Lead score Assignment

OBJECTIVES

- An Education company selling online courses to professionals.
- the company gets leads in two ways:
  - Professionals landing to company website who either fill up the form with email and phone and watch relevant videos.
  - Through referrals.
- The company receives lots of leads but the lead conversion rate is very poor (30%)
- the objective is to identify the hot leads who are like likely to be converted, so the lead conversion rate gets better by building a classification model (logistic regression model) and assign a lead score
- Higher the lead score, higher the chances of it being a hot lead
- the target lead score conversion rate must be around 80%

DATA CLEANING: Fixing Columns

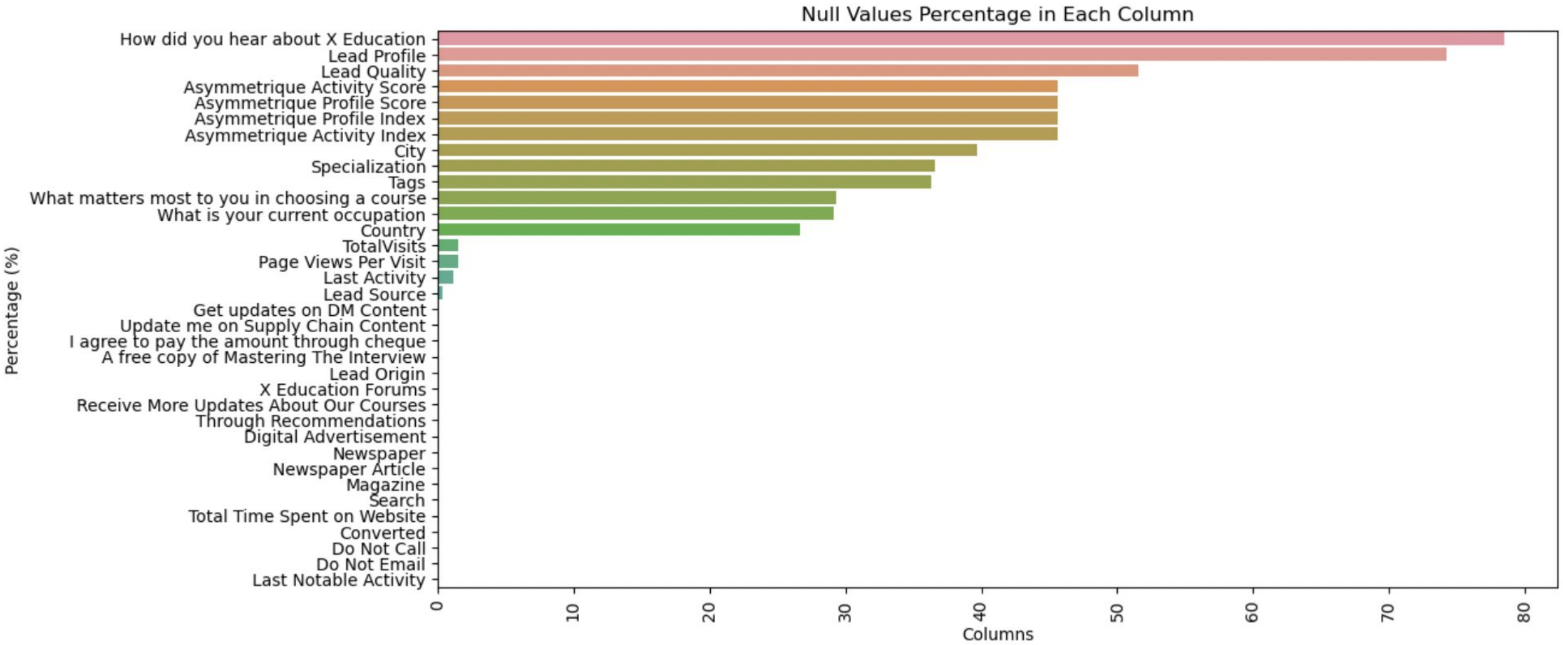
- there are no missing column names in the given dataset
- on high level all the columns looks important for analysis
- the given column names are self explanatory and there is no need to be renamed
- there are no columns that can be split to have more unique multiple columns
- there are no columns that can be merged to have a unique column

DATA CLEANING: Fixing Rows

- there are no unnecessary header rows and footer rows in the given dataset
- After going through the dataset manually, there are no summary total or subtotal rows in the given dataset
- After going through the dataset manually, there is no column descriptor row in the given dataset
- there are no blank rows in the given dataset

