Leads Score Case Study

RAJ NIMESH – DS-C44

Business Problem

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%.

The aim of this model is to identify the most promising leads by giving a score between 1 to 100 to all the leads based on various factors and increase the conversion rate to 80%.

Approach and Methodology

The problem was approached using Logistic Regression Machine Learning Algorithm

Steps followed:

1. Inspecting the Data Frame
2. Exploratory Data Analysis
3. Data Preparation
4. Building The Model
5. Feature Scaling
6. Building the Correlation Metrix
7. Build the Model
8. Plotting the ROC Curve
9. Finding the Optimal Cutoff Point
10. Making Predictions on Test Set

Inspecting the Data Frame

- Overall there are 9240 rows and 37 columns

- While most of the columns seem to be in correct data type format, some of the columns have missing values

```
In [4]: # Checking the dimesions of the dataframe
        leads.shape

Out[4]: (9240, 37)

In [5]: # Checking type of each column
        leads.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 37 columns):
 #   Column                                         Non-Null Count  Dtype
---  ------                                         --------------  -----
 0   Prospect ID                                    9240 non-null   object
 1   Lead Number                                    9240 non-null   int64
 2   Lead Origin                                    9240 non-null   object
 3   Lead Source                                    9204 non-null   object
 4   Do Not Email                                   9240 non-null   object
 5   Do Not Call                                    9240 non-null   object
 6   Converted                                      9240 non-null   int64
 7   TotalVisits                                    9103 non-null   float64
 8   Total Time Spent on Website                    9240 non-null   int64
 9   Page Views Per Visit                           9103 non-null   float64
 10  Last Activity                                  9137 non-null   object
 11  Country                                        6779 non-null   object
 12  Specialization                                 7802 non-null   object
 13  How did you hear about X Education             7033 non-null   object
 14  What is your current occupation                6550 non-null   object
 15  What matters most to you in choosing a course  6531 non-null   object
 16  Search                                         9240 non-null   object
 17  Magazine                                       9240 non-null   object
 18  Newspaper Article                              9240 non-null   object
 19  X Education Forums                             9240 non-null   object
 20  Newspaper                                      9240 non-null   object
 21  Digital Advertisement                          9240 non-null   object
 22  Through Recommendations                        9240 non-null   object
 23  Receive More Updates About Our Courses         9240 non-null   object
 24  Tags                                           5887 non-null   object
 25  Lead Quality                                   4473 non-null   object
 26  Update me on Supply Chain Content              9240 non-null   object
 27  Get updates on DM Content                      9240 non-null   object
 28  Lead Profile                                   6531 non-null   object
 29  City                                           7820 non-null   object
 30  Asymmetrique Activity Index                    5022 non-null   object
 31  Asymmetrique Profile Index                     5022 non-null   object
 32  Asymmetrique Activity Score                    5022 non-null   float64
 33  Asymmetrique Profile Score                     5022 non-null   float64
 34  I agree to pay the amount through cheque       9240 non-null   object
 35  A free copy of Mastering The Interview         9240 non-null   object
 36  Last Notable Activity                          9240 non-null   object
dtypes: float64(4), int64(3), object(30)
memory usage: 2.6+ MB
```

Exploratory Data Analysis (1/2)

- Dropping the unwanted columns from the data
  - Several columns that were irrelevant to the data were dropped

There are several columns such as 'whether the customer had seen the ad in any of the listed items' and 'index and score assigned to each customer' that can be used for other purposes but not for our modelling purpose. Hence we drop these columns

```
drop_cols = ['Search', 'Magazine', 'Newspaper Article', 'X Education Forums', 'Newspaper', 'Digital Advertisement', 'Tags',
             'Lead Quality', 'Lead Profile', 'Asymmetrique Activity Index', 'Asymmetrique Profile Index', 'Asymmetrique Activity
            'Asymmetrique Profile Score']
leads.drop(labels = drop_cols, axis = 1, inplace = True)
```

- Identification and imputing the null Columns

- The missing values in some of the columns were imputed using below methodologies
  - Dropping the columns
  - Filling missing values with Mode
  - Filling missing values with Median

