Black Friday Sales Prediction

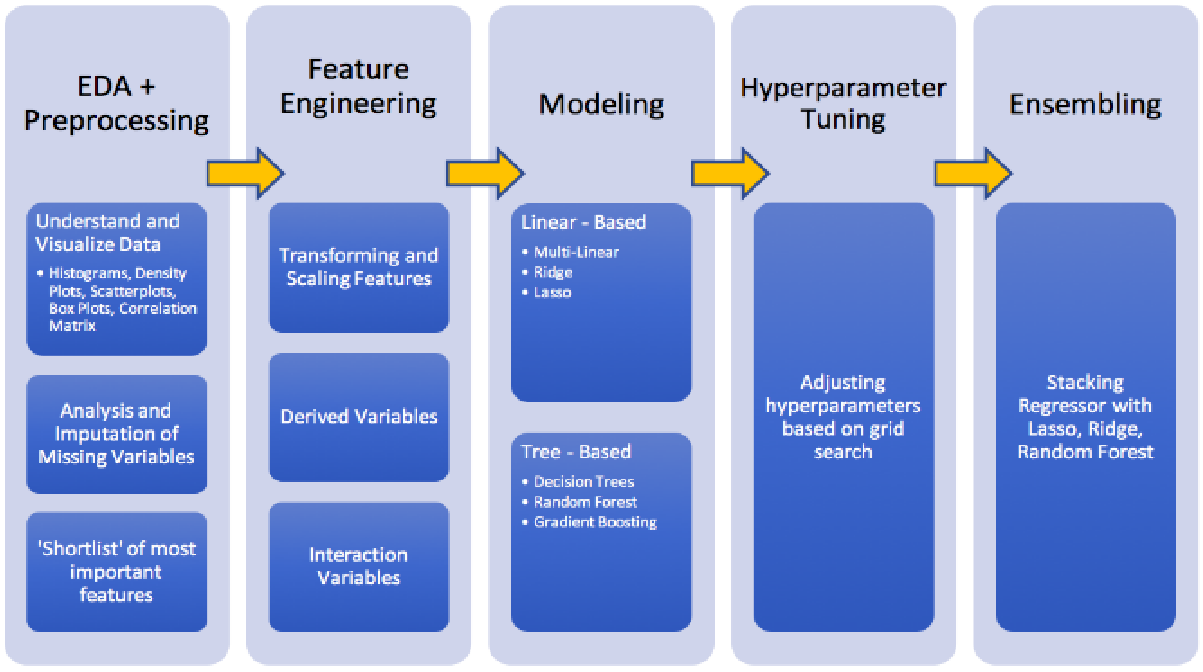
Introduction

A retail company "*Sadguru InfoTech pvt ltd*." wants to **understand the customer purchase behaviour (specifically, purchase amount) against various products of different categories.** They have shared purchase summary of various customers for a selected high volume products from last month.

They want to build a model to predict the purchase amount of customer against various products which will help them to create personalized offer for customers against different products.

Organisation of our analysis

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 five stages:

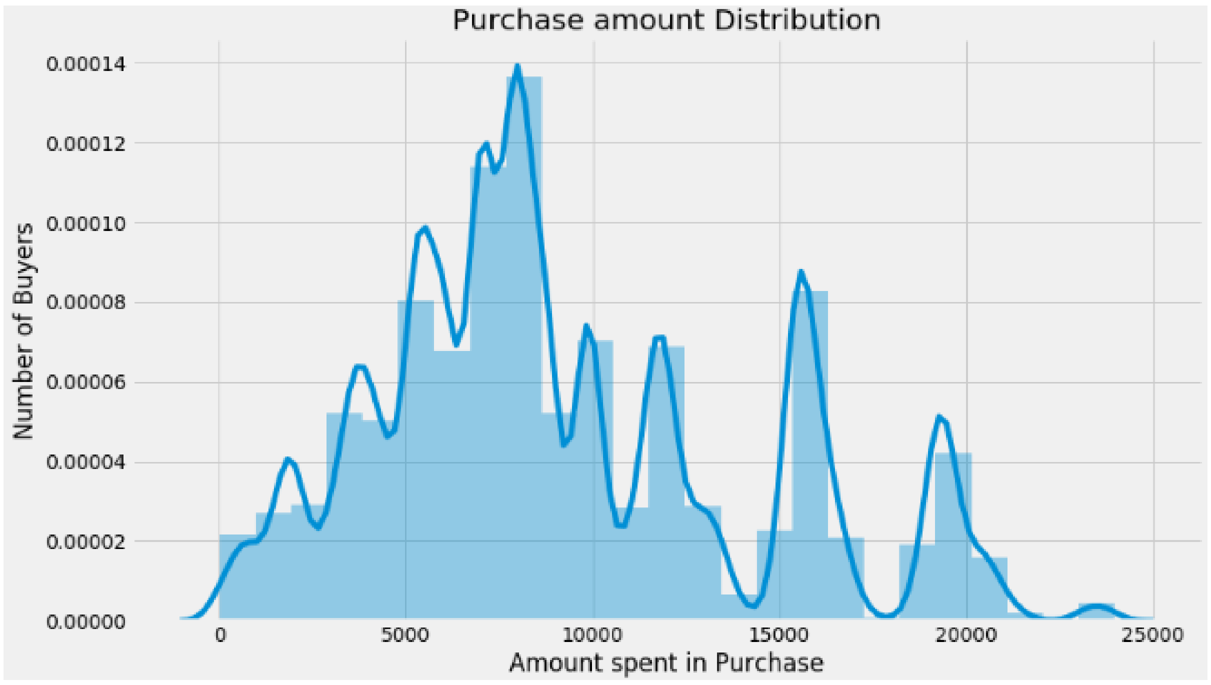

EDA + Preprocessing
Understand and Visualize Data
• Histograms, Density Plots, Scatterplots, Box Plots, Correlation Matrix
Analysis and Imputation of Missing Variables
'Shortlist' of most important features
Feature Engineering
Transforming and Scaling Features
Derived Variables
Interaction Variables
Modeling
Linear - Based
• Multi-Linear
• Ridge
• Lasso
Tree - Based
• Decision Trees
• Random Forest
• Gradient Boosting
Hyperparameter Tuning
Adjusting hyperparameters based on grid search
Ensembling
Stacking Regressor with Lasso, Ridge, Random Forest

1. Exploratory Data Analysis (EDA)

We've 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.**

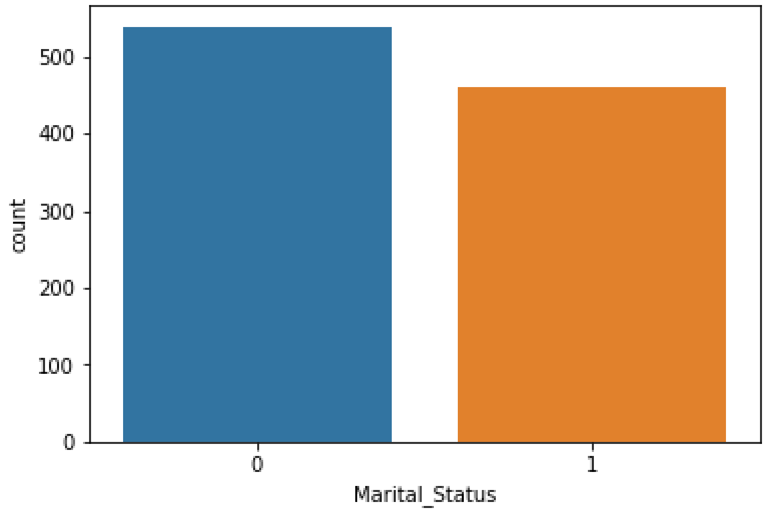
Univarate Analysis

To get an idea of the distribution of numerical variables, histograms are an excellent starting point. Let’s begin by generating one for Purchase, our target variable.



Distribution of the variable Marital Status

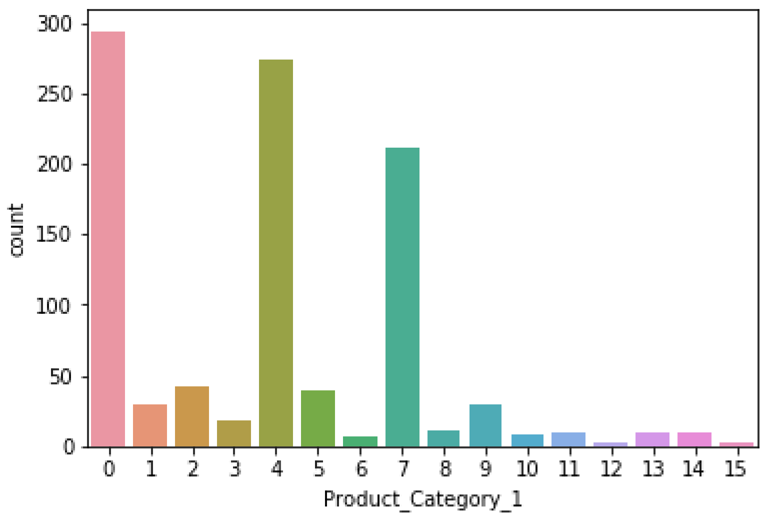
```
In [23]: seaborn.countplot(BlackFriday_frame.Marital_Status)

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x27b032f7f0>
```



