Lead Score Assignment

Submitted By:

Minakshi Maurya & Kushagra Misra

Outline

- Problem Statement
- Data Understanding
- Data Cleaning
- Data Visualization
- Data Preparation
- Modelling
- Prediction on Test dataset
- Conclusion
- Proposal

Problem Statement

IDEAL:

Sales Team of X Education should be able to make high conversion rate from leads(80%) .

REALITY:

In reality they are having 38.54 conversion rate for the leads based on different variables.

CONSEQUENCES:

Due to random calling to leads, conversion rate is very low which is resulting in business loss.

PROPOSAL:

Logistic regression model will be built to identify the most promising leads, i.e. the leads that are most likely to convert into paying customers. Lead score will be assigned to each of the leads ,with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance. Score will be assigned based on the target lead conversion rate to be around 80%.

Data Understanding

- The shape of the dataset is 9240x37
- Original conversion rate of company X is 38.54%.
- Large number of 'Select' values present for Lead Profile and City in the dataset. These values correspond to the user, having not made any selection.
- There are 7 numerical columns and 30 categorical columns.
- There are some columns with over 50% of null values.
- Lead Number and Prospect ID are columns with unique id with no duplicate value.

```
# Checking shape of dataframe
df.shape
```

```
(9240, 37)
```

```
# Checking columns name
df.columns
```

```
Index(['Prospect ID', 'Lead Number', 'Lead Origin', 'Lead Source',
       'Do Not Email', 'Do Not Call', 'Converted', 'TotalVisits',
       'Total Time Spent on Website', 'Page Views Per Visit', 'Last Activity',
       'Country', 'Specialization', 'How did you hear about X Education',
       'What is your current occupation',
       'What matters most to you in choosing a course', 'Search', 'Magazine',
       'Newspaper Article', 'X Education Forums', 'Newspaper',
       'Digital Advertisement', 'Through Recommendations',
       'Receive More Updates About Our Courses', 'Tags', 'Lead Quality',
       'Update me on Supply Chain Content', 'Get updates on DM Content',
       'Lead Profile', 'City', 'Asymmetrique Activity Index',
       'Asymmetrique Profile Index', 'Asymmetrique Activity Score',
       'Asymmetrique Profile Score',
       'I agree to pay the amount through cheque',
       'A free copy of Mastering The Interview', 'Last Notable Activity'],
      dtype='object')
```

```
# checking attributes for continuous variables
df.describe()
```

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
count	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000000
mean	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.344883
std	23405.995698	0.486714	4.854853	548.021466	2.161418	1.386694	1.811395
min	579533.000000	0.000000	0.000000	0.000000	0.000000	7.000000	11.000000
25%	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.000000
50%	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.000000
75%	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.000000
max	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.000000

Data Cleaning

- Data cleaning has been performed in 3 steps

    1. Missing / Null value imputation
    2. Column drop(not needed for model)
    3. Outliers Treatment.

- After Data cleaning we have left with 12 columns

Missing/Null Values Imputation:

- There are a large number of 'Select' values spread across the dataset. These values meant that the user had made no selection in those fields. We decided to replace these values with NaN values and treated them appropriately later.
- Then we conducted an analysis on the percentage of null columns in the dataset, tackling the ones with the highest percentage of null values first. We have removed columns that had over 70% null values and for the remaining columns assessed them individually.
- Certain columns had a large mix of values, some outliers and a small number of null values. We had to perform appropriate outlier & null value treatment for each of these.
- In certain cases, such as that of the Specialization column, we could not have taken the Mode value to impute the null columns. This is because we had to consider the fact that the mentioned options in the form might not have represented the applicant’s specialization correctly. We decided to club these values into one field and later assess them.
- In the case of numerical columns like TotalVisits we imputed the null values with the median value. This is because the difference between the median and mean was very less.

Data Cleaning Continue…….

