Lead Scoring Case Study

ANJU ABINASH

AMAN RAWAT

Problem Statement

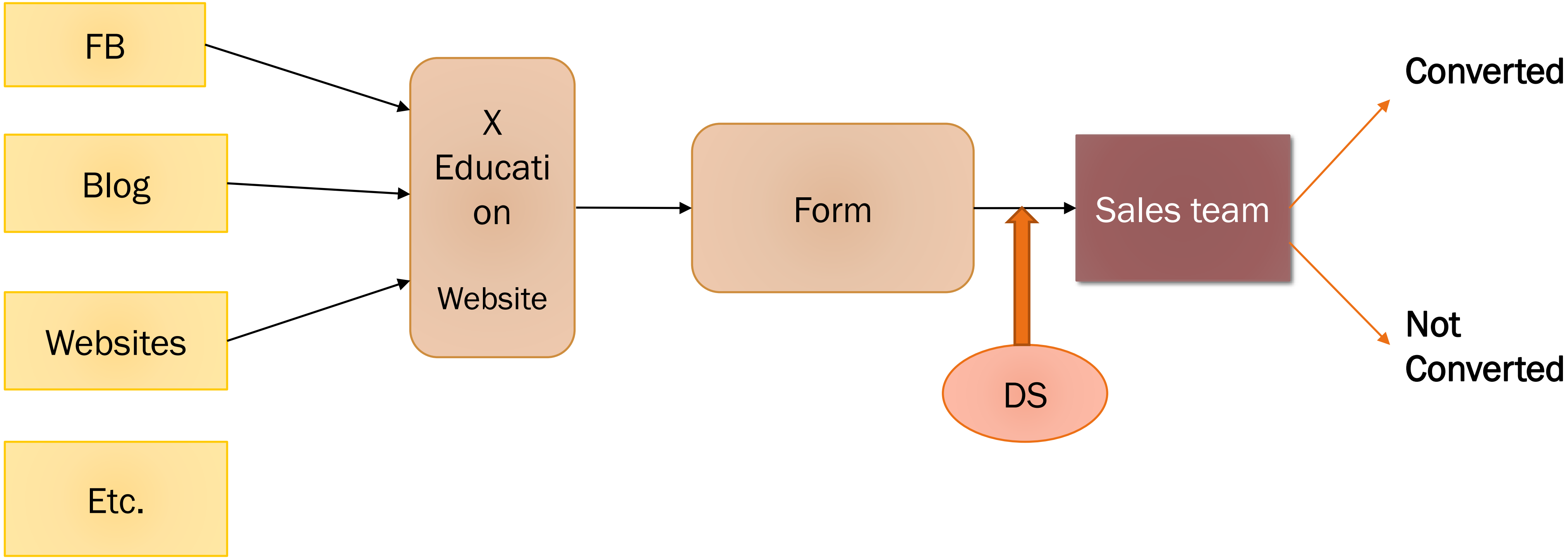
X Education sells online courses to industry professionals. X Education needs help in selecting the most promising leads, i.e., the leads that are most likely to convert into paying customers.

The company needs a model wherein a lead score is assigned to each of the leads such that the customers with higher lead scores have a higher conversion chance and the customers with lower lead scores have a lower conversion chance.

The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%.

Problem statement Flow chart

This is considered the current model. Data analysis is happening once the form has been done and before the sales term work on it.



Steps that follow:

1. **Import Libraries and dataset**

2. **Data sanity checks**

3. **Data cleaning**
   - Handle the “Select” value in the categorical variables as a Null value.
   - Drop the column which carries more than 45% of the missing value
   - Check the unique value of each categorical column, if the unique value rate is high we don’t really require that, and drop that columns
   - Bucketing has been done for some categorical columns in which the unique value percentage is very less.
   - For the columns with less percentage of missing value we can impute the value with the median for categorical columns.
   - Check the final percentage of rows retaining the data cleaning process.

4. EDA

Visualize the univariant and bivariant analysis
Outlier treatment has been done with the help of box plots.

