IS 733: Data Mining

A Report on
Black Friday Sales Prediction

By
Arushi Singh

Under the guidance of:
James Foulds
Information Systems Department
University of Maryland, Baltimore County

Table of Contents

1. Abstract

Black Friday is an informal name for the Friday following Thanksgiving Day in the United States, which is celebrated on the fourth Thursday of November. The day after Thanksgiving has been regarded as the beginning of America's Christmas shopping season since 1952, although the term "Black Friday" didn't become widely used until more recent decades. This season is crucial for the economy because around 30 percent of annual retail sales occur during the holiday season. For some retailers, such as jewelers, it's even higher, at almost 40 percent.

Research Question: The challenge is to predict the purchase of various products by users across categories given historic data of purchase amounts. This can help the market to focus on the factors that can boost up their sales.

2. Background and Related Work

If consumers spend a lot of money on Black Friday and retailers show strong numbers, investors might have an indication that it is shaping up to be a particularly profitable shopping season. This confidence can be reflected in the stock prices of the retailers that post strong sales. Conversely, many take it as a sign of trouble if retailers are unable to meet expectations on Black Friday. Black Friday is the most important day for all the companies to maximize their sales and it would be very helpful to predict the sales beforehand. We have taken the dataset from Kaggle, where many have worked on the dataset but none of them could produce a proper result by predicting the purchase.

3. Introduction

Black Friday is one of the most important day in the year for discounts, offers and various promotions. In this project, we are planning to analyze previous years Black Friday data sets and predict the sales in the upcoming years based on various attributes such as city, gender, age and the products with high sales.

Glimpse of Black Friday Dataset

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Years_of_Stay	Marital_Status	PC_1	PC_2	PC_3	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN	NaN	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	14.0	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN	NaN	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	NaN	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	NaN	NaN	7969
5	1000003	P00193542	M	26-35	15	A	3	0	1	2.0	NaN	15227
6	1000004	P00184942	M	46-50	7	B	2	1	1	8.0	17.0	19215
7	1000004	P00346142	M	46-50	7	B	2	1	1	15.0	NaN	15854
8	1000004	P0097242	M	46-50	7	B	2	1	1	16.0	NaN	15686
9	1000005	P00274942	M	26-35	20	A	1	1	8	NaN	NaN	7871

Attributes of the Dataset

UserID is the unique identity of the user.

ProductID is the unique identifier of the product.

Gender, Age, City, Marital Status, Occupation and Years of Stay are some of the properties of user.

PC_1 is the highest level of product category.

PC_2 is the second level category of a product.

PC_3 is the third level category of a product.

Purchase is the amount spent by a user on a particular product.

Every product might not have second and third level of product categories. The factors that are to be considered and which will affect the purchase are customer level, city level, product level and store level factors.

Environment

Languages: Python

IDE: Jupyter Notebook

Libraries: Pandas, Numpy, Matplotlib, sklearn

Steps by Step approach

In this project a step by step procedure is followed where the first step is to prepare the data by taking the raw data and processing it according to the need of the project where,

Exploratory Data Analysis: It a technique of exploring the data and understanding the relation between different attributes in the dataset.

Data preprocessing: It is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and lacks certain behaviors and contains errors. Data preprocessing is a proven method of resolving such issues. The next step is data cleaning which involved handling the missing values There are missing values in the data set in PC_2 and PC_3.

Train and Build Models: The next step is to train the models. We have categorized our machine learning system as Supervised Learning task as we have labeled training data and know how much a customer spends on a specific product. Regression task as we will predict the purchase amount (continuous value) of the customer on black Friday.

We have decided to use linear regression and random forest models to predict the purchase amount of customer. After predicting the purchase, we will compare the performance of the two models and the conclusion will be made based on the best predicting model.

