LEAD SCORING CASE STUDY

UPGRAD & IIITB, DATA SCIENCE DSC65 – FEB 2024

BY G B SHRUTHI AND SHUANSHU TIWARI

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 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 OUTCOME

- 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.

STEPS TO PROCEED

- Import necessary modules
- Reading and Analyzing Data
- Data Cleaning
- Data Visualization
- Data Preparation
- Model Creation
- Model Evaluation
- Understanding ROC Curve & Precision Trade-off
- Finding optimal cutoff values
- Predictions on Test dataset

STEPS TO PROCEED

- Firstly import the necessary modules required for model building.
- Next we need to analyze the data of the given excel like checking column headers, no of columns and their names with datatypes etc.
- Check the shape of the data frame.
- Check the number of non-null value rows present in each and every column and also its data types using info() function.
- Check the statistical information of data frame using describe() function.
- First we need to check the percentage of null values of columns and drop the columns with 40% null value percentage as their presence does impact the statistics.
- Drop the highly skewed columns as their presence does impact the statistics.

STEPS TO PROCEED

- Remaining columns who have less percentage of null values , we will impute the columns with Mean()/Median() – Numerical columns and Mode() – Categorical columns.
- After removing all the null values from the data, reset the index and create the data into a new data frame.
- Irrelevant columns are dropped because they are not meant for analysis if required.
- Identify the Outliers using boxplot, if any outliers calculate the IQR for them, calculate the upper bounds and lower bounds.

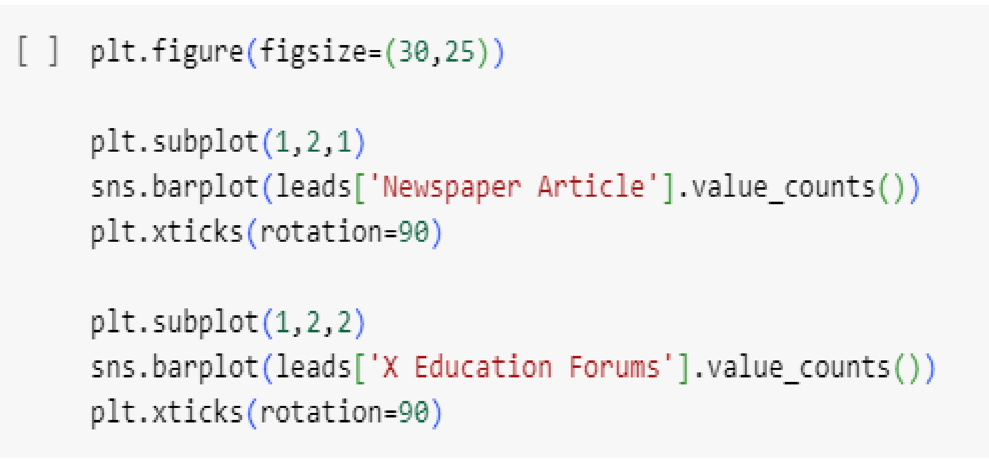
```
IQR=Q3-Q1 ;
lower_bound=Q1-1.5*IQR ;
upper_bound=Q3+1.5*IQR
app[col]=np.where(app[col]>upper_bound,upper_bound,app[col])
app[col]=np.where(app[col]<lower_bound,lower_bound,app[col])
```

