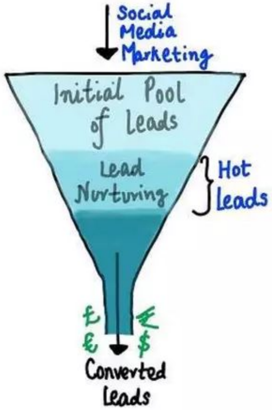
Lead Scoring Case Study

CASE STUDY DESCRIPTION

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.
When these people fill up a form providing their email address or phone number, they are classified to be a lead which will be then passed to Sales team to start making calls or send emails to convert these leads. The typical lead conversion rate at X education is around 30%.

Now, although X Education gets a lot of leads, its lead conversion rate is very poor. To make this process more efficient, the company wishes to identify the most potential leads, also known as ‘Hot Leads’. If they successfully identify this set of leads, the lead conversion rate should go up as the sales team will now be focusing more on communicating with the potential leads rather than making calls to everyone.



Problem STATEMENT

The company requires to build a model wherein a lead score is assigned 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.

This case study focuses on building 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.

Identification of such leads which can possible be converted is the focus of the case study

APPROACH

To improve the lead conversion rate to be around 80%, Logistic Regression model is created to identify the important variables and derive insights on how to improve the lead conversion count.

Below Steps are performed in the case study for the outcome :

- Data Loading & Cleaning
- Data Quality & Missing Values Check
- Handling Outliers
- Exploratory Data Analysis
- Data Preparation for Modelling
- Train-Test Data Split
- Scaling
- Feature Selection
- Recursive Model Building to find the optimal model
- Model Evaluation using Performance Metrics & Building ROC Curve
- Finding Optimal Cut-Off point
- Predictions on Test data using Final Model
- Final Evaluation using Performance Metrics on Test Data
- Calculating Lead Score

ASSUMPTIONS

- For this case study we have dropped the columns where missing value%>40% ('Lead Quality', 'Asymmetrique Activity Index', 'Asymmetrique Profile Score', 'Asymmetrique Activity Score', 'Asymmetrique Profile Index’) as applying any imputation on such huge missing values can impact the overall analysis of case study which is not recommended.

- Values coming as ‘SELECT’ in few columns have been replaced with null.

- For few Category columns, null values has been replaced with a new Category as “Others” to segregate the data.

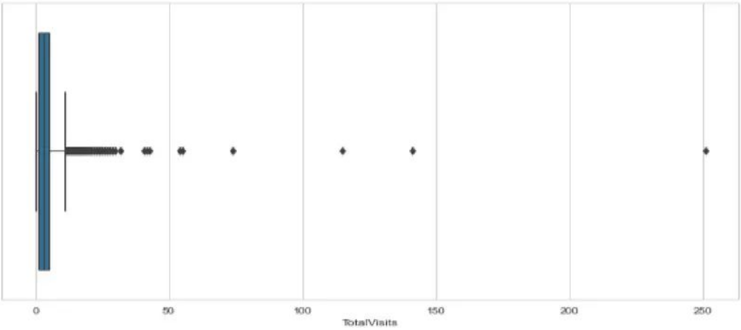
- For few Category columns, merged the category in Others category which has low volume of records.

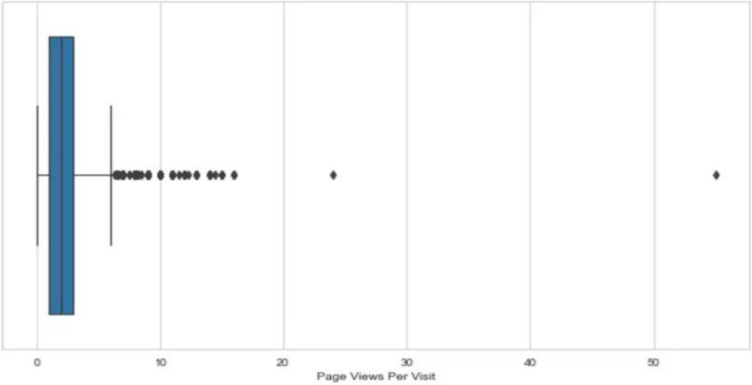
- For Numerical columns, null values has been imputed with IQR*1.5 of the variable for those where mean and median are same but max value is way out of range.

- Dropped few unnecessary columns where data was heavily skewed to not impact the overall model building.

OUTLIERS TREATMENT

- Observed Outliers with two Numerical columns which was derived using BoxPlot on them.
- For this case study we have treated any outliers for Continuous variables using Upper Bound values to be able to build proper model.
- For Category variable : we have used below two approaches:
  - Creating a new category for missing values
  - Proportionately divided the values in existing categories based on their distribution.



