Lead score case study

Lead Score Case Study

X Education, an online education company catering to industry professionals, faces a challenge with **low lead conversion rates** despite attracting numerous potential customers daily through website visits, form submissions, and referrals. The company seeks to enhance efficiency by identifying 'Hot Leads'—leads with higher conversion potential. Currently, **only about 30% of acquired leads are converted to paying customers**

X Education aims to **develop a lead scoring model** that assigns scores to leads based on their likelihood of conversion. This model is intended to aid the sales team in prioritizing communication efforts and focusing on leads that are more likely to convert, **potentially raising the overall lead conversion rate to the CEO's target of around 80%.**

Approach

Define the Business goal understand the data

- Understand the business problem.
- Familiarize with data dictionary

Data cleaning and preparation

- Missing value imputation
- Outlier analysis
- Bucketing the values
- Exploratory Data analysis

Train test Splitting and feature scaling

- Fit transform train data
- Transform test data
- RFE for finalizing feature variable

Model Building and Evaluation

- P Value and VIF analysis
- Determine the optimum cut of probability
- Accuracy, Sensitivity, and Specificity calculation

Applying on test data and get the Lead score result

Test data Evaluation.
Lead score creation

Missing Value Imputation

1.Missing values are identified in the data set using the below sample function
Columns with >30 % Missing values are dropped.

```
# Calculate the percentage of missing values in each column
missing_percentages = leads_df.isnull().sum() / len(leads_df)
```

```
missing_percentages.sort_values(ascending=False)*100
```

```
Lead Quality                          51.590909
Asymmetrique Activity Index           45.649351
Asymmetrique Profile Score            45.649351
Asymmetrique Activity Score           45.649351
Asymmetrique Profile Index            45.649351
Tags                                  36.287879
```

2. Multiple columns had selected as values which indicate users have no choice and its equivalent to 'NULL'. Select converted to Null and reviewed the missing value.

```
#Replace Select with NULL
#leads_df2 = leads_df2.applymap(lambda x: '' if x == 'Select' else x)
leads_df2=leads_df2.replace('Select',np.nan)
```

3.ForColumn What is your current Occupation which is having 29% missing value-Missing value is replaced with Mode.

```
Unemployed              5600
Working Professional     706
Student                  210
Other                     16
Housewife                 10
Businessman                8
Name: What is your current occupation, dtype: int64
```

4. Column with less missing value (Less than 2%) –missing rows are dropped

```
col_row_to_drop_missing=["Page Views Per Visit",
                         "TotalVisits",'Last Activity',
                         'Lead Source']
```

```
leads_df4=leads_df3.dropna(subset=col_row_to_drop_missing
```

```
leads_df4.shape
```

```
(9074, 25)
```

Outlier analysis

Outlier analysis was conducted in Numerical variables.
Total Visits-values above P95 are removed

```
leads_df4.TotalVisits.quantile([0.5, 0.7, 0.9, 0.95, 0.99])
```

```
0.50     3.0
0.70     4.0
0.90     7.0
0.95    10.0
0.99    17.0
Name: TotalVisits, dtype: float64
```

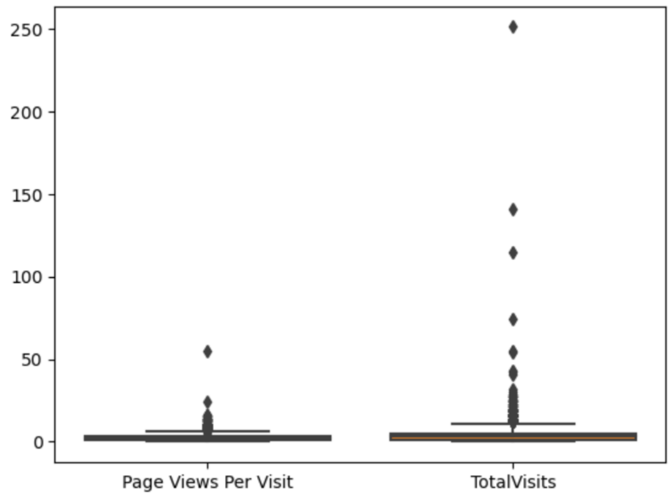
```
leads_df5=leads_df4[leads_df4['TotalVisits']<10]
leads_df5.shape
```

