

LEAD SCORE CASE STUDY

Problem Statement :

- An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses.
- Now, although X Education gets a lot of leads, its lead conversion rate is very poor.
- The company wants to increase it to 80%

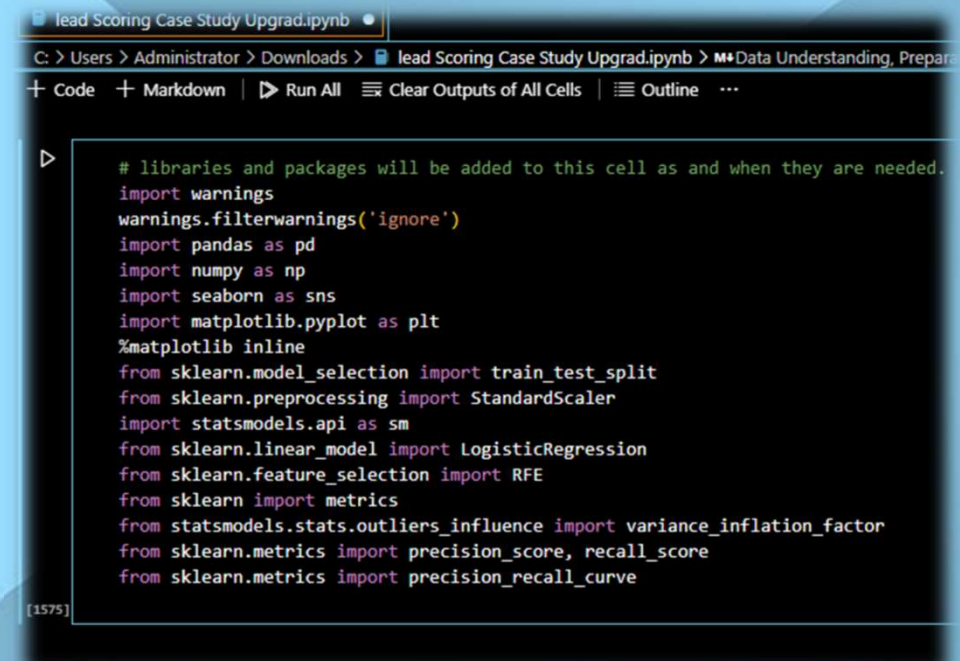
GOAL :

Build a logistic regression model to assign a lead score between 0 and 100 to each of the leads which can be used by the company to target potential leads.

A higher score would mean that the lead is hot, i.e. is most likely to convert whereas a lower score would mean that the lead is cold and will mostly not get converted.

Strategy

- Importing Data
- Cleaning and preparing the data
- EDA
- Scaling the features
- Preparing data for model building
- Assigning a lead score for each of the leads
- Testing the model on train set
- Evaluating model
- Testing the model on test set
- Measuring Accuracy of model and other metrics



The screenshot shows a Jupyter Notebook window titled "lead Scoring Case Study Upgrad.ipynb". The file path is "C: > Users > Administrator > Downloads > lead Scoring Case Study Upgrad.ipynb". The interface includes tabs for "Code", "Markdown", "Run All", "Clear Outputs of All Cells", and "Outline". A code cell is expanded, showing the following Python code:

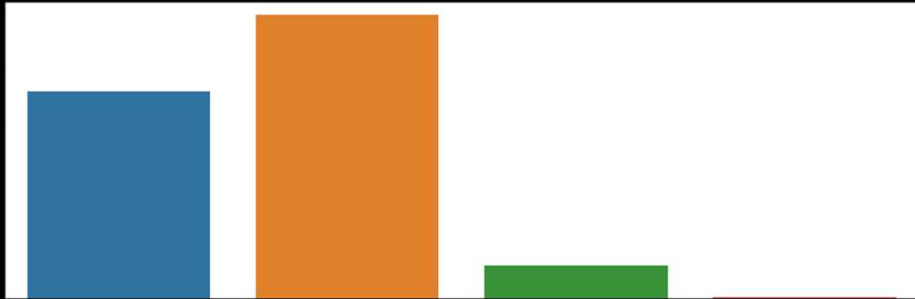
```
# libraries and packages will be added to this cell as and when they are needed.
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE
from sklearn import metrics
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import precision_recall_curve
```

The output of the cell is "[1575]".

