# Project Report
on
Analysis and Prediction of Black Friday Dataset

Team 2
Prem kumar Kamasani, Manjusha Gadupudi, Shristi Bhat, Apoorva Ramesh

December 21, 2018

## 1 Abstract

The amount of data available for predicting customer purchase behavior is unprecedented in the history of commerce. Adult users in the US are spending an incredible 5.6 hours per day with digital media, and over 50 percent of that is on mobile devices. With all that data — and the technology to crunch the numbers — marketers can know how their customers are likely to behave, sometimes before they know themselves. Most of our buying decisions are not based on well-defined logic. Emotions, trust, communication skills, culture and intuition play a big role in our buying decisions. Although humans do not follow a well-defined logic, we do have some repeated patterns. We often buy the same things, behave in a similar way and follow similar intuitions. So, if we can learn the buyer's pattern, we may be able to identify the next buyer too! Companies struggling to understand or implement descriptive analytics are now challenged to adopt this new form of predictive analytics. Organizations that incorporate predictive analytics into their daily operations in this way improve their business processes, enhancing decision making and gaining the ability to direct, optimize, and automate decisions, on demand, to meet defined business goals.

Segmentation involves separating a market into subgroups with similar attitudes, demographic, geographic, or behaviors. After segmenting your market, you position your product to appeal to the wants and needs of the chosen segment(s)–your target market. Data aids in crafting your target segment(s) and determining the most effective positioning for each. Predictive analytics also helps to identify the most profitable segments based on historical consumer behavior within each segment. Marketing managers use this data to allocate resources to reach the most profitable segments. Forecasting is the biggest use of customer behavior prediction is in developing demand models that forecast sales and revenue – the starting point for budgeting.

Demand pricing, often called yield management, involves pricing products based on differences in elasticity of demand between consumer groups. For instance, business travelers are willing to pay more for a seat on an airplane than casual travelers, so you can charge them more and reduce the price to casual travelers to make your flights more competitive and still meet ROI (Return of Investment) goals. Using predictive analytics, firms conduct a series of experiments to determine factors affecting the impact of price on demand. Using these predictive models, firms develop optimum pricing strategies that maximize ROI. Improve customer satisfaction - Customer satisfaction greatly impacts retention and loyalty. It also improves other positive consumer behaviors, such as recommending the brand to others. Any improvement in customer satisfaction impacts ROI, potentially significant. Data suggest that it's 5X less expensive to keep an existing customer than replace that customer.

## 2 Introduction

As customers we have been promised to receive better service, faster. As marketers we are supposed to have acquired intimate knowledge about the behavior, the preferences, the needs, and the wants of our customers. All of us who are touched by marketing, either on the consumer or on the vendor side, have had ample opportunity to experience what these systems can and cannot do for us.

It is critical for every company to put in place strategies for predicting customer behavior. Customer needs, situations, expectations, and demands a constantly changing and evolving and there would be no way to understand them beyond 'today' without some manner of predicting customer behavior.

The objective is to predict the Purchase field given all the other features. The dataset here is a sample of the transactions made in a retail store on Black Friday. We want to know better the customer purchase behavior against different products. We are trying to predict the amount of purchase (target) with the help of information contained in other variables (predictor).

We want to build a model to predict the purchase amount of customer against various classifiers which will help us to get insights into the spending trends of customers on a Black Friday. Dataset is taken from https://www.kaggle.com/mehdidag/black-friday

This dataset comprises of sales transactions captured at a retail store. It contains 550,000 observations and 12 variables.

| Attribute | Data Type | Description |
|---|---|---|
| User Id | Integer | user id of every customer. There are 5900 unique user id's |
| Product Id | Factor | ID of each product. There are 3600 unique product id's |
| Gender | Factor | It has two values M and F. |
| Age | Bins | It has 7 values. Ages are binned into 7 age groups |
| Occupation | Integer | It has 21 Values starting from 0 to 20 |
| City Category | Factor | There are 3 values A, B, C |
| Stay in Current City Years | Integer | This attribute has number of years customer stayed in that city |
| Marital Status | Factor | It has 2 values Yes and No |
| product category 1 | Integer | It has 18 values from 1 to 18 |
| product category 2 | Integer | This values are similar to previous attribute but stores only if the product has second categ |
| product category 3 | Integer | This attribute stores third category of the product if it has any. |
| Purchase | Integer | It stores the amount of the product in usd. |

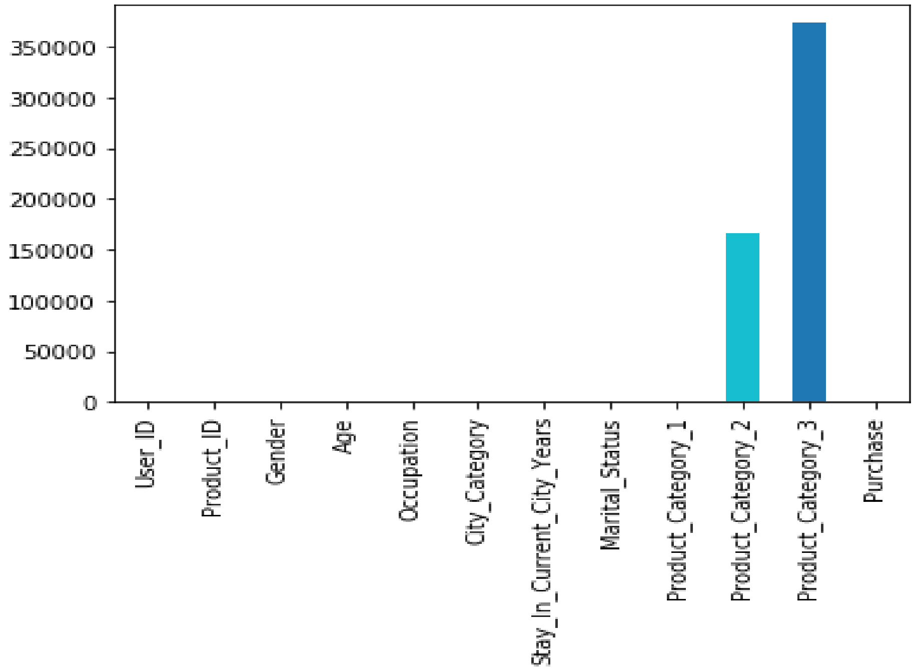


Figure 1: Bar graph of Null values in the data set
x-axis has attributes and y-axis denotes number of NUll values of each attribute

This dataset does not have any null values except product-category-2 and product-category-3. These two attributes with NULL values means that specific product falls under only one category so remaining categories are NULL values.

# 3 Related Work

To predict the purchase value there are various approaches or models. But in this project we are going to implement following 4 models to predict the purchase value.

- Linear Regression
- Decision Tree
- Random Forest
- Neural Network

## 3.1 Linear Regression

Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable. The red line in the above graph is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best. The line can be modelled based on the linear equation shown below.

$$y = a_0 + a_1 * x$$

The motive of the linear regression algorithm is to find the best values for $a_0$ and $a_1$. Before moving on to the algorithm, let's have a look at two important concepts you must know to better understand linear regression.

