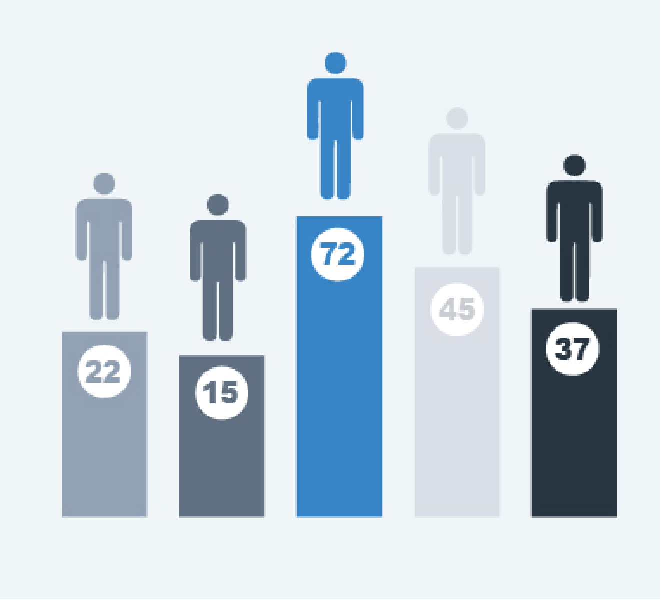
Lead Scoring Case Study



Prepared By : Abhijit Ganguly

Date : 28th Feb 2023

Problem Statement

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. The company markets its courses on several websites and search engines like Google. Once these people land on the website, they might browse the courses or fill up a form for the course or watch some videos. When these people fill up a form providing their email address or phone number, they are classified to be a lead. Moreover, the company also gets leads through past referrals. 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%. There are a lot of leads generated in the initial stage (top) but only a few of them come out as paying customers from the bottom. In the middle stage, you need to nurture the potential leads well (i.e., educating the leads about the product, constantly communicating etc.) in order to get a higher lead conversion.

X Education has appointed you to help them select the most promising leads, i.e., the leads that are most likely to convert into paying customers. The company requires you to 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 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%.

Business Objective

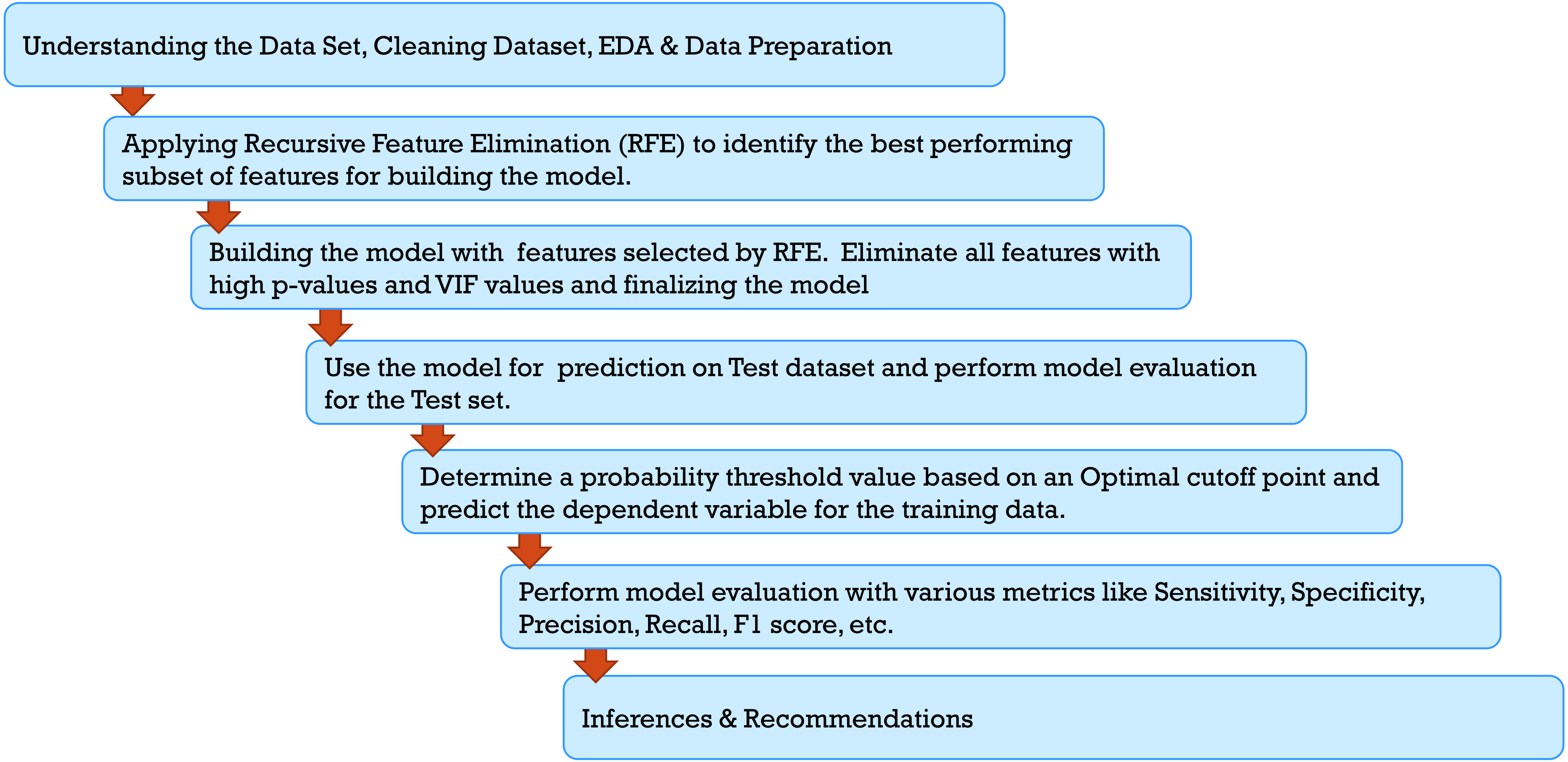
- To help X Education to select the most promising leads (Hot Leads), i.e., the leads that are most likely to convert into paying customers.
- To build a Logistic Regression (LR) model to assign a lead score value between 0 and 100 to each of the leads which can be used by the company to target potential leads.

The objective is thus classified into the following sub-goals:

- Create a Logistic Regression model to predict the Lead Conversion probabilities for each lead.
- Decide on a probability threshold value above which a lead will be predicted as converted, whereas not converted if it is below it.
- Multiply the Lead Conversion probability to arrive at the Lead Score value for each lead.

Problem Solving Approach

The entire case study solving approach has been broken down into multiple steps that are listed out below -



Understanding the Data Set, Cleaning Dataset, EDA & Data Preparation

Applying Recursive Feature Elimination (RFE) to identify the best performing subset of features for building the model.

