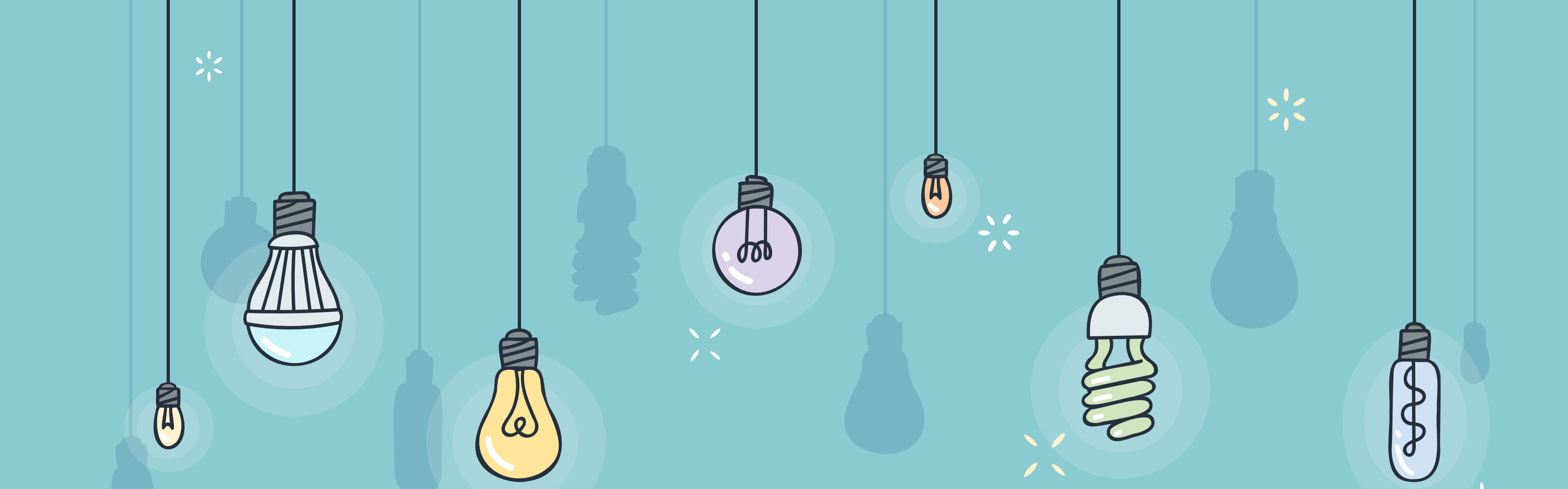


Lead Scoring Case study

Presented By :-

Rashmita Karak

Mouli Krishna Reddy Dantu

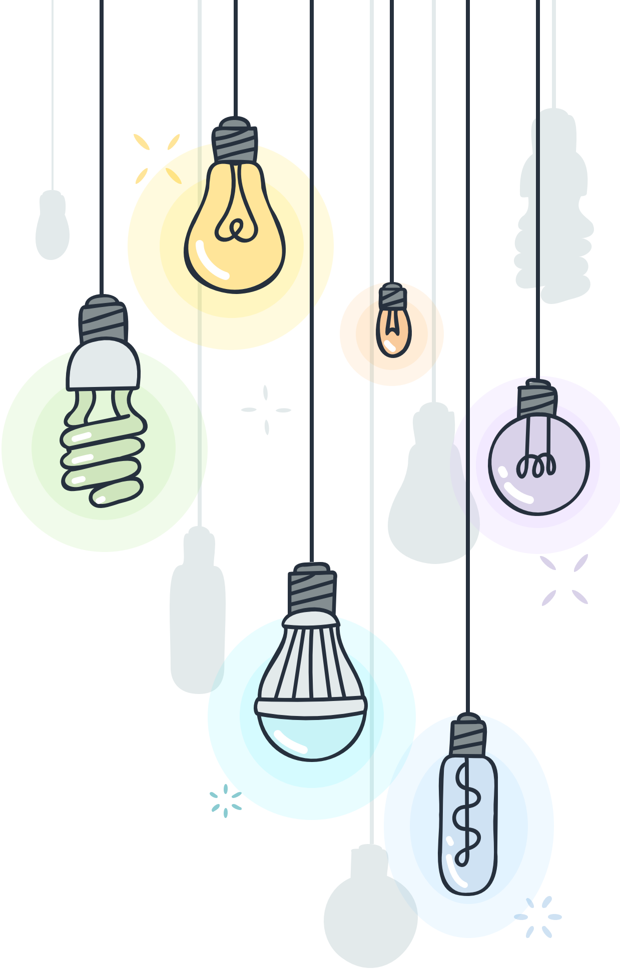
DS28 batch - Upgrad

Problem statement

X Education Company sells online courses to industry professionals.

Although there are numerous leads, conversion rate is only 30%.

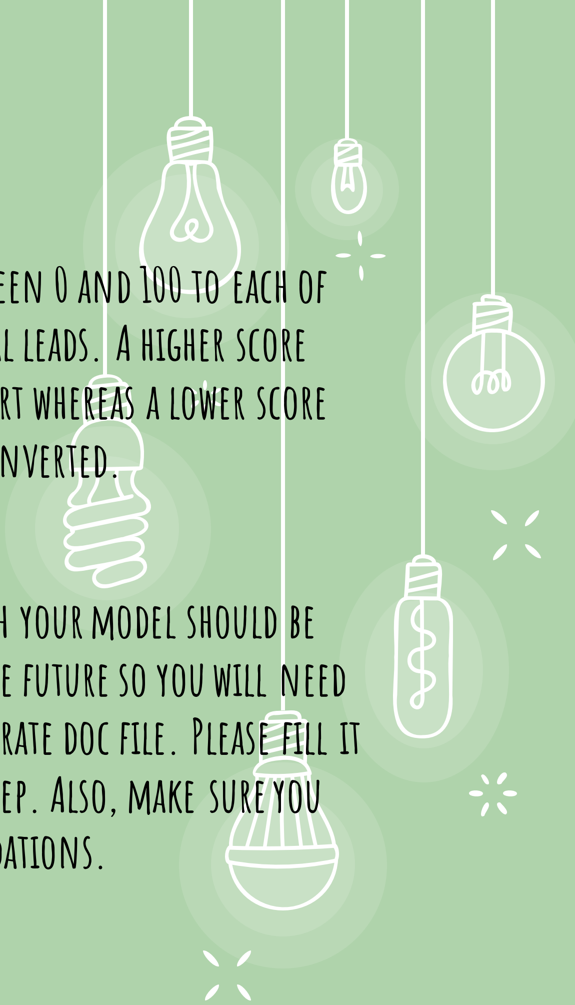
Ceo of the company wants an increase in the conversion rate to 80%.



Goal of the case study

- 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.

- There are some more problems presented by the company which your model should be able to adjust to if the company's requirement changes in the future so you will need to handle these as well. These problems are provided in a separate doc file. Please fill it based on the logistic regression model you got in the first step. Also, make sure you include this in your final PPT where you'll make recommendations.