Column Drop:

- While assessing the columns individually we found that various columns were actually summarized into one column already. Therefore, it did not make sense for us to keep these columns and we decided to drop them entirely. Examples of such columns are Search, Newspaper Article etc., they are already represented in the 'Lead Source' column. The distribution represented in these individual columns was very well represented by the data in the Lead Source column.
- There were a few columns that had highly skewed data, i.e. data pointing in one direction only. The Country, what matters most to you in choosing a course, are a few examples. Most of the leads, 95% and above, mentioned that they were from India and were looking for better career prospects. We dropped these columns as well. The tendency of skewed data to sway the model heavily towards its direction makes the model incapable of predicting the results correctly. Below are the other columns dropped for the same:

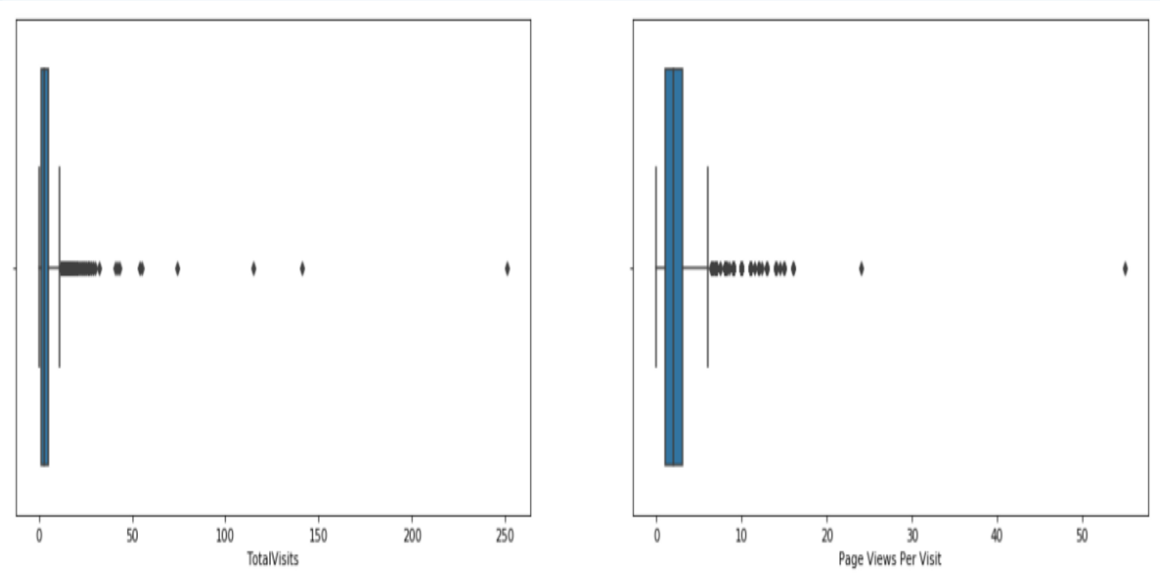
Colums	Description
Tags	Tags are added by the sales team
Country	Highly Skewed
What matters most to you in choosing a course	Highly Skewed.All candidates that take this course are looking to have a better career
Last Notable Activity	Last Activity and Last Notable seems same
Do Not Call	Highly Skewed
Do Not Email	Highly Skewed
Prospect ID	Its Unique and Lead no is also availabe, which can be used
Get updates on DM Content	Highly Skewed
Update me on Supply Chain Content	
I agree to pay the amount through cheque	
Receive More Updates About Our Courses	
Magazin	
Asymmetrique Activity Index	Null Values are high and moreover assigned by sales Team after Call
Asymmetrique Profile Index	
Asymmetrique Activity Score	
Asymmetrique Profile Score	
Lead Quality	Not Sure' are considerable high at 63.14%

Data Cleaning Continue…..