Building the model with features selected by RFE. Eliminate all features with high p-values and VIF values and finalizing the model

Use the model for prediction on Test dataset and perform model evaluation for the Test set.

Determine a probability threshold value based on an Optimal cutoff point and predict the dependent variable for the training data.

Perform model evaluation with various metrics like Sensitivity, Specificity, Precision, Recall, F1 score, etc.

Inferences & Recommendations

Data Preparation & Feature Engineering (1/3)

Following data preparation steps were taken to make the data dependable prior to creating the model and using it for making key business decisions :

Removing Columns Having Only One Unique Value

- Deleting the following columns as they have only one unique value and hence cannot be useful in predicting a successful lead case – 'Magazine', 'Receive More Updates About Our Courses', 'Update me on Supply Chain Content' , 'Update me on Supply Chain Content' and 'I agree to pay the amount through cheque'.

Removing Rows For A Particular Column Having High Missing Values

- 'Lead Source' is an important column for analysis. Hence all the rows that have null values for it were dropped.

Imputing NULL Values With Median

- The columns 'Total Visits' and 'Page Views Per Visit' are continuous variables with outliers. Hence the null values for these columns were imputed with the column median values.

Imputing NULL Values With Mode

- The columns 'Country' is a categorical variable with some Null values. Also, majority of the records belong to the Country 'India'. Thus imputed the Null values for this with Mode (most occurring value). Then binned rest of category into 'Outside India'.

Data Preparation & Feature Engineering (2/3)

Handling 'Select' Values In Some Columns

- There are some columns in dataset having a level/value called 'Select'. This might have happened because these fields on the website might be non-mandatory with drop downs options for the customer to choose from. Amongst dropdown values default option is probably 'Select' and since these aren't mandatory fields, many customers might have left it at the default value 'Select'.
- The Select values in columns were converted to Null.

Assigning A Unique Category To NULL/SELECT Values

- All Nulls in the columns were binned into a separate column 'Unknown'.
- Instead of deleting columns with huge Null value percentage (resulting in loss of data), this approach adds more information into the dataset and results in the change of variance.
- The 'Unknown' levels for each of these columns get finally dropped during dummy encoding.

Outlier Treatment

- The outliers present in the columns 'Total Visits' & 'Page Views Per Visit' were finally removed based on Interquartile Range (IQR) analysis.

Binary Encoding

- Converting the following binary variables (Yes/No) to 0/1: 'Search', 'Do Not Email', 'Do Not Call', 'Newspaper Article', 'X Education Forums', 'Newspaper', 'Digital Advertisement', 'Through Recommendations' & 'A free copy of Mastering The Interview'

Data Preparation & Feature Engineering (3/3)

Categorical Variables with Multiple Levels –One-Hot Encoding

- For the following categorical variables with multiple levels, dummy features (one-hot encoded) were created:
- 'Lead Quality','Asymmetrique Profile Index','Asymmetrique Activity Index', 'Tags', 'Lead Profile', 'Lead Origin', 'What is your current occupation', 'Specialization', 'City', 'Last Activity', 'Country' and 'Lead Source', 'Last Notable Activity'

Test-Train Split

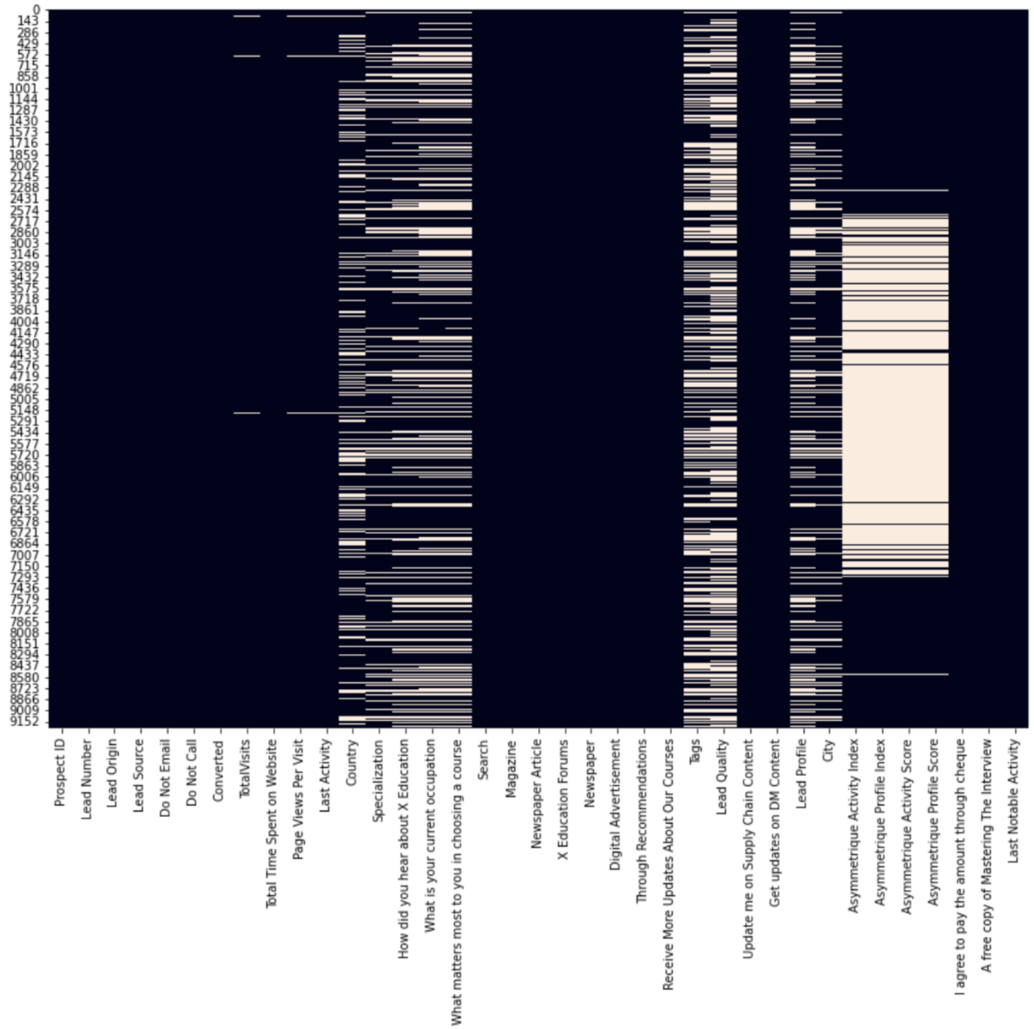
- The original dataframe was split into train and test dataset. The train dataset was used to train the model and test dataset was used to evaluate the model.

