

Lead Scoring – Case Study

Koushik Pal

Shyam Mohan Azad

Preeti Panda

Email: koushikbackin@gmail.com

Email: angelnikhilrohilla@gmail.com

Email: preeti.p0016@gmail.com

Lead Scoring – Case Study

Problem Statement :

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%.



Business Goal & Our Case Study Goal

Business Goal :

X Education needs help in selecting the most promising leads, i.e. the leads that are most likely to convert into paying customers. The company needs a model where in you a lead score is assigned to each of the leads such that the customers with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance. The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%.

Our Case Study Goal :

To 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.

To adjust to if the company's requirement changes in the future so you will need to handle these as well.

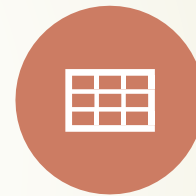
Steps Involved



1. Read and
understand the
data



2. Clean the data



3. Prepare the data
for Model Building



4. Model Building



5. Model Evaluation



6. Making
Predictions on the
Test Set

Read and Understanding on Data

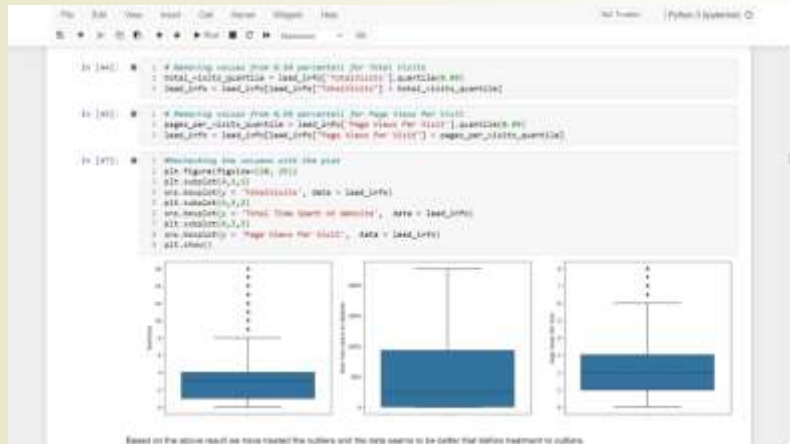
- We have imported the Lead.csv file using the pandas library and we have examined the data frame using the below commands
- `lead_info.head()`
`,lead_info.info(),lead_info.shape` and
`lead_info.describe()`
- We found the data set contains null values and there can be outliers and thus need to be handled appropriately .

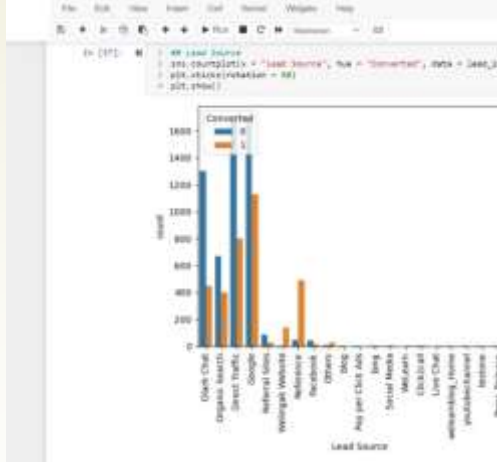
```
url>>> pandas.read_csv('Lead.csv')
In[10]: lead_info = pd.read_csv('Lead.csv')
In[11]: lead_info.info()
Out[11]:
Int64Index: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   Prospect ID         1000 non-null   int64
 1   Lead Number         1000 non-null   int64
 2   Lead Origin         1000 non-null   object
 3   Lead Source         1000 non-null   object
 4   Do Not Call         1000 non-null   object
 5   Mobile No.         1000 non-null   object
 6   Email                1000 non-null   object
 7   Total Calls         1000 non-null   int64
 8   Total Time Spent on mobile 1000 non-null   int64
 9   Page Views Per Visit 1000 non-null   int64
10   Last Activity       1000 non-null   object
11   Country              1000 non-null   object
12   Specialization       1000 non-null   object
13   How old you hear about Education 1000 non-null   object
14   What is your current occupation 1000 non-null   object
15   What matters most to you in choosing a course 1000 non-null   object
16   Name                1000 non-null   object
17   Newspaper Article    1000 non-null   object
18   Education Forum     1000 non-null   object
19   Advertisement        1000 non-null   object
20   Through Recommendation 1000 non-null   object
21   Receive News updates about our courses 1000 non-null   object
22   Page                1000 non-null   object
23   Lead Quality         1000 non-null   object
24   Consistency in Supply Chain Content 1000 non-null   object
25   Not updates on QR Content 1000 non-null   object
```

```
In[12]: lead_info.head()
Out[12]:
Prospect ID      0
Lead Number      0
Lead Origin      0
Lead Source      0
Do Not Call      0
Mobile No.       0
Email            0
Total Calls      0
Total Time Spent on mobile 0
Page Views Per Visit 0
Last Activity    0
Country          0
Specialization   0
How old you hear about Education 0
What is your current occupation 0
What matters most to you in choosing a course 0
Name            0
Newspaper Article 0
Education Forum 0
Advertisement    0
Through Recommendation 0
Receive News updates about our courses 0
Page            0
Lead Quality     0
Consistency in Supply Chain Content 0
Not updates on QR Content 0
```