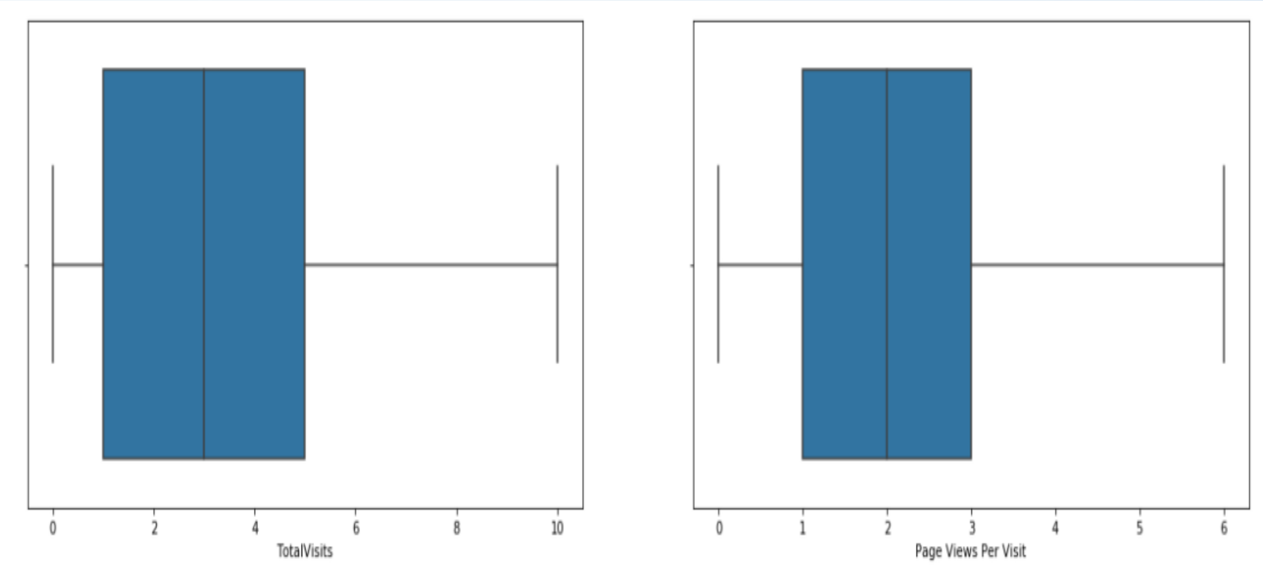
Outliers Treatment: If Outliers are present, instead of dropping them so that all the rows of data are retained. Capping was done with by replacing the lowest values with the 1%ile & highest with the 95%ile value in the column.

- ❑ TotalVisits
- ❑ Page Views per Visit

Before Outlier treatment:



After Outlier Treatment:

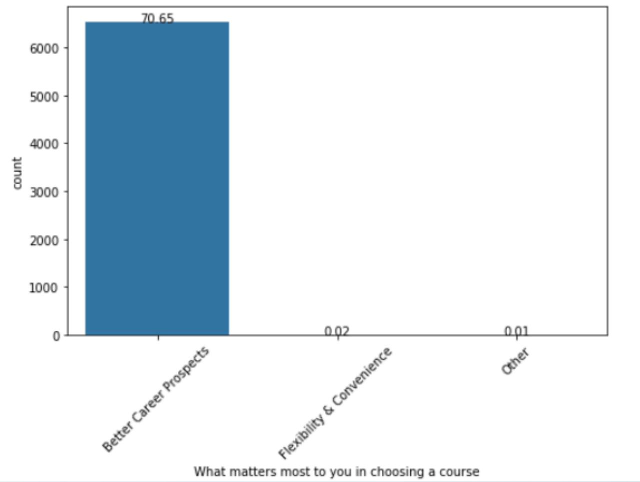


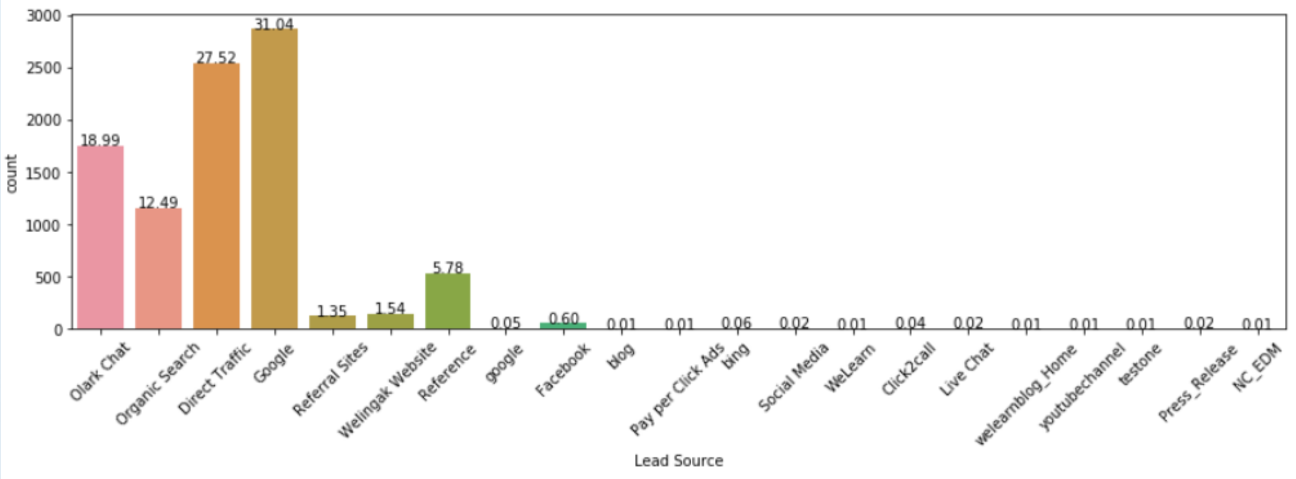
Data Visualization

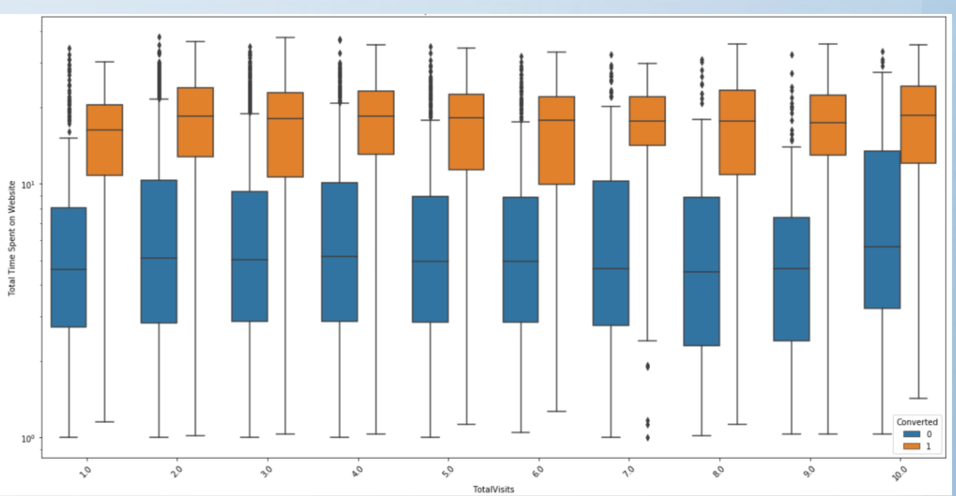
While analysing the various columns we performed univariate and bivariate analysis on them. Bivariate analysis was carried out with the Converted column as a benchmark. This analysis yielded some very important insights that we have mentioned below. Key being that the longer the user stayed on the website, the higher the chances of them converting.

Of particular interest to us was the 'Total Time Spent on The Website' column. This column had highly varied data that we had to properly convert to correct metrics to make better sense of it.