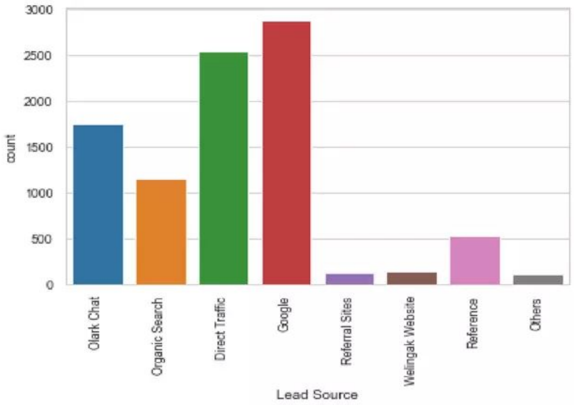


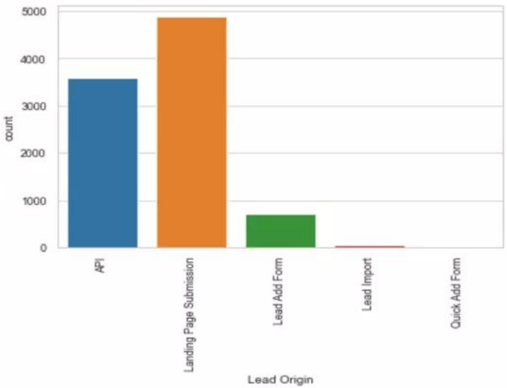
EXPLORATORY DATA ANALYSIS

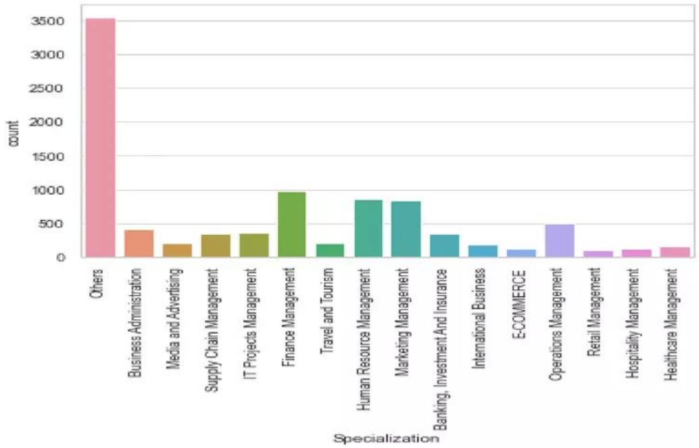
UNIVARIATE ANALYSIS

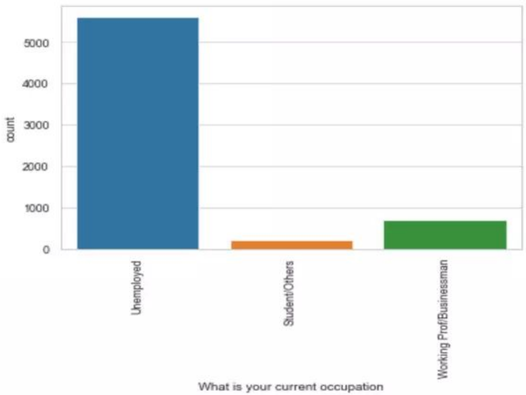
From above plots, we can infer that

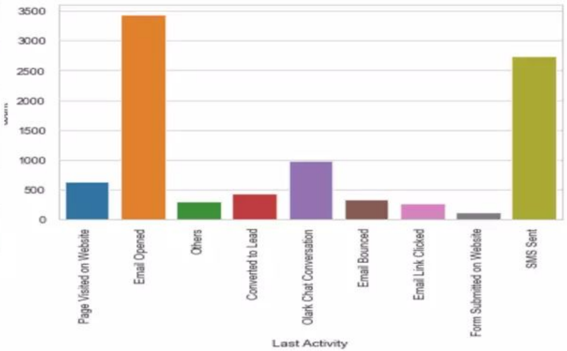
- Majority people are using either Google or Direct Traffic as lead source
- Unemployed are the majority of people who are visiting the site
- The last activity for majority of the leads is Email opened
- Majority leads have Landing Page submission as the Lead Origin