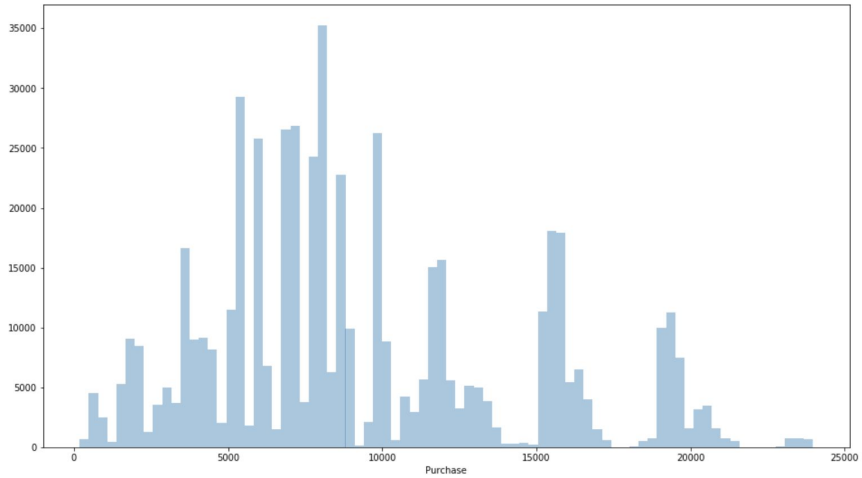
4. Exploratory Data Analysis

Exploratory data analysis explains specific sorts of initial analysis and findings done with data sets, generally at an early stage in an analytical process. Exploratory Data Analysis is an approach for data analysis that employs techniques to maximize insight into a data set. By using

visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. It is an essential way to depict the performance of the data and the task performed on it.

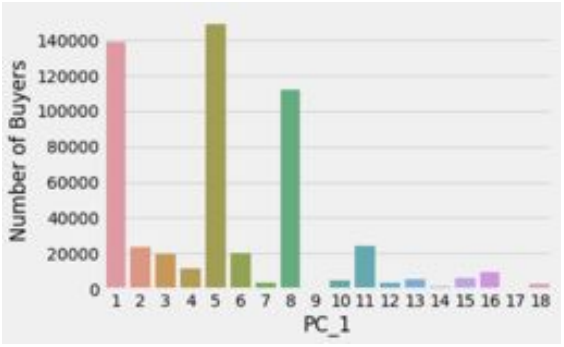
From the black Friday sales data set, the obtained graphs explain as follows:

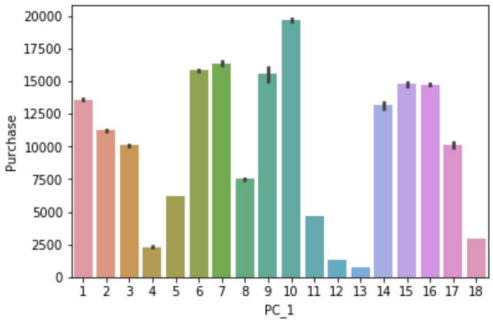
- **Buyers on Cost range Graph**:



The cost range the buyers are acquiring. A large portion of the buyers are spending at a purchase value extended between 5000-10000 $.

