Presentation of Leads Case Study

Furthermore, I have described the process that I have followed to do my Leads case study.

Problem Statement

- X Education is a company that markets courses online through websites and search engines. It's lead conversion rate is quite poor.
- In this case study we try to create a logistic regression machine learning model from the data provided by the education company of customers who have been contacted for courses in order to understand what data may play an important role in increasing the lead conversion rate so that the company can focus more on communicating the potential leads that may lean towards choosing a course.

First step upload the required libraries

- In this step, I have uploaded the required libraries like a panda, seaborn, matplotlib, sklearn, statsmodel, etc.
- This you can see in the diagram, I have attached to the left.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn import metrics
from sklearn.metrics import precision_score, recall_score
```

Import the CSV data to the notebook

- In this step, I have used the panda library to import the CSV data to the Jupiter notebook by using the read_csv function.
- After that I have used leads.head() to see the first few lines of the data that I have been imported.

```
In [2]: leads = pd.read_csv("Downloads/Leads.csv")

In [3]: leads.head()

Out[3]:
```

	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	...	Get updates on DM Content	Lead Profile	City	Asymmetrique Activity Index	Asymm Profile
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No	Select	Select	02.Medium	02.M
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No	Select	Select	02.Medium	02.M
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	...	No	Potential Lead	Mumbai	02.Medium	(

Perform some operations

- Here I performed some operations to see the data in-depth like inp0.shape(to see the number of rows and columns), inp0.dtypes(to see the data types of the different columns)
- Inp0.info() to see the information of the columns.

```
In [4]:  leads.shape
Out[4]:  (9240, 37)

In [5]:  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
```

Apply Data Cleaning on the lead data set

- The very First step is to see how many null values the columns have and calculate their percentage by using is_null().sum().
- Second, drop those columns which having null values greater than 45% by using the drop function.

```
In [12]:  # checking the null values in terms of percentage
          round(100*(leads.isnull().sum()/len(leads.index)),2)

Out[12]:  Prospect ID                                        0.00
          Lead Number                                        0.00
          Lead Origin                                        0.00
          Lead Source                                        0.39
          Do Not Email                                       0.00
          Do Not Call                                        0.00
          Converted                                          0.00
          TotalVisits                                        1.48
          Total Time Spent on Website                        0.00
          Page Views Per Visit                               1.48
          Last Activity                                      1.11
          Country                                           26.63
          Specialization                                    36.58
          How did you hear about X Education                78.46
          What is your current occupation                   29.11
          What matters most to you in choosing a course     29.32
          Search                                             0.00
          Magazine                                           0.00
          Newspaper Article                                  0.00
          X Education Forums                                 0.00
          Newspaper                                          0.00
```

Data Cleaning

- Next, the remaining columns which are having few null values. Replacing the null values with the most frequent value.
- There are some columns which have unbalanced data. So I removed those columns such as Do Not Call, Search, Magazine, etc.

```
In [68]: leads['City'].value_counts(dropna=False)

Out[68]: NaN                            3669
         Mumbai                         3222
         Thane & Outskirts               752
         Other Cities                    686
         Other Cities of Maharashtra     457
         Other Metro Cities              380
         Tier II Cities                   74
         Name: City, dtype: int64
```

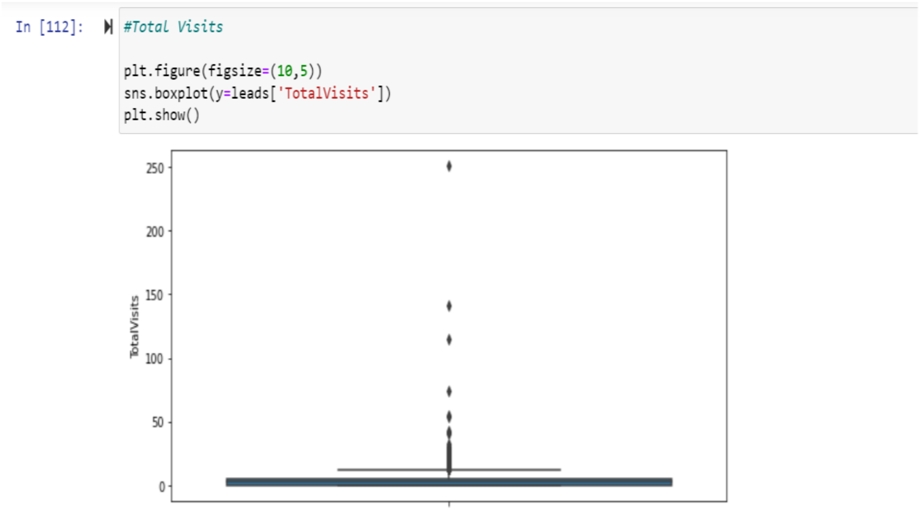
```
In [69]: # replacing Nan values with most occur value i.e mumbai
         leads['City'] = leads['City'].replace(np.nan,'Mumbai')
```

```
In [70]: leads['City'].value_counts(dropna=False)

Out[70]: Mumbai                         6891
         Thane & Outskirts               752
         Other Cities                    686
         Other Cities of Maharashtra     457
         Other Metro Cities              380
         Tier II Cities                   74
         Name: City, dtype: int64
```

Data Cleaning

- As a result of the previous operation, I don’t have any null values in my dataset and the data is balanced.
- Next, I handle the outliers for the numeric data such as TotalVisits, Total Time Spent on Website and Page Views Per Visit.



Creating Dummy variables

- Creating the dummy variables for the categorical columns which are having more than two categories.
- After creating the dummy variables, I left with 103 columns and the data is ready for the machine learning algorithm.

