**Applied Data Analytics II -Course Project**

Diallo Youssouf

Date completed: 05/04/2019

2019

1. **Research statement**

In this Course Project; I intend to apply the knowledges that I acquired during the semester to demonstrate a clear understanding of Data Analytics’ concept. I will be using three techniques that are very well known in the Data science: Classification, Clustering, and Regression Analysis. I will be studying customers behaviors in a Black Friday sale. From the dataset, I will try to answer the following questions:

1- Predict the amount purchase by a client using Regression Analysis,

2- Identify which Age group tend to buy a lot during a Black Friday sale using Classification

3- Which city is more productive

4- Segment customers based on their spending using Cluster analysis

We argue that by answering these questions, it will help the retail store to understand their businesses and make data driven decision.

1. **Dataset description**

The dataset that I am using is from Kaggle and it can be found at this link

<https://www.kaggle.com/mehdidag/black-friday/version/1> once you click on the link, you will need to go under download button to download. The dataset contains **12 columns** with numerical and categorical data types, and **537,577** observations. The dataset is in CSV format about Black Friday sales in a retail store. The Dataset has a column called **USER\_ID** which help identify user who made a purchase from the store.

**Product\_ID**: Help identify a product in the store

**Gender:** Define the user gender which will be either Male (M) or Female(F)

**Age:** Define the age of user, which is given in a range of values **[0-17[, [18-25[, [26-35[, [36-45[,**

**[46-50[, and 55+.**

**Occupation:** Define the occupation of each customer

**City\_Category:** The store is implanted in tree different city **(A, B, and C)**

**Stay\_In\_Current\_City\_Years:** Define how many years, a customer resides in that city

**Marital\_Status:** Define the customer's status, which is either married coded by 1 or not married coded by 0. The retail store is selling tree categories of products

**Product\_Category\_1**: Define the number of items purchased within the Product\_Category\_1

**Product\_Category\_2:** Define the number of items purchased within the Product\_Category\_2

**Product\_Category\_3:** Define the number of items purchased within the Product\_Category\_3

**Purchase:** Define the amount spent by each customer to purchase the three products. The amount purchased is in US dollars.

1. **Applied Data Analysis**
2. **Data exploration, analysis and visualization**

**1.a Data exploration**

The First task is to import the libraries require for our analysis. After downloading the dataset from Kaggle, I renamed the dataset as Project\_Diallo\_Youssouf\_Dataset.csv. In any Data analytics problem, we need to load the dataset into a pandas’ Data Frame first. Then, in the process of exploration, we need to call some functions to our pandas’ object such as shape (), info (), head (), tail (), and is null (). The result of shape () gives us the number of rows and columns in the dataset. We have 537577 rows and 12 columns. Calling info () to the dataset gives us the following:

Data columns (total 12 columns):

User\_ID 537577 non-null int64

Product\_ID 537577 non-null object

Gender 537577 non-null object

Age 537577 non-null object

Occupation 537577 non-null int64

City\_Category 537577 non-null object

Stay\_In\_Current\_City\_Years 537577 non-null object

Marital\_Status 537577 non-null int64

Product\_Category\_1 537577 non-null int64

Product\_Category\_2 **370591** non-null float64

Product\_Category\_3 **164278** non-null float64

Purchase 537577 non-null int64

**dtypes:** float64(2), int64(5), object (5)

This is an important step in Data Analysis, in order to work with the data, we must know first what types of data we are dealing with. We have 3 data types: Float, Int, and Object Product\_Category\_2 and Product\_Categroy\_3 are float Marital\_Status, Purchase, Product\_Catergory\_1, User\_ID, and Occupation are numerical. Product\_ID, Age, stay\_In\_Current\_City\_Years, City\_Category, Gender are object.

Head () and tail () allow us to inspect the first 5 elements and last 5 element of the dataset respectively. This process is very useful, in the sense it helps us know if the dataset still has the same structure.

We also check if dataset has missing values; the answer is yes. Among the 12 features, only Product\_Category\_2 and Product\_Category\_3 have NaN values. We will explain, how to handle to missing values in the next coming paragraph.

**1.b Data analysis**

Let’s do some analysis to better under our sales, we will ask some questions such as **what was the most profitable product on Black Friday sales?**

**A screenshot of a cell phone

Description automatically generated**

We group all the sales by Product\_ID, then we sum all the amount purchase for that product and just pick the first 5 element. From the result above the product with ID **(P00025442)** was the most profitable on Black Friday the purchase amount was $ 27,532 426 something above $ 27 million.