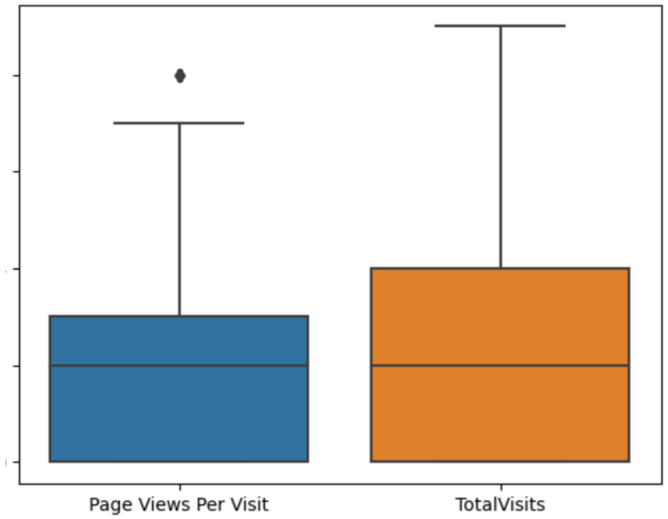
Page Views per visit-Values above P99 are removed

```
leads_df4["Page Views Per Visit"].quantile([0.5, 0.7, 0.9, 0.95, 0.99])
```

```
0.50    2.0
0.70    3.0
0.90    5.0
0.95    6.0
0.99    9.0
Name: Page Views Per Visit, dtype: float64
```

```
leads_df5=leads_df5[leads_df5['Page Views Per Visit']<9]
leads_df5.shape
```





90 % of data retained after Missing value imputation and outlier removal

Univariate Analysis

Univariate analysis was conducted on Numerical and categorical Variables and identified the Significant and non-significant parameters. Also identified few insights

The below categorical features provides significant inference about the lead conversion rate

Lead Origin-Conversion rate of the Lead Add form is high followed by Landing page submission and API

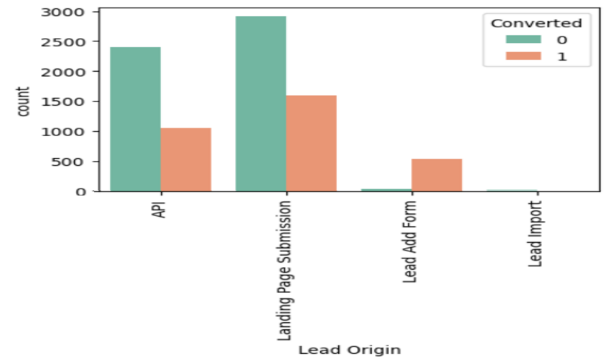
Lead source -Conversion rate is very high for Reference followed by Google, Organic search, Direct traffic, olark chat. '.

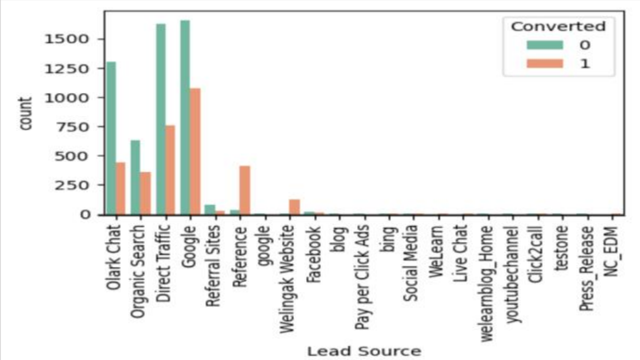
Last activity-SMS Sent have a Very high conversion rate, followed by Email Opened, Email link clicked, page visited on Website.