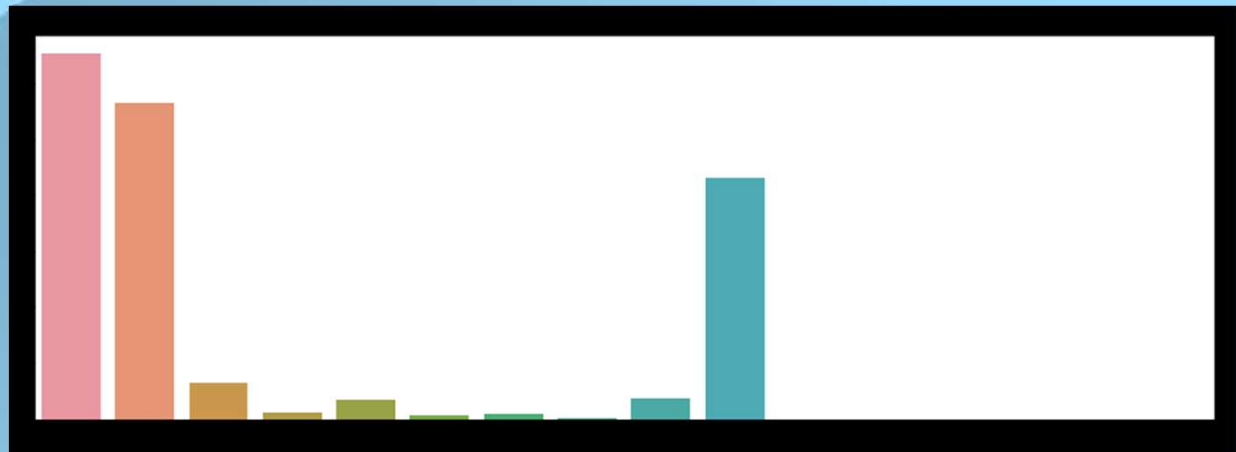
EXPLORATORY DATA ANALYSIS

Univariate Analysis

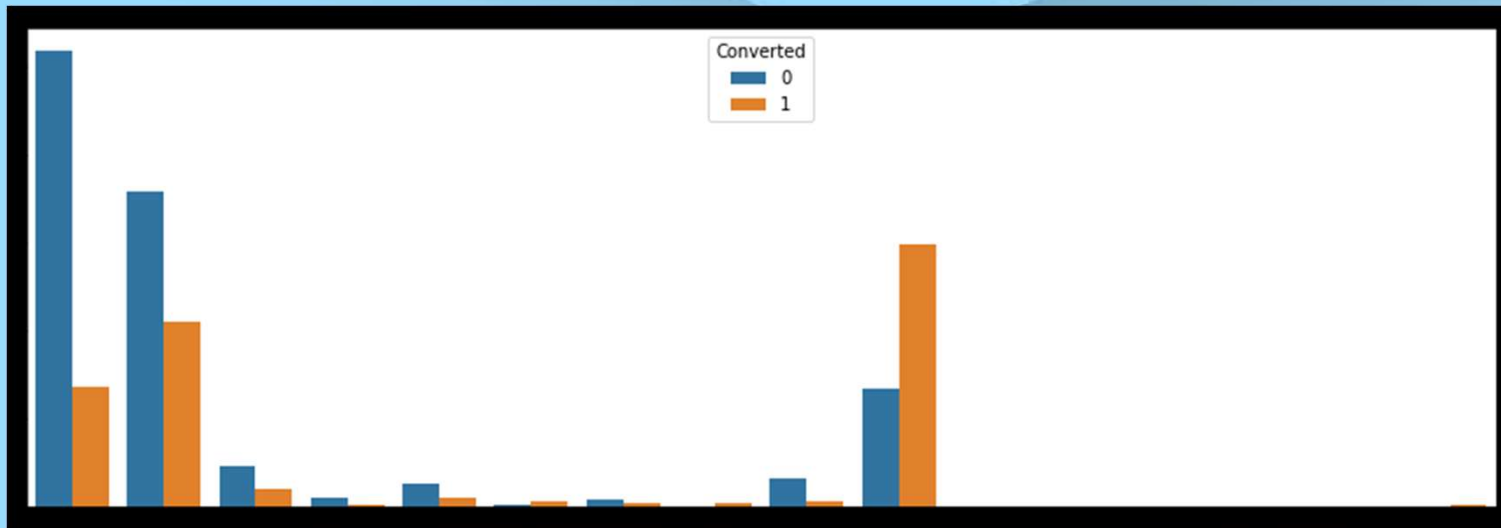
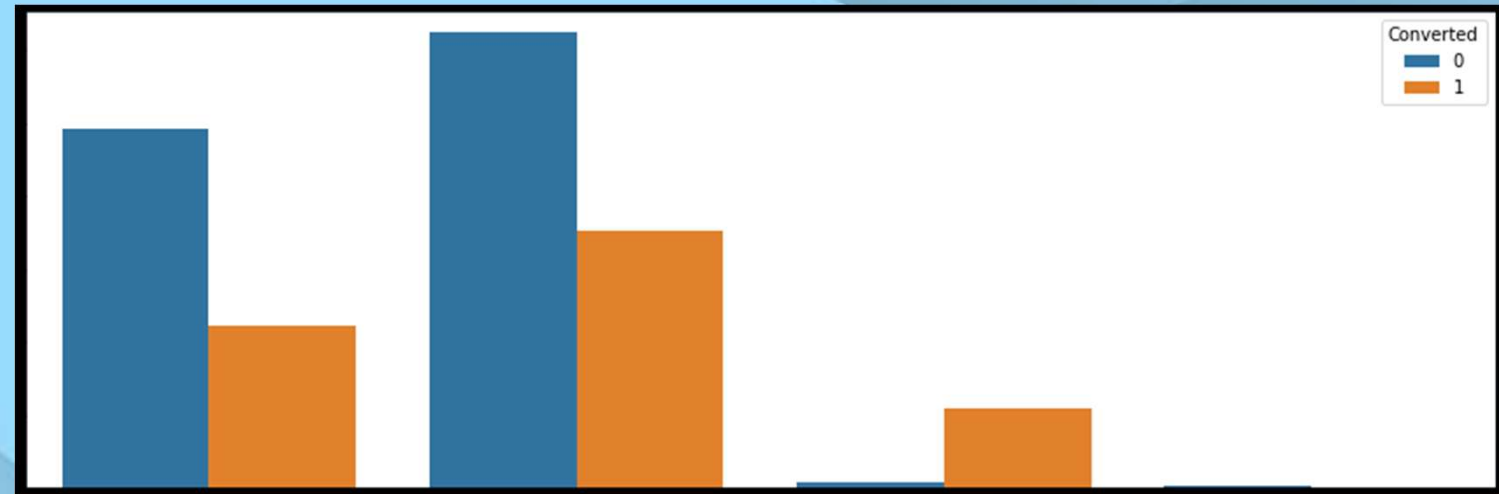
```
for column in categorical_vars.columns:  
    countplot_univariate(column)
```