---

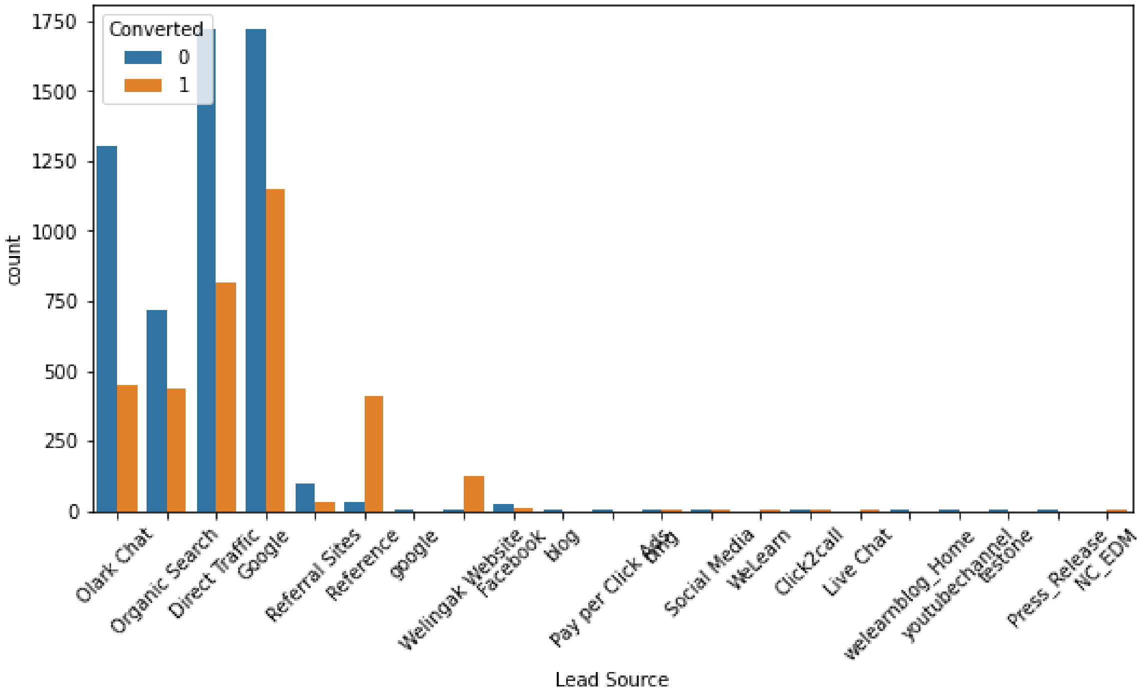
5. Data Preparation

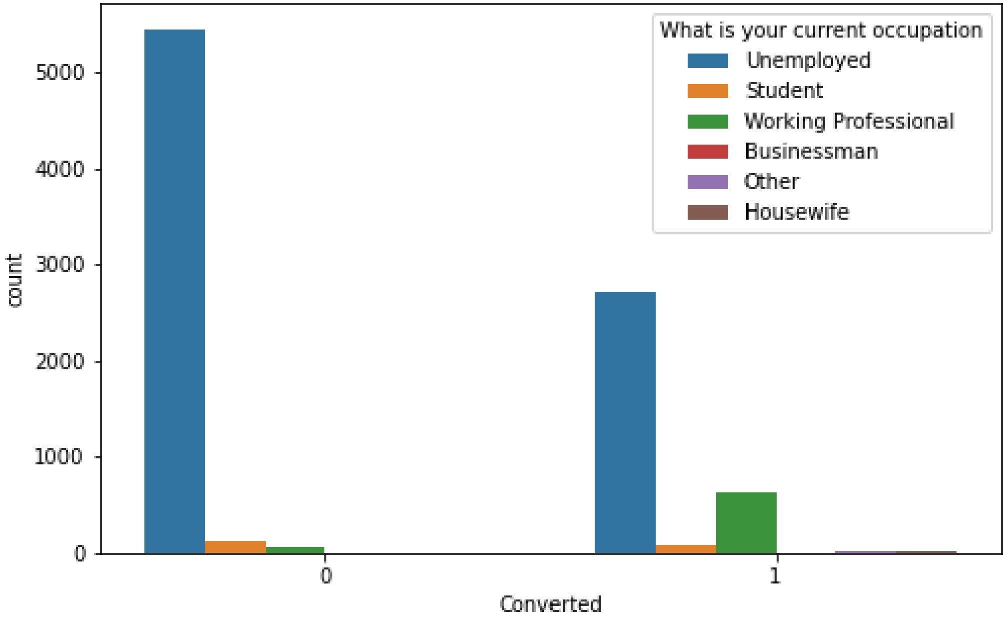
a. Create Dummy variables for all categorical variables
b. Perform Train-Test split
c. Perform scaling

