Leads Scoring
Case Study

Case Study Description

X Education has appointed you to help them select the most promising leads, i.e. the leads that are most likely to convert into paying customers. The company requires you to build a model wherein you need to assign a lead score to each of the leads such that the customers with a higher lead score have a higher conversion chance and the customers with a 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%

Goals:

1.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.

2.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.

Process Followed

Understanding Data	Cleaning Data	EDA	Model Preparation	Model building	Model Evaluation
• Understood data set shape, column types, value range, etc.	• Identified Null values and Outliers • Treated them	• Basic visualization • Converting variables to binary • Adding Dummy variables	• Split data (70-30) • Scaled numeric variables	• Utilized logistic regression • Selected features automatically using RFE • Fine tuned manually using p-value and VIF	• Checked accuracy, sensitivity, Specificity, ROC curve & AUC • Compared results for test and train dataset

Understanding Data

We started with importing the data set and understanding it based on the type of columns, the ranges and percentiles of the numerical columns and general value breakdowns.

In [154]:
```
# Importing dataset
lead_data = pd.read_csv("C:/Users/vinay/Downloads/Lead+Scoring+Case+Study/Lead Scoring Assignment/Leads.csv")
lead_data.head()
```

Out[154]:

	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	Asymmetriqu Profile Inde
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No	Select	Select	02.Medium	02.Mediu
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No	Select	Select	02.Medium	02.Mediu
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	01.Hig
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	...	No	Select	Mumbai	02.Medium	01.Hig
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.0	...	No	Select	Mumbai	02.Medium	01.Hig

5 rows × 37 columns

In [155]:
```
lead_data.shape
```

Out[155]: (9240, 37)

In [196]:
```
lead_data.describe()
```

Out[196]:

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
count	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000000
mean	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.344883
std	23405.995698	0.486714	4.854853	548.021466	2.161418	1.386694	1.811395
min	579533.000000	0.000000	0.000000	0.000000	0.000000	7.000000	11.000000
25%	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.000000
50%	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.000000
75%	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.000000
max	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.000000

In [156]:
```
lead_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 37 columns):
Prospect ID                                      9240 non-null object
Lead Number                                      9240 non-null int64
Lead Origin                                      9240 non-null object
Lead Source                                      9204 non-null object
Do Not Email                                     9240 non-null object
Do Not Call                                      9240 non-null object
Converted                                        9240 non-null int64
TotalVisits                                      9103 non-null float64
Total Time Spent on Website                      9240 non-null int64
Page Views Per Visit                             9103 non-null float64
Last Activity                                    9137 non-null object
Country                                          6779 non-null object
Specialization                                   7802 non-null object
How did you hear about X Education               7033 non-null object
What is your current occupation                  6550 non-null object
What matters most to you in choosing a course    6531 non-null object
```

Cleaning Data

Identified columns with null values and treated them based on the data distribution. Deleted the ones that were not required and changed null to different values wherever required

Some columns have "Select" as an entry, which indicates that nothing was selected there and should be replaced by Null

```
In [157]: select_present = lead_data.apply(lambda col: col.str.count('Select').sum())
          select_present_cols = select_present[select_present > 0].index.tolist()
          select_present_cols

Out[157]: ['Specialization',
           'How did you hear about X Education',
           'Lead Profile',
           'City']
```

```
# Identifying columns with null values and treat them
100*(lead_data.isnull().sum()/lead_data.shape[0]).sort_values(ascending=False)

How did you hear about X Education              78.463203
Lead Profile                                    74.188312
Lead Quality                                    51.590909
Asymmetrique Profile Score                      45.649351
Asymmetrique Activity Score                     45.649351
Asymmetrique Profile Index                      45.649351
Asymmetrique Activity Index                     45.649351
City                                            39.707792
Specialization                                  36.580087
Tags                                            36.287879
What matters most to you in choosing a course   29.318182
What is your current occupation                 29.112554
Country                                         26.634199
TotalVisits                                      1.482684
Page Views Per Visit                             1.482684
Last Activity                                    1.114719
Lead Source                                      0.389610
```

