CASE STUDY

LEAD SCORE DATA

GROUP MEMBERS:
ASHUTOSH KAUSHAL
ALAKESH KALITA
DWAIPAYAN SARKAR

PROBLEM STATEMENT

- X Education sells online courses to industry professionals.
- X Education gets a lot of leads, its lead conversion rate is very poor. For example, if,
- say, they acquire 100 leads in a day, only about 30 of them are converted.
- To make this process more efficient, the company wishes to identify the most potential leads, also known as 'Hot Leads'.
- If they successfully identify this set of leads, the lead conversion rate should go up as the sales team will now be focusing more on communicating with the potential leads rather than making calls to everyone.

Business Objective:

- X education wants to know most promising leads.
- For that they want to build a Model which identifies the hot leads.
- Deployment of the model for the future use.

STEPS TO BE FOLLOWED

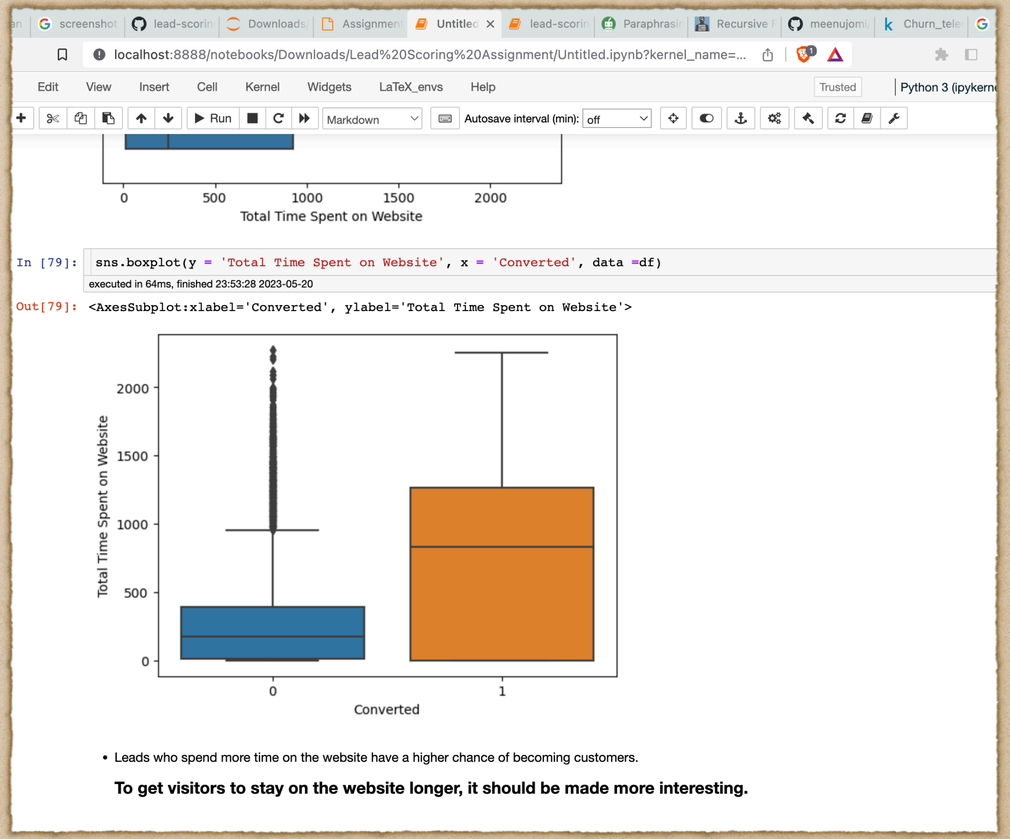
- DATA CLEANING
- EDA
- CREATING DUMMY VARIABLES
- BUILDING THE MODEL
- VALIDATING IT
- DEPLOYING IT
- CONCLUSIONS AND RESULTS

DATA CLEANING

- CHECKING NUMBER OF ROWS AND COLUMNS IN THE DATA SET.
- CHECKING FOR NULL VALUES.
- CHECKING FOR DEFAULT VALUES
- REPLACING DEFAULT VALUES WITH NAN
- DROPPING NULL VALUES