6. Modelling

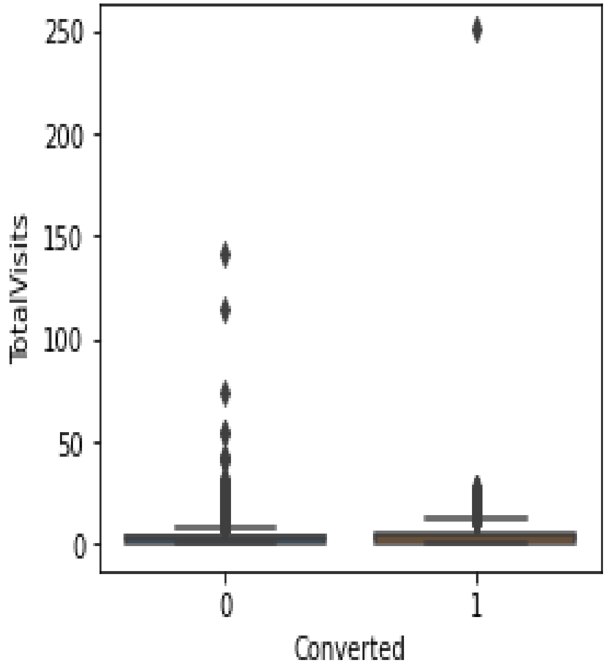
a. Use techniques like RFE to perform variable selection
b. Built Logistic regression
c. Check p-value and VIF
d. Check the ROC curve
e. Find the optimal cut-off
f. Check the model performance with a confusion matrix, Sensitivity, Recall, F1 score, etc.
g. Generate the score variable

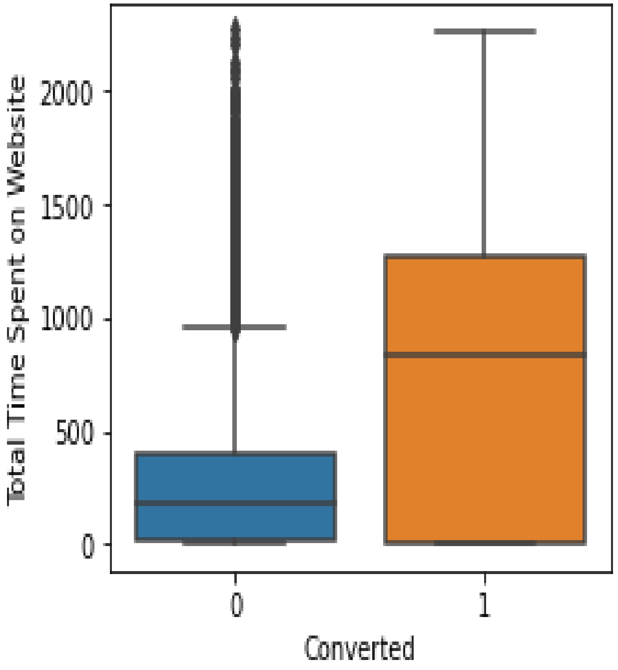
EDA: Exploratory Data Analysis
Univariate Analysis