Comments - While country and city are important factors but have too many Null values and cannot be imputed, hence we drop these columns. The null values in 'Total Visits' and 'Page Views Per Visit' can be imputed with the average values and and 'Last Avtivity' and 'Specilization' can be imputed wiht the Mode of the respective columns. Other columns with high missing values will be dropped

```
drop_cols_2 = ['Country', 'City', 'How did you hear about X Education',
               'What is your current occupation', 'What matters most to you in choosing a course']
leads.drop(labels = drop_cols_2, axis = 1, inplace = True)
```

Exploratory Data Analysis (2/2)

- Some of the columns had 'Select' as a value and they were considered as Nulls assuming that the leads might have left them as blank from the dropdown options

There are 1942 values in Specialization as "Select" which probably are values that leads did not select from dropdown and also there are 1438 Null values making a total of 3380 Null values in "Specialziation", whihc is almost 30% of the values, hence it will be good to drop this column

```
leads.drop(labels = 'Specialization', axis = 1, inplace = True)
```

```
# Imputing the Page Views Per Visit
leads['Page Views Per Visit'] = leads['Page Views Per Visit'].fillna(leads['Page Views Per Visit'].mean())
```

```
# Imputing 'Last Activity' with Mode
leads['Last Activity'] = leads['Last Activity'].fillna(leads['Last Activity'].mode()[0])
```

Data Preparation

- Converting binary variables (yes/no) to 1/0
  - The binary columns with values as yes or no were converted to 1 and 0

Converting binary variables (yes/no) to 0/1

```
# List of variables to map

varlist =  ['Do Not Email', 'Do Not Call', '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', 'A free

# Defining the map function
def binary_map(x):
    return x.map({'Yes': 1, "No": 0})

# Applying the function to the housing list
leads[varlist] = leads[varlist].apply(binary_map)
```

- Mapping dummy variables to categorical variables
  - The categorical variables with multiple levels were mapped with the dummy variables using the get_dummies function in pandas

- Once done, the repeated columns were dropped from the data frame

As we notice, the binary values have been correctly mapped to 0/1. The categorical variables with multiple levels will be mapped with dummy variables and first one will be dropped

```
# Creating a dummy variable for some of the categorical variables and dropping the first one.
dummy1 = pd.get_dummies(leads[['Lead Origin', 'Lead Source', 'Last Activity', 'Last Notable Activity']], drop_first=True)

# Adding the results to the master dataframe
leads = pd.concat([leads, dummy1], axis=1)
```

Building the Model

- We started building the model with splitting the data in Test-Train split using test-train split function in scikit-learn library

Test-Train Split

```
# Putting feature variable to X
X = leads.drop(['Prospect ID','Converted'], axis=1)

X.head()
```

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	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	...	Last Notable Activity_Form Submitted on Website	Last Notable Activity_Had a Phone Conversation	Last Notable Activity_Modified	Last N Activity_ Conver
0	0	0	0.0	0	0.0	0	0	0	0	0	...	0	0	1	
1	0	0	5.0	674	2.5	0	0	0	0	0	...	0	0	0	
2	0	0	2.0	1532	2.0	0	0	0	0	0	...	0	0	0	
3	0	0	1.0	305	1.0	0	0	0	0	0	...	0	0	1	
4	0	0	2.0	1428	1.0	0	0	0	0	0	...	0	0	1	

5 rows × 66 columns

```
# Putting response variable to y
y = leads['Converted']

y.head()
```

```
0    0
1    0
2    1
3    0
4    1
Name: Converted, dtype: int64
```

```
# 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)
```

Feature Scaling

- Once the split was done, we scale the variables in train set using standard scaler

```
scaler = StandardScaler()

X_train[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']] = scaler.fit_transform(X_train[['TotalVisits','Tota

X_train.head()
```

- The conversion rate at this stage was found to be 38%

```
### Checking the Conversion Rate
leads_converted = (sum(leads['Converted'])/len(leads['Converted'].index))*100
print(leads_converted)
```

