Lead Scoring Case Study

A Logistic Regression Model built to predict the lead for an online courses education company X Education would be successfully converted or not.

By

Madhu Varadarajanna

Rachit Dev

Business Goals

- Provide X Education a list of most promising leads(Hot Leads), i.e. the leads that are most likely to convert into paying customers.
- A logistic regression model to be built to assign a lead score value ranging from 0 to 100 to each of the leads provided that can be used by the company to target potential leads for further actions.

To satisfy the Business goal steps to be followed:

- A Logistic Regression model creation to predict the Lead Conversion probabilities for each lead provided.
- Probability threshold value to be decided above which a lead can be predicted as converted Or converted if it is below it.
- Lead Conversion probability multiplied to arrive at the Lead Score value for all lead.

Approach

The approach taken to accomplish the project is to divide the entire case study into various sub tasks to reach the goals. Below are the list of tasks:

- Data understanding and Preparation
- Use RFE to get the best performing list of features for building the model.
- Build the model with based on RFE. Remove all features with high p-values and VIF values and get the final model.
- Model evaluation done with various metrics like sensitivity, specificity, precision, recall, etc.
- Get the probability threshold value based on Optimal cutoff point and predict the dependent variable for the training data.
- Implement the model to predict the test dataset and perform model evaluation for the test set.

Data Preparation

Data preparation processes followed:

- Remove columns having only one unique value : Such columns are least significant in prediction of lead score hence eliminated – ‘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 with high missing values: ‘Lead Source’ column found to be having high missing values. So such rows are deleted.
- Imputing NULL values with Median/Mode: ‘TotalVisits’ and ‘Page Views Per Visit’ were imputed with the column median values. ‘Country’ imputed the null values for this with mode(most occurring value).
- 'Select’ handled: Some columns in dataset has value called 'Select’. The Select are converted to Nulls.
- Creating Unique Category to NULL/SELECT values: Nulls columns are made separate column ‘Unknown’. Avoiding deleting columns with large null value percentage, this decision adds more information into the dataset and results in the change of variance. The Unknown values are later dropped during dummy encoding.
- Treating Outliers: The outliers in the columns 'TotalVisits‘ and 'Page Views Per Visit‘ were finally dropped based on interquartile range analysis.
- Binary Encoding (Yes/No) to 0/1 : 'Search','Do Not Email', 'Do Not Call', 'Newspaper Article', 'X Education Forums', 'Newspaper', 'Digital Advertisement','Through Recommendations‘ and 'A free copy of Mastering The Interview’ are converted.

Data Preparation Contd…

- Dummy variable creation for columns: '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’
- Splitting the data into TestTrain: The dataframe was split into train and test dataset. The train dataset used to train the model and test dataset used for model evaluation.
- Feature Scaling :Scaling is important to have all variables(specially categorical ones which has values 0 and 1) on the same scale to interpret the model result. ‘Yeo-Johnson’ PowerTransformer scaler was used to scale the data for modelling. This makes the data more Gaussian-like. The power transform finds the optimal scaling factor to stabilize variance and minimize skewness through maximum likelihood estimation.

```
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')
```

```
logreg = LogisticRegression()
```

```
rfe = RFE(logreg, 20)
rfe = rfe.fit(X_train, y_train)
```

RFE

Recursive feature elimination is an optimization technique for finding the best performing list of features. It is an iterative process of model building and getting to best fit model based on coefficients. The RFE is applied on all features in the dataset. Features are then ranked according to when they were eliminated.RFE ran for 20 variables.

