LEAD SCORE CASE STUDY

By:
Anki Singha
Shashi Kumar S
Koushal Choudhary

CONTENTS

- Problem Statement
- Approach to Model Preparation
- Data Cleaning and Imputation
- Exploratory Data Analysis
- Selection of Dummy Variables
- Splitting into Training and Testing Sets
- Building the Model
- Evaluation of the Model – Specificity, Sensitivity, Precision, Recall

Problem Statement

- X Education offers online courses to its customer base.
- The company aims to boost leads for course enrolment.
- Streamlining lead identification, the focus is on potential hot leads.
- The objective is to selectively call high-potential leads, saving time for other productive tasks.

Approach to Model Preparation

- Data Cleaning, Imputing, and Understanding Data Variables
- Addressing null values and 'Select,' finding solutions for such values
- Checking for outliers in the data
- Exploratory Data Analysis
- Creation of Dummy Variables for Categorical Variables
- Scaling of Numerical Variables
- Logistic Regression Model
- Model Evaluation using Confusion Matrix, Precision, Recall, Specificity, etc.

Data Cleaning & Imputation

- Removed columns like 'City,' 'Country,' 'Prospect Id,' and 'Lead number' as they do not contribute significantly to the analysis.
- Excluded features with 'Asymmetric' characteristics due to their high null values (more than 50%).
- Reduced data by eliminating rows containing 'Select' values in columns like 'Specialization.'

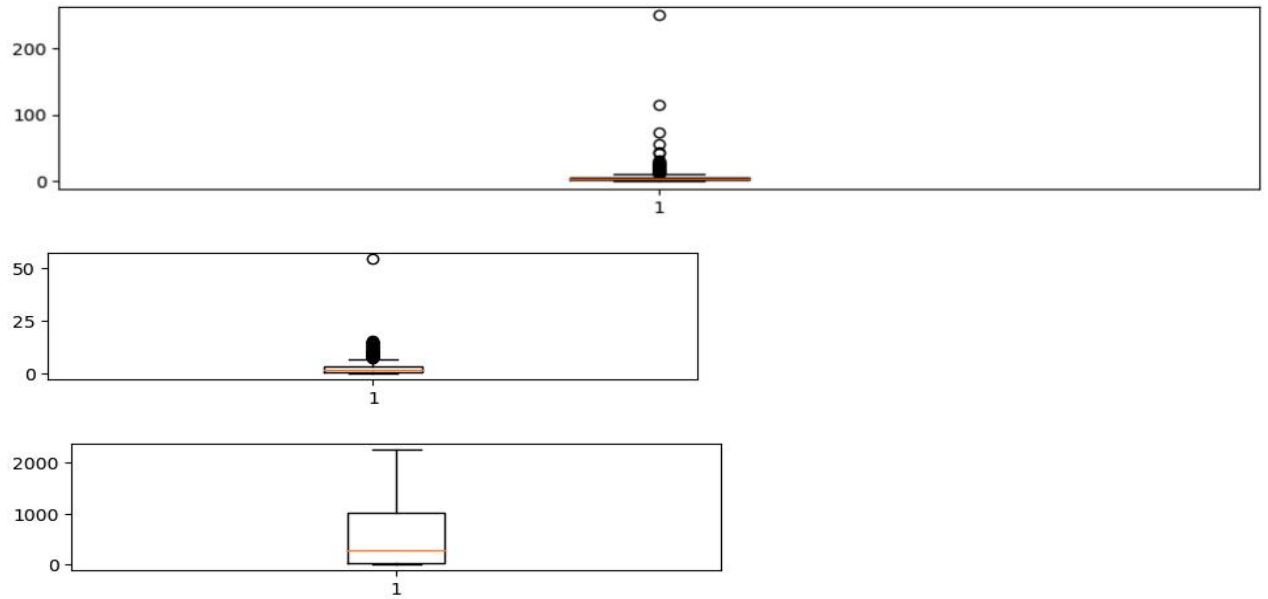
EDA

In [271]:
```
#Outliers check
plt.figure(figsize = (12,7))

plt.subplot(3,1,1)
plt.boxplot(x = 'TotalVisits', data = leadsdf)
plt.show()

plt.subplot(3,1,2)
plt.boxplot(x = 'Page Views Per Visit', data = leadsdf)
plt.show()

plt.subplot(3,1,3)
plt.boxplot(x = 'Total Time Spent on Website', data = leadsdf)
plt.show()
```



