LEAD SCORE CASE STUDY

ASHUTOSH NAYAK

MAHESH PRASAD MISHRA

PROBLEM STATEMENT:

An education company named X Education sells online courses to industry professionals.
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%.

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.

OBJECTIVES

- Build 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.

- There are some more problems presented by the company which your model should be able to adjust to if the company's requirement changes in the future so you will need to handle these as well. These problems are provided in a separate doc file. Please fill it based on the logistic regression model you got in the first step. Also, make sure you include this in your final PPT where you'll make recommendations.

PROBLEM SOLVING METHODOLOGY :

The approach for this project has been to divide the entire case study into various checkpoints to meet each of the sub-goals.

Step 1: Reading and Understanding the Data
Step 2: Data Cleansing

Step 3: Applying Recursive feature elimination to identify the best performing subset of features for building the model.

Step 5: Building the model with features selected by RFE. Eliminate all features with high p-values and VIF values and finalize the model

Step 6: Perform model evaluation with various metrics like sensitivity, specificity, precision, recall, etc.

Step 7: Decide on the probability threshold value based on Optimal cutoff point and predict the dependent variable for the training data.
Step 8: Use the model for prediction on the test dataset and perform model evaluation for the test set.

UNDERSTANDING THE DATA AND DATA CLEANING

Remove columns which has only one unique value

Deleting the following columns as they have only one unique value and hence cannot be responsible 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' etc….

Removing rows where a particular column has 45% missing values

Deleted the columns those have missing values more thatn 45%. Such columns are `'How did you hear about X Education', 'Lead Profile', 'Lead Quality' etc…..`

Imputing NULL values with Median

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

UNDERSTANDING THE DATA AND DATA CLEANING

Handling 'Select' values in some columns

There are some columns in dataset which have a level/value called 'Select'. This might have happened because these fields in the website might be non mandatory fields with drop downs options for the customer to choose from. The Select values in columns were converted to Nulls.

Assigning a Unique Category to NULL/SELECT values

All the nulls in the columns were binned into a separate column ‘Unknown’. • Instead of deleting columns with huge null value percentage(which results in loss of data), this strategy adds more information into the dataset and results in the change of variance. •

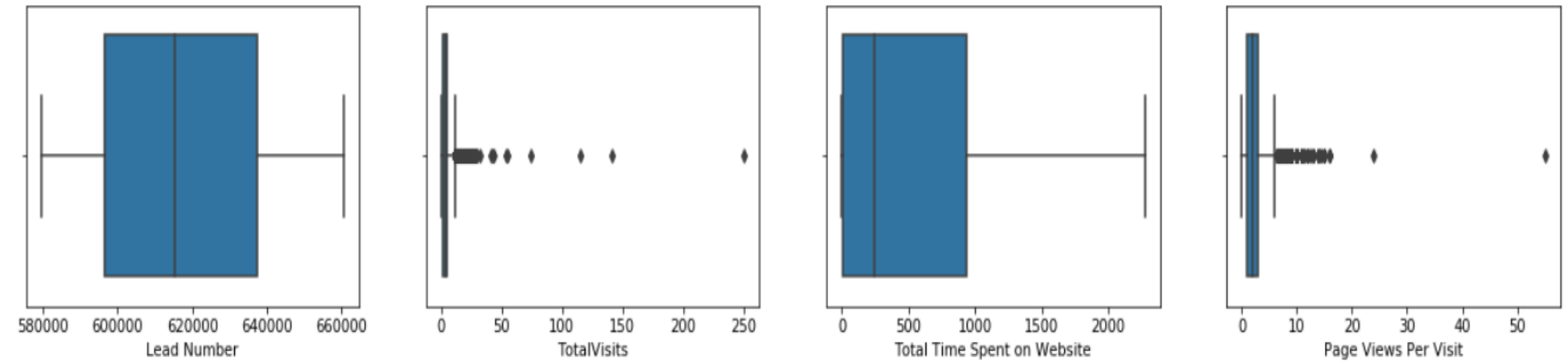
Outlier Treatment

The outliers present in the columns 'TotalVisits‘ & 'Page Views Per Visit‘ were finally removed based on iterquatile range analysis.

