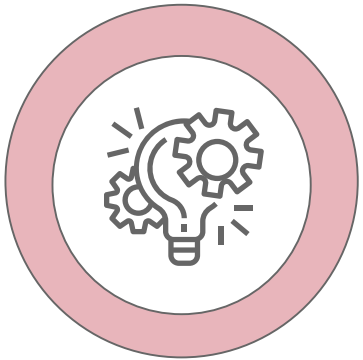
Lead Scoring Case Study Using Logistic Regression

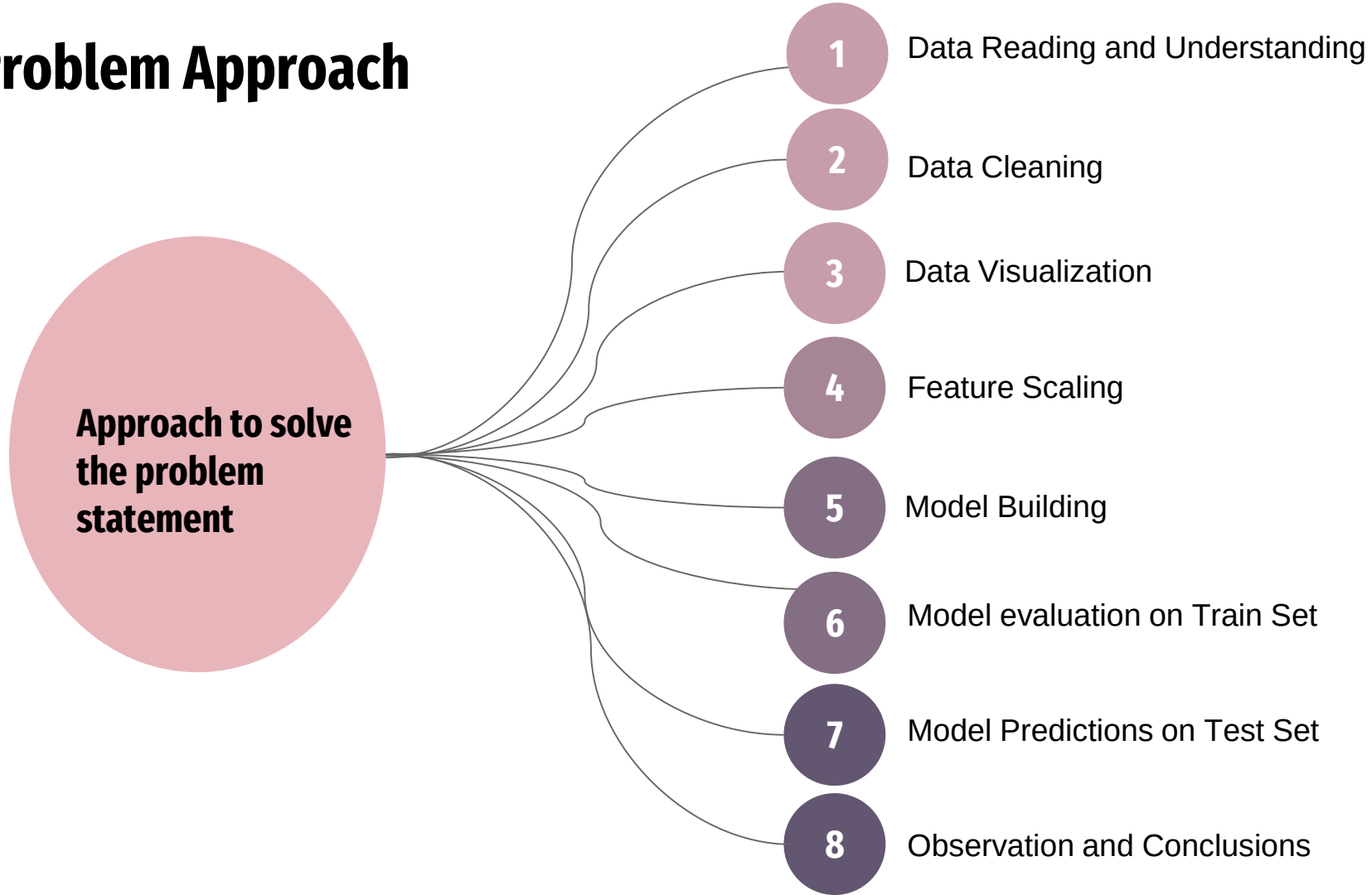
BY:
Shreyosi Chattopadhyay
Sidharth Kini
Shruti Koshti