STEPS TO PROCEED

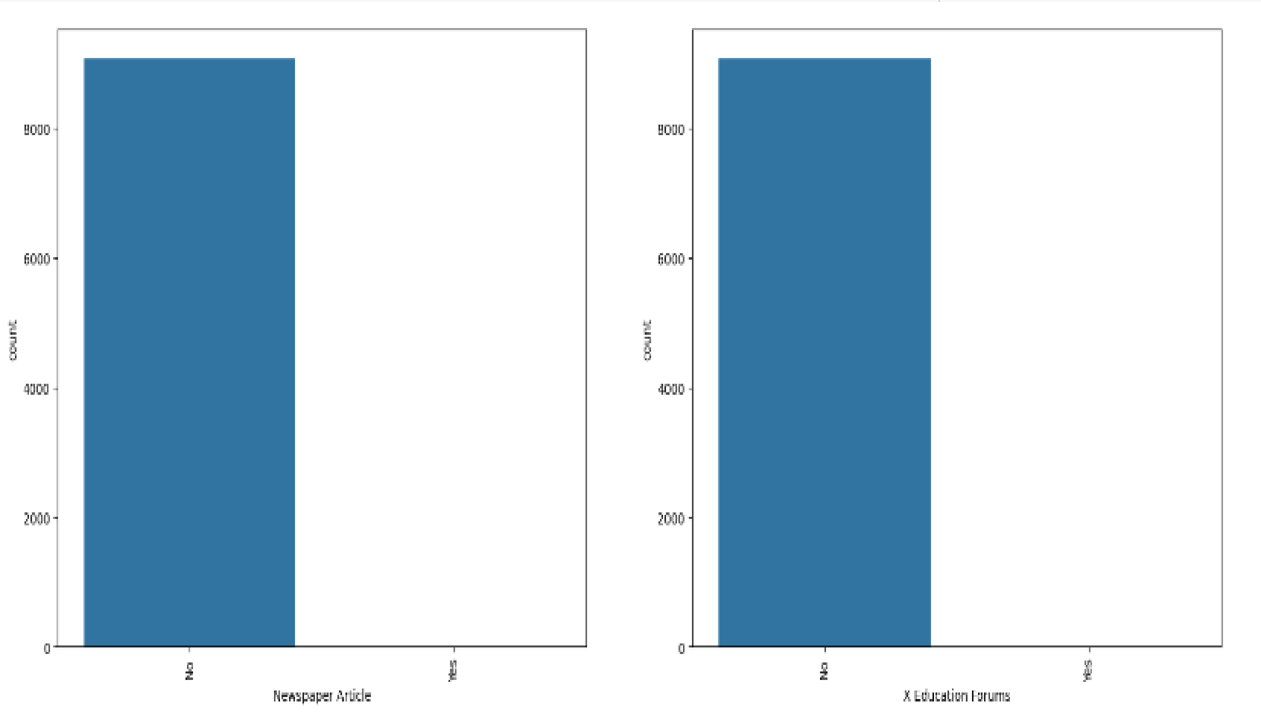
- Segregate all the columns of data frame based on this data type into Categorical and Numerical Variables for Data Visualization using MatplotLib and seaborn libraries.
- Now Univariate Analysis is done on the Categorical variables using BAR Plot and it is also done on the Numerical variables using HIST Plot and their insights are taken accordingly.
- Bivariate Analysis is done using COUNT Plot with respective to Target feature – Converted for categorical features and their insights are taken.
- Bivariate Analysis is done using PAIR plot for numerical features.
- Multivariate Analysis is done between Continuous Numerical Variables using Heat Maps.

UNIVARIATE ANALYSIS

- The univariate analysis is done for all the categorical features of the Leads Data.
- For example Newspaper Article, X Education Forums columns



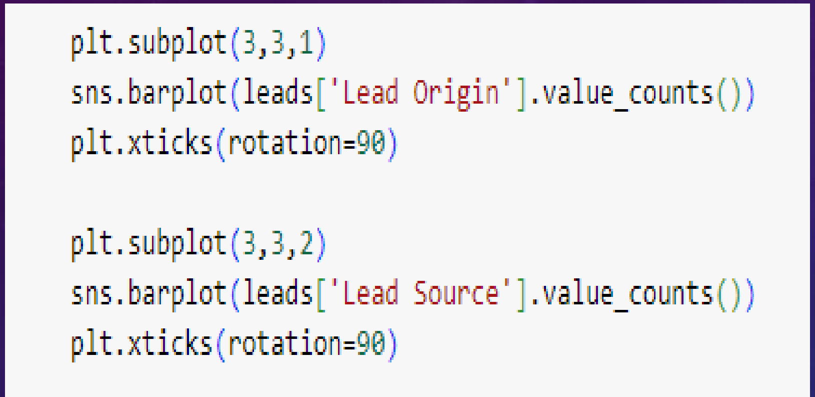
```
[ ]  plt.figure(figsize=(30,25))

     plt.subplot(1,2,1)
     sns.barplot(leads['Newspaper Article'].value_counts())
     plt.xticks(rotation=90)

     plt.subplot(1,2,2)
     sns.barplot(leads['X Education Forums'].value_counts())
     plt.xticks(rotation=90)
```



- These 2 columns are highly skewed.
- These can be dropped.