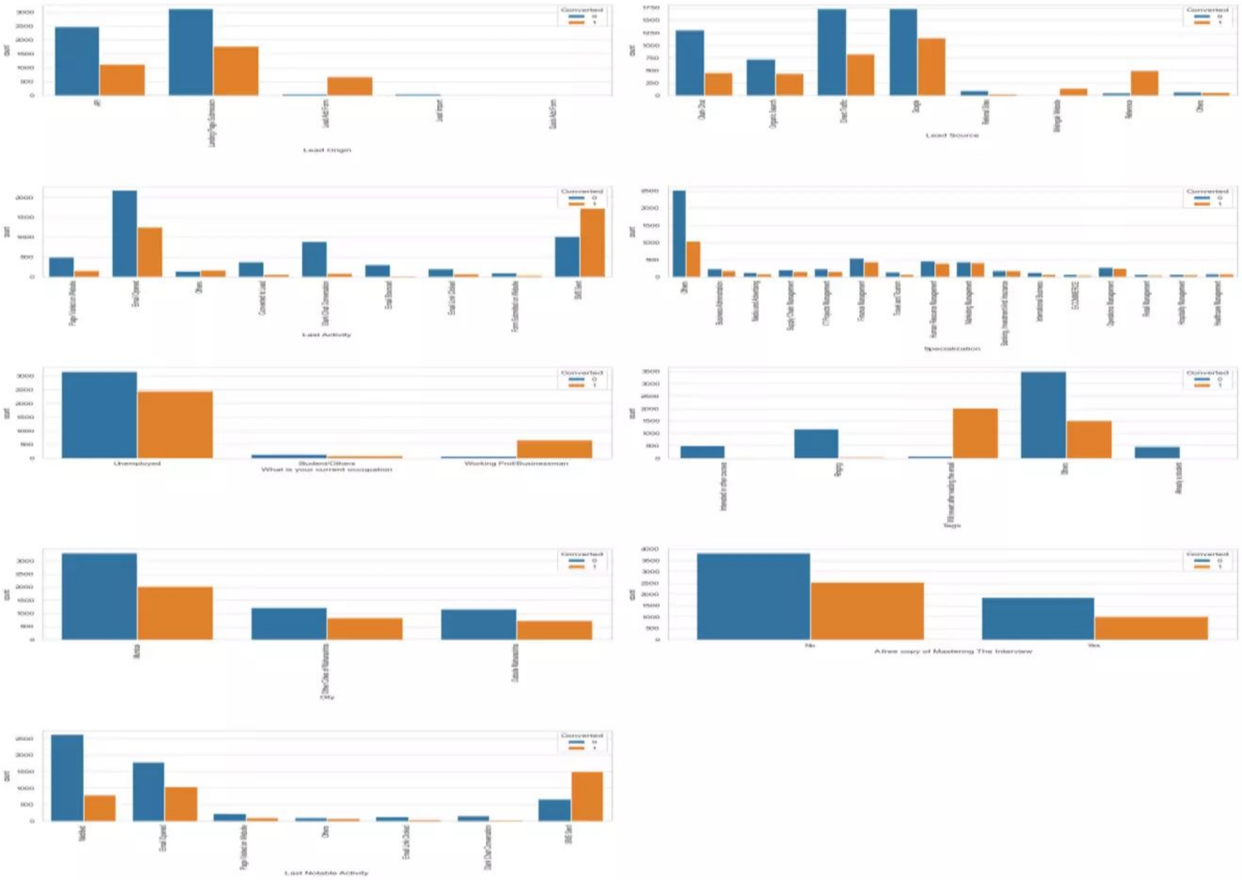




BIVARIATE ANALYSIS

From above plots it can be infer that

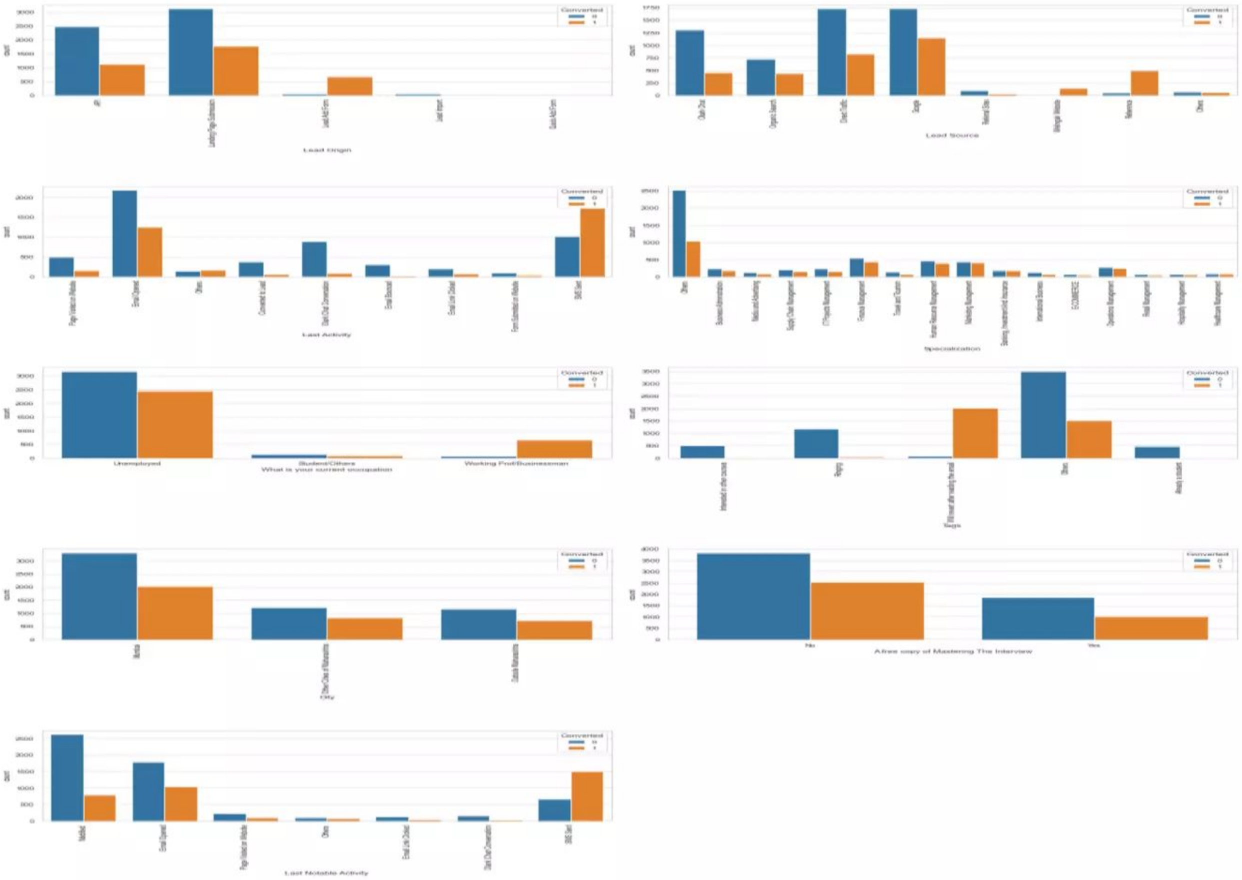
- From above plot, it can be infer that Working Professionals have the higher conversion
- Unemployed have the highest count in the lead category and additional focus can be given to them in conversion
- Google as the lead source has the highest conversion and the top two count of leads are from Direct Traffic or Google
- Lead Origin as Landing Page Submission has the highest count of leads along with most conversions
- From city plot, we can see that the conversion and lead rate is same for Other cities of Maharashtra, we can put more emphasis on advertisements in other states to get more leads
- People who said No for Free copy of Mastering the interview are highest in the conversion