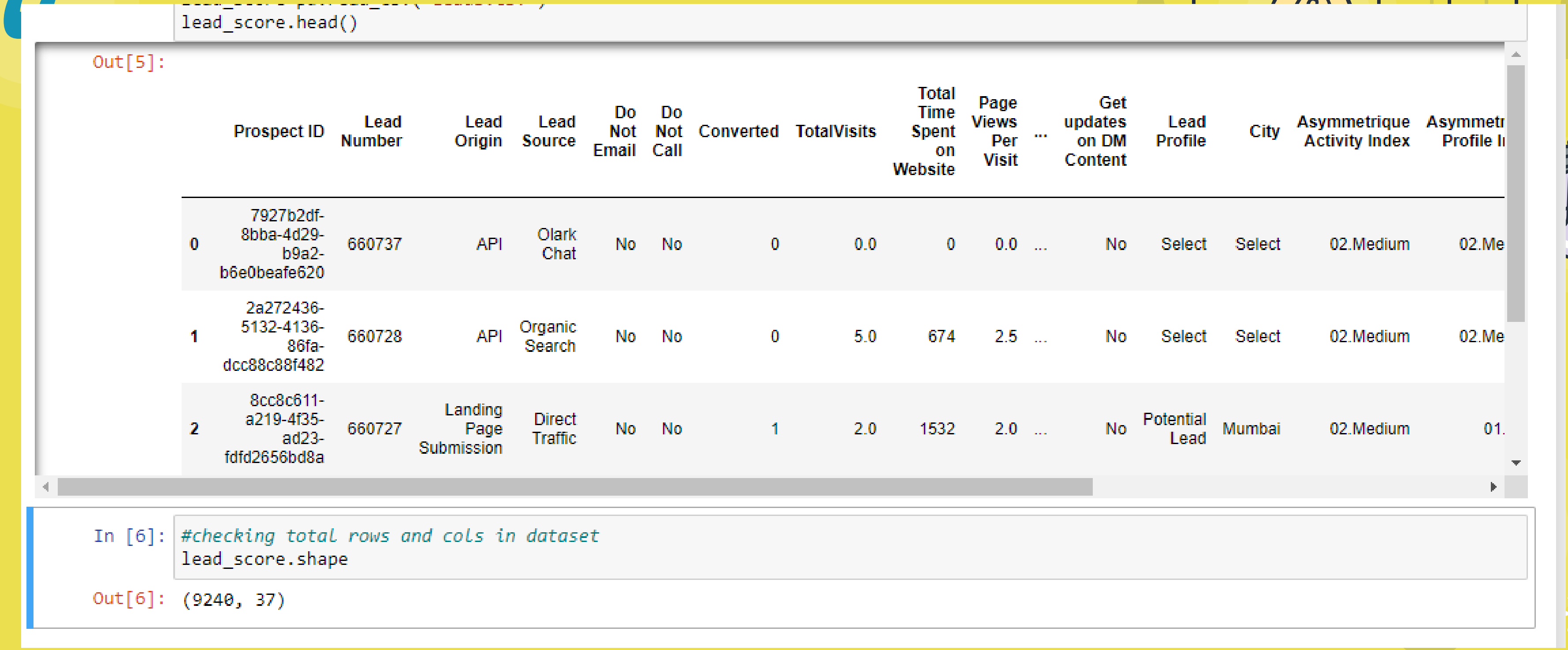
Analysis Approach

During the entire process , we follow the below steps in an detailed way : -

- Understand the data – Where we check the basic information about the dataset.
- Clean the data - Removing of the unwanted and unimportant features so that

  It is easier to analyse the dependent variables.
- Visualisation of the dataset – Using different plots we check the dependency of all the relevant categories with the target variable.
- Model building and evaluation - consiting of model building and checking if the built model is accurate.
- Conclusion from analysis - final predictions from the analysis hence evaluating which features have most effect on the target variable.

Dataset Description



```
lead_score= pd.read_csv("Leads.csv")
lead_score.head()
```

Out[5]:

	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	...	Get updates on DM Content	Lead Profile	City	Asymmetrique Activity Index	Asymmetri Profile I
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No	Select	Select	02.Medium	02.Me
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No	Select	Select	02.Medium	02.Me
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	...	No	Potential Lead	Mumbai	02.Medium	01.

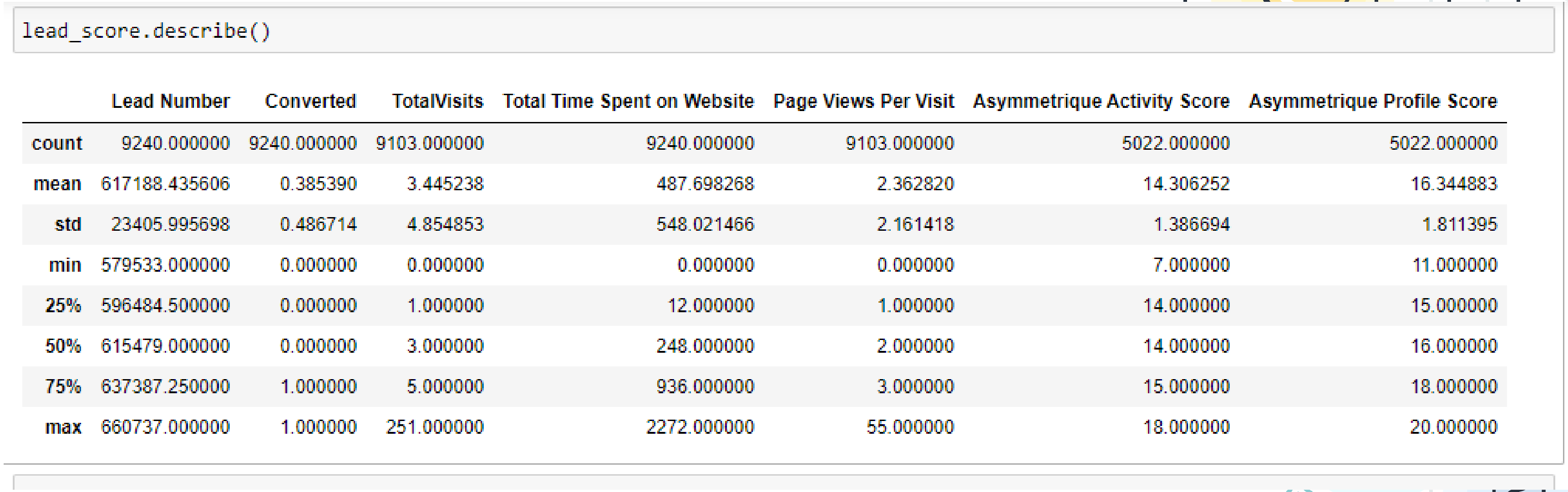
In [6]:

```
#checking total rows and cols in dataset
lead_score.shape
```

Out[6]:

```
(9240, 37)
```

DESCRIPTION OF THE DATASET



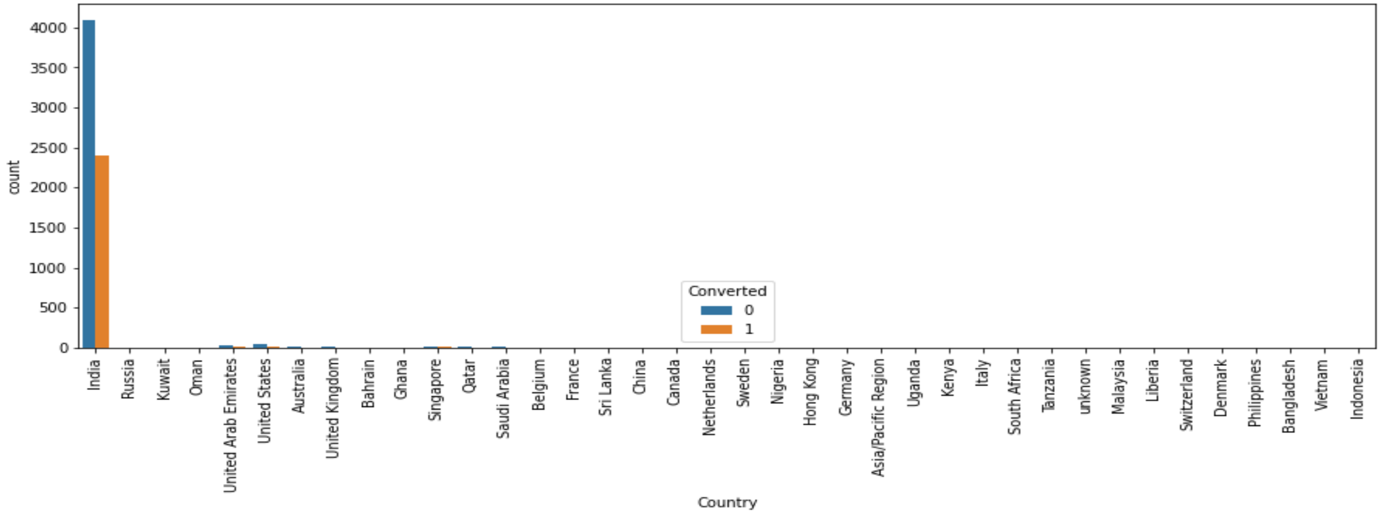
```
lead_score.describe()
```

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
count	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000000
mean	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.344883
std	23405.995698	0.486714	4.854853	548.021466	2.161418	1.386694	1.811395
min	579533.000000	0.000000	0.000000	0.000000	0.000000	7.000000	11.000000
25%	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.000000
50%	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.000000
75%	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.000000
max	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.000000

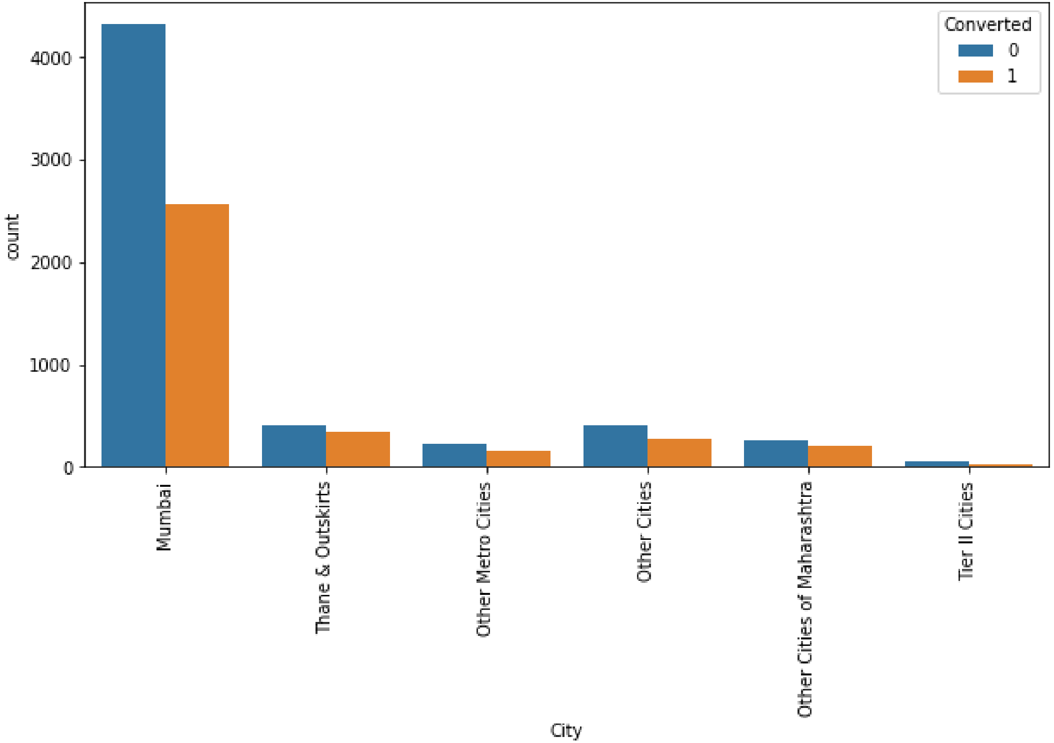
Exploratory data analysis

After Converting irrelevant data to respective values, we check according to categories.