78% missing values and no pattern; hence, it is better to drop the column

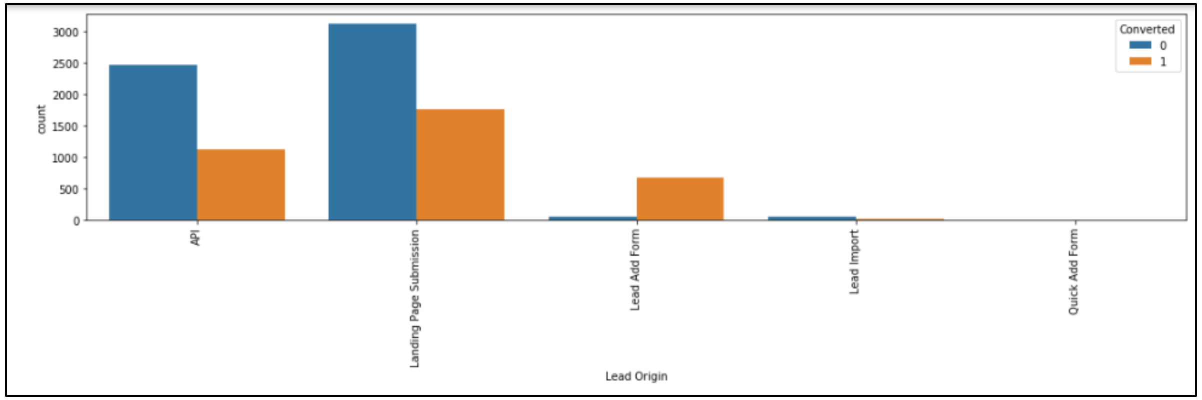
```
In [164]: lead_data.drop(["Lead Profile"],axis=1,inplace = True)

In [165]: lead_data["Lead Quality"].value_counts()

Out[165]: Might be             1560
          Not Sure             1092
          High in Relevance     637
          Worst                 601
          Low in Relevance      583
          Name: Lead Quality, dtype: int64
```

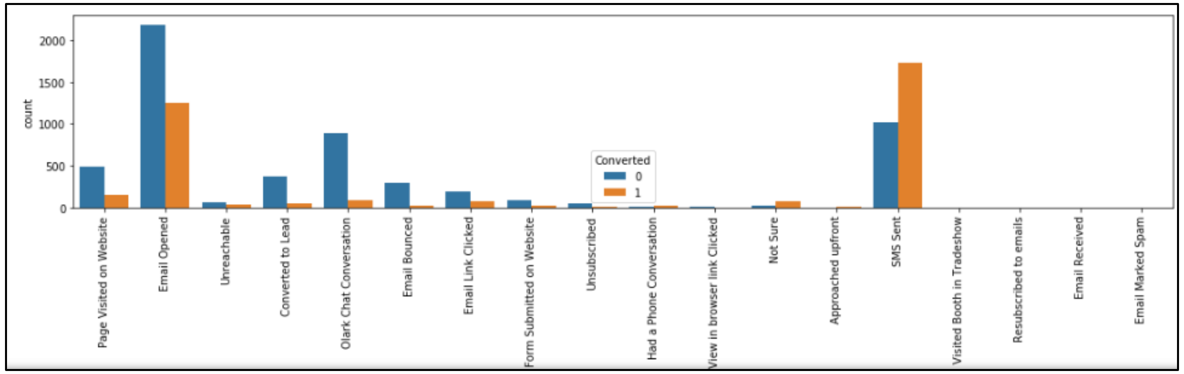
Given we have an entry that highlights the Not sure attribute, we can change NA to Not Sure

