Case study objective

An education company named X Education sells online courses to industry professionals.

- The objective of this case study is to select the most promising leads, i.e., the leads that are most likely to convert into paying customers, also known as “hot leads”
- Build a model wherein you need to assign a lead score to each of the leads such that the customers with higher lead score have a higher conversion chance
- Target is to obtain lead conversion rate to be around 80%.

Problem Solving Methodology

Phase 1: Business Understanding

Identify the most potential leads, also known as “hot leads,” to increase the conversion rate.

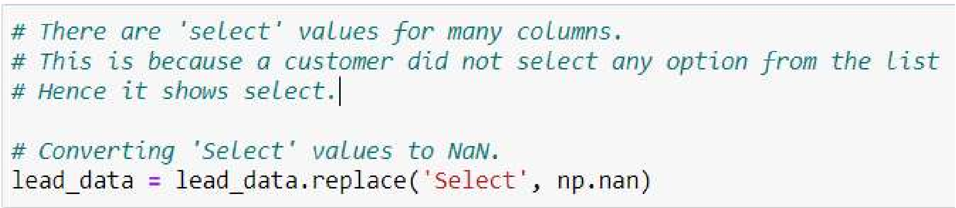
Phase 2: Data Understanding

- Dataset provided for analysis is the Leads dataset from the past with around 9000 data points.
- The dataset consists of various attributes such as Lead Source, Last Activity, etc. which may or may not be useful in ultimately deciding whether a lead will be converted or not.
- The target variable, in this case, is the column ‘Converted’ which tells whether a past lead was converted or not, wherein 1 means it was converted and 0 means it wasn’t converted.

Phase 3: Data Preparation

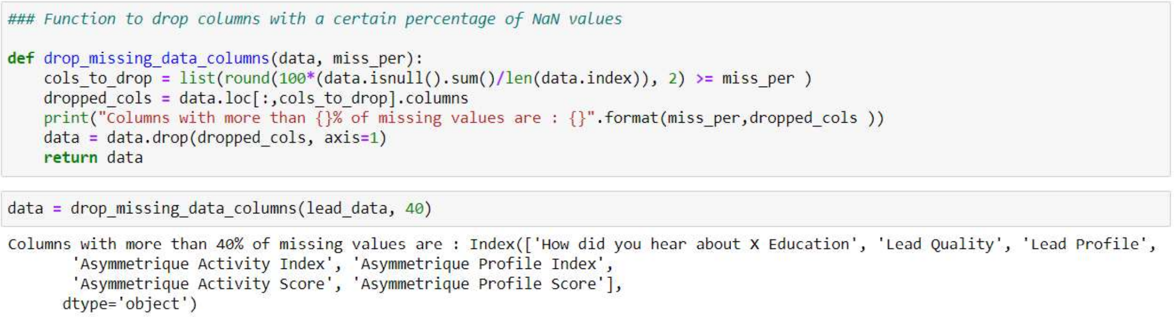
- Identification of important variables
  a. Target variable and feature variables
  b. Data types
  c. Categorical or Continuous variable

- Handled “Select” level in multiple categorical columns by replacing it with null value.



```
# There are 'select' values for many columns.
# This is because a customer did not select any option from the list
# Hence it shows select.

# Converting 'Select' values to NaN.
lead_data = lead_data.replace('Select', np.nan)
```

- Handled missing values by deleting columns which has more than 40% of missing values.

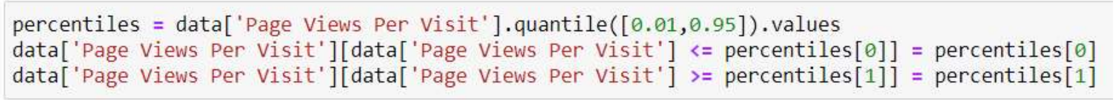


```
### Function to drop columns with a certain percentage of NaN values

def drop_missing_data_columns(data, miss_per):
    cols_to_drop = list(round(100*(data.isnull().sum()/len(data.index)), 2) >= miss_per )
    dropped_cols = data.loc[:,cols_to_drop].columns
    print("Columns with more than {}% of missing values are : {}".format(miss_per,dropped_cols ))
    data = data.drop(dropped_cols, axis=1)
    return data
```

