LEAD SCORING CASE STUDY

Group Assignment Members:

- Aarif Babulal Nadaf
- Gayatri R Nair
- Shweta Walde

PROBLEM STATEMENT

- An education company named X Education sells online courses to industry professionals.
- The company receives a lot of leads from direct traffic, organic searches, google, and so on, however the conversion rate is poor.
- To improve the conversion rate of the leads, the company wants to identify its potential customers based on the lead score or hot leads.
- On successful identification of leads, the conversion rate inevitably will improve as the sales team will focus on the potential leads instead of making calls to everyone.

OBJECTIVE

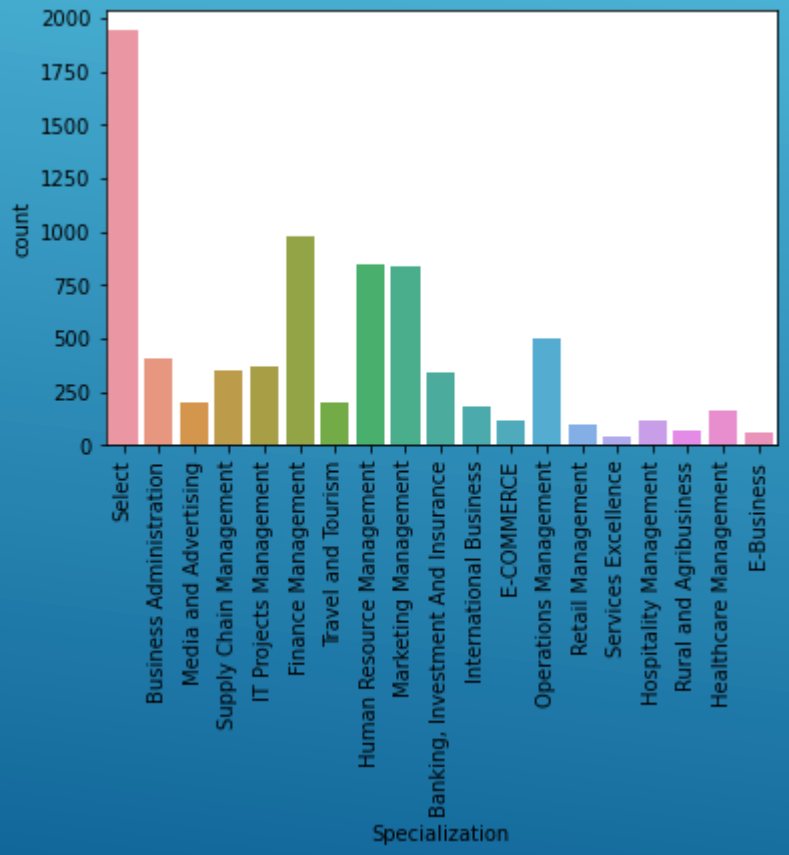
The X Education company wants to identify potential leads using logistic regression models, which can assign scores from 0 to 100 to each of the lead. A higher score would mean the lead is hot and most likely to convert.