• Cost Function
The cost function helps us to figure out the best possible values for $a_0$ and $a_1$ which would provide the best fit line for the data points. Since we want the best values for $a_0$ and $a_1$, we convert this search problem into a minimization problem where we would like to minimize the error between the predicted value and the actual value.

$$\text{minimize } \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$
$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

• Minimization and Cost Function
We choose the above function to minimize. The difference between the predicted values and ground truth measures the error difference. We square the error difference and sum over all data points and divide that value by the total number of data points. This provides the average squared error over all the data points. Therefore, this cost function is also known as the Mean Squared Error(MSE) function. Now, using this MSE function we are going to change the values of $a_0$ and $a_1$ such that the MSE value settles at the minima.

• Gradient Descent
The next important concept needed to understand linear regression is gradient descent. Gradient descent is a method of updating $a_0$ and $a_1$ to reduce the cost function(MSE). The idea is that we start with some values for $a_0$ and $a_1$ and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.

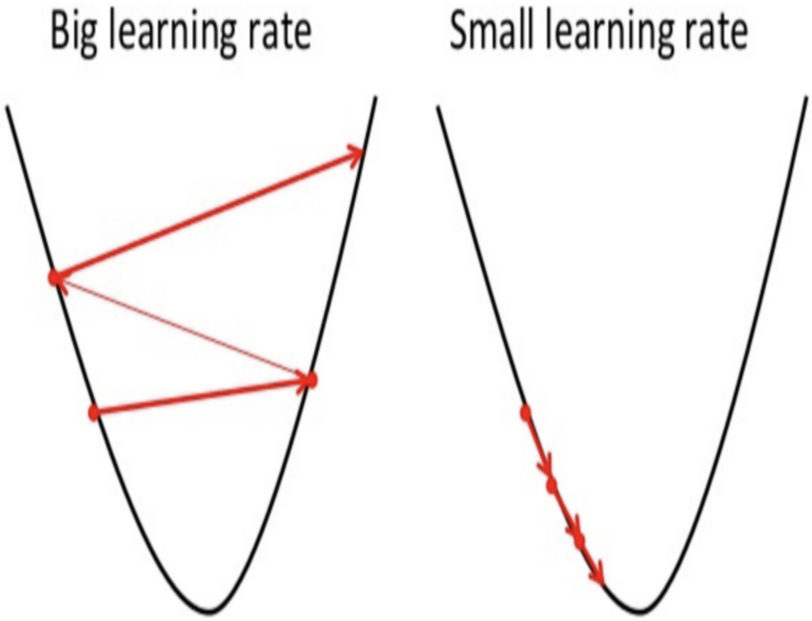


Figure 2

To draw an analogy, imagine a pit in the shape of U and you are standing at the topmost point in the pit and your objective is to reach the bottom of the pit. There is a catch, you can only take a discrete number of steps to reach the bottom. If you decide to take one step at a time you would eventually reach the bottom of the pit but this would take a longer time. If you choose to take longer steps each time, you would reach sooner but, there is a chance that you could overshoot the bottom of the pit and not exactly at the bottom. In the gradient descent algorithm, the number of steps you take is the learning rate. This decides on how fast the algorithm converges to the minima.

## 3.2 Decision Tree

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.
Learning model is based on a decision tree to go from observations about an item to conclusions about the item's target value. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. Works by recursively selecting the best attribute to split the data and expanding the leaf nodes of the tree until the stopping criterion is met.

• DECISION TREE – APPROACH
The choice of best split test condition is determined by comparing the impurity of child nodes and depends on which impurity measurement is used. Those selections would be based on what features are believed to be most valuable in a data set, they would constitute the root.

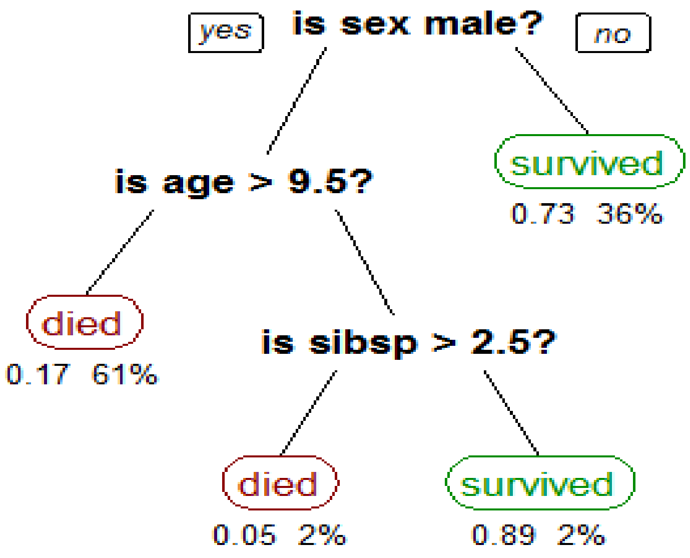


Figure 3

## 3.3 Random Forest

One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. I will talk about random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with two trees:

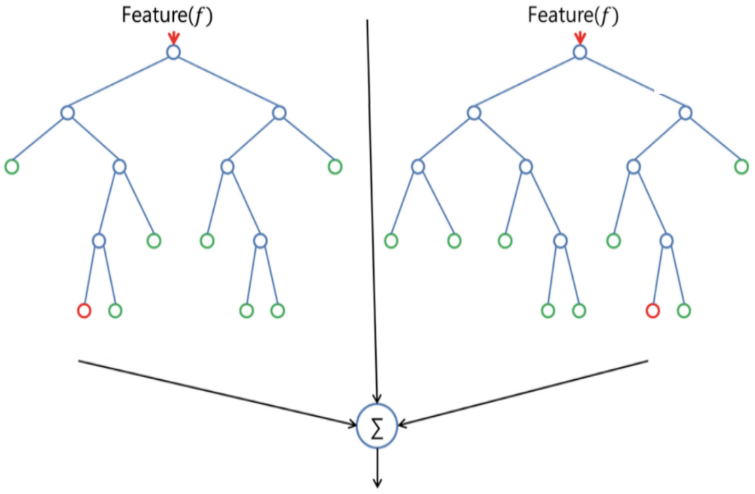


Figure 4

Random Forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, you don't have to combine a decision tree with a bagging classifier and can just easily use the classifier-class of Random Forest. Like I already said, with Random Forest, you can also deal with Regression tasks by using the Random Forest regressor.

Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.
Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

## 3.4 Neural Network

Neural Networks are a class of models within the general machine learning literature. Neural networks are a specific set of algorithms that have revolutionized machine learning. They are inspired by biological neural networks and the current so-called deep neural networks have proven to work quite well. Neural Networks are themselves general function approximations, which is why they can be applied to almost any machine learning problem about learning a complex mapping from the input to the output space.
Deep learning is the name we use for "stacked neural networks"; that is, networks composed of several layers.
The layers are made of nodes. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs for the task the algorithm is trying to learn. (For example, which input is most helpful is classifying data without error?) These input-weight products are summed, and the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal progresses further through the network to affect the ultimate outcome, say, an act of classification. Here's a diagram of what one node might look like

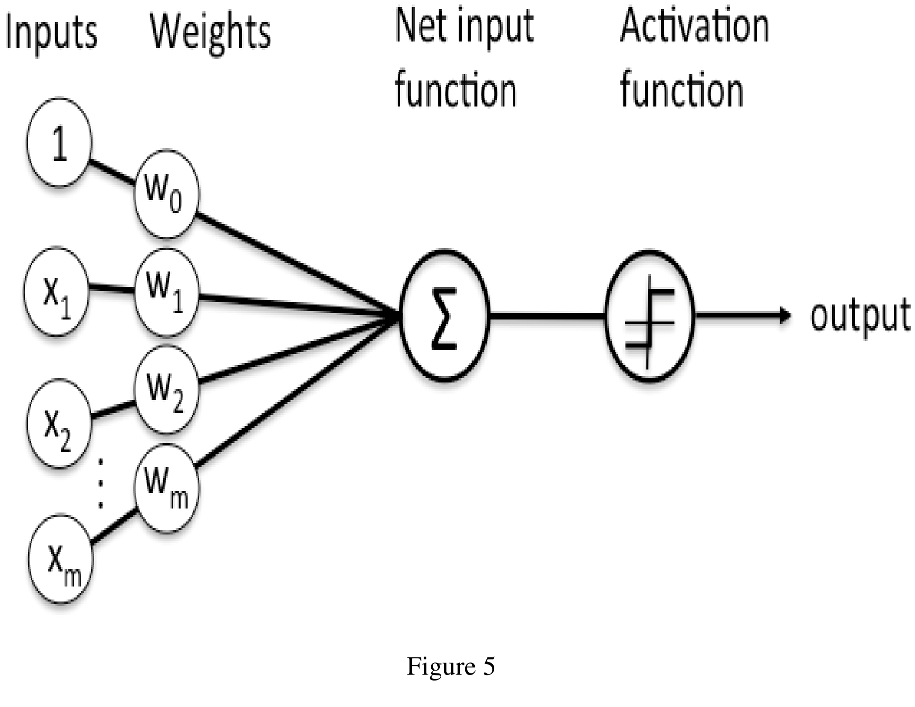


Figure 5

A node layer is a row of those neuronlike switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving your data.

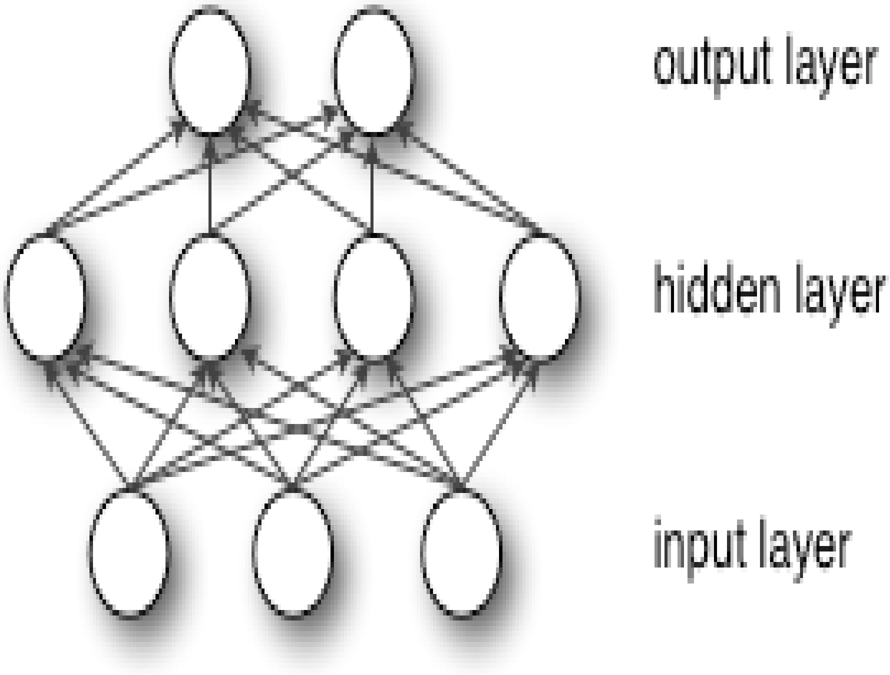


Figure 6

Pairing adjustable weights with input features is how we assign significance to those features regarding how the network classifies and clusters input.

## 4 Problem Definition

The dataset consists of Numerical and categorical data with numerical target variable which is basically a regression problem. We want to build a model to predict the purchase amount of customer against various classifiers which will help us to get insights into the spending trends of customers on a Black Friday.

## 5 Method and Methodology

In this section we would like to introduce the flow of our project. Since the problem here is regression problem and there are many implementation techniques already in use, we would like to apply some models on the dataset and compare them based on their RMSE. Following are the steps of implementation of this project:

- Data Cleaning and Processing
- Exploratory Data Analysis
- Splitting Data
- Applying different models to predict Purchase. They are:
  - Linear regression
  - Decision Tree
  - Random Forest
  - Neural Network
- Comparing the results of all Models

Two attributes(product category 2 and product category 3) has NULL values. We replace all the NULL values with 0. So 0 in those 2 attributes represents that the specific product does not fall under any category in product category 2 and product category 3.

To implement models we should not have categorical data. But our dataset has categorical attributes. So, all the categorical attributes are converted into numerical attributes.// After splitting the data into train and test dataset, we will generate models using all predictor variables available to predict the purchase. As mentioned in the introduction part there are 5900 user ID's and 3600 Product ID's which is very small value compared to total number of observations. Which means that, User ID's and Product ID's are repeated significantly more number of times. So those attributes may contain vulnerable information to predict purchase. However, these attributes will not be useful for the new customers, which may leads to data leakage. To avoid this unnecessary data leakage we will implement the model again removing the User Id and Product Id attributes from the list of predictor variables.
Later, we will group the data based on different categories and apply the models again. Even those results are compared.

# 6 Experimental Setup

Implemented Linear Regression, Decision Tree, Random Forest in:

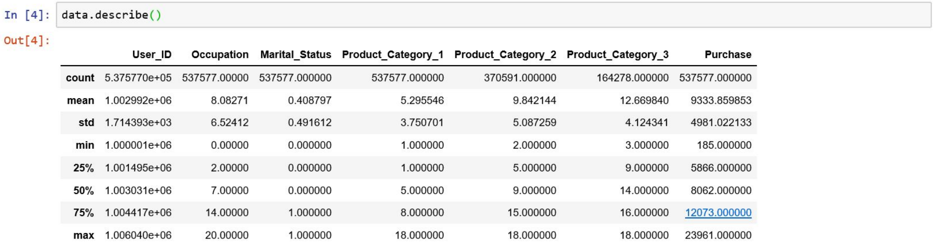
| | |
|---|---|
| Operating System | Windows 10 |
| RAM | 4GB |
| Programing Language | Python 3.7.5 (implemented in Jupyter Notebook) |
| Libraries | Pandas, Numpy, Matplotlib.pyplot, sklearn, scipy, seaborn, statsmodels.api |

Implemented Neural Network in:

| | |
|---|---|
| Operating System | Windows 10 |
| RAM | 4GB |
| Programing Language | R (implemented in R Studio) |
| Libraries | Keras, mlbench, dplyr, magrittr, neuralnet |