```
In [133]: # getting dummies and adding the result to the leads
          dummy = pd.get_dummies(leads[['Lead Origin', 'Lead Source', 'Last Activity',
                 'Specialization', 'What is your current occupation', 'Tags', 'City',
                  'Last Notable Activity']], drop_first=True)

          leads = pd.concat([leads,dummy],1)

In [134]: leads.head()
```

Splitting the data in to train and test

- Next step, is to split data into train and test by using the train-test library.
- Scaling the numeric variable by using the StandardScaler method.

In [139]:
```
# Splitting the data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=100)
```

In [140]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6468 entries, 1871 to 5640
Columns: 102 entries, TotalVisits to Last Notable Activity_View in browser link Clicked
dtypes: float64(2), int64(1), uint8(99)
memory usage: 827.4 KB
```

In [141]:
```
#scaling numeric columns

scaler = StandardScaler()

numeric_cols = X_train.select_dtypes(include=['float64', 'int64']).columns

X_train[numeric_cols] = scaler.fit_transform(X_train[numeric_cols])

X_train.head()
```

Creating the models

- After scaling the data, we create models using RFE until we get the p-values of all the columns less than 0.05.
- We also check the VIF if it is less than 3 or not.
- We eliminate the columns which are having high p-values and VIF one by one.
- When we find the model suitable for the analysis we continue with the process and do the y prediction.
- We predict the value across the target variable and find its accuracy, precision, recall and specificity.

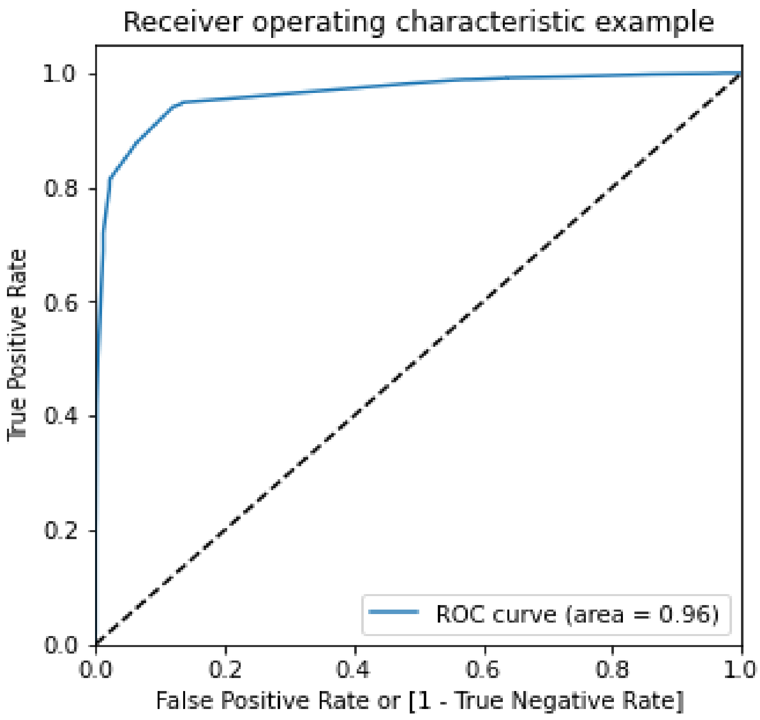
Dep. Variable:	Converted	No. Observations:	6468
Model:	GLM	Df Residuals:	6452
Model Family:	Binomial	Df Model:	15
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-1396.4
Date:	Sun, 16 Oct 2022	Deviance:	2792.7
Time:	17:44:00	Pearson chi2:	1.06e+04
No. Iterations:	8		
Covariance Type:	nonrobust		