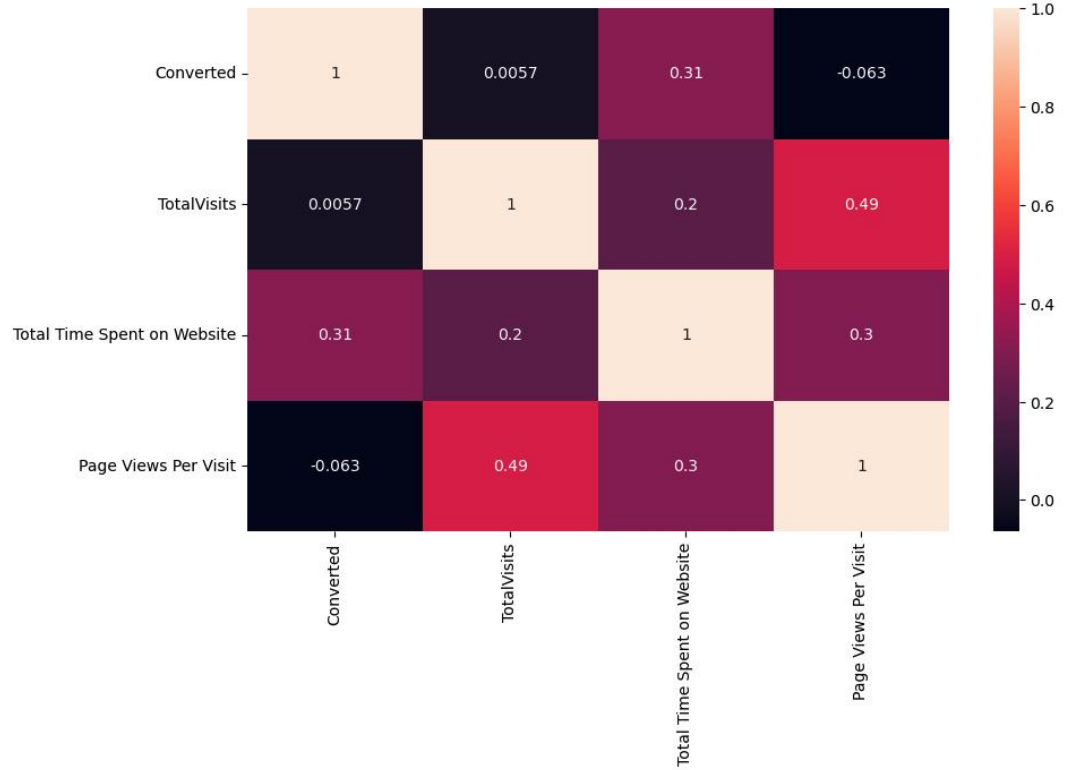
Bivariate Analysis

EDA

Correlation between ‘Converted’ & ‘Total time spent on Website’

Multivariate Analysis

In [274]:
```
#Checking the correlation among varibles
plt.figure(figsize=(10,6))
sns.heatmap(leadsdf.corr(),annot = True)
plt.show()
```



Dummy Variable Selection

The categorical variables considered for creating dummy variables include:

- Lead Origin
- Lead Source
- Do Not Email
- Last Activity
- Specialization
- What is your current occupation
- A free copy of Mastering The Interview
- Last Notable Activity

Splitting Train Test Set

The data is split in the ratio of 70 (Train) to 30 (test)

```
In [463]: ## Splitting the dataset into 70% train data and 30% test data

          X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3, random_state=100)

In [464]: X_train.shape
Out[464]: (4461, 74)

In [466]: y_train.shape
Out[466]: (4461,)

In [467]: X_test.shape
Out[467]: (1912, 74)

In [468]: y_test.shape
Out[468]: (1912,)
```

Building The Model

- Model is build using Logistic Regression classification technique
- Columns are eliminated using Recursive Feature Elimination (RFE)
- Numerical Variables are scaled using MinMaxScaler
- Variance Inflation Factor and p-values are considered for further manual elimination of the columns
- Max limit for VIF is 5 and for p-value is 0.005
- Separate individual function for logistic model and Variance inflation Factor are written for the reusability

Evaluating The Model

Measure used to evaluate the model:

- Confusion Matrix

```
In [312]: ## Creating confusion matrix
          conf_matrix = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Predicted )
          print(conf_matrix)

          [[1929  383]
           [ 560 1589]]

In [313]: ## Accuracy
          print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Predicted))

          0.7886124187401928
```

Accuracy : ~ 78%
Sensitivity : ~ 74%
Specificity : ~ 83%

```
In [313]: ## Accuracy
          print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Predicted))

          0.7886124187401928

In [314]: TP = conf_matrix[1,1] # true positive
          TN = conf_matrix[0,0] # true negatives
          FP = conf_matrix[0,1] # false positives
          FN = conf_matrix[1,0] # false negatives

In [315]: ## Sensitivity
          TP/(TP+FN)

Out[315]: 0.739413680781759

In [316]: ## Specificity
          TN/(TN+FP)

Out[316]: 0.8343425605536332
```

Precision >> achieved ~ 77%
Recall >> achieved ~ 79%

```
In [361]: ## Precision
          TP/(TP+FP)

Out[361]: 0.7771194165907019

In [362]: ## Recall

          TP/(TP+FN)

Out[362]: 0.793392275476966
```

Cutoff 0.44 is optimal one as recall is almost 80% which is fullfilling the company's target.