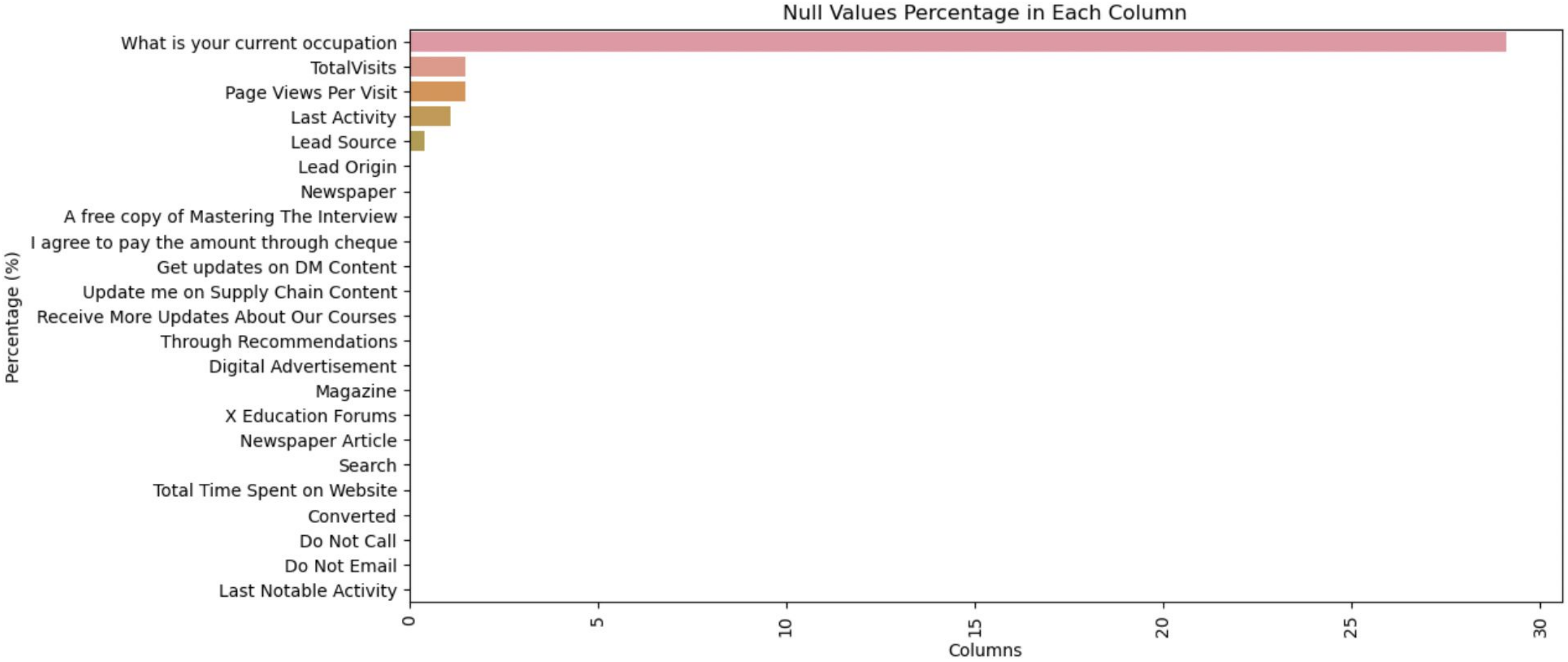
DATA CLEANING: Missing values

- After converting Select values to NaN, this is the count of missing values in percentage



DATA CLEANING: Missing values

Dropping columns whose missing values are around close to 30%, after dropping the count of missing values in %

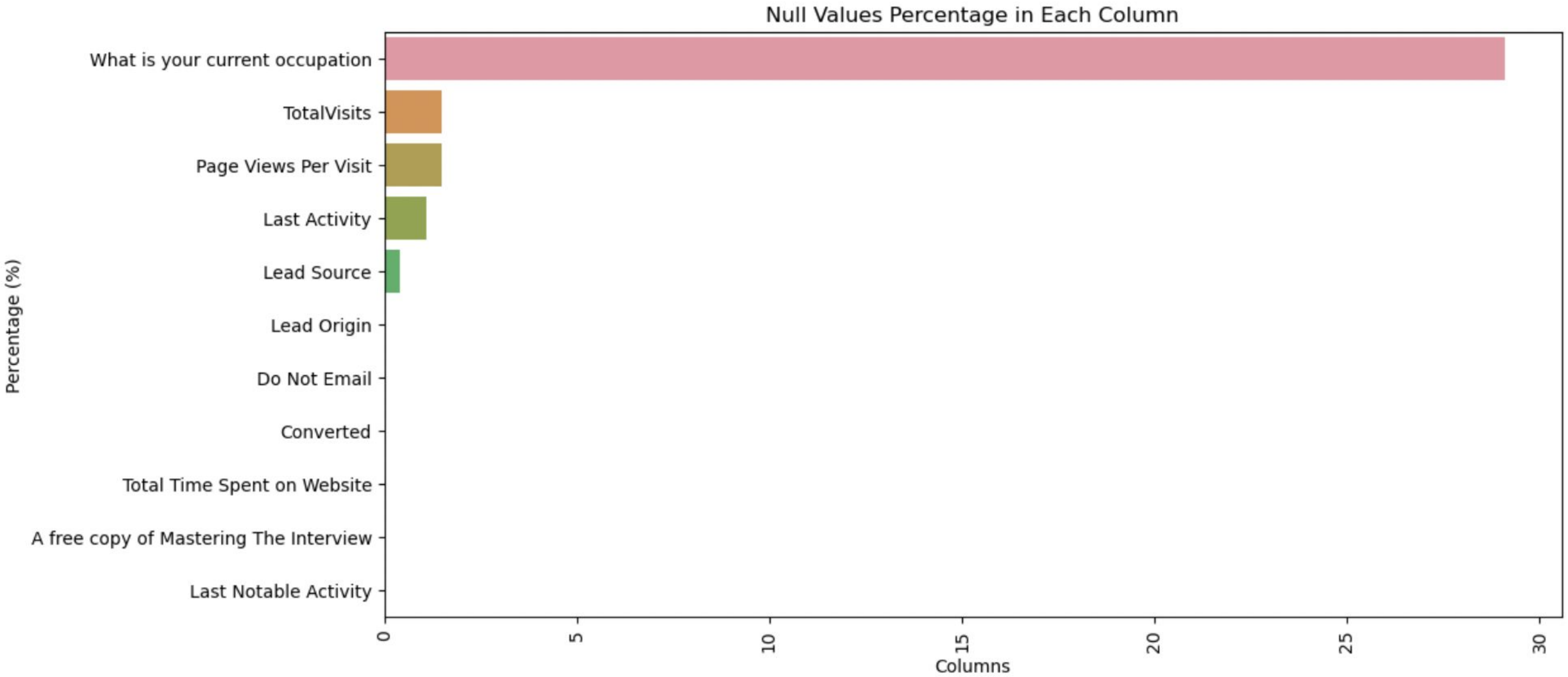


DATA QUALITY CHECK

- columns **["Do Not Call", "Search", "Magazine", "Newspaper Article", "X Education Forums", "Newspaper", "Digital Advertisement", "Through Recommendations", "Receive More Updates About Our Courses", "Update me on Supply Chain Content", "Get updates on DM Content", "I agree to pay the amount through cheque"]** have zero variance and will not have any impact on our model, hence we can drop these columns

