

Lead Score Case Study

- Submitted by:-
- **Deepa Paikar**
- **Shubham Siddharth**

Problem Statement

- An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses.
- The typical lead conversion rate at X education is around 30%.
- The Company wants to increase it to 80%.

Goal

- 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.

Methodology

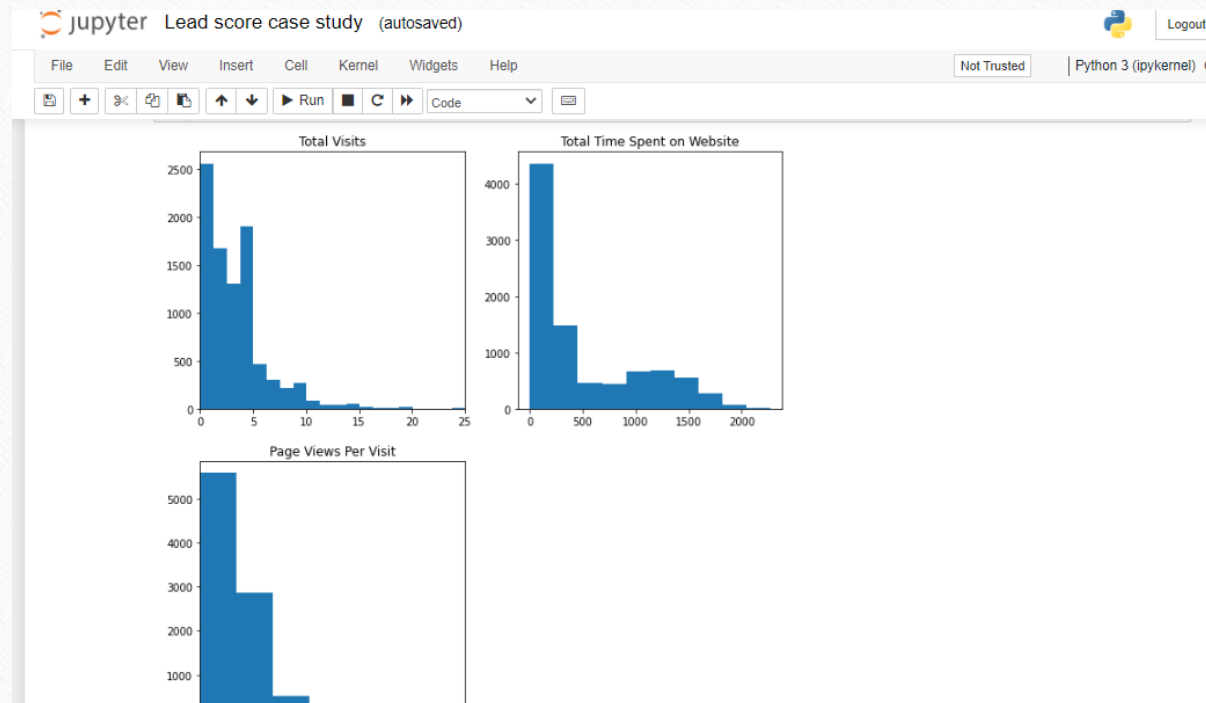
1. Data Cleaning and making data set operational.
 - Check Non values in all columns
 - Check for duplicates data in all columns
 - Drop the columns which is not useful
 - Take care of outlier if its there.
 - Impute values according to requirement
2. Use EDA for further classification analysis like univariate or Bivariate Analysis
3. Regression modelling- logistic regression build up
4. Model Analysis .
5. Conclusion and recommendation.

Data Knowledge(Impacting factors)