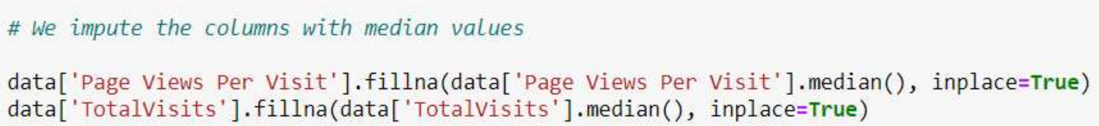
```
data = drop_missing_data_columns(lead_data, 40)
```

```
Columns with more than 40% of missing values are : Index(['How did you hear about X Education', 'Lead Quality', 'Lead Profile',
       'Asymmetrique Activity Index', 'Asymmetrique Profile Index',
       'Asymmetrique Activity Score', 'Asymmetrique Profile Score'],
      dtype='object')
```

- Checked outliers in numerical columns and performed outlier treatment.

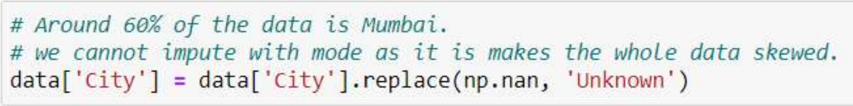


```
percentiles = data['Page Views Per Visit'].quantile([0.01,0.95]).values
data['Page Views Per Visit'][data['Page Views Per Visit'] <= percentiles[0]] = percentiles[0]
data['Page Views Per Visit'][data['Page Views Per Visit'] >= percentiles[1]] = percentiles[1]
```



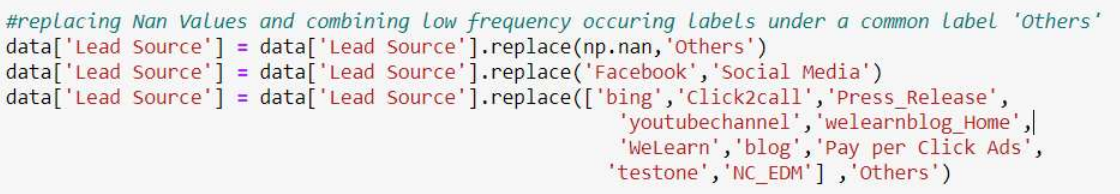
```
# We impute the columns with median values

data['Page Views Per Visit'].fillna(data['Page Views Per Visit'].median(), inplace=True)
data['TotalVisits'].fillna(data['TotalVisits'].median(), inplace=True)
```

- Identified null values in all columns and replaced them with appropriate values.



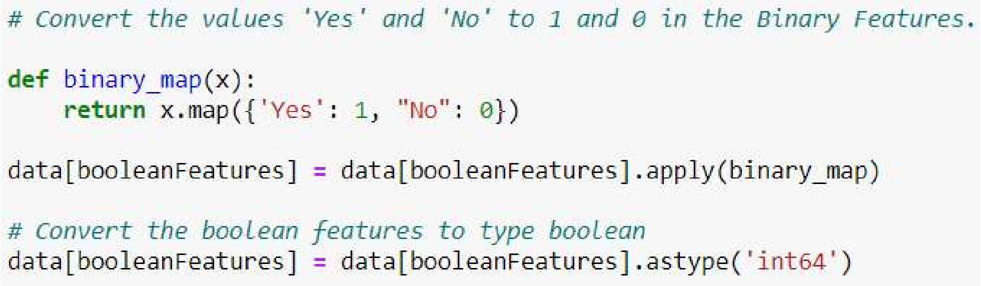
```
# Around 60% of the data is Mumbai.
# we cannot impute with mode as it is makes the whole data skewed.
data['City'] = data['City'].replace(np.nan, 'Unknown')
```