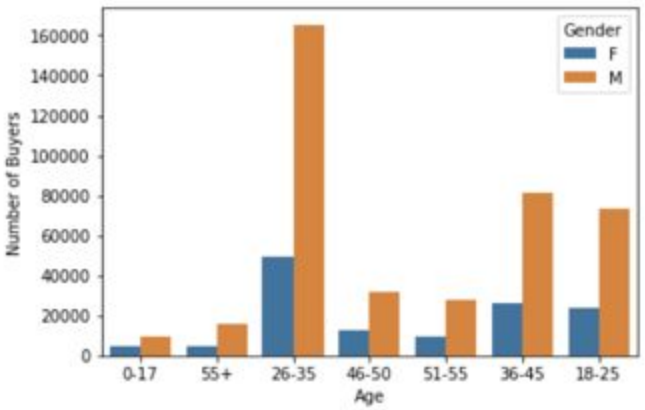
- **Buyers on Product category Graph:**

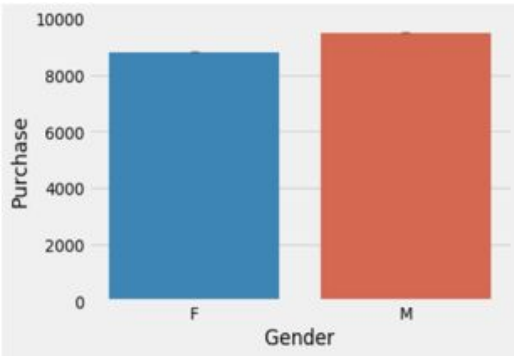


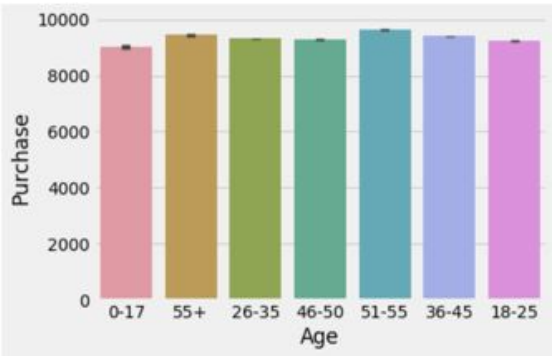


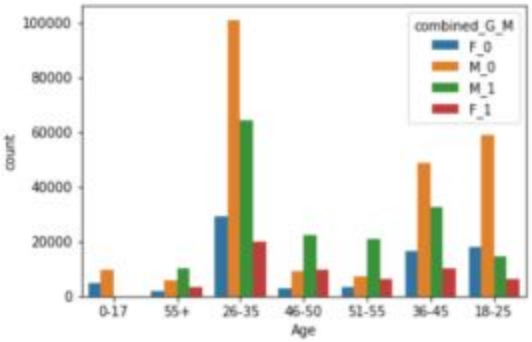
On which product category, most extreme number of buyers are from and has high purchase, through the data set it is observed that in product category-1 there are more number of buyers for product 5 and product 10 has the highest purchase amount.

- **Buyers on Age & Gender Graph:**



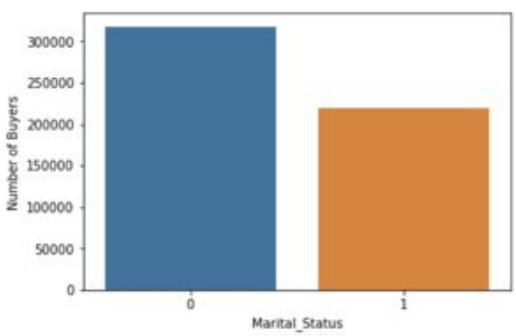


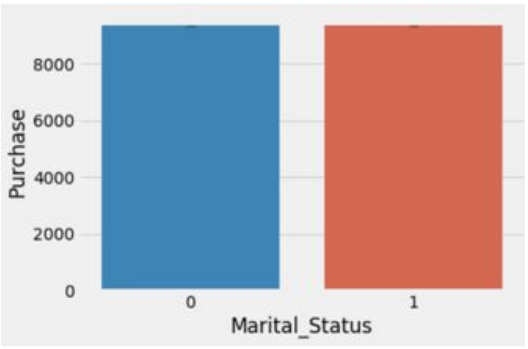




Among the different age groups which gender has attended black Friday sales more. It tends to be seen that a greater number of males spent in the sale than females between the age group 26- 35. From the graph it is observed shorter number of females attended the Black Friday sale. But it could also mean less number of females paid for the products and may be their spouse paid for them. As well as the mass purchase is also from the males compared to females but between the age group 51-55.

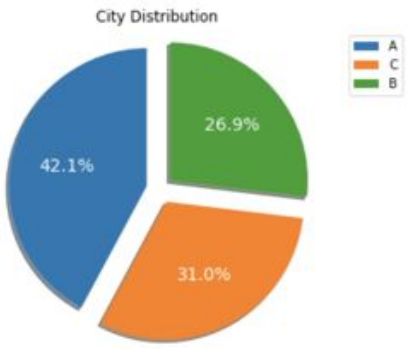
- **Buyers on marital status Graph:**

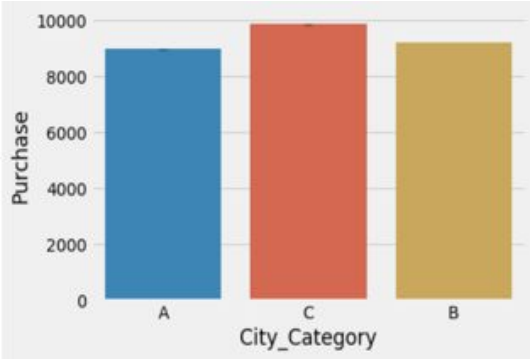




The graph explains greater part of the buyer's participation in the sales are unmarried individuals and purchase amount is similar for married and unmarried individuals.

- **Buyers & Purchase based on City Graph:**





The Pie chart depicts which city has highest number of buyers that is city A has highest number of buyers. But the maximum number of purchases are from the city C.