Problem Statement

- X Education, an online course provider, seeks to enhance lead conversion efficiency. After acquiring leads through website interactions, the sales team engages in calls and emails, with a typical conversion rate of 30%.
- To improve efficiency, the company aims to identify potential leads, or "Hot Leads," to focus sales efforts more effectively.
- Strategies include implementing lead scoring, analyzing historical data for ideal customer profiles, using automation tools, personalizing communication so that we can boost overall lead conversion rates to be around 80%.

Problem Approach



Step 1: Data Cleaning

- Importing necessary Libraries and warnings
- Importing Data and Checking Data Types

In [2]:
```
df = pd.read_csv('Leads.csv')
```

In [3]:
```
df.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	Last Activity	Country	Specialization	How did you hear about X Education	What is your current occupation
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	Page Visited on Website	NaN	Select	Select	Unemployed
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	Email Opened	India	Select	Select	Unemployed
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	Email Opened	India	Business Administration	Select	Student
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	Unreachable	India	Media and Advertising	Word Of Mouth	Unemployed

Step 2: Data Cleaning

- Checking Null values, dropping the columns with null values greater than 30%
- Dropping columns which were not needed for the analysis

In [26]:
```
# Since, NaN values are more than 30%. Hence, we are dropping 'Country' column
```

In [27]:
```
df.drop(["Country"],axis=1, inplace=True)
```

In [28]:
```
df.columns
```

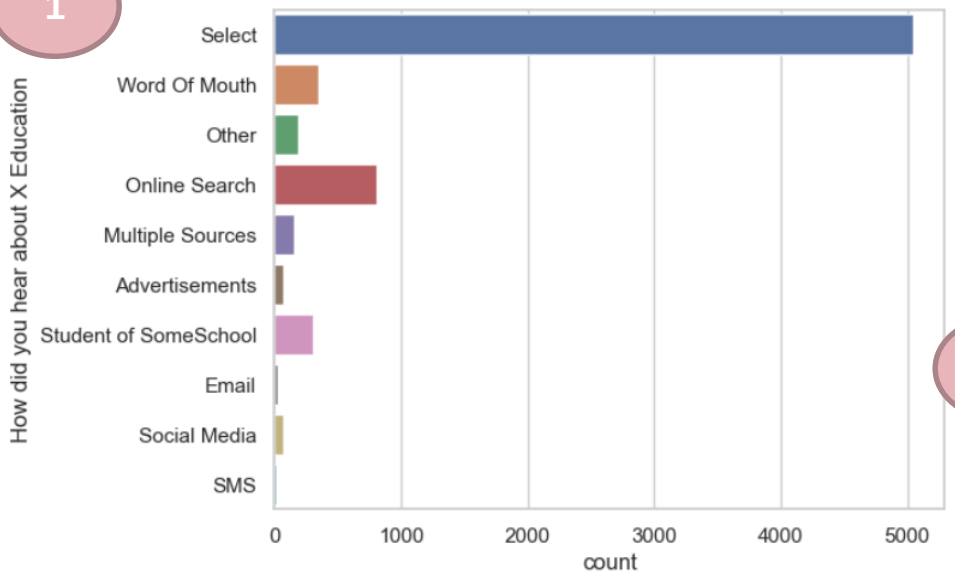
Out[28]:
```
Index(['Lead Number', 'Lead Origin', 'Lead Source', 'Do Not Email',
       'Do Not Call', 'Converted', 'TotalVisits',
       'Total Time Spent on Website', 'Page Views Per Visit', 'Last Activity',
       'Specialization', 'How did you hear about X Education',
       'What is your current occupation',
       'What matters most to you in choosing a course', 'Search', 'Magazine',
       'Newspaper Article', 'X Education Forums', 'Newspaper',
       'Digital Advertisement', 'Through Recommendations',
       'Receive More Updates About Our Courses', 'Tags',
       'Update me on Supply Chain Content', 'Get updates on DM Content',
       'Lead Profile', 'City', 'I agree to pay the amount through cheque',
       'A free copy of Mastering The Interview', 'Last Notable Activity'],
      dtype='object')
```

Step 3: Data Visualization

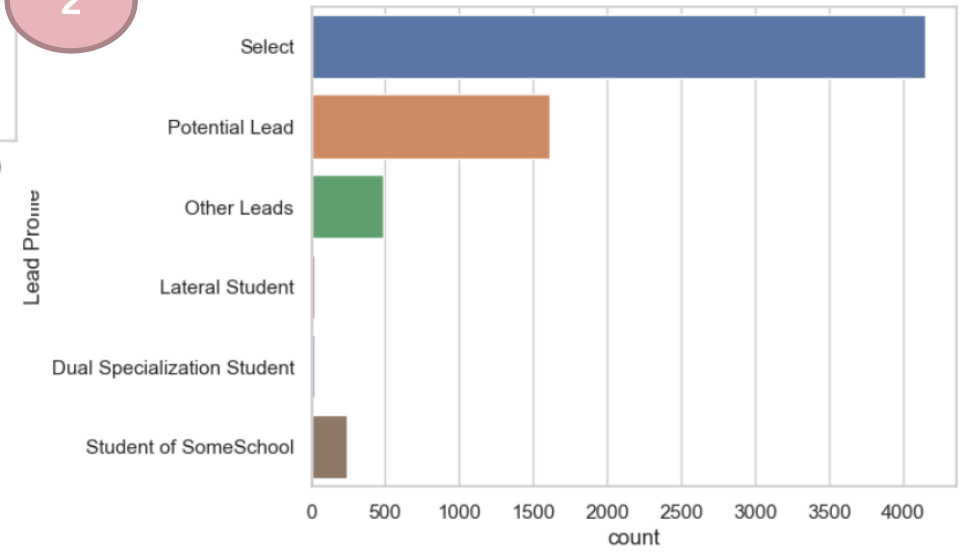