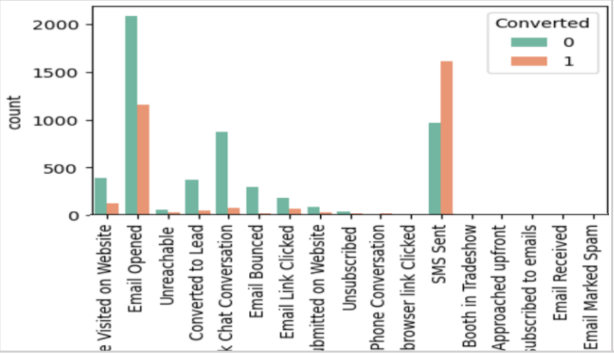
What is your current occupation-Working professionals have high conversion rate, followed by Unemployed, and students

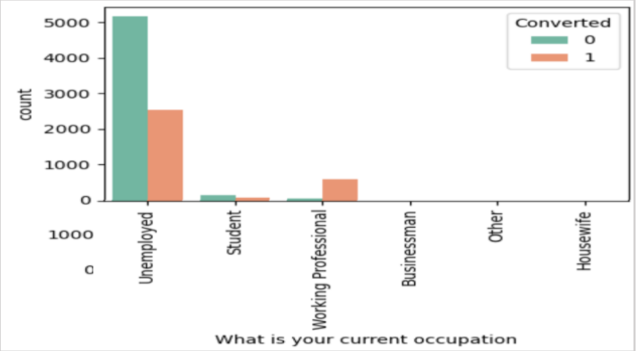
Last Notable Activity-SMS sent and Email opened to have a high conversion rate.

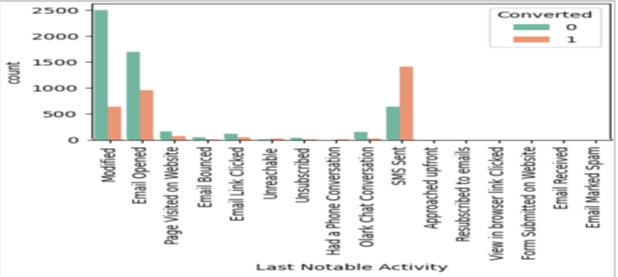
No Other Categorical features are not providing any significant inference