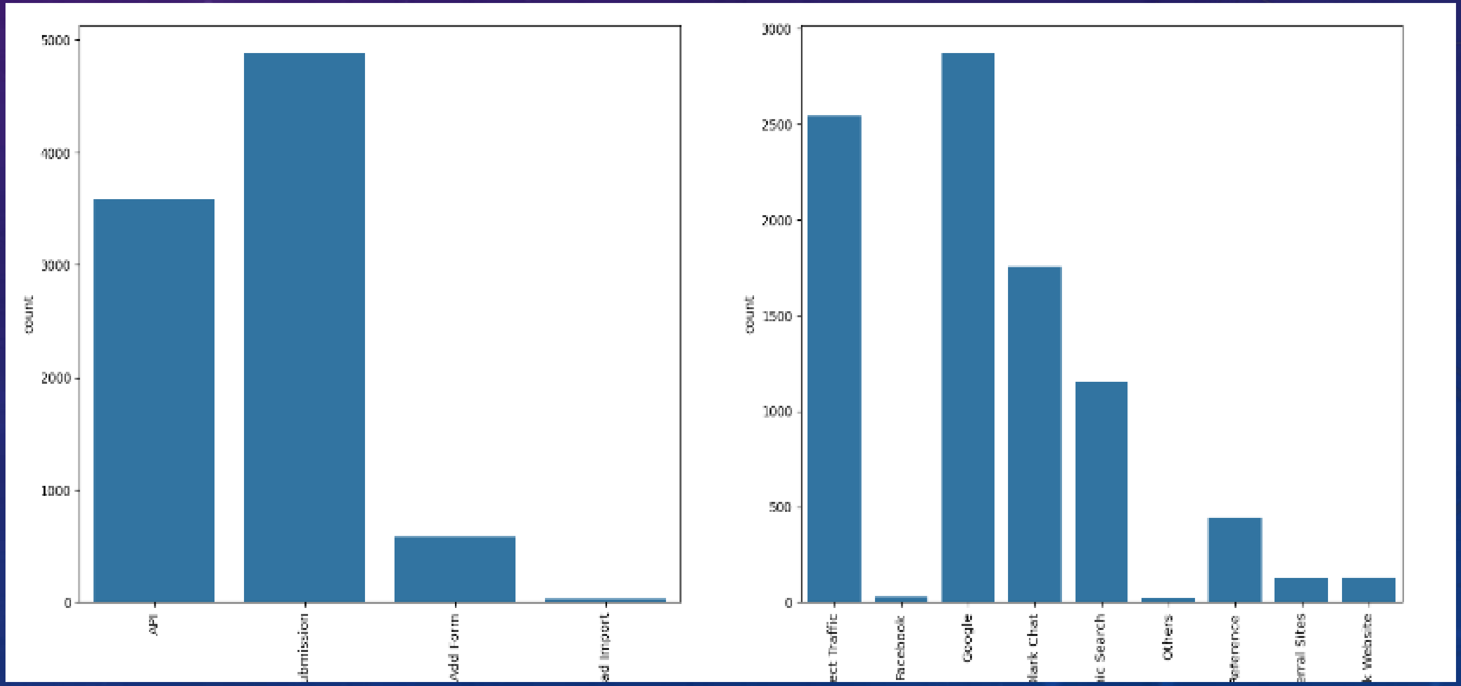
UNIVARIATE ANALYSIS

- For example Lead Origin, Lead Source columns



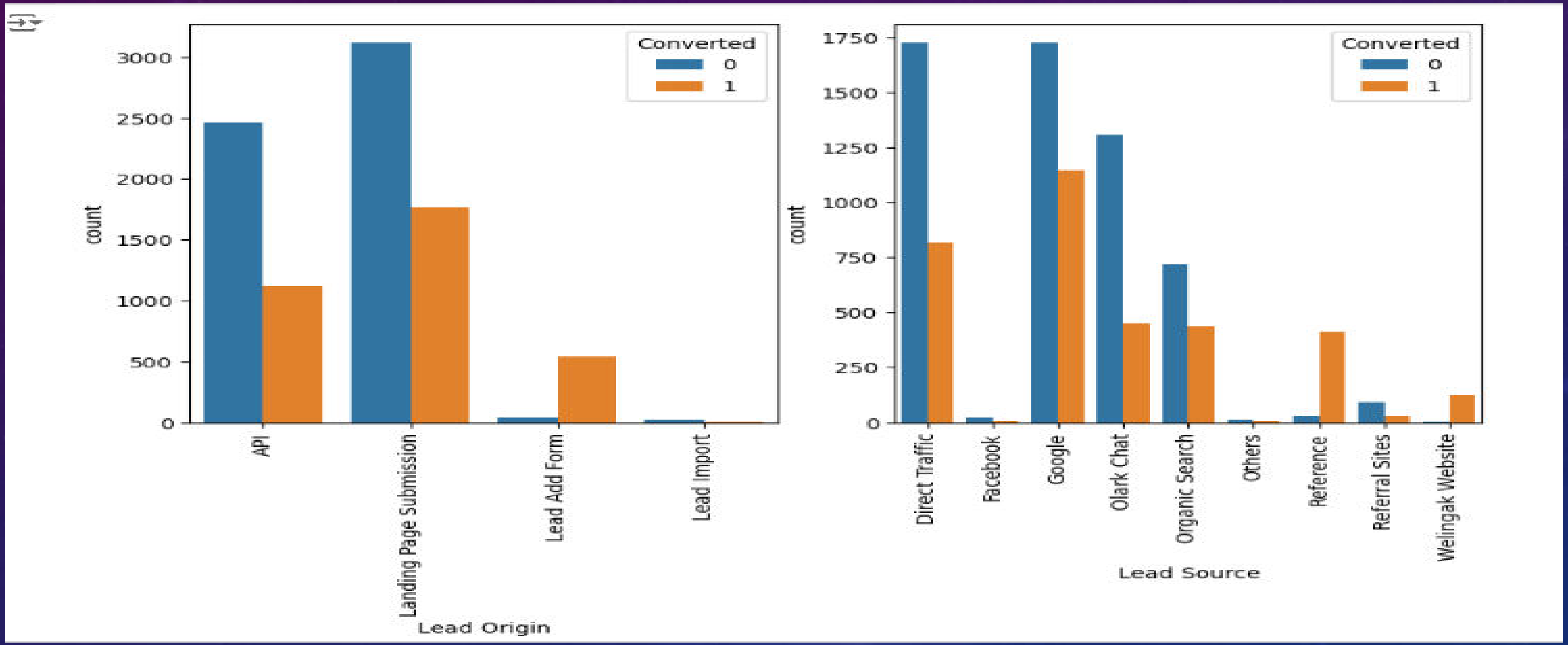
```
plt.subplot(3,3,1)
sns.barplot(leads['Lead Origin'].value_counts())
plt.xticks(rotation=90)