- Identified low frequency occurring levels in a categorical column and combined them to a common level as “Others”



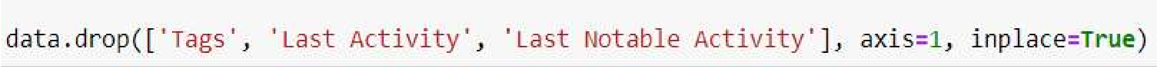
```
#replacing Nan Values and combining low frequency occuring labels under a common label 'Others'
data['Lead Source'] = data['Lead Source'].replace(np.nan,'Others')
data['Lead Source'] = data['Lead Source'].replace('Facebook','Social Media')
data['Lead Source'] = data['Lead Source'].replace(['bing','Click2call','Press_Release',
                                                    'youtubechannel','welearnblog_Home',
                                                    'WeLearn','blog','Pay per Click Ads',
                                                   'testone','NC_EDM'] ,'Others')
```

- Converted binary variables Yes/No to 1/0.



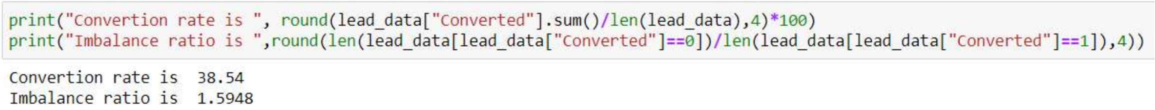
```
# Convert the values 'Yes' and 'No' to 1 and 0 in the Binary Features.

def binary_map(x):
    return x.map({'Yes': 1, "No": 0})

data[booleanFeatures] = data[booleanFeatures].apply(binary_map)

# Convert the boolean features to type boolean
data[booleanFeatures] = data[booleanFeatures].astype('int64')
```

- Dropped the columns generated by the Sales team.



```
data.drop(['Tags', 'Last Activity', 'Last Notable Activity'], axis=1, inplace=True)
```

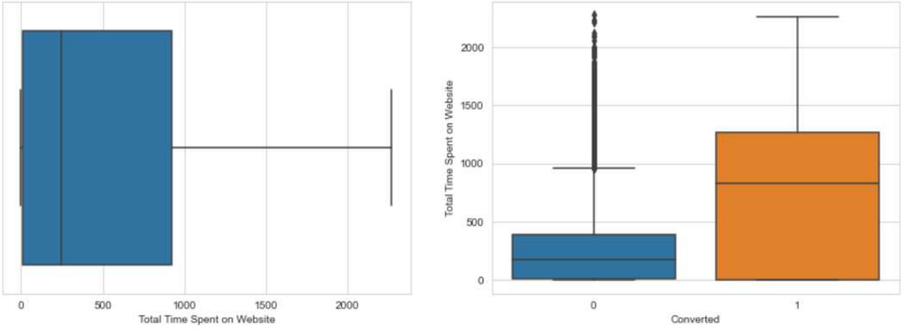
- Checked the data imbalance and the imbalance ratio.