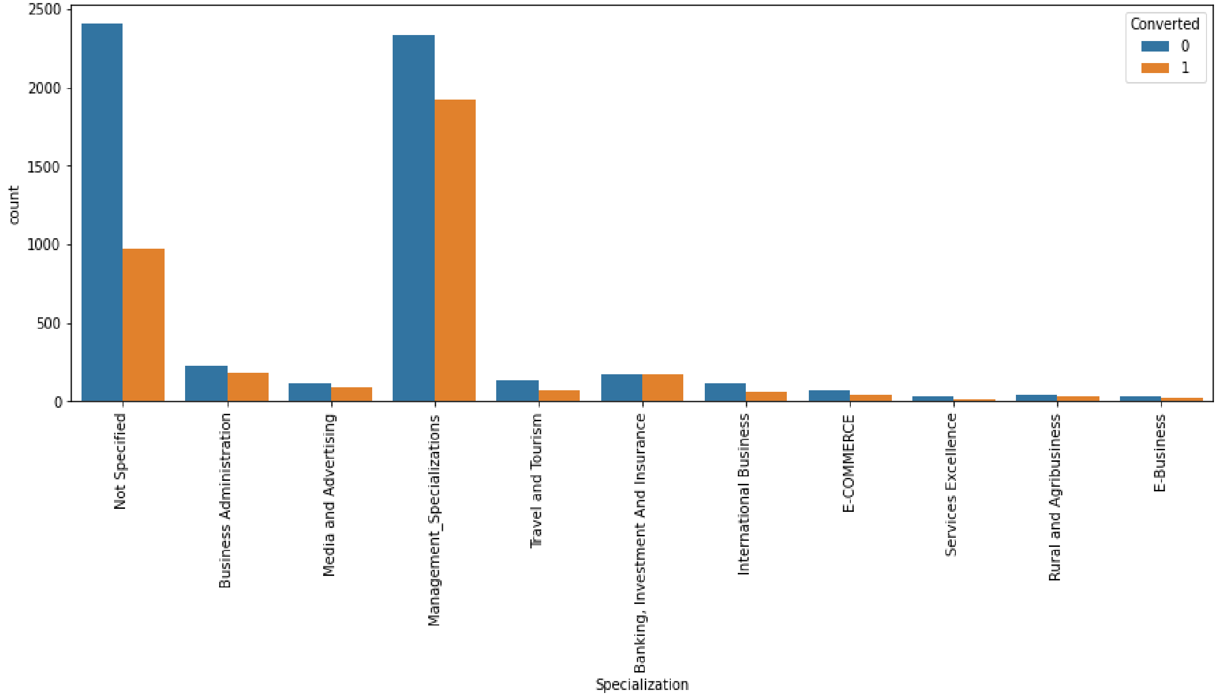
checking value counts of Country column.



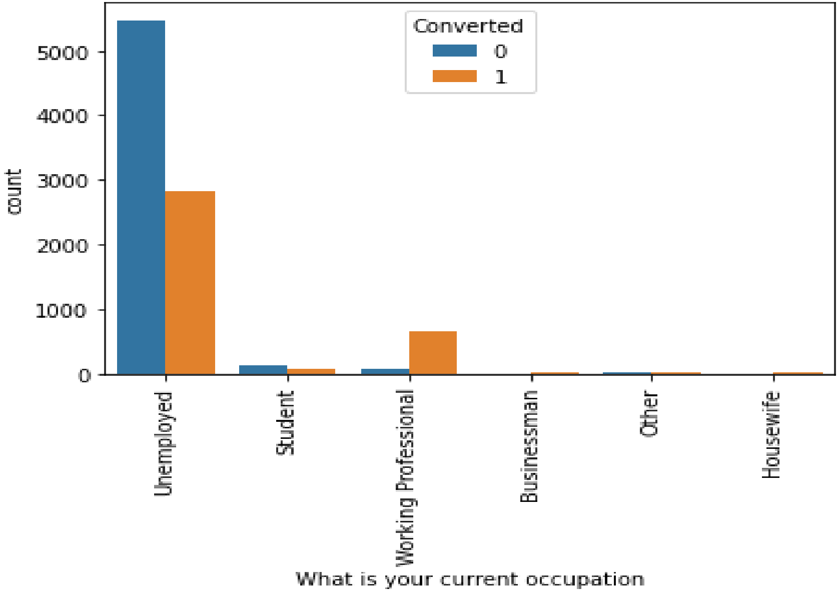
As we can see the Number of Values for India are quite high (nearly 97% of the Data), this column can be dropped. plotting spread of City column after replacing NaN values



Plotting spread of Specialization column

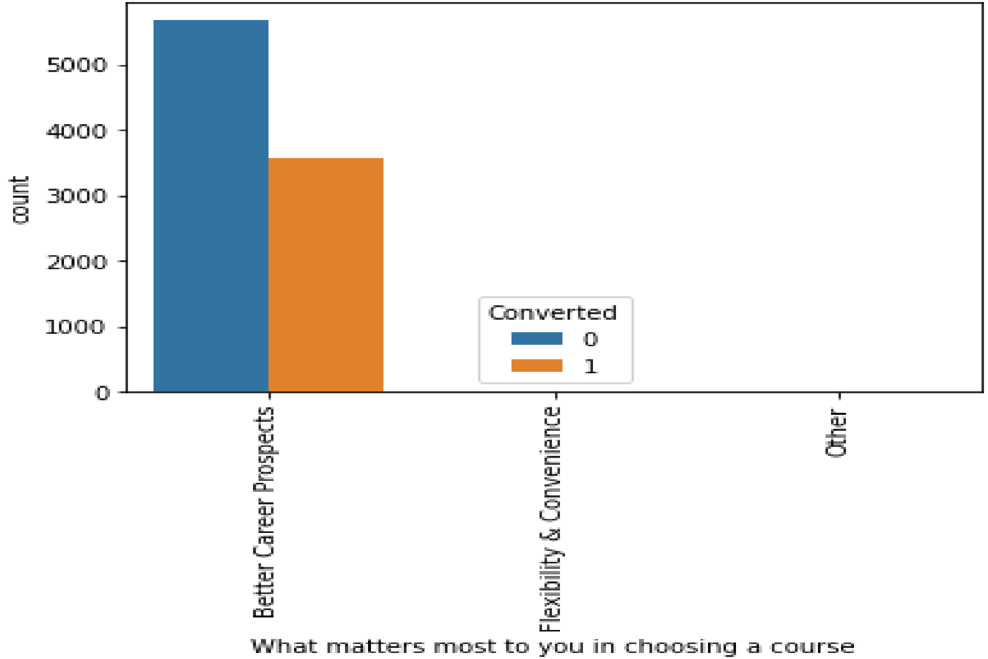


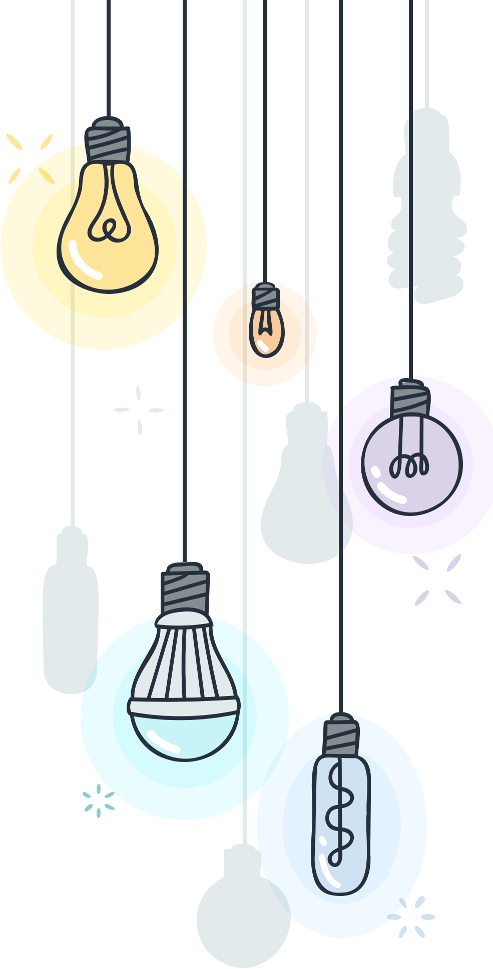
Visualizing count of Variable based on Converted value