Feature Scaling

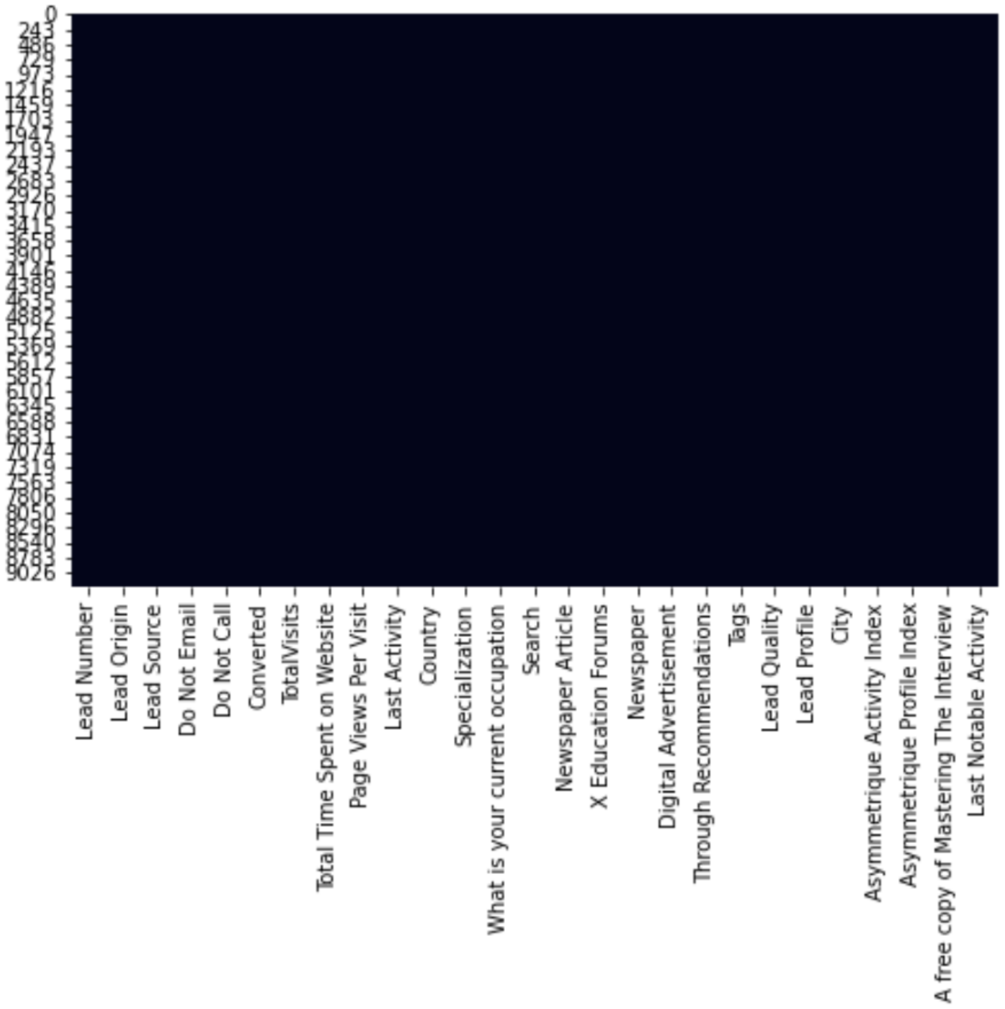
- Scaling helps in better interpretation during the modeling process. It is important to bring all variables (especially categorical ones which have values 0 and 1) on the same scale for the model to be easily interpretable.
- 'Standardisation' was used to scale the data for modelling – this basically brings all of the data into a standard normal distribution with Mean at 0 and Standard deviation to 1

Data Preparation : Null Value Handling

Distribution of NULL values in the different columns of the original dataset BEFORE Null values handling.



Distribution of NULL values in the various columns AFTER dropping columns and Null values handling.

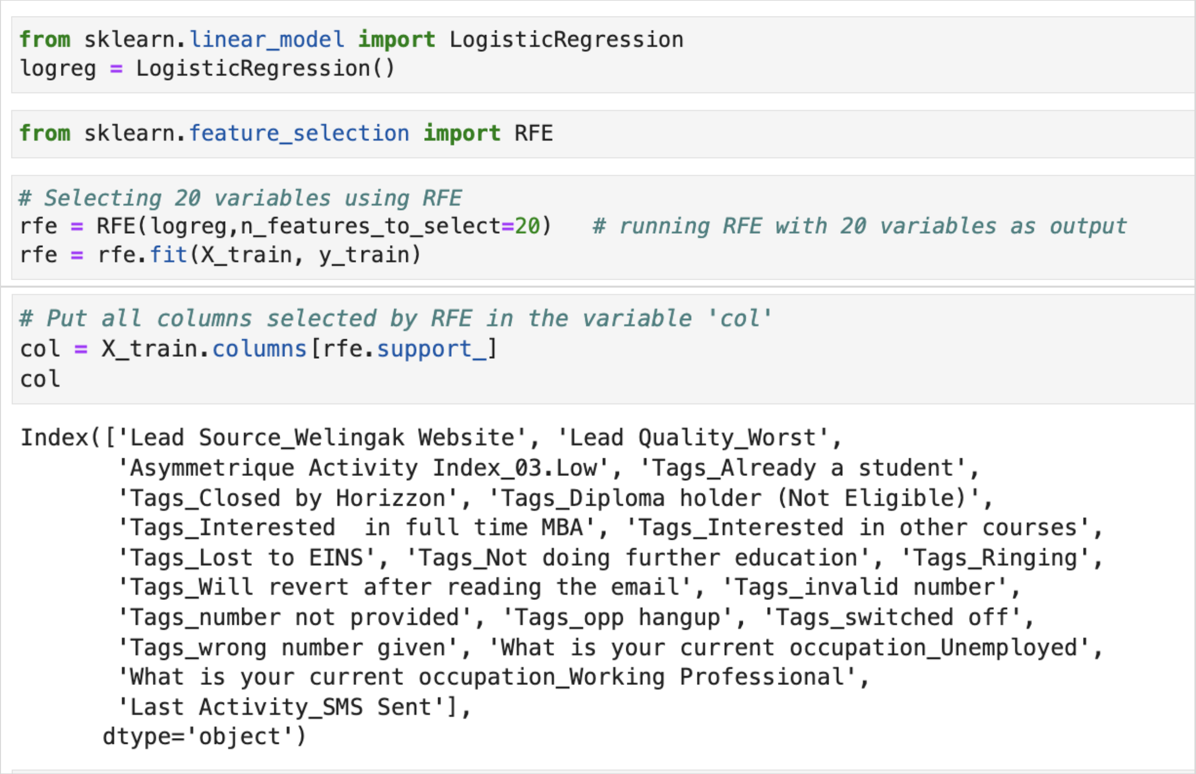


Initial Feature Selection (with RFE)

Recursive Feature Elimination (RFE)

- Recursive Feature Elimination, or RFE for short, is a feature selection algorithm. RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains.