BIVARIATE ANALYSIS

From above plots it can be infer that

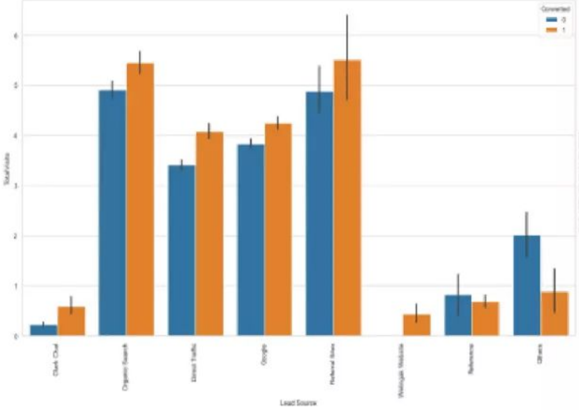
- From above plot, it can be infer that Working Professionals have the higher conversion
- Unemployed have the highest count in the lead category and additional focus can be given to them in conversion
- Google as the lead source has the highest conversion and the top two count of leads are from Direct Traffic or Google
- Lead Origin as Landing Page Submission has the highest count of leads along with most conversions
- From city plot, we can see that the conversion and lead rate is same for Other cities of Maharashtra, we can put more emphasis on advertisements in other states to get more leads
- People who said No for Free copy of Mastering the interview are highest in the conversion

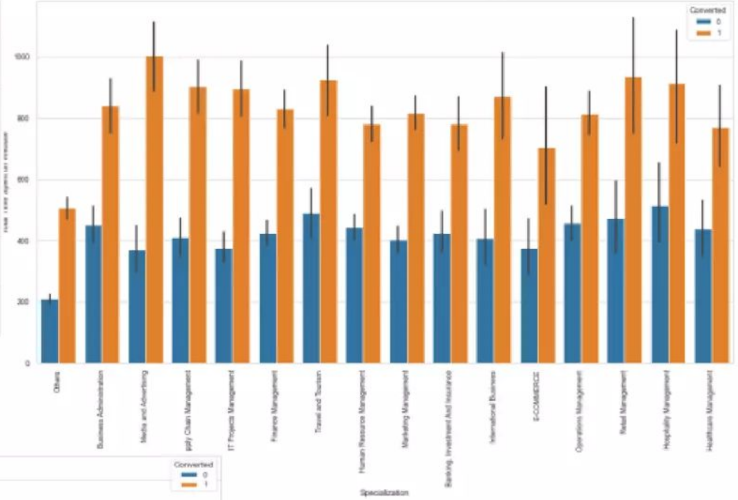


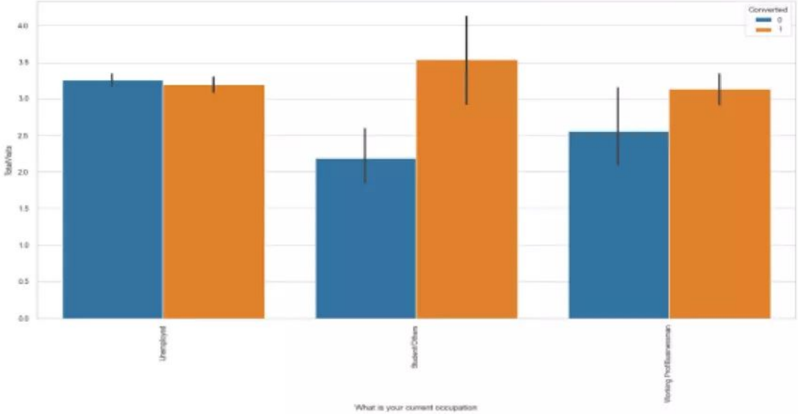
MULTIVARIATE ANALYSIS

From above plots it can be infer that

- Students/Others who visited the site regularly are more likely to be converted leads
- we can see that leads spending more time on website are majorly converted irrespective of Specialization.
- Lead Source as Wellingak Website, ClarkChat, Referral Sites & Organic Search are the ones who have most of them converted amongst the other lead source.