	Features	VIF
4	Tags_Closed by Horizzon	1.28
8	Tags_Not doing further education	1.25
13	Tags_switched off	1.18
5	Tags_Interested in full time MBA	1.11
0	Lead Source_Welingak Website	1.08
11	Tags_invalid number	1.07
2	Asymmetrique Activity Index_03.Low	1.07
7	Tags_Lost to EINS	1.06
12	Tags_opp hangup	1.02
15	What is your current occupation_Working Professional	0.78
1	Lead Quality_Worst	0.67
9	Tags_Ringing	0.59
6	Tags_Interested in other courses	0.38
3	Tags_Already a student	0.37
10	Tags_Will revert after reading the email	0.09
14	What is your current occupation_Unemployed	0.01
16	Last Activity_SMS Sent	0.00

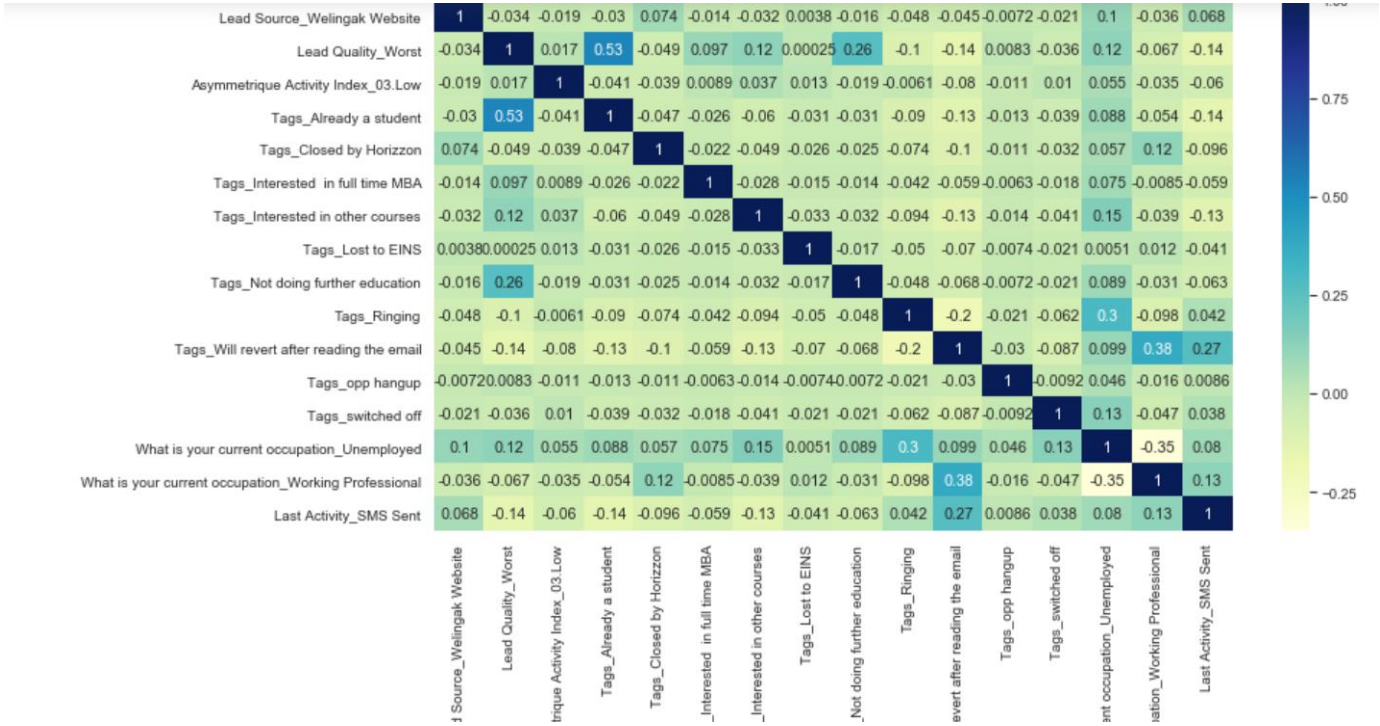
	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4751	0.088	-28.144	0.000	-2.647	-2.303
Lead Source_Welingak Website	3.6135	0.730	4.949	0.000	2.182	5.044
Lead Quality_Worst	-3.1794	0.670	-4.742	0.000	-4.494	-1.865
Asymmetrique Activity Index_03.Low	-2.3401	0.354	-6.605	0.000	-3.035	-1.646
Tags_Already a student	-3.4492	0.722	-4.776	0.000	-4.865	-2.034
Tags_Closed by Horizzon	5.4435	0.720	7.559	0.000	4.032	6.855
Tags_Interested in full time MBA	-2.6565	0.740	-3.591	0.000	-4.106	-1.207
Tags_Interested in other courses	-2.6347	0.327	-8.060	0.000	-3.275	-1.994
Tags_Lost to EINS	6.7102	0.862	7.786	0.000	5.021	8.399
Tags_Not doing further education	-3.3472	1.030	-3.250	0.001	-5.366	-1.329
Tags_Ringing	-3.8360	0.244	-15.709	0.000	-4.315	-3.357
Tags_Will revert after reading the email	3.8695	0.190	20.331	0.000	3.497	4.243
Tags_opp hangup	-3.0789	1.061	-2.903	0.004	-5.158	-1.000
Tags_switched off	-4.7274	0.722	-6.544	0.000	-6.143	-3.311
What is your current occupation_Unemployed	1.6711	0.112	14.926	0.000	1.452	1.891
What is your current occupation_Working Professional	1.8944	0.363	5.221	0.000	1.183	2.606
Last Activity_SMS Sent	1.9687	0.107	18.383	0.000	1.759	2.179

