RESFINO

RETAIL SALES FORECAST AND INVENTORY OPTIMIZATION

A Project Report Submitted
in Partial Fulfilment of the Requirements
for the Degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

by

HARSH KUMAR SINGH
(Roll No. 2015BCS0012)

to

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
KOTTAYAM - 686635, INDIA
April 2019

DECLARATION

I, **HARSH KUMAR SINGH** (**Roll No: 2015BCS0012**), hereby declare that, this report entitled **"RETAIL SALES FORECAST AND INVENTORY OPTIMIZATION"** submitted to Indian Institute of Information Technology Kottayam towards partial requirement of **Bachelor of Technology(Hon)** in **COMPUTER SCIENCE AND ENGINEERING** is an original work carried out by me under the supervision of **Dr.EBIN DENI RAJ** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Valavoor - 686 635 **HARSH KUMAR SINGH**

APRIL 2019

CERTIFICATE

This is to certify that the work contained in this project report entitled **"RETAIL SALES FORECAST AND INVENTORY OPTIMIZATION"**submitted by **HARSH KUMAR SINGH** (**Roll No: 2015BCS0012**) to Indian Institute of Information Technology Kottayam towards partial requirement of **Bachelor of Technology(Hon)** in **INDIAN INSTITUTE OF INFORMATION TECHNOLOGY KOTTAYAM** has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Valavoor - 686 635	Dr. EBIN DENI RAJ
April 2019	Project Supervisor

ABSTRACT

BigMart is a big supermarket chain, with stores all around the country.BigMart has collected sales data from the year 2013, for 1559 products across 10 stores in different cities.With this information the corporation hopes we can identify the products and stores which play a key role in their sales and use that information to take the correct measures to ensure success of their business.

The main objective is to understand whether specific properties of products and stores play a significant role in terms of increasing or decreasing sales volume. With the help of computing power and data handling methods, there comes the possibility of automating tasks that are not necessarily handled by humans. To achieve this goal, we will build a predictive model and find out the sales of each product at a particular store. This will help Big-mart to boost their sales by learning optimized product organization inside stores and finally we will make software to automate the inventory allocation so that sales can be maximized and outlet inventory is optimized.

Contents

List of Figures

Chapter 1

Introduction

Along with the emergence of digitalization, internet sales became one of the essential business methods for almost every corporation. So we have decided to make a commercial product for company which have bunch of outlets and they are supposed to allocate their stock to these bunch of outlets so that their profit is maximized and they have no residual stock left in the stock which is not getting sold.

Hence, Our main objective is to understand whether specific properties of products and stores play a significant role in terms of increasing or decreasing sales volume.

This can be achieved with the help of computing power and data handling methods, and there comes the possibility of automating tasks that are not necessarily handled by humans.

To achieve this goal, we will build a predictive model and find out the sales of each product at a particular store. This will help Big-mart to boost their sales by learning optimized product organization inside stores and finally we will make a software to automate the inventory allocation so that sales can be maximized and outlet inventory is optimized.

1.1 Importance of Machine Learning

Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition but today computers could learn from data and it can even achieve those goals which humans can't achieve manually.

The machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. Machine learning gives ability to automatically apply complex mathematical calculations to big data to analyze the data to come up with something useful.

1.2 Existing model in Market

Since sales prediction was always the one of the hottest topic of machine learning to explore, there are many solutions available in the market but nearly every solution is about predicting the sales of the product in the outlet.

But I am going to make a inventory automation software for the sales department of the Mart which is the extension of the solutions available in the market.

1.3 Motivation for this project

Sales prediction and stock price predictions are 2 most controversial problems that can be attacked by the machine learning approach. Since my previous project was related with finance i.e. Credit card fraud transaction detection so this time we are going to extend the shopping tasks but not for customers but for businessman having large pool of shops.

BigMart has collected sales data from the year 2013, for 1559 products across 10 stores in different cities. With this information the corporation hopes we can identify the products and stores which play a key role in their sales and use that information to take the correct measures to ensure success of their business.

1.4 Chapter Description

- **CHAPTER 2**: Preliminary Work

  In this chapter preliminary work that is done in phase-I of the project is discussed. Previously used algorithm, architecture, proposed algorithm, performance, and result are briefly discussed in this section.

- **CHAPTER 3:** Literature Survey

  In this chapter the existing and established theory and research is discussed. This segment gives context for our work. This area is used for filling an apparent hole in the current hypothesis.

- **CHAPTER 4:** Proposed Architecture

  In this chapter proposed architecture, terminologies and definitions regarding our projects are thoroughly discussed.

- **CHAPTER 5:** Implementation

  In this chapter implementations details with the data set features and attributes and the tools and libraries included in the project are thoroughly discussed.

- **CHAPTER 6:** Result and Discussion

  In this chapter whole summary of the project is explained as the result and brief discussion on it.

Chapter 2

Preliminary Work

The upgradation in the Information Technology slowly converges a portion of our every day life into an electronical form.The global nature of Internet along with its anonymous behaviour make it an decent instrument for committing a fraud, which results huge financial losses each year. Although prevention is the best way to reduce fraud. Methodologies for the detection of fraud are therefore essential if we want to catch fraudsters once prevention has failed. With the help of computing power and data handling methods, there comes the possibility of automating tasks that are not necessarily handled by humans.

Credit card industry is a very tempting target for these individuals and hence this causes huge financial losses every year. Credit card fraud detection is a heavily studied branch of anomaly detection.The goal is to detect the maximum number of fraudulent transactions possible, in less computation time, with reasonable number of false positives ,i.e.,transactions that are actually genuine but computed as fraud. Because of the confidentiality issues, one of the most restraining factor for this research on financial fraud detection is the lack of publicly available data sets.The data set used in this thesis contains features that have been transformed by PCA (Principal component analysis) in order to anonymize all card transactions.Thus basic approach like feature aggregation cannot be used here.

In this chapter preliminary work that is done in phase-I of the project is discussed. Previously used algorithm, architecture, proposed algorithm, performance, and result are briefly discussed in this section.

2.1 Algorithm Used

2.1.1 Logistic Regression Algorithm

Its a classification algorithm, that is used where the response variable is categorical. The idea of Logistic Regression is to find a relationship between features and probability of particular outcome.Logistic regression predicts the probability of an outcome that can only have two values.A linear regression is not appropriate for predicting the value of a binary variable because A linear regression will predict values outside the acceptable range (e.g. predicting probabilities outside the range 0 to 1) and the dichotomous experiments can only have one of two possible values for each experiment, the residuals will not be normally distributed about the predicted line.On the other hand, a logistic regression produces a logistic curve, which is limited to values between 0 and 1. Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the odds of the target variable, rather than the probability. Moreover, the predictors do not have to be normally distributed or have equal variance in each group.

Logistic Regression is an clustering algorithm hence it have some limitations like it doesn't handle large number of categorical features/variables well.

Advantage of Logistic Regression is simple and efficient way to classify the data sets. The main advantage of using the Logistic Regression is it provides the probability score for the observations and hence it can be clubbed together with the other algorithm which gives the probability score.

2.1.2 Naive Bayes

Nave Bayes classifier can be extremely fast relative to other classification algorithms. It is based on Bayes theorem of probability to predict the class of unknown data set.

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors.In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

But it depends on the type of data set we will use, .i.e it is possible that we will get bad result as compared to other algorithms for some data sets with Nave Bayes classifier.If categorical variable has a category in test data set, which was not observed in training data set, then model will assign 0 probability and will be unable to make a prediction. This is often known as Zero Frequency. Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Since Nave Bayes classifier is based on classifying every features into independent pairs but it have many limitations like if categorical variable has a category in test data set, which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as Zero Frequency issue.Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent. Also Naive Bayes is also known as a bad estimator, so the probability outputs from probability are not to be taken too seriously and the data set that we are using is very crucial data set connected with the finance and crime hence we can't be totally reliable on the probability classifier.

Advantage of using Naive Bayes for our data set is its super simple nature,you're just doing a bunch of counts. If the Naive Bayes conditional independence assumption actually holds, a Naive Bayes classifier will converge quicker than other models if we use logistic regression, so you need less training data.

2.1.3 K- Nearest Neighbours Algorithm

It is a Supervised learning technique. It is used mostly for Classification, and sometimes even for Regression.'K in KNN is the number of nearest neighbours used to classify a test sample.K-NN doesn't have a training phase as such. But the prediction of a test observation is done based on the K-Nearest (often euclidean distance) Neighbours based on weighted averages/votes.

Since k-Nearest Neighbours is an clustering algorithm hence it have many limitations like we need to set the value of the parameter k (number of nearest neighbour) this different values of k will effect the accuracy of the algorithm.It's a distance based learning algorithm hence it is not clear which type of distance to use and which attribute to use to obtain the best results.

We have very large data set hence the computation cost is quite high because we need to compute the distance of each query instance to all training samples.Imbalanced data causes problems hence k-NN doesn't perform well on imbalanced data. If we consider two classes, A and B, and the majority of the training data is labeled as A, then the model will ultimately give a lot of preference to A. This might result in getting the less common class B wrongly classified.

Advantage of KNN for our data set is it can quickly respond to changes in input. kNN employs lazy learning, which generalizes during testing, this allows it to change during real-time use. and this algorithm is simple to implement and give high accuracy.

2.1.4 Support Vector Machine

Support Vector Machine is a supervised machine learning algorithm which can be used for both classification or regression challenges.However,it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well. Support Vector Machines (SVM) seek such a hyper plane that maximizes its distance between classified data points.Its training speed grows rapidly with the number of training samples and therefore it is not suitable for large amount of data. There are many cons of using SVM such as, it doesn't perform well when we have large data set because the required training time is much higher.It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping.

Since SVM is about finding the hyper-plane that differentiate the two classes very well hence it have many limitations such as Support Vector Machine Algorithm is not suited to larger data sets as the training time with SVMs can be very much high as compared to other algorithms.If we have noisier data sets with overlapping classes then effectiveness of the algorithm decreases.For our data set which is too large it will take much time as compared to other algorithms and there are noises present in data set hence this is the concern part of this algorithm.

Advantage of SVM for our data set is its high accuracy, it gives more accuracy than Naive Nayes.It is effective in high dimensional spaces since our data set have high dimensions.It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

2.2 Architecture

As we have discussed earlier best model will be that model which will cover every aspect of the data set hence the combination of all the algorithm will be the best model to classify the data sets into fraud and non-fraud transactions as shown in Fig 2.1.

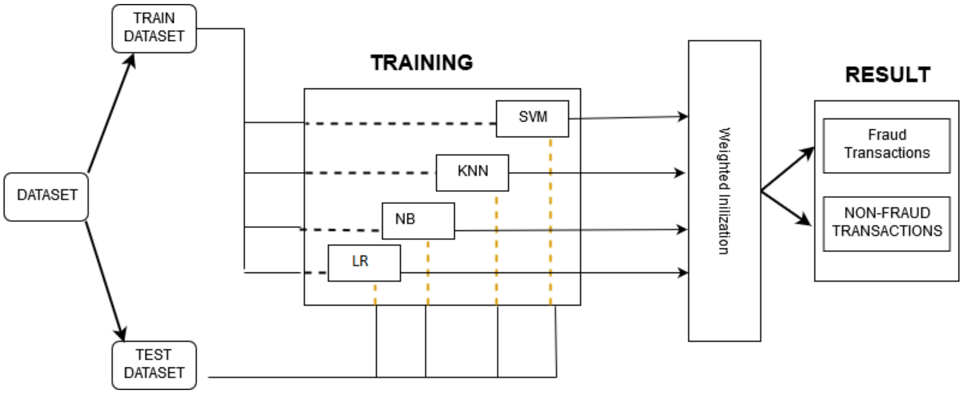


Figure 2.1: Block diagram of Proposed Algorithm

2.3 Proposed Algorithm

In this section we had constructed an algorithm for our proposed model.