- ❑ Most applicants would like to join a course to have better career prospects
- ❑ X Education has the highest conversion rate of individuals who are referred to them
- ❑ Overall, it is safe to say that the more time the user spends on the website, the better their chances of becoming a student.







Data Preparation

- We created dummy variables from final 12 variables and correctly dropped all the original columns, other category variables that we had created.
- Numerical column has been scaled using Standard scaler so that all the variables follow similar units
- train-test split has been performed using the 70-30 method for splitting.
- Assessed the split datasets and plotted a correlation heatmap to identify any variables with high collinearity. We found 2 such variables and dropped them from both the training and test sets.
- We assessed the split datasets and plotted a correlation heatmap to identify any variables with high collinearity. We found 2 such variables and dropped them from both the training and test sets

```
# Splitting the data into train and test

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3, random_state=100)
```

```
X_train.shape
```

```
(6468, 58)
```

```
X_test.shape
```

```
(2772, 58)
```

Modelling

Final Model has been generated with 12 variables on which satisfies our condition of 80% sensitivity and below attributes

- The VIF values are under 3
- The p values are under 0.05
- No Multi Collinearity

- **Basic Model :** Logistic Regression model has been created with all our features from the scaled training dataset. The GLM summary *report* from this model provided us the base benchmarks for our model.
- **Model Based on RFE Selection :** Based on the above criteria we performed RFE with 15 variables and began creating and the models. We eliminated any variables that had high p values and VIF values. Eventually we generated a model with 12 variables that we performed training and testing on.

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:**	-2799.5
Date:	Sun, 06 Sep 2020	**Deviance:**	5599.0
Time:	15:15:07	**Pearson chi2:**	9.11e+03
No. Iterations:	7		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.3837	0.179	7.726	0.000	1.033	1.735
Total Time Spent on Website	1.0659	0.038	27.980	0.000	0.991	1.141
Lead Source_Olark chat	1.1280	0.100	11.322	0.000	0.933	1.323
Lead Source_Reference	3.5984	0.201	17.914	0.000	3.205	3.992
Lead Source_Welingak website	5.4963	0.727	7.556	0.000	4.071	6.922
Last Activity_Converted to Lead	-1.2127	0.217	-5.585	0.000	-1.638	-0.787
Last Activity_Email Bounced	-1.7984	0.282	-6.383	0.000	-2.351	-1.246
Last Activity_Had a Phone Conversation	2.1604	0.652	3.314	0.001	0.883	3.438
Last Activity_Olark Chat Conversation	-1.4009	0.162	-8.624	0.000	-1.719	-1.083
Last Activity_SMS Sent	1.1884	0.072	16.449	0.000	1.047	1.330
What is your current occupation_Other	-2.8435	0.819	-3.471	0.001	-4.449	-1.238
What is your current occupation_Student	-2.3752	0.286	-8.313	0.000	-2.935	-1.815
What is your current occupation_Unemployed	-2.7984	0.180	-15.540	0.000	-3.151	-2.445

	Features	VIF
11	What is your current occupation_Unemployed	2.01
1	Lead Source_Olark chat	1.73
8	Last Activity_SMS Sent	1.50
7	Last Activity_Olark Chat Conversation	1.43
0	Total Time Spent on Website	1.21
2	Lead Source_Reference	1.09
4	Last Activity_Converted to Lead	1.09
5	Last Activity_Email Bounced	1.07
3	Lead Source_Welingak website	1.04
10	What is your current occupation_Student	1.03
6	Last Activity_Had a Phone Conversation	1.01
9	What is your current occupation_Other	1.00

Modelling Continues…..

Confusion Matrix:

On both our training and testing models we predicted the probability score for converting the leads and correctly added them to a new table along with the Lead ID with cut-off .5 . Once we had our scores and probability of converting a lead, we checked for the accuracy, specificity, sensitivity. Our model is having less sensitivity.