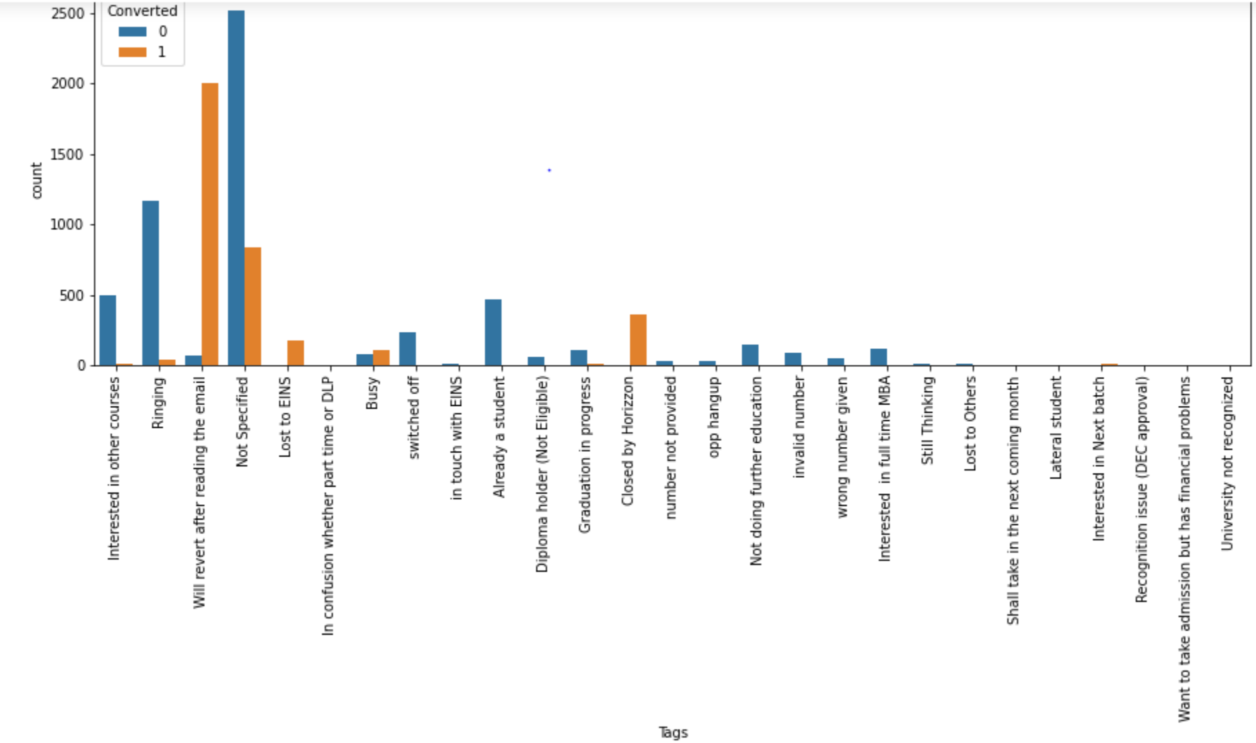
- Working Professionals going for the course have high chances of joining it.
- Unemployed lead_score are the most in terms of Absolute numbers.

VISUALIZING COUNT OF 'WHAT MATTERS MOST TO YOU IN CHOOSING A COURSE' BASED ON CONVERTED VALUE

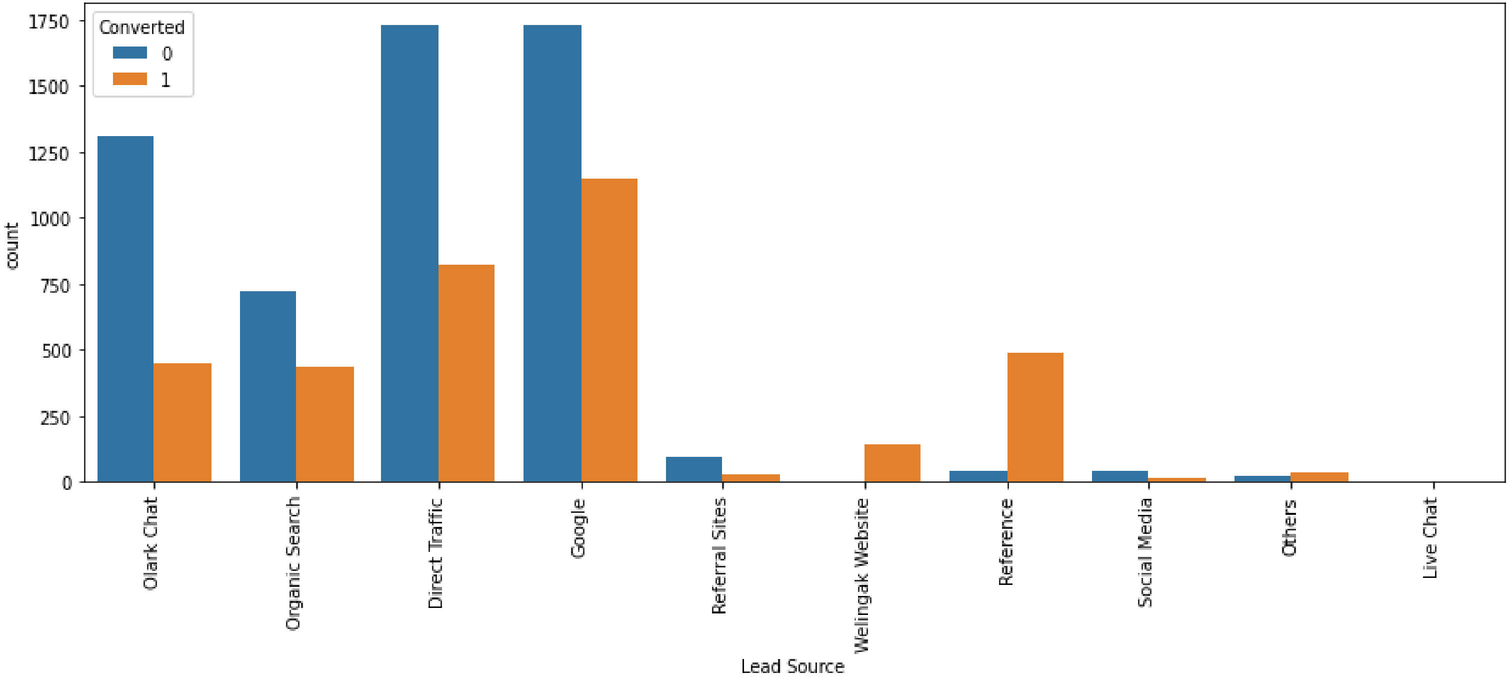




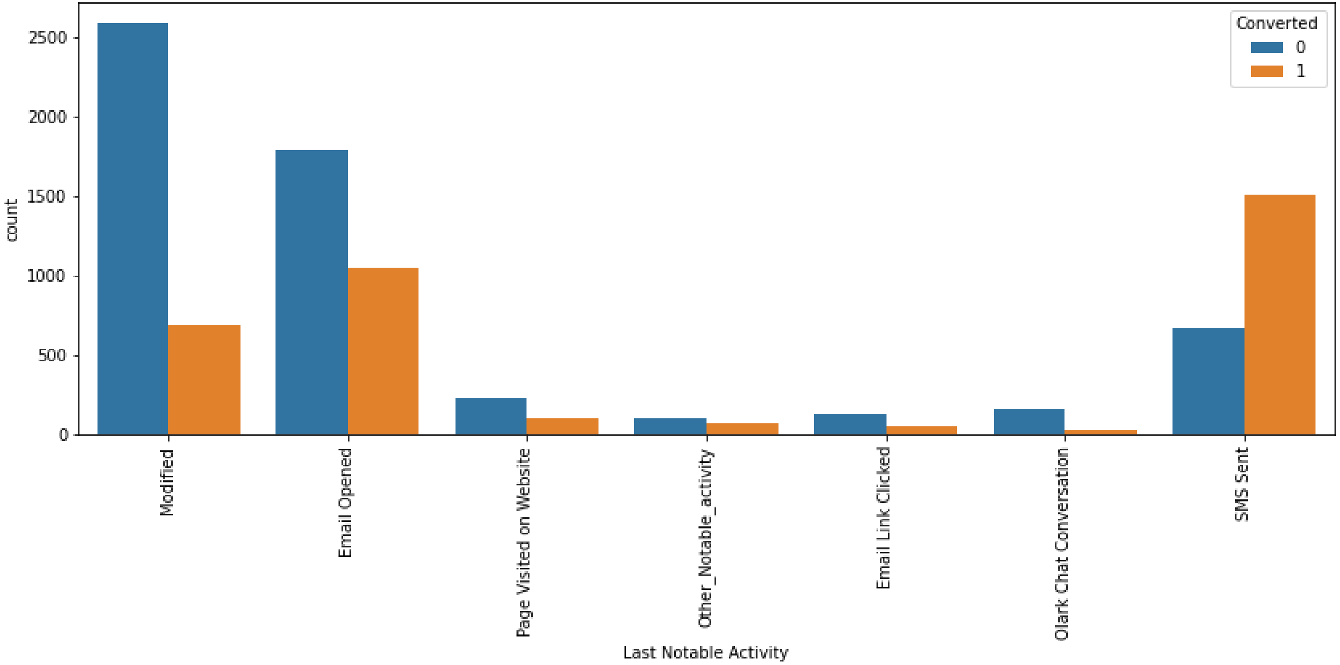
VISUALIZING COUNT OF 'TAGS' BASED ON CONVERTED VALUE