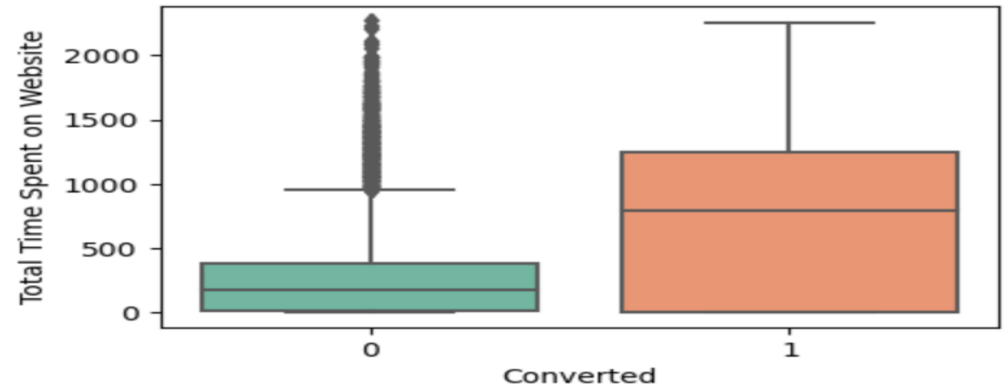


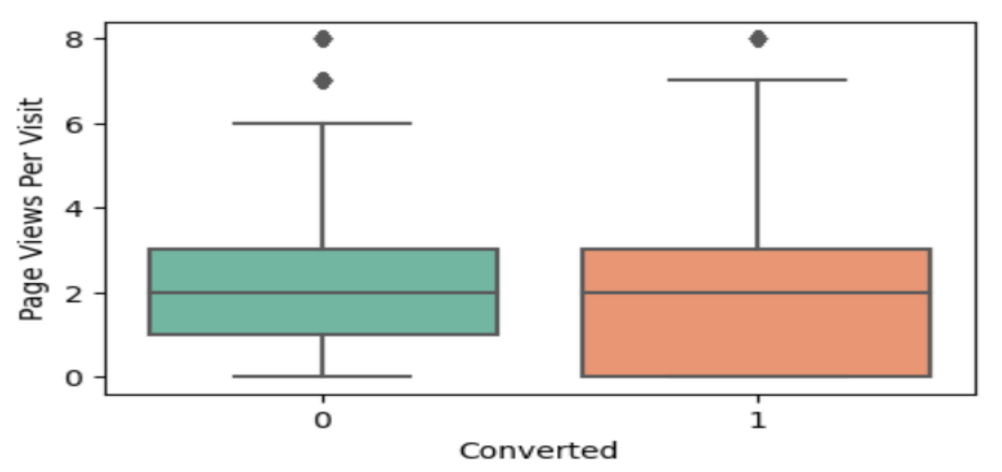


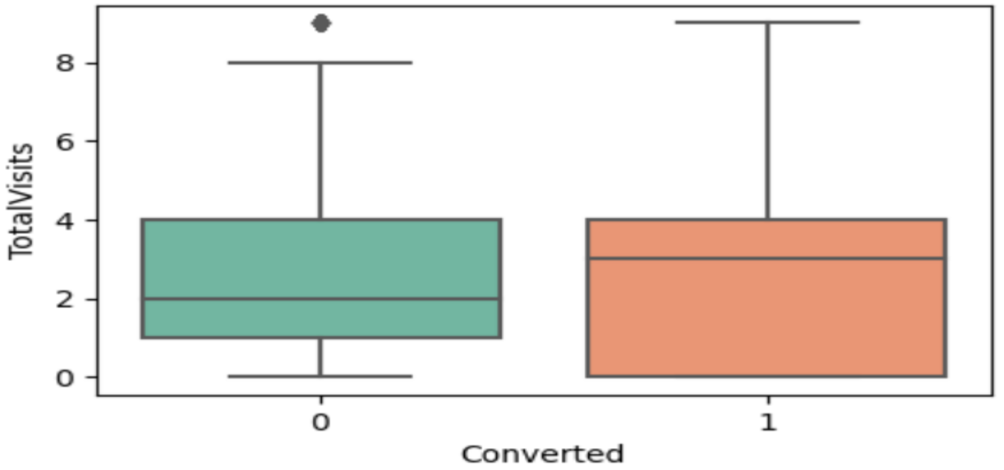
Univariate Analysis

Numerical features

Conversion rates are high for features Total time spent on the website.
Median of Total visits for converted customers is high
The converted and Non converted user count is the same for feature page views per visit







Based on the Univariate analysis below columns identified as insignificant and dropped while the model building

col_to_drop=['Prospect ID', 'Lead Number','Search','Magazine','Newspaper Article', 'X Education Forums', 'Newspaper', 'Digital Advertisement', 'Through Recommendations', 'Receive More Updates About Our Courses','Update me on Supply Chain Content', 'Get updates on DM Content','I agree to pay the amount through cheque']

Data Preparation for Model Building

1. Categorical variable with two values converted to 1 and 0

Converting some binary variables (Yes/No) to 0/1

```
63]: # List of variables to map

varlist =  ['Do Not Email', 'Do Not Call','A free copy of Mastering The Interview']

# Defining the map function
def binary_map(x):
    return x.map({'Yes': 1, "No": 0})

# Applying the function to the housing list
leads_df6[varlist] = leads_df6[varlist].apply(binary_map)
```

2. For categorical variables with multiple levels, create dummy features (one-hot encoded).

```
# Creating a dummy variable for some of the categorical variables and dropping the first one.
dummy1 = pd.get_dummies(leads_df6[['Lead Origin', 'Lead Source', 'Last Activity',
       'What is your current occupation','Last Notable Activity']], drop_first=True)

# Adding the results to the master dataframe
leads_df7 = pd.concat([leads_df6, dummy1], axis=1)
```

```
#droping the repeated variable for which dummy creation done

leads_df7=leads_df7.drop(['Lead Origin', 'Lead Source', 'Last Activity',
       'What is your current occupation','Last Notable Activity'],axis=1)
```

3. Train -Test Split

```
from sklearn.model_selection import train_test_split
# Putting feature variable to X
X = leads_df7.drop(['Converted'], axis=1)

X.head()
```

```
# Putting response variable to y
y=leads_df7["Converted"]
```