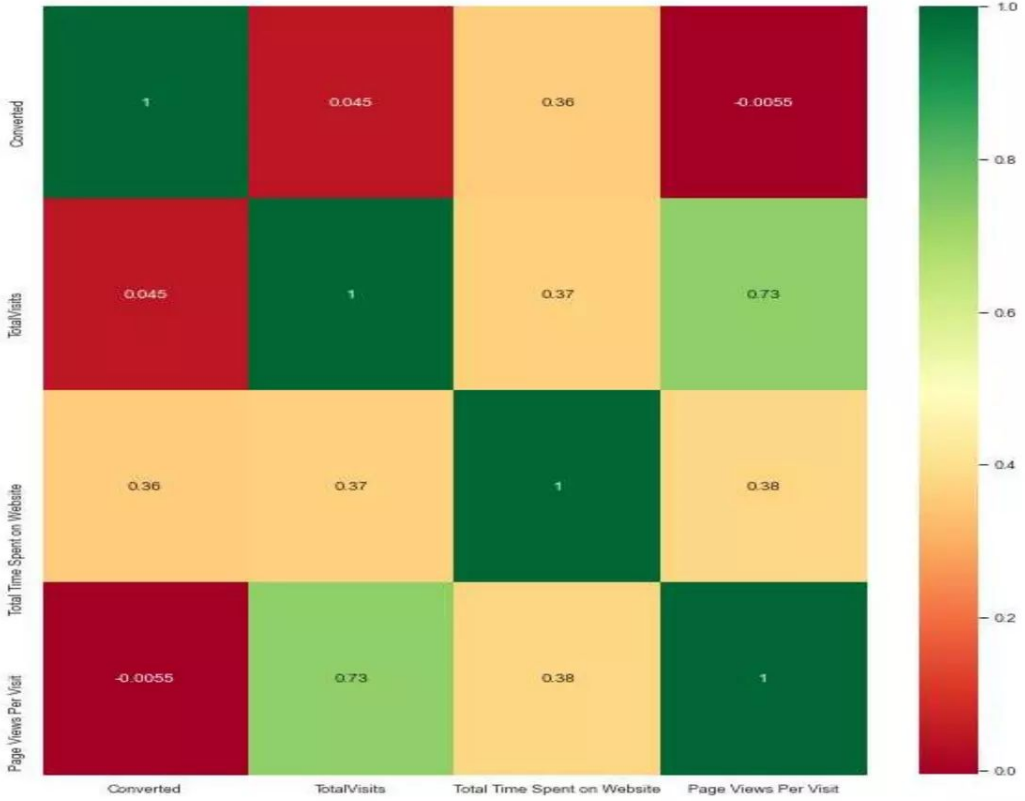




CORRELATION MATRIX

From above Correlation Matrix, we can see that

- Converted is having positive correlation with Total Time Spent on Website
- and negative relationship with Page Views Per Visit



MODEL BUILDING

DATA PREPARATION STEPS

- Converted binary variables (Yes/No) to 0 and 1 for model building.
- Created Dummy variables for all category columns using pd.get_dummies

TRAIN-TEST SPLIT

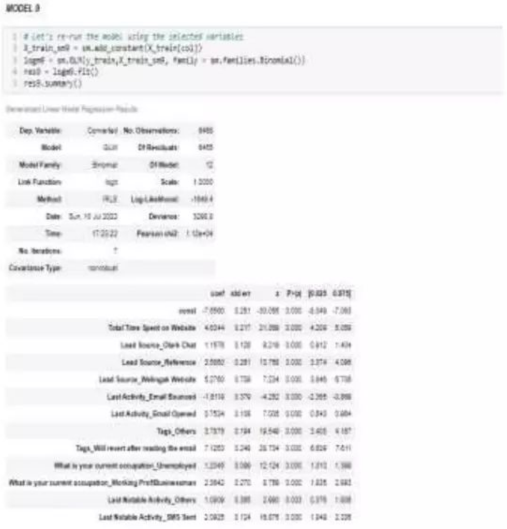
- Split the data into train and test data frame using 70-30% ratio. At this stage we have imported train-test-split library from sklearn

FEATURE SCALING

- We have used MinMax Scaler to convert the numerical columns so that they have comparable scales. If we don't have comparable scales, then some of the coefficients as obtained by fitting the model might be very large or very small as compared to the other coefficients which is not good at the time of model evaluation

MODEL BUILDING

- Build 1st Logistic Regression training model using all features.
- To build best fit model, we used Recursive Feature Elimination technique to get the top 20 features to build out next model
- For each model build, we have checked for p-value should be less than 0.05
- To remove Multicollinearity, calculated Variance Inflation Factor(VIF) to check if feature variables are not correlated with each other.
- Dropped the features which have high p-value and highly correlated one by one and recursively build the model to get optimal model.