VISUALIZING COUNT OF 'LEAD SOURCE' BASED ON CONVERTED VALUE

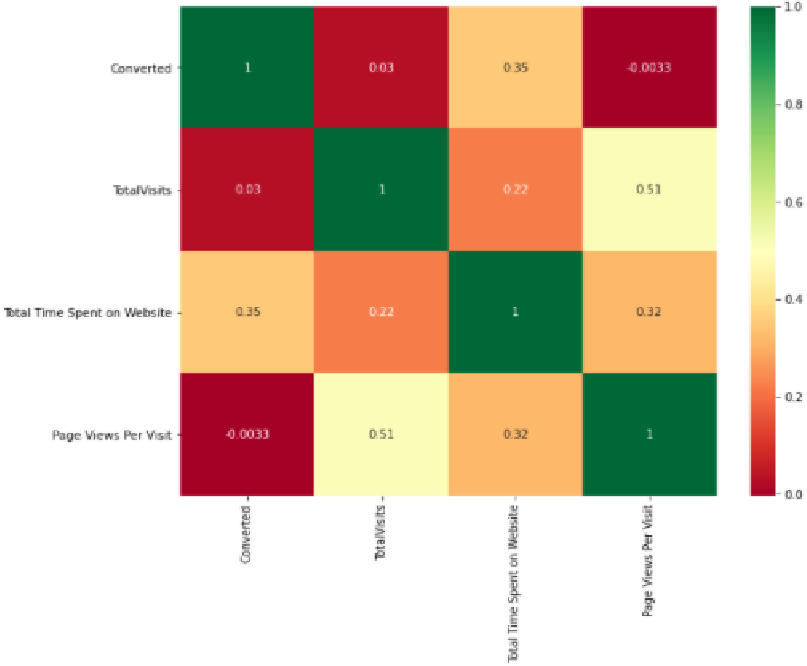


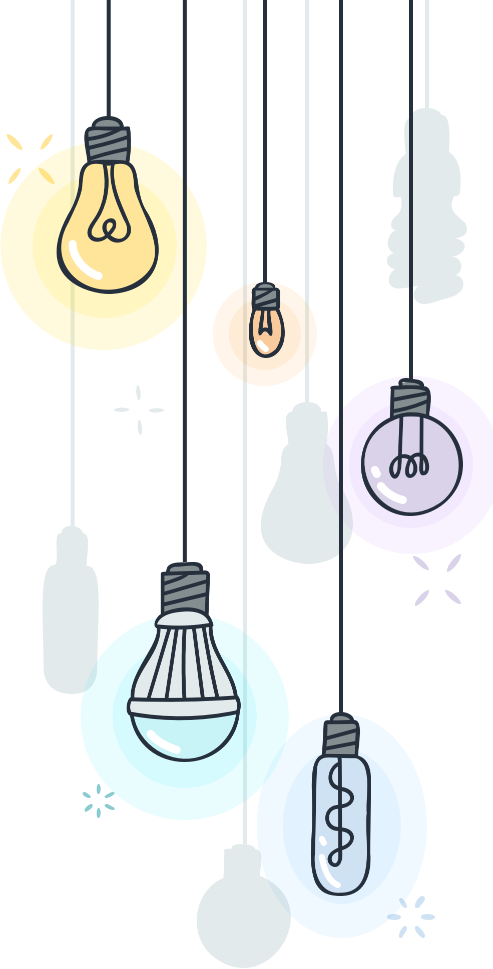
VISUALIZING COUNT OF 'LAST NOTABLE ACTIVITY' BASED ON CONVERTED VALUE



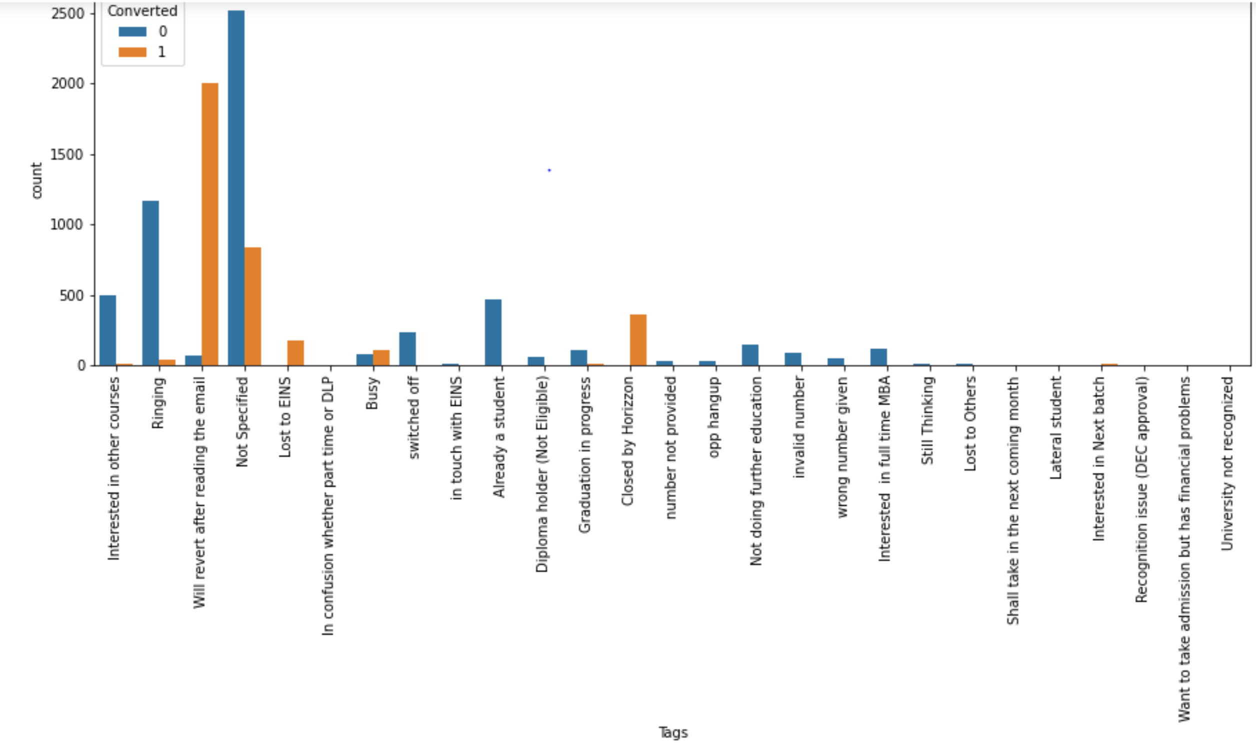
Checking if the % of Data that has Converted Values = 1

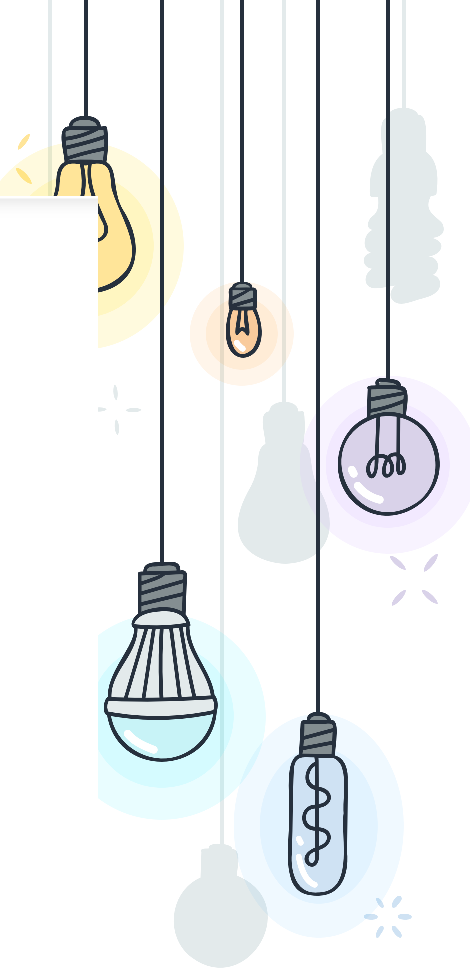
We found the percentage of data that has converted value equals 38.020.