# 7 Experimental Reults and Analysis

- Exploratory Data Analysis:



```
In [4]: data.describe()
Out[4]:
```

| | User_ID | Occupation | Marital_Status | Product_Category_1 | Product_Category_2 | Product_Category_3 | Purchase |
|---|---|---|---|---|---|---|---|
| **count** | 5.375770e+05 | 537577.00000 | 537577.000000 | 537577.000000 | 370591.000000 | 164278.000000 | 537577.000000 |
| **mean** | 1.002992e+06 | 8.08271 | 0.408797 | 5.295546 | 9.842144 | 12.669840 | 9333.859853 |
| **std** | 1.714393e+03 | 6.52412 | 0.491612 | 3.750701 | 5.087259 | 4.124341 | 4981.022133 |
| **min** | 1.000001e+06 | 0.00000 | 0.000000 | 1.000000 | 2.000000 | 3.000000 | 185.000000 |
| **25%** | 1.001495e+06 | 2.00000 | 0.000000 | 1.000000 | 5.000000 | 9.000000 | 5866.000000 |
| **50%** | 1.003031e+06 | 7.00000 | 0.000000 | 5.000000 | 9.000000 | 14.000000 | 8062.000000 |
| **75%** | 1.004417e+06 | 14.00000 | 1.000000 | 8.000000 | 15.000000 | 16.000000 | 12073.000000 |
| **max** | 1.006040e+06 | 20.00000 | 1.000000 | 18.000000 | 18.000000 | 18.000000 | 23961.000000 |

Figure 7: Variable Metrics Table

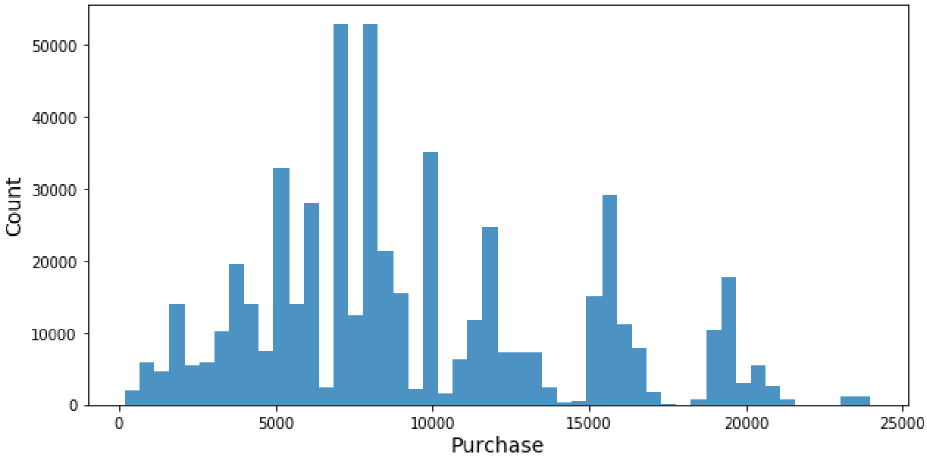


Figure 8: Purchase vs count

In this graph we can see the count of the purchases made in the the purchase range, we observe that most of the purchases are in the range of 5000 USD to 10000 USD

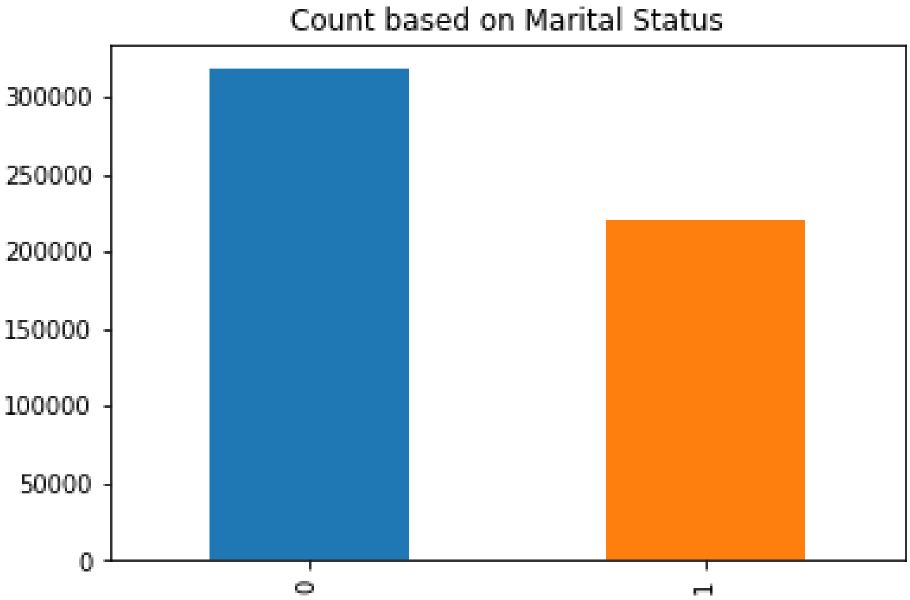


Figure 9: Marital Status vs count

In this graph '0' indicates that the person has been married and '1' indicates the person to be unmarried. We observe that there are more married people shopping during this time of the year.

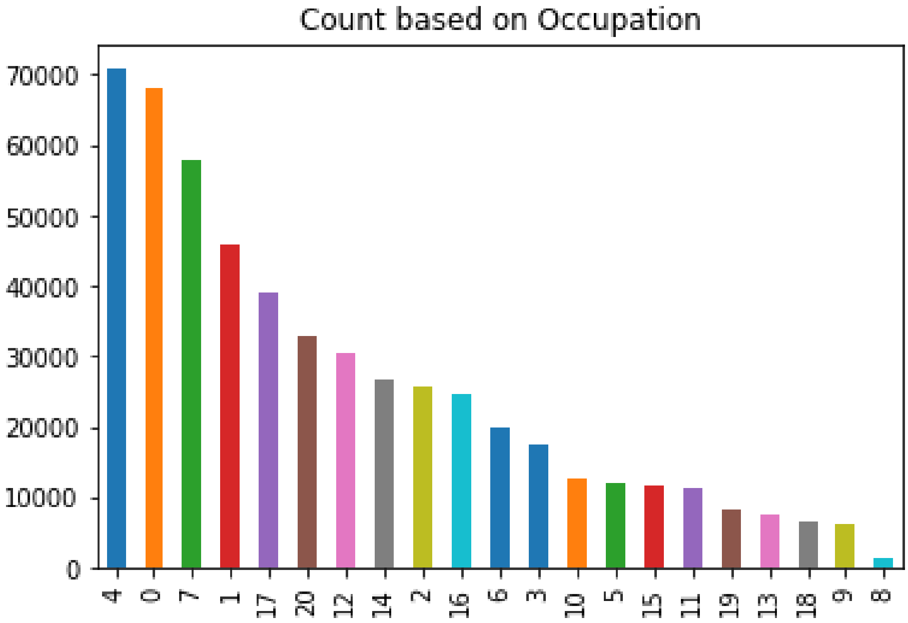
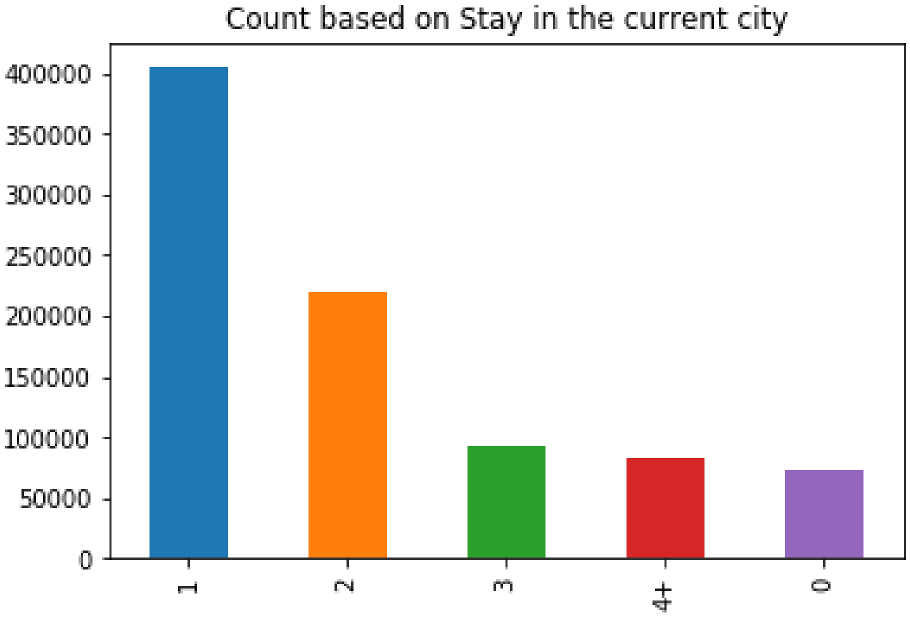