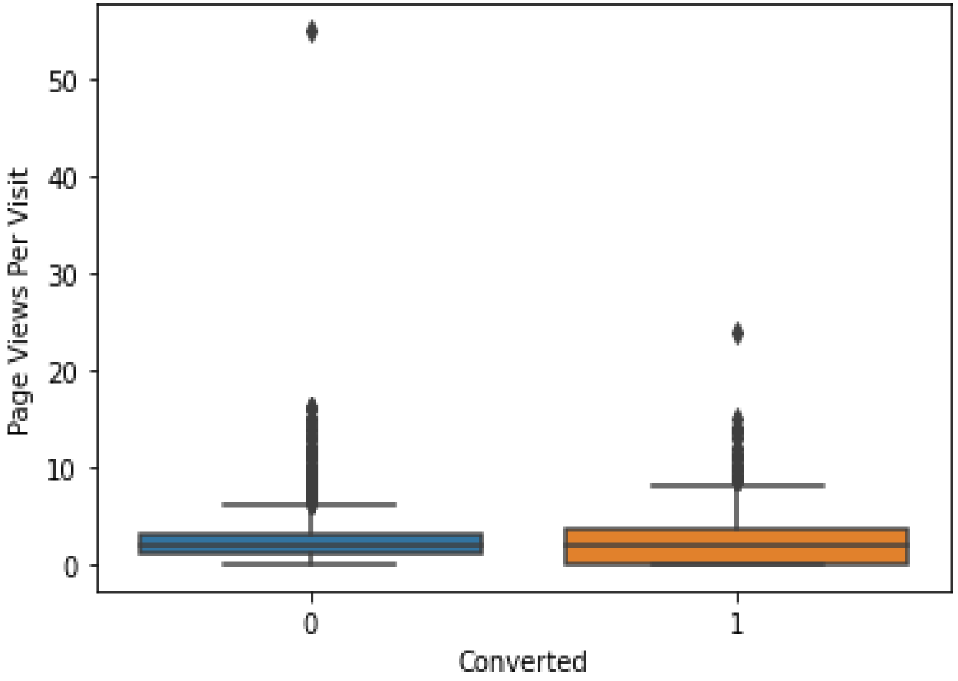




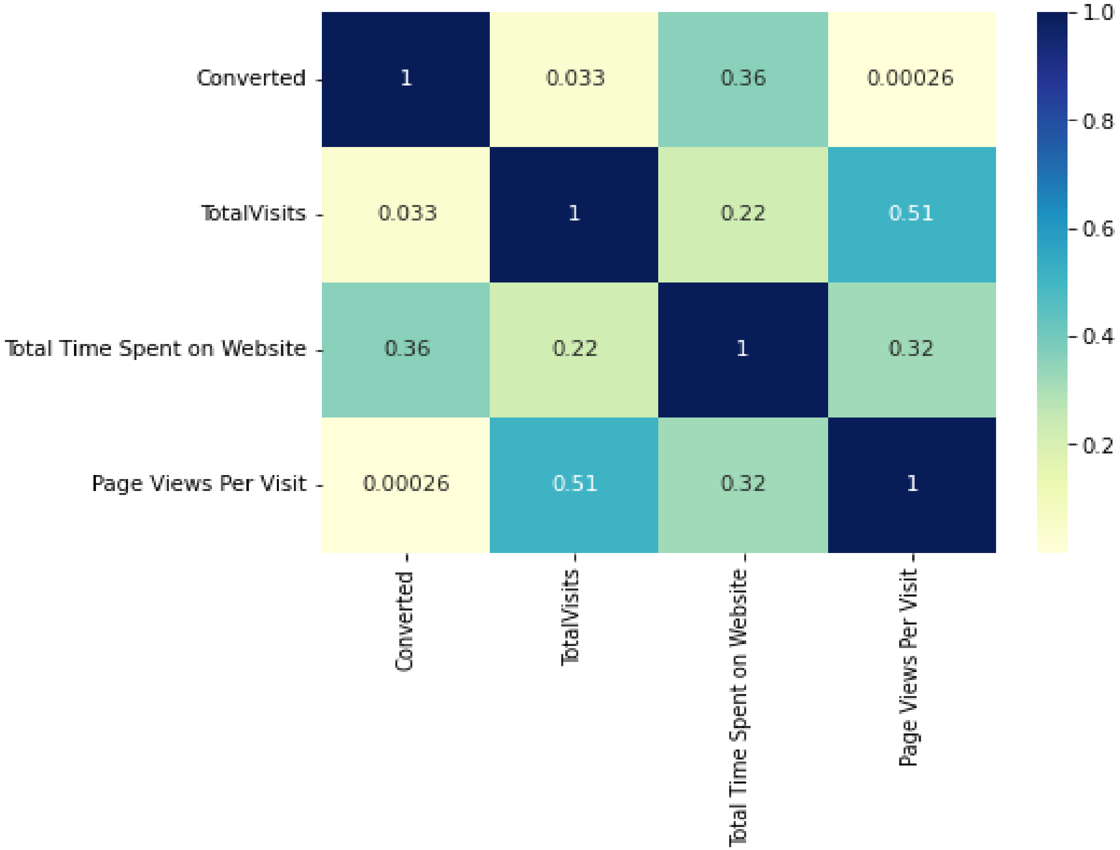
Bivariate Analysis







Correlation Heat map

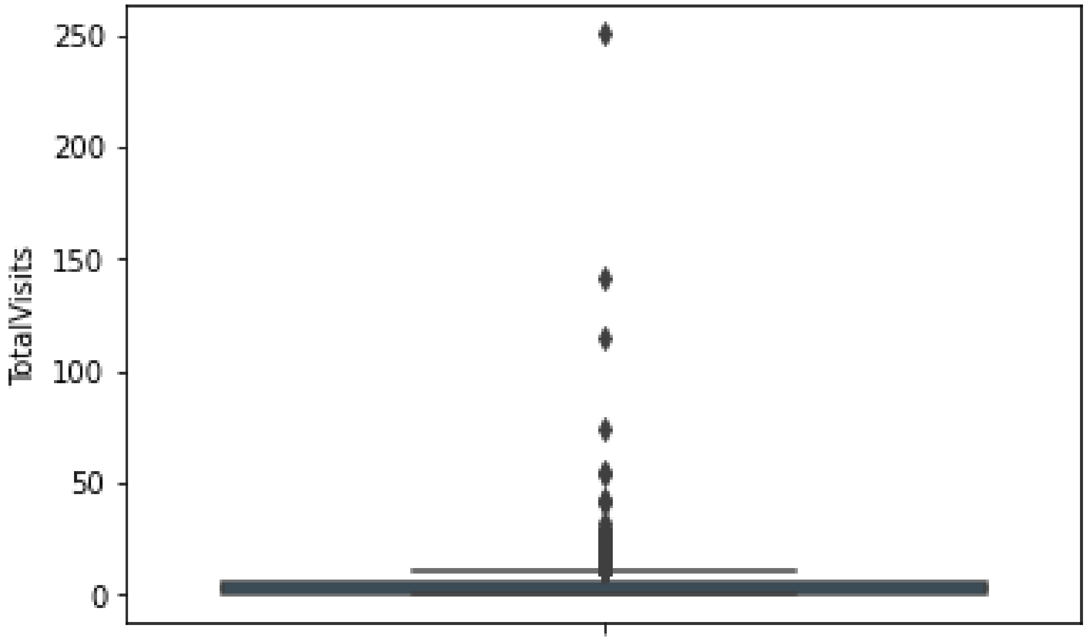