OUTLIERS TREATMENT

The outliers present in the columns 'TotalVisits‘ & 'Page Views Per Visit‘ were finally removed based on iterquatile range analysis.

```
plot_Outlier(['Lead Number','TotalVisits','Total Time Spent on Website','Page Views Per Visit'])
```



DUMMY ENCODING

For the following categorical variables with multiple levels, dummy features (one-hot encoded) were created:

- 'Lead Origin',
- 'Lead Source',
- 'Do Not Email',
- 'Last Activity',
- 'Country',
- 'Specialization',
- 'What is your current occupation',
- 'What matters most to you in choosing a course',
- 'Tags',
- 'City',
- 'Last Notable Activity'

Creation of Dummy variables for categorical variables

```
]: colmn=[]
for i in lead_data.describe(include='object').columns:
    # Creating a dummy variable
    colmn.append(i)
    cont = pd.get_dummies(lead_data[i],prefix=i,drop_first=True)
    #Adding the results to the master dataframe
    lead_data = pd.concat([lead_data,cont],axis=1)
```

```
]: colmn
```

```
]: ['Lead Origin',
 'Lead Source',
 'Do Not Email',
 'Last Activity',
 'Country',
 'Specialization',
 'What is your current occupation',
 'What matters most to you in choosing a course',
 'Tags',
 'City',
 'Last Notable Activity']
```

```
]: #created dummies for the below variables, so drop the same
lead_data = lead_data.drop(columns = colmn,axis=1)
```

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.

Step-3 rain-Test Split & Logistic Regression Model Building:

```
# Putting feature variable to X
X = lead_data.drop(['Converted','Lead Number'], axis=1)
# Putting response variable to y
y = lead_data['Converted']
# 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

Scaling helps in interpretation. It is important to have all variables(specially categorical ones which has values 0 and 1 on the same scale for the model to be easily interpretable.

- 'Standardisation' was used to scale the data for modelling. It basically brings all of the data into a standard normal distribution with mean at zero and standard deviation one.

Step 4: Feature Scaling

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

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

X_train.head()
```

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	A free copy of Mastering The Interview	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Origin_Quick Add Form	Lead Source_Direct Traffic	Source
5182	-0.74	-0.77	-0.65	0	0	0	0	0	0	
8469	-0.33	-0.53	-0.05	1	1	0	0	0	1	
8382	0.08	1.69	-0.35	0	0	0	0	0	0	
8031	0.08	-0.08	0.56	0	1	0	0	0	1	
6712	0.08	2.94	-0.05	0	0	1	0	0	0	

5 rows × 120 columns

FEATURE SELECTION USING RFE

- Recursive feature elimination is an optimization technique for finding the best performing subset of features.
- It is based on the idea of repeatedly constructing a model and choosing either the best (based on coefficients), setting the feature aside and then repeating the process with the rest of the features.
- This process is applied until all the features in the dataset are exhausted. Features are then ranked according to when they were eliminated.

- Running RFE with the output number of the variable equal to 15

Step 5: Feature Selection Using RFE

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

```
from sklearn.feature_selection import RFE
rfe = RFE(logreg, 15)             # running RFE with 15 variables as output
rfe = rfe.fit(X_train, y_train)
```

```
rfe.support_
```

```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False,  True, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False,  True, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
        True,  True,  True, False, False, False, False,  True, False,
        True,  True, False,  True, False, False,  True, False, False,
       False, False,  True, False,  True, False, False,  True, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False,  True,  True, False, False, False,
       False, False, False])
```

BUILDING THE MODEL

- Generalized Linear Models from StatsModels is used to build the Logistic Regression model.
- The model is built initially with the 15 variables selected by RFE.
- Unwanted features are dropped serially after checking p values (< 0.5) and VIF (< 5) and model is built multiple times.
- The final model with 11 features, passes both the significance test and the multi-collinearity test.