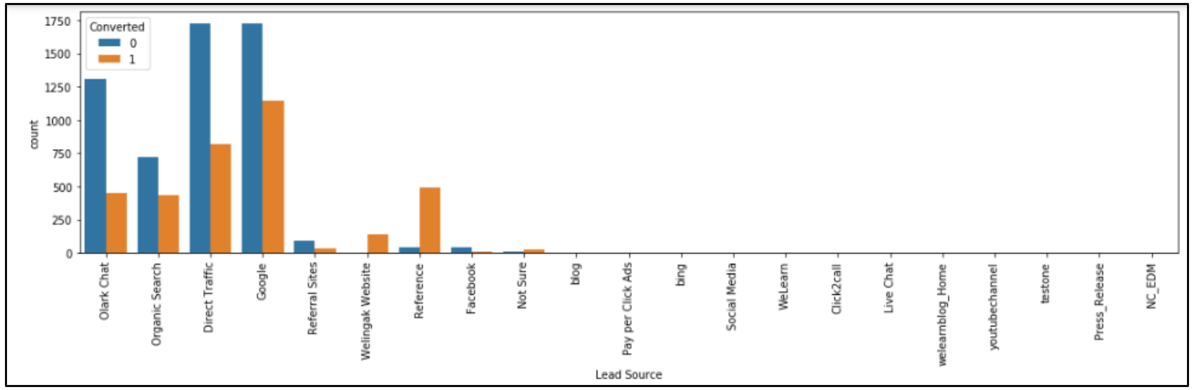
```
In [166]: lead_data["Lead Quality"].replace(np.nan,"Not Sure", inplace = True)

In [167]: lead_data["Lead Quality"].value_counts()

Out[167]: Not Sure             5859
          Might be             1560
          High in Relevance     637
          Worst                 601
          Low in Relevance      583
          Name: Lead Quality, dtype: int64
```

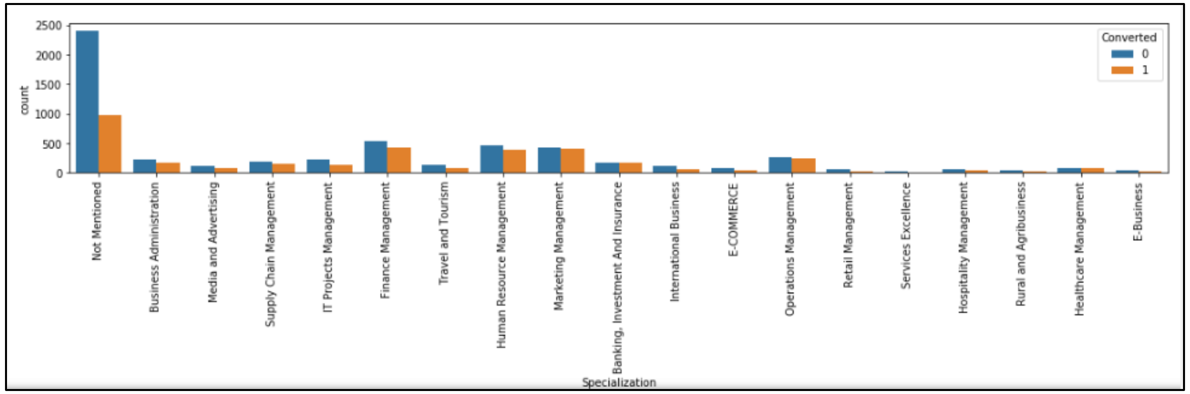
EDA

General Observations: SMS sent feature has the highest conversion, Followed by Email Opened, Modified and page visit. Lead Source feature we observe referrals ,Welinkak website have the highest conversion compared to others. Lead To Conversion Ratio for these features were noticed to be above 80% Business Swelinck & Advertising, Marketing management, Insurance & banking, OS management, health care & hospitality management









Model building

We dropped multiple variables due to their high VIF scores or their insignificant p-values. Our final model included 13 variables that were significant and provided a stable result. Organization should focus on the source from Wellingak Website and consider the time a person is spending on the website while reaching out for a sales call.