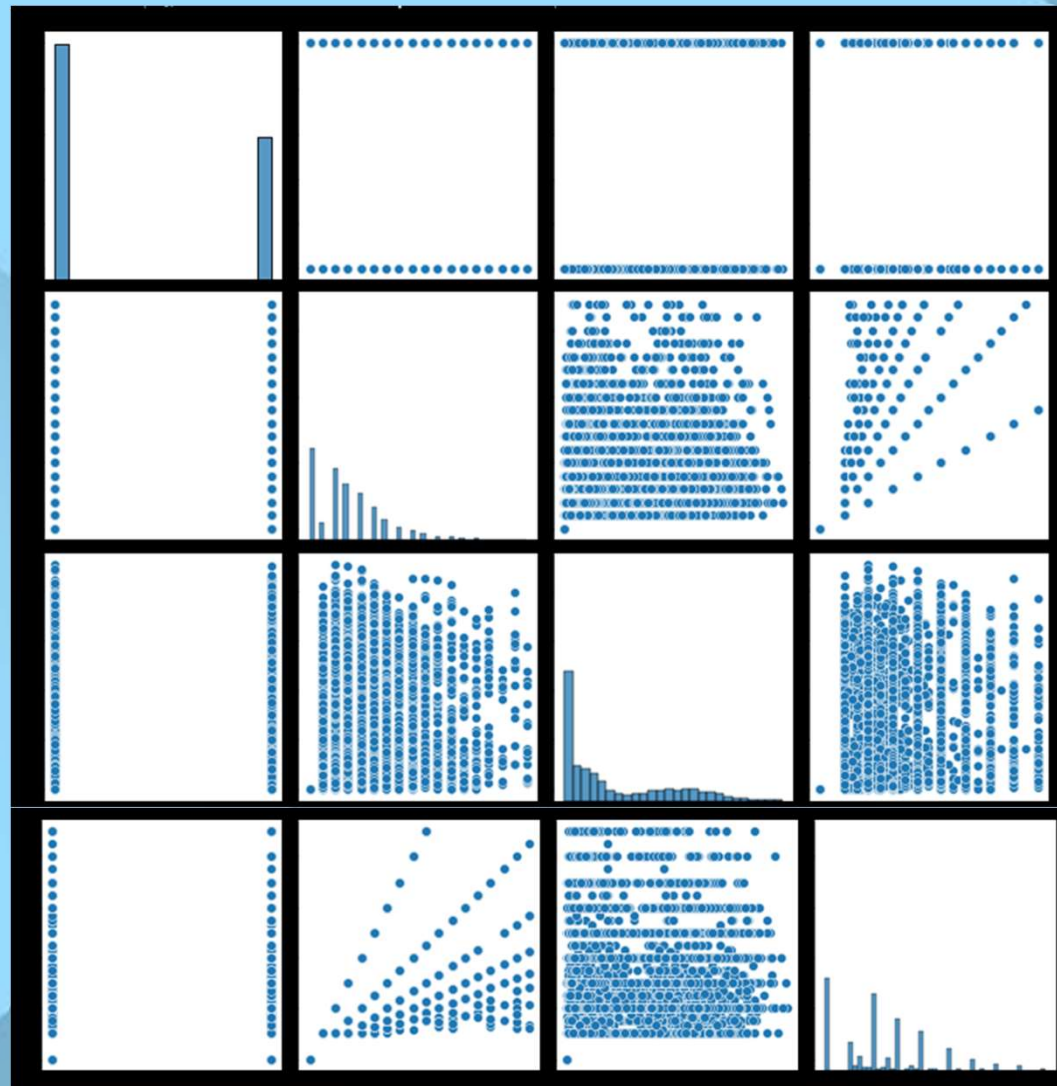
Google searches had high conversions as compared to other modes.



