# LEAD SCORING

Presented by:

-Urvashi Singh

-Nitish Rathore

# CONTENT:

- Problem Statement
- Data Inspection
- Missing value treatment
- Analyzing different Variables
- Checking Outliers
- Dummy variable and train-test division
- Model Building
- Model Evaluation
- Conclusion

# PROBLEM STATEMENT

- An education firm named X Education has appointed us to help them select the most promising leads, i.e. the leads that are most likely to convert into paying customers.
- The company requires to build a model wherein we need to assign a lead score to each of the leads such that the customers with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance. The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%.

# DATA INSPECTION

- First of all we have imported all the libraries required to build the logistic regression model for assigning a score to every lead which are likely to be converted where we can assign a lead score to each of the leads such that the customers with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance.
- We further proceeded by importing the dataframe and inspected its size and shape.
- Using .describe() method we got the mean, median and standard deviation for the numerical variables.
- After this we have checked for the null values or missing values.

# MISSING VALUE TREATMENT

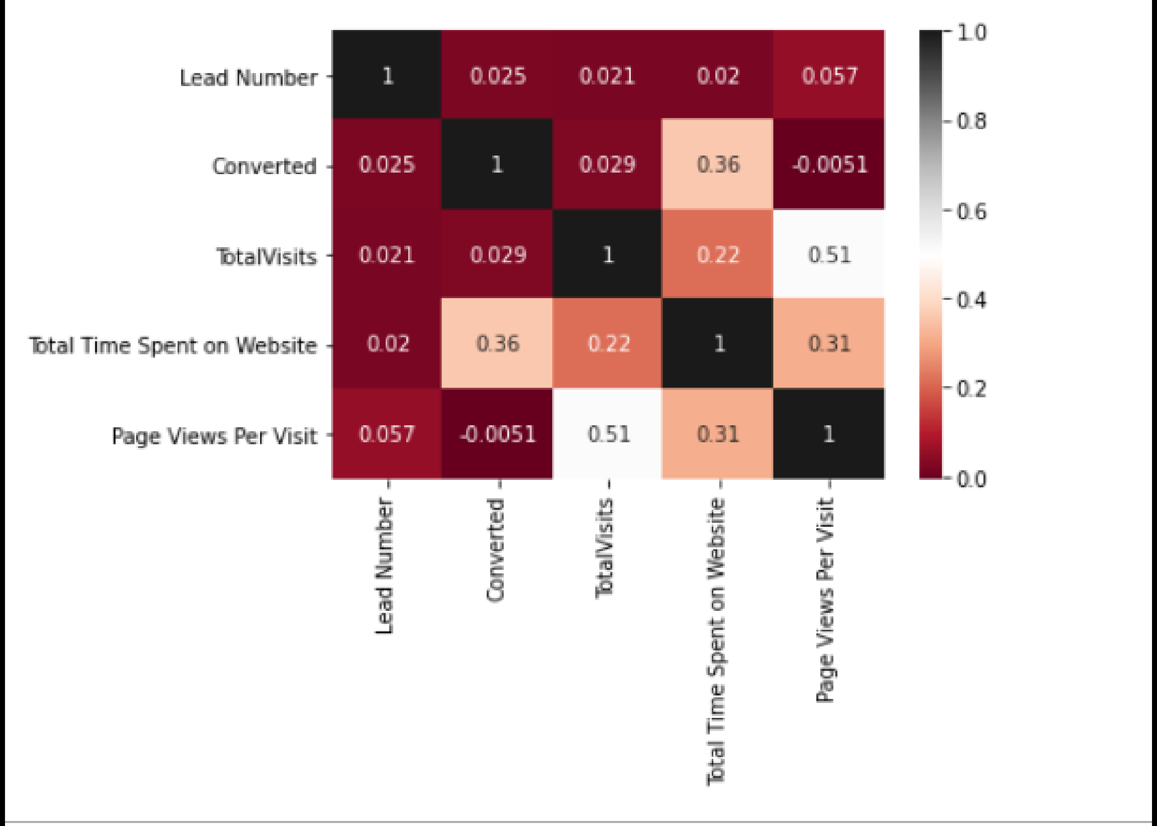
- We observed that there is a value "select" in many columns which is where customer have not selected any option. So, here we have converted "select" to "NaN“.
- After this we calculated the percentage of missing values and dropped those columns which have more than 50% missing values and having only one category value as these would not have any impact in further prediction.
- We also dropped all the sales related columns as they are not necessary for the model and also having high missing values.
- We have dropped 'What matters most to you in choosing a course’ column as it doesn’t seem to be much significant after looking at its values.