Step 1: Load the Dataset

Step 2: Using the Logistic Regression to classify the dataset into the fraud and genuine transactions and take the probability score entries which is an vector and named it as 'a'.

Step 3: Using the Naive Bayes and logistic regression classify the dataset into the fraud and genuine transactions and take the probability score entries which is an vector and named it as 'b'.

Step 4: Using the k-nearest neighbors algorithm to classify the dataset into the fraud and genuine transactions and take the probability score entries which is an vector and named it as 'c'.

Step 5: Using the Support Vector Machine algorithm to classify the dataset into the fraud and genuine transactions and take the probability score entries which is an vector and named it as 'd'.

Step 6: Let the coefficients of LA , NB, KNN, SVM algorithms be CL, CN, CK, CS respectively.Now using these weights coefficients make another class as vector 'e' which is summation of product of coefficient and algorithms class result value .i.e e = CL*a + CN*b + CK*c + CS*d

Step 7: Now we have created another vector of same dimension of the datasets with same respective entries now and it contains the probability value of fraud and genuine for each transaction.

Step 8: Now with these probability values we will classify the dataset entries the first column is 'i' and second column is 'j' and if i¿j it means fraud transaction and substitute it with value 1 and vice versa .

Step 9: Now the got a new vector 'f' of the same dimension of e which contains all entries and these entries will tell whether the transaction is Fraud or Genuine.

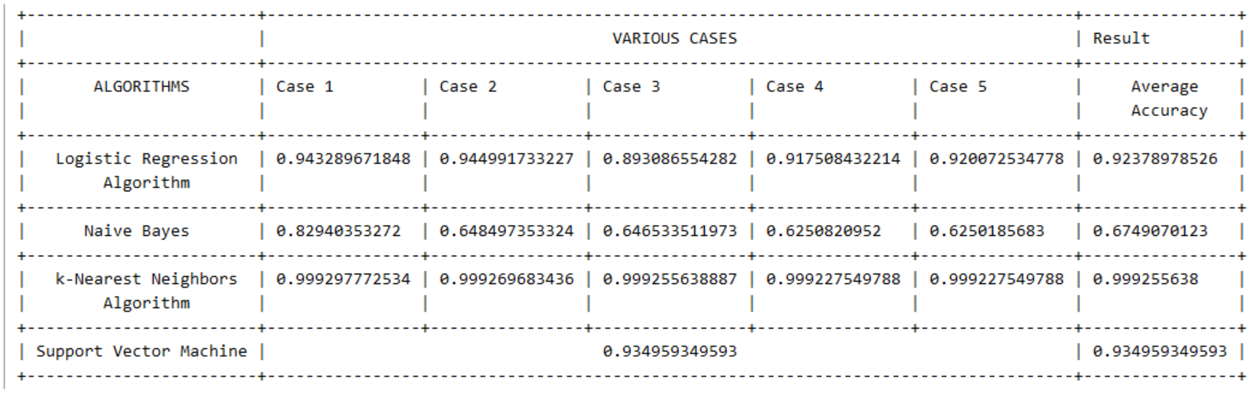
2.4 Performance Analysis of different Algorithm separately

Following Fig 2.2 is a table which contains the performance or accuracy of different algorithm in the given dataset.

For the following observations we have observe nearly 5 different cases for each algorithm each.

- In Logistic Regression we took 5 different situations where we took 5 different C parameter values as 0.001, 0.01, 0.1, 1, 10 respectively.

- In Naive Bayes we also took 5 different cases. So to begin I will initially add all the variables to the model and remove one by one variable and see removing which ones increases the accuracy the most.For case 1 we dropped class ,log amount, time. For case 2 we dropped class ,log amount, V1. For case 3 we dropped class ,log amount, V2. For case 4 we dropped class ,log amount, V3. For case 5 we dropped class ,log amount, V4.



```
+------------------------+------------------------------------------------------------------------------------+----------------+
|                        |                                    VARIOUS CASES                                   | Result         |
+------------------------+------------------------------------------------------------------------------------+----------------+
|       ALGORITHMS       | Case 1         | Case 2         | Case 3         | Case 4         | Case 5         |    Average     |
|                        |                |                |                |                |                |    Accuracy    |
+------------------------+----------------+----------------+----------------+----------------+----------------+----------------+
|   Logistic Regression  | 0.943289671848 | 0.944991733227 | 0.893086554282 | 0.917508432214 | 0.920072534778 | 0.92378978526  |
|        Algorithm       |                |                |                |                |                |                |
+------------------------+----------------+----------------+----------------+----------------+----------------+----------------+
|       Naive Bayes      | 0.82940353272  | 0.648497353324 | 0.646533511973 | 0.6250820952   | 0.6250185683   | 0.6749070123   |
+------------------------+----------------+----------------+----------------+----------------+----------------+----------------+
|   k-Nearest Neighbors  | 0.999297772534 | 0.999269683436 | 0.999255638887 | 0.999227549788 | 0.999227549788 | 0.999255638    |
|        Algorithm       |                |                |                |                |                |                |
+------------------------+----------------+----------------+----------------+----------------+----------------+----------------+
| Support Vector Machine |                                   0.934959349593                                   | 0.934959349593 |
+------------------------+------------------------------------------------------------------------------------+----------------+
```

Figure 2.2: Performance table of various algorithms in given dataset

- In KNN classifier we also took 5 different cases in which the number of neighbours was the distinguishing factor.So we took five different cases with number of neighbours 5,6,7,8,9 respectively.

- For the Support Vector Machine we took only one case in which the whole dataset is used to train the model.

After obtaining the different result for the various algorithm we calculated the average accuracy score of each algorithm separately so that we can use that result for the purpose of weight initialization.

2.5 Weight initialization

Since we got all the accuracy result of various algorithm, so while observing these result we found some inference like the performance of every algorithm is different hence if we want to build an new model which contains all the above algorithms then we should have to give the different weightage to different algorithms result,hence we need to find the different coefficients values for all the algorithms.
For finding the weighted coefficients we compare the accuracy values with each other and came to the conclusion that coefficients are :

A = Average accuracy of Logistic Regression Algorithm
B = Average accuracy of Naive Bayes
C = Average accuracy of k-Nearest Neighbors Algorithm
D = Average accuracy of Support Vector Machine
hence,

-

Coefficient of Logistic Regression Algorithm = A/(A+B+C+D) = 0.26148113552

Coefficient of Naive Bayes and Logistic Regression = B/(A+B+C+D) = 0.191034210122

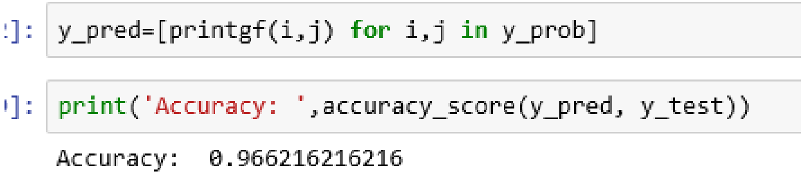
Coefficient of k-Nearest Neighbors Algorithm = C/(A+B+C+D) = 0.2828419442

Coefficient of Support Vector Machine = D/(A+B+C+D) = 0.264642710167

2.6 Performance of new Algorithm

Since we got all the coefficients value for all algorithm, we can easily classify the dataset into frauds and non fraud transactions.Performance of this algorithm is increased a bit since it contains all the various different aspects of the datasets.

The accuracy of new algorithm for our dataset is 96.6216216216 percent which is way better than the Naive Bayes and it have more precision as compared to other all algorithm. Here in Fig 2.3 y_pred is prediction score.



```
]: y_pred=[printgf(i,j) for i,j in y_prob]

]: print('Accuracy: ',accuracy_score(y_pred, y_test))

   Accuracy:  0.966216216216
```

Figure 2.3: Performance of the Proposed Algorithm

2.7 Inference from preliminary work and future work

We have researched related work and decided to use the hybrid approach .i.e. combination of various algorithms because of its advantages over other anomaly detection methods.

Our Fraud detection algorithm architecture was presented along with its analysis. We described each component of the detection system and explained the parameter choices for Proposed model and it has dramatically improved our classifier performance.

The previous work was about the finance where it was helping the vendor to catch the fraudulent and decrease the financial loss but in future we can make a application related with finance which will increase the sales profit by automating the inventory allocation process of the Big-Mart inventory into its different outlets inventory for maximizing the profit for the successful business.

Chapter 3

Literature Survey

In this chapter the existing and established theory and research is discussed. This segment gives context for our work. This area is used for filling an apparent hole in the current hypothesis.

[4] **A*S Tomar, M Singh, G Sharma, KV Arya*** in their paper " **Traffic Management using Logistic Regression with Fuzzy Logic** " have proposed a real time traffic information for intelligent decision making to decide the route preference is required. Some certain parameters such as distance, weather condition, road location, day of week and time are considered to formulate the problem and to find solutions to these problems. This paper outlines a combination of logistic regression with fuzzy logic such that a smart decision to preferred path can be taken. Traffic management problem can be seems as like an inventory management. Somewhere it is more crowded and somewhere its less crowded. Now we have to manage it so that crowd can be optimized.

[2] ***Aly Megahed, Peifeng Yin, Hamid Reza Motahari Nezhad***

in their paper "**An Optimization Approach to Services Sales Forecasting in a Multi-staged Sales Pipeline** " have formulated this problem, considering the service-specific context, as a machine learning problem over the set of historical services sales data. They introduced a optimization approach for finding the optimized weights of a sales forecasting function. They evaluated the presented method, with the existing method without weights then they come to conclusion that their method was having superior performance (in terms of absolute and relative errors) compared to basic method.

[5] ***Fernando Jimnez, Gracia Snchez, Jos M. Garca, Guido Sciavicco, Luis Miralles*** in their paper "**Multi-Objective Evolutionary Feature Selection for Online Sales Forecasting**" have made a regression model for online sales forecasting obtained via a novel feature selection methodology by the application of the multi-objective evolutionary algorithm ENORA (Evolutionary non-dominated Radial slots-based Algorithm). Then they have also used Random Forest algorithm. They have integrated feature selection for regression, model evaluation, and decision making, in order to choose the most satisfactory model.

[6] ***Indre liobaite, Jorn Bakker, Mykola Pechenizkiy*** in their paper "**Towards Context Aware Food Sales Prediction**" have worked on the context aware sales prediction approach, they have selected the base predictor depending on the structural properties of the historical sales. They have also shown how the dependencies between product categorization accuracies and sales prediction accuracies.