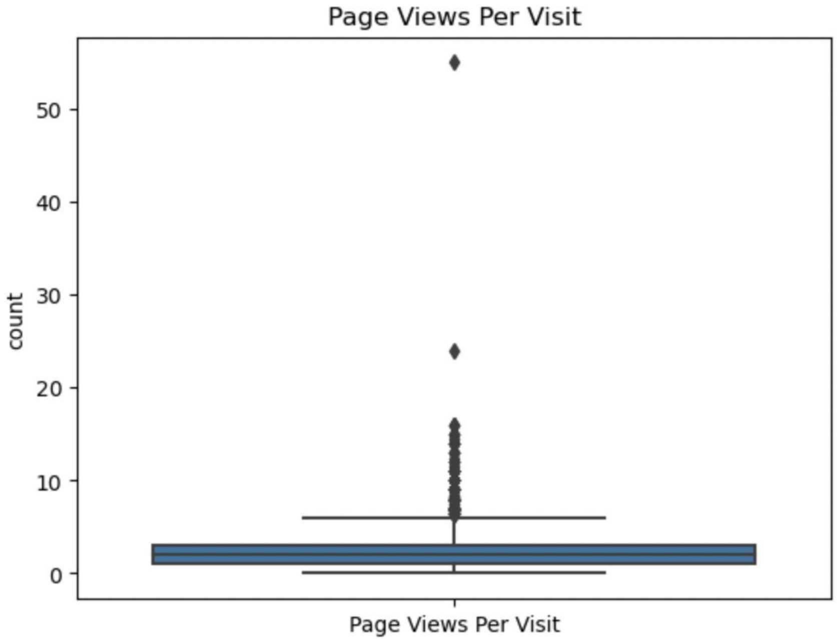
DATA QUALITY CHECK

After dropping columns with zero variance, we are left with 11 columns



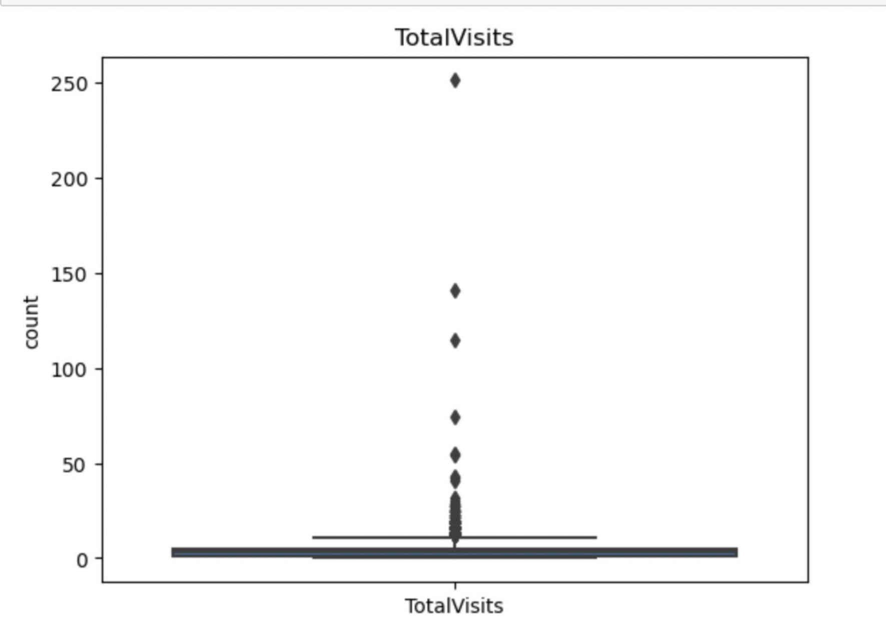
DEALING WITH OUTLIERS: Page views per Request

- outliers detected for **Page Views Per Visit** but they are well within realistic limits, hence we cannot treat them as outliers