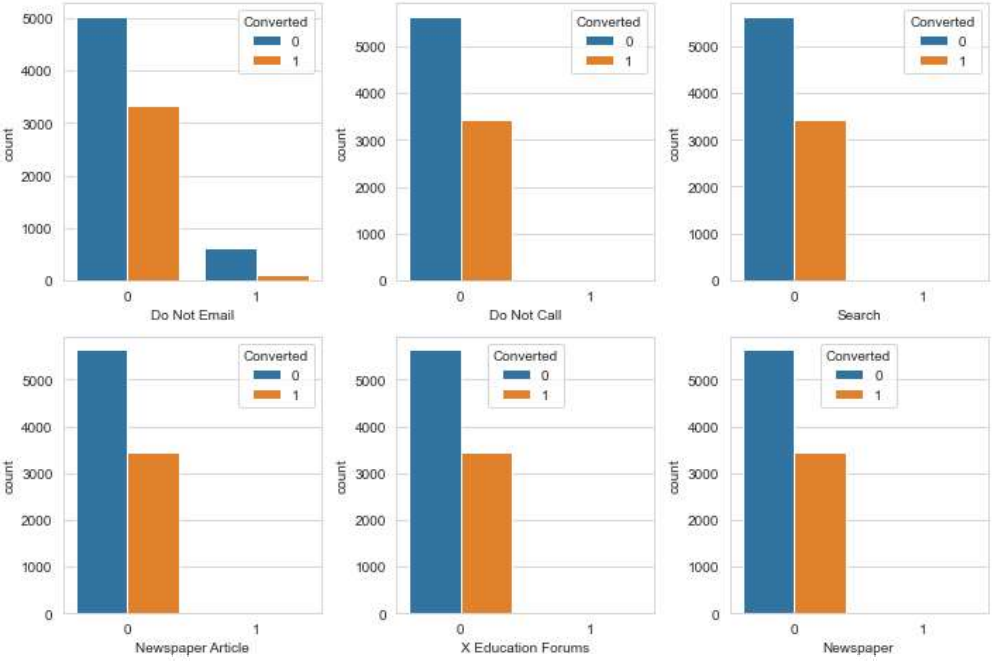


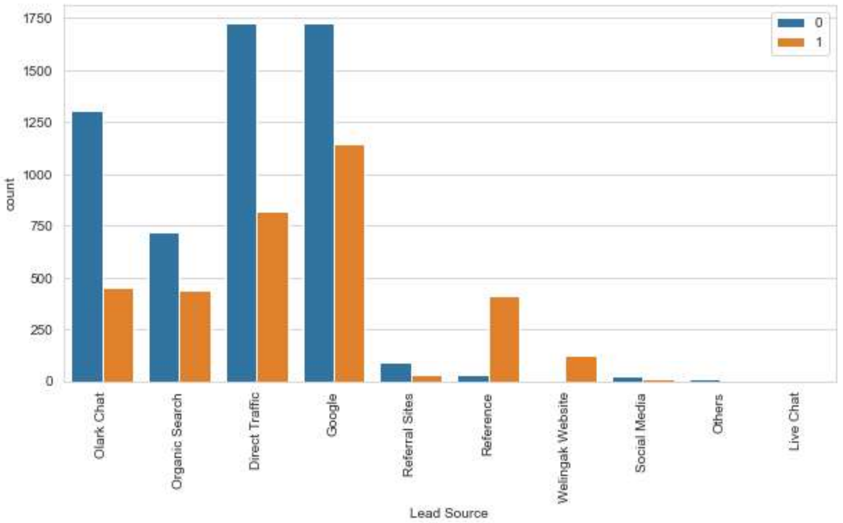
```
print("Convertion rate is ", round(lead_data["Converted"].sum()/len(lead_data),4)*100)
print("Imbalance ratio is ",round(len(lead_data[lead_data["Converted"]==0])/len(lead_data[lead_data["Converted"]==1]),4))
```