- We plotted graph for those columns which have “select” values and dropped them later on as they were equivalent to null values
- Dropped those columns which have “No” values mostly present in the data set
- Performed Analysis on Numerical columns by plotting pair plot and Heatmap.
- Performed Analysis on the categorical columns by plotting count plot.

Step 3: Data Visualization

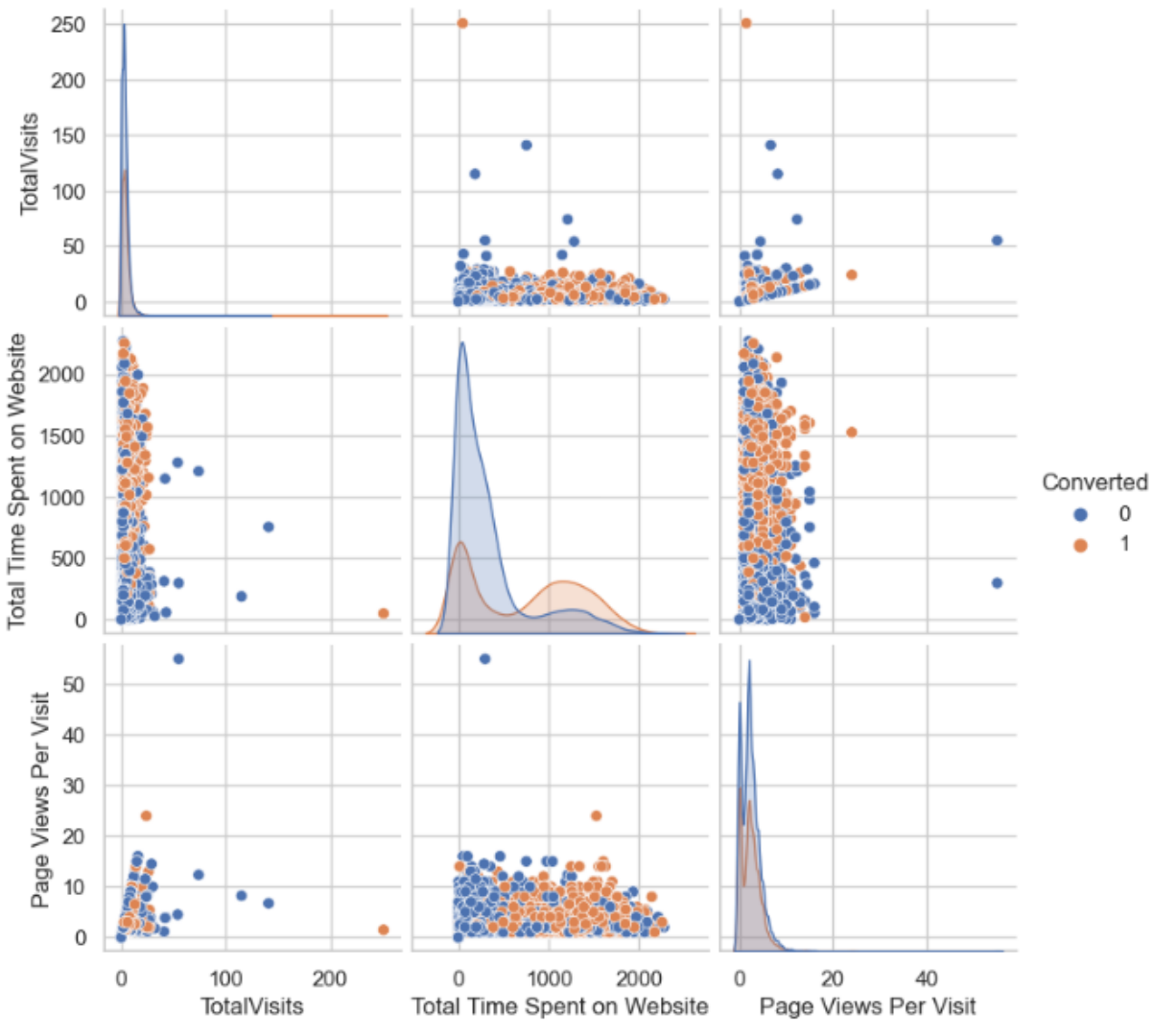
1



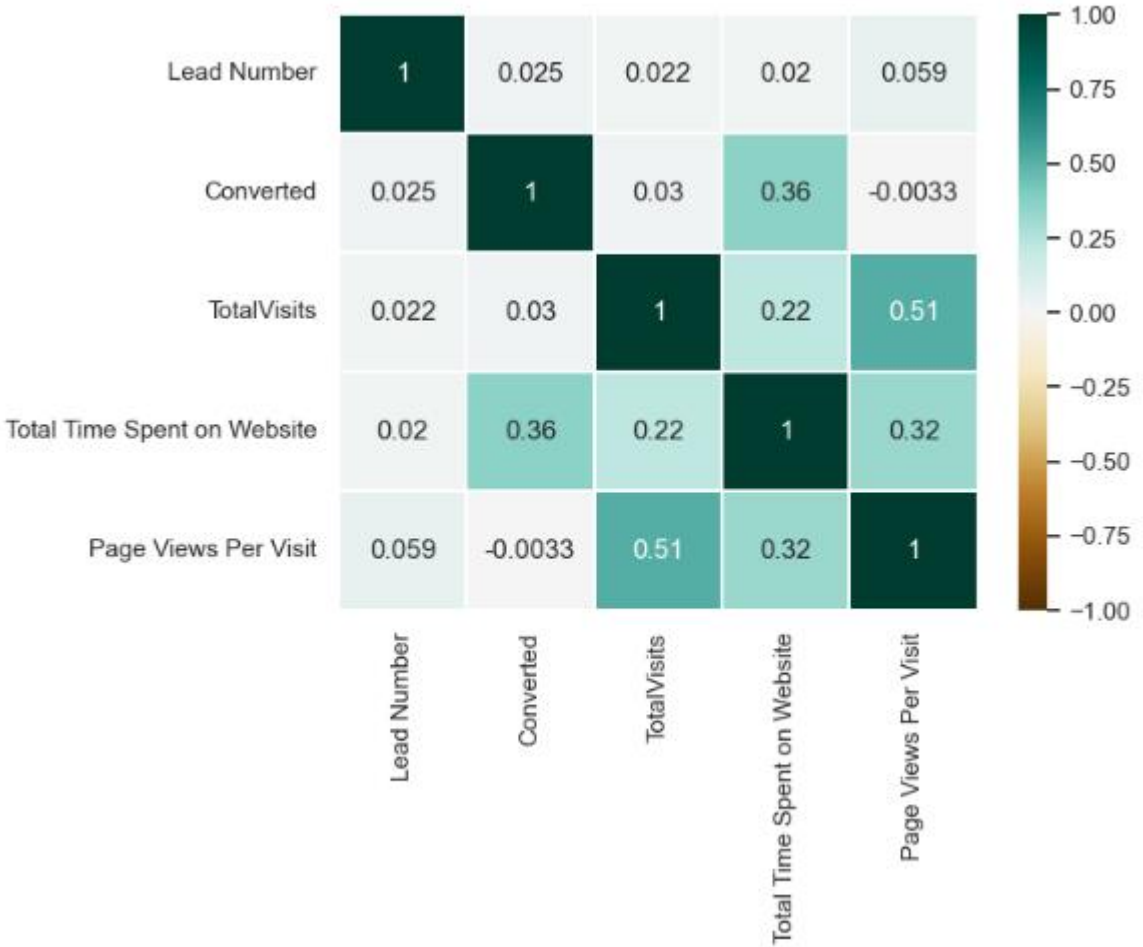
2



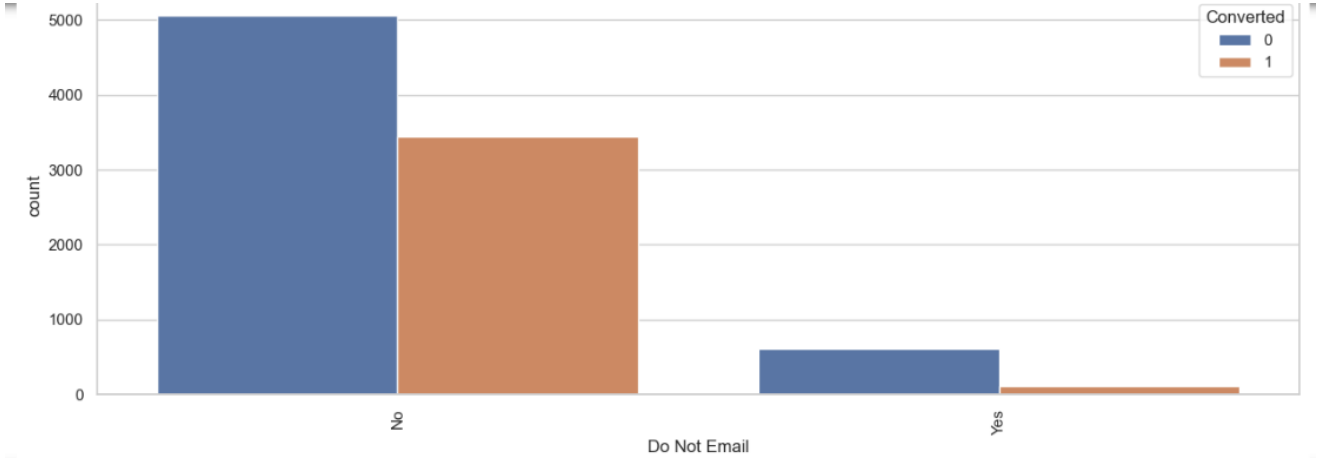
Analyzing
Numerical columns

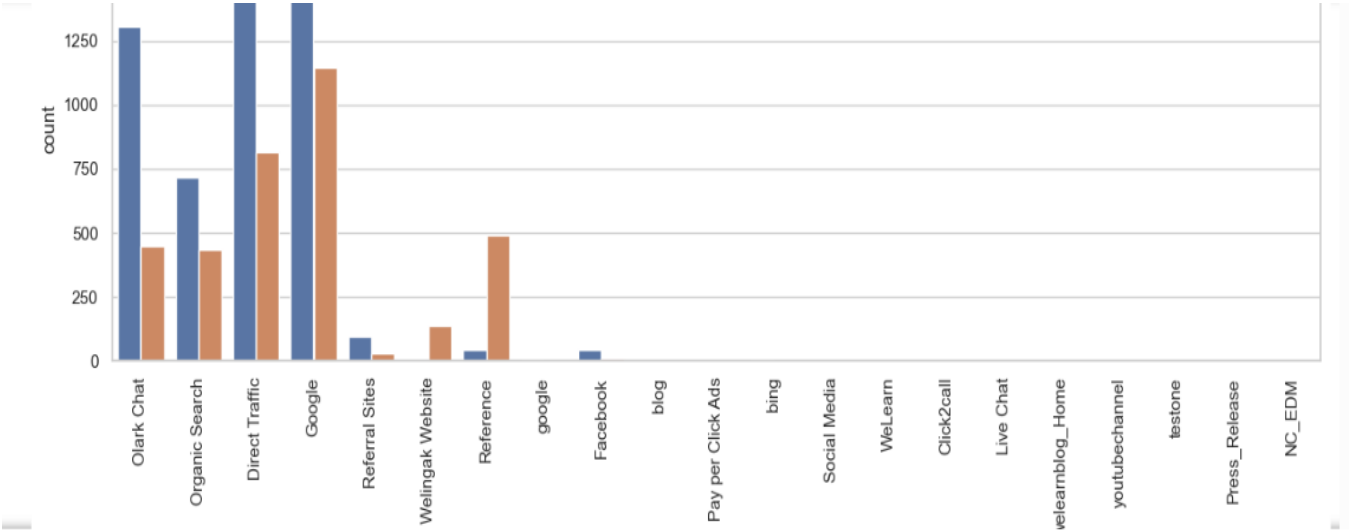


Analyzing Numerical columns



Visualizing Categorical Columns





Categorical column analysis

Conclusion as per above graphs-

1) Among Lead Origin- The people who landed to page submission got successfully converted,
2) People mostly used google as source search engine,
3) People who opened email and whom we sent message got converted as lead, Finance, HRM and Marketing management mostly converted as lead compared to other specialization,
4) People who were unemployed mostly got converted as lead

Step 4: Feature Scaling

- We converted all the categorical columns to numerical by converting them to dummy variables.
- We scaled all the Numerical variables using Standard scaler function

In [70]:
```
scale = StandardScaler()