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6075
Model:	GLM	**Df Residuals:**	6063
Model Family:	Binomial	**Df Model:**	11
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1334.7
Date:	Sun, 19 Apr 2020	**Deviance:**	2669.4
Time:	21:30:17	**Pearson chi2:**	1.39e+04
No. Iterations:	8		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.3452	0.090	-14.878	0.000	-1.522	-1.168
Lead Source_Welingak Website	5.8679	1.033	5.678	0.000	3.842	7.893
Last Activity_SMS Sent	2.2380	0.113	19.861	0.000	2.017	2.459
What matters most to you in choosing a course_Other	-0.7261	0.116	-6.249	0.000	-0.954	-0.498
Tags_Busy	0.5888	0.243	2.418	0.016	0.112	1.066
Tags_Closed by Horizzon	7.5815	0.723	10.482	0.000	6.164	8.999
Tags_Lost to EINS	6.6521	0.603	11.036	0.000	5.471	7.833
Tags_Ringing	-3.3505	0.239	-14.009	0.000	-3.819	-2.882
Tags_Will revert after reading the email	4.9927	0.192	25.989	0.000	4.616	5.369
Tags_switched off	-2.9732	0.699	-6.455	0.000	-5.343	-2.607

PREDICTING CONVERSION PROBABILITY

Creating a dataframe with the actual Converted flag and the predicted probabilities. Showing top 5 records of the dataframe in the picture on the right.

	Conversion	Conversion_Prob	LeadID
0	0	0.03	5182
1	0	0.03	8469
2	0	0.01	8382
3	0	0.21	8031
4	1	0.99	6712

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

	Conversion	Conversion_Prob	LeadID	predicted
0	0	0.03	5182	0
1	0	0.03	8469	0
2	0	0.01	8382	0
3	0	0.21	8031	0
4	1	0.99	6712	1

FINDING OPTIMAL PROBABILITY THRESHOLD

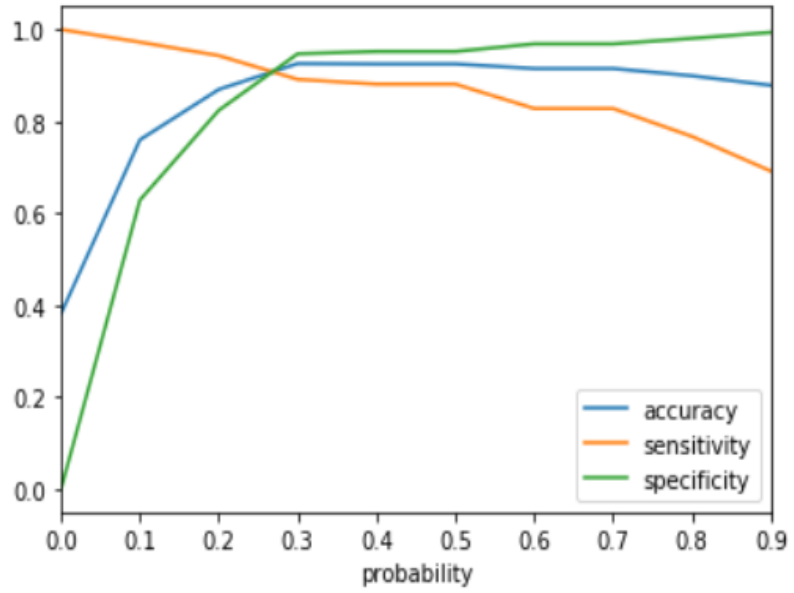
Optimal cutoff probability is that prob where we get balanced sensitivity and specificity.

• The accuracy sensitivity and specificity was calculated for various values of probability threshold and plotted in the graph to the right.

• From the curve above, 0.28 is found to be the optimum point for cutoff probability.