[7] ***Indu Kumar, Kiran Dogra, Chetna Utreja, Premlata Yadav***

in their paper "**A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction**" have worked on many machine learning techniques that have been applied for the stock price prediction in order to overcome difficulties. In the implemented work, five models have been developed and their performances are compared in predicting the stock market trends. These models are based on five supervised learning techniques i.e., Support Vector Machine (SVM), Random Forest, K-Nearest Neighbour (KNN), Naive Bayes. The experimental results show that Random Forest algorithm performs the best for large datasets and Naive Bayesian Classifier is the best for small datasets.The results also reveal that reduction in the number of technical indicators reduces the accuracies of each algorithm.

[10] **M Hlsmann, D Borscheid, CM Friedrich, D Reith** in thei paper "**General Sales Forecast Model for Automobile Markets and their Analysis**" in this paper they have developed the forecast model for the US-American automobile market. In this paper, various enhanced sales forecast methodologies and models for the automobile market are presented.The methodology mainly consists of time series analysis and classical Data Mining algorithms, whereas the data is composed of absolute market-specific parameters.It can be concluded that the monthly forecasts were especially improved by this enhanced methodology using absolute, normalized parameters.

[14] **Wenyi Huang, Xiao Qin, Hongyan Dai** "**Sales Forecast for O2O Services - Based on Incremental Random Forest Method**" in this paper they have propose an incremental random forest method to forecast the sales for the O2O (online to offline) take-out business.The proposed

method has two characteristics. First, they identify the important features that contribute most to the forecast accuracy by deleting the noisy features. This feature selection process helps to improve the forecast accuracy. Second, they used an incremental method based on random forest by adding incremental features and focus on sales increment prediction. This incremental random forest method could further help to control the forecast error.

Chapter 4

Proposed Architecture

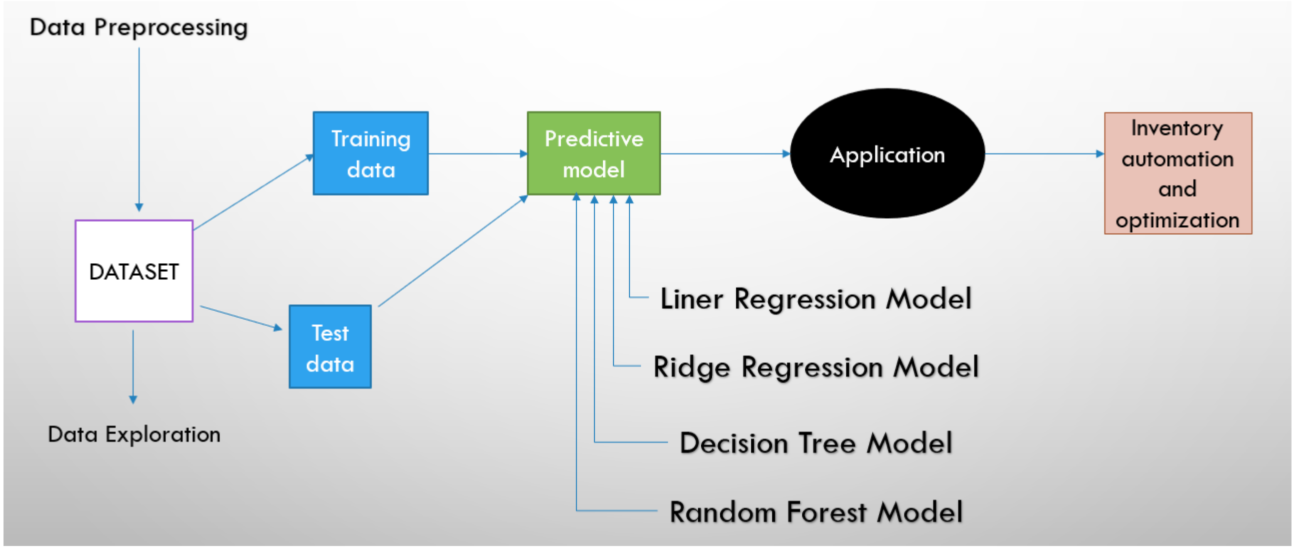


Figure 4.1: Architecture of the System

As in the diagram Fig 4.1, we have done Data prepossessing in the starting with the project on the inconsistent data set to make it more sensible for our analysis.

Then we have done the Exploratory Data Analysis (EDA) or Data Exploration with the feature engineering methods on the data set and after that we have exported our data into train and test CSV file to train our model.

After exporting the CSV files we have created four machine learning models(i.e. Linear Regression Model, Ridge regression Model, Decision Tree model, Random Forest Model) for predicting the sales, then after created those model we have developed an application which will take input from the used (2 inputs are Item_ID, Quantity) and it will automatically allocate those inventory items into those outlets so that we will get maximum sales result.

4.1 Definitions

In this section we will discuss about the various terminologies used in the project.

4.1.1 Hypothesis Generation

Understanding the problem better by brainstorming possible factors that can impact the outcome.

This is done before looking at the data, and we end up creating a laundry list of the different analysis which we can potentially perform if data is available.

4.1.2 Data Exploration

Looking at categorical and continuous feature summaries and making inferences about the data. Data exploration is an approach similar to initial data analysis, whereby a data analyst uses visual exploration to understand what is in a dataset and the characteristics of the data, rather than through traditional data management systems. These characteristics can include size or amount of data, completeness of the data, correctness of the data, possible relationships among data elements or files/tables in the data.

Data exploration is typically conducted using a combination of automated and manual activities.Automated activities can include data profiling or data visualization or tabular reports to give the analyst an initial view into the data and an understanding of key characteristics.

This is often followed by manual drill-down or filtering of the data to identify anomalies or patterns identified through the automated actions. Data exploration can also require manual scripting and queries into the data (e.g. using languages such as SQL or R) or using Excel or similar tools to view the raw data.

4.1.3 Data Cleaning

Data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores.

The process of Data Cleaning

1. **Parsing :** for the detection of syntax errors. A parser decides whether a string of data is acceptable within the allowed data specification. This is similar to the way a parser works with grammars and languages.

2. **Data transformation:**Data transformation allows the mapping of the data from its given format into the format expected by the appropriate application. This includes value conversions or translation functions, as well as normalizing numeric values to conform to minimum and maximum values.

3. **Duplicate elimination:**Duplicate detection requires an algorithm for determining whether data contains duplicate representations of the same entity. Usually, data is sorted by a key that would bring duplicate entries closer together for faster identification.

4. **Statistical methods:**By analyzing the data using the values of mean, standard deviation, range, or clustering algorithms, it is possible for an expert to find values that are unexpected and thus erroneous. Although the correction of such data is difficult since the true value is not known, it can be resolved by setting the values to an average or other statistical value.

4.1.4 Feature Engineering

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive. The quality and quantity of the features will have great influence on whether the model is good or not. You could say the better the features are, the better the result is.Here mostly we modify existing variables and create new ones for analysis.

The process of feature engineering:

1. Testing features
2. Deciding what features to create
3. Creating features
4. Checking how the features work with your model
5. Improving your features if needed
6. Go back to creating more features until the work is done.

4.1.5 Model Building

The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions.

Here we will train your model as per the data set and our model can predict the outcome for the other new data values that are not available in the data set that we have used to train the model.

Chapter 5

Implementation

In this chapter implementations details with the data set features and attributes and the tools and libraries included in the project are thoroughly discussed.

5.1 About the Data set

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.

Data reference :- https://www.kaggle.com/aakash2016/big-mart-sales-prediction

Using this model, BigMart will try to understand the properties of products and stores which play a key role in increasing sales. The data may have missing values as some stores might not report all the data due to technical glitches. Hence, it will be required to treat them accordingly.

5.1.1 Data Set Attributes

This data set contains 8523 observations and 12 features.The target variable is "Item_Outlet_Sales"

Now we will explain each and every attribute properties:

1. **Item_Identifier :** it is Unique product ID, Numeric, Discrete, it is expected that this attribute have low impact on the target sales value.

2. **Item_Weight :** it is Weight of product, Numeric, Continuous, it is expected that this attribute have medium impact on the target sales value.

3. **Item_Fat_Content :** it tells whether the product is low fat or not, Categorical, Ordinal, it is expected that this attribute have medium impact on the target sales value because low fat items are generally used more than others.

4. **Item_Visibility:** The percent % of total display area of all products in a store allocated to the particular product, Numeric, Continuous, it is expected that this attribute have high impact on the target sales value.

5. **Item_Type :** The category to which the product belongs, Categorical, Nominal, it is expected that this attribute have high impact on the target sales value.

6. **Item_MRP :** It is Maximum Retail Price (list price) of the product, Numeric, Discrete, it is expected that this attribute have medium impact on the target sales value.

7. **Outlet_Identifier :** It is a Unique store ID, Numeric, Discrete, it is expected that this attribute have low impact on the target sales value.

8. **Outlet_Establishment_Year :** The year in which store was established, Numeric, Discrete, it is expected that this attribute have low impact on the target sales value.

9. **Outlet_Size :** The size of the store in terms of ground area covered, Categorical, Ordinal, it is expected that this attribute have high impact on the target sales value.

10. **Outlet_Location_Type :** The type of city in which the store is located, Categorical, Ordinal, it is expected that this attribute have high impact on the target sales value.

11. **Outlet_Type :** Whether the outlet is just a grocery store or some sort of supermarket, Categorical, Ordinal, it is expected that this attribute have high impact on the target sales value as it tells about the stores capacity.

12. **Item_Outlet_Sales :** Sales of the product in the particular store. This is the outcome variable to be predicted, Numeric, Discrete, it is a Target variable.

5.1.2 Attributes Analysis

From this first look at the data, the variables that will have higher impact on the products sale price are: Item_Visibility , Item_Type ,Outlet_Size , Outlet_Location_Type , Outlet_Type . These attributes can have higher impact on sales than others because these are the categorical data and rest others are numerical.

The target variable is Item_Outlet_Sales.

If we look at variable Item_Identifier , we can see different group of letters per each product such as FD (Food), DR(Drinks) and NC (Non-Consumable).

Regarding Item_Visibility there are items with the value zero . This does not make lot of sense, since this is indicating those items are not visible on the store.

From the 12 features, 5 are numeric and 7 categorical.

5.1.3 Hypotheses about the data

Data may have missing values as some stores might not report all the data due to technical glitches. Hence, it will be required to treat them accordingly. So first Step is to develop hypotheses about the data.

Store Level Hypotheses:

1. **Store Capacity:** Stores which is having very big in size should have higher sales as they act like one-stop-shops and people would prefer getting everything from one place.

2. **Population Density:** Stores located in densely populated areas should have higher sales because of more demand.

3. **City type:** Stores located in urban or Tier 1 cities should have higher sales because of the higher income levels of people there.

4. **Competitors:** Stores having similar establishments nearby should have less sales because of more competition.

5. **Marketing:** Stores which have a good marketing division should have higher sales as it will be able to attract customers through the right offers and advertising.

6. **Location:** Stores located within popular marketplaces should have higher sales because of better access to customers.

7. **Customer Behavior:** Stores keeping the right set of products to meet the local needs of customers will have higher sales.

8. **Ambiance:** Stores which are well-maintained and managed by polite and humble people are expected to have higher footfall and thus higher sales.

Product Level Hypotheses:

1. **Packaging:** Products with good packaging can attract customers and sell more.

2. **Brand :** Branded products should have higher sales because of higher trust in the customer.

3. **Utility:** Daily use products should have a higher tendency to sell as compared to the specific use products.

4. **Display Area:** Products which are given bigger shelves in the store are likely to catch attention first and sell more.