# CONT…

- We checked the value counts of the categorical variables and also checked the skewness, and dropped the columns with high skewness.
- Again we checked for the missing values after dropping the unnecessary variables.
- Hereafter we imputed the missing value of the categorical columns with its mode and continuous variables with median.

# ANALYZING DIFFERENT VARIABLES

- After missing value treatment we have calculated the correlation between different variables which has given the following result:

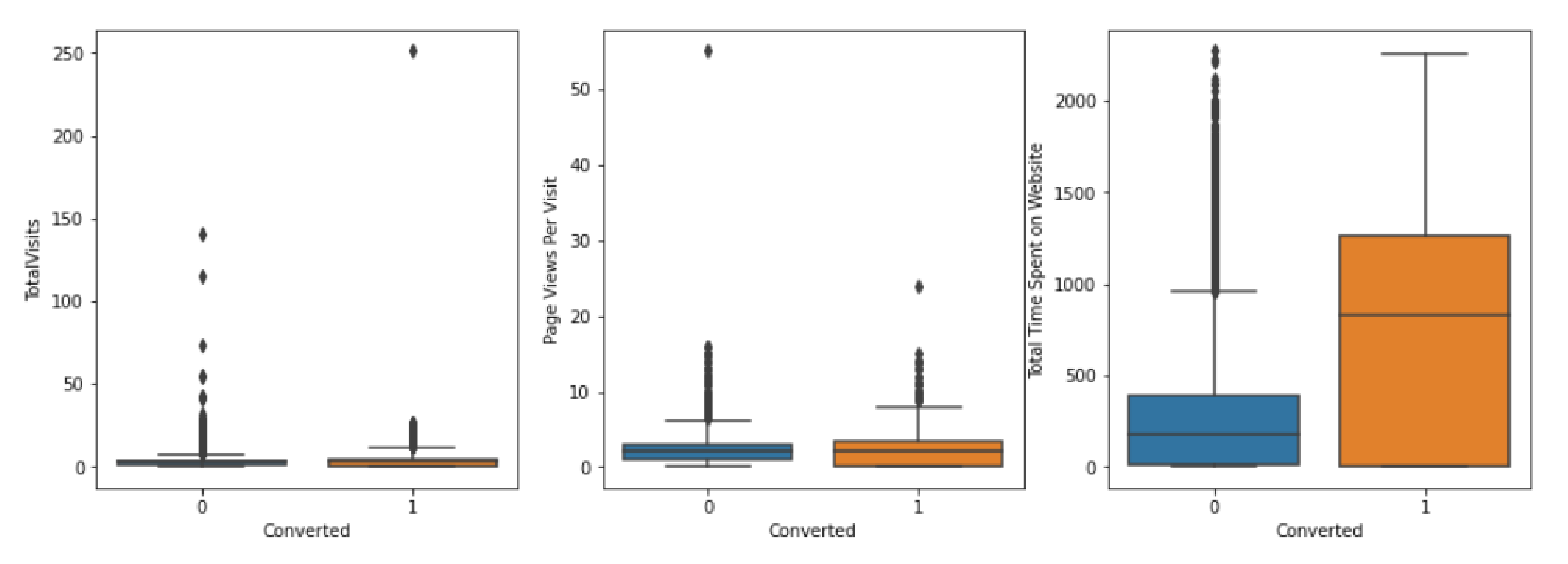


| | Lead Number | Converted | TotalVisits | Total Time Spent on Website | Page Views Per Visit |
|---|---|---|---|---|---|
| Lead Number | 1 | 0.025 | 0.021 | 0.02 | 0.057 |
| Converted | 0.025 | 1 | 0.029 | 0.36 | -0.0051 |
| TotalVisits | 0.021 | 0.029 | 1 | 0.22 | 0.51 |
| Total Time Spent on Website | 0.02 | 0.36 | 0.22 | 1 | 0.31 |
| Page Views Per Visit | 0.057 | -0.0051 | 0.51 | 0.31 | 1 |