X_train[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']] = scale.fit_transform(X_train[['TotalVisits','Tota
X_train.head()
```

Out[70]:

	Do Not Email	TotalVisits	Total Time Spent on Website	Page Views Per Visit	A free copy of Mastering The Interview	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Direct Traffic	Lead Source_Facebook	Lead Source_Google	Le Source_L C
8003	0	0.064874	-0.824395	-0.223652	1	1	0	0	1	0	0	
218	0	0.064874	-0.611929	0.753710	1	1	0	0	1	0	0	
4171	0	0.431907	-0.804919	1.731071	1	1	0	0	1	0	0	
4037	0	-0.669191	-0.943022	-1.201013	0	0	0	0	0	0	0	
3660	0	-0.669191	-0.943022	-1.201013	0	0	1	0	0	0	0	

Step 5: Model Building

- We used RFE to select 15 columns to build the model.
- We performed logistic regression and optimised variables based on p-values (<0.05) and VIF values
- Repeated above step in order to optimise our model

Dep. Variable:	Converted	**No. Observations:**	4461
Model:	GLM	**Df Residuals:**	4448
Model Family:	Binomial	**Df Model:**	12
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-869.32
Date:	Sat, 18 Nov 2023	**Deviance:**	1738.6
Time:	17:16:15	**Pearson chi2:**	5.60e+03
No. Iterations:	8	**Pseudo R-squ. (CS):**	0.6304
Covariance Type:	nonrobust		

Final Model

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.9796	0.132	-22.525	0.000	-3.239	-2.720