Figure 10: Occupation vs Count

The data has 21 categories of occupation which are in the range of 0-21 and the people belonging to category 4 shop a lot according to the data.



.png

Figure 11: Number of years stay in the current city vs Count
This shows the number of years these people have stayed in the particular city. It has 5 categories 0-4+ and people belonging to 1 year shop the most.

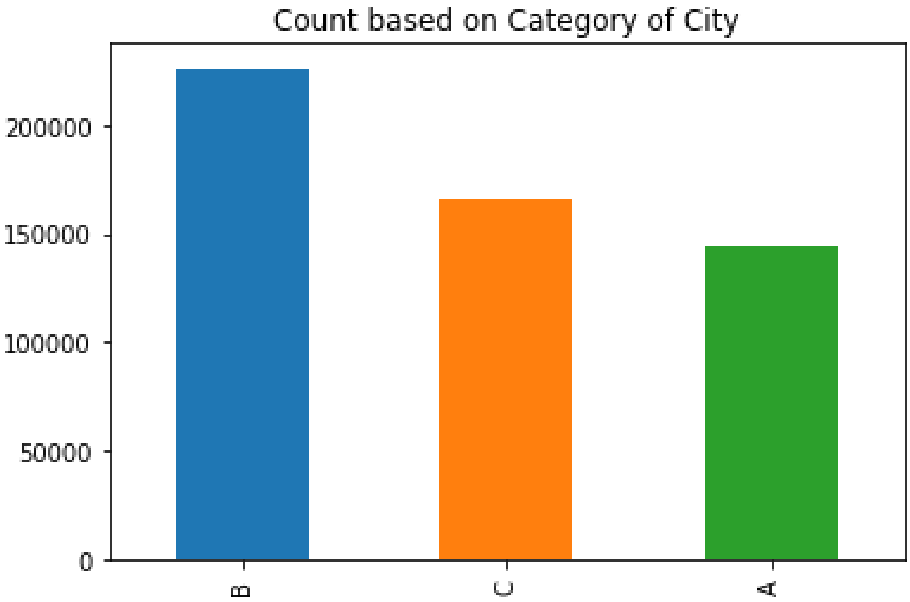


Figure 12: Type of City vs Count
This graph shows that there are 3 categories of cities, ’A’,’B’, and ’C’ and we can observe that ’B’ has a lot of people shopping.

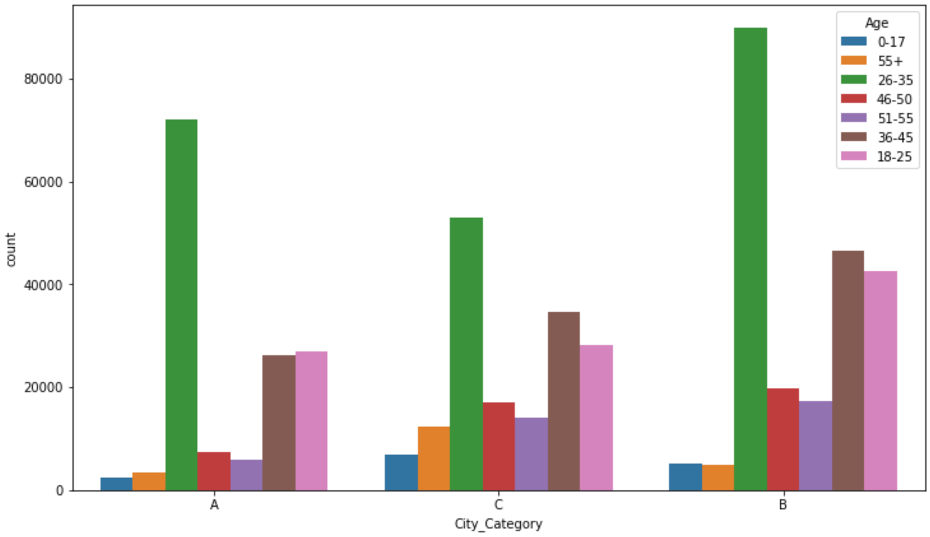


Figure 13: Type of City vs Count
This shows the type of city and the different age groups that shop in these cities. Each age group has different colors.