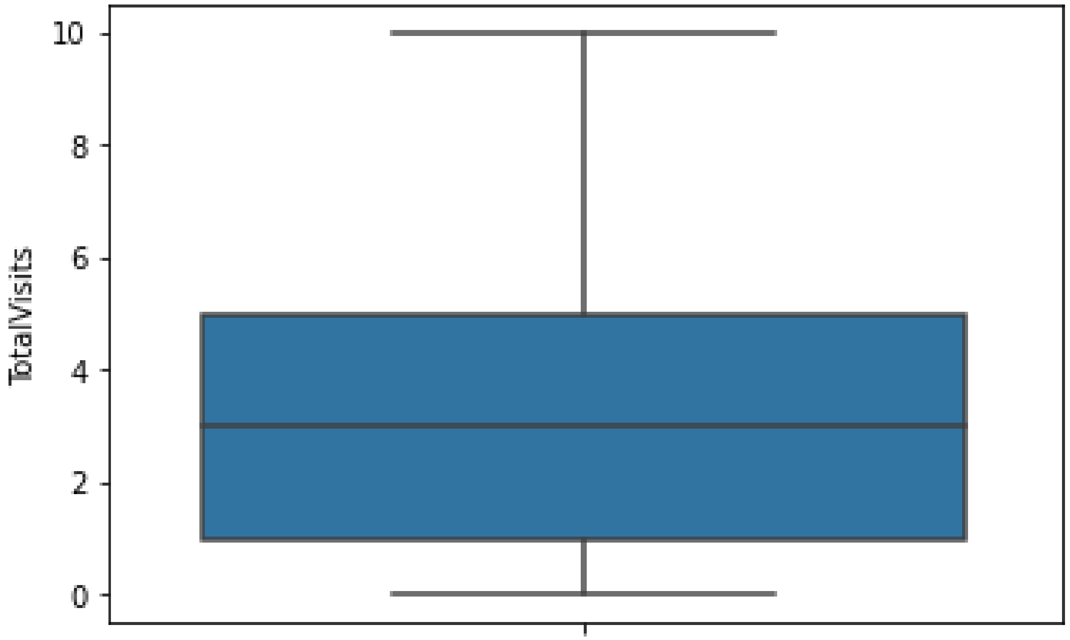
This is the heat map with the targeted variable (Converted) and the continuous predicted variables

All the variables are moderately correlated.