5. **Visibility in Store:** The location of product in a store will impact sales. Ones which are right at entrance will catch the eye of customer first rather than the ones in back.

6. **Advertising:** Better advertising of products in the store will should higher sales in most cases.

7. **Promotional Offers:** Products accompanied with attractive offers and discounts will sell more.

5.2 Tools and Libraries

In this section tools and the libraries that is used in the project are discussed.

5.2.1 Tools

We have tools used several tools to make our project such as :

1. **The Jupyter Notebook :** The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

2. **Qt Designer :** Qt Designer is the Qt tool for designing and building graphical user interfaces. It allows you to design widgets, dialog or complete main windows using on-screen forms and a simple drag-and-drop interface. It has the ability to preview your designs to ensure they work as you intended, and to allow you to prototype them with your users, before you have to write any code.

5.2.2 Libraries

We have tools used several libraries to make our project such as :

1. **Pandas :** pandas is an open source, it is a library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

2. **NumPy :** NumPy is the fundamental package for scientific computing with Python. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases. It contains following features:

   - a powerful N-dimensional array object
   - sophisticated (broadcasting) functions
   - tools for integrating C/C++ and Fortran code
   - useful linear algebra, Fourier transform, and random number capabilities

3. **SciPy :** SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering.

Our goal as a Data Scientist is to identify the most important variables and to define the best regression model for predicting out target variable. Hence, this analysis will be divided into four stages:

1. Exploratory data analysis (EDA);
2. Data Pre-processing;
3. Feature engineering;
4. Modeling;

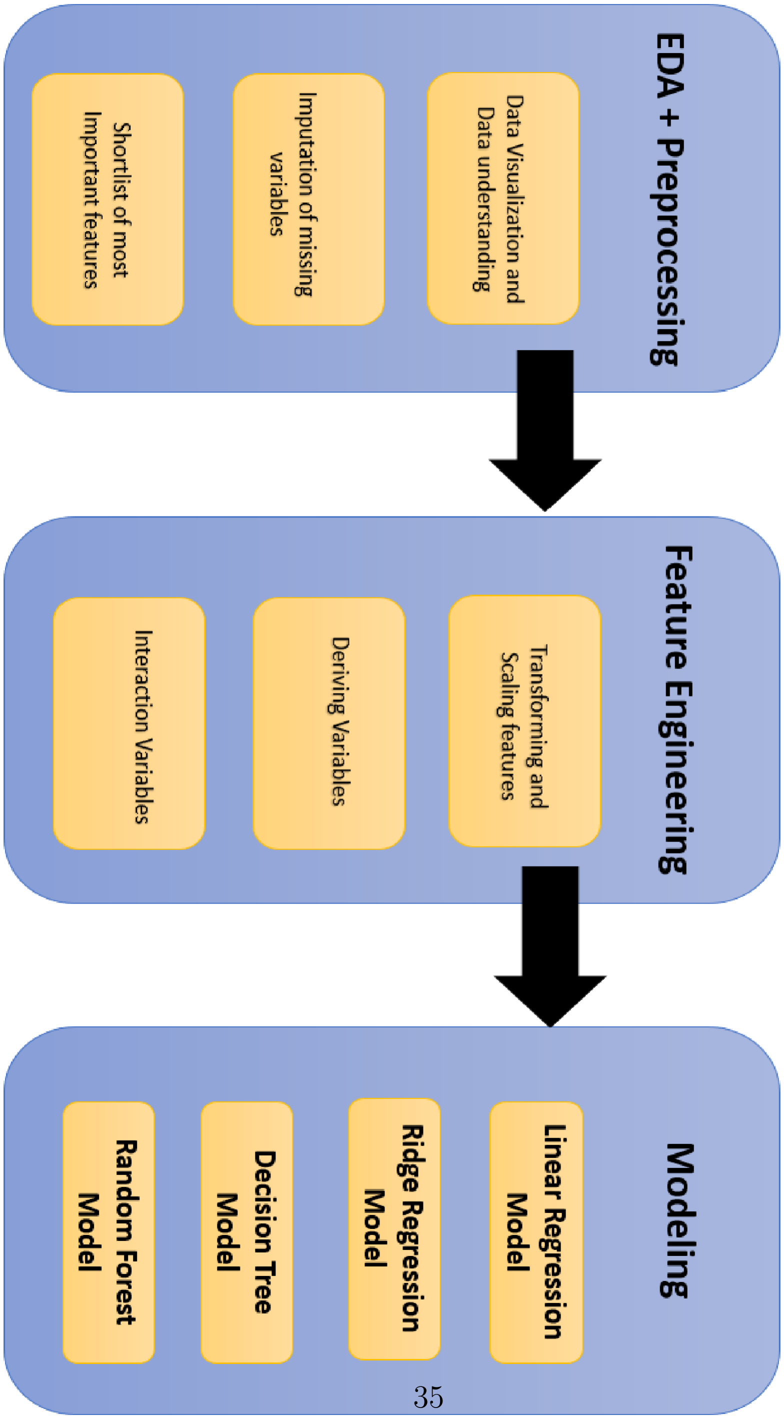


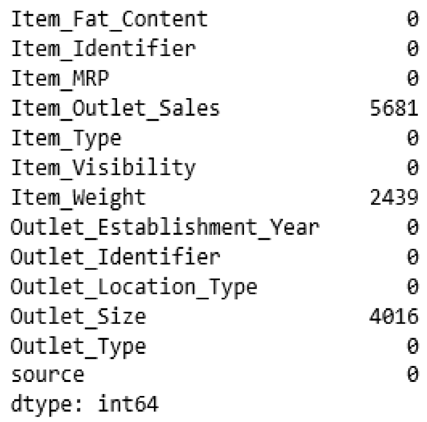
Figure 5.1: This is a workflow chart illustrating the four stages

5.3 Exploratory Data Analysis (EDA)

Weve made our first assumptions on the data and now we are ready to perform some basic data exploration and come up with some inference. Hence, the goal for this section is to take a glimpse on the data as well as any irregularities so that we can correct on the next section, Data Pre-Processing. Before starting the analysis it is interesting to check if the dataset suffers from duplicate values. In this case, the way to see if there are any duplicates is using the Item_Identifier feature. Since a product can exist in more than one store it is expected for this repetition to exist. Curious fact is that there seems to be 1562 unique items only available in one single store.

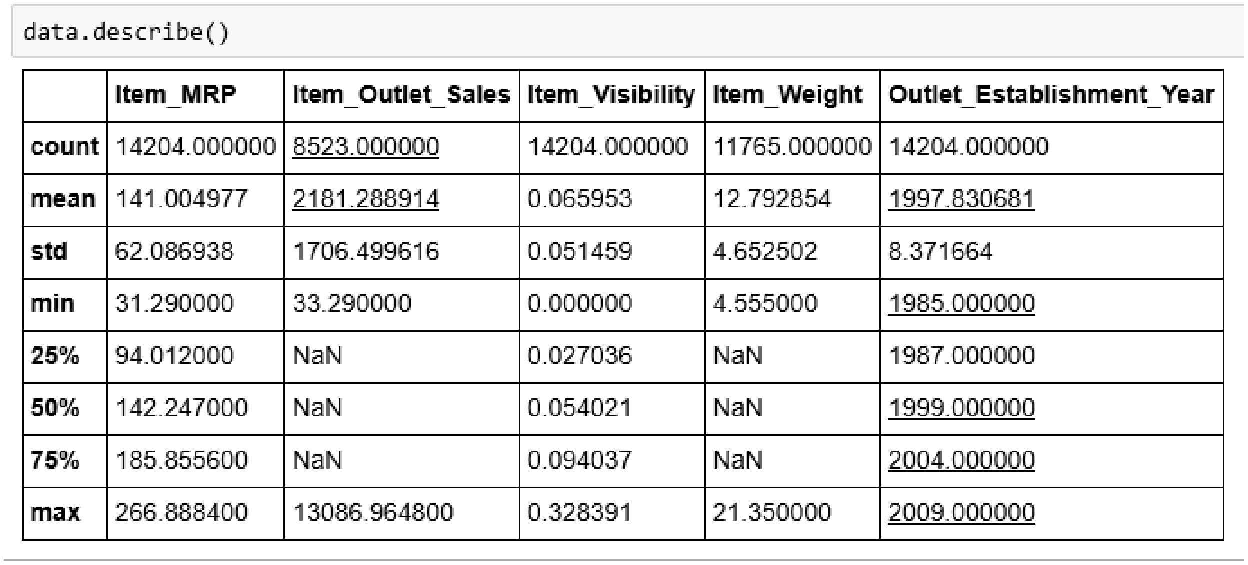
5.3.1 Missing Values analysis

One of the key challenges in any data set is missing values. Lets start by checking which columns contain missing values. So we can obtain the missing count and it is shown in Fig 5.2.



```
Item_Fat_Content                0
Item_Identifier                 0
Item_MRP                        0
Item_Outlet_Sales            5681
Item_Type                       0
Item_Visibility                 0
Item_Weight                  2439
Outlet_Establishment_Year       0
Outlet_Identifier               0
Outlet_Location_Type            0
Outlet_Size                  4016
Outlet_Type                     0
source                          0
dtype: int64
```

Figure 5.2: Number of missing value cells in the dataset



```
data.describe()
```

	Item_MRP	Item_Outlet_Sales	Item_Visibility	Item_Weight	Outlet_Establishment_Year
count	14204.000000	8523.000000	14204.000000	11765.000000	14204.000000
mean	141.004977	2181.288914	0.065953	12.792854	1997.830681
std	62.086938	1706.499616	0.051459	4.652502	8.371664
min	31.290000	33.290000	0.000000	4.555000	1985.000000
25%	94.012000	NaN	0.027036	NaN	1987.000000
50%	142.247000	NaN	0.054021	NaN	1999.000000
75%	185.855600	NaN	0.094037	NaN	2004.000000
max	266.888400	13086.964800	0.328391	21.350000	2009.000000

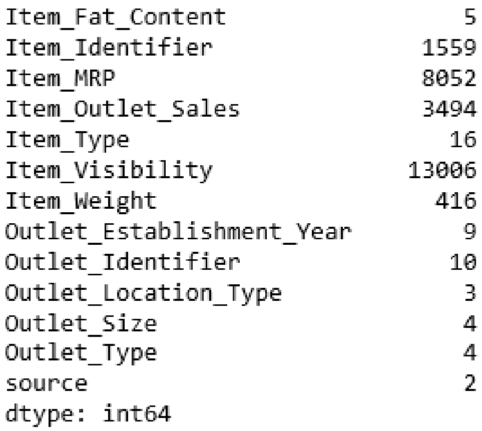
Figure 5.3: Some basic statistics for numerical variables

Note that the Item_Outlet_Sales is the target variable and missing values are ones in the test set. So we need not worry about it. But well impute the missing values in Item_Weight and Outlet_Size in the data cleaning section.

5.3.2 Statistics for numerical variables

Now we will look into some basic statistics for numerical variables. In Fig 5.3 we can see that count, mean, standard deviation, minimum, maximum, quantile, and many more mathematical features of the data.