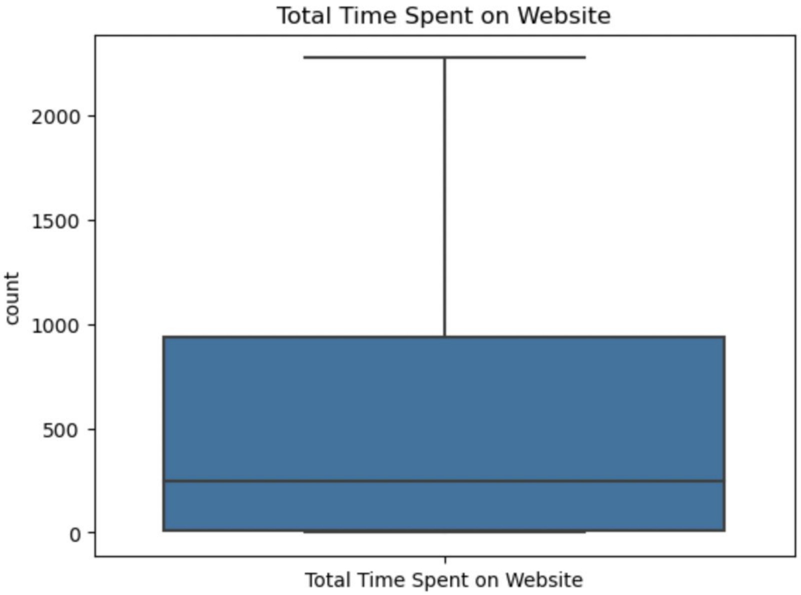
DEALING WITH OUTLIERS: TotalVisits

- outliers detected for **TotalVisits** but they are well within realistic limits, hence we cannot treat them as outliers



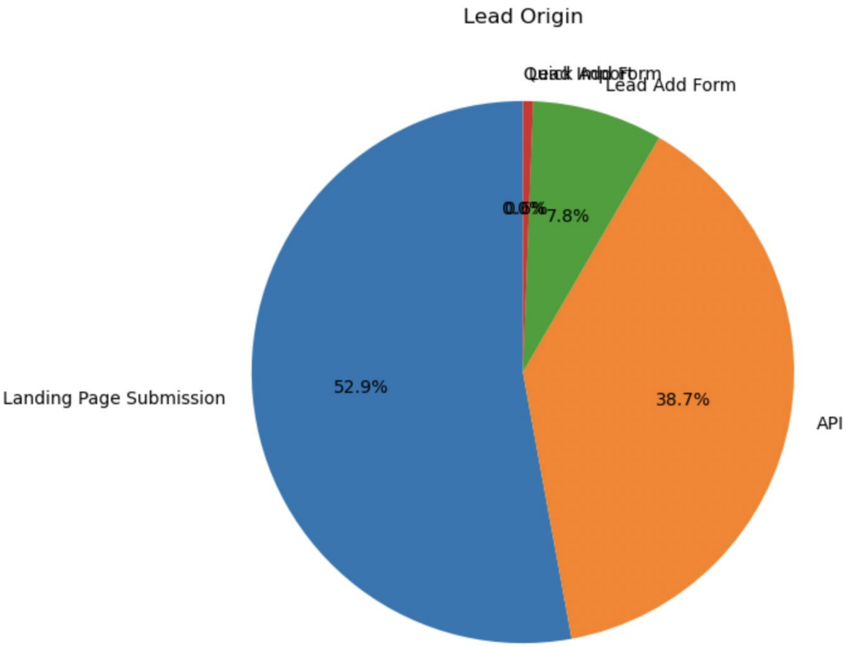
DEALING WITH OUTLIERS: Total Time Spent on Website