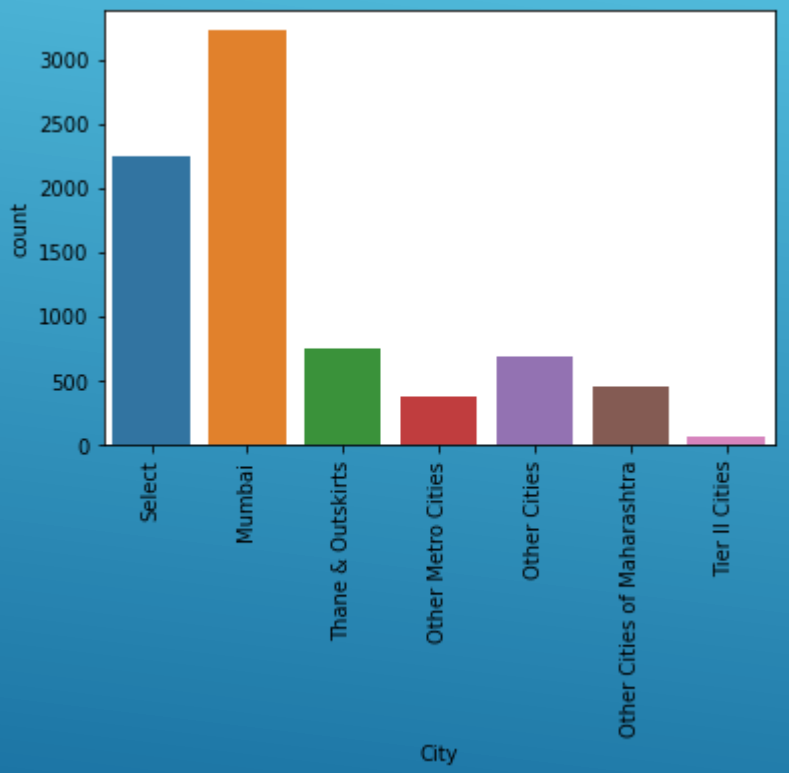
SOLUTION METHODOLOGY

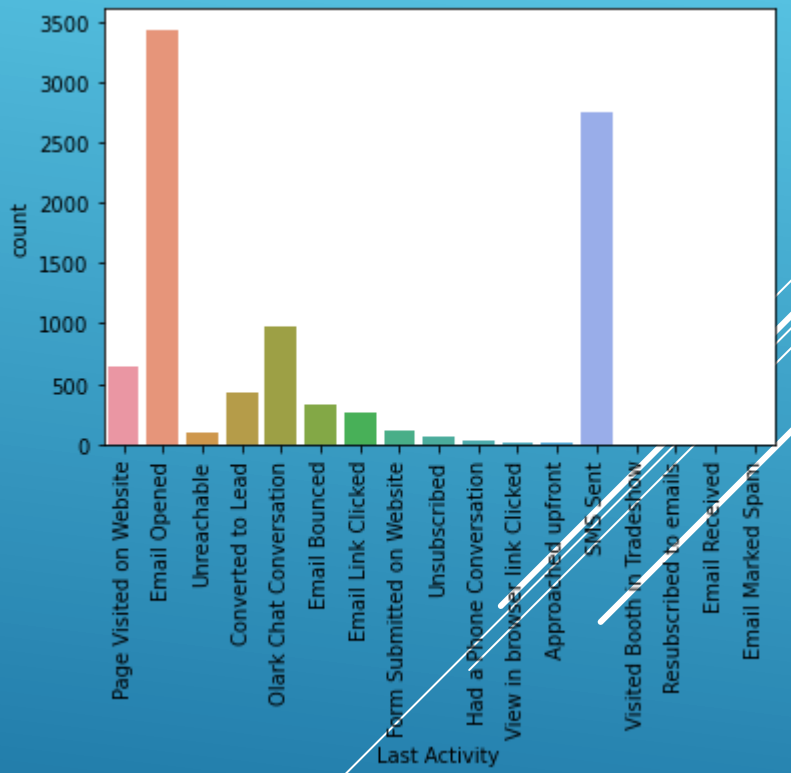
- Data understanding and preparation
  - Verifying and handling duplicate data
  - Verifying and handling missing values
  - Dropping columns if it includes large amount of missing values and not useful for the analysis
  - Imputing values as needed
  - Verifying and handling outliers in data
- Exploratory Data Analysis
- Scaling and Creating Dummy Variables while encoding the data
- Using logistic regression to make the model and perform prediction
- Validating the model
- Presenting the model
- Conclusions and recommendation

MISSING VALUES

- Identified missing values in the columns by representing it with the label “Select”