Item_Visibility has a min value of zero. This makes no practical sense because when a product is being sold in a store, the visibility cannot be 0. Outlet_Establishment_Years vary from 1985 to 2009. The values might not be apt in this form. Rather, if we can convert them to how old the particular store is, it should have a better impact on sales.The lower count of Item_Weight and Item_Outlet_Sales confirms the findings from the missing value check.



```
Item_Fat_Content                 5
Item_Identifier               1559
Item_MRP                      8052
Item_Outlet_Sales             3494
Item_Type                       16
Item_Visibility              13006
Item_Weight                    416
Outlet_Establishment_Year        9
Outlet_Identifier               10
Outlet_Location_Type             3
Outlet_Size                      4
Outlet_Type                      4
source                           2
dtype: int64
```

Figure 5.4: Number of unique values

Now we Move to nominal (categorical) variable, lets have a look at the number of unique values in each of them in Fig 5.4.

This tells us that there are 1559 products and 10 outlets.Another thing that we can observe is that Item_Type has 16 unique values.

5.3.3 Frequency distribution

Now we will explore further using the frequency of different categories in each nominal variable. This following figures shows theses frequencies of the outlets attributes and products attributes.

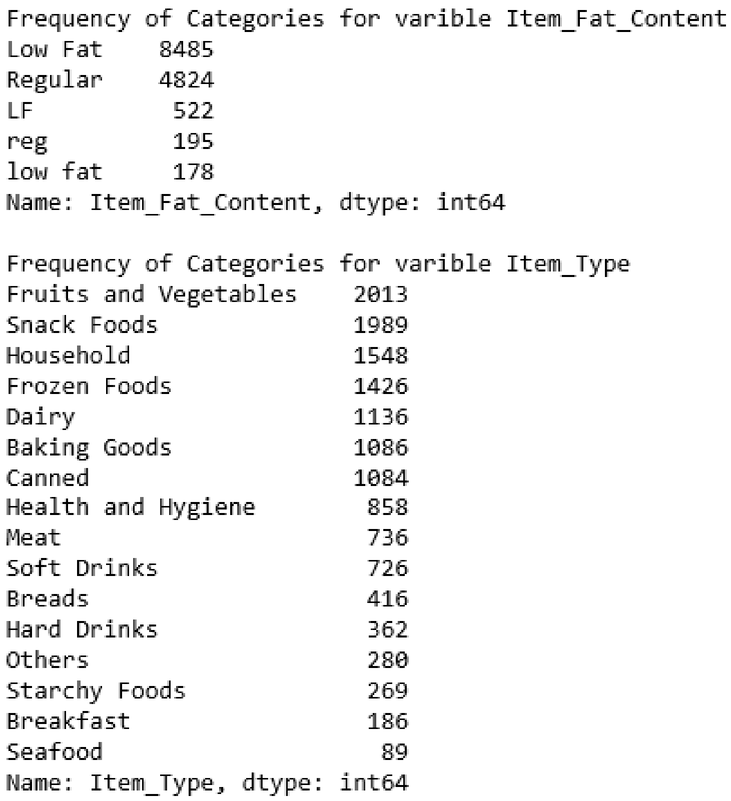
As we can see in the frequencies table Fig 5.5 that Item_Fat_Content there are Some Low Fat values mis-coded as low fat and LF and also, some of Regular are mentioned as regular. In Fig 5.5 in Item_Type not all categories have substantial numbers. It looks like combining them can give better results. In Fig 5.6 in Outlet_Type supermarket Type2 and Type3 can be combined. But we should check if thats a good idea before doing it.

5.4 Data Cleaning

This process involves imputing missing values and treating outliers. Though outlier removal is very important in regression techniques, advanced tree based algorithms are not allows outliers.

5.4.1 Imputing Missing Values

We found two variables with missing values Item_Weight and Outlet_Size. So we have imputed the former by the average weight of the particular item.

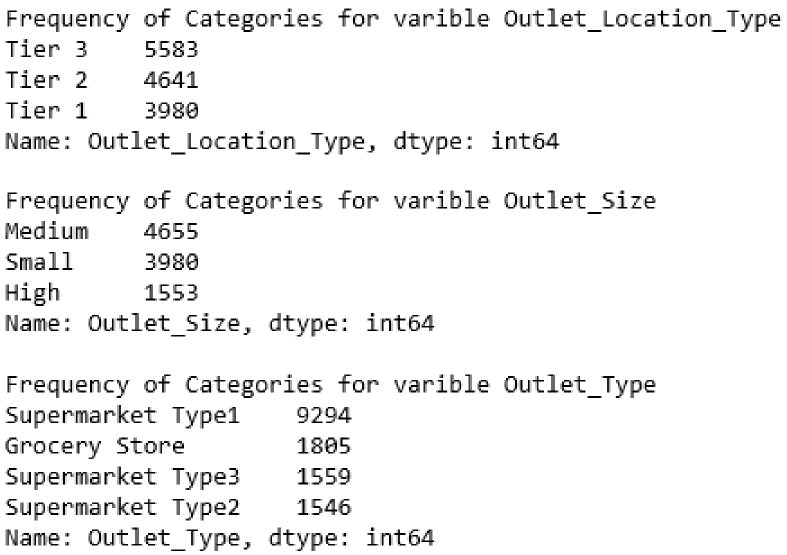


```
Frequency of Categories for varible Item_Fat_Content
Low Fat    8485
Regular    4824
LF          522
reg         195
low fat     178
Name: Item_Fat_Content, dtype: int64

Frequency of Categories for varible Item_Type
Fruits and Vegetables    2013
Snack Foods              1989
Household                1548
Frozen Foods             1426
Dairy                    1136
Baking Goods             1086
Canned                   1084
Health and Hygiene        858
Meat                      736
Soft Drinks               726
Breads                    416
Hard Drinks               362
Others                    280
Starchy Foods             269
Breakfast                 186
Seafood                    89
Name: Item_Type, dtype: int64
```

Figure 5.5: Frequencies of Item related attributes

The Fig 5.7 confirms that the column has no missing values now. Lets impute Outlet_Size with the mode of the Outlet_Size for the particular type of outlet. The Fig 5.8 confirms that there are no missing values in the data. So we will move on to feature engineering.

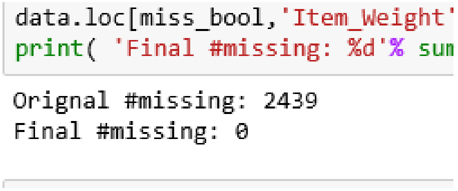


```
Frequency of Categories for varible Outlet_Location_Type
Tier 3    5583
Tier 2    4641
Tier 1    3980
Name: Outlet_Location_Type, dtype: int64

Frequency of Categories for varible Outlet_Size
Medium    4655
Small     3980
High      1553
Name: Outlet_Size, dtype: int64

Frequency of Categories for varible Outlet_Type
Supermarket Type1    9294
Grocery Store        1805
Supermarket Type3    1559
Supermarket Type2    1546
Name: Outlet_Type, dtype: int64
```

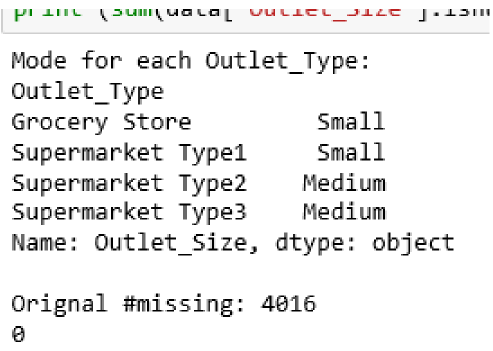
Figure 5.6: Frequencies of outlet related attributes



```
data.loc[miss_bool,'Item_Weight'
print( 'Final #missing: %d'% sum

Orignal #missing: 2439
Final #missing: 0
```

Figure 5.7: Count of new missing values in data set



```
print (sum(data['Outlet_Size'].isn

Mode for each Outlet_Type:
Outlet_Type
Grocery Store          Small
Supermarket Type1      Small
Supermarket Type2     Medium
Supermarket Type3     Medium
Name: Outlet_Size, dtype: object

Orignal #missing: 4016
0
```

Figure 5.8: Imputation through mode in Outlet_Size

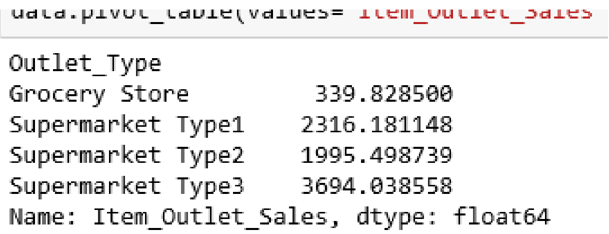
5.5 Feature Engineering

We have already explored some nuances in the data in the data exploration section. We have resolve them and make our data ready for the analysis in the feature engineering section. We have also create some new variables using the existing ones here.

5.5.1 Consider combining Outlet_Type

During exploration, we had decided to consider combining the Supermarket Type2 and Type3 variables, but we don't know exactly whether that is good idea or not.

To check whether its good or bad we will analyze mean sales by type of store.If they have similar sales, then keeping them separate wont help much and we can combine them.



```
data.pivot_table(values='Item_Outlet_Sales'
```

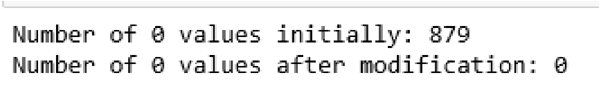
```
Outlet_Type
Grocery Store         339.828500
Supermarket Type1    2316.181148
Supermarket Type2    1995.498739
Supermarket Type3    3694.038558
Name: Item_Outlet_Sales, dtype: float64
```

Figure 5.9: Mean sales by type of store

The fig 5.9 shows that there is significant difference between them and well leave them as it is.

5.5.2 Modify Item_Visibility

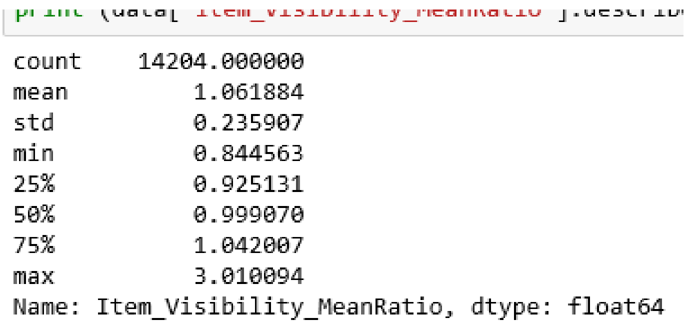
We noticed that the minimum value here is 0, which makes no practical sense. Lets consider it like missing information and impute it with mean visibility of that product.Refer Fig 5.10.



```
Number of 0 values initially: 879
Number of 0 values after modification: 0
```

Figure 5.10: Number of 0 in Item_Visibility

We hypothesized that products with higher visibility are likely to sell more. But along with comparing products on absolute terms, we should look at the visibility of the product in that particular store as compared to the mean visibility of that product across all stores. This will give some idea about how much importance was given to that product in a store as compared to other stores. We can use the visibility_avg variable made above to achieve this.