- RFE was used for the initial set of features selection from a long list of variables – with output number of variables set equal to 20



```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
```

```
from sklearn.feature_selection import RFE
```

```
# Selecting 20 variables using RFE
rfe = RFE(logreg,n_features_to_select=20)   # running RFE with 20 variables as output
rfe = rfe.fit(X_train, y_train)
```

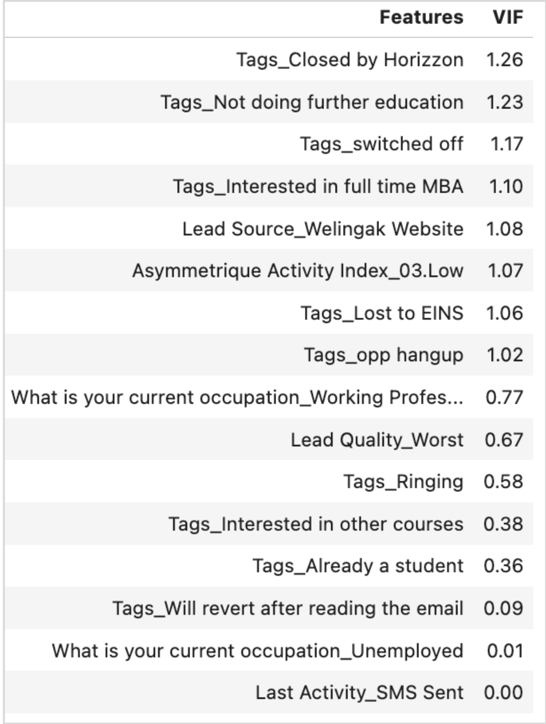
```
# Put all columns selected by RFE in the variable 'col'
col = X_train.columns[rfe.support_]
col
```

```
Index(['Lead Source_Welingak Website', 'Lead Quality_Worst',
       'Asymmetrique Activity Index_03.Low', 'Tags_Already a student',
       'Tags_Closed by Horizzon', 'Tags_Diploma holder (Not Eligible)',
       'Tags_Interested  in full time MBA', 'Tags_Interested in other courses',
       'Tags_Lost to EINS', 'Tags_Not doing further education', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_invalid number',
       'Tags_number not provided', 'Tags_opp hangup', 'Tags_switched off',
       'Tags_wrong number given', 'What is your current occupation_Unemployed',
       'What is your current occupation_Working Professional',
       'Last Activity_SMS Sent'],
      dtype='object')
```

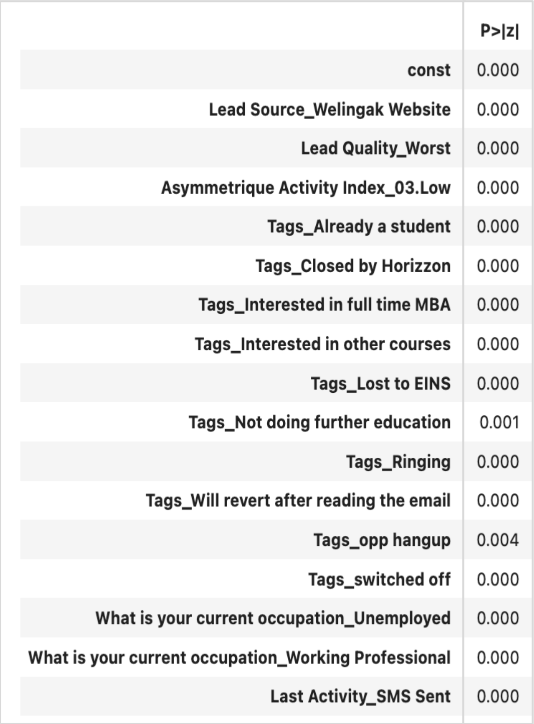
Converging To The Final Model

Model Building & Iterative Finalizing Process

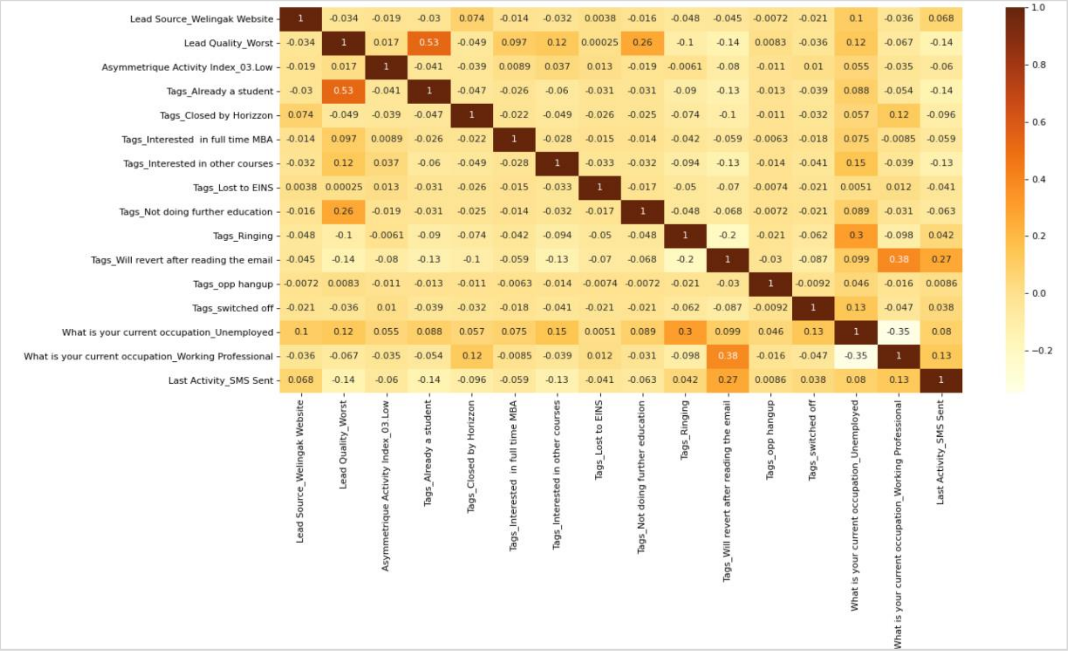
- Generalized Linear Models from StatsModels is used to build the Logistic Regression model.
  - To start with first version of the model was built with the 20 variables selected from running RFE.
  - Unwanted features were dropped (one at a time) serially after checking p-values (to keep them at < 0.5) and VIFs (< 5) and in the process building the model iteratively.
  - Final version of the model has16 features and passes both the significance test and Multi-collinearity test.