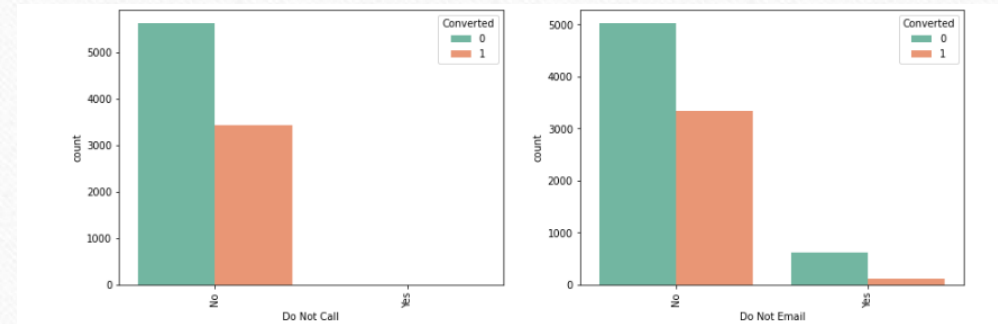
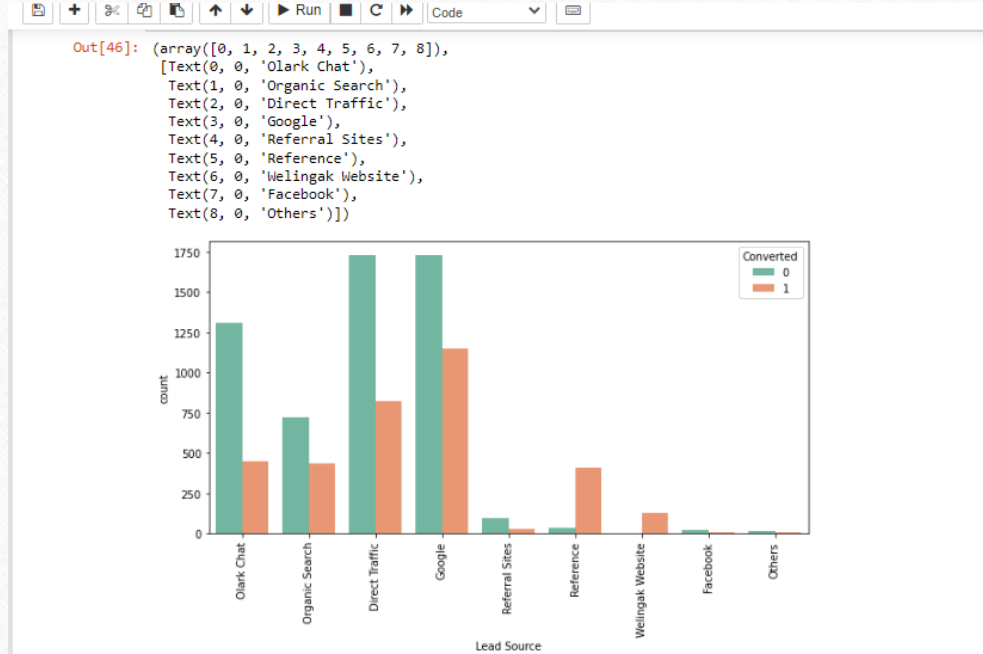
- Dropping of few columns where Null values are greater than 40% like Asymmetrique Activity Index, Asymmetrique Profile Index, Asymmetrique Activity Score, Asymmetrique Profile Score and etc....
- Tola number of rows- 97 and column- 9240
- Project Profile also dropped.

EDA- Univariate analysis

Numerical Variables



Lead Score Visualization



Data Preparation(creating dummy variable)

Creating Dummy variables

```
In [53]: 1 # Creating a dummy variable for the categorical variables and dropping the first one.
2 dummy_data = pd.get_dummies(df[['Lead Origin', 'Lead Source', 'Last Activity', 'Specialization', 'What is your current occupa
3         'City', 'Last Notable Activity']], drop_first=True)
4 dummy_data.head()
```

Out[53]:

	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	Lead Source_Ork Chat	Lead Source_Organic Search	Lead Source_Others	Lead Source_Reference	Lead Source_Refere Si
0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0
4	1	0	0	0	1	0	0	0	0	0

5 rows × 10 columns

Splitting the data into train and test set

```
In [57]: 1 from sklearn.model_selection import train_test_split
2
3 # Putting feature variable to X
4 X = df.drop(['Prospect ID', 'Converted'], axis=1)
5 X.head()
6
```

Out[57]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	...	Last Notable Activity_Form Submitted on Website	Last Notable Activity_Had a Phone Conversation	Activ
0	0	0	0.0	0	0.0	0	0	0	0	0	...	0	0	0
1	0	0	5.0	674	2.5	0	0	0	0	0	...	0	0	0
2	0	0	2.0	1532	2.0	1	0	0	0	0	...	0	0	0
3	0	0	1.0	305	1.0	1	0	0	0	0	...	0	0	0
4	0	0	2.0	1428	1.0	1	0	0	0	1	...	0	0	0