Data Cleaning

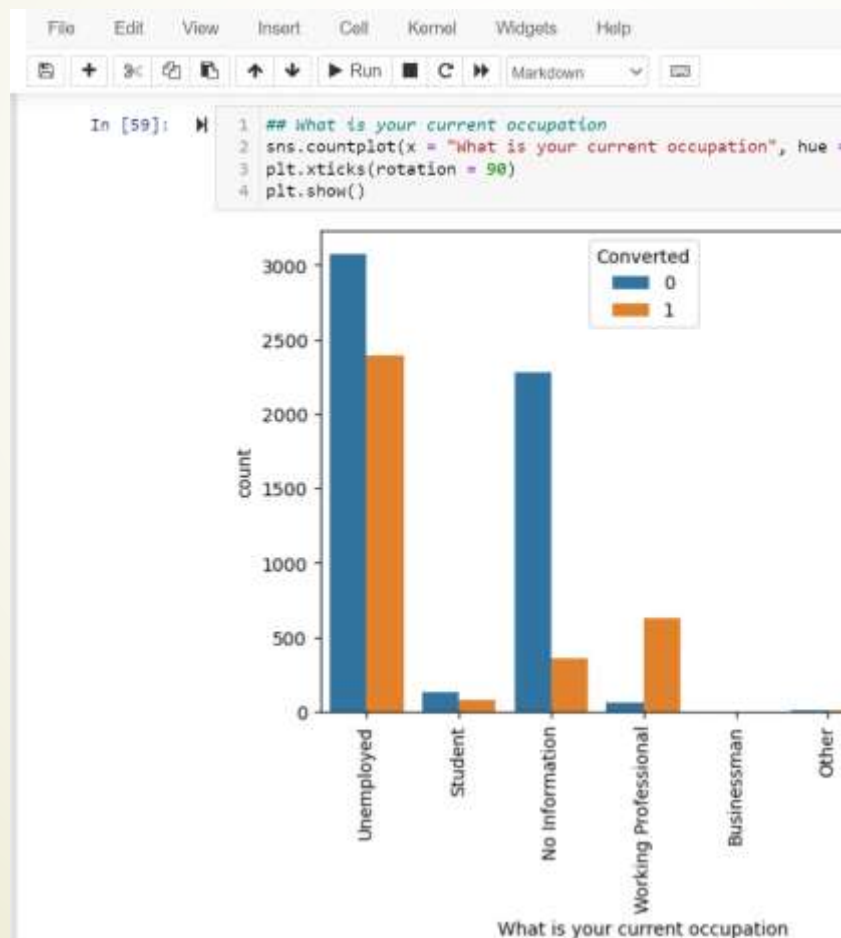
- Initially we checked for duplicates and the percentage of the null values.
- We also know based on the case study few columns have the value 'Select' and we converted it to NAN.
- We have dropped null value columns > 30 %.
- Also, a few columns were dropped that have max data under one category 'NO'.
- We have treated the outliers for numerical column .