plt.subplot(3,3,2)
sns.barplot(leads['Lead Source'].value_counts())
plt.xticks(rotation=90)
```

- Columns with different counts.



BIVARIATE ANALYSIS

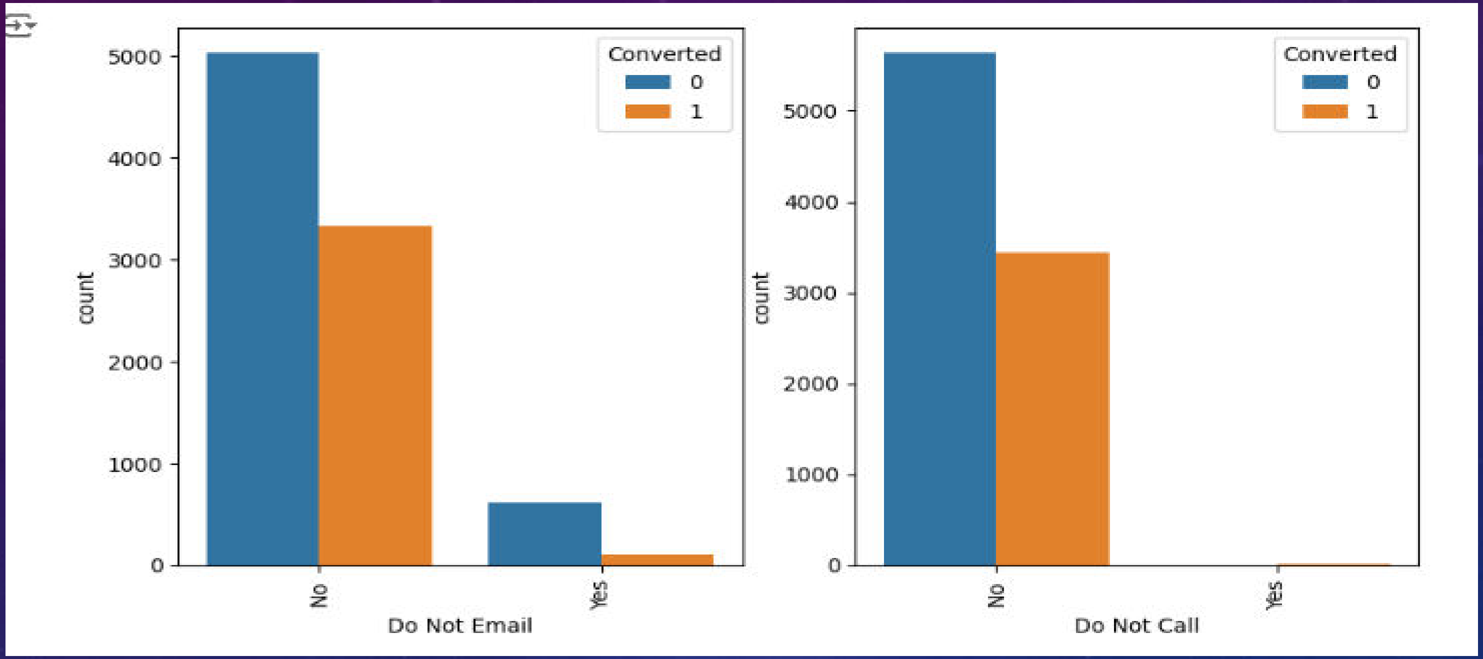
- Between Lead Origin and Converted ; Between Lead Source and Converted