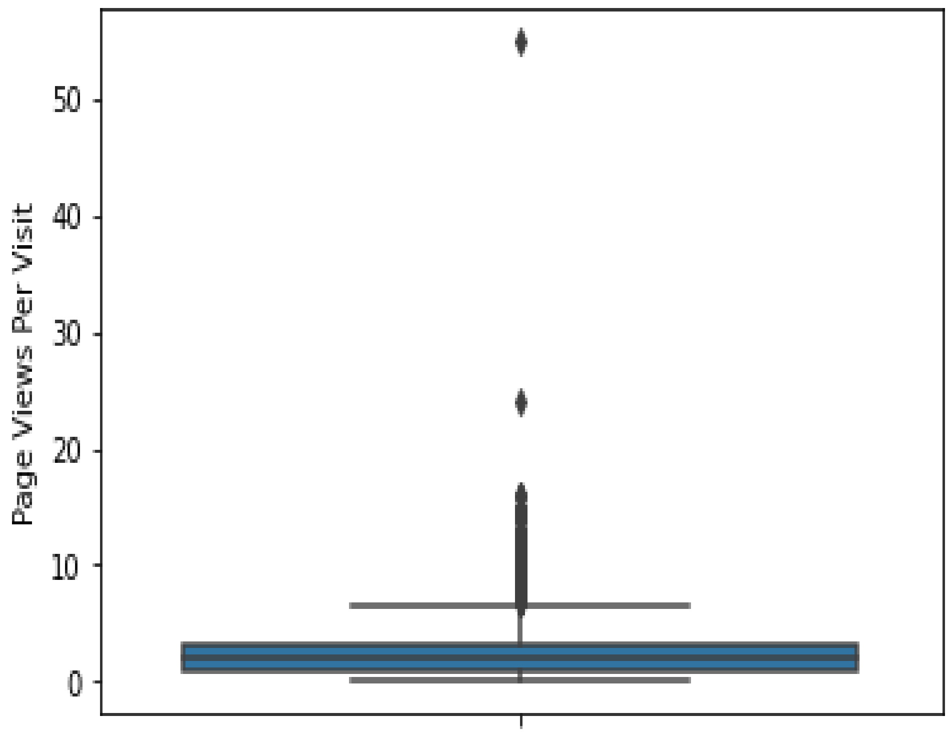
Outlier Treatment

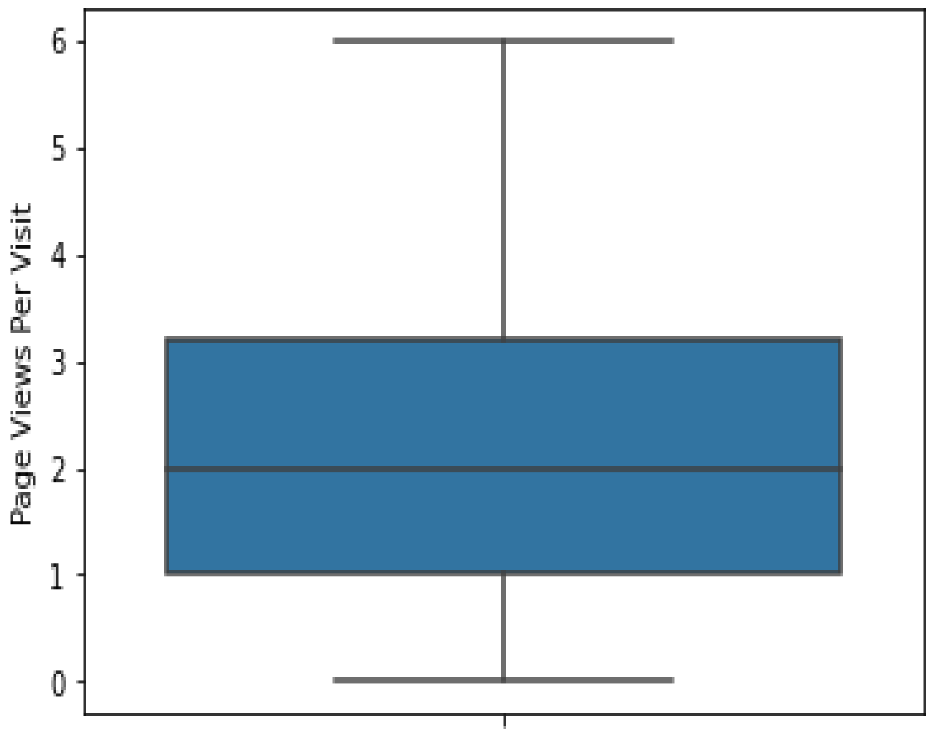
Visualization of “TotalVisits” before and after outlier treatment