Black Friday Sales Prediction

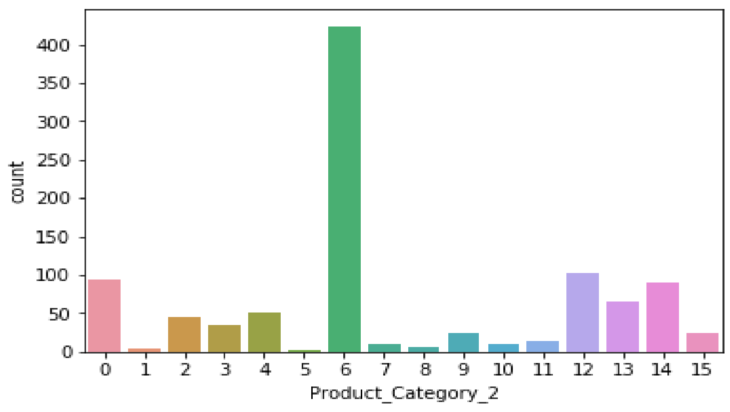
Distribution of the variable Product_Category_1

```
In [24]: seaborn.countplot(data.Product_Category_1)
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x27af5be908>
```



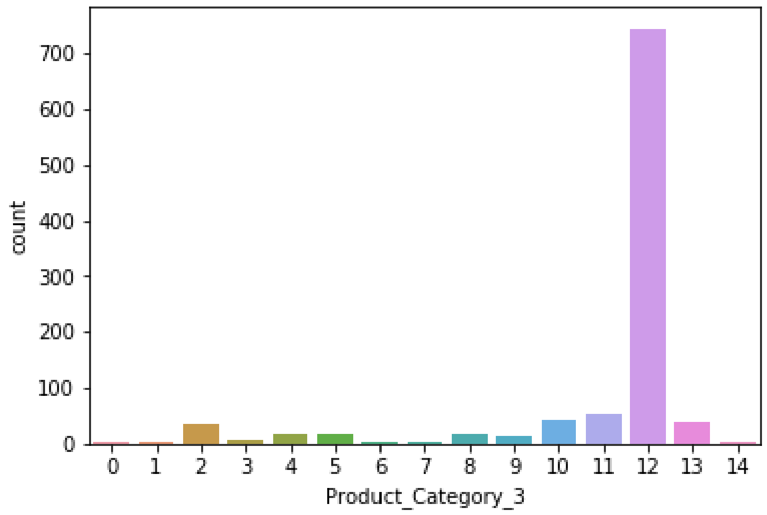
Distribution of the variable Product_Category_2

```
In [25]: seaborn.countplot(data.Product_Category_2)
Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x27a92553c8>
```



Distribution of the variable Product_Category_3

```
In [26]: seaborn.countplot(data.Product_Category_3)

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x27b031a9e8>
```

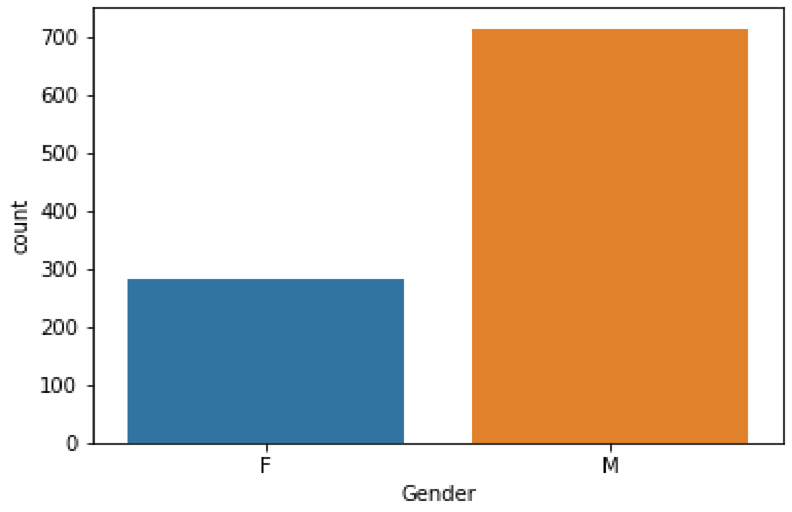


Correlation between Numerical Predictors and Target variable

```
In [27]: corr = data.corr()
         print (corr['Purchase'].sort_values(ascending=False)[:10], '\n')
         print (corr['Purchase'].sort_values(ascending=False)[-10:])

Purchase              1.000000
City_Category         0.075906
Gender                0.055498
Age                  -0.009903
Product_ID           -0.088289
Product_Category_2   -0.112573
Product_Category_3   -0.189807
Product_Category_1   -0.386427
Name: Purchase, dtype: float64

Purchase              1.000000
City_Category         0.075906
Gender                0.055498
Age                  -0.009903
Product_ID           -0.088289
Product_Category_2   -0.112573
Product_Category_3   -0.189807
Product_Category_1   -0.386427
Name: Purchase, dtype: float64
```