4. Feature scaling for numerical variable

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()

X_train[['TotalVisits', 'Total Time Spent on Website','Page Views Per Visit']] = scaler.fit_transform(X_train[['TotalVisits', 'T

X_train.head()
```

5. Feature selection using RFE
20 features decided to select

```
from sklearn.feature_selection import RFE
rfe = RFE(estimator=logreg, n_features_to_select=20)
rfe = rfe.fit(X_train, y_train)
```

Model Finalization

Feature variables with High P value and VIF are discarded one by one.

- Features are discarded based on below criteria-
  - High P value High VIF- Discarded first
  - High P value Low VIF- Second
  - Low P value High VIF- Third

- After multiple iterations **Model 8 provided the below result of**

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.7451	0.237	-7.371	0.000	-2.209	-1.281
Do Not Email	-1.4332	0.208	-6.882	0.000	-1.841	-1.025
Total Time Spent on Website	1.1644	0.043	27.061	0.000	1.080	1.249
Lead Source_Olark Chat	1.4800	0.108	13.658	0.000	1.268	1.692
Lead Source_Reference	4.2285	0.239	17.711	0.000	3.761	4.696
Lead Source_Welingak Website	6.8106	1.019	6.684	0.000	4.814	8.808
Last Activity_Email Opened	0.7160	0.119	6.034	0.000	0.483	0.949
Last Activity_Olark Chat Conversation	-0.9908	0.202	-4.894	0.000	-1.388	-0.594
Last Activity_Other_Activity	2.6063	0.626	4.165	0.000	1.380	3.833
Last Activity_SMS Sent	2.0259	0.122	16.657	0.000	1.788	2.264
Last Activity_Unreachable	1.2945	0.358	3.619	0.000	0.593	1.996
Last Activity_Unsubscribed	1.8883	0.571	3.304	0.001	0.768	3.008
What is your current occupation_Unemployed	-0.5657	0.219	-2.579	0.010	-0.996	-0.136
What is your current occupation_Working Professional	2.1140	0.286	7.399	0.000	1.554	2.674

Features	VIF
What is your current occupation_Unemployed	5.20
Last Activity_Email Opened	3.08
Last Activity_SMS Sent	2.59
Lead Source_Olark Chat	1.89
Last Activity_Olark Chat Conversation	1.89
What is your current occupation_Working Profes...	1.48
Total Time Spent on Website	1.34
Do Not Email	1.26
Lead Source_Reference	1.26
Last Activity_Unsubscribed	1.08
Lead Source_Welingak Website	1.06
Last Activity_Unreachable	1.05
Last Activity_Other_Activity	1.02

Model Result when chosen the probability threshold as 0.5

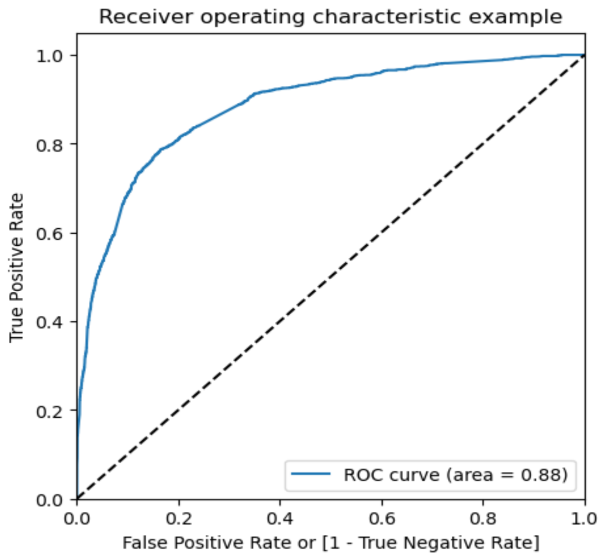
Creating a confusion matrix

