_error

tree\_accuracy=round(regressor.score(X,y)\*100,3)

tree\_accuracy

56.266

# RANDOM\_FOREST\_ALGORITHM

from sklearn.ensemble import RandomForestRegressor

regressor=RandomForestRegressor(n\_estimators=100,max\_depth=10,min\_samples\_leaf=50,n\_jobs=4)

regressor.fit(X,y)

RandomForestRegressor(bootstrap=True, criterion='mse', max\_depth=10,

max\_features='auto', max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=50, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=100, n\_jobs=4,

oob\_score=False, random\_state=None, verbose=0, warm\_start=False)

B\_pred=regressor.predict(A)

B\_pred

array([1578.15541105, 1368.99594012, 593.15201242, ..., 1816.28452195,

3601.81114234, 1336.98061369])

rf\_accuracy=round(regressor.score(X,y)\*100,3)

rf\_accuracy

62.616

# KNN ALGORITHM

from sklearn.neighbors import KNeighborsRegressor

for K in range(10):

K=K+1

regressor=KNeighborsRegressor(n\_neighbors=K)

regressor.fit(X,y)

B\_pred=regressor.predict(A)

KNN\_accuracy=round(regressor.score(X,y)\*100,3)

print(KNN\_accuracy)

100.0

76.778

68.633

64.363

61.99

59.766

58.553

57.372

56.361

55.351

CHAPTER 5

RESULTS:

The Final Accuaracies and the predicted training sets by each Algorithm are given below:

**Linear Regression:**

**Graphical user interface, text, application, email

Description automatically generated**

**Decision Tree Algorithm:**

**Text, application

Description automatically generated**

**Random Forest Algorithm:**

**Text

Description automatically generated**

**KNN Algorithm:**

**Graphical user interface, text, application

Description automatically generated**

CHAPTER 6

CONCLUSION

The objective of this framework is to predict the future sales from given data of the previous year's using machine Learning techniques. In this paper, we have discussed how different machine learning models are built using different algorithms like Linear regression, Random forest regressor, Decision tree classification and KNN algorithms. These algorithms have been applied to predict the final result of sales. We have addressed in detail about how the noisy data is been removed and the algorithms used to predict the result. Based on the accuracy predicted by different models we conclude that the random forest approach and XG Booster approach are best models. Our predictions help big marts to refine their methodologies and strategies which in turn helps them to increase their profit.

FUTURE WORK

* Multiple instances parameters and various factors can be used to make this sales prediction more innovative and successful.
* Accuracy, which plays a key role in prediction-based systems, can be significantly increased as the number of parameters used are increased.
* The project can be further collaborated in a web-based application with an in-built intelligence by virtue of Internet of Things (IoT), to be more feasible for use.
* When combined with effective data mining methods and properties, the traditional means could be seen to make a higher and positive effect on the overall development of corporation’s tasks.

## REFERENCES

[1]. Beheshti-Kashi, S., Karimi, H.R., Thoben, K.D., L¨utjen, M., Teucke, M.: A survey on retail sales forecasting and prediction in fashion markets. Systems Science & Control Engineering 3(1), 154–161 (2015)

[2]. Bose, I., Mahapatra, R.K.: Business data mininga machine learning perspective. Information & management 39(3), 211–225 (2001)

[3]. Chu, C.W., Zhang, G.P.: A comparative study of linear and nonlinear models for aggregate retail sales forecasting. International Journal of production economics 86(3), 217–231 (2003)

[4]. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combing content-based and collaborative filters in an online newspaper (1999)

[5]. Das, P., Chaudhury, S.: Prediction of retail sales of footwear using feedforward and recurrent neural networks. Neural Computing and Applications 16(4-5), 491–502 (2007)

[6]. Domingos, P.M.: A few useful things to know about machine learning. Commun. acm 55(10), 78–87 (2012)

[7]. Langley, P., Simon, H.A.: Applications of machine learning and rule induction. Communications of the ACM 38(11), 54–64 (1995)

[8]. Loh, W.Y.: Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(1), 14–23 (2011)

[9]. Makridakis, S., Wheelwright, S.C., Hyndman, R.J.: Forecasting methods and applications. John wiley & sons (2008)

[10]. Ni, Y., Fan, F.: A two-stage dynamic sales forecasting model for the fashion retail. Expert Systems with Applications 38(3), 1529–1536 (2011)

[11]. Punam, K., Pamula, R., Jain, P.K.: A two-level statistical model for big mart sales prediction. In: 2018 International Conference on Computing, Power and Communication Technologies (GUCON). pp. 617–620. IEEE (2018)

[12]. Ribeiro, A., Seruca, I., Dur˜ao, N.: Improving organizational decision support: Detection of outliers and sales prediction for a pharmaceutical distribution company. Procedia Computer Science 121, 282–290 (2017)

[13]. Shrivas, T.: Big mart dataset@ONLINE (Jun 2013), https://datahack. analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/

[14]. Smola, A.J., Sch¨olkopf, B.: A tutorial on support vector regression. Statistics and computing 14(3), 199–222 (2004) 12 Gopal Behera and Neeta Nain

[15]. Smyth, B., Cotter, P.: Personalized electronic program guides for digital tv. Ai Magazine 22(2), 89–89 (2001)

[16]. Wang, Y., Witten, I.H.: Induction of model trees for predicting continuous classes (1996)

[17]. Xia, M., Wong, W.K.: A seasonal discrete grey forecasting model for fashion retailing. Knowledge-Based Systems 57, 119–126 (2014)