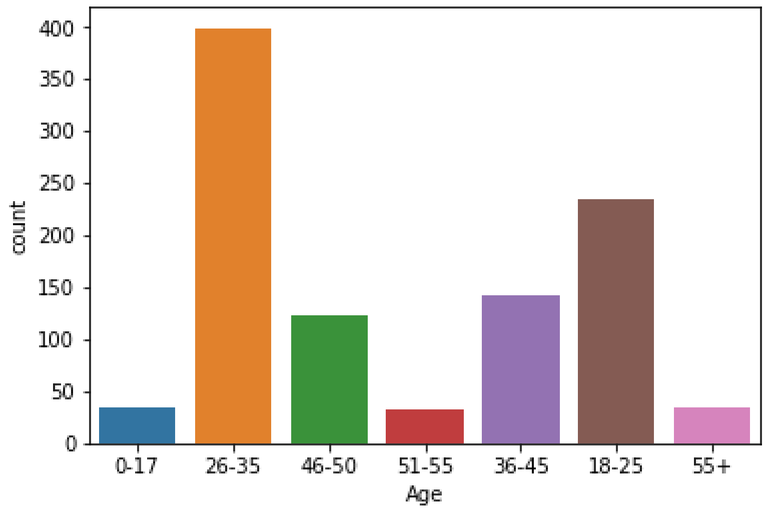
Distribution of the variable Gender

```
In [28]: seaborn.countplot(BlackFriday_frame.Gender)

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x27a939feb8>
```



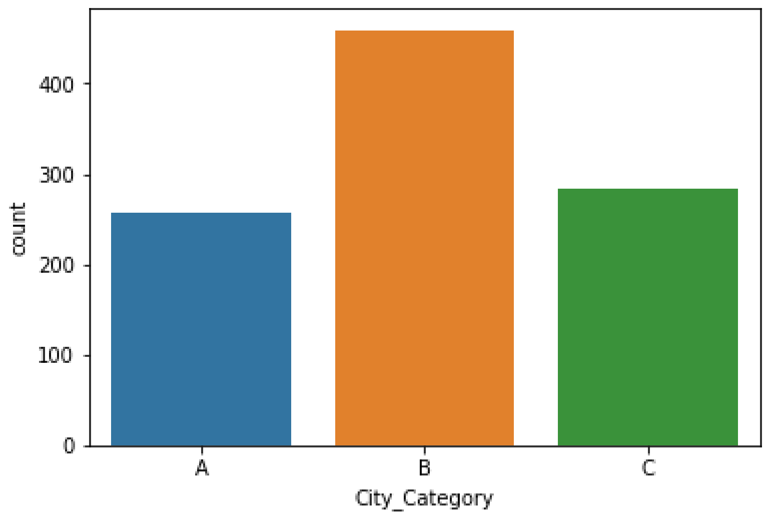
Distribution of the variable Age

```
In [29]: seaborn.countplot(BlackFriday_frame.Age)

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x27af5ff208>
```



Distribution of the variable City_Category

```
In [30]: seaborn.countplot(BlackFriday_frame.City_Category)
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x27af662128>
```



Black Friday Sales Prediction

Life Cycle of Data Science

Step 1: Define the Problem :

There is a company "Sadguru InfoTech pvt ltd" has given the data set so that we can predict sales ,so that company would take Business decision . We need to understand the problem what exactly customer or client want us to predict.