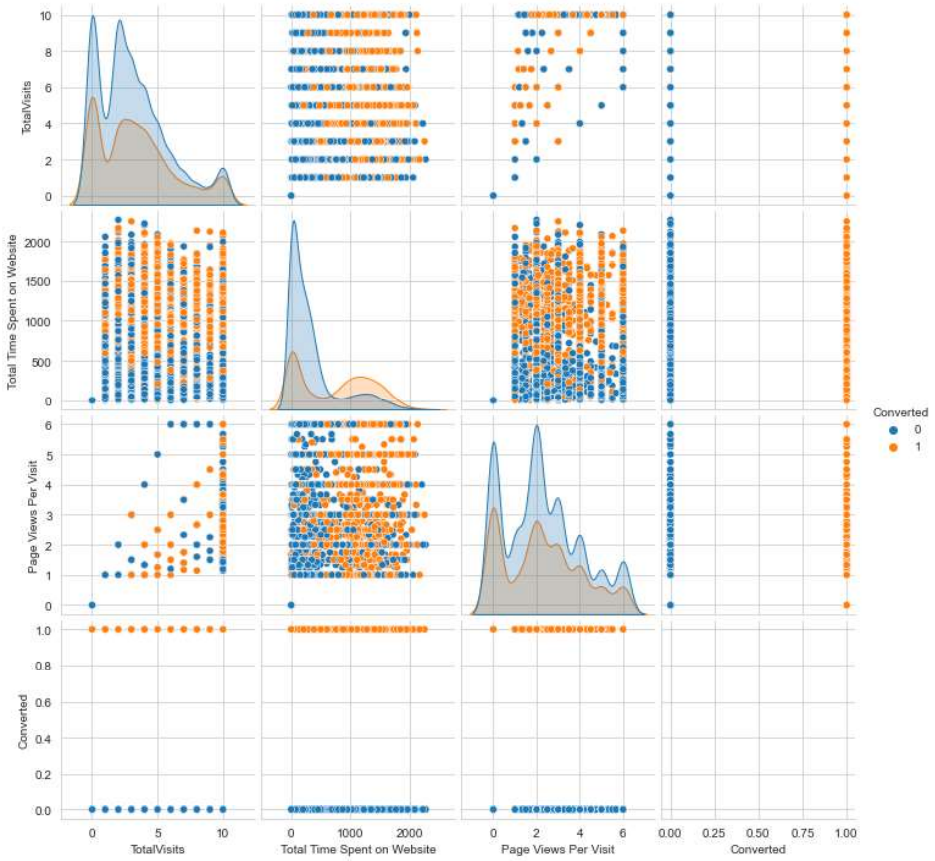
```
Convertion rate is  38.54
Imbalance ratio is  1.5948
```

- Performed univariate, bivariate, and multivariate analysis.

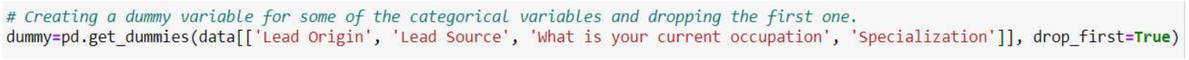






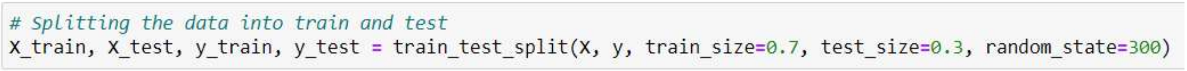


- Created dummy variables for categorical variables with multiple levels.



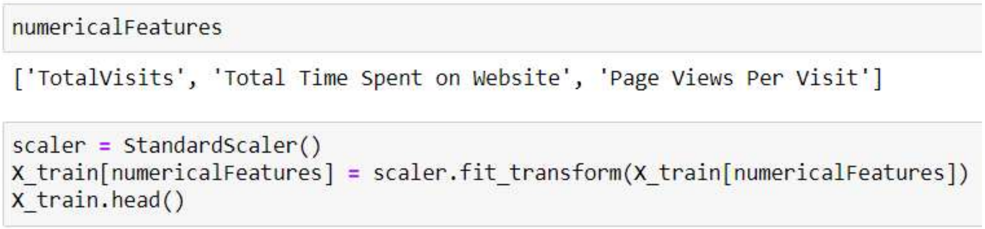
```
# Creating a dummy variable for some of the categorical variables and dropping the first one.
dummy=pd.get_dummies(data[['Lead Origin', 'Lead Source', 'What is your current occupation', 'Specialization']], drop_first=True)
```

- Performed 70%-30% train-test split.



```
# Splitting the data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3, random_state=300)
```

- Identified and performed scaling of numerical variables using Standard Scalar.



```
numericalFeatures
```

```
['TotalVisits', 'Total Time Spent on Website', 'Page Views Per Visit']
```