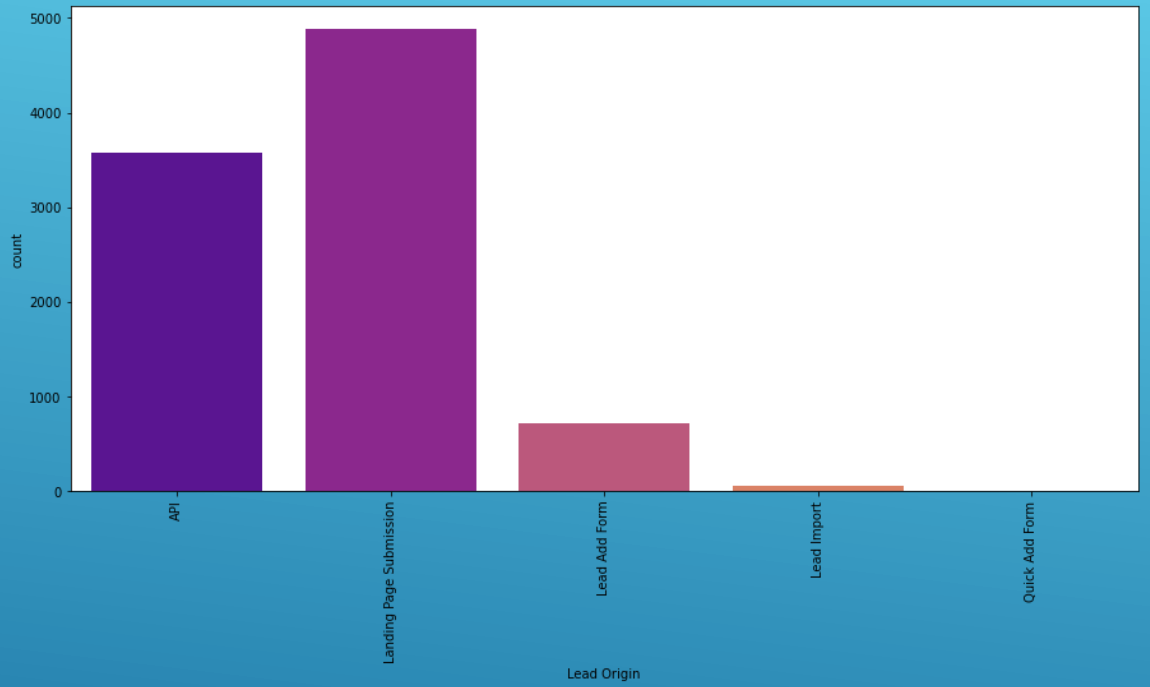


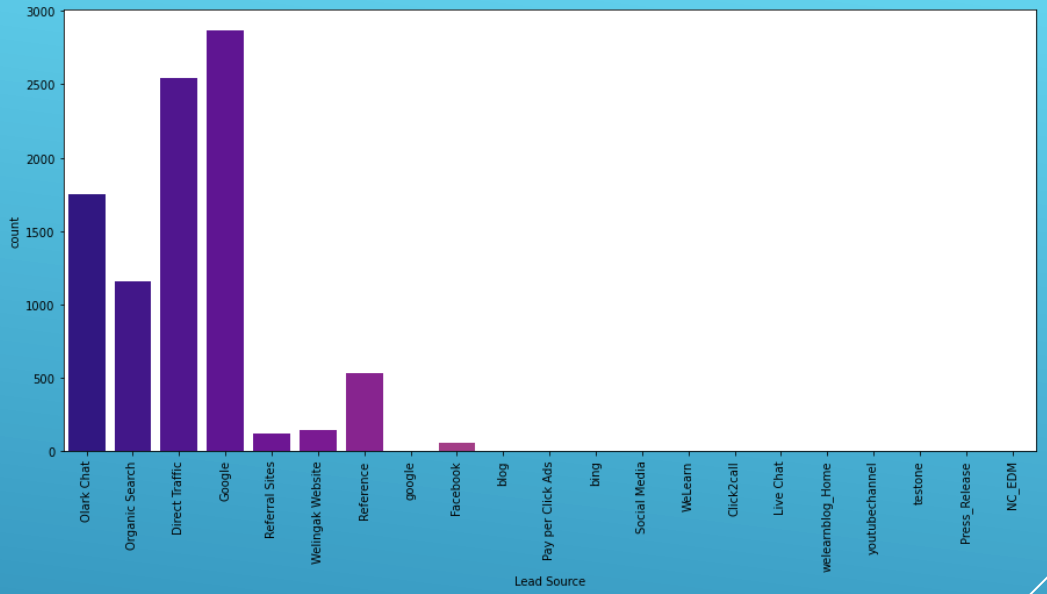
NULL VALUES

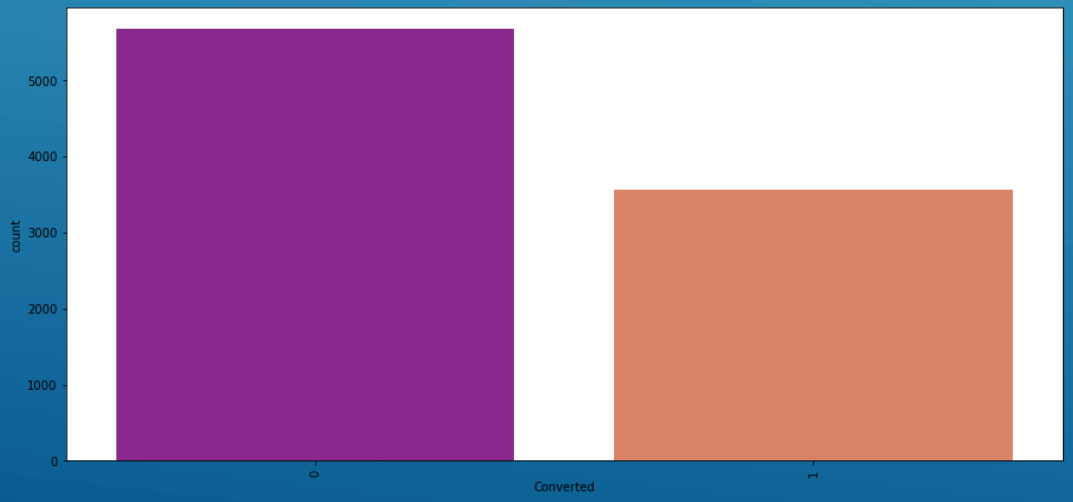
- Identified null values. As there are few columns with null values, removing it.

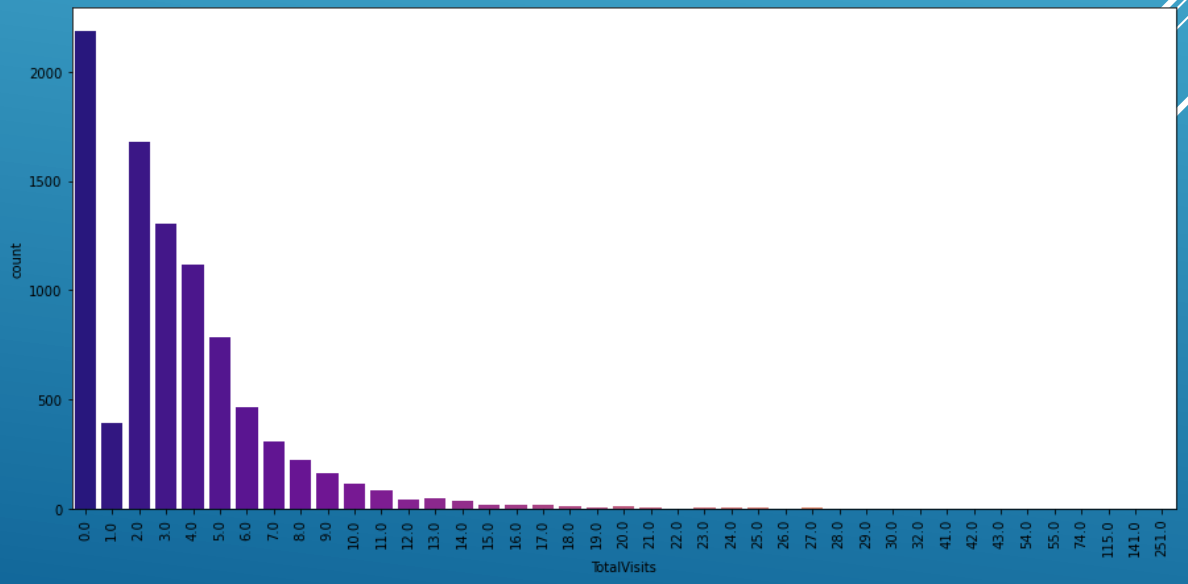
```
In [18]: #checking the null value
         df.isnull().sum()

Out[18]: Lead Origin                                  0
         Lead Source                                 36
         Converted                                    0
         TotalVisits                                137
         Total Time Spent on Website                  0
         Page Views Per Visit                       137
         Last Activity                              103
         A free copy of Mastering The Interview       0
         Last Notable Activity                        0
         dtype: int64
```