Model Building

Stats Models is used to build the Logistic Regression model. 20 variables selected by RFE to build the model. Features that are not required are dropped sequentially after checking p values (< 0.5) and VIF (< 5) and model is built multiple times. The final model having 16 features, passes significance test and the multi-collinearity test.

Model Building Contd…

Heat map for 16 features:



	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

	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

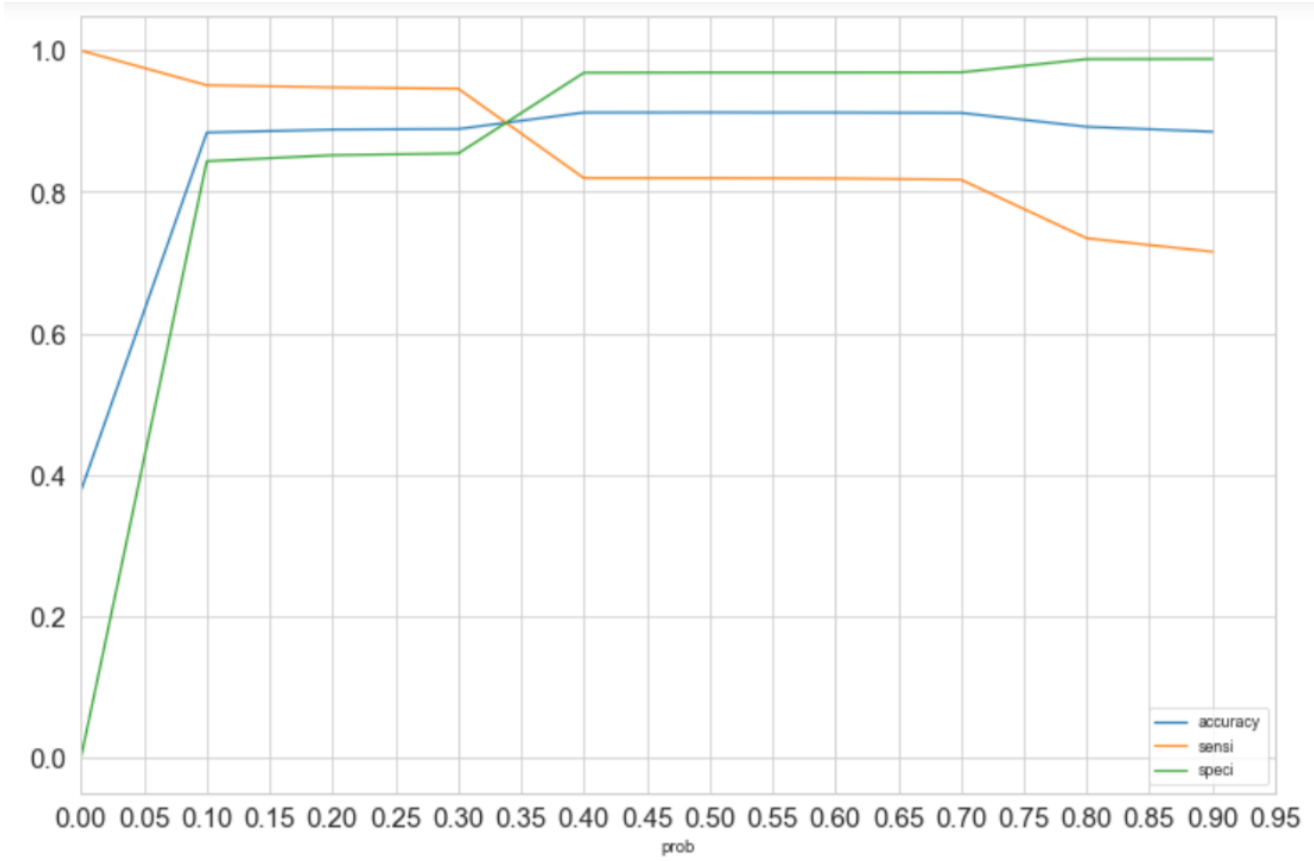
Conversion probability predicted columns