Do Not Email	-1.2151	0.284	-4.277	0.000	-1.772	-0.658
Lead Source_Welingak Website	3.2735	1.021	3.206	0.001	1.272	5.275
Last Activity_SMS Sent	1.6686	0.149	11.180	0.000	1.376	1.961
Tags_Closed by Horizzon	8.4918	0.731	11.619	0.000	7.059	9.924
Tags_Lost	6.1381	0.397	15.471	0.000	5.360	6.916
Tags_No phone number	-2.3347	1.015	-2.300	0.021	-4.324	-0.345
Tags_Others	3.5747	0.161	22.252	0.000	3.260	3.890
Tags_Want to take admission but has financial problems	3.6371	1.086	3.348	0.001	1.508	5.766
Tags_Will revert after reading the email	6.4821	0.207	31.285	0.000	6.076	6.888
Tags_in touch with EINS	2.6733	0.817	3.270	0.001	1.071	4.275
Last Notable Activity_Modified	-1.4311	0.168	-8.540	0.000	-1.760	-1.103
Last Notable Activity_Olark Chat Conversation	-1.0705	0.479	-2.237	0.025	-2.009	-0.133

	Features	VIF
2	Last Activity_SMS Sent	1.479403
8	Tags_Will revert after reading the email	1.375697
6	Tags_Others	1.226306
10	Last Notable Activity_Modified	1.184516
1	Lead Source_Welingak Website	1.147730
0	Do Not Email	1.091474
3	Tags_Closed by Horizzon	1.060418
5	Tags_No phone number	1.046848
4	Tags_Lost	1.044958
11	Last Notable Activity_Olark Chat Conversation	1.016366
9	Tags_in touch with EINS	1.002191
7	Tags_Want to take admission but has financial problems	1.002174

Features & VIF

Continued…

Step 5: Model Building