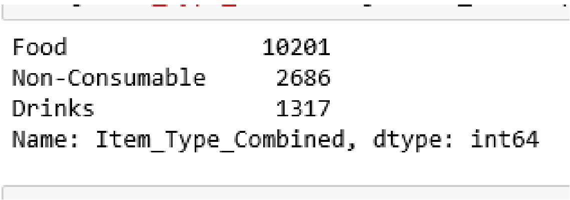
```
print (data['Item_Visibility_MeanRatio'].describ

count    14204.000000
mean         1.061884
std          0.235907
min          0.844563
25%          0.925131
50%          0.999070
75%          1.042007
max          3.010094
Name: Item_Visibility_MeanRatio, dtype: float64
```

Figure 5.11: New statistics of Item_Visibility after imputation

5.5.3 Create a broad category of Type of Item

We have seen that the Item_Type variable has 16 categories which might prove to be very useful in analysis. So its a good idea to combine them.But if we look at the Item_Identifier, i.e. the unique ID of each item, it starts with either FD, DR or NC and If you see the categories, these look like being Food, Drinks and Non-Consumables, so we have used the Item_Identifier variable to create a new column. Refer Fig 5.12.



```
Food              10201
Non-Consumable     2686
Drinks             1317
Name: Item_Type_Combined, dtype: int64
```

Figure 5.12: Category generalization

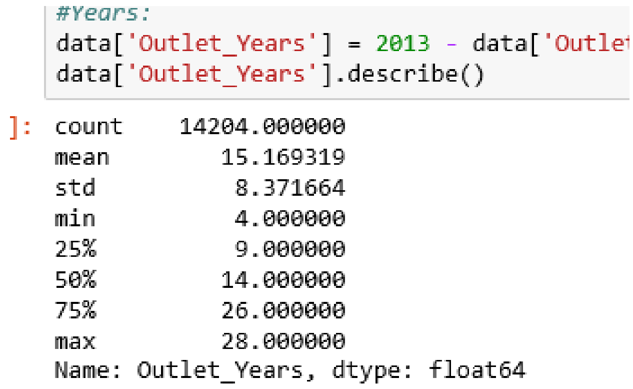
5.5.4 Determine the years of operation of a store

We have made a new column depicting the years of operation of a store. In Fig 5.13 we can see that stores are 4 to 28 years old.

5.5.5 Modify categories of Item_Fat_Content

We found typos and difference in representation in categories of Item_Fat_Content variable. This we have corrected using simple techniques. Refer Fig 5.14.

But we have seen there were some non-consumables as well and a fat-content should not be specified for them. So we have created a separate category for such kind of observations as seen in Fig 5.15.



```
#Years:
data['Outlet_Years'] = 2013 - data['Outle
data['Outlet_Years'].describe()
```

```
]: count    14204.000000
   mean        15.169319
   std          8.371664
   min          4.000000
   25%          9.000000
   50%         14.000000
   75%         26.000000
   max         28.000000
   Name: Outlet_Years, dtype: float64
```

Figure 5.13: Years of operation of a store

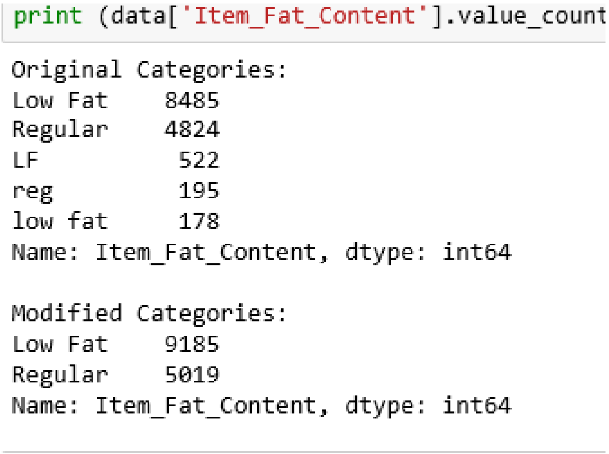
5.5.6 Numerical and One-Hot Coding of Categorical variables

Since scikit-learn accepts only numerical variables, I converted all categories of nominal variables into numeric types. We have created a new variable Outlet same as Outlet_Identifier and coded that.

5.6 Model Building

We have used LabelEncoder from sklearns preprocessing module for coding all categorical variables as numeric. One-Hot-Coding refers to creating dummy variables, one for each category of a categorical variable.

All variables are now float and each category has a new variable. Fig 5.16 shows 3 columns formed from Item_Fat_Content.

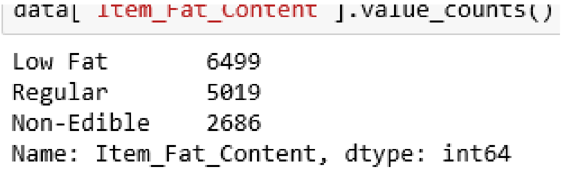


```
print (data['Item_Fat_Content'].value_count

Original Categories:
Low Fat    8485
Regular    4824
LF          522
reg         195
low fat     178
Name: Item_Fat_Content, dtype: int64

Modified Categories:
Low Fat    9185
Regular    5019
Name: Item_Fat_Content, dtype: int64
```

Figure 5.14: Categories of Item_Fat_Content



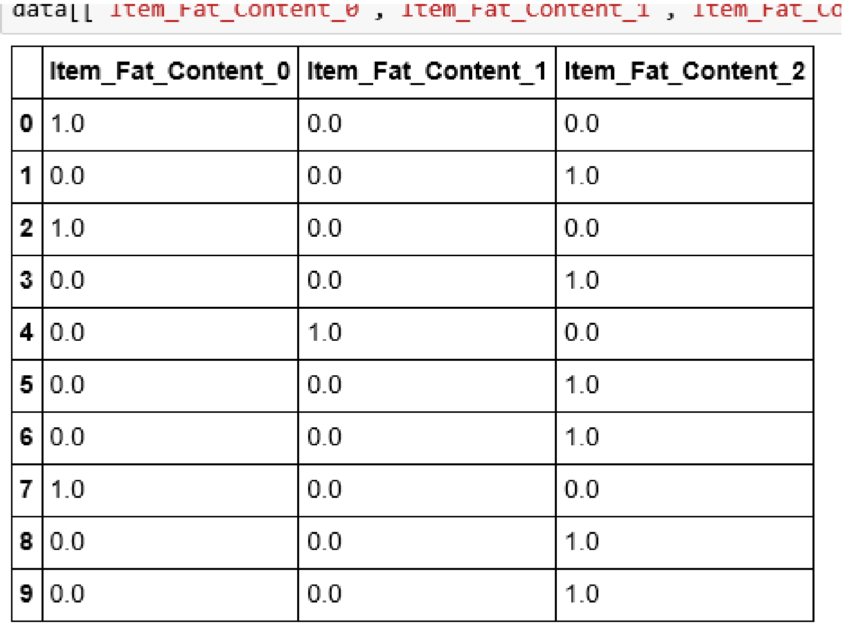
```
data['Item_Fat_Content'].value_counts()

Low Fat       6499
Regular       5019
Non-Edible    2686
Name: Item_Fat_Content, dtype: int64
```

Figure 5.15: Categories of Item_Fat_Content after making new Categories

5.6.1 Exporting Data

Final step is to convert data back into train and test data sets. Its generally a good idea to export both of these as modified data sets so that they can be re-used for multiple sessions.



```
data[[ Item_Fat_Content_0 , Item_Fat_Content_1 , Item_Fat_Co
```

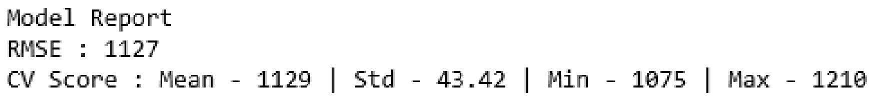
	Item_Fat_Content_0	Item_Fat_Content_1	Item_Fat_Content_2
0	1.0	0.0	0.0
1	0.0	0.0	1.0
2	1.0	0.0	0.0
3	0.0	0.0	1.0
4	0.0	1.0	0.0
5	0.0	0.0	1.0
6	0.0	0.0	1.0
7	1.0	0.0	0.0
8	0.0	0.0	1.0
9	0.0	0.0	1.0

Figure 5.16: Categories of Item_Fat_Content after making new variables

Now we will make predictive models, here we have made 4 models i.e. linear regression,ridge regression model, decision tree and random forest. Lets go through each model one by one:

5.6.2 Linear Regression Model

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. We can see the result of Linear Regression Model in Fig 5.17.

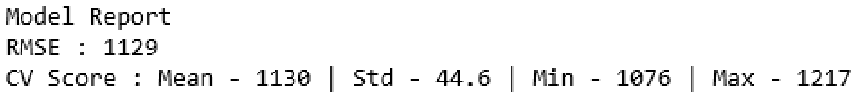


```
Model Report
RMSE : 1127
CV Score : Mean - 1129 | Std - 43.42 | Min - 1075 | Max - 1210
```

Figure 5.17: Result of Linear Regression Model

5.6.3 Ridge Regression Model

Ridge Regression is a variation of Linear Regression. We always prefer a model that catches general patterns. The other is that our goal is predicting it from new data, not specific data. Therefore, model evaluation should be based on new data (testing set), not given data (training set).In ridge regression, we can tune the lambda parameter so that model coefficients change.We can see the result of Ridge Regression Model in Fig 5.18.



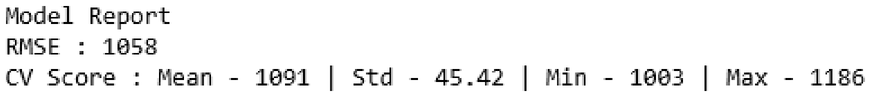
```
Model Report
RMSE : 1129
CV Score : Mean - 1130 | Std - 44.6 | Min - 1076 | Max - 1217
```

Figure 5.18: Result of Ridge Regression Model

5.6.4 Decision tree Model

A decision tree is a machine learning algorithm that partitions the data into subsets. The partitioning process starts with a binary split and continues until no further splits can be made. Various branches of variable length are formed. The goal of a decision tree is to encapsulate the training data in the smallest possible tree. Small trees produce decisions faster than large trees, and they are much easier to look at and understand.

This model is very popular among non-statisticians as they produce a model that is very easy to interpret. Each leaf node is presented as an if/then rule. Cases that satisfy the if/then statement are placed in the node. It can be useful for detecting important variables, interactions, and identifying outliers.We can see the result of Decision tree Model in Fig 5.19.Here you can see that the RMSE is 1058 and the mean CV error is 1091. This tells us that the model is slightly overfitting.



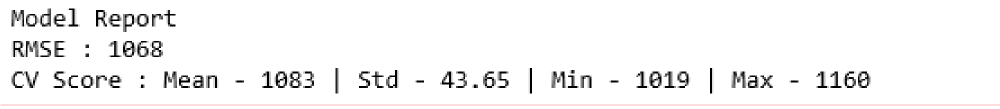
```
Model Report
RMSE : 1058
CV Score : Mean - 1091 | Std - 45.42 | Min - 1003 | Max - 1186
```

Figure 5.19: Result of decision tree Model

5.6.5 Random Forest Model

Random Forest is a supervised learning algorithm and it is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because its simplicity and the fact that it can be used for both classification and regression tasks.