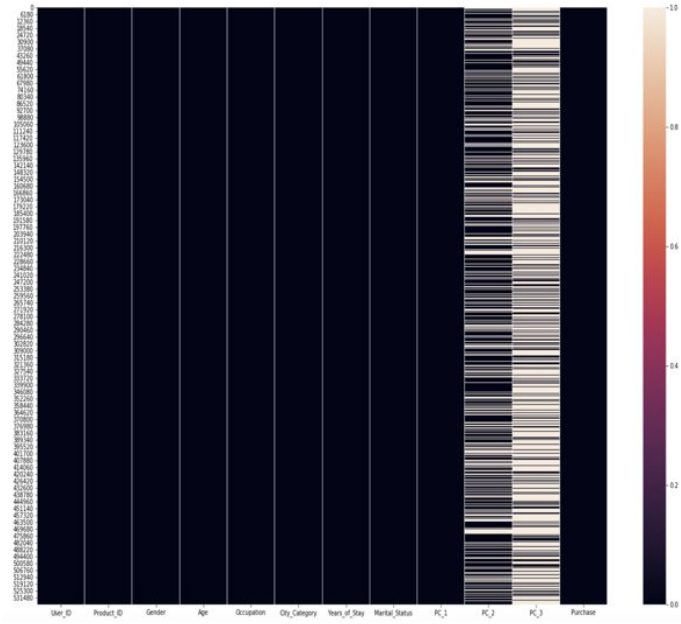
5. Data Preparation

5.1 Data Pre-Processing

Real world data, many times is incomplete, inconsistent or likely to contain error. Data Preprocessing is a method to solve such problems. Data preprocessing, in Data mining is a technique to convert raw data into understandable format. It prepares data for further processing. In order to perform data pre-processing techniques, it is very necessary to understand the dataset first. During our initial exploratory data analysis, we came to conclusion regarding our initial assumptions and our dataset:

1. Age: This column represents the age range instead of exact age of the customer which is in string format. It should be converted into numeric.
2. City_Category: This column is also in a string type representing the city category in which customer resides. It is converted into numeric form as well for which we created dummy variables using label encoder.
3. Gender: As there are only two genders available across the dataset, we converted it into binary M, F are represented as 0,1.

4. Stay_In_Current_City: This column also needs to be converted into int from string datatype. We dealt with '+' symbol by removing it making four highest value of the column.
5. Product_Id and User_Id: These columns are not required for our Prediction model so we dropped them.
6. PC_2 and PC_3: Contains missing values.




```
df.isna().any()
```

User_ID	False
Product_ID	False
Gender	False
Age	False
Occupation	False
City_Category	False
Years_of_Stay	False
Marital_Status	False
PC_1	False
PC_2	True
PC_3	True
Purchase	False
dtype: bool	

From the above heat map plotted for the whole dataset it is observed that PC_1 and PC_2 have missing values.

We first checked the total percentage of missing values to decide whether we can drop them or not. If a small percentage of values are missing, we can drop them as it won't affect much to our model.

```
def missing_values_table(df):
    # fetch Total missing values
    missing_values = df.isnull().sum()
    missing_values_percentage = 100 * missing_values / len(df)

    #creating the table for above variables
    table1 = pd.concat((missing_values, missing_values_percentage), axis = 1)
    print("table1 created!")
    column_name = table1.rename(columns ={
        0: 'Number of Missing values',
        1: '% of total values'
    })

    #sorting the columns with max at first
    column_name = column_name[column_name.iloc[:, 1] != 0].sort_values('% of total values', ascending = False).round(1)

    return column_name

# calling the function with our dataframe
missing_values_table(df)
```

Output:

	Number of Missing values	% of total values
product_category_3	373299	69.4
product_category_2	166986	31.1