```
4]: from sklearn import metrics

# Confusion matrix
confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.predicted )
print(confusion)

[[3293  380]
 [ 625 1409]]
```

```
5]: # Predicted     not_converted    converted
# Actual
# not_converted        3293     380
# converted            625      1409
```

Accuracy=82.3%
Sensitivity=69.3%
Specificity=89.7%

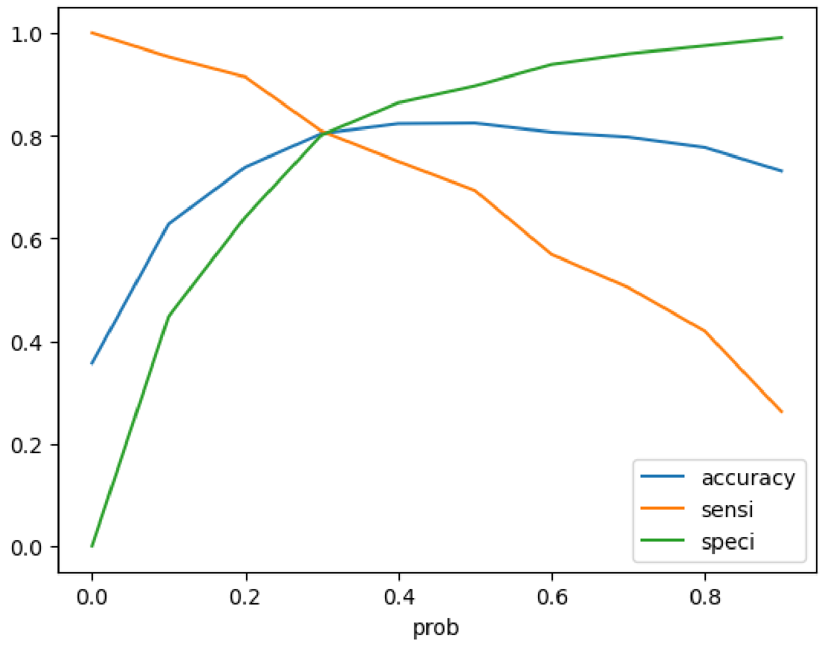
Sensitivity is less with this model



The ROC curve covers 88% of the area that means Model is good

Finding the Optimal Cutoff point

Plot accuracy,Sensitivity, and Specificity to find the cut-off point



From the curve above, 0.3 is the optimum point to take it as a cutoff probability

The optimal **cut-off point is suggested as 0.3** and below is the result which is better than the cutoff point 0.5

Accuracy=80.4%
Sensitivity=80.8%
Specificity=80.1%

Making prediction with test data

1:Transform Numerical variable

```
X_test[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']] = scaler.transform(X_test[['TotalVisits',
                                                                                                          'Total Time Spent on Webs
                                                                                                          'Page Views Per Visit']])
```

2:Assigning the model selected by the final model

```
# Assigning the columns selected by the final model to the X_test
X_test = X_test[col7]
X_test.head()
```

3:Making a prediction and append it with test data

```
# Making predictions on the test set

y_test_pred = res.predict(X_test_sm)
```

```
# Appending y_test_df and y_pred_1
y_pred_final = pd.concat([y_test_df, y_pred_1],axis=1)
```

3.Assigning lead score

```
#Assigning Lead score
y_pred_final['Lead_Score'] = y_pred_final.Converted_prob.map( lambda x: round(x*100))

y_pred_final.head()
```

	Converted	Cust_index	Converted_prob	final_predicted	Lead_Score
0	1	7482	0.996060	1	100
1	0	6071	0.070909	0	7
2	1	7793	0.099749	0	10
3	1	4564	0.379176	1	38
4	0	1674	0.262904	0	26

4. Making a confusion matrix and result

```
# Making the confusion matrix
confusion2 = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.final_predicted )
confusion2
```

```
array([[1220,  338],
       [ 175,  714]], dtype=int64)
```

Accuracy=79%
Sensitivity=80.3%
Specificity=78.3%

The above result shows that the Model Performed well on test data

The Conclusion from the EDA and Model

To bring the lead conversion rate to 80% X education should consider contacting the Lead whose

- Lead Source is through references, Wellingak website, followed by Google and organic search
- Lead activity others, SMS followed by Email opened and link clicked.
- Contact the leads who spent more time on the x education website than the leads who just visit the page
- Lead origin is Lead add form, Landing page submission and
- X education should focus more on Working professionals than students, focus on students only if they sent sms or email