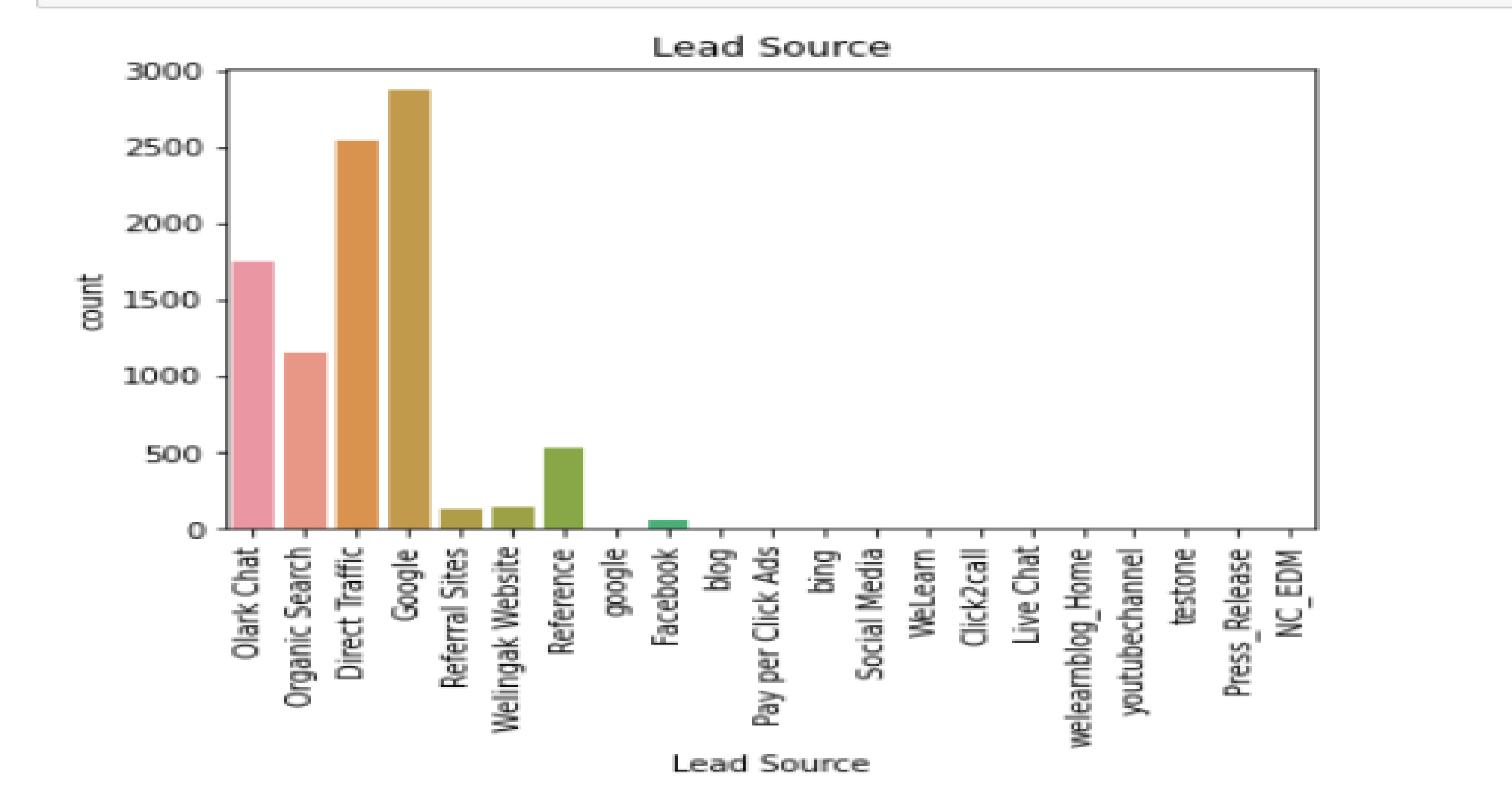
# CONT…

- We have performed some EDA on the numerical variables by making subplots and box plots.
- From these box plots we saw how our target variable ‘Converted’ depends on other variables like ‘TotalVisits’, ‘Page Views Per Visit’ and "Total Time Spent on Website“.
- We further analysed the ‘Lead Source’ variable and found that "Google" has the highest count and NULL values are only 0.39 % so we have safely replaced them with "Google"

# CHECKING OUTLIERS



# DISTRIBUTION OF LEAD SOURCE



# DUMMY VARIABLE AND TRAIN-TEST DIVISION

- Once the null values are treated and we found that there are no missing values anymore, we then have created the dummy variables for the variables having two or more distinct categories in order to correctly analyze attribute variables.
- After this we have split the data into train and test data set, where train dataset consists of 70% of the data and test data contains 30% of the data, and we have then used the min-max scaler to scale the numerical variables of the training dataset.
- Once scaling is done, we will proceed to build the model.