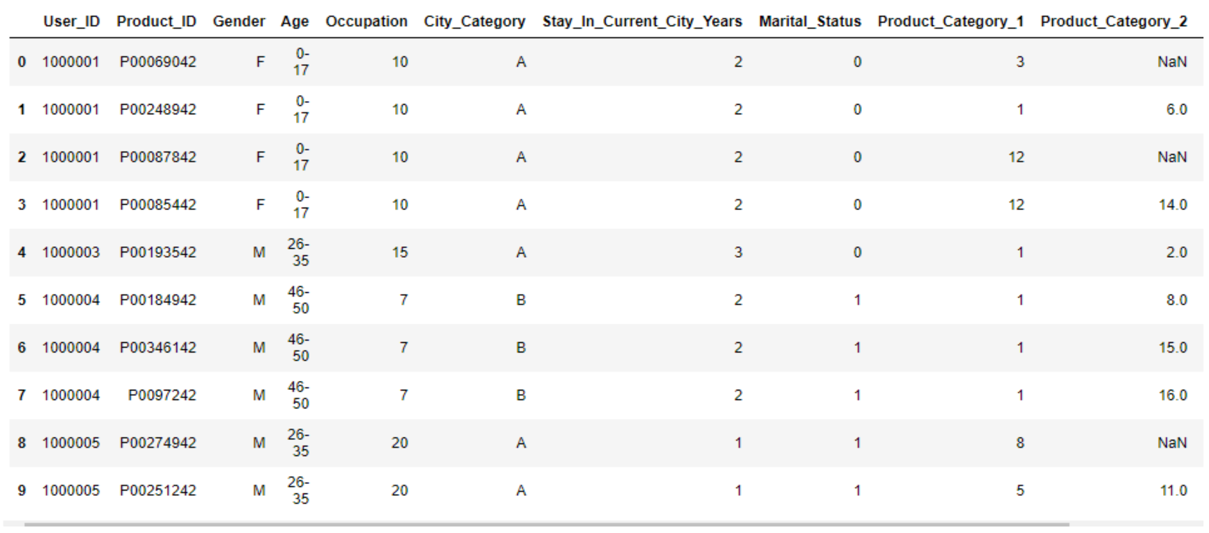
In this project the problem is **" to predict sales prediction "**

Step 2: Collecting data :

Once the problem is defined and identified its quite important for Data-Scientist or Data-Analyst to collect the required data which needed for prediction (Datasets could be provided from client side and based on that data set only Data- Scientist need to do prediction.)

BlackFriday_frame = pandas.read_csv('F:\BlackFriday.csv')

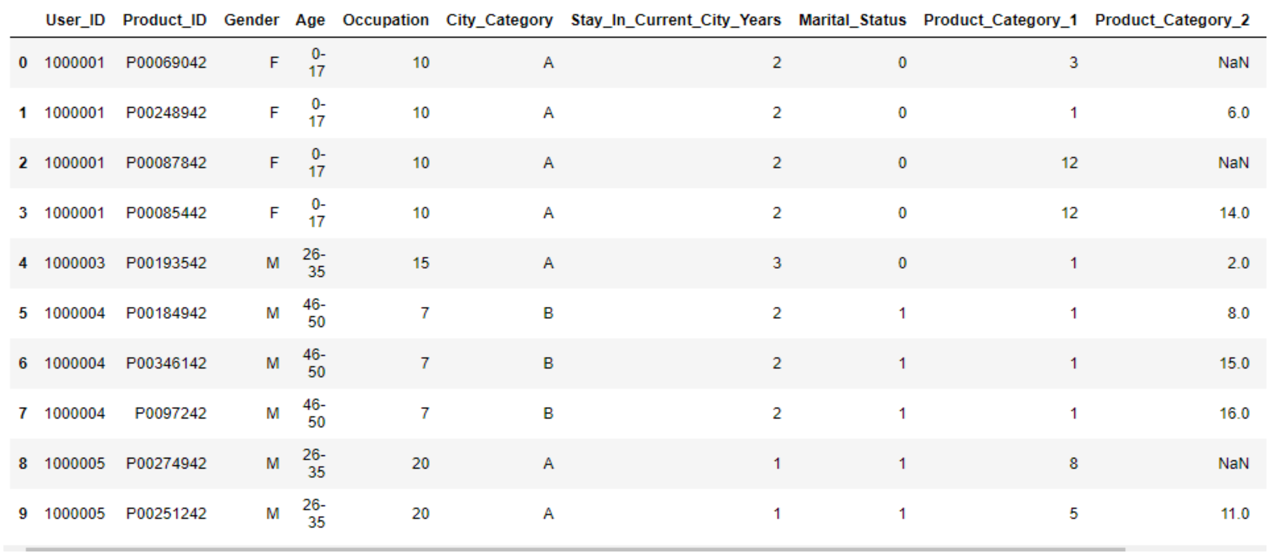
BlackFriday_frame.head(10)



	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0
4	1000003	P00193542	M	26-35	15	A	3	0	1	2.0
5	1000004	P00184942	M	46-50	7	B	2	1	1	8.0
6	1000004	P00346142	M	46-50	7	B	2	1	1	15.0
7	1000004	P0097242	M	46-50	7	B	2	1	1	16.0
8	1000005	P00274942	M	26-35	20	A	1	1	8	NaN
9	1000005	P00251242	M	26-35	20	A	1	1	5	11.0

Step 3: Prepare the data :

Client provides huge amount of data and Data Scientist need to understand what and data are required for prediction and only that data will be used for prediction. In this Data-Scientist basically understand the data and its types.