Features	VIF
Tags_Closed by Horizzon	1.26
Tags_Not doing further education	1.23
Tags_switched off	1.17
Tags_Interested in full time MBA	1.10
Lead Source_Welingak Website	1.08
Asymmetrique Activity Index_03.Low	1.07
Tags_Lost to EINS	1.06
Tags_opp hangup	1.02
What is your current occupation_Working Profes...	0.77
Lead Quality_Worst	0.67
Tags_Ringing	0.58
Tags_Interested in other courses	0.38
Tags_Already a student	0.36
Tags_Will revert after reading the email	0.09
What is your current occupation_Unemployed	0.01
Last Activity_SMS Sent	0.00

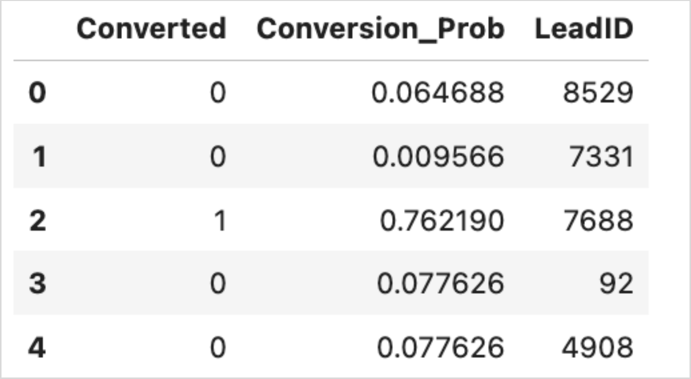


	P>|z|
const	0.000
Lead Source_Welingak Website	0.000
Lead Quality_Worst	0.000
Asymmetrique Activity Index_03.Low	0.000
Tags_Already a student	0.000
Tags_Closed by Horizzon	0.000
Tags_Interested in full time MBA	0.000
Tags_Interested in other courses	0.000
Tags_Lost to EINS	0.000
Tags_Not doing further education	0.001
Tags_Ringing	0.000
Tags_Will revert after reading the email	0.000
Tags_opp hangup	0.004
Tags_switched off	0.000
What is your current occupation_Unemployed	0.000
What is your current occupation_Working Professional	0.000
Last Activity_SMS Sent	0.000



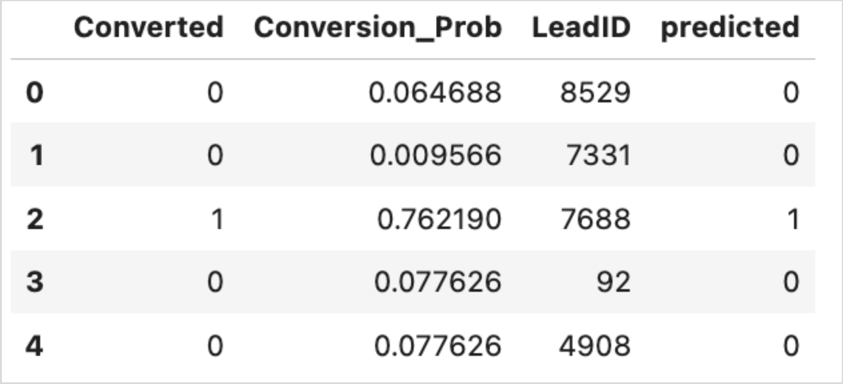
- A heat map consisting of the final 16 features included in the final model proves that there is no significant correlation between the independent variables.

Predicting Conversion Prob & Adding ‘Predicted’ Column



	Converted	Conversion_Prob	LeadID
0	0	0.064688	8529
1	0	0.009566	7331
2	1	0.762190	7688
3	0	0.077626	92
4	0	0.077626	4908

- Creating a dataframe with the actual Converted flag and predicted probabilities.
- Top 5 records of the dataframe are displayed in image.



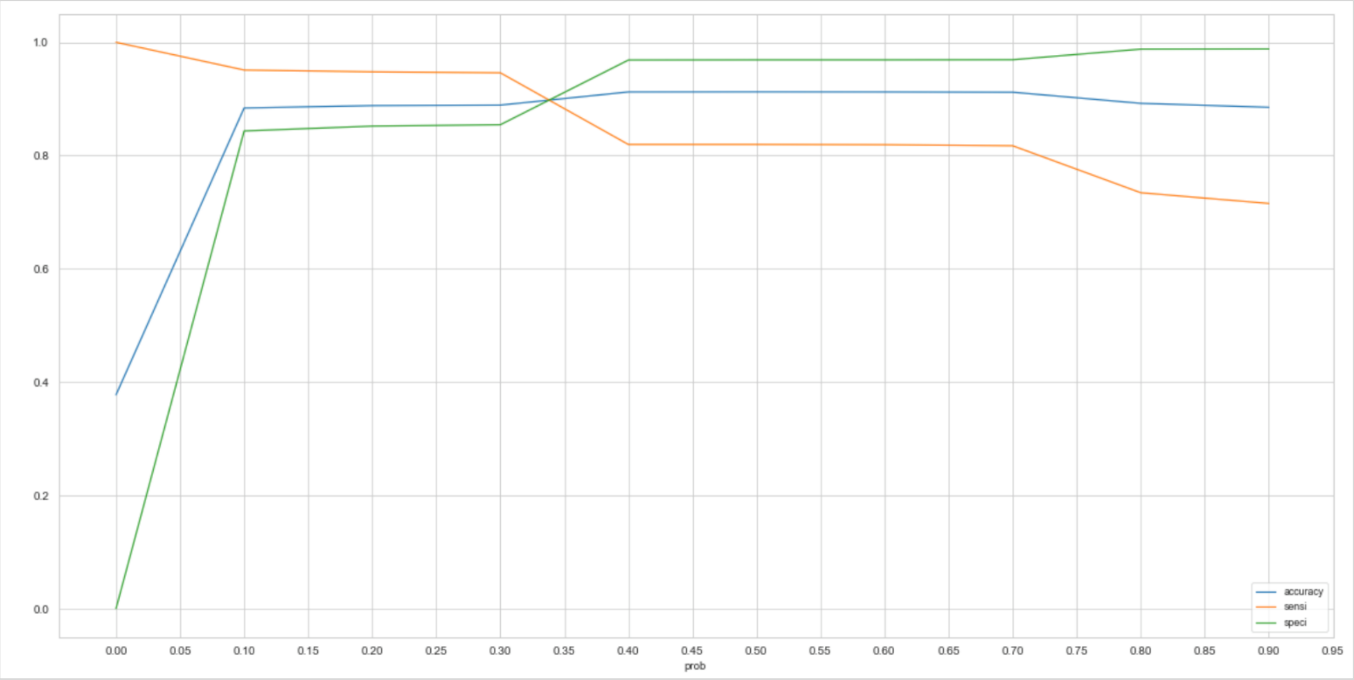
	Converted	Conversion_Prob	LeadID	predicted
0	0	0.064688	8529	0
1	0	0.009566	7331	0
2	1	0.762190	7688	1
3	0	0.077626	92	0
4	0	0.077626	4908	0

- Added new column 'predicted' with 1 if Conversion Probability > 0.5 else 0

- Showing top 5 records of the dataframe in the image.