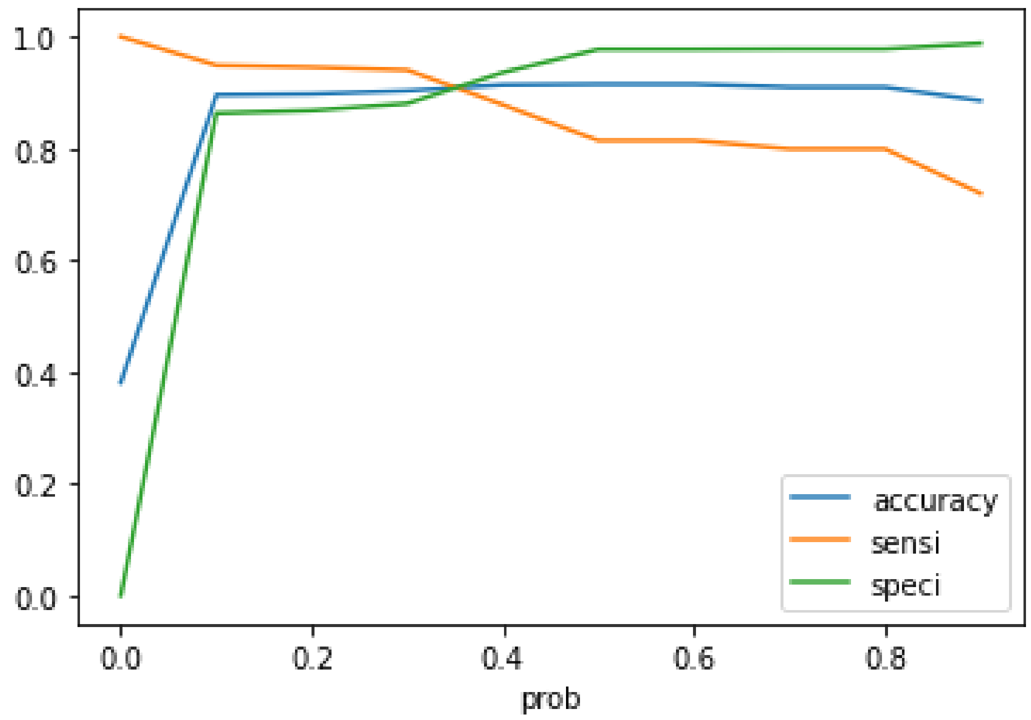
	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.5631	0.088	-29.114	0.000	-2.736	-2.391
What is your current occupation_Unemployed	2.2634	0.117	19.370	0.000	2.034	2.492
What is your current occupation_Working Professional	2.5416	0.356	7.149	0.000	1.845	3.238
Lead Source_Welingak Website	2.9473	0.733	4.021	0.000	1.511	4.384
Last Activity_SMS Sent	2.0669	0.109	18.950	0.000	1.853	2.281
Tags_Already a student	-4.8662	0.718	-6.774	0.000	-6.274	-3.458
Tags_Closed by Horizzon	5.8904	1.010	5.829	0.000	3.910	7.871
Tags_Graduation in progress	-2.5781	0.493	-5.228	0.000	-3.545	-1.612
Tags_Interested in full time MBA	-3.8136	0.736	-5.184	0.000	-5.255	-2.372
Tags_Interested in other courses	-3.4071	0.325	-10.479	0.000	-4.044	-2.770
Tags_Lost to EINS	5.5233	0.727	7.601	0.000	4.099	6.948
Tags_Not doing further education	-4.6677	1.015	-4.599	0.000	-6.657	-2.678
Tags_Other_tags	-3.0611	0.271	-11.300	0.000	-3.592	-2.530
Tags_Ringing	-4.4302	0.233	-19.033	0.000	-4.886	-3.974
Tags_Will revert after reading the email	3.3450	0.183	18.256	0.000	2.986	3.704

	Features	VIF
5	Tags_Closed by Horizzon	1.33
11	Tags_Other_tags	1.29
14	Tags_switched off	1.24
10	Tags_Not doing further education	1.14
7	Tags_Interested in full time MBA	1.10
2	Lead Source_Welingak Website	1.09
6	Tags_Graduation in progress	1.09
9	Tags_Lost to EINS	1.06
1	What is your current occupation_Working Profes...	0.95
8	Tags_Interested in other courses	0.44
13	Tags_Will revert after reading the email	0.32
4	Tags_Already a student	0.29
12	Tags_Ringing	0.20
3	Last Activity_SMS Sent	0.12
0	What is your current occupation_Unemployed	0.11

ROC Curve

- Before the ROC Curve, we are using a 0.5 cutoff for all the observations. But the ROC curve gives us the best cutoff value.
- We use this cutoff value to predict the output across the target variable.
- We again find accuracy, precision, and specificity which are the final solution for the train data.
- Next step is to apply the model to the test data which is not seen by the model till now.
- Depending on the outcome we find out how the model is behaving on the test data set.


Receiver operating characteristic example
True Positive Rate
False Positive Rate or [1 - True Negative Rate]
ROC curve (area = 0.96)
0.0
0.2
0.4
0.6
0.8
1.0


accuracy
sensi
speci
prob
0.0
0.2
0.4
0.6
0.8
1.0

Results and Conclusion

- From the final ML model we get 91% accuracy, sensitivity of around 81% and specificity of around 97% and after selecting the best cut-off which is 0.3 we calculate the accuracy of 90%, sensitivity of 94% and specificity of around 88% and after applying model to the test data set we concluded that model gives us accuracy of 90%, sensitivity of 96%, specificity of around 87%, precision score of 83% and recall score of 96%.
- From these results as determine that the model has made good predictions and the company should focus on leads that are closed by 'Horizzon', lost to EINS, should avoid leads that are too busy/marked as switched off, already a student and who are not interested in further education.
  The marketing team should also consider leads that are interested in full time MBA courses and will revert back after reading the email.