First Model

Dep. Variable:	Converted	No. Observations:	6298
Model:	GLM	Df Residuals:	6277
Model Family:	Binomial	Df Model:	20
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-2101.1
Date:	Mon, 17 Jul 2023	Deviance:	4202.2
Time:	22:50:51	Pearson chi2:	6.28e+03
No. Iterations:	21		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.3121	0.163	1.919	0.055	-0.007	0.631
Do Not Email	-1.1883	0.185	-6.438	0.000	-1.550	-0.827
TotalVisits	1.4539	0.337	4.318	0.000	0.794	2.114
Total Time Spent on Website	3.8660	0.158	24.537	0.000	3.557	4.175
Page Views Per Visit	-1.4852	0.304	-4.893	0.000	-2.080	-0.890
Newspaper	-23.6145	4.82e+04	-0.000	1.000	-9.45e+04	9.44e+04
Through Recommendations	21.9195	2.15e+04	0.001	0.999	-4.21e+04	4.21e+04
Lead Origin_Lead Add Form	2.7647	0.257	10.773	0.000	2.262	3.268
Lead Source_Not Sure	0.9915	0.723	1.372	0.170	-0.425	2.408
Lead Source_Olark Chat	1.4167	0.143	9.913	0.000	1.137	1.697
Lead Source_Welingak Website	4.3983	1.037	4.241	0.000	2.366	6.431
Last Activity_Not Sure	-1.8102	0.488	-3.706	0.000	-2.767	-0.853
Last Activity_Olark Chat Conversation	-1.2341	0.182	-6.785	0.000	-1.591	-0.878
Last Activity_SMS Sent	0.4349	0.152	2.863	0.004	0.137	0.733
Last Activity_View in browser link Clicked	-22.4482	1.89e+04	-0.001	0.999	-3.7e+04	3.7e+04
Lead Quality_Might be	-1.3692	0.152	-8.991	0.000	-1.668	-1.071
Lead Quality_Not Sure	-3.3786	0.136	-24.924	0.000	-3.644	-3.113
Lead Quality_Worst	-5.5328	0.391	-14.166	0.000	-6.298	-4.767
Last Notable Activity_Had a Phone Conversation	2.5851	1.227	2.106	0.035	0.180	4.991

Final Model

Dep. Variable:	Converted	No. Observations:	6298
Model:	GLM	Df Residuals:	6284
Model Family:	Binomial	Df Model:	13
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-2125.6
Date:	Mon, 17 Jul 2023	Deviance:	4251.1
Time:	22:50:53	Pearson chi2:	6.23e+03
No. Iterations:	7		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0406	0.150	0.272	0.786	-0.253	0.334
Do Not Email	-1.1640	0.180	-6.450	0.000	-1.518	-0.810
TotalVisits	0.5633	0.280	2.014	0.044	0.015	1.111
Total Time Spent on Website	3.8495	0.157	24.588	0.000	3.543	4.156
Lead Origin_Lead Add Form	3.1901	0.243	13.145	0.000	2.714	3.666
Lead Source_Olark Chat	1.6511	0.133	12.449	0.000	1.391	1.911
Lead Source_Welingak Website	4.2686	1.035	4.123	0.000	2.240	6.298
Last Activity_Not Sure	-2.0750	0.497	-4.177	0.000	-3.049	-1.101
Last Activity_Olark Chat Conversation	-1.2601	0.180	-6.995	0.000	-1.613	-0.907
Lead Quality_Might be	-1.2917	0.151	-8.582	0.000	-1.587	-0.997
Lead Quality_Not Sure	-3.2962	0.133	-24.704	0.000	-3.558	-3.035
Lead Quality_Worst	-5.4565	0.390	-13.996	0.000	-6.221	-4.692
Last Notable Activity_Had a Phone Conversation	2.4469	1.248	1.961	0.050	0.001	4.893
Last Notable Activity_SMS Sent	1.7106	0.090	18.967	0.000	1.534	1.887

Model building

We witnessed a good increase in the number of respondents categorized as positives, which is good in our case as correct leads will result to higher revenue. A high sensitivity and specificity means that categorization is good and the organization would not waste time in reaching out to folks that might not be our customer