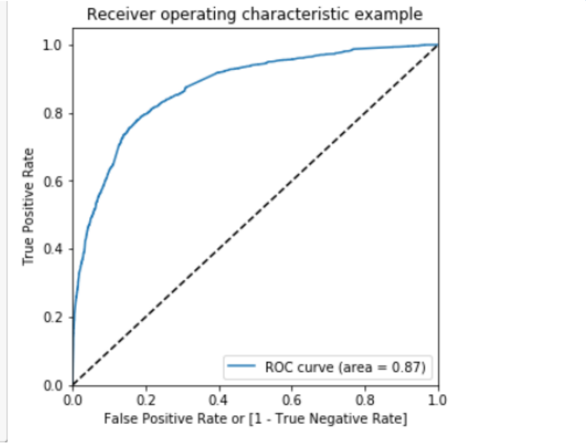
```
In [183]: y_train_pred_final['Final_Predicted_Hot_Lead'] = y_train_pred_final.Lead_Score_Prob.map(lambda x: 1 if x > 0.5 else 0)
          y_train_pred_final.head()
Out[183]:
```

	Converted	Lead_Score_Prob	Lead	Final_Predicted_Hot_Lead
0	0	0.226122	1871	0
1	0	0.196527	6795	0
2	0	0.264404	3516	0
3	0	0.773650	8105	1
4	0	0.226122	3934	0

```
In [184]: confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Final_Predicted_Hot_Lead)
          print(confusion)

          [[3575  427]
           [ 884 1582]]

In [185]: # Let's check the overall accuracy.
          print(round(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Final_Predicted_Hot_Lead),2))

          0.8
```

	Converted	Lead_Score_Prob	Lead	Final_Predicted_Hot_Lead	Lead_Score
0	0	0.226122	1871	0	23
1	0	0.196527	6795	0	20
2	0	0.264404	3516	0	26
3	0	0.773650	8105	1	77
4	0	0.226122	3934	0	23

```
In [188]: TP = confusion[1,1] # true positive
          TN = confusion[0,0] # true negatives
          FP = confusion[0,1] # false positives
          FN = confusion[1,0] # false negatives

In [189]: # Let's see the sensitivity of our logistic regression model
          round((TP / float(TP+FN)),2)
Out[189]: 0.64

In [190]: # Let us calculate specificity
          round((TN / float(TN+FP)),2)
Out[190]: 0.89
```



ROC Curve:

- ❑ We generated the ROC curve shows the tradeoff between sensitivity and specificity
- ❑ ROC curve shows that model accuracy is good as closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.

Modeling Continues……..

Getting Optimal Cut-off : Optimal cutoff probability is that prob where we get balanced sensitivity and specificity. For the model we got optimal cutoff at .33

- After getting Optimal cut-off confusion matrix, accuracy, sensitivity, specificity, recall and precision has been calculated.

- Through Optimal cutoff Accuracy, sensitivity and specificity for train model are 80%,80% and 79% respectively.

- Precision and recall after optimal cutoff are 70% and 80%

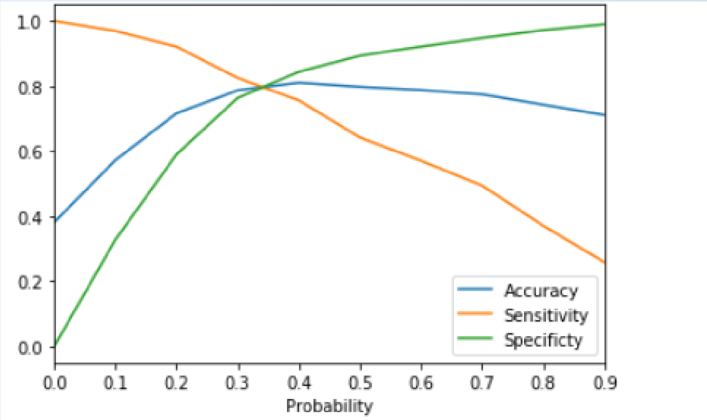
```
### Calculating Precision
precision =round(TP/float(TP+FP),2)
precision
```

```
0.7
```

```
### Calculating Recall
recall = round(TP/float(TP+FN),2)
recall
```

```
0.8
```