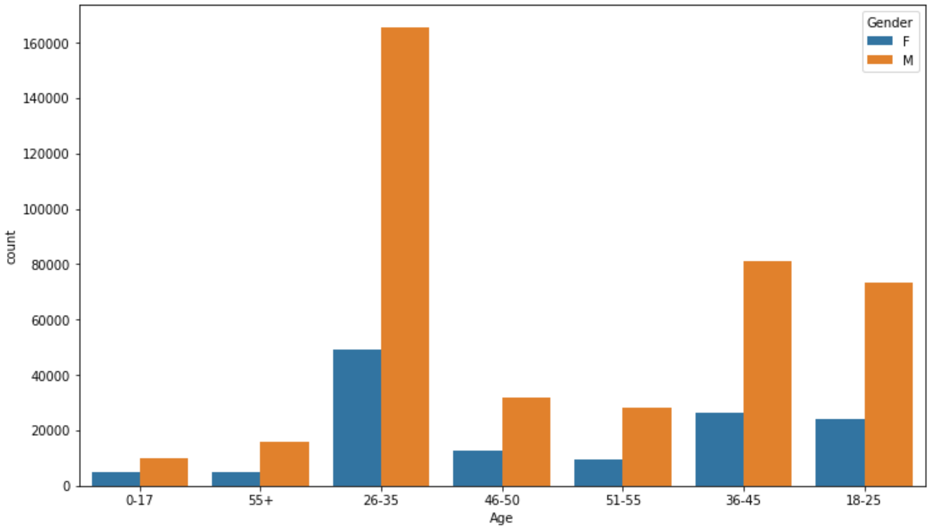
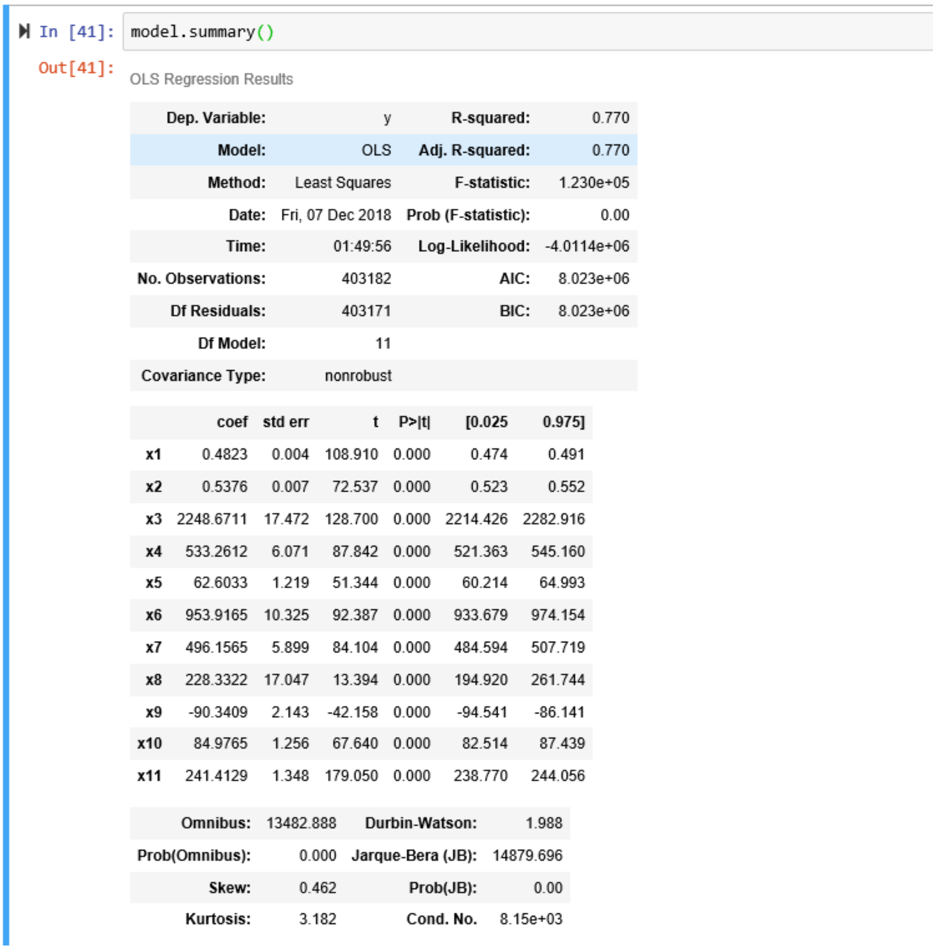


Figure 14: Age vs Count
This shows the count of people spending in each age group and we grouped them by marital status. Females are denoted by blue bars and males are denoted by Orange bars

In this section we have done exploratory data analysis on whole dataset. We found lot of interesting observations in the data.

- Linear Regression (Ordinary Least Square):

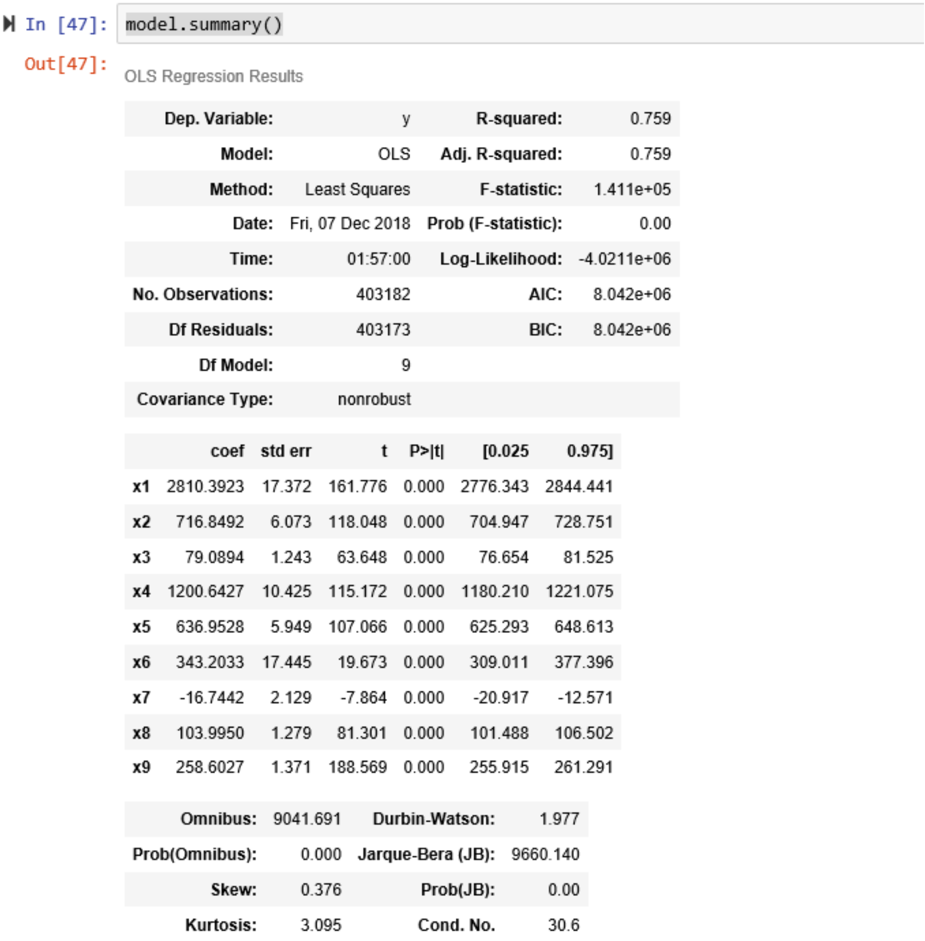


```
In [41]: model.summary()
```

```
Out[41]:
OLS Regression Results

    Dep. Variable:             y         R-squared:       0.770
           Model:            OLS    Adj. R-squared:       0.770
          Method:  Least Squares       F-statistic:   1.230e+05
            Date: Fri, 07 Dec 2018 Prob (F-statistic):     0.00
            Time:       01:49:56    Log-Likelihood: -4.0114e+06
No. Observations:         403182               AIC:   8.023e+06
    Df Residuals:         403171               BIC:   8.023e+06
        Df Model:             11
 Covariance Type:      nonrobust

          coef  std err        t  P>|t|    [0.025    0.975]
 x1     0.4823    0.004  108.910  0.000     0.474     0.491
 x2     0.5376    0.007   72.537  0.000     0.523     0.552
 x3  2248.6711   17.472  128.700  0.000  2214.426  2282.916
 x4   533.2612    6.071   87.842  0.000   521.363   545.160
 x5    62.6033    1.219   51.344  0.000    60.214    64.993
 x6   953.9165   10.325   92.387  0.000   933.679   974.154
 x7   496.1565    5.899   84.104  0.000   484.594   507.719
 x8   228.3322   17.047   13.394  0.000   194.920   261.744
 x9   -90.3409    2.143  -42.158  0.000   -94.541   -86.141
x10    84.9765    1.256   67.640  0.000    82.514    87.439
x11   241.4129    1.348  179.050  0.000   238.770   244.056

      Omnibus: 13482.888    Durbin-Watson:     1.988
Prob(Omnibus):     0.000 Jarque-Bera (JB): 14879.696
         Skew:     0.462         Prob(JB):      0.00
     Kurtosis:     3.182         Cond. No.  8.15e+03
```