# MODEL BUILDING

- We have used statsmodel  and sklearn library here in order to look at the statistics part and to be able to select the features better.
- As statsmodel does not add a constant or an intercept by default we have added a constant and have provided the family as binomial.
- Then we will fit the model and will check its summary to get the number of observations and features.
- We checked for the P-values and coefficients, then doing some course tuning, we have eliminated the variables and selected the best 18 variables.
- After this we will be using recursive feature elimination to eliminate the variables one by one, this we call the shortlisting of the variables.

# RFE, VIF & P-Value Check

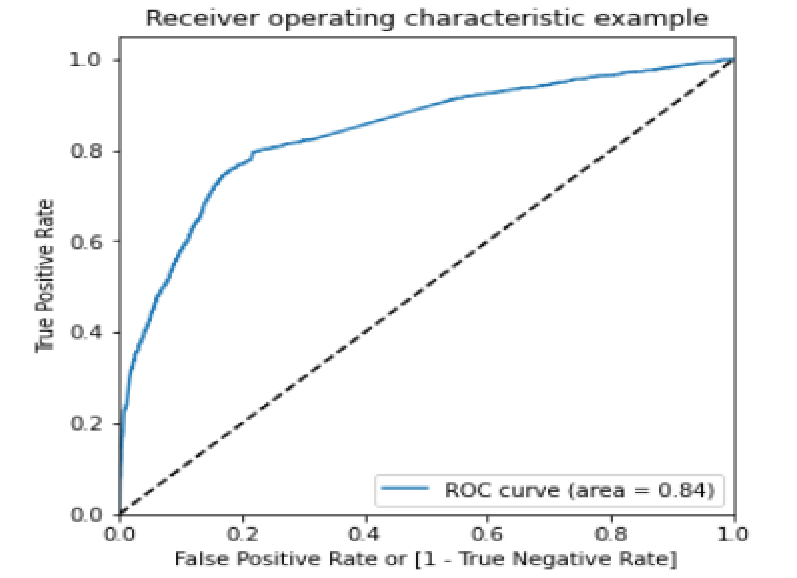
- After this we have Created X_train dataframe with RFE selected variables and added a constant to the X_train dataframe and again built the model of the X training dataset.
- We checked for the summary of the resulting model for the P-values and also checked the VIF (variance inflation factor) by creating a VIF dataframe.
- Further we have dropped the variables one by one with high P-values first and also considering their VIFs.

# Cont..

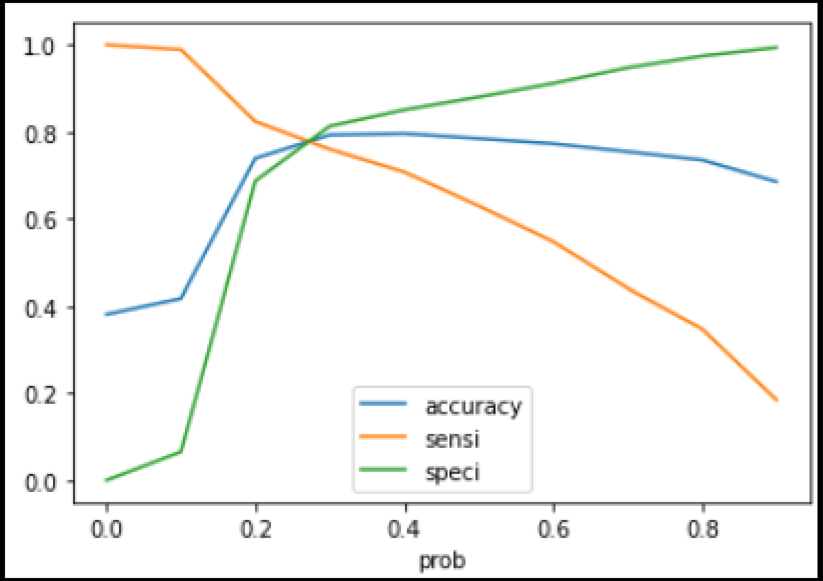
- Then after 9 iterations i.e after dropping 9 variable we have reached to a point where we can predict as to whether a lead will be converted or not and also by checking the probability of its conversion.
- Here we have set the cut-off as 0.5 for the probability i.e we have assigned a 1 to the leads which have greater than 0.5 probability and assigned 0 to those who has a probability of less than 0.5.

