# LEAD SCORING CASE STUDY

**Group member:**

1. **Ha Hoang Hai**
2. **Truong Hieu Thuan**

# Problem statement

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

Once these leads are acquired, employees from the sales team start making calls, writing emails, etc. Through this process, some of the leads get converted while most do not. The typical lead conversion rate at X education is around 30%.

To make this process more efficient, the company wishes to identify the most potential leads, also known as ‘Hot Leads’

X Education has appointed you to help them select the most promising leads. And final the CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%.

# Goals of the Case Study

There are quite a few goals for this case study:

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.

There are some more problems presented by the company which your model should be able to adjust to if the company's requirement changes in the future so you will need to handle these as well.

# Content implementation

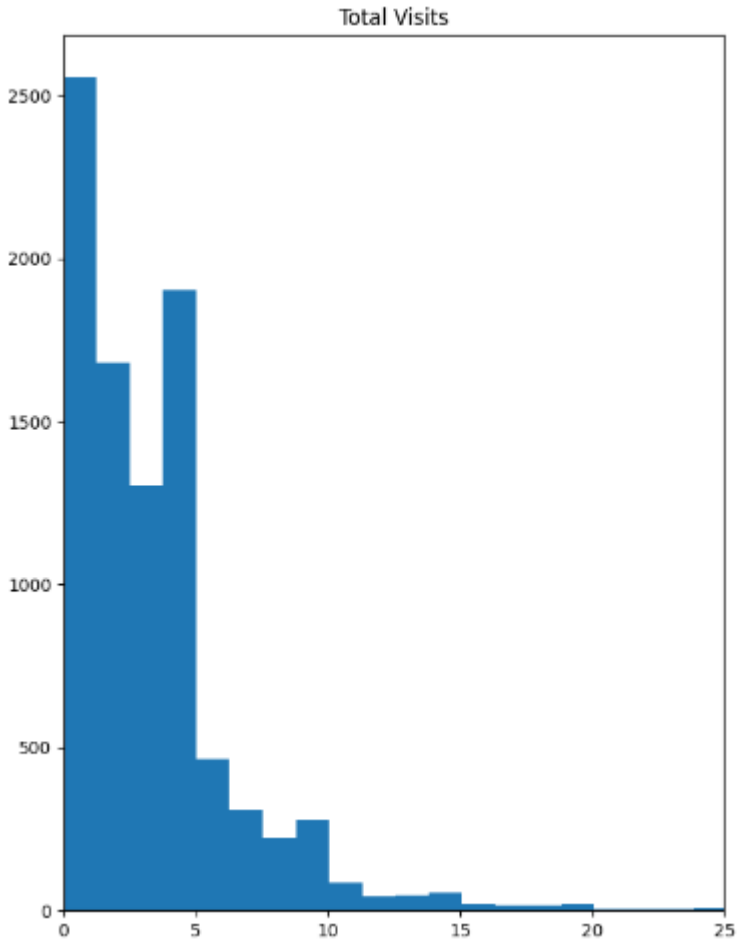
- Data Cleaning and Manipulation
- Exploratory data analysis
- Data preparation
- Splitting the data into training and testing sets
- Building a linear model
- Residual anylysis of the train data
- Making the confusion matrix
- Optimise cut off (ROC curve)
- Prediction on the test set
- Conclusion