- **Lead Origin:** Focus should be on improving lead conversion of API and Landing Page Submission origin and generate more leads from Lead Add Form.
- **Lead Source:** Focus should be on improving lead conversion of olark chat, organic search, direct traffic, and google leads and generate more leads from reference and welingak website.

BIVARIATE ANALYSIS

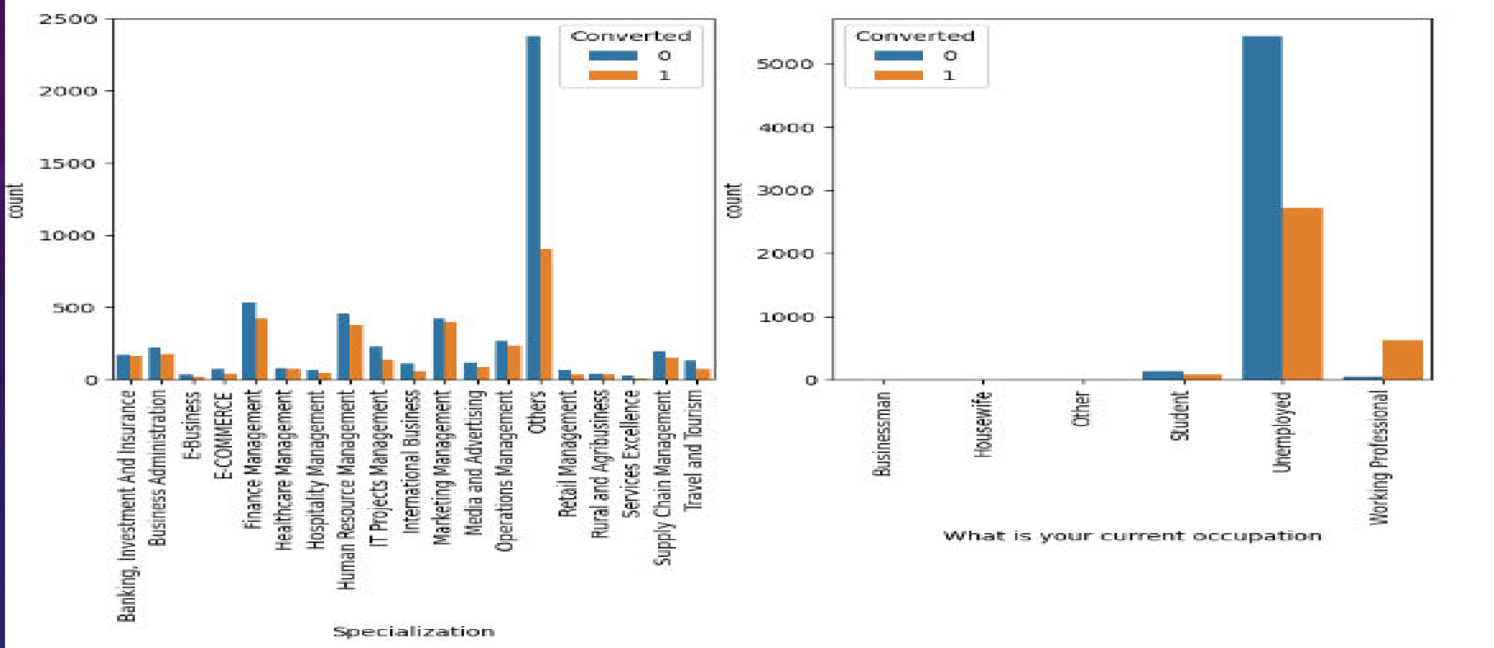
- Between Do Not Call and Converted ; Between Do Not Email and Converted



- **Do Not Email:** Most entries are 'No'. No Inference can be drawn with this parameter.
- **Do Not Call:** Most entries are 'No'. No Inference can be drawn with this parameter.