- F1- Score is 75% , hence we can say model is accurate.




```
# Accuracy
round(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Final_Predicted_Hot_Lead),2)
```

```
0.8
```

```
confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Final_Predicted_Hot_Lead )
confusion2
```

```
array([[3169,  833],
       [ 490, 1976]], dtype=int64)
```

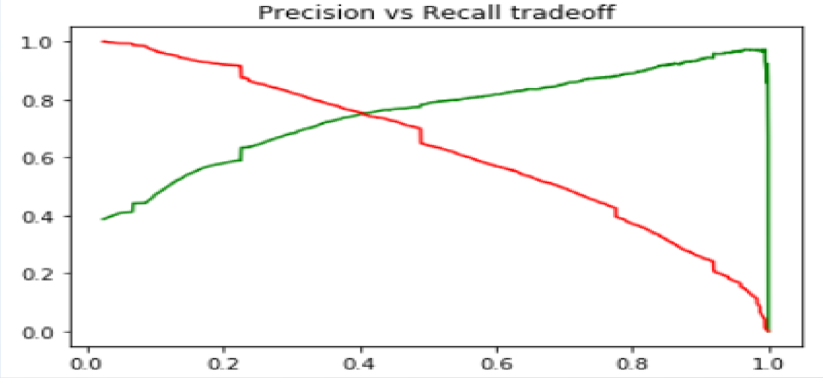
```
TP = confusion2[1,1] # true positive
TN = confusion2[0,0] # true negatives
FP = confusion2[0,1] # false positives
FN = confusion2[1,0] # false negatives
```

```
# Let's see the sensitivity of our logistic regression model
round(TP / float(TP+FN),2)
```

```
0.8
```

```
# Let us calculate specificity
round(TN / float(TN+FP),2)
```

```
0.79
```



Prediction on Test Dataset

With model Cut-off at .33 Test data is also having all 3 Specificity, Accuracy and Specificity are 80%

Making predictions on the test set

```
y_test_pred = res.predict(X_test_sm)
```

```
y_test_pred[:10]
```

```
4269    0.690380
2376    0.918972
7766    0.635218
9199    0.067156
4359    0.775571
9186    0.505920
1631    0.405401
8963    0.137610
8007    0.052129
5324    0.297859
dtype: float64
```

```
# Converting y_pred to a dataframe which is an array
y_pred_1 = pd.DataFrame(y_test_pred)
y_pred_1.head()
```

	0
4269	0.690380
2376	0.918972
7766	0.635218
9199	0.067156
4359	0.775571

```
y_pred_final.shape
```

```
(2772, 3)
```

```
# Renaming the column
y_pred_final= y_pred_final.rename(columns={ 0 : 'Lead_Score_Prob'})
```

```
# Rearranging the columns
y_pred_final = y_pred_final.reindex(['Lead','Converted','Lead_Score_Prob'], axis=1)
```

```
# Adding Lead_Score column
y_pred_final['Lead_Score'] = round((y_pred_final['Lead_Score_Prob'] * 100),0)
y_pred_final['Lead_Score'] = y_pred_final['Lead_Score'].astype(int)
```

```
# Let's see the head of y_pred_final
y_pred_final.head()
```

	Lead	Converted	Lead_Score_Prob	Lead_Score
0	4269	1	0.690380	69
1	2376	1	0.918972	92
2	7766	1	0.635218	64
3	9199	0	0.067156	7
4	4359	1	0.775571	78

```
y_pred_final['Final_Predicted_Hot_Lead'] = y_pred_final.Lead_Score_Prob.map(lambda x: 1 if x > 0.33 else 0)
```