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.We can see the result of Random Forest Model in Fig 5.20.



```
Model Report
RMSE : 1068
CV Score : Mean - 1083 | Std - 43.65 | Min - 1019 | Max - 1160
```

Figure 5.20: Result Random Forest Model

Chapter 6

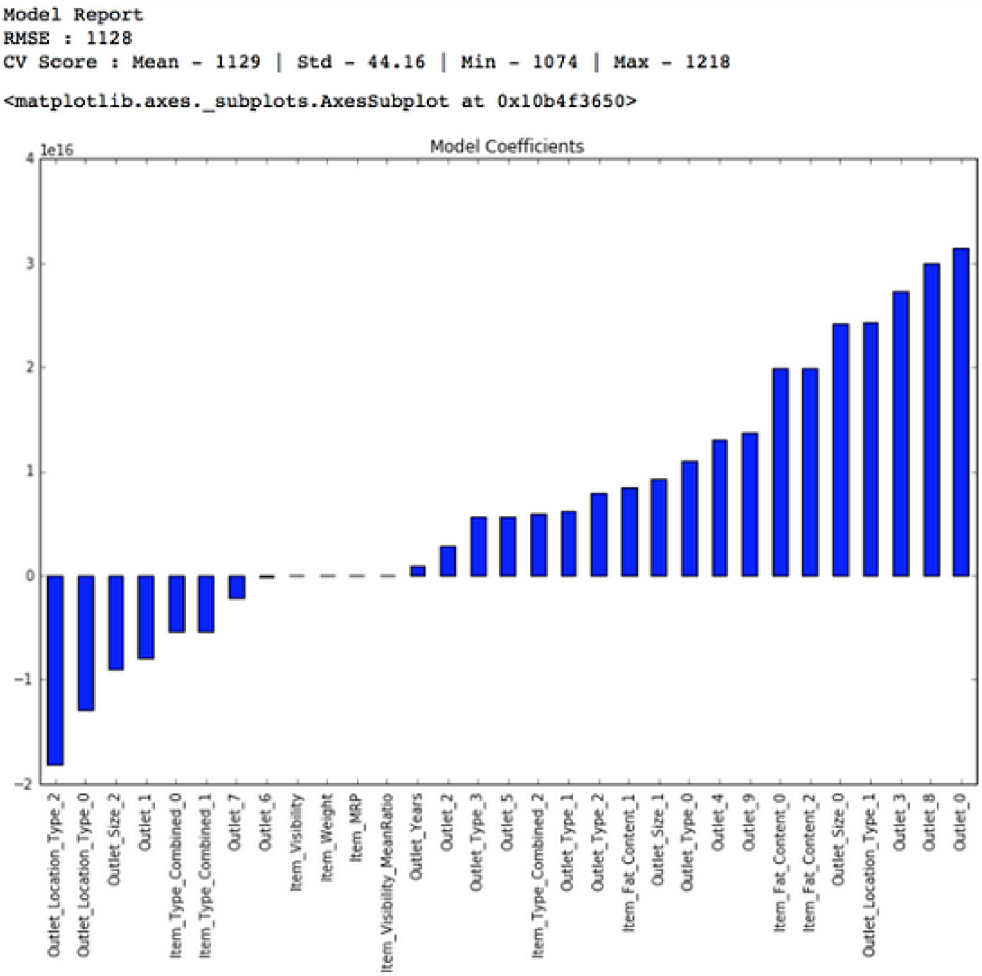
Result and Discussionn

With the information obtained from this product the corporation can identify the products and stores which play a key role in their sales and use that information to take the correct measures to ensure success of their business.

We have successfully developed an application which can allocate the main inventory to the different outlets inventory so that we will get maximum sales result.

We have also got few result from observation like smallest locations produced the lowest sales, However, the largest location did not produce the highest sales.The location that produced the highest sales was the OUT027 location. This location was Supermarket Type3 and its size was medium. This outlet performed much better than any other location.The location that was second was the OUT035 location.

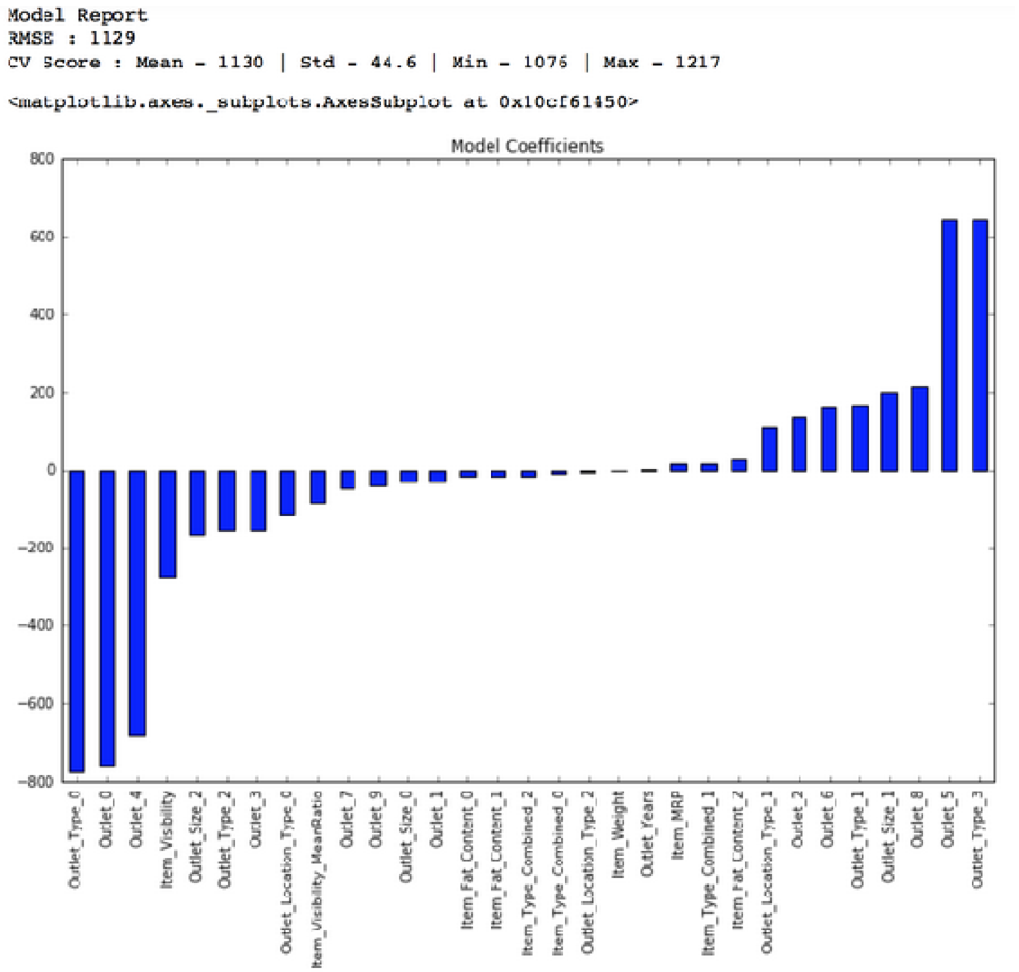
If Big Mart were to try to increase sales at all locations, it may consider switching more locations to Supermarket Type3. Other things Big Mart could do to increase sales is to see which Items had the highest sales. They may also consider how product visibility affected outlet sales. However, the model built in this report should be good for helping Big Mart predict future sales at its locations.



```
Model Report
RMSE : 1128
CV Score : Mean - 1129 | Std - 44.16 | Min - 1074 | Max - 1218
<matplotlib.axes._subplots.AxesSubplot at 0x10b4f3650>
```

Figure 6.1: Result of Linear Regression Model

In Fig 6.1 the coefficients are very large in magnitude which signifies over fitting. To cater to this, we have used Ridge regression model.

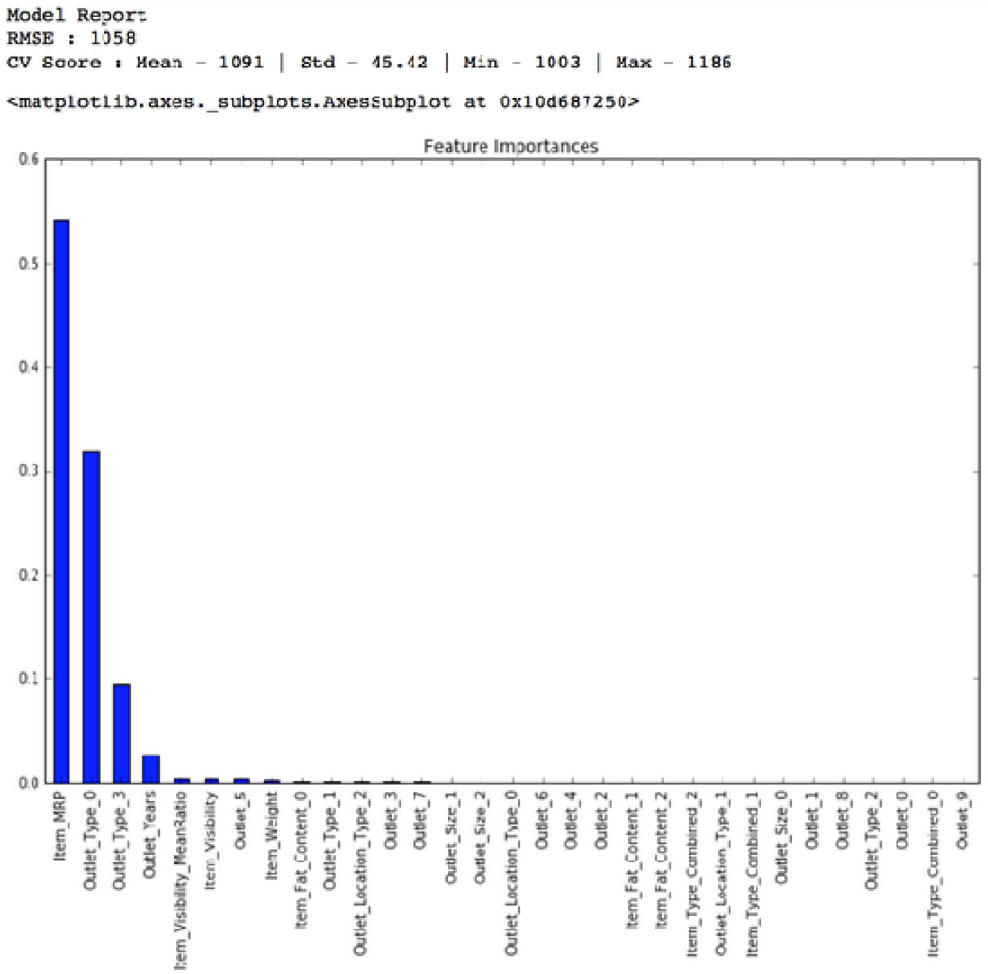


```
Model Report
RMSE : 1129
CV Score : Mean - 1130 | Std - 44.6 | Min - 1076 | Max - 1217

<matplotlib.axes._subplots.AxesSubplot at 0x10cf61450>
```

Figure 6.2: Result of Ridge Regression Model

In Fig 6.2 the coefficients are very large in magnitude which signifies over fitting, but here regression coefficient look better now, the score is about the same. To cater to this, we have used decision tree model.