BIVARIATE ANALYSIS

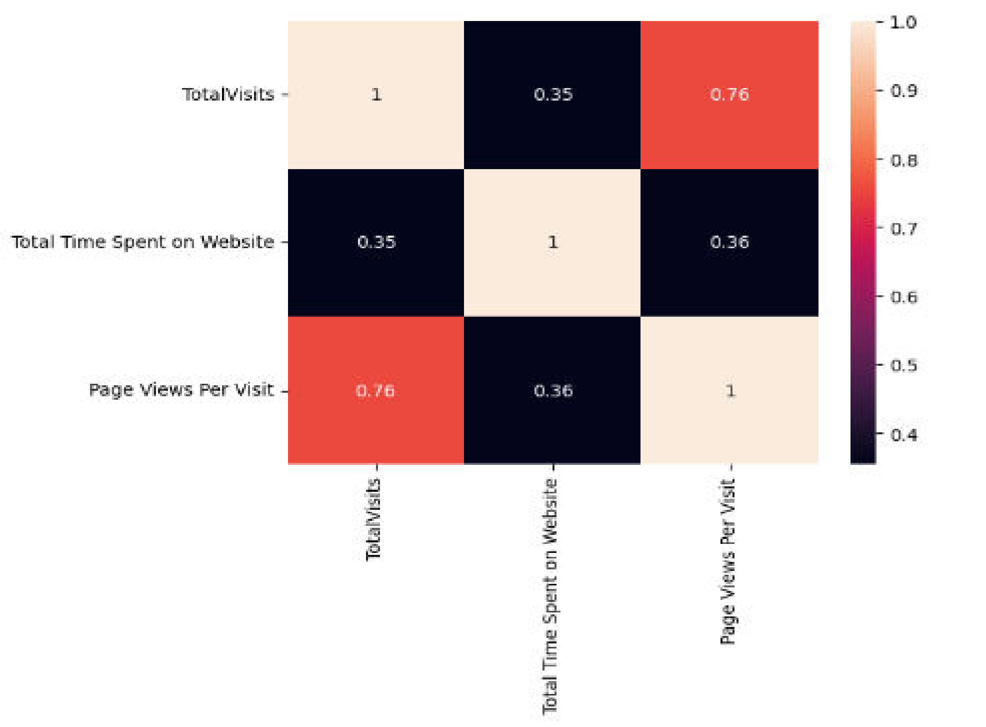
- Between Specialization and Converted ; Between What is your current occupation and Converted



- **Specialization:** We need to focus more on Others Specialization for leads conversion rate.
- **What is your current occupation:** Unemployed leads are the most in numbers for conversion rate. Employed leads are high in number converted.

MULTI-VARIATE ANALYSIS

- Heat map is plotted against the numerical features.



- **TotalVisits and Pages Views Per Visit are highly correlated to each other rather than others.**

DATA PREPARATION

1. Dummy variables creation:

- Convert Yes/No -- valued categorical feature into -- 1/0 (numerical values) for model building.
- Create the non-binary valued categorical feature into dummy variables.

By using the function -- pd.get_dummies(leads[col_list],drop_first = True, dtype = int)

2. Train-Test data splitting:

- We need to split the complete data set into train dataset and test dataset.

df_train, df_test = train_test_split(dataset, test_size=0.3, random_state=42)

- Here the train_size = 70% and test_size = 30%

3. Rescaling:

- Instantiate the object scalar, for using MinMaxScalar() function.
- Fit and transform the train dataset – scalar.fit_transform(train dataset)

MODEL BUILDING

1. We need to create the Dependent variables set (y) and Independent variables set(X) for train dataset.

- X_train = df_train.drop(['Prospect ID','Converted'],axis = 1)
- y_train = df_train['Converted’]

2. Create a logistic regression model

- log_reg = LogisticRegression()

3. By using RFE function we can eliminate unnecessary features from the train dataset.

- rfe = RFE(log_reg, n_features_to_select=20)
- rfe = rfe.fit(X_train, y_train)

4. Accumulate the supported columns by rfe.support_ attribute

- Store the columns in a list and use it for building a logistic regression model.

MODEL BUILDING

Create the model using GLM() and fit the model and print the summary.