	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0
4	1000003	P00193542	M	26-35	15	A	3	0	1	2.0
5	1000004	P00184942	M	46-50	7	B	2	1	1	8.0
6	1000004	P00346142	M	46-50	7	B	2	1	1	15.0
7	1000004	P0097242	M	46-50	7	B	2	1	1	16.0
8	1000005	P00274942	M	26-35	20	A	1	1	8	NaN
9	1000005	P00251242	M	26-35	20	A	1	1	5	11.0

In the above image we can see that data set contains both categorical and numerical data and also NaN values which we need to apply few statistics and mathematics concepts.

Step 4: Split data in Training and Testing

Once the data is easily understood by Data scientist then he need to split those data sets in Training and Testing data .

But before splitting those in Training and Testing data we must clean those data because won't understand high dimensional data and we must clean and convert into low dimensional data so that machine could understand and do prediction and generate output

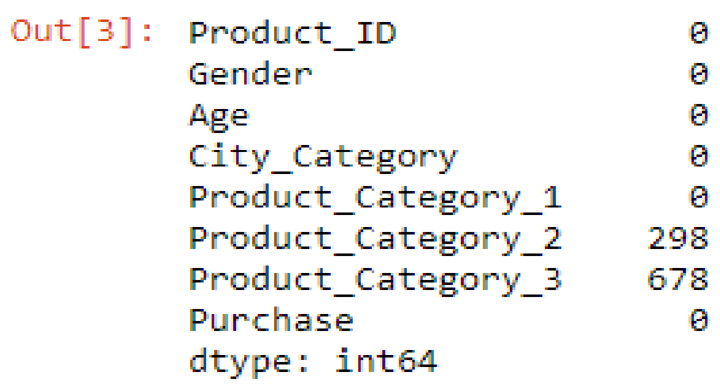
We can see that product id(high dimensional data)must be converted into low dimensional data ,gender(categorical data),age(range of values) which must be converted into numerical value ,product_category_2 consist **NaN** values which we need to take care.

We will also check how many values contain NaN values and based on that we will decide what statistics we need to apply on that.

We also need to remove all unwanted columns which are not required.

data=BlackFriday_frame.drop(['User_ID','Occupation','Stay_In_Current_City_Years','Marital_Status'],axis = 1)

data.isnull().sum()



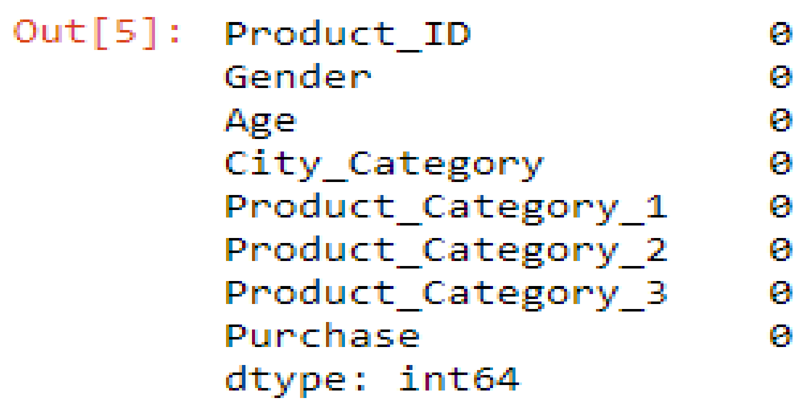
```
Out[3]: Product_ID               0
        Gender                   0
        Age                      0
        City_Category            0
        Product_Category_1       0
        Product_Category_2     298
        Product_Category_3     678
        Purchase                 0
        dtype: int64
```

In this project To fill NaN values we applied

for i in ['Product_Category_2','Product_Category_3']:

exec("data.%s.fillna(data.%s.value_counts().idxmax(), inplace=True)" %(i,i))

data.isnull().sum()



```
Out[5]: Product_ID            0
        Gender                0
        Age                   0
        City_Category         0
        Product_Category_1    0
        Product_Category_2    0
        Product_Category_3    0
        Purchase              0
        dtype: int64
```

Once the NaN values is filled we need to split the data in Training and Testing

Bfriday_data_x=data[['Product_ID','Gender','City_Category','Product_Category_1','Product_Category_2','Product_Category_3']]

```
Bfriday_data_y = data[['Purchase']]

X_train, X_test, y_train, y_test = train_test_split(Bfriday_data_x, Bfriday_data_y, test_size=0.3)
```

Step 5: Algorithm Selection:

Once the data is splited into **Training and Testing** data we need to select appropriate Algorithm which gives > 70% accuracy

Data Scientist need to applies multiple algorithm (at least 2 or 3) just to check which one is giving better accuracy and once it confirmed that algorithm will be implemented .

In this Project We have applied two algorithm **Decision Tree Regressor and Random Forest Regressor** which giving(98 and 90)% accuracy.