**What was the most profitable City on Black Friday?**

With two lines of codes, we grouped by City\_Category and add all the amount purchased by each city

City\_Category

A $ 1295668797

B $ 2083431612

C $ 1638567969

We clearly see from the result above that the city category B was the most profitable on Black Friday the purchase amount was $ 2,083 431 612 something above 2 Billion dollars.

Determine between men and women, which category made a lot of purchase.

The result from our analysis shows that the retail company recorded 405380 transactions from Males

which represent 75.40 % and 132197 transactions which also represent 24.6 % from Females.

We noticed that, among all customers who made purchase on Black Friday sales, 59 % of them are married and 41% are unmarried.

Now, let’s do some descriptive analysis

A screenshot of a cell phone

Description automatically generated

This result above shows us that customers spent minimum **$185** on Black Friday sales and maximum

**$ 23,961** on sales.

Customers spent on average **$ 9,333** on Black Friday sales. We also found out that every customer who came to the store on Black Friday purchased the **Product\_category\_1**, and the **Product\_Category\_3** was the least sold. But it is not clear to me why it was the least sold. It might be that the store did not have a lot of quantity for Product\_Categroy\_3 and it was sold out early or something else. Customers who purchased during Black Friday sales had different occupation which is coded between 0 to 20. I do not know what exactly the number of occupations really represent. The Martial status is 0 for unmarried and 1 for married.

**1.c Data visualization**

Let’s plot the distribution of purchase and its violin plot

A screenshot of a social media post

Description automatically generated

Plotting the distribution will give us an idea, how the data point is spread out and what model we should apply.

A screenshot of a social media post

Description automatically generated

The violin plot is used to visualize the distribution of the data and its probability density. We can see from this plot that a large portion of amount purchase is under 10,000.

1. **Data preparation**

I did some data preprocessing on the dataset, there was a lot of NaN values for Product\_Category\_2 and Product\_Category\_3 columns. For example, if someone did not purchase a specific Product Category, no value is being enter for that. Because Product\_Category\_2 and Product\_Category\_3 represent the number of product purchase by a customer. I find very useful to replace the NAN values by 0, instead deleting the entire columns. Another data transformation that I did is to change the types of Product\_Category\_2 and Product\_Category\_3 from float to int. I wanted to keep everything with same data types as Product\_Catgory\_1 which is int. It makes more sense because someone cannot purchase a half item. The data type of the column Stay\_In\_Current\_City\_Years is int which represent how long a person resides in a specific City\_Category, but I had to remove the (+) sign that is associated with those who resides in the City over 4 years. If I did not remove the (+) sign, I will get an error when training and testing the model later.

1. **Algorithm #1 Linear Regression**

The first algorithm that I am using is multiple linear regression. Let’s define what is multiple regression.

A regression model that involves two or more independent variable is called multiple regression. Regression analysis is a tool for building mathematical and statistical models that characterize relationship between a dependent variable which must be a numerical variable and one or more independent variable. In any regression problem we have two things: Independent variables (X) and dependent variables(Y).

The research question that I will answer is to predict the purchase amount for a customer on a Black Friday sale.

Y = Purchase (called target data in python, and referred to as the dependent variable or response variable)

X = independent variables, or explanatory variables (Occupation, Marital Status, Stay\_In\_Current\_City\_Years, Product\_category1, Product\_Category\_2 and Product\_Category\_3). Both the independent and dependent variables are continuous. We will use the linear regression model to fit and predict the purchase amount. We will use the R squared method to estimate the accuracy of the model. We build and train the model with more than half million records. We reserve 70 % for training and 30% of testing. After using the linear Regression model on Scikit Learn, we compute the accuracy of the model and found out the model is very terrible 13 % accuracy, which means only 13 % purchase amount can be explained by the chosen independent variables. Let’s plot the graph to visualize the result of the model.

A screenshot of a cell phone

Description automatically generated

We can see from the plot that the model performs very poorly. Let’s try some technique to see if we can improve the model. For that I will be using XGBoost.