- No outliers detected in column **Total Time Spent on Website**



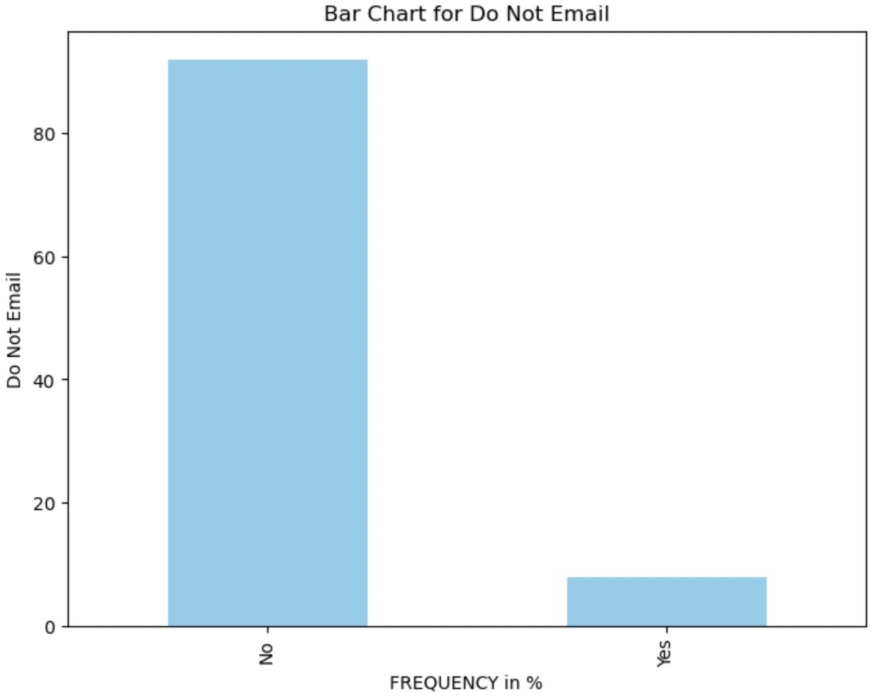
UNIVARIATE ANALYSIS: Lead Origin

- majority of the leads have originated from landing page submission and API



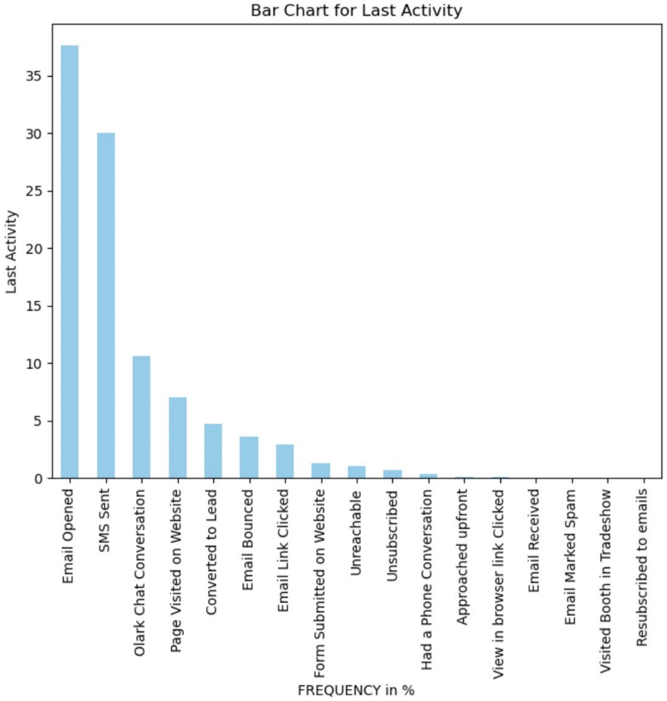
UNIVARIATE ANALYSIS: Do Not Email

92% of the leads have opted for No to "Do not Email"



UNIVARIATE ANALYSIS: Last Activity

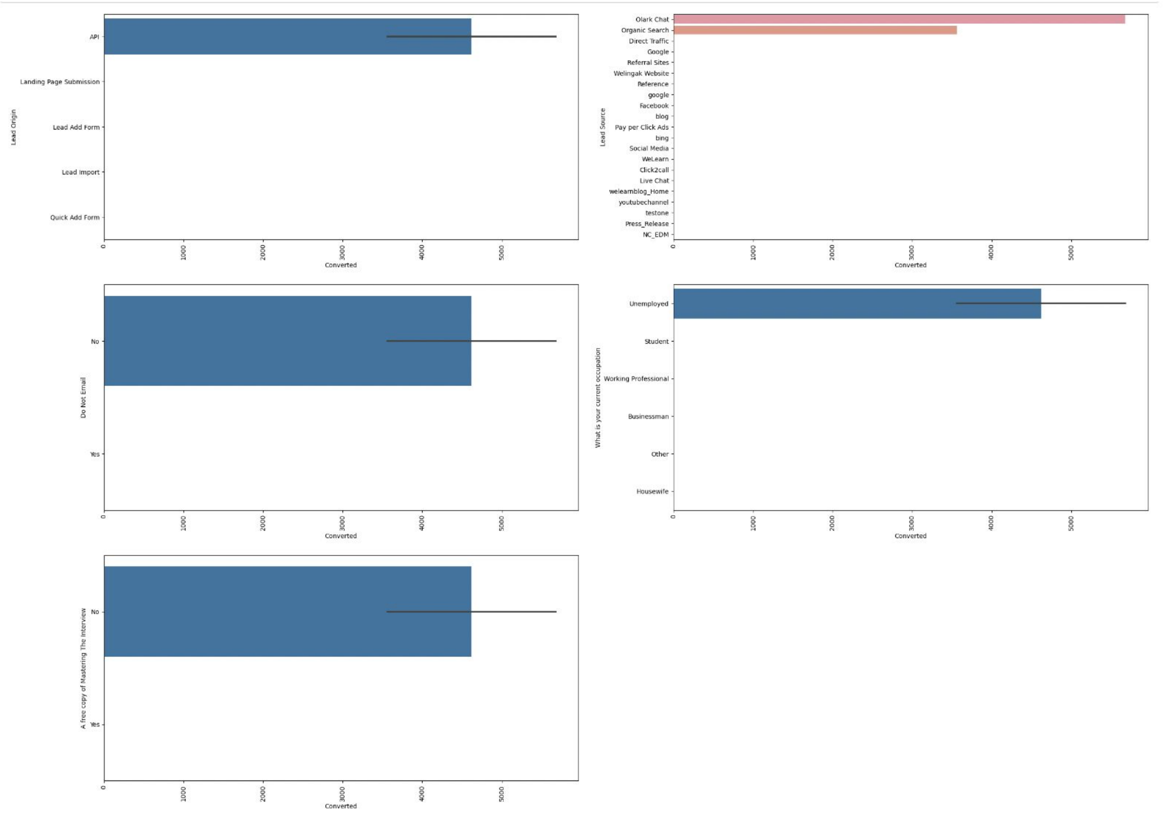
Email opened (37.62%) and SMS sent (30%) contribute to majority of 'Last Activity' by the leads



BIVARIATE ANALYSIS: with target variable

- If the lead is originated from API source then they are most likely to be converted
- If the lead is sourced from Olark chat or Organic Search then they are most likely to be converted
- if the lead didn't opt for No to Email, they are most likely to be converted
- if the lead is unemployed, then they are more likely to be converted
- if the lead didn't opt for A free copy of Mastering The Interview, they are most likely to be converted