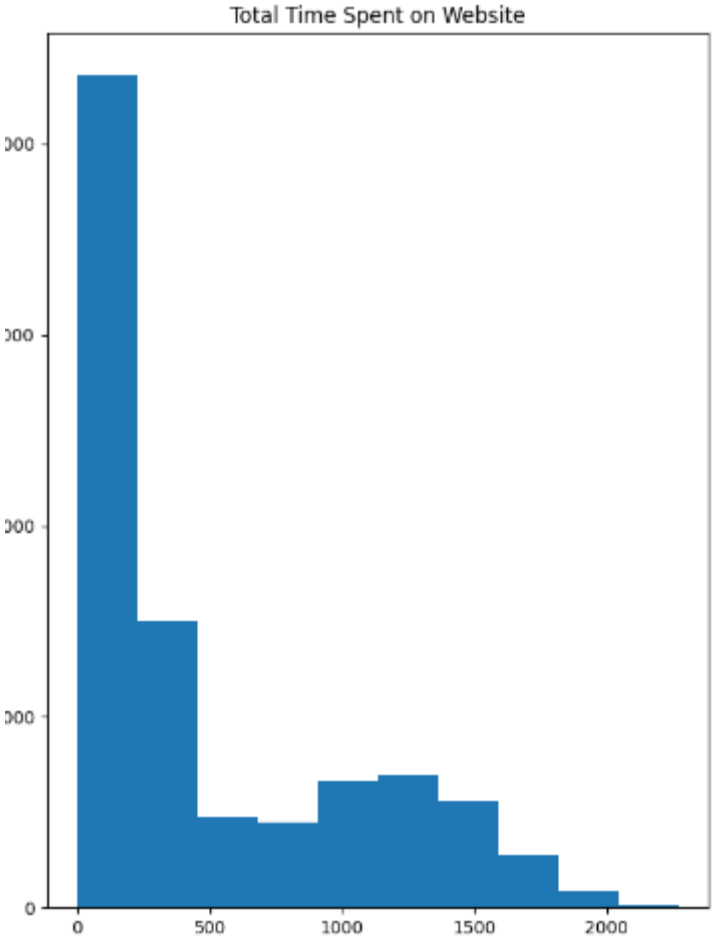
# Data Cleaning and Manipulation

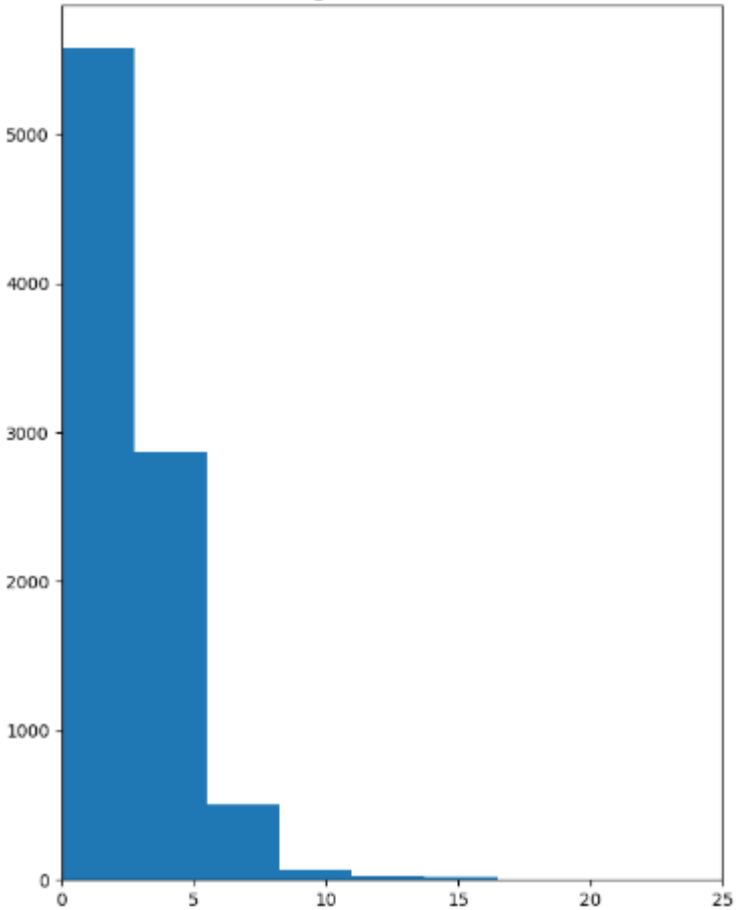
- The data have 9240 rows and total 37 columns
- We removed columns:
  - Have missing values more than 35%
  - Columns have unique value
- Which columns less than 30% null values, we inspected and update value for these columns
- After cleaning data, we have 9074 rows and total 22 columns