XGBoost is one of the popular machine learning algorithms these days. “Regardless of the type of prediction task at hand; regression or classification” [2] (Data Camp). Extreme Gradient Boosting “belongs to a family of boosting algorithm and uses the gradient boosting algorithm framework at its core” (Data camp). Let us talk about what is booting and how it works? Boosting is a sequential technique uses in Machine Learning that convert a weak learner to a strong one. “It combines a set of weak learners and delivers improved prediction accuracy. At any instant t, the model outcomes are weighed based on the outcomes of previous instant t-1” (Data camp) [ 2]. XGBoost has an in-built routine to handle missing values.

The scalability of the algorithm is made possible because it uses a novel tree learning algorithm for handling sparse data, and parallel and distributed computing to make learning faster. It uses an approximation of the exact greedy algorithm to find the best possible split for the tree. It is an ensemble learning method, which is computationally fast compared to other gradient boosting algorithms (Brownlee, Jason) [1]. It is also excellent when used on structured or tabular datasets for classification or prediction problems [1].

We will be using the same input and output for XGboost as the linear regression. We build the model using XGBRegressor and predict the model with the test data. Surprisingly the model performs very well. After computing the explain variance score, we observe that XGboost has improved the model and the accuracy moves from **0.13 % to 0.459 %**. Let’s plot the regression line

A close up of text on a white background

Description automatically generated

We see from the graph above that the data points are clustered around the regression line.

To conclude, we use linear regression to predict the purchase amount on Black Friday sales, the model performs very poorly, and we improve the accuracy of the model by using Extreme Gradient Boosting regressor (XGBoost).

1. **Algorithm #2 Decision trees for classification**

Decision trees for classification also simply know as classification trees. At a very general level, the classification tree algorithm seeks to split data sets into homogenous groups, ultimately identifying and creating groups that have similar variables to one another. It does so by analyzing the data set and using the analysis to create a set of rules, breaking down the dataset into smaller and smaller subsets. The creation of these rules is not governed by a human being and are determined by the algorithm itself. The variables which go into the classification can be numerical or categorical, although the output of the classification tree will always be a discrete category (class).

We are doing two classifications:

First, we will classify the age of customers who made purchase on Black Friday sales.

In the describe statistics, we calculate the total purchase made by each age group.

A screenshot of a cell phone

Description automatically generated

We see from the result above that customers whose age group is between [26-35[ are those who did purchase a lot, followed by the [36-45[, [18-25[, [46-50[, [51-55[, 55+ and [0-17[. Knowing which age group purchased a lot will help the retail store to focus on brining products and target the specific age that they know will buy it.

The output of the classifications is age which is categorical variable. This is the possible values of age **([0-17[, [18-25[, [26,35[, [36-45[, [46-50[, [51-55[, and 55+)**

The inputs of the model are: Occupation, Stay\_In\_Current\_City\_Years, Marital\_Status, Purchase, Product\_Category\_1, Product\_Category\_2, and Product\_Category\_3. The input is continuous. We use Decision tree classifier from sklearn to build, train and test the model. We evaluate the model by using accuracy score from metrics. We find that the accuracy of the decision tree classifier is 41.68%.

Let ‘s see how we can improve the model. We will add some parameter such that criterion="entropy", and max\_depth=8. When we run again the model, we see that the model improves, and the accuracy of the model is now 50.6 % a bit of improvement.

The second model that we build is to classify the City Category variable which represent three different cities where the retail company was selling their products on Black Friday. We have three values for city. City A, B and C.

We will be using the same input as the previous classification, the only different here is the target variable. We build, train and test the model and we find that the accuracy is 42.23 %.

We are going to see if we can improve the model by using Extreme Gradient Boosting (XGboost) Classifier. We are going to use Label encoder to transform the categorical variable into a numerical variable because Extreme Gradient Boosting (XGboost) does not accept categorical variable. We will be using the same input, and now our output is discrete. We build, train and test the model using XGboost classifier. We find out that the model did improve slightly to 44.88 %. We think that if we use XGboost classifier the model will improve by a lot. However, it did not happen.

Now, we are going to try different techniques which is to use K-Fold cross Validation.

What is K-Fold Cross Validation how does it works? K -Fold Cross Validation is a technique used to evaluate models on a limited data. K-Fold Cross validation split the data into K number of sections or folds where each fold is used as testing set at some point. For instance, if k=3, the data set is split into 3 folds. Each iteration is used as testing and the rest are used to train the model. The process will continue until each fold has been used as the testing set [5]. In our example we set the number of folding to three (nfold=3) , num\_class =3 (which determine the number of class) and so on.