EDA

- VISUALIZATION WAS PERFORMED WITH EVERY ATTRIBUTES OF THE DATE SET.
- MOST OF THEM WERE NOT FOUND USEFUL FOR ANALYSIS
- ATTRIBUTES THAT WERE FOUND USEFUL WERE 'LEAD SOURCE',' LAST ACTIVITY','LEAD ORIGIN','TOTAL TIME SPENT ON WEBSITE'



ADDING DUMMY VARIABLES

- DUMMY VARIABLES WERE ADDED TO THESE COLUMNS OF THE DATA SET

- dummy_var= pd.get_dummies(df[['Lead Origin', 'Lead Source', 'Last Activity', 'Specialization','What is your current occupation', 'City','Last Notable Activity']], drop_first=True)

executed in 13ms, finished 23:53:28 2023-05-20

	Prospect ID	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	...	Last Notable Activity_Form Submitted on Website	Last Notable Activity_Had a Phone Conversation	Ac
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	API	Olark Chat	0	0	0	0.0	0	0.0	Page Visited on Website	...	0	0	
1	2a272436-5132-4136-86fa-dcc88c88f482	API	Organic Search	0	0	0	5.0	674	2.5	Email Opened	...	0	0	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	Landing Page Submission	Direct Traffic	0	0	1	2.0	1532	2.0	Email Opened	...	0	0	
3	0cc2df48-9de9-19797f9b38cc	Landing Page Submission	Direct Traffic	0	0	0	1.0	305	1.0	Unreachable	...	0	0	
4	3256f628-e534-4826-9d63-4a8b88782852	Landing Page Submission	Google	0	0	1	2.0	1428	1.0	Converted to Lead	...	0	0	

to scroll output; double click to hide

5 rows × 78 columns

```
# droping varibles which dummies created
df = df.drop(['Lead Origin', 'Lead Source', 'Last Activity', 'Specialization','What is your current occupation',
              'City','Last Notable Activity'], axis = 1)
```

executed in 4ms, finished 23:53:28 2023-05-20

```
df
```

MODEL BUILDING

- FOR CREATION FOR MODEL STEPS WERE FOLLOWED BY REMOVING THE ATTRIBUTES WITH HIGH VALUES TO ALMOST ZERO
- WITH VIF RANGING BELOW 3
- TOTAL 14 MODELS WERE MADE AND 14TH MODEL WAS CONSIDERED AS THE FINAL MODEL WITH 0 P-VALUE AND VIF VALUE BELOW 3