5 rows × 14 columns

Model Building

Model Building

```
In [64]: 1 # Import 'LogisticRegression'
2 from sklearn.linear_model import LogisticRegression
3 # Import RFE
4 from sklearn.feature_selection import RFE
5 # Running RFE with 15 variables as output
6
7 rfe = RFE(estimator=LogisticRegression(), n_features_to_select=15)
8 rfe = rfe.fit(X_train, y_train)
```

```
In [65]: 1 rfe.support_
```

```
Out[65]: array([[False, False, False, True, False, True, True, False, False,
False, True, False, False, False, False, True, False, True,
False, False, False, False, False, True, True, 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, True, False, False, False,
True, False, False, False, False, False, True, False, False,
False, False, False, True, False, False, False, False, True,
True, False, False]])
```

```
In [66]: 1 list(zip(X_train.columns, rfe.support_, rfe.ranking_))
```

Model Building

Model-1

```
In [68]: 1 import statsmodels.api as sm
```

```
In [69]: 1 #BUILDING MODEL #1
2
3 X_train_sm = sm.add_constant(X_train[cols])
4 logml = sm.GLM(y_train, X_train_sm, family = sm.families.Binomial())
5 res = logml.fit()
6 res.summary()
```

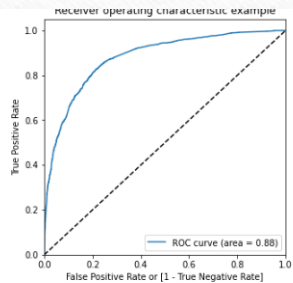
```
Out[69]:
```

Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	6351	
Model:	GLM	Df Residuals:	6335	
Model Family:	Binomial	Df Model:	15	
Link Function:	Logit	Scale:	1.0000	
Method:	IRLS	Log-Likelihood:	-2648.0	
Date:	Mon, 17 Oct 2022	Deviance:	5296.1	
Time:	21:52:40	Pearson chi2:	6.48e+03	
No. Iterations:	23	Pseudo R-squ. (CS):	0.3899	
Covariance Type:	nonrobust			
	coef	std err	z P> z [0.025 0.975]	
const	-0.4348	0.123	-3.522 0.000	-0.677 -0.193
Total Time Spent on Website	1.0884	0.040	27.528 0.000	1.011 1.166

ROC Curve

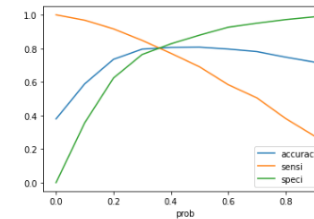
optimise



Since we have higher (0.88) area under the ROC curve, therefore our model is a good one.

Final optimal

```
In [97]: 1 # Let's plot accuracy sensitivity and specificity for various probabilities.  
2 cutoff_df.plot.line(x='prob', y=['accuracy', 'sensi', 'speci'])  
3 plt.show()
```



From the curve above, 0.35 is the optimum point to take it as a cutoff probability.

Result evaluation

- Train Data:
- Accuracy : 80%
- Sensitivity : 82 %
- Specificity : 79 %
- Test Data:
- Accuracy : 80 %
- Sensitivity : 81 %
- Specificity : 78 %

Results :

Comparing the values obtained for Train & Test:

Train Data:

Accuracy : 80%
Sensitivity : 82 %
Specificity : 79 %

Test Data:

Accuracy : 80 %
Sensitivity : 81 %
Specificity : 78 %

Thus we have achieved our goal of getting a ballpark of the target lead conversion rate to be around 80% . The Model seems to predict the Conversion Rate very well and we should be able to give the CEO confidence in making good calls based on this model to get a higher lead conversion rate of 80%.

Conclusion Part

- Lead Origin Lead Add Form are the ones the to whom the calls should be made as conversion rate is high
- The company should make calls to the leads who are the "working professionals" as they are more likely to get converted.
- The company should make calls to the leads who spent "more time on the websites" as these are more likely to get converted.
- The company should make calls to the leads coming from the lead sources "Olark Chat" as these are more likely to get converted.
- The company should not make calls to the leads who chose the option of "Do not Email" as "yes" as they are not likely to get converted.

Thank You