The metrics we use to evaluate the model is the Logarithmic Loss (mlogloss). The goal is to minimize this value. A perfect model would have a log loss of 0. Log loss increases as the predicted probability diverges from the actual label. From our result, we do not have mlogloss training and testing mean close to zero.

1. **Algorithm #3 Density Based Spatial Clustering of Applications with Noise (DBSCAN)**

Density-Based Spatial Clustering of Applications with Noise is a very popular unsupervised learning algorithm uses for data clustering. “It is a density-based clustering non-parametric algorithm” [3]. In other words, given a set of points, DBSCAN groups points that are very close together based on the density. Any point alone, or outside the density is consider an outlier. DBSCAN is affected by the presence of noise and outliner in the data. DBSCAN does not require to specify the number of clusters in the data a priori.

I use DBSCAN because I want to segment customer based on their spending, this is very useful for retail store, they can target customers who purchase a lot with some major discounts.

Before, we apply the data set on DBSCAN, I found that it is very interesting to try some techniques to see how the model behave. I apply Principal Component Analysis (PCA) which is a statistical procedure that uses to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components [6]. The reason why I am using this is because I want to see if the features that I are using are correlated, and if I can reduce the dimension in order to get something that I will be able to plot into two dimensions. I use columns like Product\_category\_1, Product\_category\_2 and Product\_Category\_3, Purchase as input of the model. My target variable is gender which is either Male (M) or Female (F). After building the PCA model and training it. We get the following graph.

A screenshot of a cell phone

Description automatically generated

The result from this graph clearly shows us that the classes are not well separated. That means it will be very hard to find a good pattern of clustering. Because the PCA did not give a clear separation of classes, I will try another algorithm called K-means.

K-means clustering is one of the simplest and popular unsupervised machine learning.

K-means “starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids” [7]. I build the K-means model using three clusters and the inputs were the same as the PCA. I find out that the model did not cluster well the data.

A close up of text on a white background

Description automatically generated

We can see from the graph above that we have 3 clusters, but the data points are not well clustered.

We will now try the last option which DBSCAN, I build and train the model with the same inputs as K-means. The only difference in K-Means, I did specify the number of clusters, however, with DBSCAN I did not. Here is the output of the model.

A close up of a logo

Description automatically generated

With DBSCAN, it is possible to determine the number of clusters and the noisy.

Estimated number of clusters: 195 and Estimated number of noise points: 176

We find that we have 195 clusters and 176 noises. From the result presented in the graph above that the data points are not well clustered. We evaluate DBSCAN model and we find the accuracy is 42.80 %.

To evaluate the model, we will use the silhouette score which is “calculated utilizing the mean intra- cluster distance between points, and the mean nearest-cluster distances” [4].

The silhouette score varies from -1 to 1. The score of -1 indicates a worse performance, and silhouette score of 1 signifies a best performance. Silhouette score of 0 indicates that there is an overlapping among the clusters.

1. **Selected Algorithm validation**

Throughout this process I apply many different algorithms, in each research questions, I use at least two algorithms to see how I can enhance the accuracy. But all my accuracy score is roughly below 50 %.

I find it is not useful to test again the algorithm with the validation set.

**IV: Conclusion**

Throughout this paper**,** I use linear regression to predict the purchase amount and I enhance the model by using XGboost Regressor. Then, I classify the age group and the city using Decision Tree Classifier, improver it with XGboost Classifier, and finally I run Principal Component Analysis, K-Means and DBSCAN to see if I can segment customers based on their spending. After trying different model and technique, I was able the predict the purchase amount, however, the classification and clustering did not perform well. Therefore, I cannot answer with clarity to those research questions.

For further research, I recommend looking at the classification and clustering to see if we can come up with an accurate response. I suggest looking into Deep learning. Excellent learning experience.

**V: Appendix**

[1] <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>

[2] <https://www.datacamp.com/community/tutorials/xgboost-in-python>

[3] <https://en.wikipedia.org/wiki/DBSCAN>

[4] <https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818>

[5] <https://medium.com/datadriveninvestor/k-fold-cross-validation-6b8518070833>

[6] <https://en.wikipedia.org/wiki/Principal_component_analysis>

[7] <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>