In [135]:
```
# Checking Vif again
vif = pd.DataFrame()
vif['Features'] = X_train[col1].columns
vif['VIF'] = [variance_inflation_factor(X_train[col1].values, i) for i in range(X_train[col1].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

executed in 50ms, finished 23:53:29 2023-05-20

Out[135]:

	Features	VIF
10	Specialization_Others	2.16
3	Lead Source_Olark Chat	2.03
12	Last Notable Activity_Modified	1.78
2	Lead Origin_Landing Page Submission	1.70
6	Last Activity_Olark Chat Conversation	1.59
8	Last Activity_SMS Sent	1.57
1	Total Time Spent on Website	1.29
4	Lead Source_Reference	1.24
0	Do Not Email	1.21
11	What is your current occupation_Working Profes...	1.19
5	Lead Source_Welingak Website	1.09
9	Last Activity_Unsubscribed	1.08
7	Last Activity_Other_Activity	1.01

It can be observed here the p values for columns are now 0 and VIF values are not too high hence model 14 can be considered as the final model for evaluation

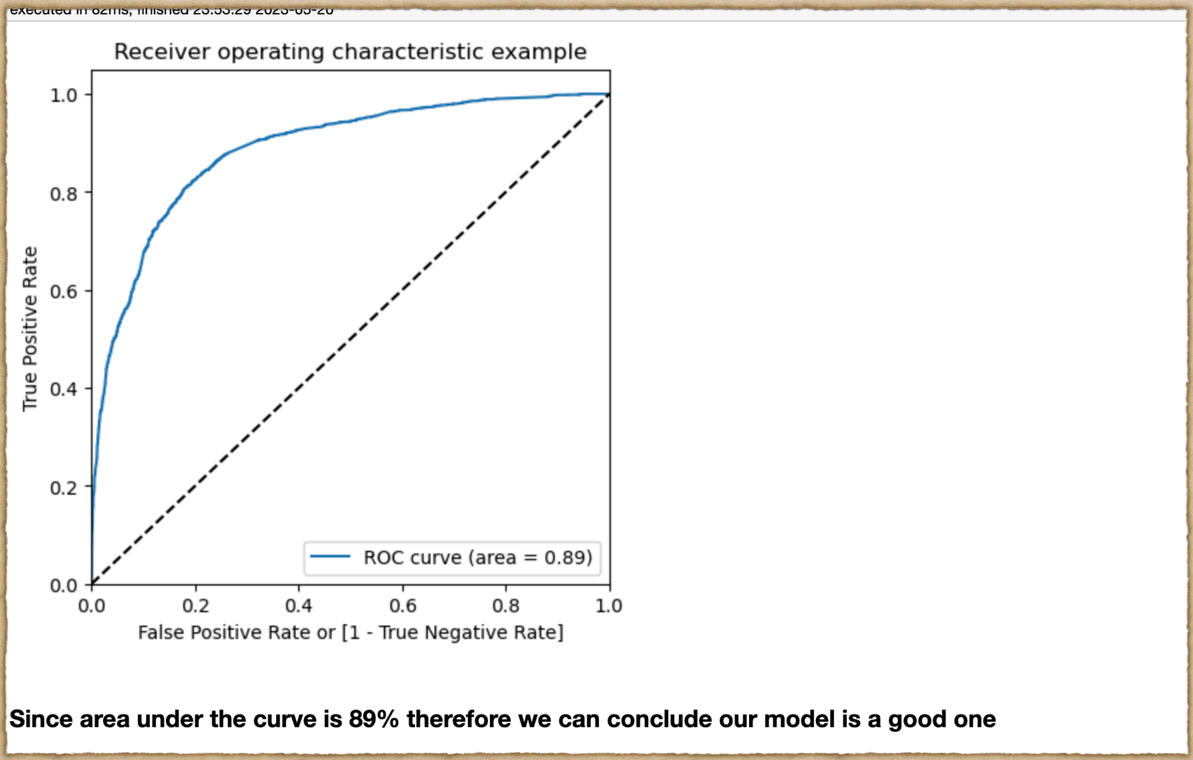
TRAINING THE MODEL

- THE WERE THE OUTPUTS AFTER TRAINING THE MODEL

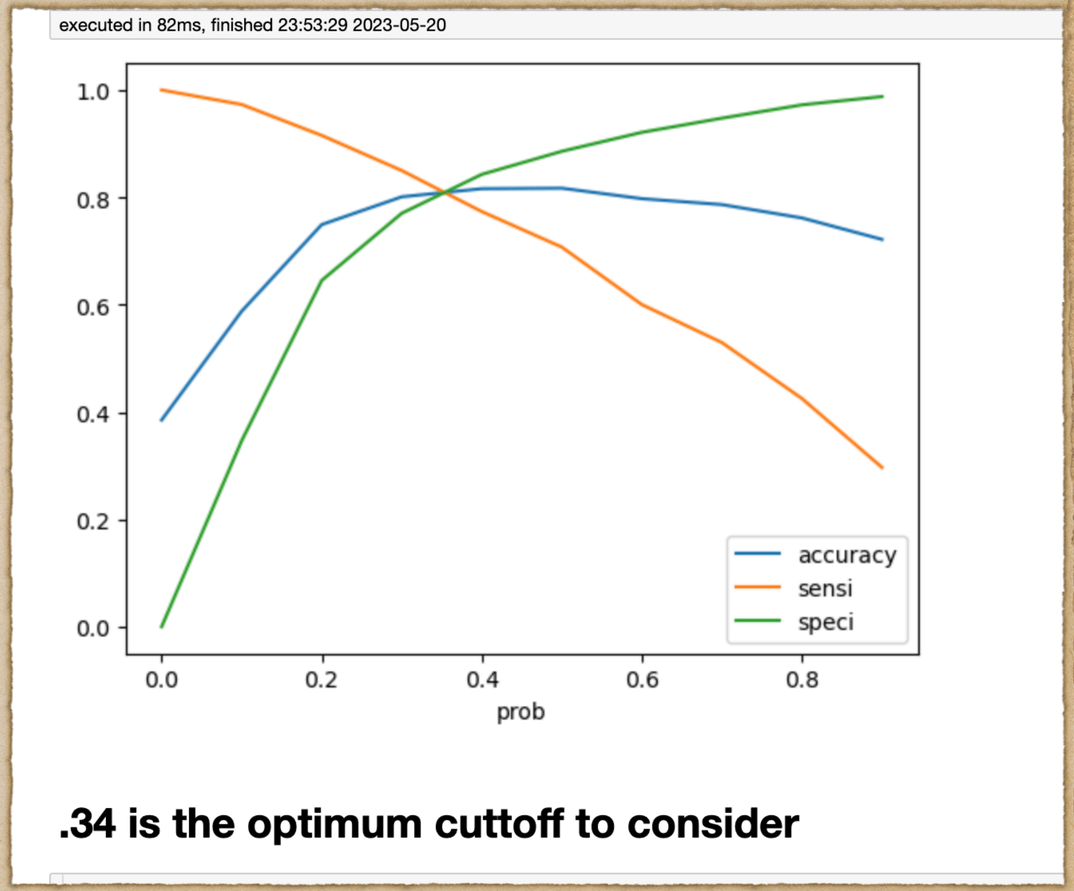
- Specificity was determined to be good (88.5%) but our sensitivity was just 70.7%. As a result, this needs to be resolved.

-Sensitivity was 70%, mostly as a result of the arbitrary cut-off point of 0.5 that we used. The ROC curve will be used to optimise this cut-off point in order to obtain a respectable value for sensiti

ROC CURVE AND OPTIMUM CUTTOFF



Since area under the curve is 89% therefore we can conclude our model is a good one



.34 is the optimum cuttoff to consider

MODEL EVALUATION

Train Data:

- Accuracy : 81.1 %
- Sensitivity : 81.8 %
- Specificity : 80.6 %

Train Data:

- Accuracy : 80.4 %
- Sensitivity : 80.4 %
- Specificity : 80.3 %