# MODEL EVALUATION

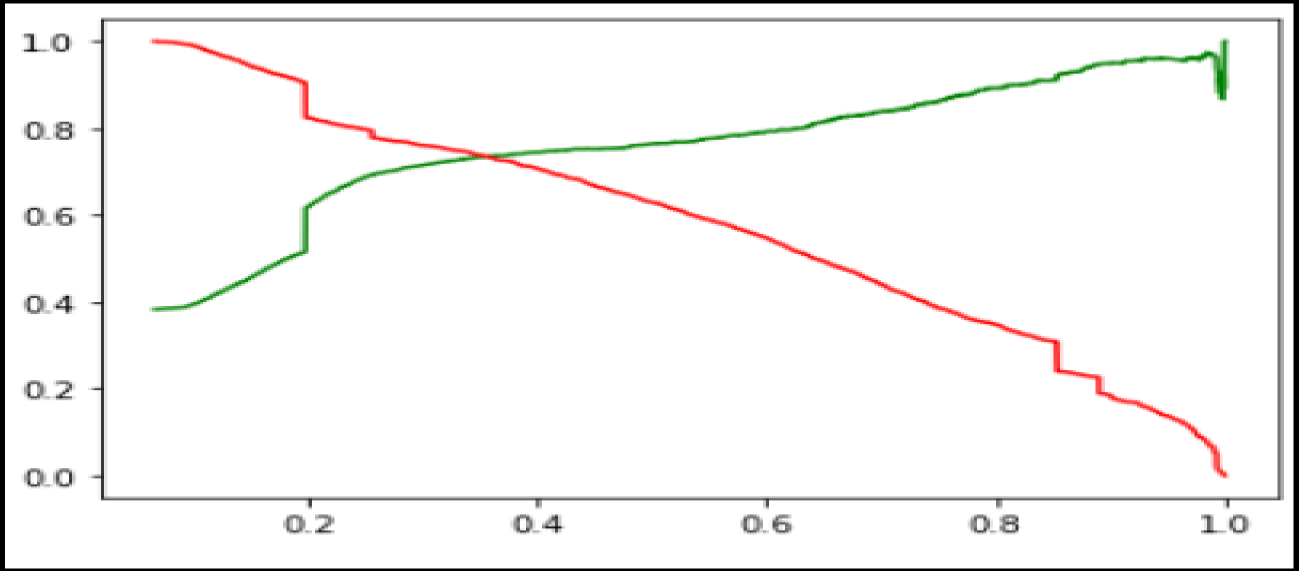
- In evaluating any model we check how good our model is. For this we have created a confusion matrix first and checked the overall accuracy.
- After this we have calculated the true positive, true negative, false positive and false negative values to calculate the Sensitivity and Specificity and also drew the ROC curve.



- To find the optimal cut-off value we have first created the columns with different probabilities and looked at the sensitivity, accuracy, precision and specificity.
- We have then plotted the Sensitivity, Specificity and accuracy graph which gives a cut-off value of 0.3.



- From the graph taking the optimal cut-off value as 0.30 we have created another column and assigned a value of 0 and 1 based on the cut-off value.
- We have again checked the accuracy of the model and have got the recall curve as follows where the cut-off comes to be around 0.4:



- We scaled the numerical variables using Min-Max Scaler of the training data-set and splitted the test data into y and x.
- Then we have taken all the columns from the final model and predicted the value of y.
- We then created Data frame with given conversion rate and probability of predicted ones and made prediction using cut off of 0.30.
- Finally, we have generated the lead score and lead number and checked the overall accuracy and other parameters.

# CONCLUSION

- From the final model developed we are able to generate a lead score which shows how likely the lead is getting converted.
- From the final predictions it is evident that the people with lead score more than 30 are likely to get converted and sales team can concentrate people with lead score. more than 30 to convert them to a lead.