# Exploratory data analysis

**Univariate analysis**

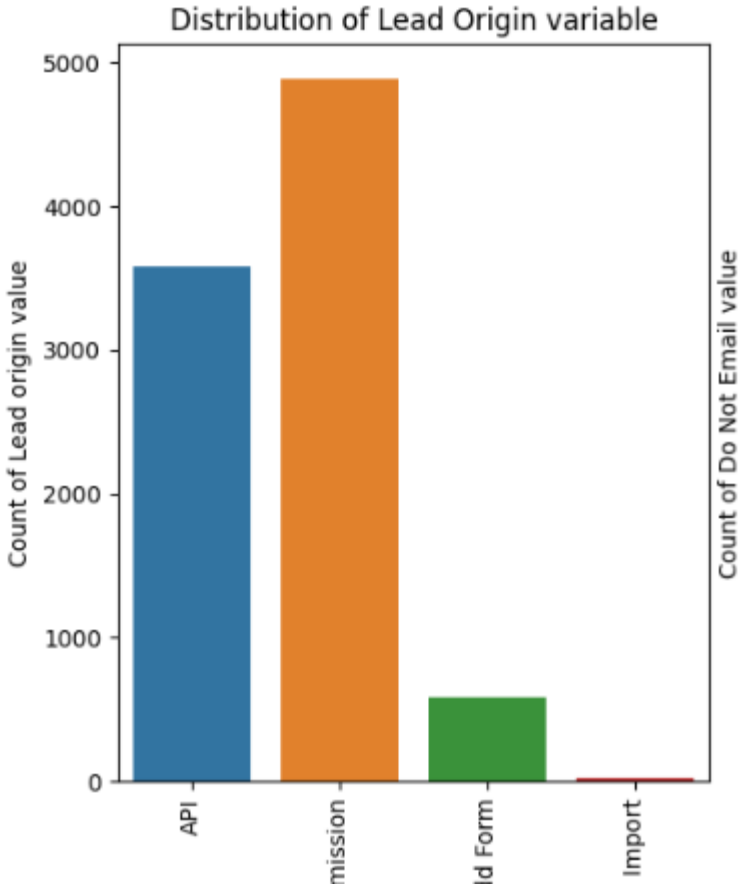


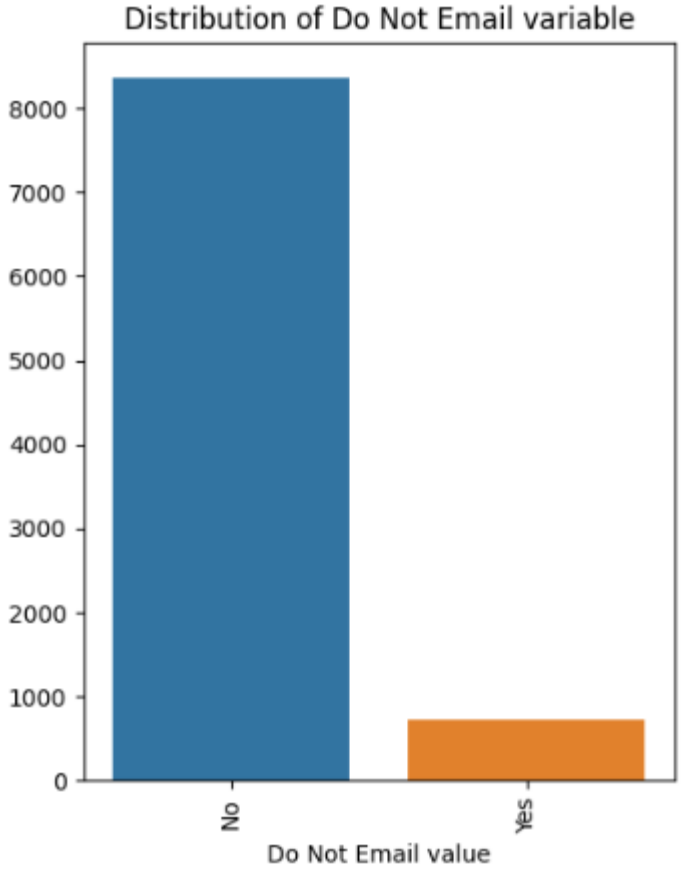



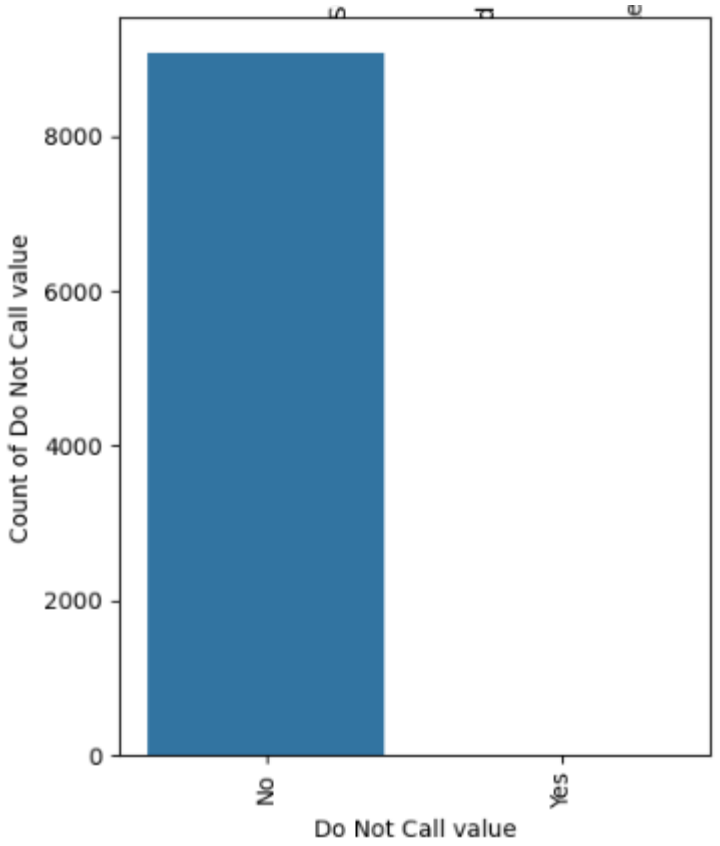



# Exploratory data analysis

**Visualising Categorical Variables**

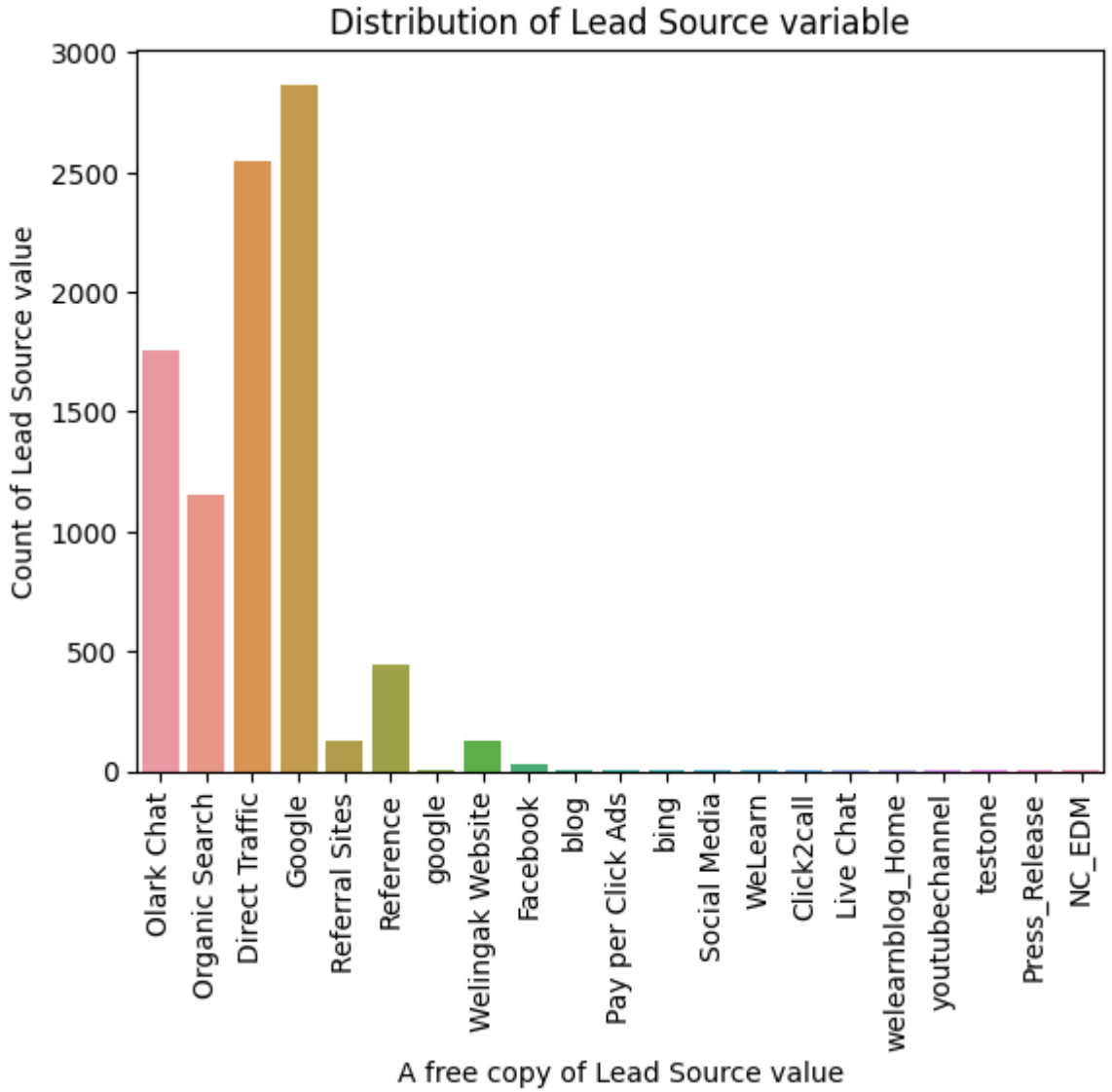


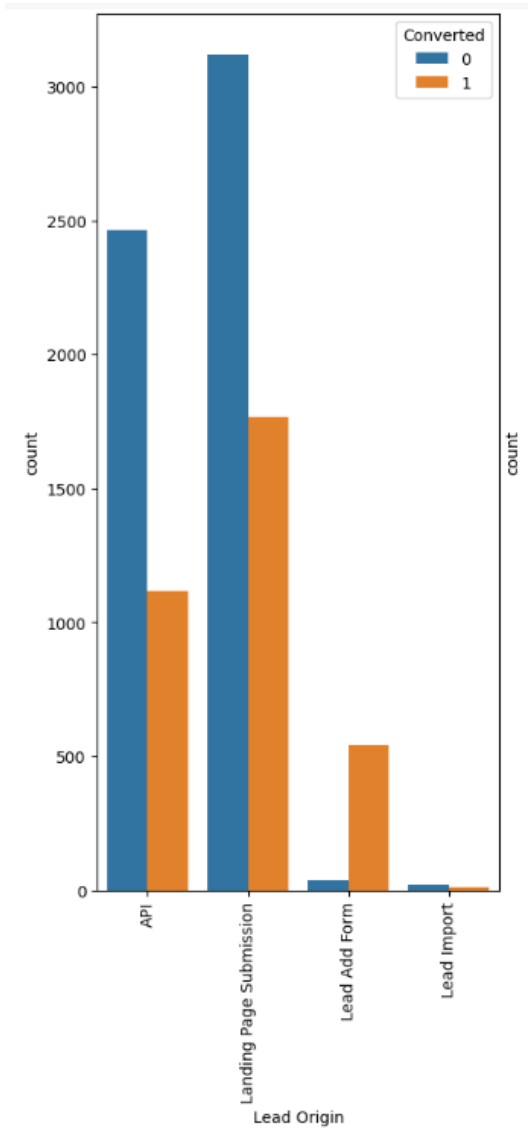
# Exploratory data analysis

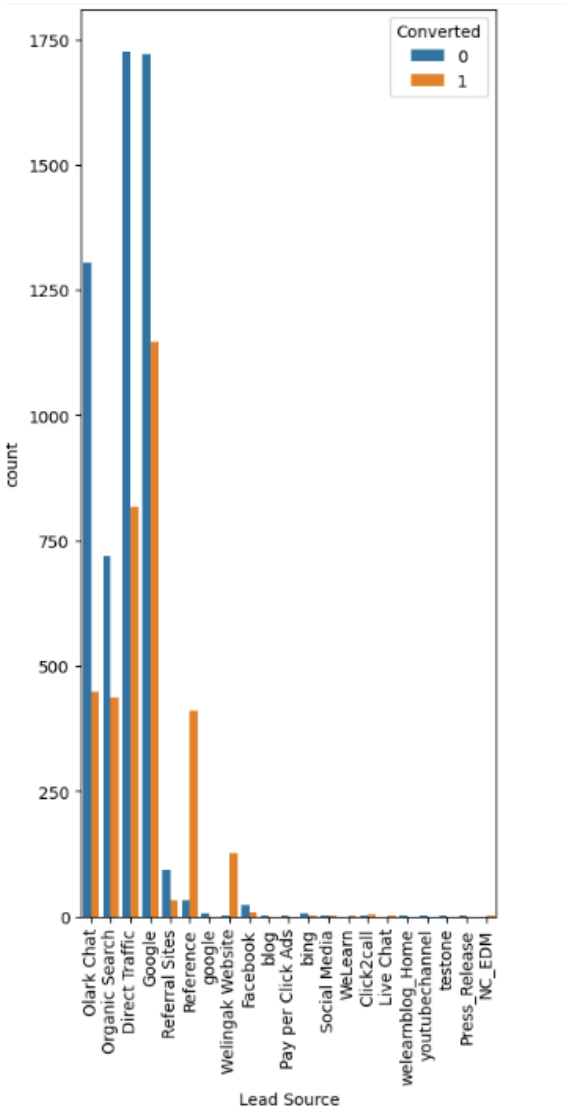
**Visualising Categorical Variables**



# Exploratory data analysis

**Associate all categorical variables with Converted**



# Data preparation

## Create Dummy variable

```