```
Out[169]: array([[1394,  340],
                 [ 193,  796]])
```

```
In [170]: # Lets check TP TN FP FN
          TP = confusion[1,1] # true positive
          TN = confusion[0,0] # true negatives
          FP = confusion[0,1] # false positives
          FN = confusion[1,0] # false negatives
```

executed in 3ms, finished 23:53:30 2023-05-20

```
In [171]: print("Sensitivity : ",TP / float(TP+FN)*100)
          print("Specificity : ",TN / float(TN+FP)*100)
          print("False Positive Rate :",FP/ float(TN+FP)*100)
          print("Positive Predictive Value :",TP / float(TP+FP)*100)
          print ("Negative predictive value :",TN / float(TN+ FN)*100)
```

executed in 3ms, finished 23:53:30 2023-05-20

```
Sensitivity :  80.48533872598584
Specificity :  80.3921568627451
False Positive Rate : 19.607843137254903
Positive Predictive Value : 70.07042253521126
Negative predictive value : 87.83868935097668
```

```
In [172]: # Assinging the score to final test data
          y_predict['Lead_Score'] = y_predict.Converted_probality.map( lambda x: round(x*100))
          y_predict.head()
```

executed in 7ms, finished 23:53:30 2023-05-20

Out[172]:

	ID	Converted	Converted_probality	final_predicted	Lead_Score
0	3271	0	0.129393	0	13
1	1490	1	0.968762	1	97
2	7936	0	0.111751	0	11
3	4216	1	0.804505	1	80
4	3830	0	0.132210	0	13

CONCLUSION

Thus, we have succeeded in achieving our objective of estimating the desired lead conversion rate to be somewhere around 80%. We should be able to provide the CEO confidence to make wise decisions based on this model in order to achieve a higher lead conversion rate of 80% because the model appears to anticipate the conversion rate quite effectively.