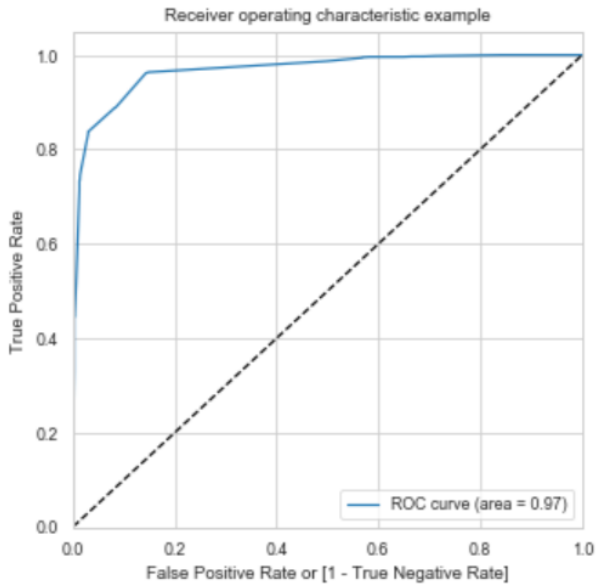
The actual Converted flag and the predicted probabilities. Showing top 5 records of the dataframe in the picture on the left.

Creating new column 'predicted' with 1 if Conversion_Prob > 0.5 else 0 Showing top 5 records of the dataframe in the picture

Optimal probability threshold

- Accuracy sensitivity and specificity was calculated for various values of probability threshold and plotted in the graph to the right.
- 0.33 is found to be the optimum point for cutoff probability.
- All the 3 metrics - accuracy sensitivity and specificity was found to be well above 80% which is a well acceptable value.





As a rule of thumb, an AUC can be classed as follows,

0.90 - 1.00 = excellent

0.80 - 0.90 = good

0.70 - 0.80 = fair

0.60 - 0.70 = poor

0.50 - 0.60 = fail

Auc is 0.9678, our model seems to be doing well on the test dataset.

ROC Curve and AUC Calculation

- Receiver Operating Characteristics (ROC) Curve
- Area under the Curve (GINI)

Model Evaluation – Train Dataset

Confusion Matrix

Accuracy Score

(TP +TN/ (TP+TN+FN+FP)):

```
confusion1 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
confusion1
```

```
array([[3411,  325],
       [ 256, 2010]], dtype=int64)
```

```
TP = confusion1[1,1] # true positive
TN = confusion1[0,0] # true negatives
FP = confusion1[0,1] # false positives
FN = confusion1[1,0] # false negatives
```

```
#sensitivity
TP / float(TP+FN)
```

```
0.8870255957634599
```

```
#specificity
TN / float(TN+FP)
```

```
0.9130085653104925
```

```
#false postive rate
print(FP/ float(TN+FP))
```

```
0.0869914346895075
```

```
# Positive predictive value
print (TP / float(TP+FP))
```

```
0.860813704496788
```

```
# Negative predictive value
print (TN / float(TN+ FN))
```

```
0.9301881647122989
```

```
metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
```

```
0.9031989336887704
```