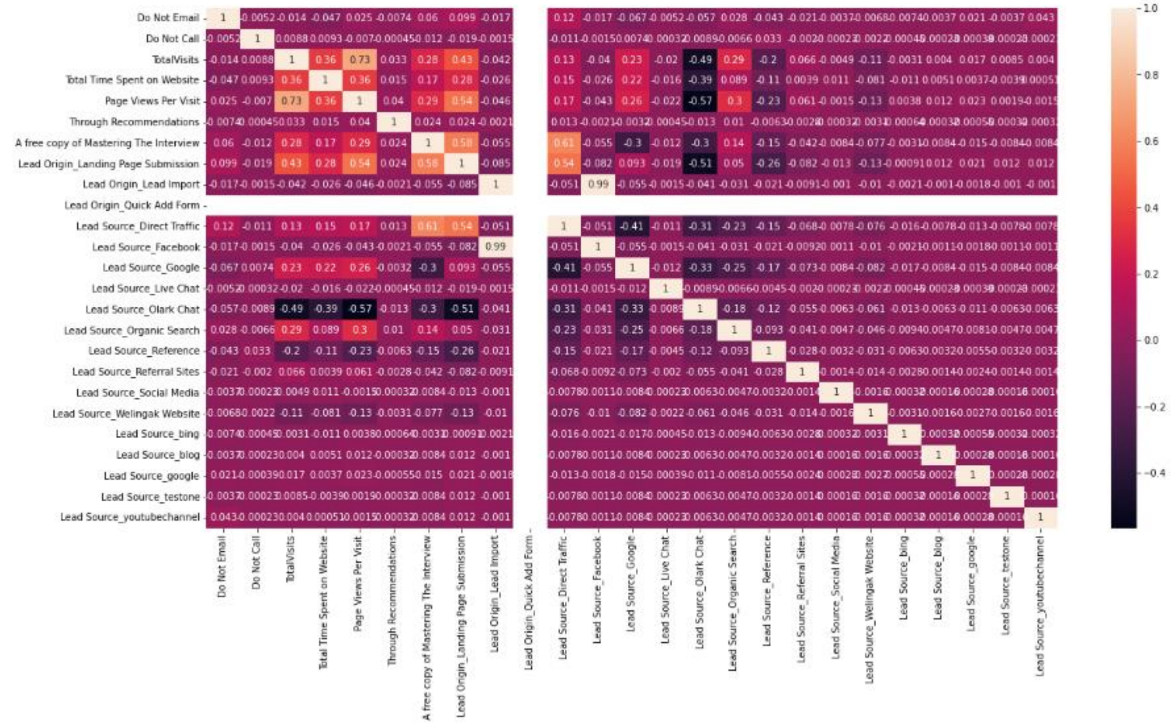
```
38.20765271872902
```

As we notice, the conversion rate is approximately 38%

Building the Correlation Metrix

- Several attempts were made to create the correlation Metrix and final one had the below mentioned output:

```
plt.figure(figsize = (20,10))
sns.heatmap(X_train.corr(),annot = True)
plt.show()
```



Further, the column ‘Lead_Origin_Quick_Add form was dropped as all the values in the column were zero

Model Building (1/3)

Once the data was cleaned and brought into right format, model building process was started:

- There were quite a few variables with high p-values hence, RFE method was used to initially eliminate few variables:

There are quite a few values with higher P-values and we will take the help of Recursive Feature Elimination (RFE) to select the features

```
logreg = LogisticRegression()
```

```
rfe = RFE(logreg,n_features_to_select = 15)             # running RFE with 13 variables as output
rfe = rfe.fit(X_train, y_train)
```

- The remaining variables were assessed and 2 more variables were removed using VIF method
- A column called Lead Score was added to give a score of 1-100 to all the leads
- The accuracy of the model was tested at this stage

```
# Checking the overall accuracy of the model
print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.predicted))
```

```
0.7800511508951407
```

As we notice, the prediction probabilty of the model is 78%. We will not check the VIFs to further improve the accuracy of the model

Model Building (2/3)

- Removing features using VIF method:

```
# 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 Import	43.00
5	Lead Source_Facebook	43.00
1	Total Time Spent on Website	1.25
7	Lead Source_Olark Chat	1.15
0	Do Not Email	1.11
4	Lead Source_Direct Traffic	1.09
6	Lead Source_Google	1.06
8	Lead Source_Organic Search	1.03
9	Lead Source_Reference	1.01
11	Lead Source_Welingak Website	1.01
2	Through Recommendations	1.00
10	Lead Source_Referral Sites	1.00
12	Lead Source_bing	1.00
13	Lead Source_blog	1.00
14	Lead Source_google	1.00

The featues with high VIF can be eliminated. Hence we will remove "Lead Origin_Lead Import" and Lead Source_Facebook" from the model

- The Lead Origin_Lead Import and Lead_Origin Source_Facebook were removed from the model since the VIF for these two features was very high

Model Building (3/3)

- Rebuilding the Model