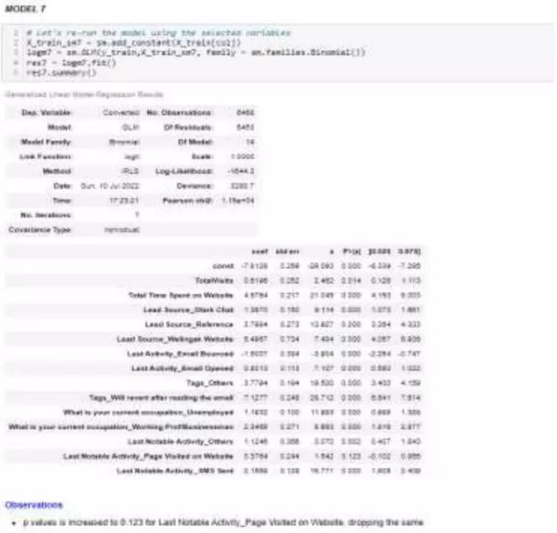


MODEL 9

```
# Let's re-run the model using the selected variables
X_train_sm9 = sm.add_constant(X_train[col])
logm9 = sm.GLM(y_train,X_train_sm9, family = sm.families.Binomial())
res9 = logm9.fit()
res9.summary()
```

```
Generalized Linear Model Regression Results
   Dep. Variable:       Converted  No. Observations:     6468
           Model:             GLM      Df Residuals:     6455
    Model Family:        Binomial          Df Model:       12
   Link Function:           logit             Scale:   1.0000
          Method:            IRLS    Log-Likelihood:  -1649.4
            Date: Sun, 10 Jul 2022         Deviance:   3298.8
            Time:        17:23:22     Pearson chi2: 1.12e+04
  No. Iterations:               7
 Covariance Type:       nonrobust

                                                      coef  std err        z  P>|z|  [0.025  0.975]
                                            const  -7.5560    0.251  -30.088  0.000  -8.049  -7.063
                      Total Time Spent on Website   4.6344    0.217   21.358  0.000   4.209   5.059
                          Lead Source_Olark Chat   1.1578    0.126    9.216  0.000   0.912   1.404
                           Lead Source_Reference   3.6862    0.231   15.758  0.000   3.374   4.098
                    Lead Source_Welingak Website   5.2760    0.730    7.234  0.000   3.846   6.706
                    Last Activity_Email Bounced   -1.6118    0.379   -4.252  0.000  -2.355  -0.868
                     Last Activity_Email Opened    0.7524    0.106    7.095  0.000   0.545   0.960
                                     Tags_Others   3.7879    0.194   19.546  0.000   3.408   4.167
         Tags_Will revert after reading the email   7.1253    0.248   28.734  0.000   6.639   7.611
       What is your current occupation_Unemployed   1.2049    0.099   12.124  0.000   1.010   1.399
What is your current occupation_Working Prof/Businessman   2.3642    0.270   8.756  0.000   1.835   2.893
                     Last Notable Activity_Others   1.0909    0.365    2.990  0.003   0.376   1.806
                   Last Notable Activity_SMS Sent   2.0925    0.124   16.875  0.000   1.849   2.336
```



MODEL 7

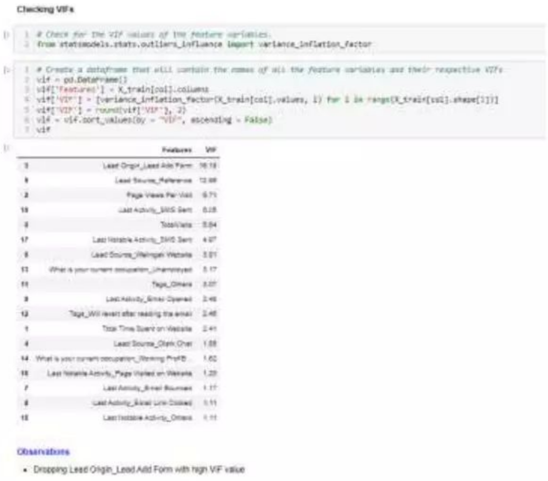
```
# Let's re-run the model using the selected variables
X_train_sm7 = sm.add_constant(X_train[col])
logm7 = sm.GLM(y_train,X_train_sm7, family = sm.families.Binomial())
res7 = logm7.fit()
res7.summary()
```

```
Generalized Linear Model Regression Results
   Dep. Variable:       Converted  No. Observations:     6468
           Model:             GLM      Df Residuals:     6453
    Model Family:        Binomial          Df Model:       14
   Link Function:           logit             Scale:   1.0000
          Method:            IRLS    Log-Likelihood:  -1644.3
            Date: Sun, 10 Jul 2022         Deviance:   3288.7
            Time:        17:23:21     Pearson chi2: 1.15e+04
  No. Iterations:               7
 Covariance Type:       nonrobust

                                                      coef  std err        z  P>|z|  [0.025  0.975]
                                            const  -7.8128    0.269  -29.093  0.000  -8.339  -7.286
                                      TotalVisits   0.6196    0.252    2.462  0.014   0.126   1.113
                      Total Time Spent on Website   4.5784    0.217   21.049  0.000   4.152   5.005
                          Lead Source_Olark Chat   1.3870    0.160    8.674  0.000   1.073   1.701
                           Lead Source_Reference   3.7994    0.273   13.927  0.000   3.264   4.333
                    Lead Source_Welingak Website   5.4967    0.734    7.484  0.000   4.057   6.936
                    Last Activity_Email Bounced   -1.5157    0.394   -3.854  0.000  -2.284  -0.747
                     Last Activity_Email Opened    0.8013    0.113    7.107  0.000   0.580   1.022
                                     Tags_Others   3.7794    0.194   19.530  0.000   3.400   4.159
         Tags_Will revert after reading the email   7.1277    0.248   28.712  0.000   6.641   7.614
       What is your current occupation_Unemployed   1.1632    0.100   11.633  0.000   0.967   1.359
What is your current occupation_Working Prof/Businessman   2.3469    0.271   8.653  0.000   1.815   2.877
                     Last Notable Activity_Others   1.1246    0.366    3.072  0.002   0.407   1.842
   Last Notable Activity_Page Visited on Website   0.3764    0.244    1.542  0.123  -0.102   0.855
                   Last Notable Activity_SMS Sent   2.1589    0.128   16.777  0.000   1.908   2.409
```