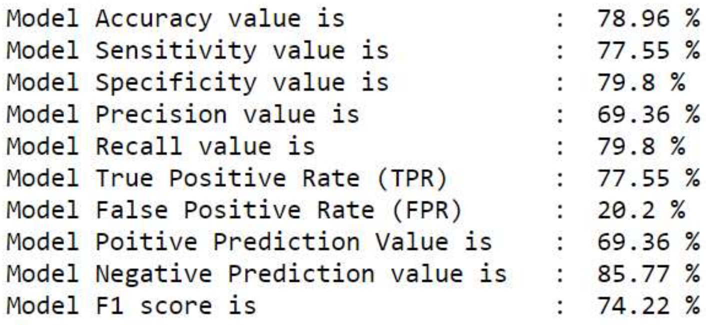
```
scaler = StandardScaler()
X_train[numericalFeatures] = scaler.fit_transform(X_train[numericalFeatures])
X_train.head()
```

Phase 4: Model Building

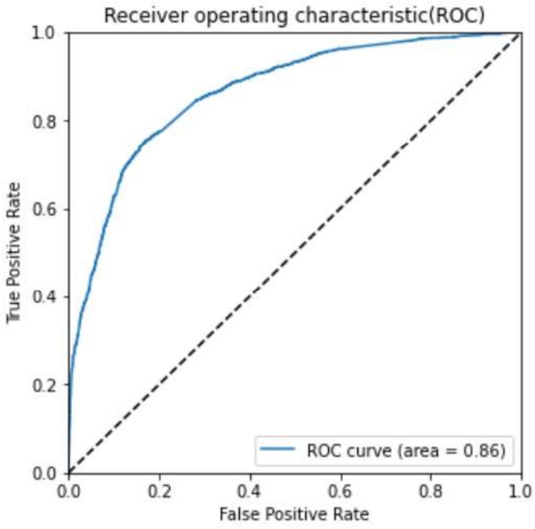
- Recursive Feature Elimination (RFE) was used to select top 20 variables.
- Logistic Regression Model was built and VIF and p-values were checked.
- Using Manual Feature Elimination, a variable was recursively eliminated until VIF and p-values were significant.
- Initially, an arbitrary cutoff value of 0.5 was used to calculate the predicted probably of getting converted on the train dataset.

Phase 5: Model evaluation:

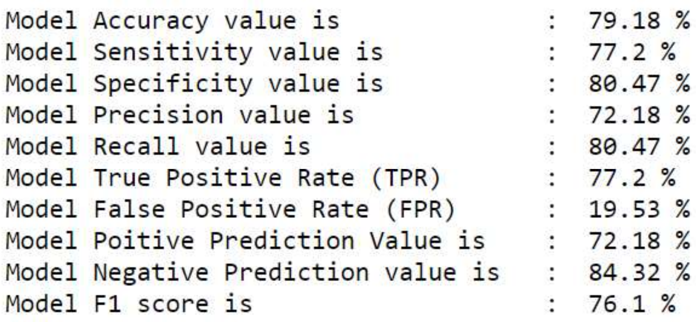
- Optimal threshold point was found using Accuracy, Sensitivity, and Specificity tradeoff graph.
- The conversion probability was calculated using the optimal cutoff point and model’s performance was checked over train dataset.



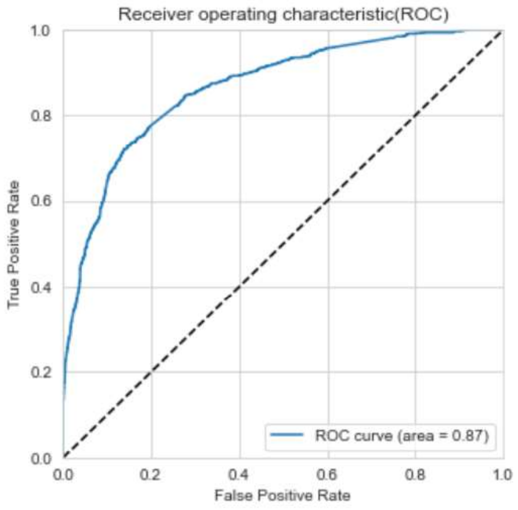
```
Model Accuracy value is              :  78.96 %
Model Sensitivity value is           :  77.55 %
Model Specificity value is           :  79.8 %
Model Precision value is             :  69.36 %
Model Recall value is                :  79.8 %
Model True Positive Rate (TPR)       :  77.55 %
Model False Positive Rate (FPR)      :  20.2 %
Model Poitive Prediction Value is    :  69.36 %
Model Negative Prediction value is   :  85.77 %
Model F1 score is                    :  74.22 %
```



- The model was tested using optimal cutoff point and the model performance was checked using test dataset.