BIVARIATE ANALYSIS: with target variable

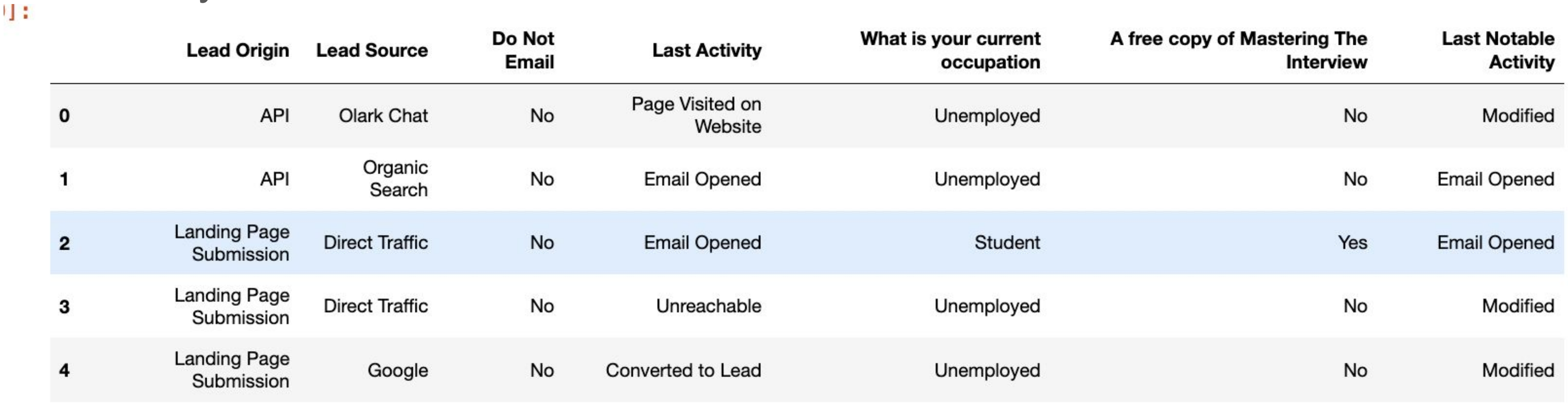


EDA OBSERVATIONS:

- majority of the leads have originated from landing page submission and API
- If 92% of the leads have opted for No to "Do not Email"
- Email opened (37.62%) and SMS sent (30%) contribute to majority of 'Last Activity' by the leads
- If the lead is originated from API source then they are most likely to be converted
- If the lead is sourced from Olark chat or Organic Search then they are most likely to be converted
- if the lead didn't opt for No to Email, they are most likely to be converted
- if the lead is unemployed, then they are more likely to be converted
- if the lead didn't opt for A free copy of Mastering The Interview, they are most likely to be converted

BUILDING MODEL: Creating Dummy variables

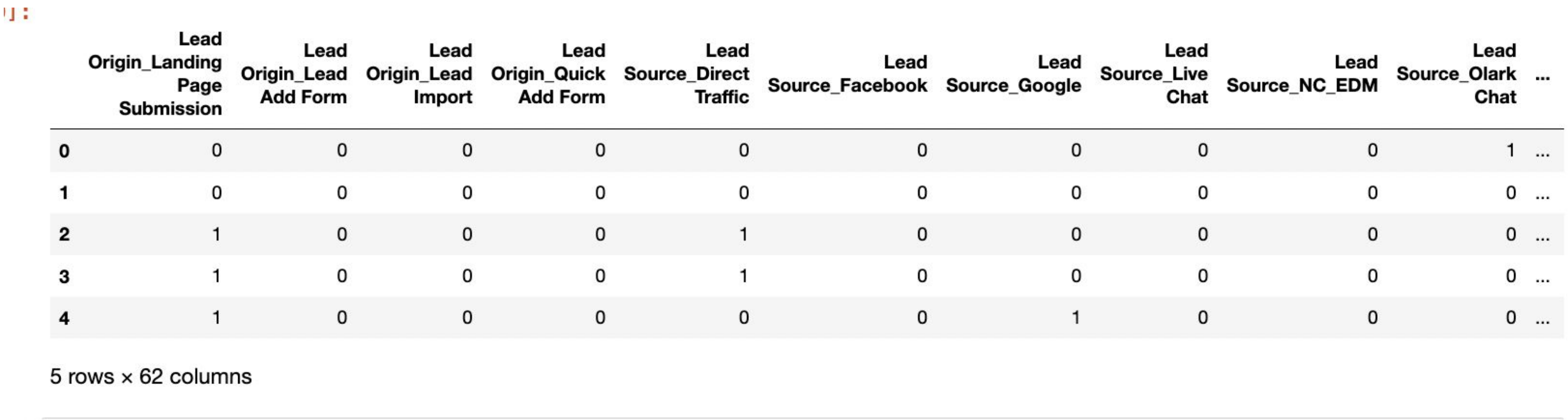
- Since we have categorical variables and the machine learning algorithm deals with values in numerical inputs. So we create dummy variables
- We have 7 categorical variables in our dataset for which we have to create dummy variables.



	Lead Origin	Lead Source	Do Not Email	Last Activity	What is your current occupation	A free copy of Mastering The Interview	Last Notable Activity
0	API	Olark Chat	No	Page Visited on Website	Unemployed	No	Modified
1	API	Organic Search	No	Email Opened	Unemployed	No	Email Opened
2	Landing Page Submission	Direct Traffic	No	Email Opened	Student	Yes	Email Opened
3	Landing Page Submission	Direct Traffic	No	Unreachable	Unemployed	No	Modified
4	Landing Page Submission	Google	No	Converted to Lead	Unemployed	No	Modified

BUILDING MODEL: Creating Dummy variables

- We have created 62 dummy columns, now we have to merge them with actual dataset and drop all the categorical columns