lead_dummy = pd.get_dummies(lead_data_final[['Lead Origin','Specialization' ,'Lead Source', 'Do Not Email', 'Last Activity',
'What is your current occupation','A free copy of Mastering The Interview', 'Last Notable Activity']], drop_first=True)
# Add the results to the master dataframe
lead_data_dum = pd.concat([lead_data_final, lead_dummy], axis=1)
lead_data_dum
```

## Drop columns unwanted

```
lead_data_dum = lead_data_dum.drop(['Lead Origin', 'Lead Source', 'Do Not Email', 'Do Not Call','Last Activity', 'Country',
'Specialization', 'Specialization_not provided','What is your current occupation','What matters most to you in choosing a
course', 'Search','Newspaper Article', 'X Education Forums', 'Newspaper','Digital Advertisement', 'Through Recommendations','A
free copy of Mastering The Interview', 'Last Notable Activity','City'], 1)
lead_data_dum
```

# Model building

- Split data into Train and Test Sets
- Building a linear model
- Building model using statsmode: have 24 model created
- Residual analysis of the train data

```
[147] y_train_pred = y_train_pred.values.reshape(-1)
      y_train_pred[:10]
```

```
array([0.66202482, 0.08136772, 0.18285728, 0.07964972, 0.17838694,
       0.89870285, 0.16339015, 0.98521708, 0.73250302, 0.25457765])
```

+ Code + Text

```
[148] y_train_pred_final = pd.DataFrame({'Converted':y_train.values,'Conversion_Prob':y_train_pred})
      y_train_pred_final.head()
```

| | Converted | Conversion_Prob |
|---|---|---|
| 0 | 1 | 0.662025 |
| 1 | 0 | 0.081368 |
| 2 | 0 | 0.182857 |
| 3 | 0 | 0.079650 |
| 4 | 0 | 0.178387 |

```
[149] y_train_pred_final['Predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x >0.5 else 0)
      y_train_pred_final.head()
```

| | Converted | Conversion_Prob | Predicted |
|---|---|---|---|
| 0 | 1 | 0.662025 | 1 |
| 1 | 0 | 0.081368 | 0 |
| 2 | 0 | 0.182857 | 0 |
| 3 | 0 | 0.079650 | 0 |
| 4 | 0 | 0.178387 | 0 |