• At this threshold value, all the 3 metrics accuracy sensitivity and specificity was found to be well above 80% which is a well acceptable value

```
# Let's plot accuracy sensitivity and specificity for various probabilities.
cutoff_df.plot.line(x='probability', y=['accuracy','sensitivity','specificity'])
plt.show()
```



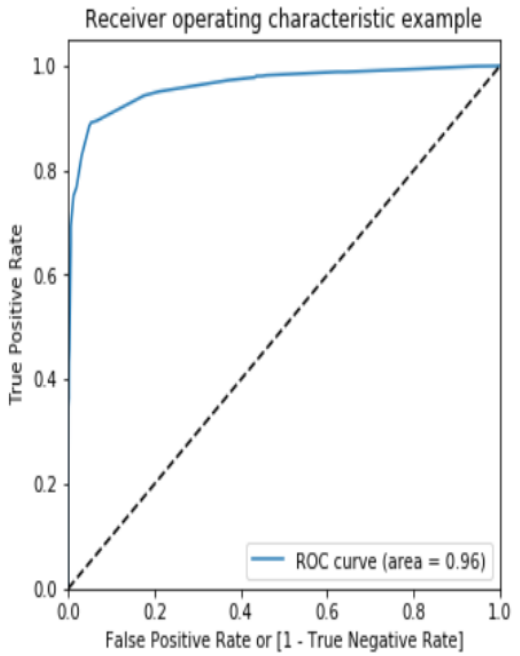
From the curve above, 0.28 is the optimum point to take it as a cutoff probability

PLOTING ROC CURVE AND CALCULATING AUC

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

- 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 0.9678

```
draw_roc(y_train_pred_final.Conversion, y_train_pred_final.Conversion_Prob)
```



MAKING PREDICTIONS ON TEST DATA

•The final model on the train dataset is used to make predictions for the test dataset

•The train 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 0.28, 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.

	Converted	LeadID	Conversion_Prob
0	1	475	1.00
1	1	2461	1.00
2	0	1890	0.11
3	0	6007	0.21
4	0	4052	0.08

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

```
[[1490  101]
 [ 133  880]]
```

```
CalculateMetrics(confusion_test)
```

```
Sesitivity For the Model :  0.8687068114511353
specificity For the Model :  0.9365179132620993
false postive rate For the Model :  0.06348208673790069
precision/false postive rate For the Model :  0.8970438328236493
Negative predictive value For the Model :  0.9180529882932841
```

```
p, r, thresholds = precision_recall_curve(y_pred_final.Converted, y_pred_final
```

DETERMINING FEATURE IMPORTANCE

- 11 features have been used by our model to successfully predict if a lead will get converted or not.

- The Coefficient (beta) values for each of these features from the model parameters are used to determine the order of importance of these features.
- Features with high positive beta values are the ones that
contribute most towards the probability of a lead getting converted.
- Similarly, features with high negative beta values contribute the least.

```
new_params.sort_values(ascending=False)
```

Tags_Closed by Horizzon	7.58
Tags_Lost to EINS	6.65
Lead Source_Welingak Website	5.87
Tags_Will revert after reading the email	4.99
Last Activity_SMS Sent	2.24
Tags_Busy	0.59
What matters most to you in choosing a course_Other	-0.73
Last Notable Activity_Olark Chat Conversation	-1.65
Last Notable Activity_Modified	-2.01
Tags_Ringing	-3.35
Tags_switched off	-3.87

dtype: float64

INFERENCE

After trying several models, we finally chose a model with the following characteristics:

- All variables have p-value < 0.05.
- All the features have very low VIF values, meaning, there is hardly any multicollinearity among the features. This is also evident from the heat map.
- The overall accuracy of 0.91 at a probability threshold of 0.28 on the test dataset is also very acceptable.
- Using this model, the dependent variable value was predicted as per the following threshold values of Conversion probability:
- Final Observation:
- Threshold=0.28
- **Train Data:**
- Accuracy : 92.5%
- Sensitivity : 89.1%
- Specificity : 94.7%
- **Test Data:**
- Accuracy : 91.01%
- Sensitivity : 86.8%
- Specificity : 93.6%

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

Features with Positive Coefficient Values
Tags_Busy 0.59
Last Activity_SMS Sent 2.24
Tags_Will revert after reading the email 4.99
Lead Source_Welingak Website 5.87
Tags_Lost to EINS 6.65
Tags_Closed by Horizzon 7.58

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

Features with Negative Coefficient Values
Tags_switched off -3.87
Tags_Ringing -3.35
Last Notable Activity_Modified -2.01
Last Notable Activity_Olark Chat Conversation -1.65
What matters most to you in choosing a course_Other -0.73