PERFORMING EDA

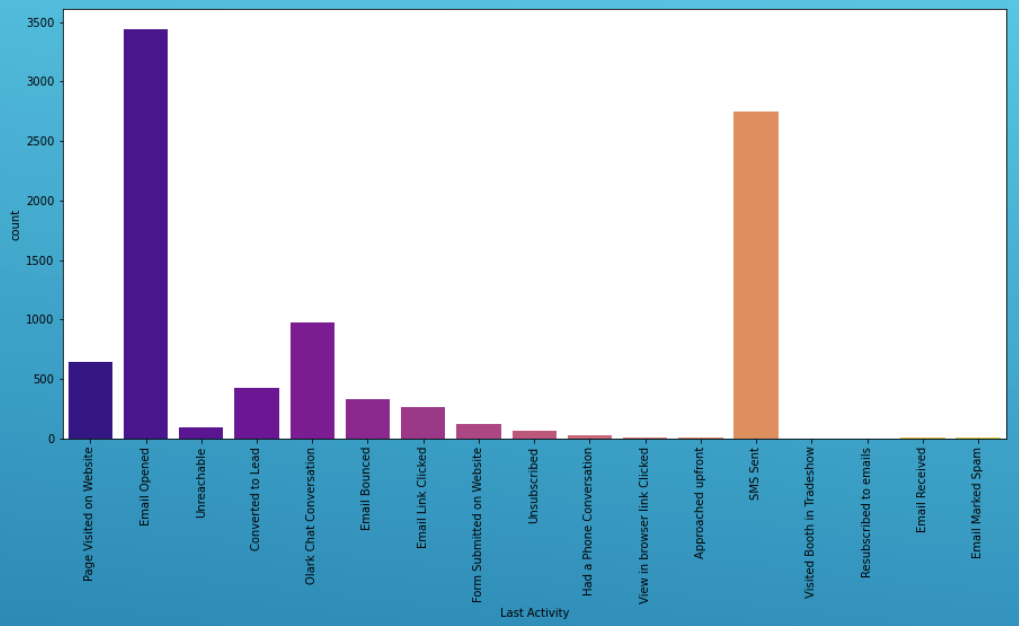


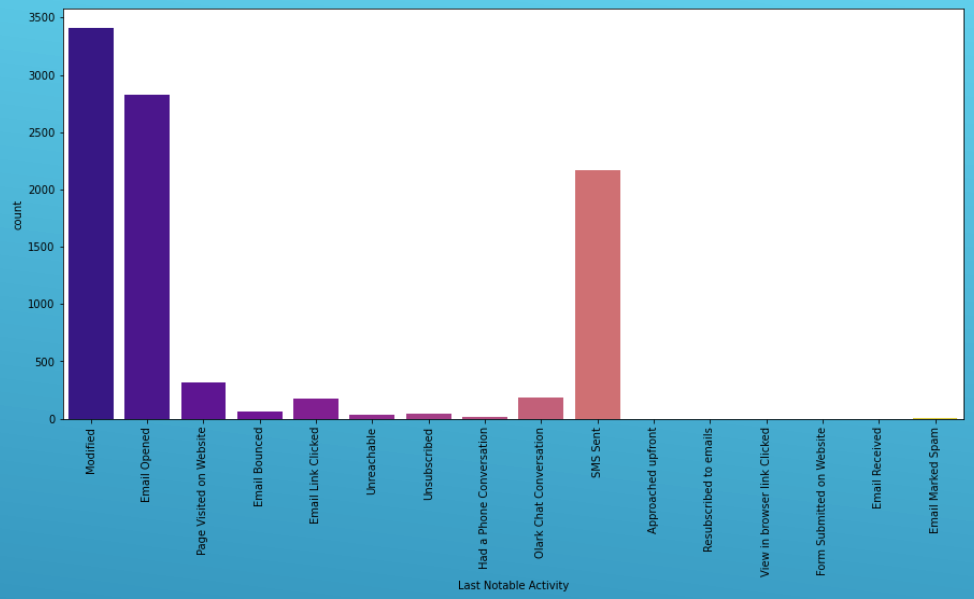


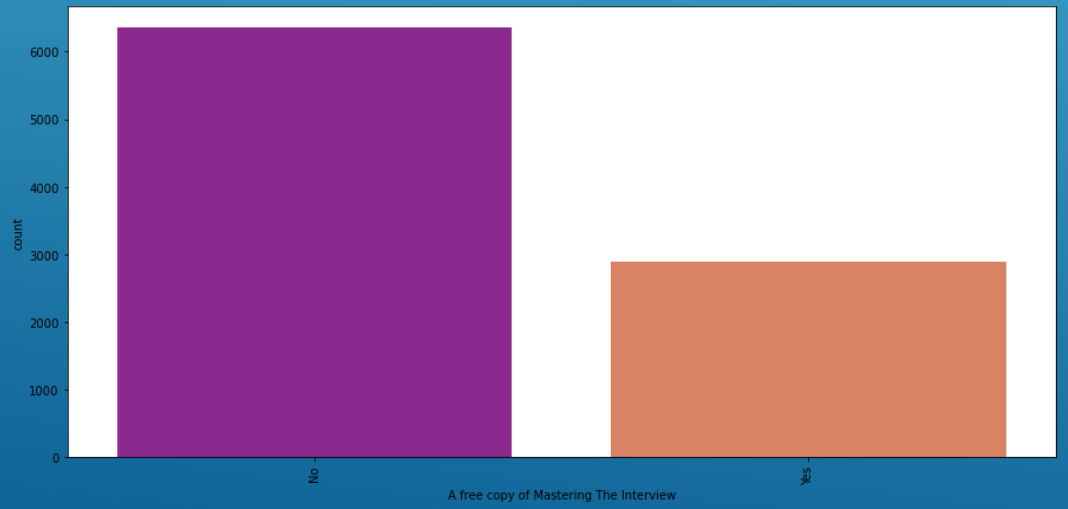




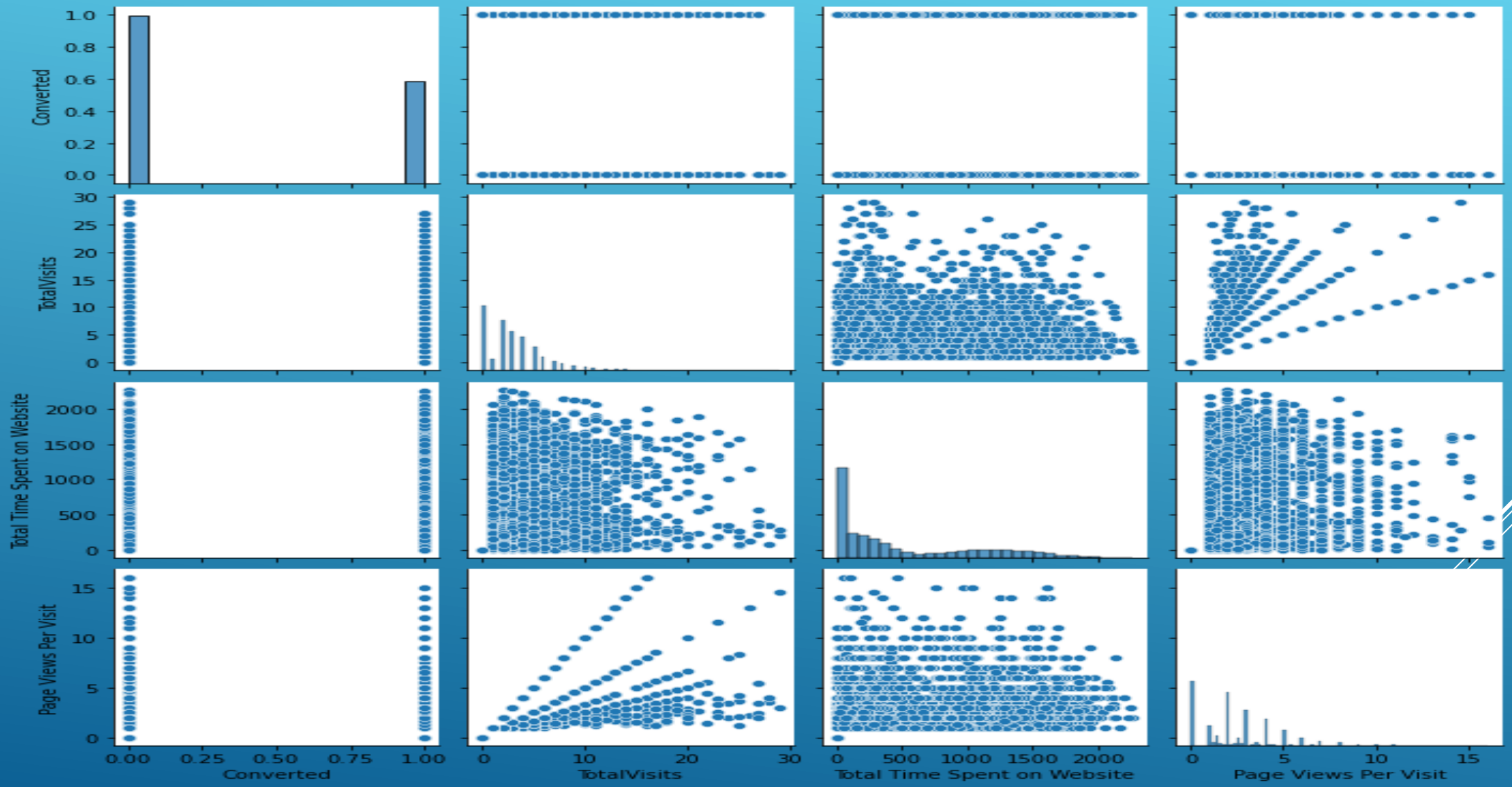
PERFORMING EDA







PERFORMING EDA



DATA PREPROCESSING

- Numeric variables are normalized
- Dummy variables are created for object type variables

```
In [49]: #checking info
         df_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9062 entries, 0 to 9239
Data columns (total 9 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Lead Origin                             9062 non-null   object