Precision TP / TP + FP

```
precision = confusion1[1,1]/(confusion1[0,1]+confusion1[1,1])
precision
```

```
0.860813704496788
```

Recall TP / TP + FN

```
recall = confusion1[1,1]/(confusion1[1,0]+confusion1[1,1])
recall
```

```
0.8870255957634599
```

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

```
confusion_test = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.final_predicted )
print(confusion_test)
```

```
[[1445  132]
 [ 111  885]]
```

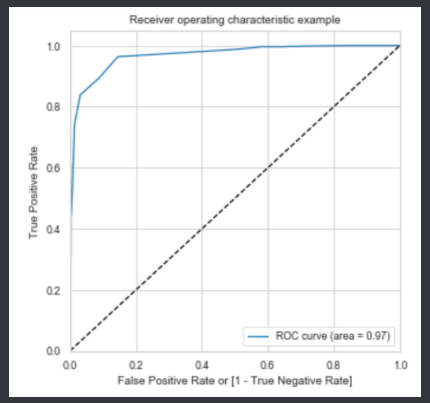
```
TP = confusion_test[1,1] # true positive
TN = confusion_test[0,0] # true negatives
FP = confusion_test[0,1] # false positives
FN = confusion_test[1,0] # false negatives
```

Predictions on test dataset

The Predicted probabilities are added to the leads in the test dataframe. Using the probability threshold value of 0.33, the leads from the test dataset were predicted if they will convert or not.

Model evaluation on test dataset

ROC Curve



Cross Validation Score To avoid overfitting, calculate the Cross Validation Score to see how our model performs

```
lr = LogisticRegression(solver = 'lbfgs')
scores = cross_val_score(lr, X, y, cv=10)
scores.sort()
accuracy = scores.mean()

print(scores)
print(accuracy)
```

```
[0.84364061 0.89731622 0.90898483 0.91248541 0.91501746 0.92424242
 0.92665891 0.92882147 0.92998833 0.9369895 ]
0.9124145163173883
```

```
#accuracy.
acc_score=metrics.accuracy_score(y_pred_final.Converted, y_pred_final.final_predicted)
acc_score
```

```
0.9055577147298873
```

Recall TP / TP + FN

```
Recall = confusion_test[1,1]/(confusion_test[1,0]+confusion_test[1,1])
Recall
```

```
0.8885542168674698
```

```
F1 = 2*(Precision*Recall)/(Precision+Recall)
F1
```

```
0.879284649776453
```

```
print(classification_report(y_pred_final.Converted, y_pred_final.final_predicted))
```

```
              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
```

```
# false postive rate
print(FP/ float(TN+FP))
```

```
0.08370323398858592
```

```
# Positive predictive value
print (TP / float(TP+FP))
```

```
0.8702064896755162
```

```
# Negative predictive value
print (TN / float(TN+ FN))
```

```
0.9286632390745502
```

Sensitivity TP / TP + FN

```
#sensitivity
TP / float(TP+FN)
```

```
0.8885542168674698
```