Finding Optimal Probability Threshold

Optimal cutoff on LR probabilities is that intersection point at which there is a balance between metrics Sensitivity and Specificity.



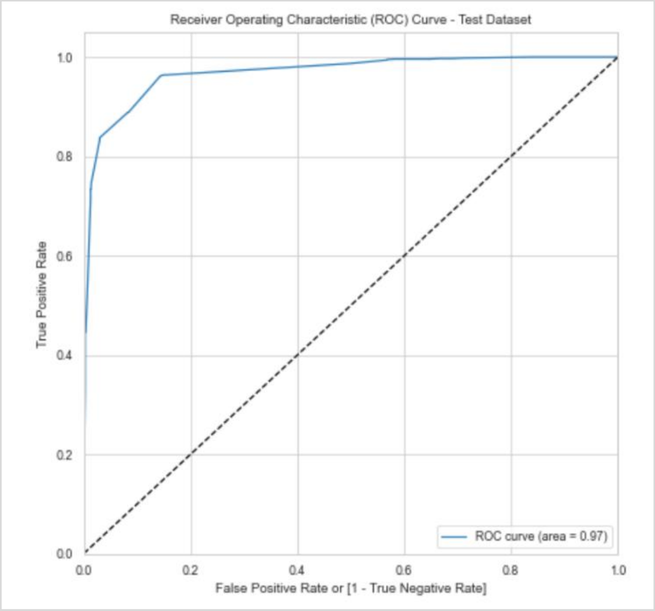
Optimal Probability Threshold

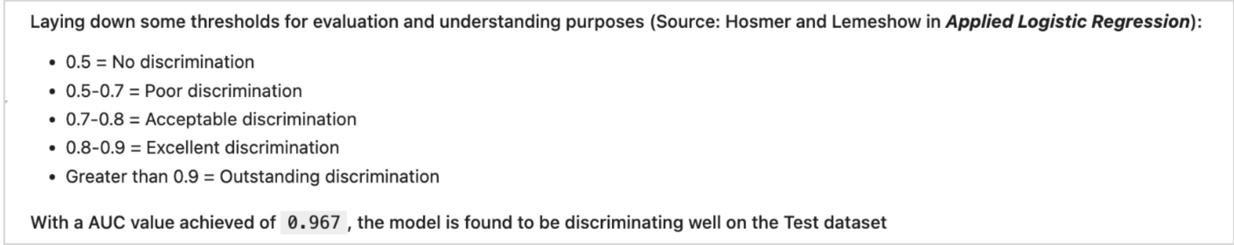
- Model evaluation metrics – Accuracy, Sensitivity and Specificity was calculated for various values of probability threshold and plotted in the graph.

- As can be observed from the curve in the plot, 0.34 is found to be the optimum point for cutoff probability.

- At this threshold value, all 3 metrics - Accuracy, Sensitivity and Specificity were found to be well above 80% which are in an acceptable range.

Plotting ROC Curve & Calculating AUC





Laying down some thresholds for evaluation and understanding purposes (Source: Hosmer and Lemeshow in ***Applied Logistic Regression***):

- 0.5 = No discrimination
- 0.5-0.7 = Poor discrimination
- 0.7-0.8 = Acceptable discrimination
- 0.8-0.9 = Excellent discrimination
- Greater than 0.9 = Outstanding discrimination

With a AUC value achieved of `0.967`, the model is found to be discriminating well on the Test dataset

Receiver Operating Characteristic (ROC) Curve

- It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).

Area Under Curve (GINI)

- By determining the Area under the curve (AUC) of the ROC curve, the goodness of the model is determined. Since the ROC curve is more towards the upper-left corner of the graph, it means that the model is very good. The larger the AUC, the better will is the model.
- The value of AUC for our model is 0.967.

Model Evaluation (on Train Dataset) | Metrics

Confusion Matrix

Predicted ⇨	Not Converted	Converted
Actual ⇩		
Not Converted	3411	325
Converted	256	2010

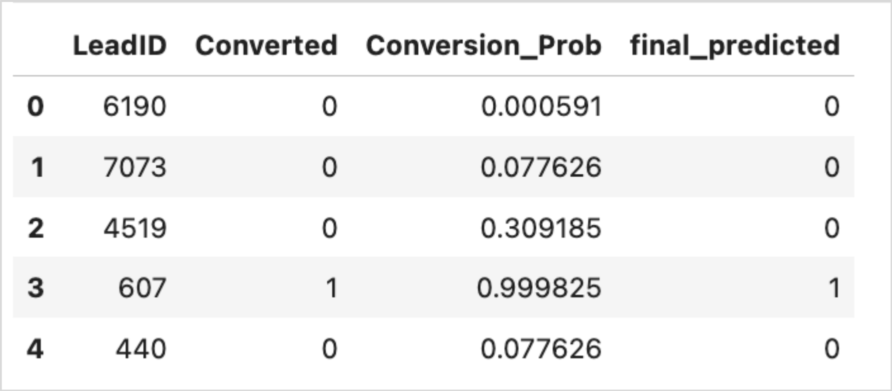
Probability
Threshold
Cutoff
0.34

Model Evaluation Metrics

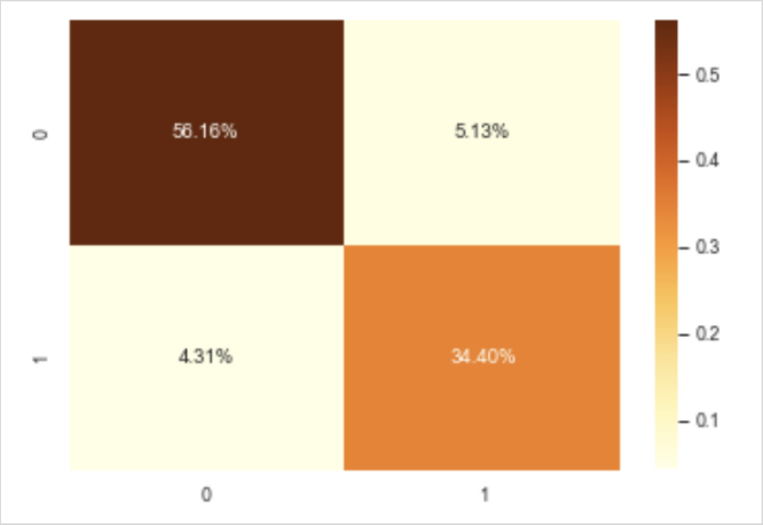
Metrics		Scores
Accuracy	TP +TN / (TP+TN+FN+FP)	0.903
Sensitivity	TP / (TP+FN)	0.887
Specificity	TN / (TN+FP)	0.913
False Positive Rate	FP/ (TN+FP)	0.087
Positive Predictive Value	TP / (TP+FP)	0.860
Negative Predictive Value	TN / (TN+ FN)	0.930
Precision	TP / TP + FP	0.861
Recall	TP / TP + FN	0.887
F1 Score	2×(Precision*Recall)/(Precision + Recall)	0.874
AUC (GINI)		0.962