 1   Lead Source                             9062 non-null   object
 2   Converted                               9062 non-null   int64
 3   TotalVisits                             9062 non-null   float64
 4   Total Time Spent on Website             9062 non-null   int64
 5   Page Views Per Visit                    9062 non-null   float64
 6   Last Activity                           9062 non-null   object
 7   A free copy of Mastering The Interview  9062 non-null   int64
 8   Last Notable Activity                   9062 non-null   object
dtypes: float64(2), int64(3), object(4)
memory usage: 966.0+ KB
```

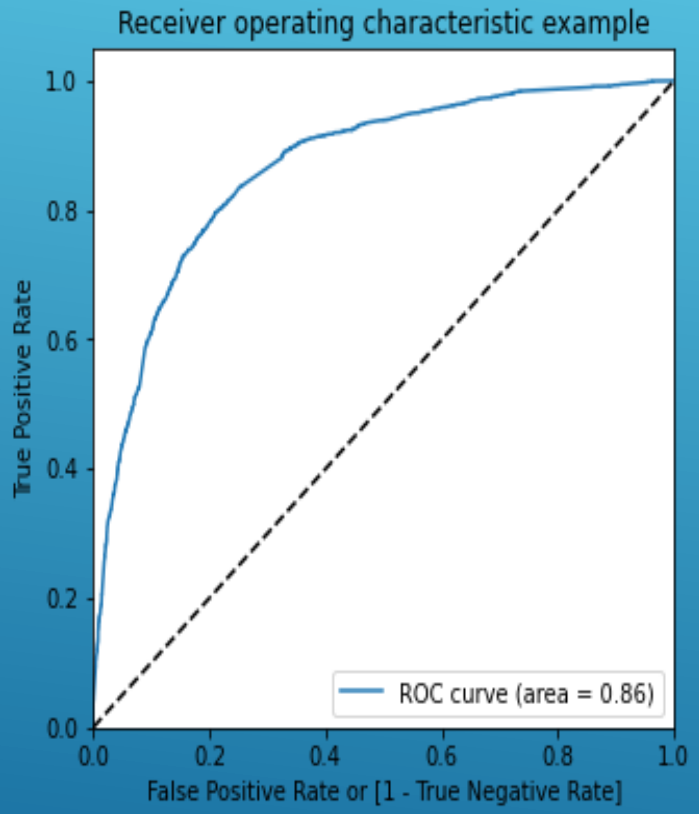
Creating dummy variable

```
In [50]: #Creating dummy of the categorical variable
         dummy1 = pd.get_dummies(df_cleaned[['Lead Origin', 'Lead Source', 'Last Activity', 'Last Notable Activity']], drop_first=True)

In [51]: # Adding the results to the master dataframe
         df_cleaned = pd.concat([df_cleaned, dummy1], axis=1)

In [52]: #Dropping column already created to dummy.
         df_cleaned=df_cleaned.drop(['Lead Origin', 'Lead Source', 'Last Activity', 'Last Notable Activity'],axis=1)

In [53]: #Checking shape of the data
         df_cleaned.shape

Out[53]: (9062, 37)
```