- X = X_train[col]
- X_train_lm = sm.add_constant(X)
- logm1 = sm.GLM(y_train, X_train_lm, family = sm.families.Binomial())
- res1 = logm1.fit()
- print(res1.summary())
- Now check the p-values of all the 20 features. They all should be less than 0.05 – if not those columns should be dropped and re-build the model again.
- If all the p-values are less than 0.05, then check VIF for all the remaining columns and that value should be less than 5 -- if not those columns should be dropped and re-build the model again.
- This process is continued until the features are free from multi-collinearity.

MODEL EVALUATION

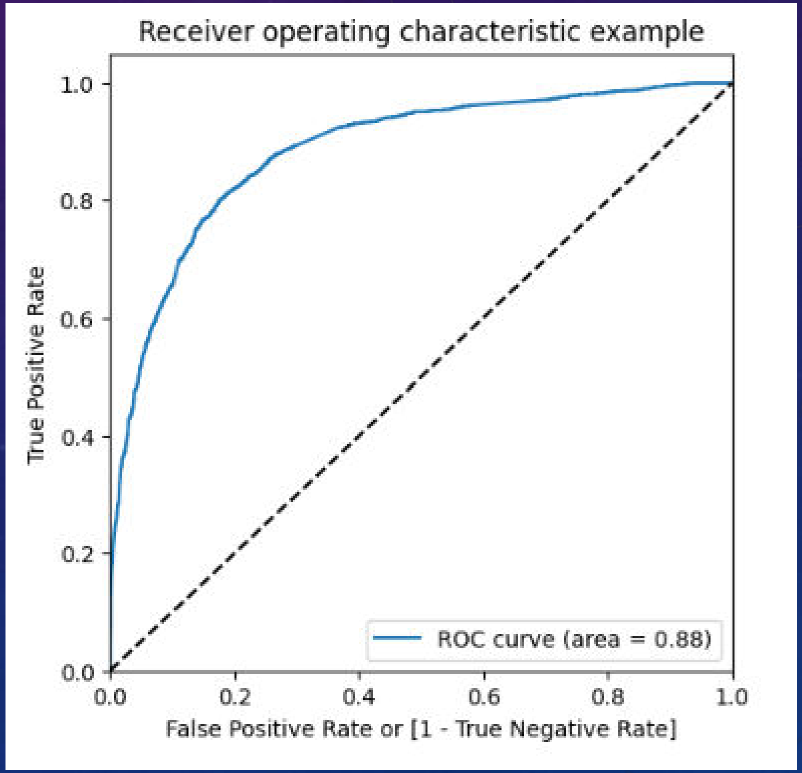
Metrics for train dataset:

- Accuracy of the Train dataset is 80.0% with 0.5 optimal cut-off value.
- With the current cut off as 0.5 we have sensitivity of around 65.5%.
- With the current cut off as 0.5 we have specificity of around 89.9%.

Understanding ROC Curve:

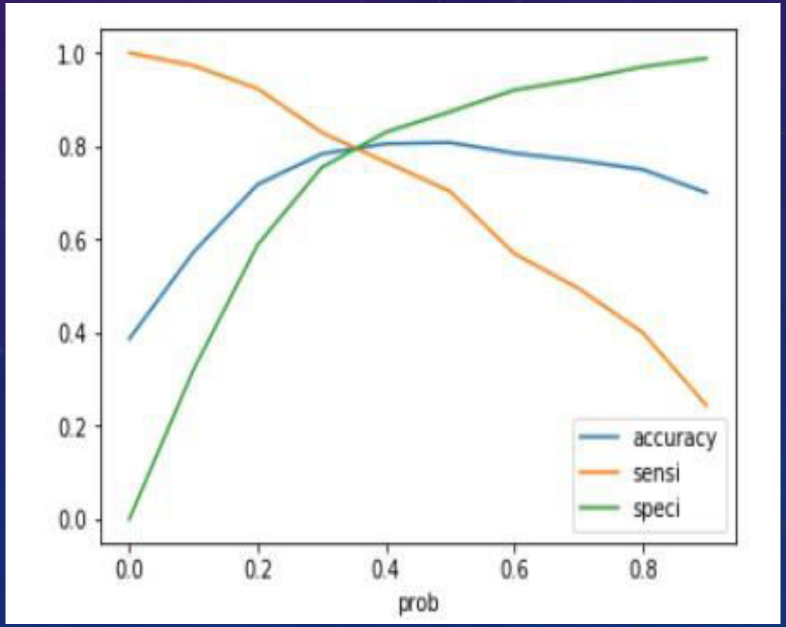
- Plot a ROC Curve to check the area covered between TPR and FPR.
- The area of the ROC curve is 0.88 which is a very good value because the value is nearer to 1.