Box plot of "Page Views Per Visit" before and after Outlier treatment





Model building and VIF code

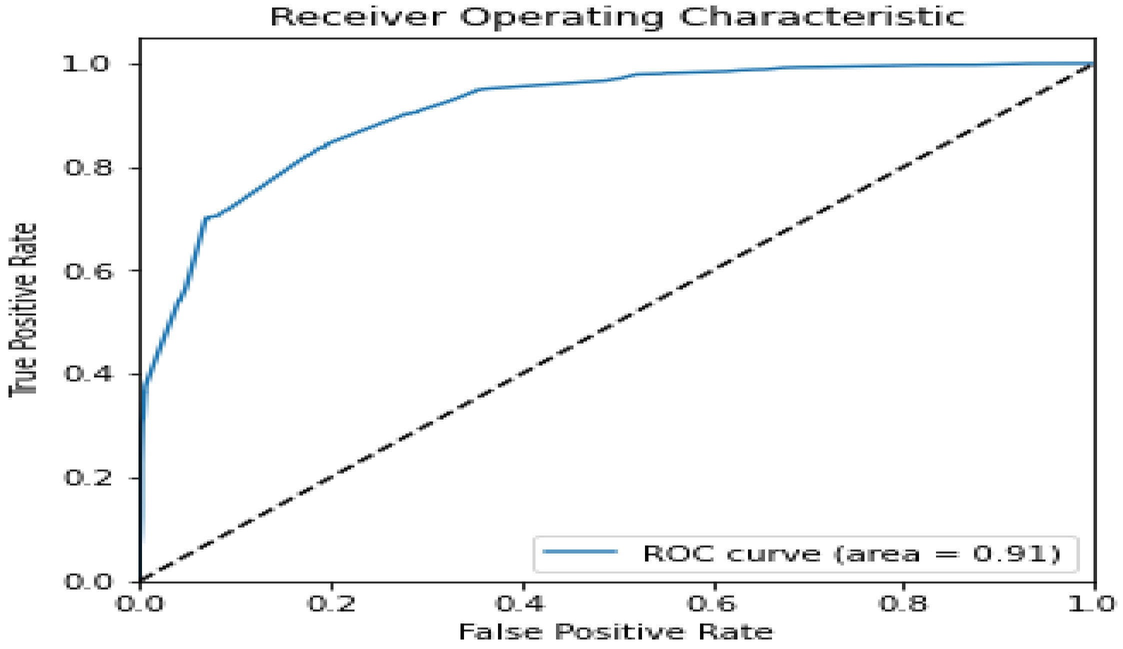
```
x_train_sm = sm.add_constant(x_train[col])

logm = sm.GLM(y_train,x_train_sm, family = sm.families.Binomial())

res = logm.fit()

res.summary()
```

```
def calculate_vif(x_train):

    vif_df = pd.DataFrame()

    vif_df['Features'] = x_train.columns

    vif_df['Variance Inflation Factor’] = [variance_inflation_factor(x_train.values, i) for i in range(x_train.shape[1]) ]

vif_df['Variance Inflation Factor'] = round(vif_df['Variance Inflation Factor'], 2)

vif_df = vif_df.sort_values(by = 'Variance Inflation Factor', ascending = False)

print(vif_df)

calculate_vif(x_train[col])
```

Plotting the ROC Curve

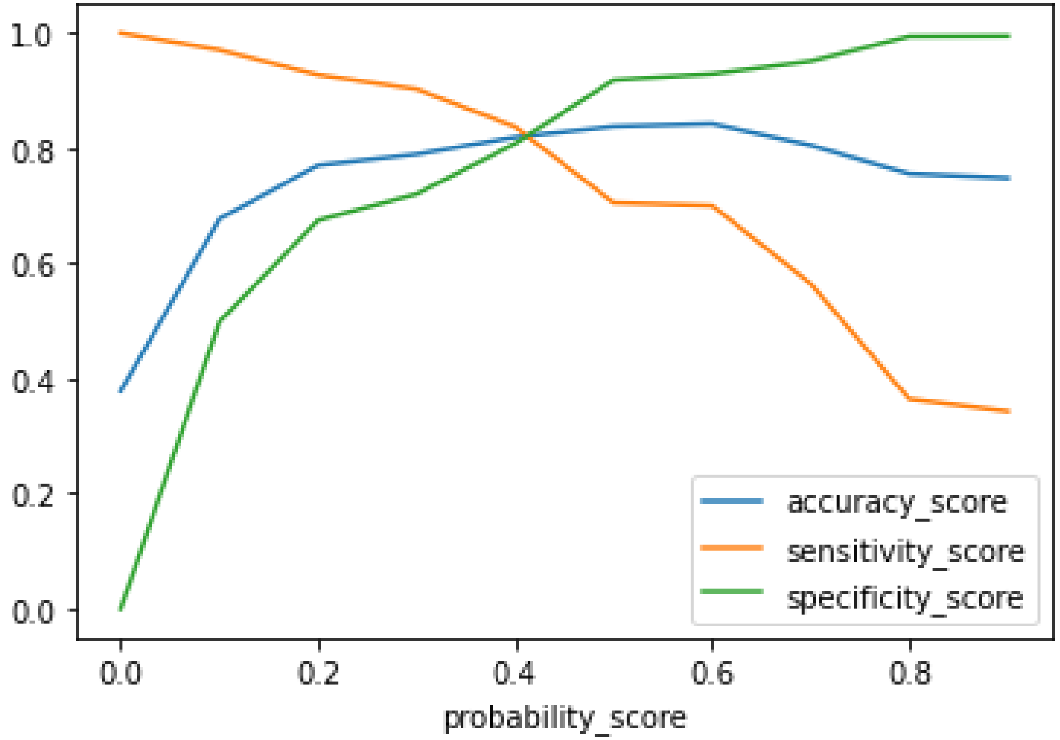
ROC provides a simple way to summarize the information related to different thresholds and resulting True Positive Rate and False Positive Rate values.