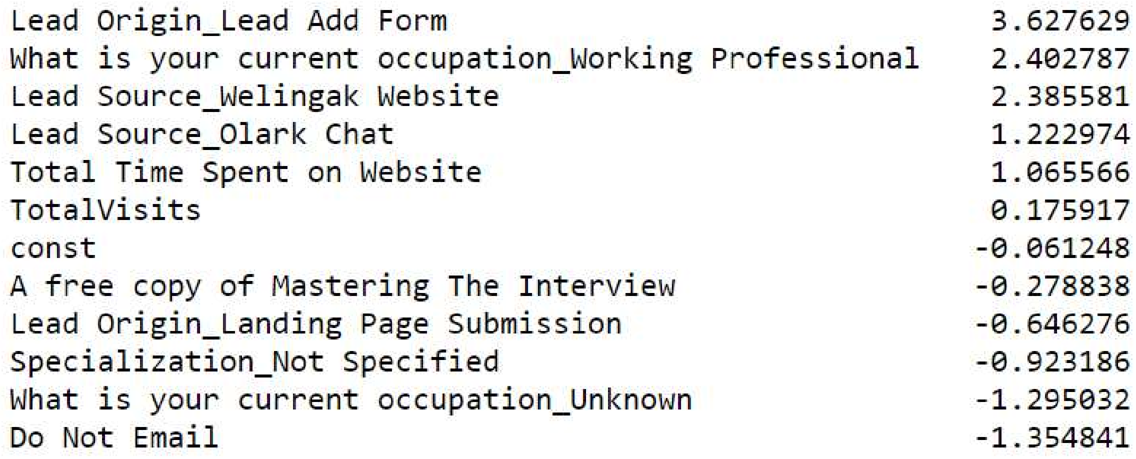


```
Model Accuracy value is              :  79.18 %
Model Sensitivity value is           :  77.2 %
Model Specificity value is           :  80.47 %
Model Precision value is             :  72.18 %
Model Recall value is                :  80.47 %
Model True Positive Rate (TPR)       :  77.2 %
Model False Positive Rate (FPR)      :  19.53 %
Model Poitive Prediction Value is    :  72.18 %
Model Negative Prediction value is   :  84.32 %
Model F1 score is                    :  76.1 %
```



- Finally, lead score value for each lead was calculated. Using this score, the “Hot Leads” for X Education were identified.

Driver variables for lead scoring



```
Lead Origin_Lead Add Form                               3.627629
What is your current occupation_Working Professional    2.402787
Lead Source_Welingak Website                            2.385581
Lead Source_Olark Chat                                  1.222974
Total Time Spent on Website                             1.065566
TotalVisits                                             0.175917
const                                                  -0.061248
A free copy of Mastering The Interview                 -0.278838
Lead Origin_Landing Page Submission                    -0.646276
Specialization_Not Specified                           -0.923186
What is your current occupation_Unknown                -1.295032
Do Not Email                                           -1.354841
```

Recommendations

The company should make calls to the leads who match the below criteria as these are more likely to get converted:

- lead sources such as Welingak Websites and Olark Chat.
- leads who spent more time on the websites or visit it often.
- leads who are the working professionals.
- leads which originate from "Add form".

The company should not make calls to the leads who match the below criteria as these are not likely to get converted:

- lead origin is "Landing Page Submission"
- leads whose Specialization was "Not Specified"
- leads whose Occupation is "Unknown"
- leads who chose the option of "Do not Email" as "Yes"
- leads who opt for a free copy of "Mastering the Interview" book.