Figure 15: Model Summary of OLS Linear Regression using all variables

Here we can observe the R-squared value = 0.770. All the coefficients are really high which means no variable is significant to predict Purchase.



```
In [47]: model.summary()
```

Out[47]:

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | y | **R-squared:** | 0.759 |
| **Model:** | OLS | **Adj. R-squared:** | 0.759 |
| **Method:** | Least Squares | **F-statistic:** | 1.411e+05 |
| **Date:** | Fri, 07 Dec 2018 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 01:57:00 | **Log-Likelihood:** | -4.0211e+06 |
| **No. Observations:** | 403182 | **AIC:** | 8.042e+06 |
| **Df Residuals:** | 403173 | **BIC:** | 8.042e+06 |
| **Df Model:** | 9 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **x1** | 2810.3923 | 17.372 | 161.776 | 0.000 | 2776.343 | 2844.441 |
| **x2** | 716.8492 | 6.073 | 118.048 | 0.000 | 704.947 | 728.751 |
| **x3** | 79.0894 | 1.243 | 63.648 | 0.000 | 76.654 | 81.525 |
| **x4** | 1200.6427 | 10.425 | 115.172 | 0.000 | 1180.210 | 1221.075 |
| **x5** | 636.9528 | 5.949 | 107.066 | 0.000 | 625.293 | 648.613 |
| **x6** | 343.2033 | 17.445 | 19.673 | 0.000 | 309.011 | 377.396 |
| **x7** | -16.7442 | 2.129 | -7.864 | 0.000 | -20.917 | -12.571 |
| **x8** | 103.9950 | 1.279 | 81.301 | 0.000 | 101.488 | 106.502 |
| **x9** | 258.6027 | 1.371 | 188.569 | 0.000 | 255.915 | 261.291 |

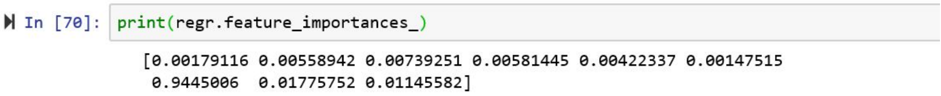
| | | | |
|---|---|---|---|
| **Omnibus:** | 9041.691 | **Durbin-Watson:** | 1.977 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 9660.140 |
| **Skew:** | 0.376 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 3.095 | **Cond. No.** | 30.6 |

Figure 16: Model Summary of OLS Linear Regression using low-level variables

Here we can observe the R-squared value = 0.759. We can observe decrease in the R-squared value compared to previous model.

- Decision Tree:
  Decision Tree model is applied on twice, one using all variables to predict purchase and other, only using the low-level variables. For the implementation of this model we have used maximum depth of the tree 11 which, gave the better RMSE value compared to other depths.

- Random Forest:
  Similar to the previous two models we have applied Random forest model on the dataset twice. We have implemented this model using maximum depth, 12 and number of estimators, 100. Random Forest model gave slightly better results than Decision tree model. We have calculate feature importance using this model. Those are shown in the below figure:



```
In [70]: print(regr.feature_importances_)

         [0.00179116 0.00558942 0.00739251 0.00581445 0.00422337 0.00147515
          0.9445006  0.01775752 0.01145582]
```

Figure 17: Calculated feature importance

From the above output we can observe that Product category 1 variable has higher importance in predicting the Purchase value.

- Neural Network:

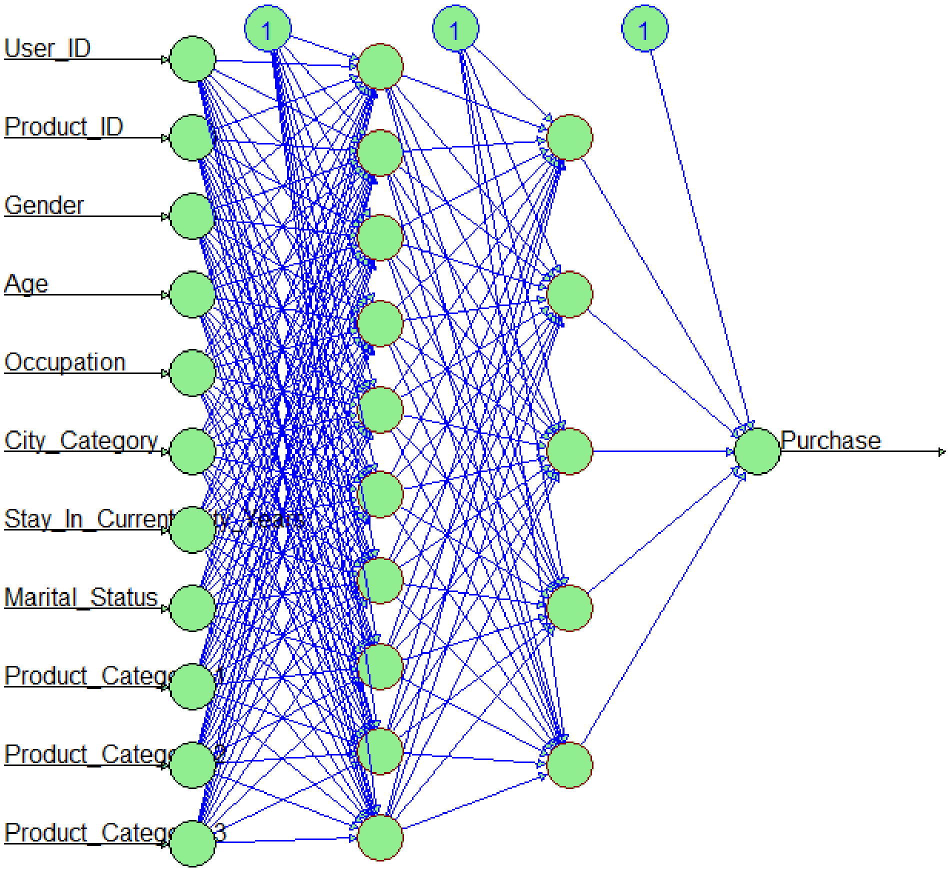


Figure 18: Sample Neural Network
This image is the Neural Network of our dataset with 2 hidden layers with 10 and 5 neurons.