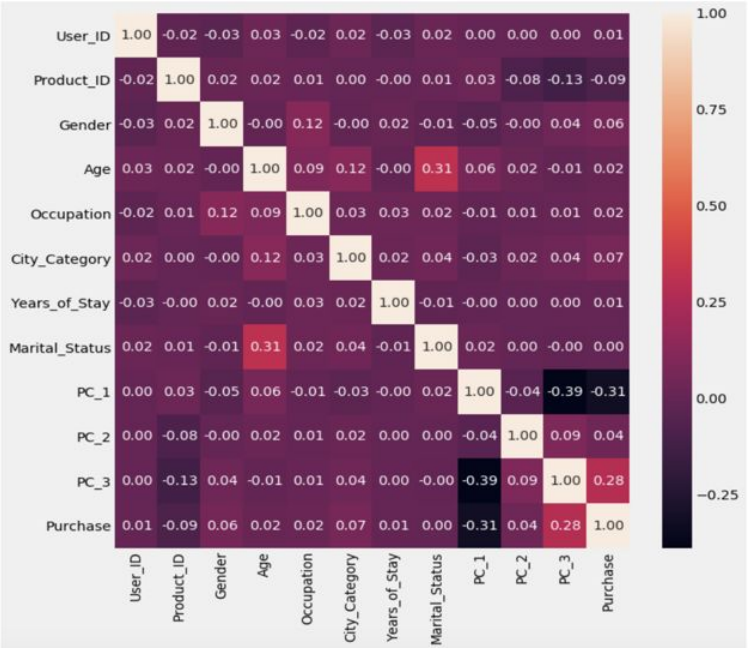
As it is clear from output, PC_3 have almost 70% of total values missing. We cannot drop these rows as it fill result into underfitting of our model. Also, columns PC_2 and PC_3 are dependent on PC_1. Moreover, there are no details provided with dataset about these columns. So, Imputing the missing values with mean, median or mode might result into wrong predictions. So, We're treating missing values as an independent category. Hence, replaced them with zero.

```
df.fillna(value=0,inplace=True)
df["PC_2"] = df["PC_2"].astype(int)
df["PC_3"] = df["PC_3"].astype(int)
print('PC_2', df['PC_2'].unique())
print('PC_3', df['PC_3'].unique())
```

```
PC_2 [ 0  6 14  2  8 15 16 11  5  3  4 12  9 10 17 13  7 18]
PC_3 [ 0 14 17  5  4 16 15  8  9 13  6 12  3 18 11 10]
```

5.2 Data Correlation

A heatmap is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colors. The correlation matrix is a table showing coefficients between sets of variables. Each random variable in the table is correlated with each of the other values in the table. The seaborn python package allows the creation of annotated heatmaps which can be tweeted using Matplotlib tools as per the requirement. A correlation matrix is a table showing correlation coefficients between variables. It is used as a way to summarize data, as an input into a more advanced analysis, and a diagnostic for advanced analyses. The three broad reasons for computing a correlation matrix is to summarize a large amount of data where the goal is to see patterns, to input into other analyses, as a diagnostic when checking other analyses. The diagonal of the table is always set of ones, because the correlation between a variable and itself is always 1.



In our data set to apply models we thought of implementing the correlation matrix to know the correlation between the variables to use them for the analyses. Out of the correlation matrix we found that the user_id and product_id have very less correlation and it won't contribute to the models as other attributes do and hence we have dropped these two attributes.

5.3 Training and Testing Sets

```
from sklearn import linear_model
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,random_state = 42)
print('Training Features Shape ', x_train.shape)
print('Training Labels Shape:', y_train.shape)
print('Testing Features Shape:', x_test.shape)
print('Testing Labels Shape:', y_test.shape)
```

```
Training Features Shape  (430061, 9)
Training Labels Shape: (430061,)
Testing Features Shape: (107516, 9)
Testing Labels Shape: (107516,)
```