```
y_train_pred = res.predict(X_train_sm).values.reshape(-1)
```

```
y_train_pred[:10]
```

```
array([0.98978906, 0.48062338, 0.83268365, 0.16782147, 0.46331452,
       0.16839177, 0.2010038 , 0.80191374, 0.73678744, 0.25427899])
```

```
y_train_pred_final['Convert_Prob'] = y_train_pred
```

```
# Creating new column 'predicted' with 1 if Churn_Prob > 0.5 else 0
y_train_pred_final['predicted'] = y_train_pred_final.Convert_Prob.map(lambda x: 1 if x > 0.5 else 0)
y_train_pred_final.head()
```

	Converted	Convert_Prob	Prospect_ID	Lead_Score	predicted
0	1	0.989789	1046	99.0	1
1	0	0.480623	8738	48.0	0
2	1	0.832684	7818	83.0	1
3	0	0.167821	498	17.0	0
4	0	0.463315	2169	47.0	0

```
# Checking the overall accuracy.
print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.predicted))
```

```
0.7774936061381074
```

- It was noticed that the accuracy didn’t change much and hence the model was accepted with these features

The Confusion Matrix

Actual ⬇ Predicted ➡	Not Converted	Converted
Not Converted	3422	465
Converted	912	1442

True Positives = 1442
True Negatives = 3442
False Positives = 465
False Negatives = 912

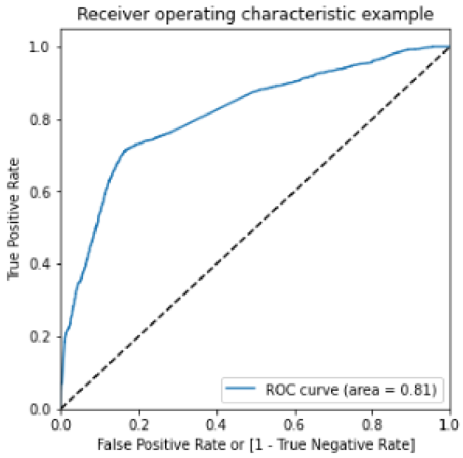
As we notice, our model has rightly called 4864 (3422+1442)) values out of 6241 values.
Whihc is an accuracy of approximately 78%

Plotting the ROC Curve

An ROC curve demonstrates several things:

- It 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.

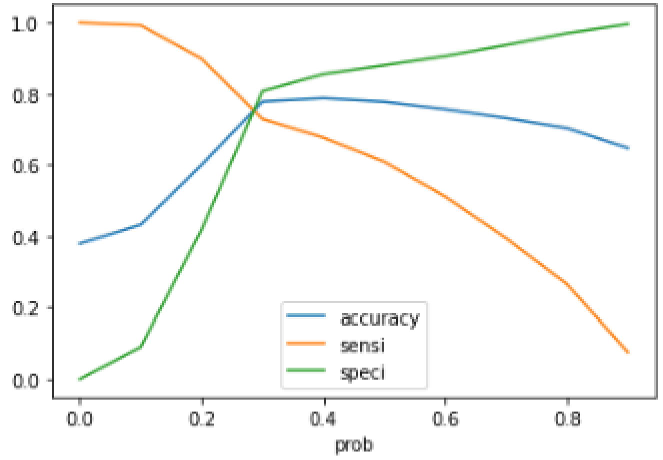
```
draw_roc(y_train_pred_final.Converted, y_train_pred_final.Convert_Prob)
```



Finding the Optimal Cutoff Point

We plot the Sensitivity, Specificity and Accuracy to find the optimal cut-off point

```
# Plotting the accuracy sensitivity and specificity for various probabilities.
cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'])
plt.show()
```



We notice that the optimal cut-off point is 0.25

Making Predictions on Test Set

- The model was tested on the test set and an accuracy of 78% was found

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

```
0.7803877703206562
```

- The confusion Matrix

Actual Predicted	Not Converted	Converted
Not Converted	1399	237
Converted	352	694

True Positives = 1399
True Negatives = 694
False Positives = 352
False Negatives = 237

Thank You!