Best positive threshold value will get with a maximum area under the curve.

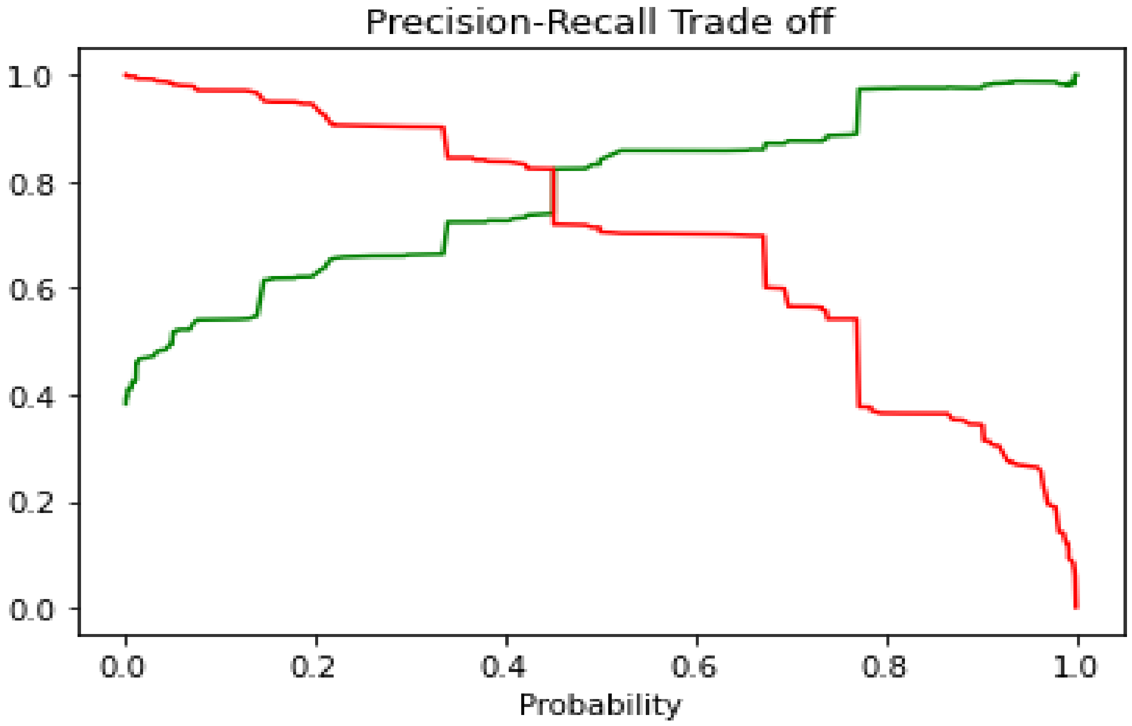


Finding the optimal value of the cutoff

Intersection point at which there is a balance between sensitivity and specificity; it corresponds to the optimal cutoff on logistic regression probabilities.
In this, we have chosen 0.4 as the optimal cutoff value.



Precision -Recall Trade off



Conclusion

1. The logistic regression model is used to predict the probability conversion of a customer.
2. Optimum cut-off is chosen to be 0.4
3. Our final Logistic Regression Model is built with 13 features.
4. Final model Sensitivity of train and test: 84%
5. Final mode Specificity of train and test : 81%
6. Final model Accuracy of train and test: 82%
7. Final mode Precision of train and test: 73%