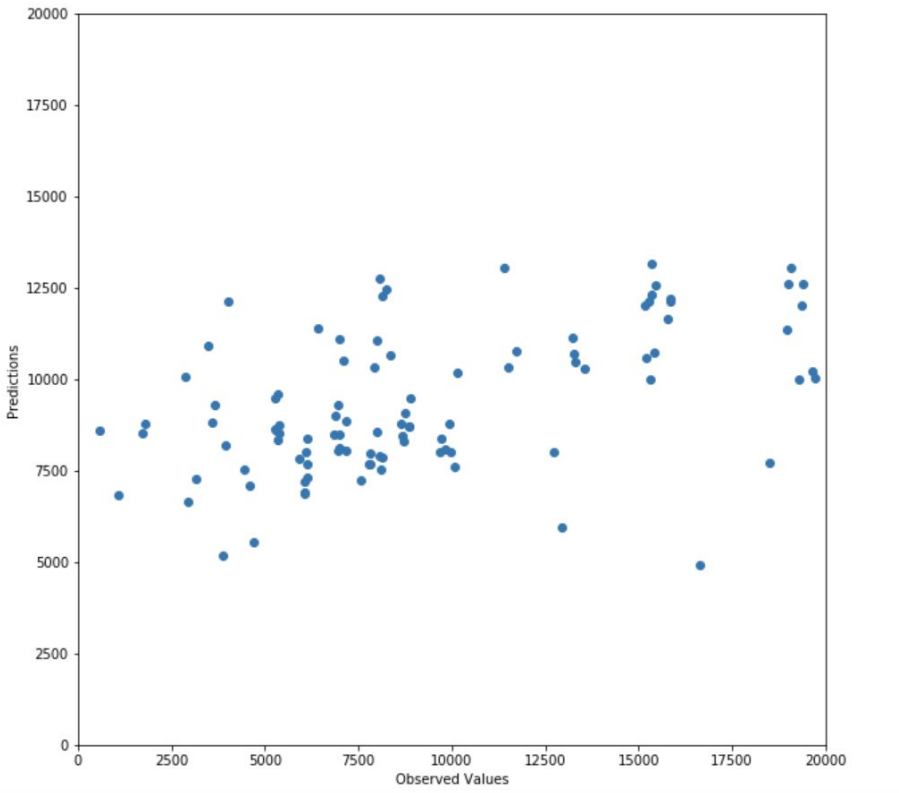
The final step in data preparation is splitting data into training and testing sets. We have split the data set as 80% training data and 20% test data. We train the model using the training data and then make predictions on the test data and compare the results with the observed values in the dataset. In our data set the class label is purchase and we have predicted the purchase by splitting the data on eighty-twenty bases and applied the model on training set.

6. Linear Regression

Linear regression is a basic and commonly used type of predictive analysis. Linear regression is similar to correlation in that the purpose is to measure to what extent there is a linear relationship between two variables. In particular, the purpose of linear regression is to predict the value of the dependent variable based upon the values of one or more independent variables. In multiple linear regression two or more independent variable are used to predict the value of a dependent variable. The dependent variable must be measured on a continued measurement scale. In our data set the purchase value was a continuous variable which is the reason we choose linear regression to predict it. Our main goal is to predict the purchase of the users. So, our dependent variable is purchase in this case and all other variables are independent variables. We have applied linear regression with multiple variables.

```
lm = linear_model.LinearRegression()
model=lm.fit(x_train,y_train)
predictions_linear_regression=lm.predict(x_test)
from matplotlib import pyplot as plt
plt.figure(figsize=(10, 10))
plt.scatter(y_test[0:100], predictions_linear_regression[0:100])
plt.xlabel('Observed Values')
plt.ylabel('Predictions')
plt.xlim(0,20000)
plt.ylim(0,20000)
plt.show()
```

We have trained the model with the training data and then predicted the purchase for the test data. After making predictions, we have compared the predicted values with the actual values in the test data. We plotted a scatter plot for random 100 predicted variables.



It is found that only two out of five values were closer to the actual values. The root mean square error (RMSE) is a standard deviation of the residuals. Residuals are a measure of how far from the regression line data points are, it tells you how concentrated the data is around the line of best fit. In our case the RMSE score is about 4636.86.