Observations

- p values is increased to 0.123 for Last Notable Activity_Page Visited on Website. dropping the same



Checking VIFs

```
# Check for the VIF values of the feature variables.
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

```
# Create a dataframe that will contain the names of all the feature variables and their respective VIFs
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

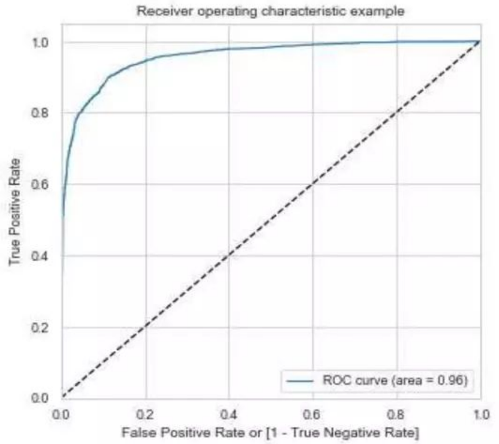
```
                                             Features    VIF
 3                          Lead Origin_Lead Add Form  16.18
 5                              Lead Source_Reference  12.89
 2                               Page Views Per Visit   6.71
10                             Last Activity_SMS Sent   6.18
 0                                        TotalVisits   5.64
17                     Last Notable Activity_SMS Sent   4.87
 6                       Lead Source_Welingak Website   3.51
13         What is your current occupation_Unemployed   3.17
11                                        Tags_Others   3.07
 9                         Last Activity_Email Opened   2.48
12           Tags_Will revert after reading the email   2.46
 1                        Total Time Spent on Website   2.41
 4                             Lead Source_Olark Chat   1.88
14  What is your current occupation_Working Prof/B...   1.62
16      Last Notable Activity_Page Visited on Website   1.23
 7                        Last Activity_Email Bounced   1.17
 8                   Last Activity_Email Link Clicked   1.11
15                       Last Notable Activity_Others   1.11
```

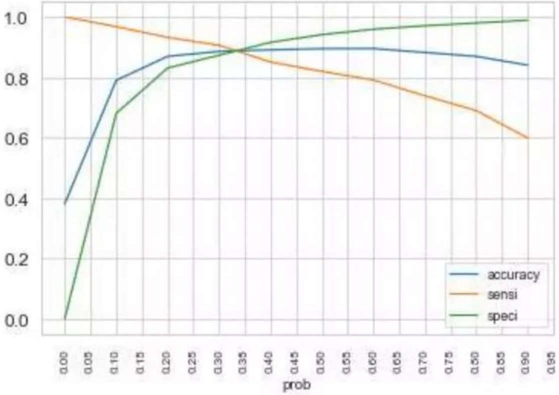
Observations

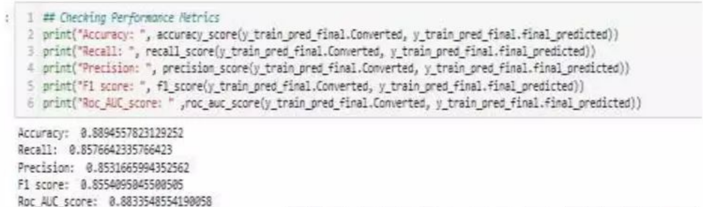
- Dropping Lead Origin_Lead Add Form with high VIF value

MODEL EVALUATION – TRAIN DATA

- After getting optimal model, evaluated performance metrics score Accuracy, Recall, Precision, F1 score.
- ROC curve plotted that shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
    - The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
    - The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.
- Calculated Optimal cutoff point between sensitivity & specificity. From below plot, we have received 0.33 as the optimal cut-off point.
- Also checked Precision and Recall trade-off as this will help us to identify the predicted CONVERTED is actual CONVERTED
- The Precision and Recall tradeoff came out to be 0.38, we have considered that as our cut-off probability on test data.



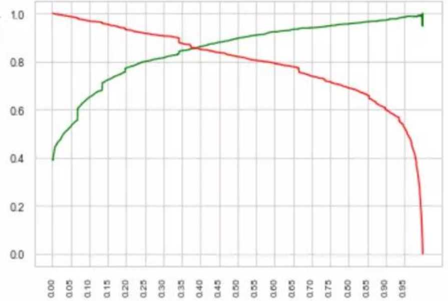




```
## Checking Performance Metrics
print("Accuracy: ", accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted))
print("Recall: ", recall_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted))
print("Precision: ", precision_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted))
print("F1 score: ", f1_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted))
print("Roc_AUC_score: " ,roc_auc_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted))
```

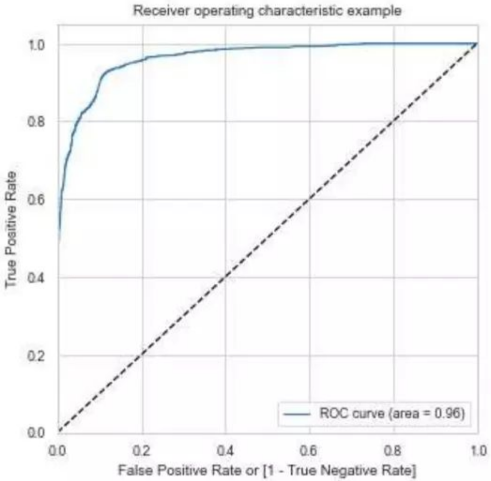
```
Accuracy:  0.8894557823129252
Recall:  0.8576642335766423
Precision:  0.8531665994352562
F1 score:  0.8554095045500505
Roc_AUC_score:  0.8833548554190058
```