Making Predictions on Test Dataset

Top 5 records from the Final Test Dataset



	LeadID	Converted	Conversion_Prob	final_predicted
0	6190	0	0.000591	0
1	7073	0	0.077626	0
2	4519	0	0.309185	0
3	607	1	0.999825	1
4	440	0	0.077626	0



- The final model on the Train dataset is used to make predictions for the Test dataset
- The Test data set was scaled using the scaler.transform function that was used to scale the Train dataset.
- The Predicted probabilities were added to the leads in the Test dataframe.
- Using the probability threshold value of 0.34, the leads from the Test dataset were predicted if they will convert or not.

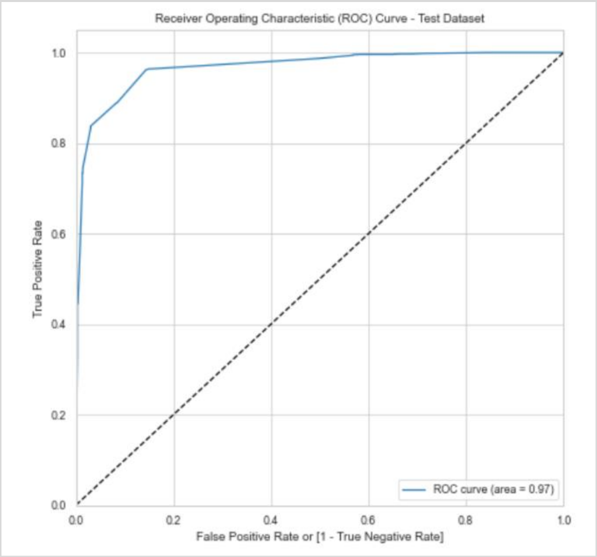
- The Conversion Matrix was calculated based on the Actual and Predicted 'Converted' columns.

Model Evaluation (on Test Dataset) | Metrics

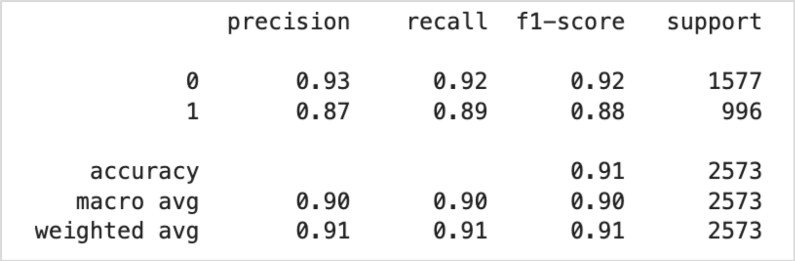
Model Evaluation Metrics

Metrics		Scores
Accuracy	TP +TN / (TP+TN+FN+FP)	0.906
Sensitivity	TP / (TP+FN)	0.889
Specificity	TN / (TN+FP)	0.916
False Positive Rate	FP/ (TN+FP)	0.084
Positive Predictive Value	TP / (TP+FP)	0.870
Negative Predictive Value	TN / (TN+ FN)	0.928
Precision	TP / TP + FP	0.870
Recall	TP / TP + FN	0.889
F1 Score	2×(Precision*Recall)/(Precision + Recall)	0.879
AUC (GINI)		0.968

Area Under Curve (GINI)



Classification Report



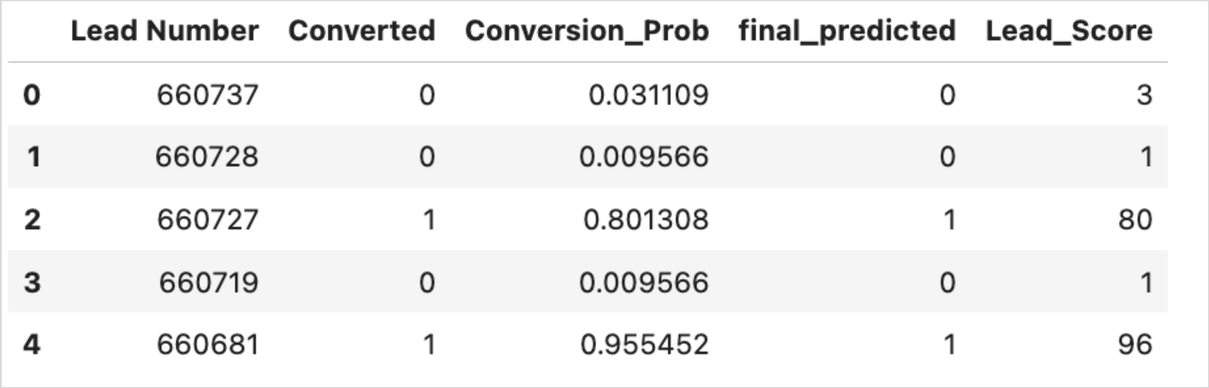
```
              precision    recall  f1-score   support

           0       0.93      0.92      0.92      1577
           1       0.87      0.89      0.88       996

    accuracy                           0.91      2573
   macro avg       0.90      0.90      0.90      2573
weighted avg       0.91      0.91      0.91      2573
```

Lead Score Calculation

Lead Score is calculated for all leads from the original dataset

Lead Score is derived using the below formula :

Lead Score = 100 * Conversion Probability



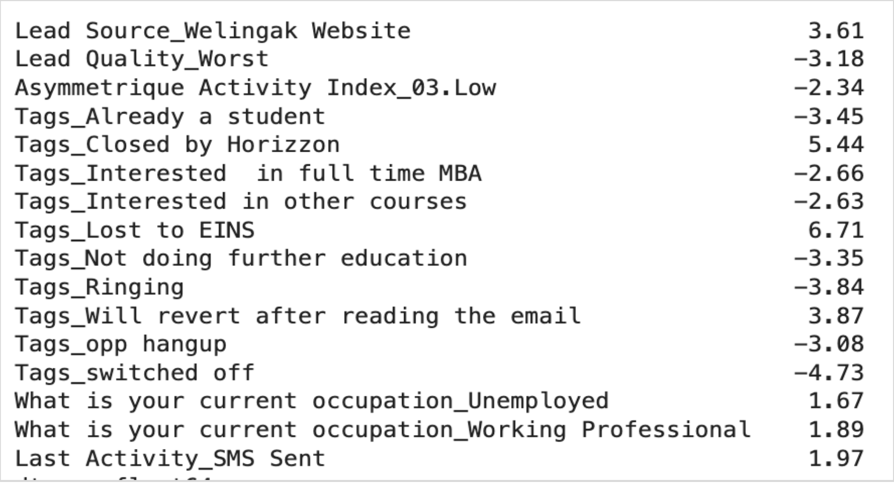
	Lead Number	Converted	Conversion_Prob	final_predicted	Lead_Score
0	660737	0	0.031109	0	3
1	660728	0	0.009566	0	1
2	660727	1	0.801308	1	80
3	660719	0	0.009566	0	1
4	660681	1	0.955452	1	96