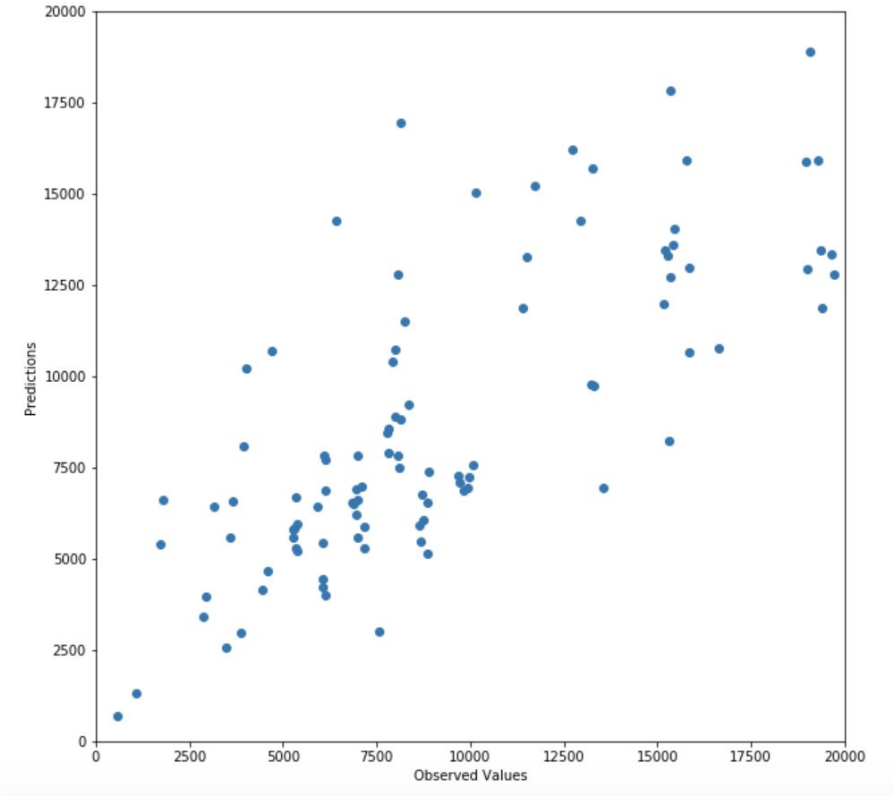
Model performance is estimated in terms of its accuracy to predict the occurrence of an event on unseen data. A more accurate model is seen as a more valuable model. The RMSE score is used to evaluate accuracy. The accuracy for our model is 31.78%.

7. Random Forest

Random forest is an ensemble learning method for regression and classification. Random forest operates by including multiple decision trees. A decision tree consists of a set of questions to be answered to predict the class label. Random forest not only includes one decision tree but a set of decision trees. If class is categorical the final output will be the majority of class values predicted by all decision trees and if the class is continuous value the final output will be the mean of the values of all the decision trees. In our case we are predicting the purchase amount which is a continuous value. We have applied random forest regression model by first training the model with 80% training data and predicted the purchase amount for 20% test data.

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 10)
model_random_forest=regressor.fit(x_train, y_train)
predictions_random_forest = regressor.predict(x_test)
plt.figure(figsize=(10, 10))
plt.scatter(y_test[0:100], predictions_random_forest[0:100])
plt.xlabel('Observed Values')
plt.ylabel('Predictions')
plt.xlim(0, 20000)
plt.ylim(0, 20000)
plt.show()
```

We have plotted a scatter plot for the predicted values and the actual values in which we have just taken 100 random instances from test data for better comparison. We have observed that four out of five values were nearly closer.

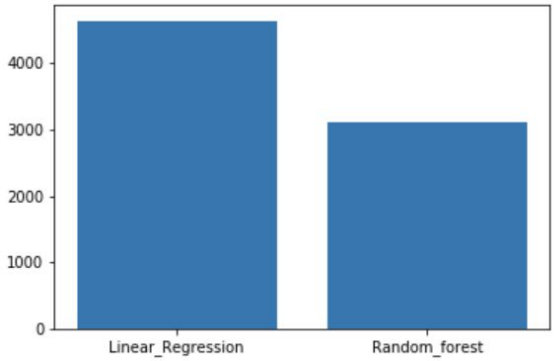


The Root mean square error is 3100.33 and accuracy is determined to be 66.89%.
We have also calculated the accuracy by hyperparameter tuning. Calculated the accuracy with different number of estimators i.e., the number of decision trees and max_depth i.e., the depth of the decision tree but we did not observe any increase in the accuracy/ performance.

8. Conclusion

- Random Forest model can be used to predict class efficiently as it not only depends on the outcome of individual decision tree but considers the outcomes of multiple decision trees.

- Comparing the root mean square error of the two algorithms Random Forest is found to have less root mean square error.



- We have made our predictions using Random Forest algorithm and were successfully able to predict the purchase amount with 66.89% accuracy. Age, Marital status, gender and city were observed to be the highest contributors to the purchase amount.
- Considering the insights from exploratory data analysis companies can concentrate on particular age group, city and gender and target the products related to these users to improve their sales.

9. Future work

- We can look forward to apply boosting algorithms such as XG boost algorithm which might give better performance. Boosting algorithms might have less root mean square error compared to regression models.
- We can also visualize the decision trees.
- Variable importance can be calculated which represents how much prediction changes by including a particular variable and we can completely eliminate the attributes with less importance and calculate the performance.

10. References

1. BlackFriday Dataset has been taken from https://www.kaggle.com/mehdidag/black-friday