	Lead	Converted	Lead_Score_Prob	Lead_Score	Final_Predicted_Hot_Lead
0	4269	1	0.690380	69	1
1	2376	1	0.918972	92	1
2	7766	1	0.635218	64	1
3	9199	0	0.067156	7	0
4	4359	1	0.775571	78	1

```
# Let's check the overall accuracy.
round(metrics.accuracy_score(y_pred_final.Converted, y_pred_final.Final_Predicted_Hot_Lead),2)
```

```
0.8
```

```
confusion3 = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.Final_Predicted_Hot_Lead )
confusion3
```

```
array([[1346,  331],
       [ 217,  878]], dtype=int64)
```

```
TP = confusion3[1,1] # true positive
TN = confusion3[0,0] # true negatives
FP = confusion3[0,1] # false positives
FN = confusion3[1,0] # false negatives
```

```
# Let's see the sensitivity of our logistic regression model
round((TP / float(TP+FN)),2)
```

```
0.8
```

```
# Let us calculate specificity
round(TN / float(TN+FP),2)
```

```
0.8
```

Conclusion

Logistic Regression Model has been created with the below acceptance criteria

log odds = 1.3837 +(1.0659 Total Time Spent on Website) + (1.1280 Lead Source_Olark chat) + (3.5984 Lead Source_Reference) + (5.4963 Lead Source_Welingak website) + (-1.2127 Last Activity_Converted to Lead) + (-1.7984 Last Activity_Email Bounced) + (2.1604 Last Activity_Had a Phone Conversation) + (-1.4009 Last Activity_Olark Chat Conversation) + (1.1884 Last Activity_SMS Sent)+(-2.8435 What is your current occupation_Other)+(-2.3752 What is your current occupation_Student)+(-2.7984 * What is your current occupation_Unemployed)

- ❑ The model does not over-fit
- ❑ The model is simple enough to be understood
- ❑ The model is built using significant features.
- ❑ The VIF value is under 5 & the p value is under 0.05 for each feature
- ❑ The accuracy, sensitivity and specificity of our model after test are at least 80% (+- 1% between all 3 parameters)
- ❑ Lead Score has been assigned on both Train and test dataset with cut-off .33 Probability as well as whole dataframe.

Hot Leads: Leads are having more than .33 probability are Hot leads

Cold Leads: Leads are having less than .33 probability are Cold leads

Top 3 Variables, contributing more towards probability of a lead getting converted

- ✓ Lead Source
- ✓ Total Time Spent on Website
- ✓ Last Activity

Top 3 Variables, contributing more towards probability of a lead getting converted

- ✓ Lead Source_Welingak website
- ✓ LeadSource_Reference
- ✓ Last Activity_Had a Phone Conversation

Proposal

X Education Sales Team should give attention to below key pointes inferred from the model to make conversion rate high:

- ❑ Organization should come-up with more effective incentive offers(referral bonus, discount as per company policy) to convert the leads into student.
- ❑ Working professionals will be better leads due to their high conversion percentage.
- ❑ They should follow-up more with the referred individuals and the leads spending more time on website as their conversion rate are high.
- ❑ Overall, it is safe to say that the more time the user spends on the website, the better their chances of becoming a student.
- ❑ Leads who had call to our customer care or sent messages as their last activity should be our targeted customer.
- ❑ They should first focus on the 'Hot Leads‘(Leads having score of 33 or above)
- ❑ Higher the Lead Score, higher the chances of conversion of 'Hot Leads' into 'Paying Customers‘
- ❑ The 'Cold Leads'(Customer having lead score < 33) should be focused after the Sales Team is done with the 'Hot Leads‘ and should provide some discount as per company policy and do follow-ups to clear their doubts about the platform.

Thank you