```
In [90]: print("The final Variables selected are:")
         cols

         The final Variables selected are:

Out[90]: Index(['Do Not Email', 'Lead Source_Welingak Website',
                'Last Activity_SMS Sent', 'Tags_Closed by Horizzon', 'Tags_Lost',
                'Tags_No phone number', 'Tags_Others',
                'Tags_Want to take admission but has financial problems',
                'Tags_Will revert after reading the email', 'Tags_in touch with EINS',
                'Last Notable Activity_Modified',
                'Last Notable Activity_Olark Chat Conversation'],
               dtype='object')
```

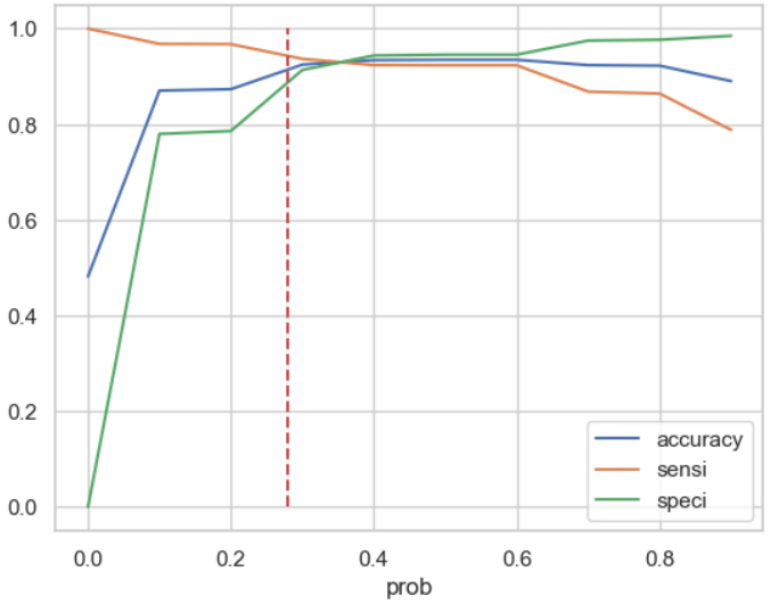
Final Variables