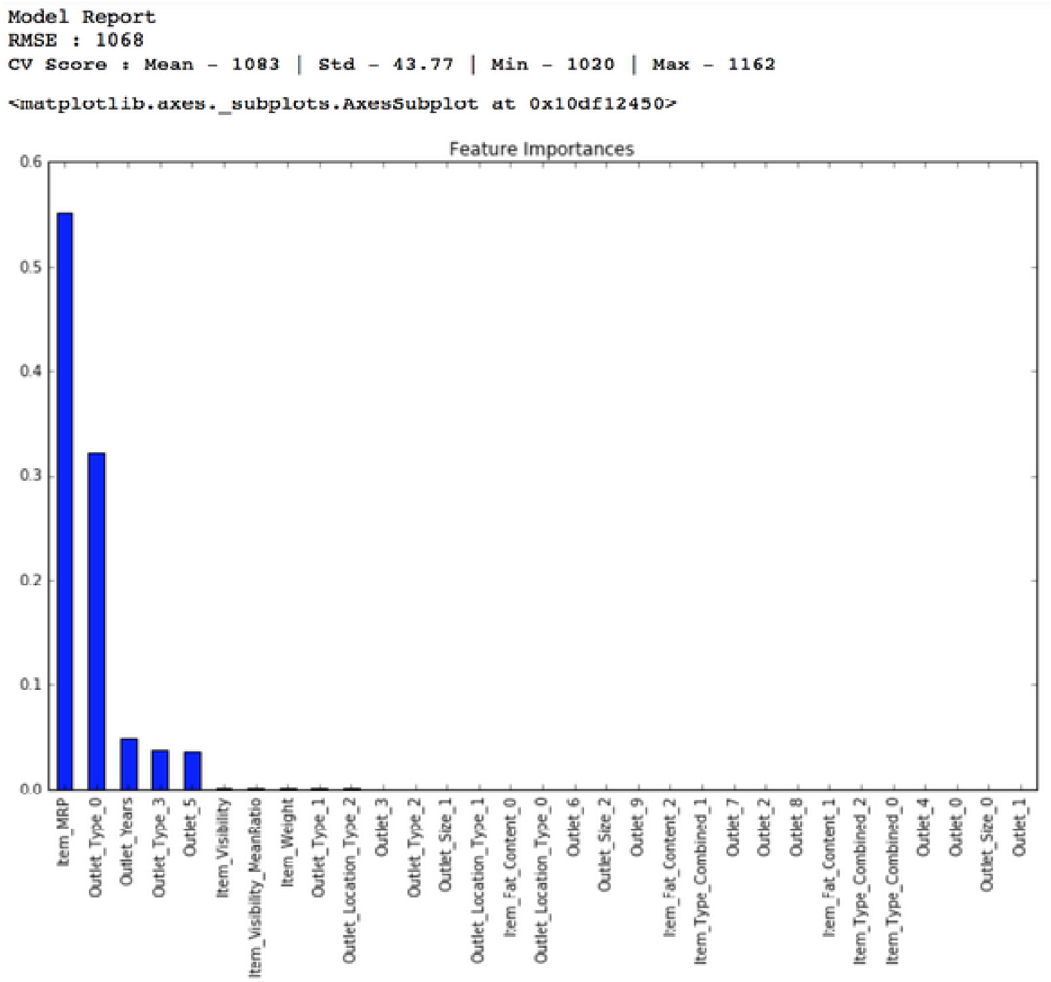


```
Model Report
RMSE : 1058
CV Score : Mean - 1091 | Std - 45.42 | Min - 1003 | Max - 1186

<matplotlib.axes._subplots.AxesSubplot at 0x10d687250>
```

Figure 6.3: Result of Decision tree Model

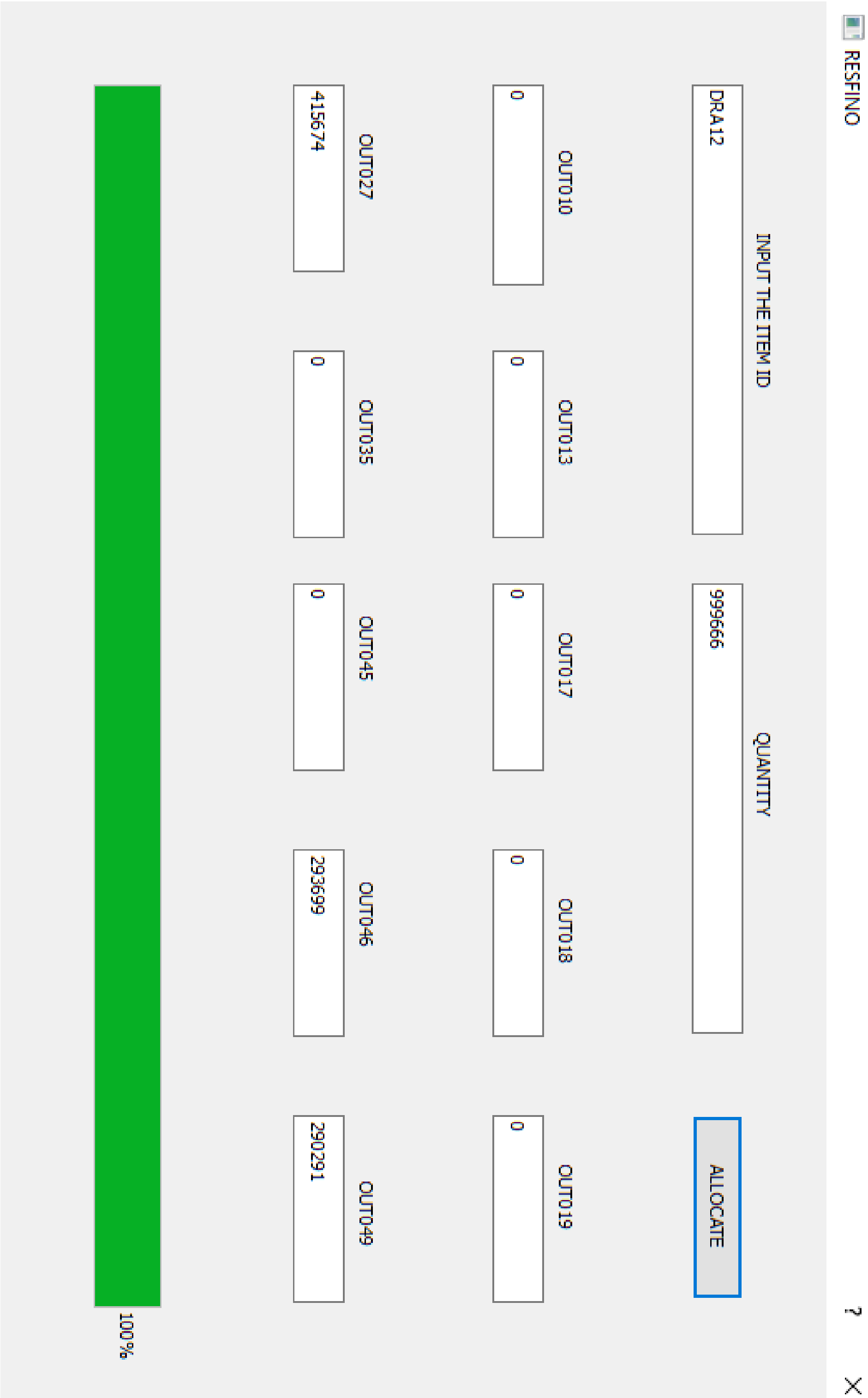
In Fig 6.3 the coefficients are not very large in magnitude which signifies regression coefficient in the model is slightly over fitting . Then we have checked random forest model.



```
Model Report
RMSE : 1068
CV Score : Mean - 1083 | Std - 43.77 | Min - 1020 | Max - 1162
<matplotlib.axes._subplots.AxesSubplot at 0x10df12450>
```

Figure 6.4: Result of Random Forest Model

In Fig 6.4 the coefficients are not very large in magnitude which signifies regression coefficient in the model is slightly over fitting . Hence there is very small improvement.



INPUT THE ITEM ID: DRA12

QUANTITY: 999666

OUT010: 0
OUT013: 0
OUT017: 0
OUT018: 0
OUT019: 0
OUT027: 415674
OUT035: 0
OUT045: 0
OUT046: 293699
OUT049: 290291

100%

Figure 6.5: Graphical user interface of RESFINO Application

Bibliography

[1] H. M. Al-Hamadi. Long-term electric power load forecasting using fuzzy linear regression technique. *IEEE Power Engineering and Automation Conference*, 2011.

[2] Hamid Reza Motahari Nezhad Aly Megahed, Peifeng Yin. An optimization approach to services sales forecasting in a multi-staged sales pipeline. *IBM Almaden Research Center*, 2016.

[3] Hamid Reza Motahari Nezhad Aly Megahed, Peifeng Yin. An optimization approach to services sales forecasting in a multi-staged sales pipeline. *International Conference on Services Computing (SCC)*, 2016.

[4] G Sharma KV Arya AS Tomar, M Singh. *Traffic Management using Logistic Regression with Fuzzy Logic.* Temporal Data Mining via Unsupervised Ensemble Learning, Procedia Computer Science. Elsevier, 2018.

[5] Jos M. Garca Guido Sciavicco Luis Miralles Fernando Jimnez, Gracia Snchez. Multi-objective evolutionary feature selection for online sales forecasting. In *Neurocomputing*, pages 75–92. Elsevier, 2017.

[6] Mykola Pechenizkiy Indre liobaite, Jorn Bakker. Towards context aware food sales prediction. *IEEE International Conference on Data Mining Workshops*, 56, 2009.

[7] Chetna Utreja Premlata Yadav Indu Kumar, Kiran Dogra. A comparative study of supervised machine learning algorithms for stock market trend prediction. *Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 88, 2018.

[8] Shu-Cherng Fang Jian Luo, Tao Hong. Robust regression models for load forecasting. *IEEE Transactions on Smart Grid*, 2018.

[9] Guangwu Hu Yongzhong Huang Jiangtao Ma, Yaqiong Qiao. De-anonymizing social networks with random forest classifier. *IEEE Access*, 6:10139 – 10150, 2017.

[10] CM Friedrich D Reith M Hlsmann, D Borscheid. General sales forecast model for automobile markets and their analysis. In *Industrial Conference on Data Mininng, ICDM 2011*, pages 255–269. Springer, 2011.

[11] M. Costa P. Kela, J. Turkka. Borderless mobility in 5g outdoor ultra-dense networks. *IEEE*, pages 1462–1476, 2015.

[12] TJ Choi S Ren, CP Hui. *A survey of machine learning techniques for food sales prediction.* Artificial Intelligence Review, Artificial Intelligence for Fashion Industry in the sales prediction 2018. Springer, 2018.

[13] George P. Koudouridis James Gross Sahar Imtiaz, Hadi Ghauch. Random forests resource allocation for 5g systems: Performance and robust-

ness study. *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2018.

[14] Hongyan Dai Nina Yan Wenyi Huang, Xiao Qin. *Sales Forecast for O2O Services - Based on Incremental Random Forest Method*. 15th International Conference on Service Systems and Service Management (ICSSSM). 2018.

[15] Yanming Yang. Prediction and analysis of aero-material consumption based on multivariate linear regression model. *3rd IEEE International Conference on Cloud Computing and Big Data Analysis*, 2018.

[16] ShanShan Wang YouLi Feng. A forecast for bicycle rental demand based on random forests and multiple linear regression. *IEEE*, 2013.

[17] katsutoshi yada Yuta Kaneko. A deep learning approach for the prediction of retail store sales. In *16th International Conference on Data Mining Workshops (ICDMW)*, pages 49–54. IEEE, 2016.