To implement Neural Network model we have used 3 hidden layers with 100, 50 and 20 neurons respectively.

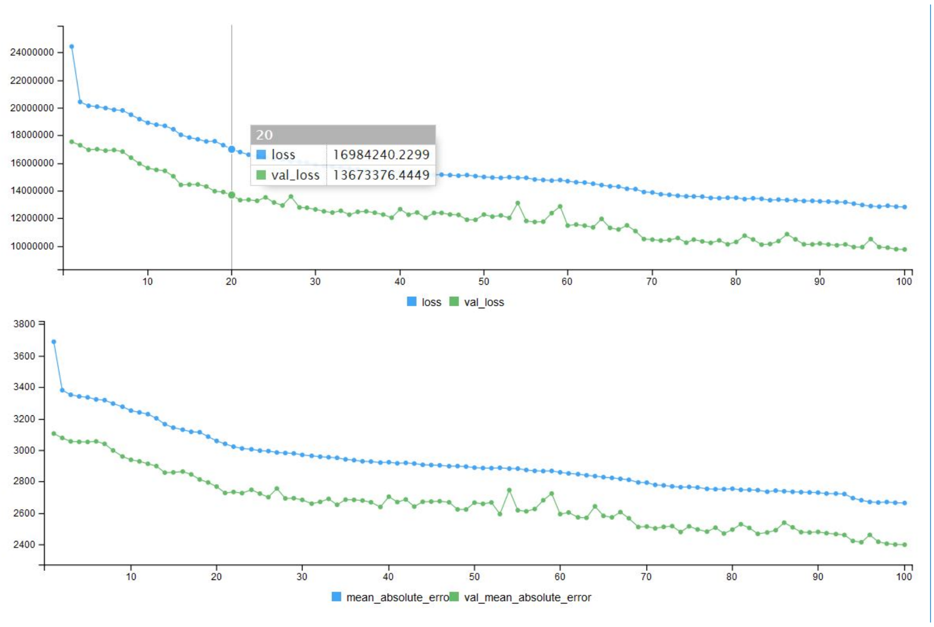
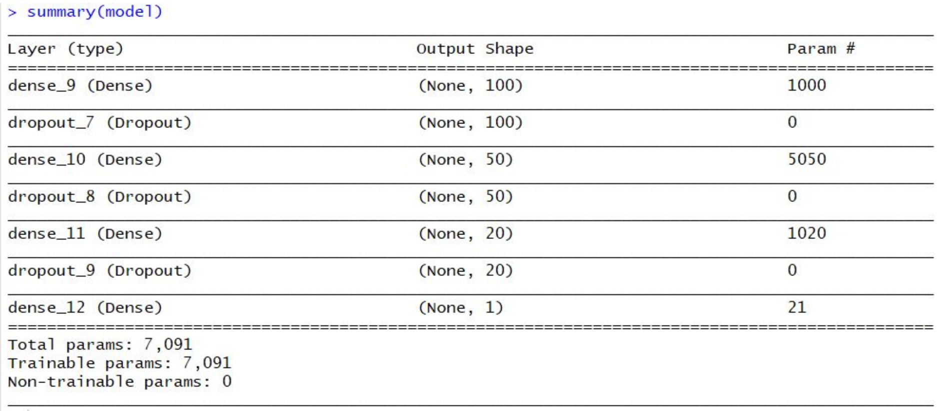


Figure 19: Error loss and Mean Absolute Error
Since the data is huge We have taken 20 percent of data sample to plot this graph. In this graph error loss and Mean Absolute Error are plotted along with their sample loss and sample Mean Absolute Error



```
> summary(model)
______________________________________________________________________________________________
Layer (type)                              Output Shape                           Param #
==============================================================================================
dense_9 (Dense)                           (None, 100)                            1000
______________________________________________________________________________________________
dropout_7 (Dropout)                       (None, 100)                            0
______________________________________________________________________________________________
dense_10 (Dense)                          (None, 50)                             5050
______________________________________________________________________________________________
dropout_8 (Dropout)                       (None, 50)                             0
______________________________________________________________________________________________
dense_11 (Dense)                          (None, 20)                             1020
______________________________________________________________________________________________
dropout_9 (Dropout)                       (None, 20)                             0
______________________________________________________________________________________________
dense_12 (Dense)                          (None, 1)                              21
==============================================================================================
Total params: 7,091
Trainable params: 7,091
Non-trainable params: 0
______________________________________________________________________________________________
```

Figure 20: Summary of Neural Network model

- Comparison of RMSE values for all models:

| | Using all Variables | Using only low-level variables |
|---|---|---|
| Linear Regression | 25731316.365 | 26986687.830 |
| Decision Tree | 7934321.792 | 8870391.228 |
| Random Forest | 7635218.020 | 8677330.898 |
| Neural Network | 4332.358142 | 4349.638378 |

RMSE values of models for full dataset

Above given table is the RMSE values of the 4 Models on whole dataset implemented using all available variables and using only low-level Variables. From the table we can observe that RMSE values are very high for this dataset in all models. Among all the models Neural Networks RMSE values are significantly less compared reamining 3 models. We expected RMSE value of the model using only low-level varaiables would be higher than the value of the model using all the variables. But, surprisingly values of models build using only low level variables are less compared to others.
As mentioned in the exploratory data analysis part, customer of Age group 26-35 are more in this dataset. So we used data of only customers with Age from 26-35. Results for that data are as follows:

| | Using all Variables | Using only low-level variables |
|---|---|---|
| Linear Regression | 21533833.553 | 21619301.481 |
| Decision Tree | 8318541.705 | 9191738.517 |

RMSE values of models for Age group 26-35

We was expecting better results for this data then the full data. But, both the values are almost similar.

# 8 Conclusion

We have explored Black Friday dataset. We found out some interesting insights in exploratory data analysis part of this project. We have implemented Linear regression, decision tree, random forest and neural networks on the whole data set and data of customers with age from 26 to 35. Results of this models are disappointing. However, results seems reasonable for this synthetically generated data. Predictor variables are not significant to predict target variable.

# 9 References

1. https://www.hausmanmarketingletter.com/predictive-analytics/
2. https://www.practicalecommerce.com/predictive-analysis
3. https://www.dummies.com/business/customers/the-benefits-of-customer-analytics/
4. https://www.kaggle.com/mehdidag/black-friday

5. https://becominghuman.ai/predicting-buying-behavior-using-machine-learning-a-case-study-on-sales-prospecting-part-i-3bf455486e5d

6. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

7. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

8. https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLS.html

9. https://www.statsmodels.org/dev/examples/notebooks/generated/ols.html

10. https://www.statsmodels.org/stable/regression.html

11. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

12. https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html

13. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

14. https://www.analyticsvidhya.com/blog/2017/09/creating-visualizing-neural-network-in-r/

## 10 Reference for Figures

Figure2:https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a
Figure3:https://en.wikipedia.org/wiki/Decision_tree_learning
Figure4:https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd
Figure5:https://skymind.ai/wiki/neural-network
Figure6: https://skymind.ai/wiki/neural-network