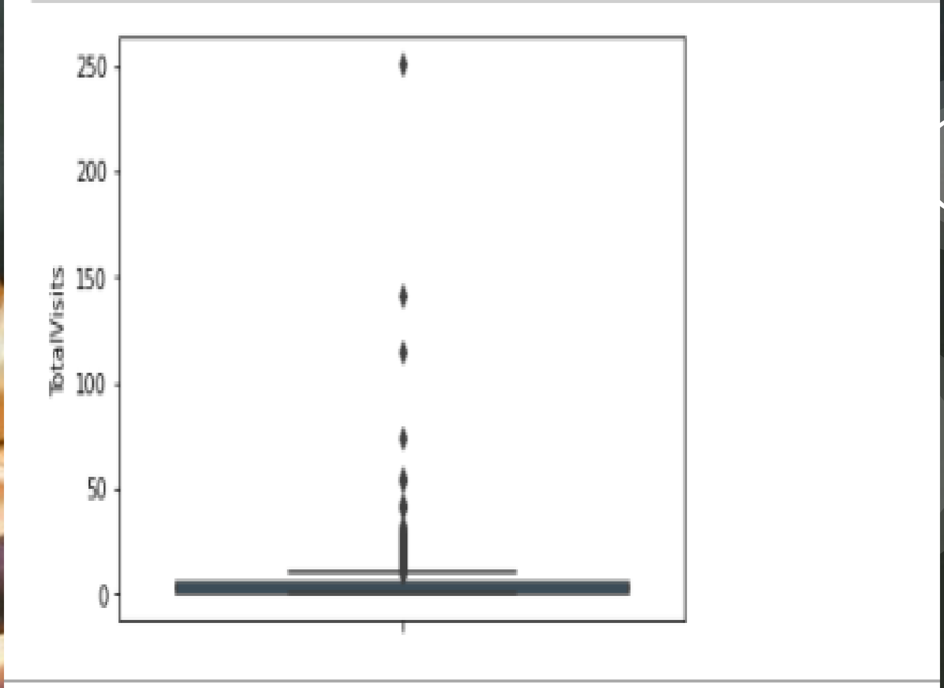
Visualizing count of 'Tags' based on Converted value



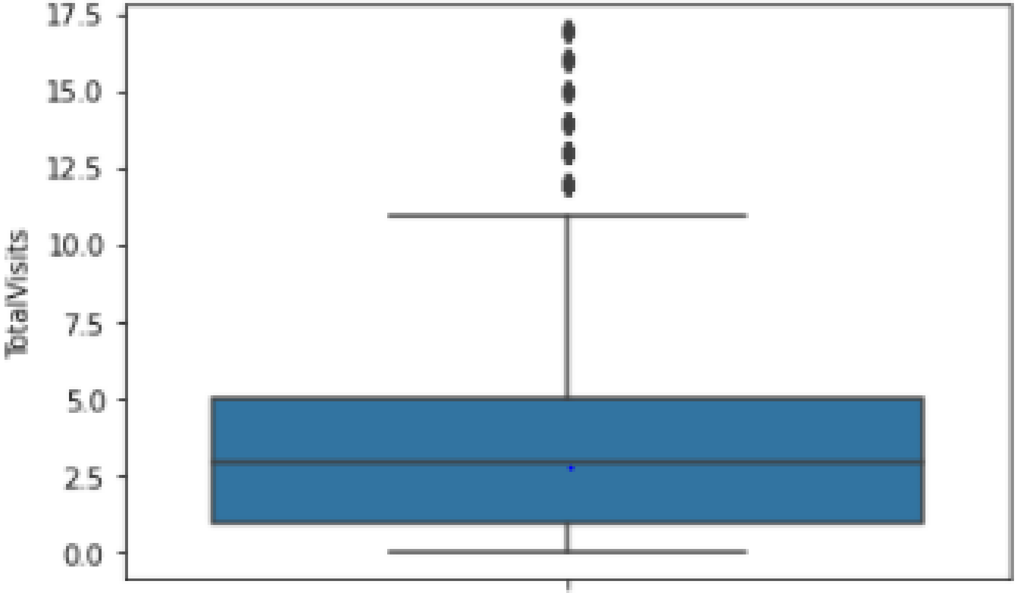


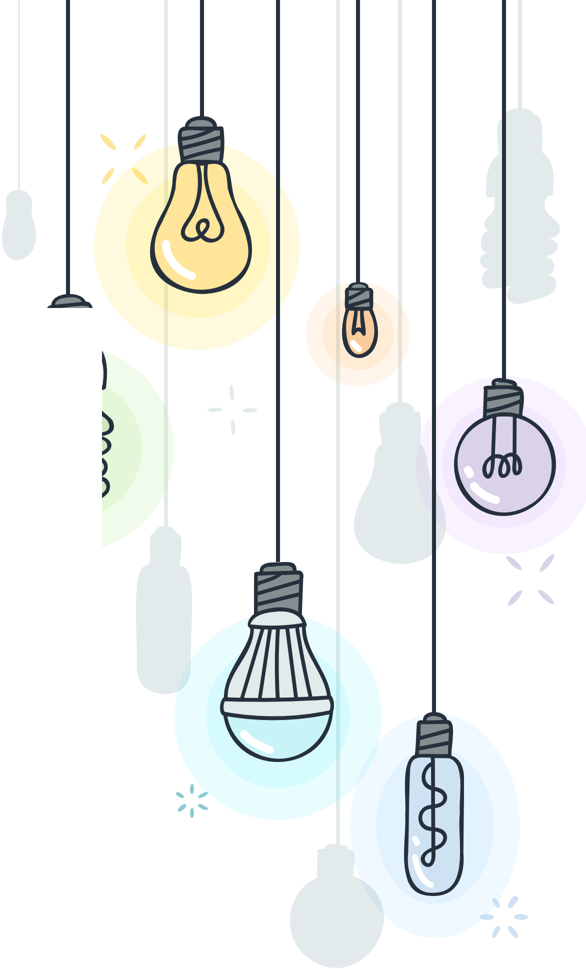
Search for outliers

We find that there are some outliers in 'Total Visits' features.

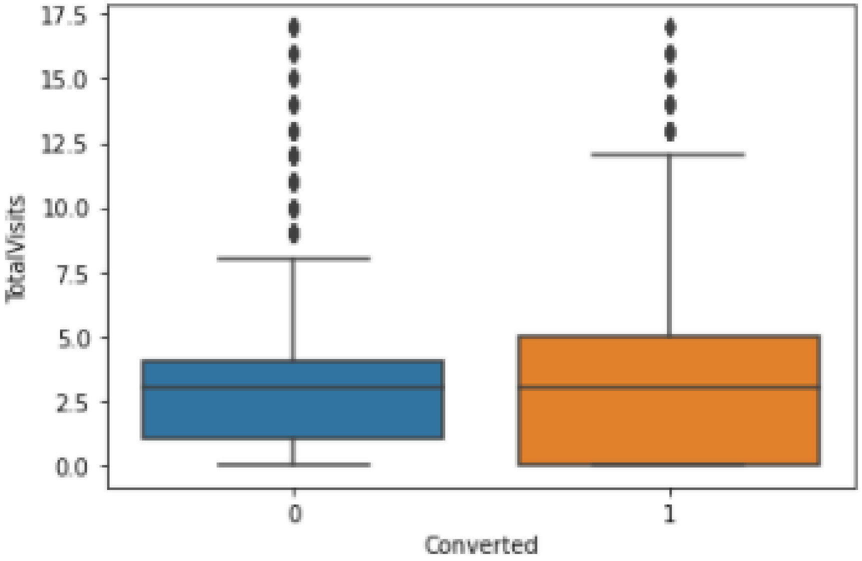


Outlier Treatment: Remove top & bottom 1% of the Column Outlier values





CHECKING SPREAD OF "TOTAL VISITS" VS CONVERTED VARIABLE



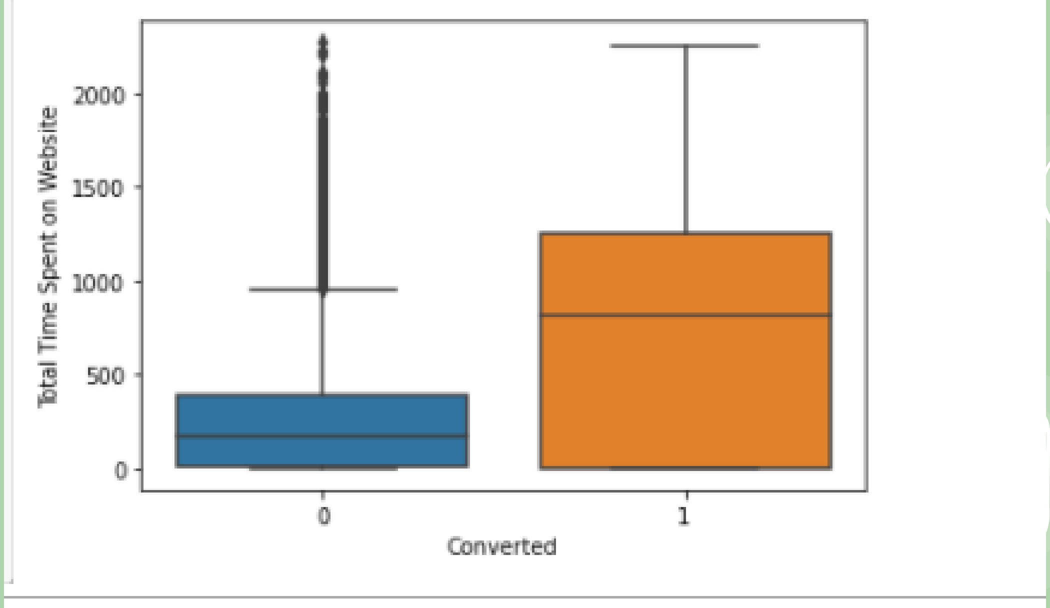
Inference

- Median for converted and not converted lead_score are the close.
- Nothng conclusive can be said on the basis of Total Visits