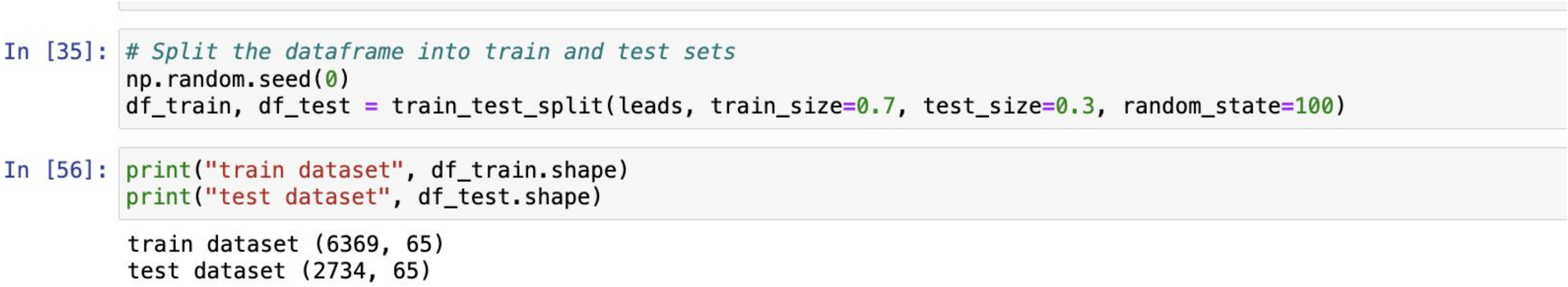


	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Origin_Quick Add Form	Lead Source_Direct Traffic	Lead Source_Facebook	Lead Source_Google	Lead Source_Live Chat	Lead Source_NC_EDM	Lead Source_Olark Chat	...
0	0	0	0	0	0	0	0	0	0	1	...
1	0	0	0	0	0	0	0	0	0	0	...
2	1	0	0	0	1	0	0	0	0	0	...
3	1	0	0	0	1	0	0	0	0	0	...
4	1	0	0	0	0	0	1	0	0	0	...

5 rows × 62 columns

BUILDING MODEL: Split dataset (7:3)

- After splitting the data set with a 70% data as training data and 30% data as the testing data we have
  - 6468 rows in train dataset
  - 2734 rows in test dataset



```
In [35]: # Split the dataframe into train and test sets
         np.random.seed(0)
         df_train, df_test = train_test_split(leads, train_size=0.7, test_size=0.3, random_state=100)
```

```
In [56]: print("train dataset", df_train.shape)
         print("test dataset", df_test.shape)
```

```
train dataset (6369, 65)
test dataset (2734, 65)
```

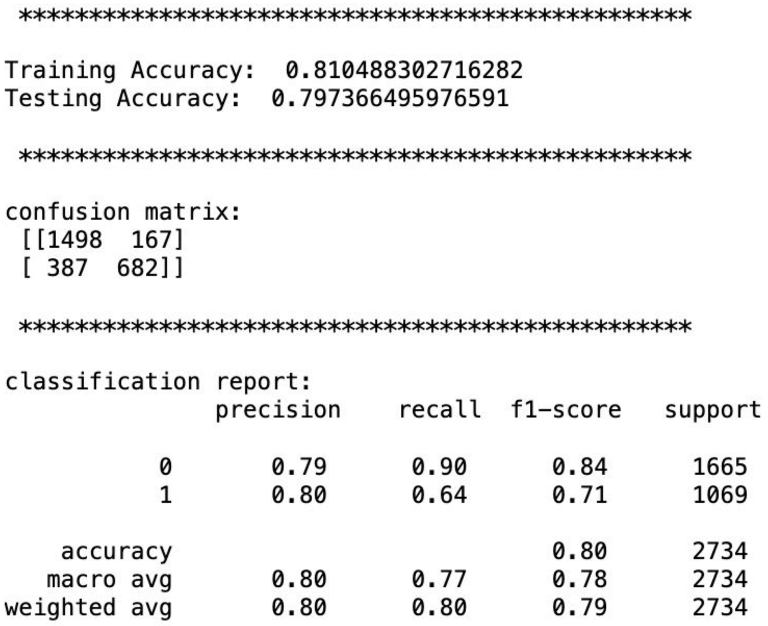
BUILDING MODEL: Split dataset (7:3)

- After creating test and train dataset we are seeing NaN values
- there are 99 records in the train set and 38 records in the test set which have NaN for two columns, totalVisits and Page Views Per Visit, hence it is better to remove these rows as it gives an error while training the model if it encounters NaN values
- Dropping this rows as NaN values are not accepted by Machine learning algorithm

BUILDING MODEL: Logistic Regression Model Results

- The model has an accuracy of 81.04% for training dataset and 79.73% for test dataset
- the model has False Positives (FP) of 387 out of 2734 leads
- The model had False Negatives (NP) of 167 out of 2737 leads

BUILDING MODEL: Logistic Regression Model Results



```
 ************************************************

Training Accuracy:  0.810488302716282
Testing Accuracy:  0.797366495976591

 ************************************************

confusion matrix:
 [[1498  167]
 [ 387  682]]

 ************************************************

classification report:
               precision    recall  f1-score   support

           0       0.79      0.90      0.84      1665
           1       0.80      0.64      0.71      1069

    accuracy                           0.80      2734
   macro avg       0.80      0.77      0.78      2734
weighted avg       0.80      0.80      0.79      2734
```