Highest Accuracy id given by Decision Tree Regressor 98% so we chosen this algorithm for Prediction.

Once the algorithm is chosen we need to import required packages and methods which are required for prediction.

```
from sklearn.tree import DecisionTreeRegressor

dtr = DecisionTreeRegressor()
```

Step 6 : Training the algorithm with data for machine:

Once the algorithm is selected and finalized we need to train the data and based on that it going to predict the output and accuracy score .

Here we fit training data to Decision Tree Regressor

```
fit2 = dtr.fit(X_train,y_train)
```

Step 7 : Evaluation on test data :

Once the data is trained, test data will be evaluated and Accuracy score will be calculated .

```
print("Accuracy Score of Decision Tree on train set",fit2.score(X_train,y_train)*100)
```

output :

```
Accuracy Score of Decision Tree on train set 98.217037210071
```

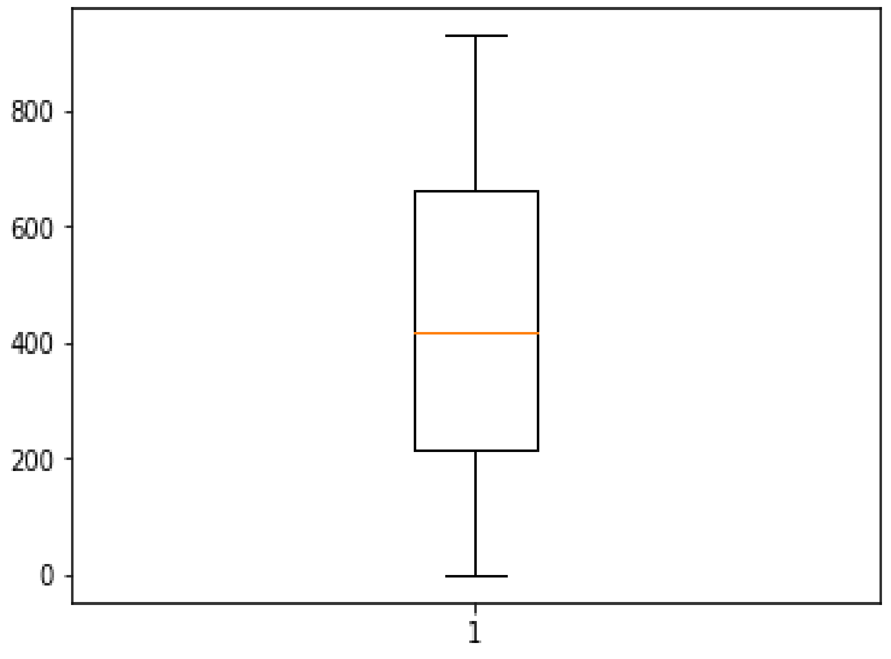
Step 8: Deployment:

This is the final step where the model is deployed and implemented once after getting accuracy score > 70 % .

VISUALIZATION

Box Plot

It is the visual representation of the depicting groups of numerical data through their quartiles. Boxplot is also used for detect the outlier in data set. It captures the summary of the data efficiently with a simple box and whiskers and allows us to compare easily across groups.

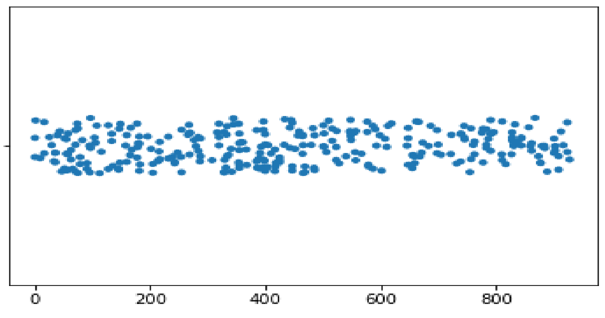


Stripplot :

A strip plot can be drawn on its own, but it is also a good complement to a box or violin plot in cases where you want to show all observations along with some representation of the underlying distribution.

```
seaborn.stripplot(y_pred,jitter=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27af721828>
```



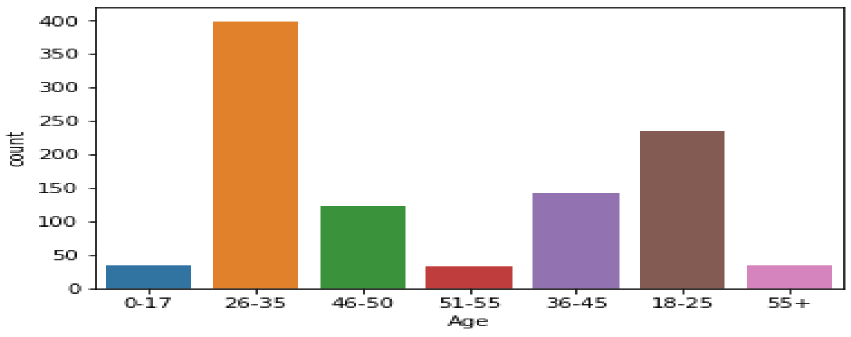
Count plot:

A count plot can be thought of as a histogram across a categorical, instead of quantitative, variable. The basic API and options are identical to those for barplot, so you can compare counts across nested variables.