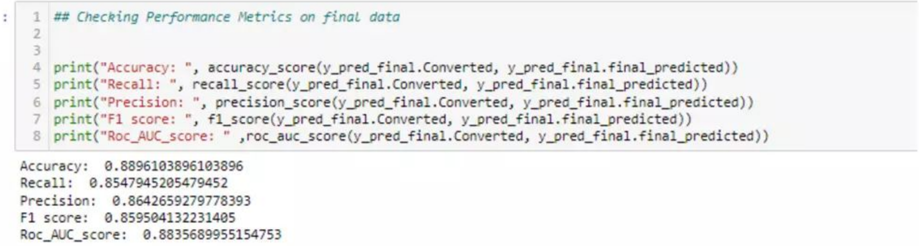
- Recall value is now 0.86 and Accuracy is 0.89



MODEL EVALUATION – TEST DATA

- Run the final optimal model on test dataset with below observations :
- ROC Curve came out similar to what we got on our train data.
- Recall/Sensitivity Score is 85.4%
- Accuracy – 88.9%
- Precision – 86.4%





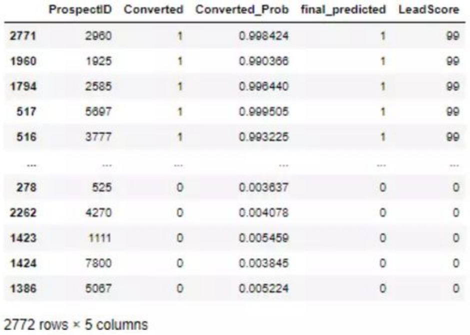
```
## Checking Performance Metrics on final data


print("Accuracy: ", accuracy_score(y_pred_final.Converted, y_pred_final.final_predicted))
print("Recall: ", recall_score(y_pred_final.Converted, y_pred_final.final_predicted))
print("Precision: ", precision_score(y_pred_final.Converted, y_pred_final.final_predicted))
print("F1 score: ", f1_score(y_pred_final.Converted, y_pred_final.final_predicted))
print("Roc_AUC_score: " ,roc_auc_score(y_pred_final.Converted, y_pred_final.final_predicted))
```

```
Accuracy:  0.8896103896103896
Recall:  0.8547945205479452
Precision:  0.8642659279778393
F1 score:  0.859504132231405
Roc_AUC_score:  0.8835689955154753
```

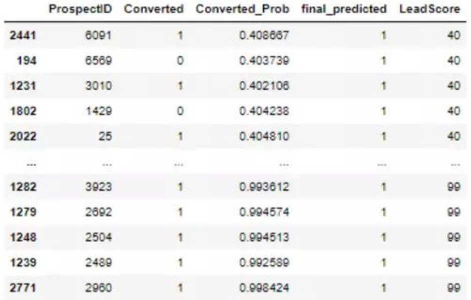
LEAD SCORE PREDICTION

- The final_predicted column shows the conversion probability of prospective lead
- Lead Score above 39 have a high tendency of converting to a Hot Lead category



	ProspectID	Converted	Converted_Prob	final_predicted	LeadScore
2771	2960	1	0.998424	1	99
1960	1925	1	0.990366	1	99
1794	2585	1	0.996440	1	99
517	5697	1	0.999505	1	99
516	3777	1	0.993225	1	99
...	...	...	...	...	...
278	525	0	0.003637	0	0
2262	4270	0	0.004078	0	0
1423	1111	0	0.005459	0	0
1424	7800	0	0.003845	0	0
1386	5067	0	0.005224	0	0

2772 rows × 5 columns



	ProspectID	Converted	Converted_Prob	final_predicted	LeadScore
2441	6091	1	0.408667	1	40
194	6569	0	0.403739	1	40
1231	3010	1	0.402106	1	40
1802	1429	0	0.404238	1	40
2022	25	1	0.404810	1	40
...	...	...	...	...	...
1282	3923	1	0.993612	1	99
1279	2692	1	0.994574	1	99
1248	2504	1	0.994513	1	99
1239	2489	1	0.992589	1	99
2771	2960	1	0.998424	1	99

RECOMMENDATIONS

- Prospect spending more time on website have high changes of becoming Hot Leads therefore Sales team can provide more focus on reaching out to those.
- Lead Score with Welingak websites and referral are the ones who have the highest amount of conversions therefore additional marketing can be done on the websites and sales team can sent the course details and promotional offers to existing users to get more Hot Leads
- Leads contacted via email/sms has higher chances of conversion.
- Unemployed/Working Professionals as Occupation category can generate more leads by reaching out to them and providing information about the courses available.

CONCLUSION

- The final model shows 88.9% accuracy with Recall as 85.7% and Precision as 85.3%
- The optimal cut-off was selected based on Precision and Recall trade off score.
- The model also worked fine on test dataset with Recall as 85.4% and Precision as 86.4%
- Overall the model looks good and is able to identify the correct leads which has high chances of conversion using Lead Score prediction