CHECKING SPREAD OF "TOTAL TIME SPENT ON WEBSITE" VS CONVERTED VARIABLE

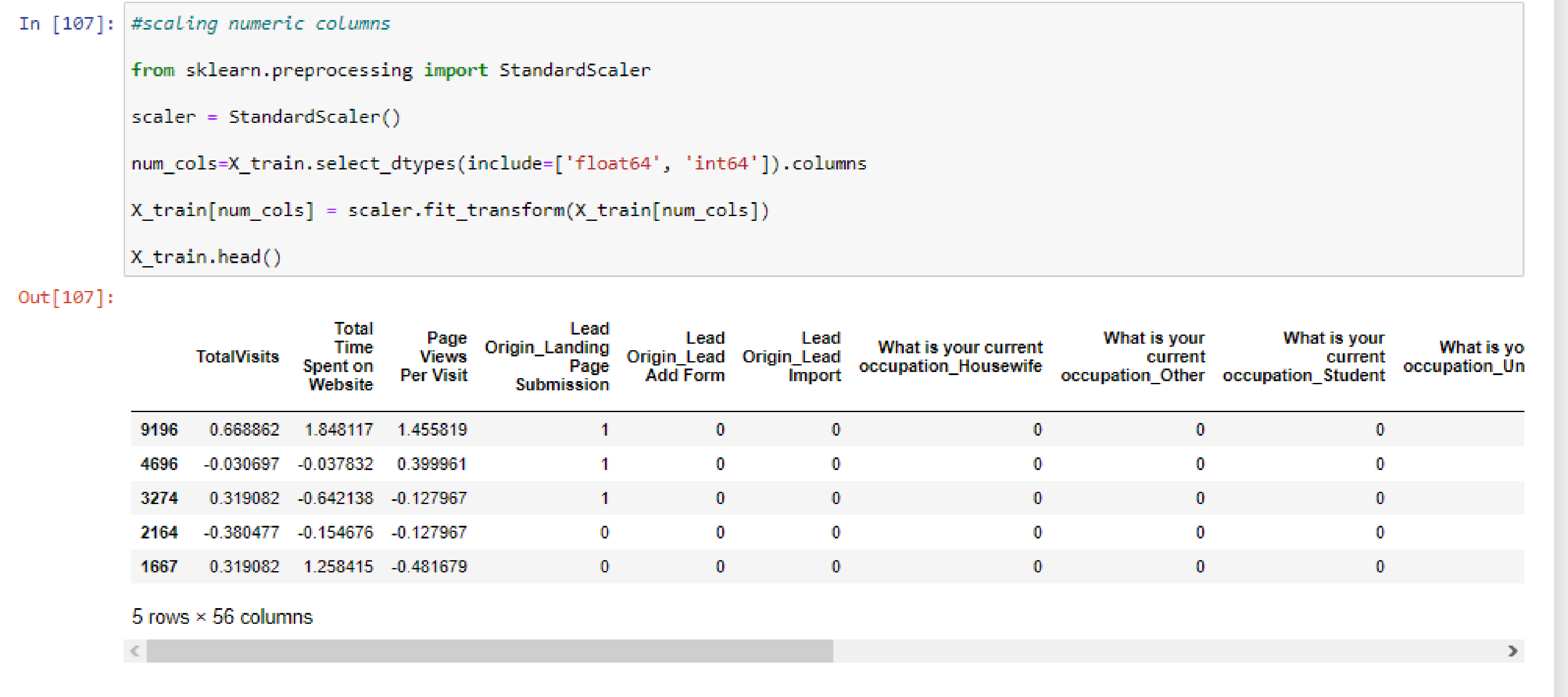
INFERENCE

- LEADS SPENDING MORE TIME ON THE WEBSITE ARE MORE LIKELY TO BE CONVERTED.
- WEBSITE SHOULD BE MADE MORE ENGAGING TO MAKE LEAD _SCORE SPEND MORE TIME.



Find more maps at slidescarnival.com/extra-free-resources-icons-and-maps

Scaling of Data



```
In [107]: #scaling numeric columns

          from sklearn.preprocessing import StandardScaler

          scaler = StandardScaler()

          num_cols=X_train.select_dtypes(include=['float64', 'int64']).columns

          X_train[num_cols] = scaler.fit_transform(X_train[num_cols])

          X_train.head()
```

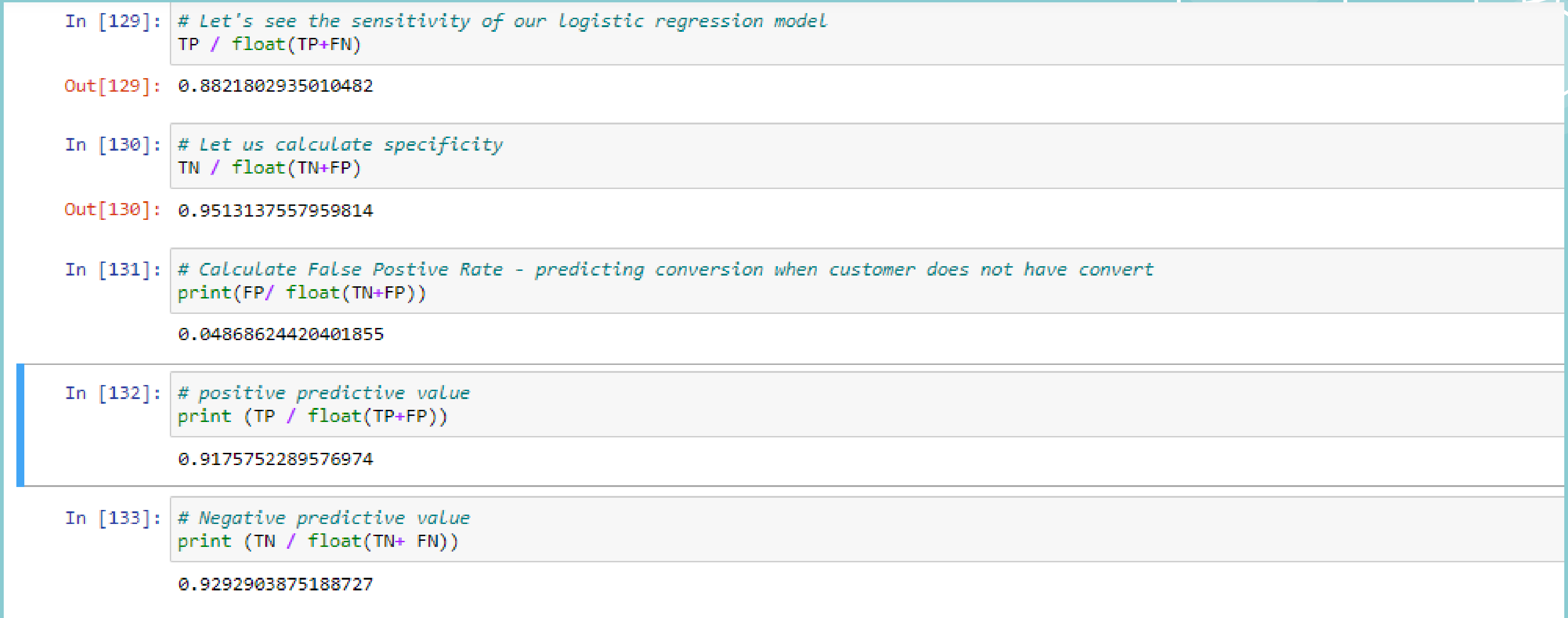
Out[107]:

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	What is your current occupation_Housewife	What is your current occupation_Other	What is your current occupation_Student	What is yo occupation_Un
9196	0.668862	1.848117	1.455819	1	0	0	0	0	0	
4696	-0.030697	-0.037832	0.399961	1	0	0	0	0	0	
3274	0.319082	-0.642138	-0.127967	1	0	0	0	0	0	
2164	-0.380477	-0.154676	-0.127967	0	0	0	0	0	0	
1667	0.319082	1.258415	-0.481679	0	0	0	0	0	0	