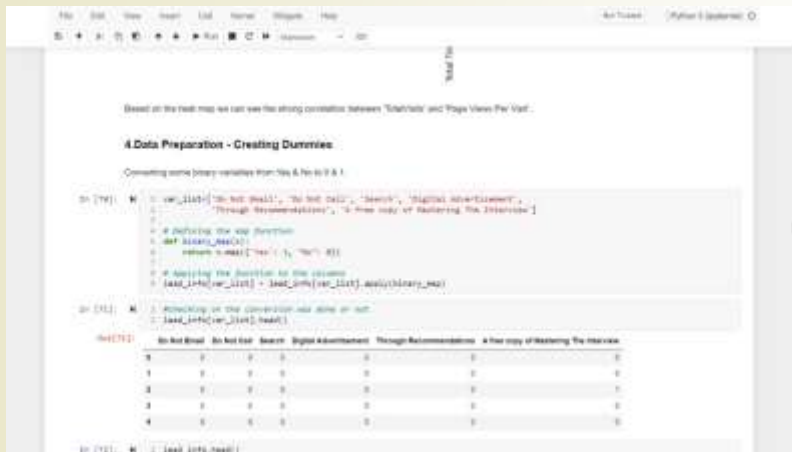
Data Analysis

- We did a Uni-variate/Bi-variate analysis of data using EDA for few columns and made the inferences on the same .



Data Preparation

- We have converted the columns that are with values as Yes/No to 0 and 1.
- We also created dummies for columns with more than one category.



Based on the heat map we can see the strong correlation between 'Search' and 'Page Views Per Visit'.

Data Preparation - Creating Dummies

Converting some binary variables from Yes & No to 0 & 1

```
In [17]: var_list = ["Do Not Email", "Do Not Call", "Search", "Digital Advertisement",  
                  "Through Recommendations", "A New Way of Reaching the Interview"]  
  
# Defining the var function  
def binary_var(x):  
    return 0 if x == "No" else 1  
  
# Applying the function to the columns  
data_list[var_list] = data_list[var_list].apply(binary_var)
```

```
In [18]: # Applying on the conversion has done or not  
data_list["conversion"] = data_list["conversion"].apply(lambda x: 1 if x == "Yes" else 0)
```

	Do Not Email	Do Not Call	Search	Digital Advertisement	Through Recommendations	A New Way of Reaching the Interview
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	1
3	0	0	0	0	0	0
4	0	0	0	0	0	0

```
In [19]: data_list.to_csv("data.csv")
```


Model Building

- We have created test-train-split using the ski-learn libraries for logistic regression model building.
- We have also scaled the data using MinMaxScaler.
- We also used RFE for the selection of the top 20 columns for model building.
- We have used the stats model to build a logistic regression model.
- We rebuild the model multiple models until we had the stable p-values and VIF <5.
- Finally, we arrived at the model 8 with all stable p-values and VIF less than 5.



Summary of Dataset:

Dep. Variable:	Count:	No. Observations:
Model:	1000	1000
Model Period:	1000	1000
Link Function:	Logit	1000
Method:	GLS	1000
Time:	1000	1000
No. Variables:	20	1000
Dependent Type:	Logistic	1000

Summary of Features:

Feature	Count	No. Observations:
Model	1000	1000
Model Period	1000	1000
Link Function	1000	1000
Method	1000	1000
Time	1000	1000
No. Variables	20	1000
Dependent Type	Logistic	1000

Summary of Features (continued):

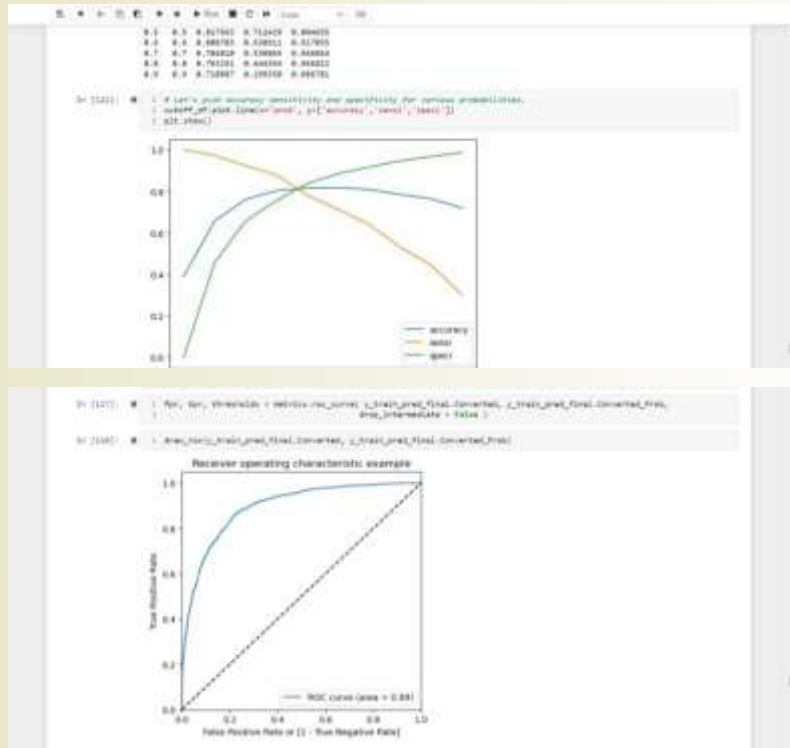
Feature	Count	No. Observations:
Model	1000	1000
Model Period	1000	1000
Link Function	1000	1000
Method	1000	1000
Time	1000	1000
No. Variables	20	1000
Dependent Type	Logistic	1000