```
In [97]: #classification report
         print(classification_report(y_train_pred_final['Converted'], y_train_pred_final['predicted'] ))

                       precision    recall  f1-score   support

                    0       0.93      0.95      0.94      2312
                    1       0.94      0.92      0.93      2149

             accuracy                           0.93      4461
            macro avg       0.94      0.93      0.93      4461
         weighted avg       0.93      0.93      0.93      4461
```

Train Set Report

Step 6: Model evaluation on Train Set

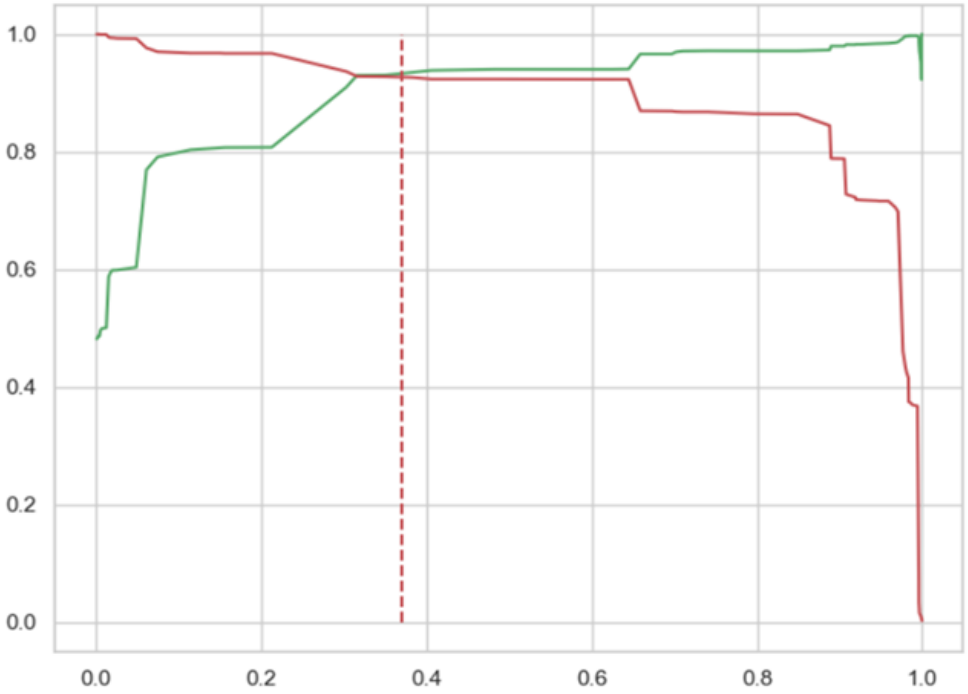
- We created confusion matrix and calculated the metrics- accuracy, sensitivity, specificity, precision and recall
- Plotted graph for all the matrix and found optimum cutoff as 0.28
- Plotted ROC curve to find the area under the curve



ROC curve

Step 7: Model Predictions on Test Set

• After calculating optimum cut off by plotting graph which we got as 38%, then we calculated the metrics and predicted data on the test set.



Any Prospect Lead with **Conversion Probability higher than 38 % to be a hot Lead.**

Test Set Report

In [140]:
```
#classification report
print(classification_report(y_pred_final.Converted, y_pred_final.final_predicted))
```

```
              precision    recall  f1-score   support

           0       0.94      0.95      0.95       996
           1       0.94      0.94      0.94       916

    accuracy                           0.94      1912
   macro avg       0.94      0.94      0.94      1912
weighted avg       0.94      0.94      0.94      1912
```

Step 8: Observation and Conclusions

When the Company has limited time and resources, it should approach Hot_leads i.e. those leads who have more than 80% of conversion chances to achieve maximum conversion & to avoid useless phone calls.

- Train Data:
  Accuracy : 91.23%
  Sensitivity : 93.6%
  Specificity : 91.3%
  Precision : 91%
  Recall : 94%

- Test Data:
  Accuracy : 94%
  Sensitivity :93.8%
  Specificity : 94.7%
  Precision : 94%
  Recall : 94%

Additional conclusion

- Lead Source- Company can focus on the lead source from “Google”, “Direct Traffic” and “Reference”
- Lead Origin- Company can focus on the customers who have landed on the “Landing Page Submission”
- What is your current occupation- Company can focus on customers who are unemployed.
- Specialization- “Finance Management”, “Marketing Management” and “Human Resource Management”.
- Last Activity- Company can focus on the customers who have – “SMS Sent” and “Email Opened”