5 rows × 56 columns

Model Building using Stats Model & RFE

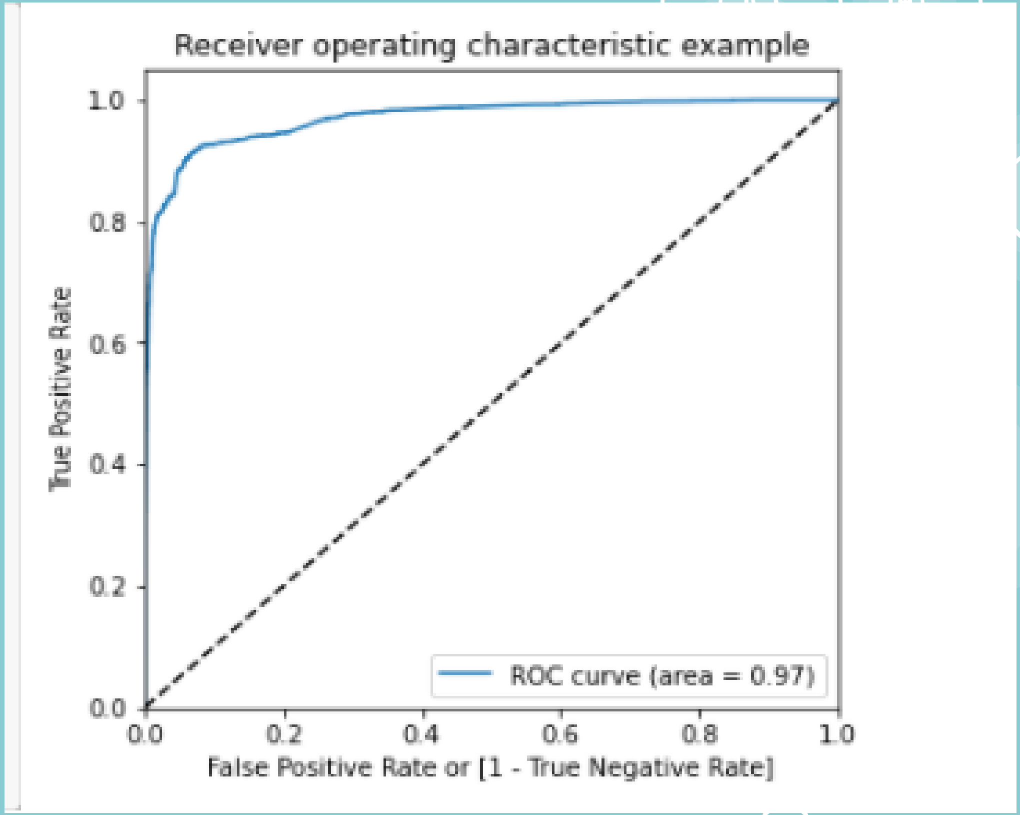
After dropping multiple columns due to high VIF and building the model 3 times, below is the list of properties we can observe :-



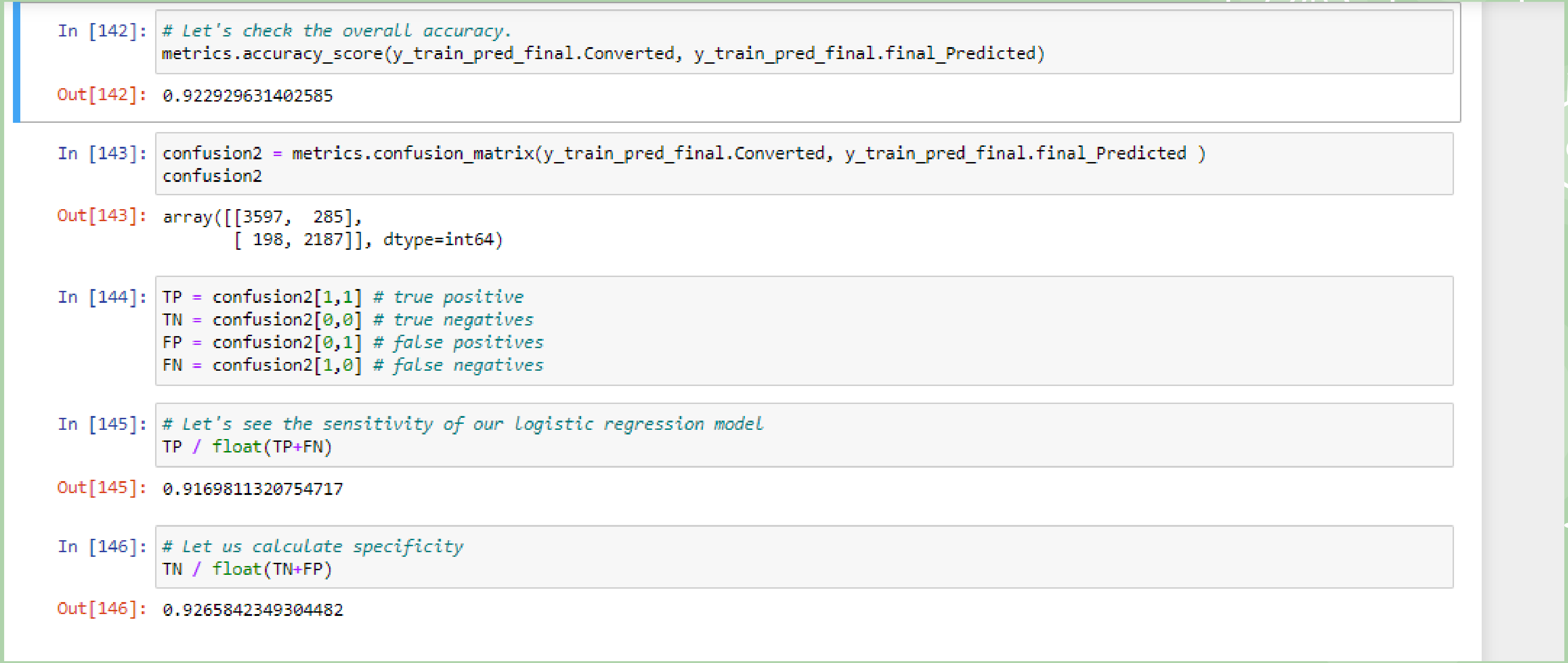
```
In [129]: # Let's see the sensitivity of our logistic regression model
          TP / float(TP+FN)

Out[129]: 0.8821802935010482

In [130]: # Let us calculate specificity
          TN / float(TN+FP)

Out[130]: 0.9513137557959814

In [131]: # Calculate False Postive Rate - predicting conversion when customer does not have convert
          print(FP/ float(TN+FP))

          0.04868624420401855

In [132]: # positive predictive value
          print (TP / float(TP+FP))

          0.9175752289576974

In [133]: # Negative predictive value
          print (TN / float(TN+ FN))

          0.9292903875188727
```

ROC Curve

We are getting a good value of 0.97 indicating a good predictive model.



There are some other properties that are required to be checked. Here we will check those informations if they are fine.



```
In [142]: # Let's check the overall accuracy.
          metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_Predicted)

Out[142]: 0.922929631402585

In [143]: confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.final_Predicted )
          confusion2

Out[143]: array([[3597,  285],
                 [ 198, 2187]], dtype=int64)

In [144]: TP = confusion2[1,1] # true positive
          TN = confusion2[0,0] # true negatives
          FP = confusion2[0,1] # false positives
          FN = confusion2[1,0] # false negatives

In [145]: # Let's see the sensitivity of our logistic regression model
          TP / float(TP+FN)

Out[145]: 0.9169811320754717

In [146]: # Let us calculate specificity
          TN / float(TN+FP)

Out[146]: 0.9265842349304482
```

Find more maps at slidescarnival.com/extra-free-resources-icons-and-maps

PREDICTION

Final Observation:

Let us compare the values obtained for Train & Test:

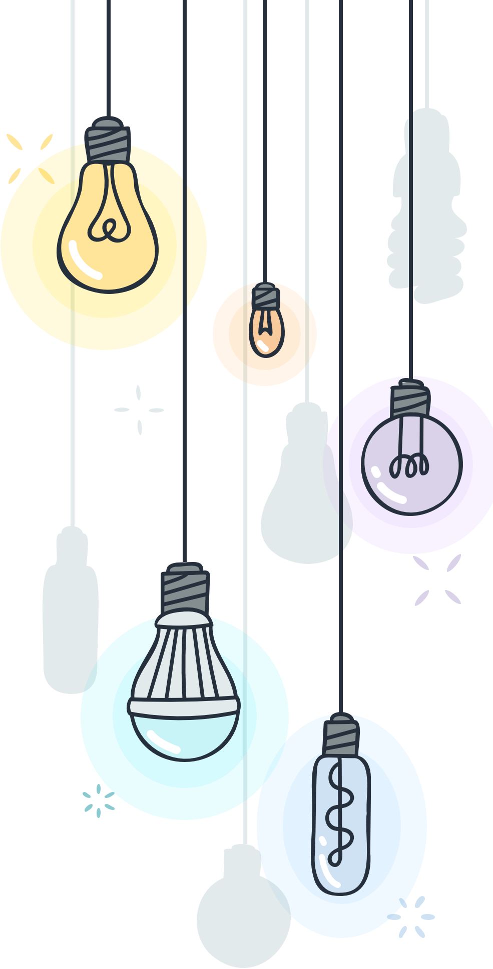
Train Data:

- Accuracy : 92.29%
- Sensitivity : 91.70%
- Specificity : 92.66%

Test Data:

- Accuracy : 92.78%
- Sensitivity : 91.98%
- Specificity : 93.26%

Thus we can see that this model predicts the Conversion Rate very well and the CEO will surely like this model better due to high conversion rate.



Thanks!