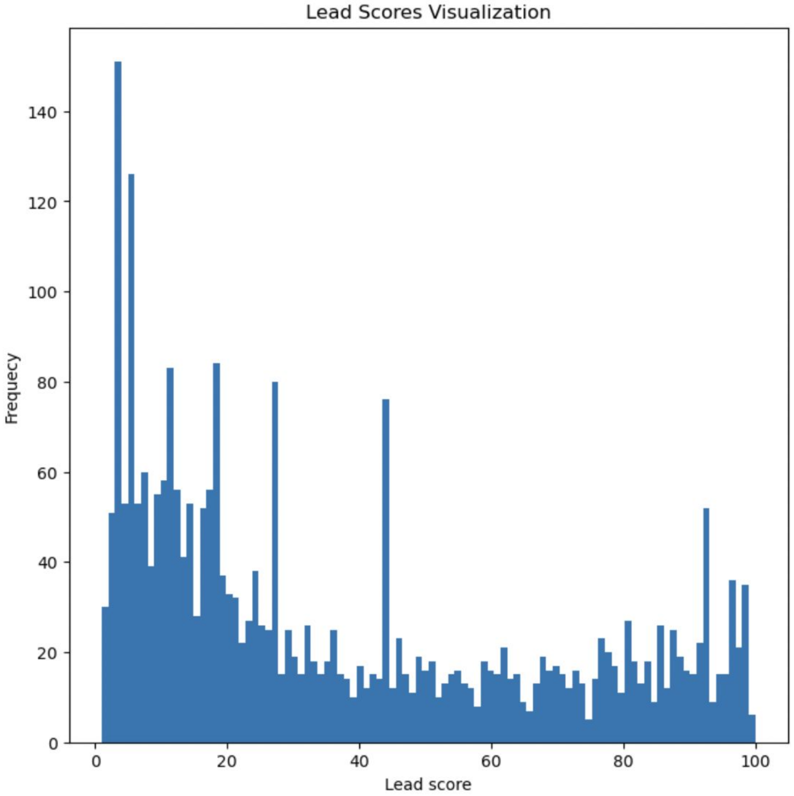
BUILDING MODEL: RFE Rankings on model 1

- All these features are high important for the lead to be converted
  - ('Lead Source_Social Media', True, 1),
  - ('Do Not Email_Yes', True, 1),
  - ('Last Activity_Page Visited on Website', True, 1),
  - ('Last Activity_SMS Sent', True, 1),
  - ('What is your current occupation_Student', True, 1),
  - ('What is your current occupation_Unemployed', True, 1),
  - ('What is your current occupation_Working Professional', True, 1),
  - ('Last Notable Activity_Resubscribed to emails', True, 1).

BUILDING MODEL: Lead Scores

- We can get the lead scores values from the logreg.predict_proba(X_test) method for the test dataset
- It is a 2d array, which we have to convert to 1D array to get an 1D array of lead scores
- Plotting the frequency distribution of lead scores for all the test data in the next slide

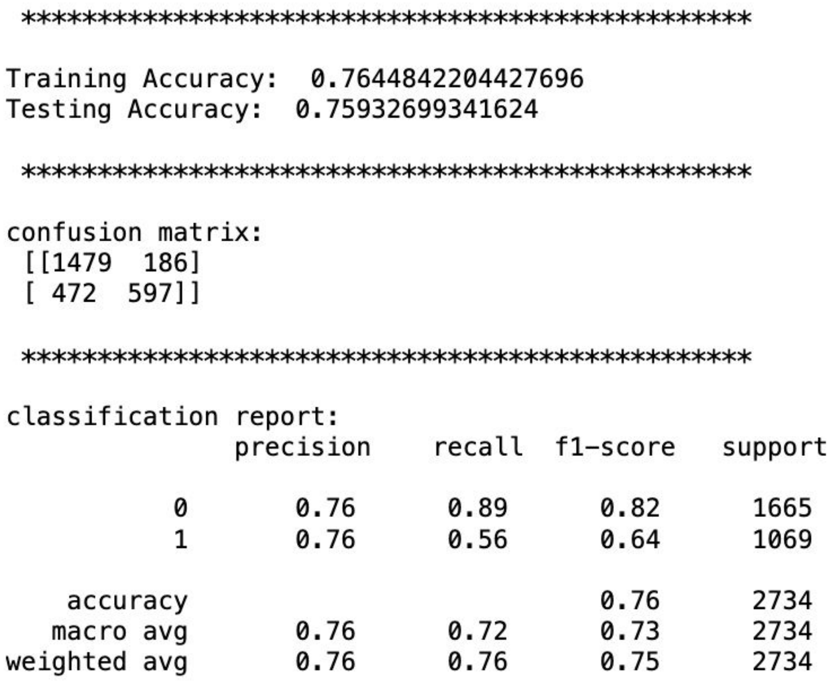
BUILDING MODEL: Lead Scores



BUILDING MODEL: Select top 15 features and rebuild

- The second model has an accuracy of 76.44% for training dataset and 75.94% for test dataset
- the second model has False Positives (FP) of 472 out of 2734 leads
- The second model had False Negatives (NP) of 186 out of 2737 leads
- The second model has lesser accuracy than the first model and has higher False Positives (FP) and False Negatives (FN)

BUILDING MODEL: Select top 15 features and rebuild



```
 ************************************************

Training Accuracy:  0.7644842204427696
Testing Accuracy:  0.75932699341624

 ************************************************

confusion matrix:
 [[1479  186]
 [ 472  597]]

 ************************************************

classification report:
               precision    recall  f1-score   support

           0       0.76      0.89      0.82      1665
           1       0.76      0.56      0.64      1069

    accuracy                           0.76      2734
   macro avg       0.76      0.72      0.73      2734
weighted avg       0.76      0.76      0.75      2734
```

CONCLUSION:

- the first model has better accuracy with accuracy of 81% for train data and 79.73% for test data when compared to second model with accuracy of 76.44% for train data and 75.93% for test data
- the first model has less False Positives (FP) 387 when compared to the second model which has False Positives (FP) of 472
- the first model and second model has almost very similar False Negatives (FN)
- Hence we can go ahead with the first model which has an accuracy of 81.04% for training dataset and 79.73% for test dataset

Thank you!