BUILDING THE MODEL

- Splitting the Data into Training and Testing sets
- Using RFE for feature selection
- Running RFE on 15 variables as output
- Removing the variable whose p- value is greater than 0.05 and vif value is greater than 5 to build the model
- Making predictions on the test data set
- Overall accuracy is 80%

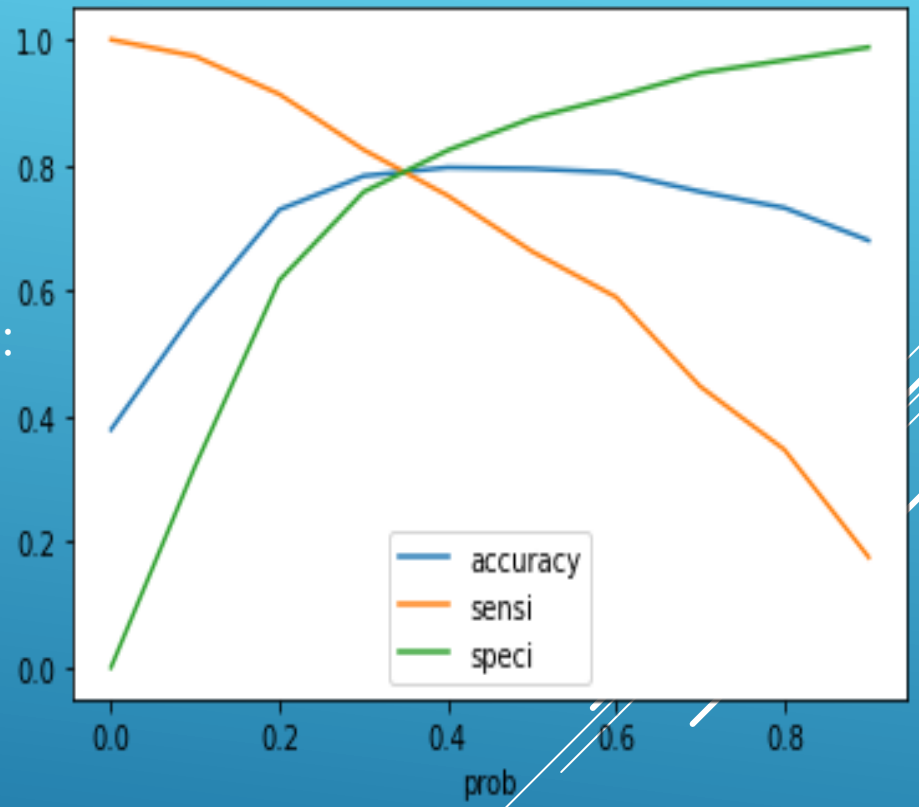
ROC CURVE



From the **Accuracy, Sensitiviy and Specificity Graph** graph, it is apparent that the optimal cut off is at 0.35.

Evaluation after cut off points 0.35:

- Accuracy - 0.7914
- Sensitivity - 0.7903
- False Positive Rate - 0.2096
- Positive Predicted - 0.6977
- Precision - 0.6977
- Recall - 0.79322



CONCLUSION

These inferences were made based on the analysis:

- The following sources impacted the potential leads:
  - Google
  - Direct traffic
  - Organic Search
- The last activity that affected the leads were:
  - Opened emails
  - SMS
  - Olark Chat Conversation
- The total time spent on the website and the total number of visits had their share of impact
- Working professions contributed to the lead
- Final Model (res) res = logm4.fit()
- The cut off probability is 0.35
- More than 0.35 were converted as lead
- Less than 0.35 will not be converted as lead
- Accuracy of the train data 0.791
- Accuracy of the test data 0.784
- When the lead is increased or decreased, the Cut off can be adjusted
- The lead score targeting can be done from the top.