In most cases, it is possible to use numpy or Python objects, but pandas objects are preferable because the associated names will be used to annotate the axes.

```
seaborn.countplot(BlackFriday_frame.Age)
```

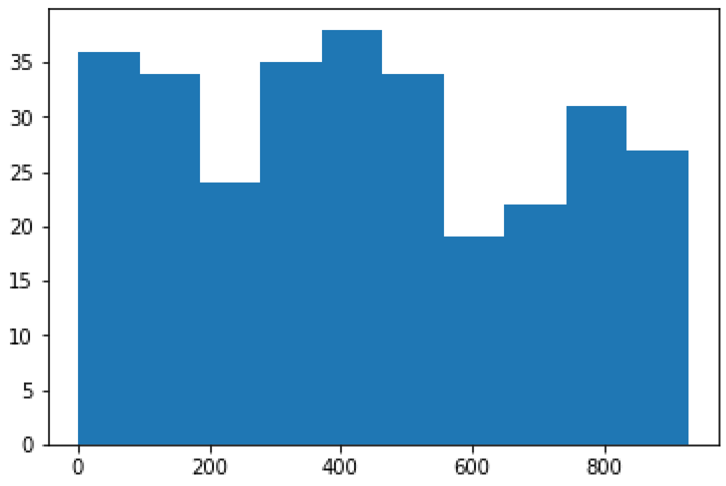
```
<matplotlib.axes._subplots.AxesSubplot at 0x27af5ff208>
```



Y_Pred Histogram

```
plt.hist(y_pred)
```

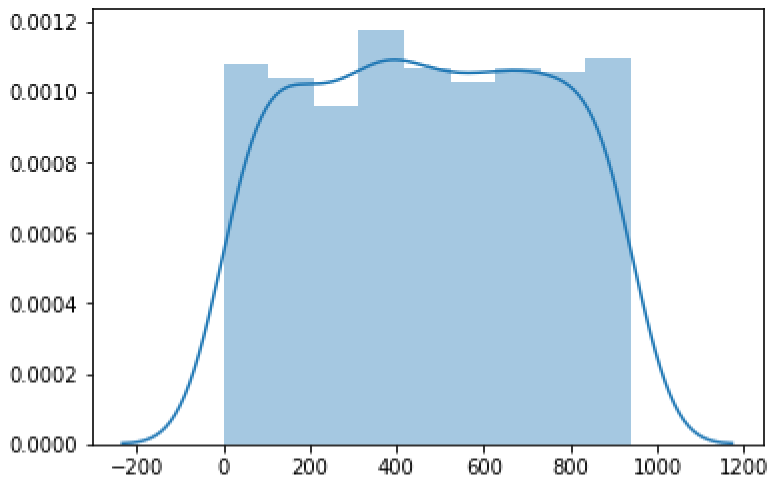
```
(array([36., 34., 24., 35., 38., 34., 19., 22., 31., 27.]),
 array([  1. ,  93.7, 186.4, 279.1, 371.8, 464.5, 557.2, 649.9, 742.6,
        835.3, 928. ]),
 <a list of 10 Patch objects>)
```



Y_train distplot

```
seaborn.distplot(y_train)
```

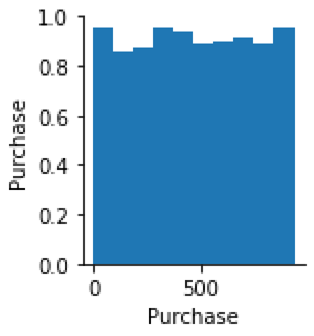
```
<matplotlib.axes._subplots.AxesSubplot at 0x27b039ed68>
```



Pair plot of purchase

```
seaborn.pairplot(y_train)
```

```
<seaborn.axisgrid.PairGrid at 0x27a9344b38>
```



Conclusion:

The future of Black Friday will rely on customer experience and the offerings driving demand. People need to feel like the While more bargain hunters opened their wallets for Black Friday this year, on average they spent a little less than in last year. Black Friday sales are still increasing, which is promising for retail. Effectively utilising this strong online trend is key for retailers to ensure Black Friday is still a profitable event. But a slow down in the intent of customers to buy things may indicate that interest is waning and expectation is increasing.