Above table displays Lead Scores for top 5 records from the dataset

- The Train and Test dataset is concatenated to get the complete list of leads available.
- The Conversion Probability is multiplied by 100 to obtain the Lead Score for each lead.
- Higher the lead score, higher is the probability of a lead getting converted and vice versa
- Since, we had used 0.34 as our final Probability threshold for deciding if a lead will convert or not, any lead with a lead score of 35 or above will have a value of '1' in the final_predicted column.

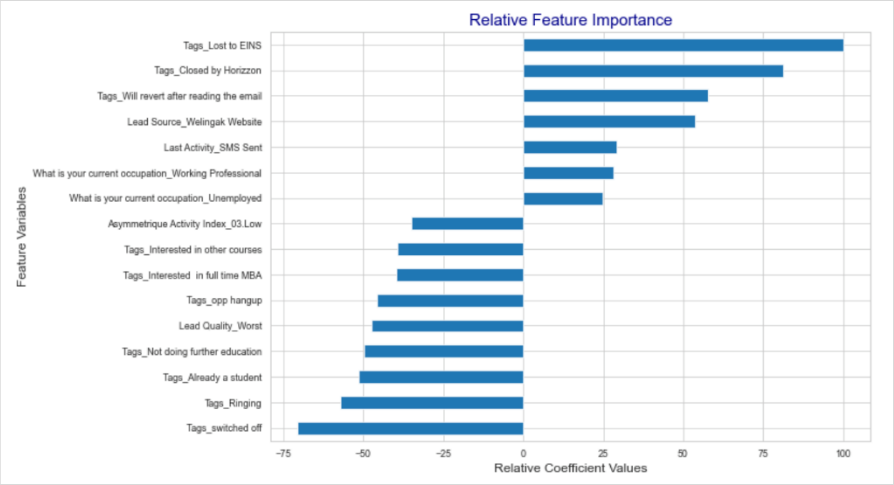
Calculating Relative Feature Importance

16 features that have been used by the model :



```
Lead Source_Welingak Website                          3.61
Lead Quality_Worst                                   -3.18
Asymmetrique Activity Index_03.Low                   -2.34
Tags_Already a student                               -3.45
Tags_Closed by Horizzon                               5.44
Tags_Interested  in full time MBA                    -2.66
Tags_Interested in other courses                     -2.63
Tags_Lost to EINS                                     6.71
Tags_Not doing further education                     -3.35
Tags_Ringing                                         -3.84
Tags_Will revert after reading the email              3.87
Tags_opp hangup                                      -3.08
Tags_switched off                                    -4.73
What is your current occupation_Unemployed            1.67
What is your current occupation_Working Professional  1.89
Last Activity_SMS Sent                                1.97
```

Feature Relative Importance Plot



- 16 features have been used by the model to successfully predict if a lead will get converted or not.
- Coefficient (beta) values for each of these features from model parameters are used to determine the order of importance for these features.
- Features with high +ve beta values are ones that contribute most towards the probability of a lead getting converted.
- Similarly, features with high –ve beta values contribute the least.

- The Relative Importance of each feature is determined on a scale of 100 with the feature with highest importance having a score of 100.
  - Feature importance = 100.0 * (feature_importance / feature_importance.max())
- The features are then sorted using np.quicksort algorithm.
- Lastly, the sorted features are plotted in a bar graph in descending order of their relative importance.

Summing It Up (1/2)

Post running multiple models and dropping variables iteratively, a model with following characteristics was finally chosen :

- The final 16 features have very low VIF values, indicating hardly any multi-collinearity among the features. This is also evident from the final correlations heatmap that is plotted.
- All variables have p-values < 0.05.
- Overall accuracy achieved is 0.9055 at a probability threshold of 0.34 which is also very acceptable.

Using this model, the dependent variable value was predicted as per the threshold value of Conversion probability (0.34):

Metrics (Test Dataset)	Scores
Accuracy	0.906
Sensitivity	0.889
Specificity	0.916
False Positive Rate	0.084
Positive Predictive Value	0.870
Negative Predictive Value	0.928
Precision	0.870
Recall	0.889
F1 Score	0.879
AUC (GINI)	0.968

Summing It Up (2/2)

Based on final model, some features are identified which contribute most to success of Lead conversion -

Lead Conversion Probability Increases With Increase In Values Of These Features (In Descending Order):

Features with Positive Coefficients Values
Tags_Lost to EINS
Tags_Closed by Horizzon
Tags_Will revert after reading the email
Lead Source_Welingak Website
Last Activity_SMS Sent
What is your current occupation_Working Professional
What is your current occupation_Unemployed

Lead Conversion Probability Increases With Decrease In Values Of These Features (In Descending Order):

Features with Negative Coefficients Values
Tags_switched off
Tags_Ringing
Tags_Already a student
Tags_Not doing further education
Lead Quality_Worst
Tags_opp hangup
Tags_Interested in full time MBA
Tags_Interested in other courses
Asymmetrique Activity Index_03.Low

Recommendations

Top 3 Variables Contributing Most To Leads Conversion Success

Top three variables in the model which contribute most towards the probability of a lead getting converted :

- **Tags_Lost to EINS**
- **Tags_Closed by Horizzon**
- **Tags_Will revert after reading the email**

Top 3 Categorical Variables To Increase Probability Of Lead Conversion

Top three categorical variables which should be focused the most on in order to increase the probability of lead conversion are:

- **Tags_Lost to EINS**
- **Tags_Closed by Horizzon**
- **Tags_Will revert after reading the email**

Strategy To Deploy When More Resources Available With Sales Teams

A **lower threshold value** for Conversion Probability cut-off can be chosen to ensure Sensitivity rating is very high which in turn will make sure almost all leads that are likely to Convert are identified correctly and Sales team members can make phone calls to as much of such leads as possible.

Strategy To Deploy When Sales Target Is Complete For Quarter

A higher threshold value for Conversion Probability can be chosen to ensure Specificity rating is very high, which in turn will make sure almost all leads that are on the brink of the probability of getting Converted or not are not selected. As a result, the Sales team will not have to make unnecessary phone calls and can focus on some new work.

THANK YOU!