# Model building

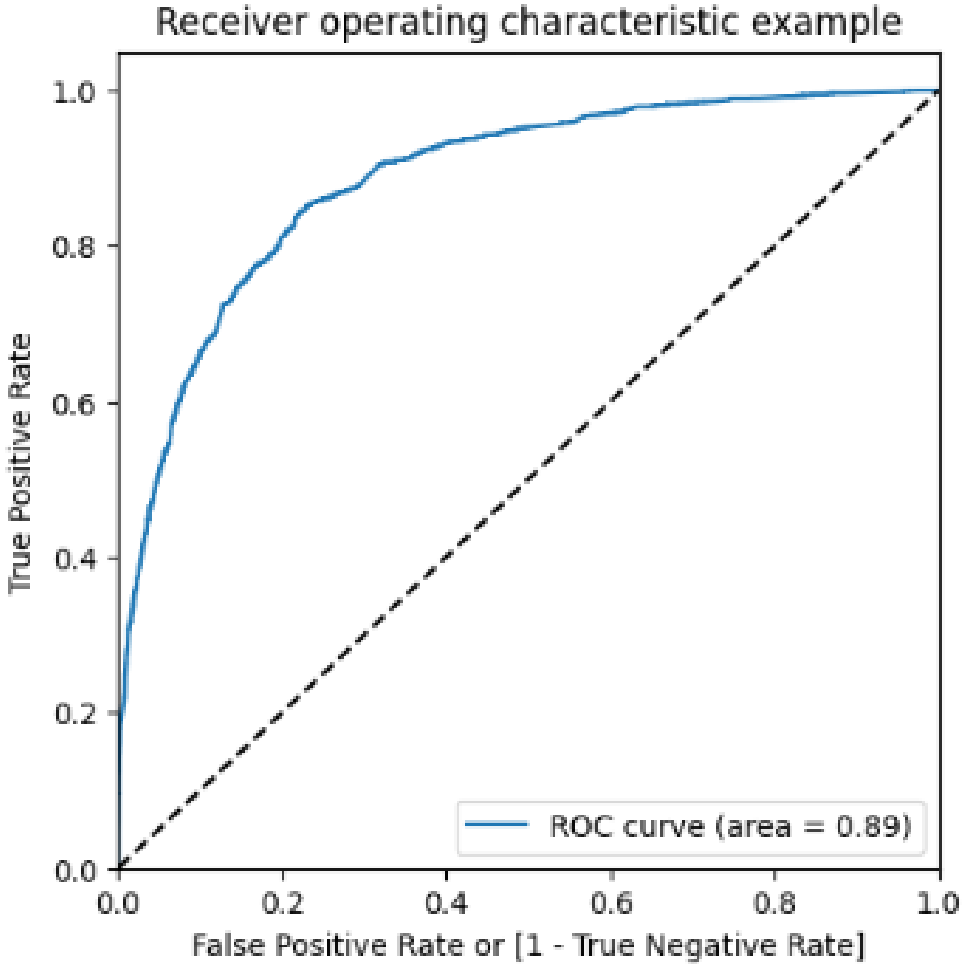
Making the confusion matrix:

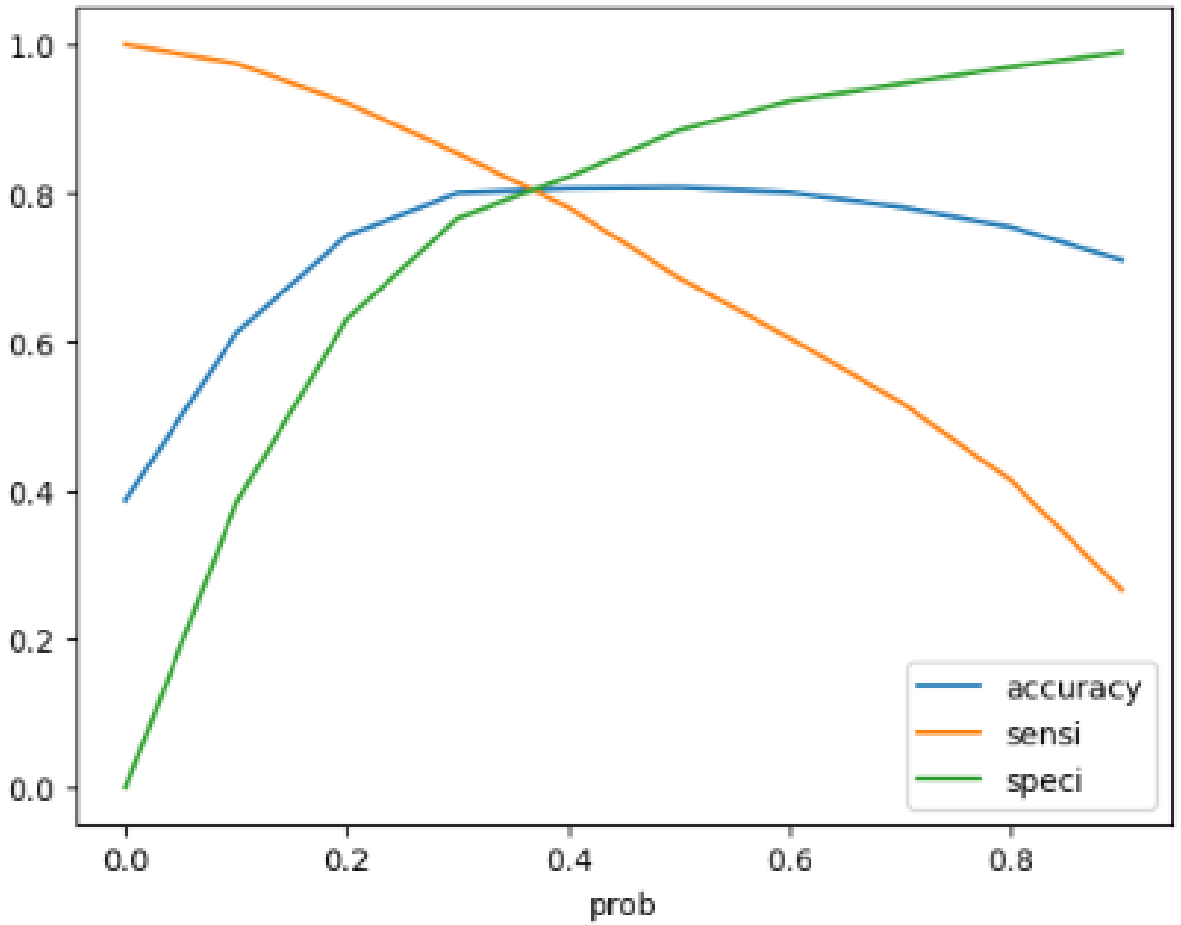
```
[151] # Confusion matrix

      confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Predicted)
      print (confusion)

      [[3443  452]
       [ 772 1684]]
```

We found that:

- Accuracy was good: 80%
- Specificity was good: 88%
- Sensitivity not good, it only: 68%

# Optimise Cut Off (ROC Curve)




*Insight:*

- *We have higher (0.89) area under the ROC curve, therefore our model is a good value.*
- *From the curve above, 0.36 is the optimize point to take it as a cutoff probability*

# Conclusion

First of all, in the process of searching for the best model, we found that there are too many variables affecting the selection process, leading to too many models to run. In general, running many test models in this case we find is not optimal.
However, within the requirements we also found variables that influence potential buyers:

- The total time spend on the Website.
- Total number of visits.

Lead source (In descending order):

- Google
- Direct traffic
- Organic search
- Welingak website

Priority Active:

- SMS
- Chat conversation

Priority should be given to the following industries to have a high chance of achieving leads:

- Working professional

# THANK YOU