Summary of Features (continued):

Feature	Count	No. Observations:
Model	1000	1000
Model Period	1000	1000
Link Function	1000	1000
Method	1000	1000
Time	1000	1000
No. Variables	20	1000
Dependent Type	Logistic	1000

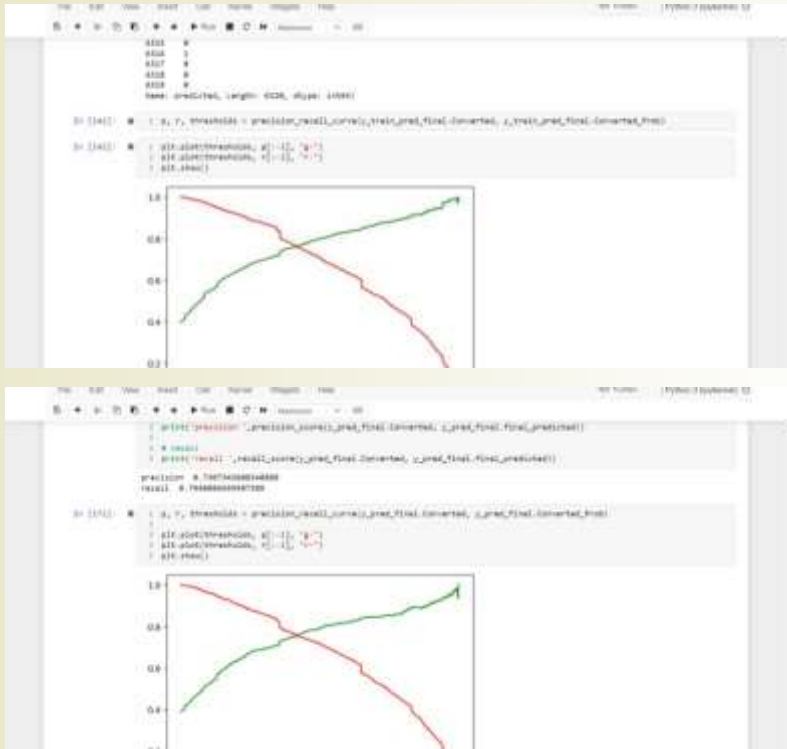
Model Predictions

- After building the model we have to predict the model based on the metrics to see its overall performance on the train data set.
- We used the confusion matrix and calculated the accuracy score.
- We also didn't stick with the accuracy score and tried to calculate the sensitivity, specificity, and false positive rate.
- Also, we checked by plotting the ROC curve.
- We also found the optimal cut-off point based on the probability of balanced sensitivity and specificity.



Model Prediction- Cont'd

- we will need to now predict with the obtained cut-off on the test set.
- We continue the overall metrics to make predictions on the test set with accuracy, sensitivity, specificity, and false positive rate.
- We also made a confusion matrix.
- We also calculated the Precision and Recall for both the train and test data sets.





Conclusion



- As we have checked Sensitivity, Specificity, Precision, and Recall as Metrics, we have considered the optimal cutoff limit as 0.37 for calculating the final prediction.
- Accuracy, Sensitivity, and Specificity values of the test set are around 81%, 79%, and 82% which are approximately closer to the respective values calculated using the trained set.
- Also, the lead score calculated in the trained set of data shows the conversion rate on the final predicted model is around 80%
- Hence overall this model seems to be good.
- Firstly, need to sort out the best prospects from the leads you have generated. 'TotalVisits', 'Total Time Spent on Website', and 'Page Views Per Visit' which contribute most towards the probability of a lead getting converted.
- Focus on converted leads.
- Hold question-answer sessions with leads to extract the right information you need about them.
- Make further inquiries and appointments with the leads to determine their intention and mentality to join online courses.