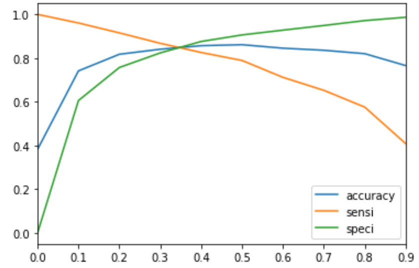
Train Data Prediction at cut-off 0.5

Predicted/ Actual	Not Lead	Lead
Not Lead	3524	366
Lead	508	1900

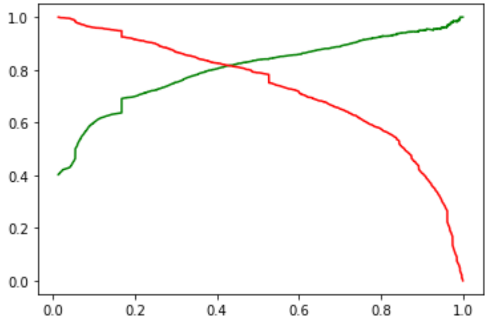
Accuracy	86%
Sensitivity	79%
Specificity	91%

Finding Optimal Cut-off

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




```
plt.plot(thresholds, p[:-1], "g-")
plt.plot(thresholds, r[:-1], "r-")
plt.show()
```



Train Data Prediction at cut-off 0.33

Predicted/ Actual	Not Lead	Lead
Not Lead	3283	607
Lead	351	2057

Accuracy	85%
Sensitivity	85%
Specificity	84%

Model Evaluation

Both the models witnessed a similar evaluation scores indicating that the model is performing well and can be considered for business purposes. A high score across evaluation metrics on test data indicates the model performs well and utilizing this by the organization would work in their favor

Final Model on Train Data

Predicted/ Actual	Not Lead	Lead
Not Lead	3283	607
Lead	351	2057

Accuracy	85%
Sensitivity	85%
Specificity	84%

Result on Test Data

Predicted/ Actual	Not Lead	Lead
Not Lead	1366	309
Lead	166	859

Accuracy	82%
Sensitivity	84%
Specificity	82%

Lead Scores

Lead scores were assigned to every row to highlight if a row is a hot lead or not. While the accuracy was around 80% for the train data, it went over 80% for the test data. These lead scores can be directly utilized before a sales call and conversation should always start with the higher ones.

Lead Score on Train Data

Addilng Lead Score

In [105]:
```
y_train_pred_final['Lead_score'] = y_train_pred_final.Converted_Prob.map(lambda x: round(x*100))
y_train_pred_final.head(10)
```

Out[105]:

	Converted	Converted_Prob	Id	predicted	Lead_score
4714	0	0.167349	4714	0	17
5794	1	0.974656	5794	1	97
876	1	0.053928	876	0	5
7767	1	0.721225	7767	1	72
7895	1	0.296118	7895	0	30
1182	0	0.246172	1182	0	25
959	0	0.389443	959	0	39
1444	1	0.974985	1444	1	97
106	0	0.063224	106	0	6
4615	0	0.051103	4615	0	5

In [106]:
```
#Checking if we have 80% correct prediction [WHICH IS TRUE POSITIVE]
check_df = y_train_pred_final.loc[y_train_pred_final['Converted']==1,['Converted','predicted']]
check_df['predicted'].value_counts()
```

Out[106]:
```
1    1900
0     508
Name: predicted, dtype: int64
```

In [107]:
```
1900/(1900+508)
```

Out[107]:
```
0.7890365448504983
```

Lead Score on Test Data

Addilng Lead Score

In [159]:
```
y_pred_final['Lead_score'] = y_pred_final.Converted_Prob.map(lambda x: round(x*100))
y_pred_final.head(10)
```

Out[159]:

	Converted	ID	Converted_Prob	final_predicted	Lead_score
0	0	9122	0.167349	0	17
1	0	4366	0.051053	0	5
2	1	4044	0.967776	1	97
3	0	5527	0.053928	0	5
4	0	2752	0.045786	0	5
5	1	4165	0.375775	1	38
6	0	8744	0.001530	0	0
7	0	1600	0.603254	1	60
8	1	7641	0.996399	1	100
9	0	4741	0.057797	0	6

In [157]:
```
#Checking if we have 80% correct prediction [WHICH IS TRUE POSITIVE]
check_df = y_pred_final.loc[y_pred_final['Converted']==1,['Converted','final_predicted']]
check_df['final_predicted'].value_counts()
```

Out[157]:
```
1    859
0    166
Name: final_predicted, dtype: int64
```

In [158]:
```
859/(859+166)
```

Out[158]:
```
0.8380487804878048
```