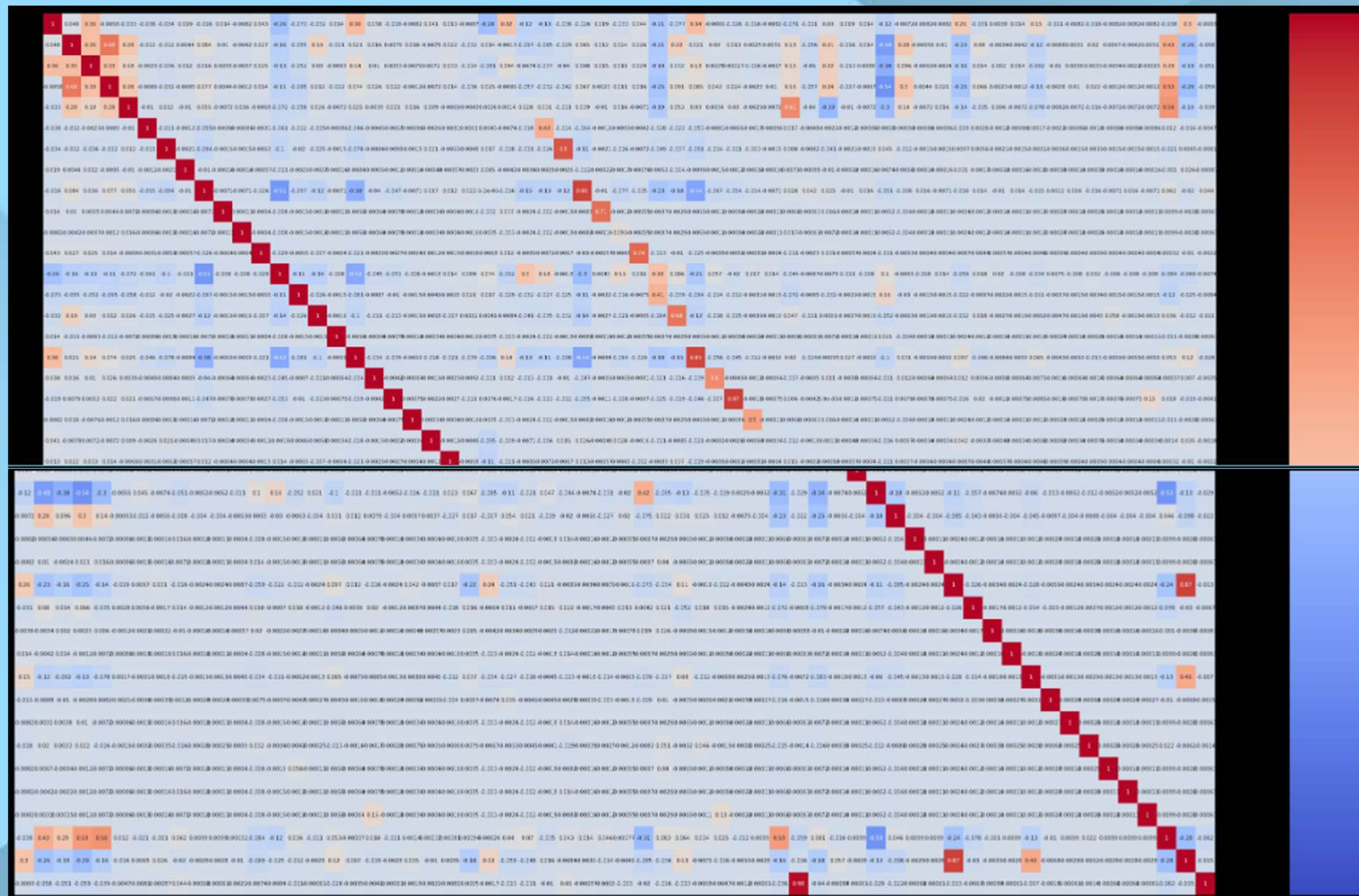
Bi – Variate Analysis



Pair plot to check the relationship between the converted and the numeric columns



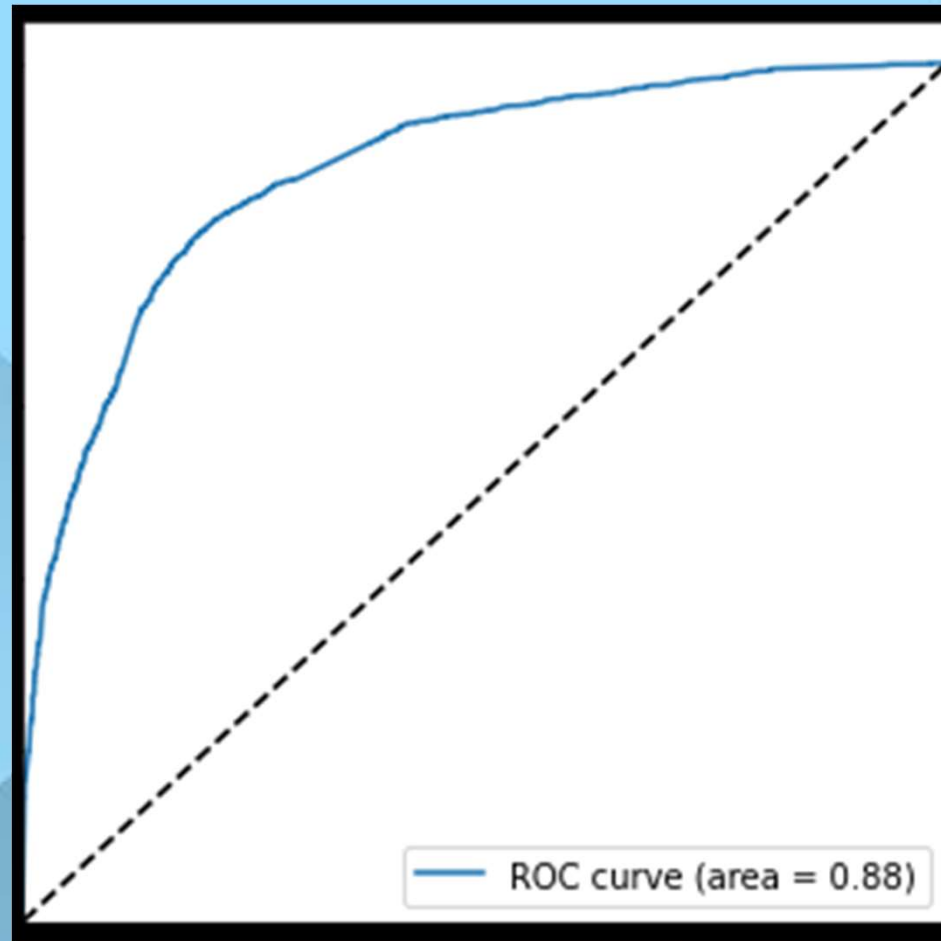
Correlation Matrix



MODEL BUILDING

- To split into train and test set
- Scale the variable
- Building first model
- Using RFE
- Building next model
- Eliminating variable based on high p-values
- Checking VIF
- Predicting accuracy
- Evaluating accuracy
- Predicting by using test set
- Precision and recall analysis