Specificity TN / TN + FP

```
#specificity
TN / float(TN+FP)
```

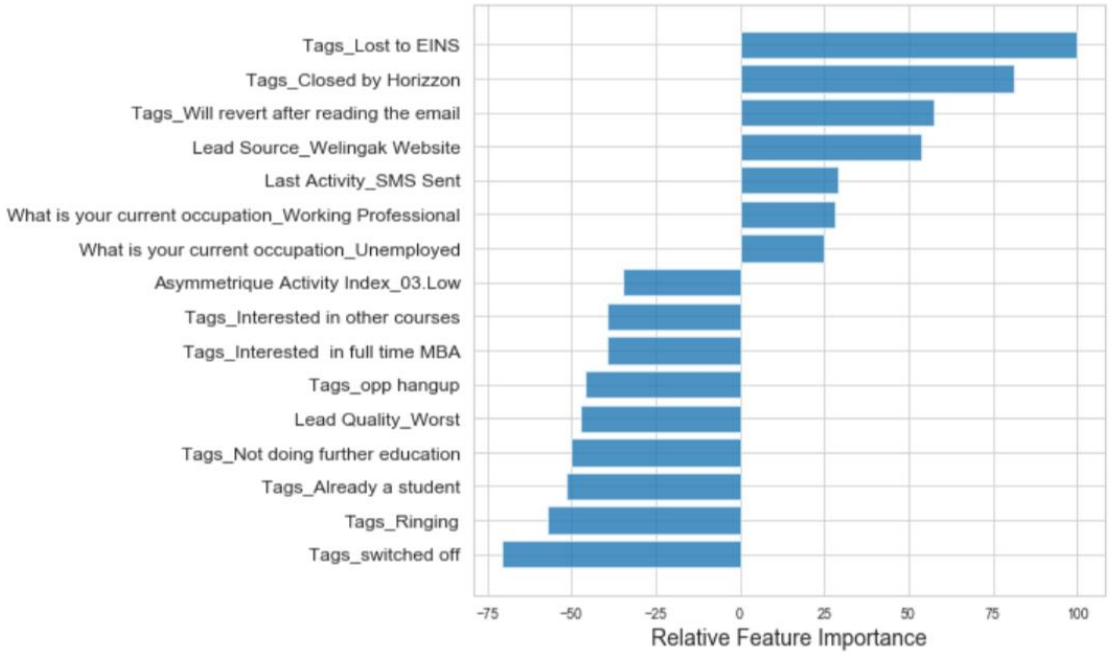
```
0.9162967660114141
```

Calculating the lead score

- Lead Score = 100 * Conversion Probability
- The train and test dataset is concatenated to get the entire list of leads available.
- Higher the lead score, higher is the probability of a lead getting converted and vice versa,
- We have used 0.33 as our final Probability threshold for deciding if a lead will convert or not, any lead with a lead score of 34 or above will have a value of ‘1’ in the final_predicted column.

	Lead Number	Conversion_Prob	Converted	final_predicted	Lead_Score
0	660737	0.031109	0	0	3
1	660728	0.009566	0	0	1
2	660727	0.801308	1	1	80
3	660719	0.009566	0	0	1
4	660681	0.955452	1	1	96
5	660680	0.077626	0	0	8
6	660673	0.955452	1	1	96
7	660664	0.077626	0	0	8
8	660624	0.077626	0	0	8
9	660616	0.077626	0	0	8

Feature determination



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

Asymmetrique Activity Index_03.Low

Tags_Interested in other courses

Tags_Interested in full time MBA

Tags_opp hangup

Lead Quality_Worst

Tags_Not doing further education

Tags_Already a student

Tags_Ringing

Tags_switched off

Conclusion

Based on our model, some features are identified which contribute most to a Lead getting converted successfully:

The conversion probability of a lead increases with increase in values of the following features in descending order:

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

The conversion probability of a lead increases with decrease in values of the following features in descending order:

Problem Solution with recommendations

Which are the top three variables in your model that 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

What are the top 3 categorical/dummy variables in the model which get maximum focus in order to increase the probability of lead conversion?

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

X Education has a period of 2 months every year during which they hire few interns. The sales team, in particular, has around 10 interns allotted to them. So, during this phase, they wish to make the lead conversion more aggressive. So they want almost all of the potential leads (i.e. the customers who have been predicted as 1 by the model) to be converted and hence, want to make phone calls to as much of such people as possible. Suggest a good strategy they should employ at this stage.

We will choose a lower threshold value for Conversion Probability. This will ensure the Sensitivity rating is very high which in turn will make sure almost all leads that are likely to Convert are identified correctly and the agents can make phone calls to as much of such people as possible.

Similarly, at times, the company reaches its target for a quarter before the deadline. During this time, the company wants the sales team to focus on some new work as well. So during this time, the company's aim is to not make phone calls unless it's extremely necessary, i.e. they want to minimize the rate of useless phone calls. Suggest a strategy they should employ at this stage.

We will choose a higher threshold value for Conversion Probability. This will ensure the 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 agents won't have to make unnecessary phone calls and can focus on some new work.