We can use both the metrics Accuracy and Precision-Recall.



MODEL EVALUATION

- The point of intersection of accuracy, sensitivity and specificity gives the optimal cutoff.
- **The optimal cutoff is 0.35**



- After getting the optimal cutoff,

Metrics for train dataset against Accuracy:

- With the current cut off as 0.35 we have around 81.1% accuracy.
- With the current cut off as 0.35 we have sensitivity of around 80.8%.
- With the current cut off as 0.35 we have specificity of around 81.3%.

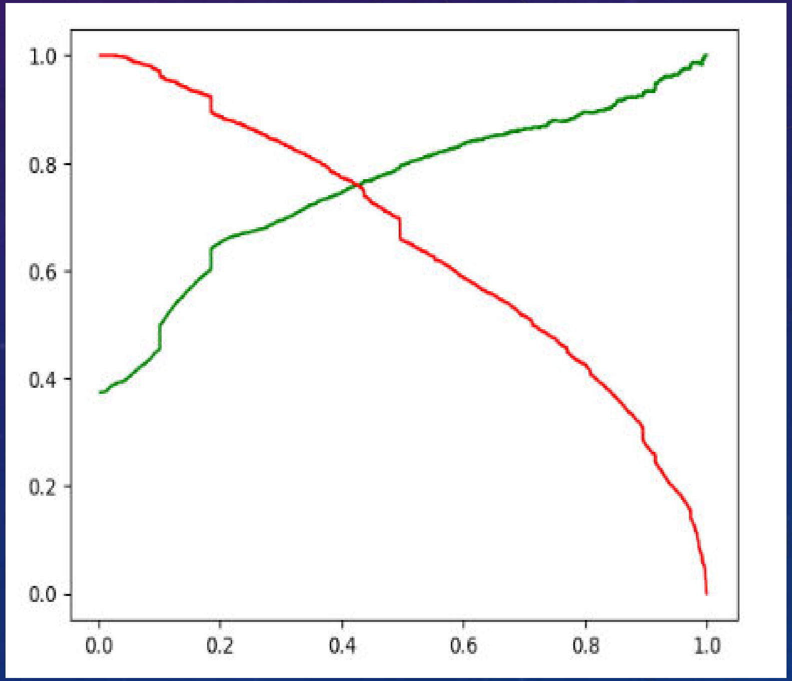
MODEL EVALUATION

- The point of intersection of precision and recall gives the optimal cutoff.
- **The optimal cutoff is 0.41**

- After getting the optimal cutoff,

Metrics for train dataset against Precision-Recall:

- With the current cut off as 0.41 we have around 81.1% accuracy.
- With the current cut off as 0.41 we have sensitivity of around 75.1%.
- With the current cut off as 0.41 we have specificity of around 76.7%.



PREDICTIONS ON TEST DATASET

1. We need to create the Dependent variables set (y) and Independent variables set(X) for test dataset.

- X_test = df_test.drop(['Prospect ID','Converted'],axis = 1)
- Y_test = df_test['Converted’]

2. Re-scaling the test dataset numerical columns.

3. Add a constant for Test dataset

4. Predict the values for test dataset with optimal cutoff value 0.35 against **Accuracy, Sensitivity and Specificity**.

5. Predict the values for test dataset with optimal cutoff value 0.41 against **Precision and Recall**.

PREDICTIONS ON TEST DATASET

Metrics for test dataset for against Accuracy:

- With the current cut off as 0.35 we have around 80.4% accuracy.
- With the current cut off as 0.35 we have sensitivity of around 80.1%.
- With the current cut off as 0.35 we have specificity of around 80.7%.

These values are equal to trained dataset values with cutoff 0.35

Metrics for test dataset against Precision-Recall:

- With the current cut off as 0.41 we have around 81.1% accuracy.
- With the current cut off as 0.41 we have Precision of around 75.4%.
- With the current cut off as 0.41 we have Recall of around 76.4%.

These values are equal to trained dataset values with cutoff 0.41

CONCLUSION

From the metrics, we can see that the trained dataset and test dataset are having the similar accuracy rates, sensitivity and specificity also Precision-Recall values. So we can say that this logistic regression model is good to go.

Key factors influencing potential buyers (in descending order):

1. Total time spent on the website
2. Total number of visits
3. Lead Source: Google, Direct traffic, Organic search
4. Last Activity: SMS
5. Lead Origin: API, Lead ad format
6. Current Occupation: Working professional

With these insights, X Education can significantly increase the likelihood of converting potential buyers into customers.

THANK YOU