Plotting the ROC curve

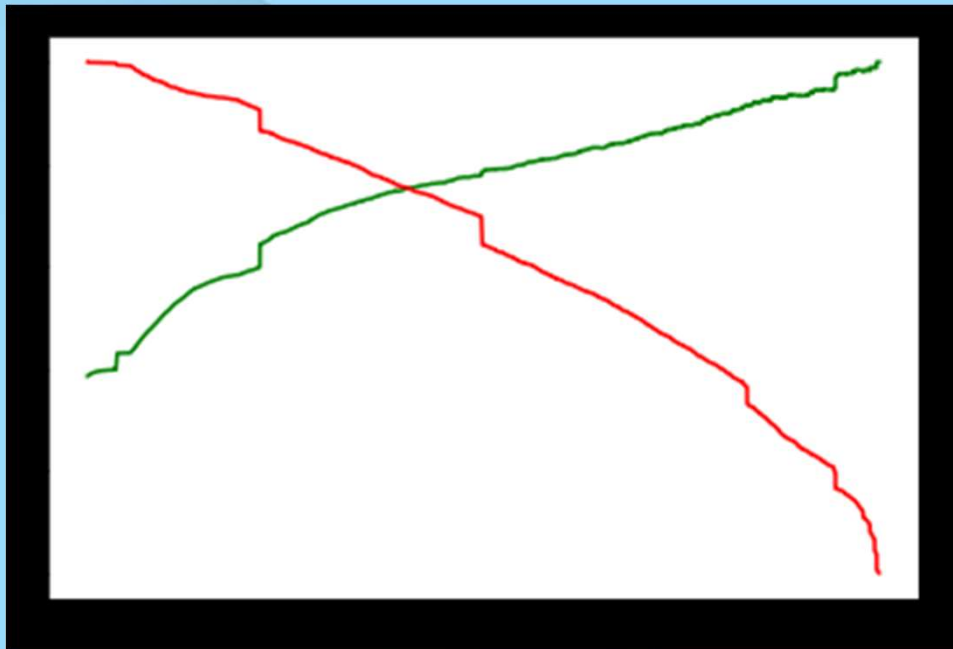


FINAL OBSERVATIONS

```
# Checking the parameters and their value to the model
result.params.sort_values(ascending=False)

[732]
... Lead Source_Welingak Website        6.439660
     Lead Source_Reference                4.199732
     Last Notable Activity_Had a Phone Conversation  2.439575
     Last Notable Activity_Unreachable            1.969361
     Lead Origin_Lead Import                1.444759
     Lead Source_Olark Chat                1.372520
     Last Activity_SMS Sent                1.225233
     Total Time Spent on Website            1.170855
     Last Activity_Had a Phone Conversation        1.049196
     What is your current occupation_Working Professional  0.988228
     const                                0.266553
     What is your current occupation_Student        -1.478473
     Last Activity_Olark Chat Conversation        -1.683645
     Last Activity_Email Bounced              -1.697975
     What is your current occupation_Unemployed    -1.801652
     dtype: float64
```

Model Evaluation test



- **72.5% Precision**
- **76.5% Recall**
- **80.6% Accuracy**
- **76.5% Sensitivity**
- **83.1% Specificity**

Model Evaluation Train

accuracy = 80.9%
sensitivity = 70.1%
specificity = 87.5%

Conclusion

Logistic Regression :

- ❑ The threshold has been selected from Accuracy, sensitivity, specificity and precision, recall curves
- ❑ The model shows high close to 80% accuracy
- ❑ Overall this model is accurate.

EDA :

- ❖ People who spend more than the average time are promising leads hence approaching them can be helpful in conversions.
- ❖ Landing page submissions can help out more leads
- ❖ Reference lead can be good source for higher conversions.