Lead Scoring Case Study

Group Members:

Ramakrishnan Ramasamy
Dinesh
Rajshri

Date : 04/14/2024

Agenda

Problem Statement

Solution Methodology

Data Manipulation

EDA

Model Building

Model Evaluation

Conclusion

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 Objective

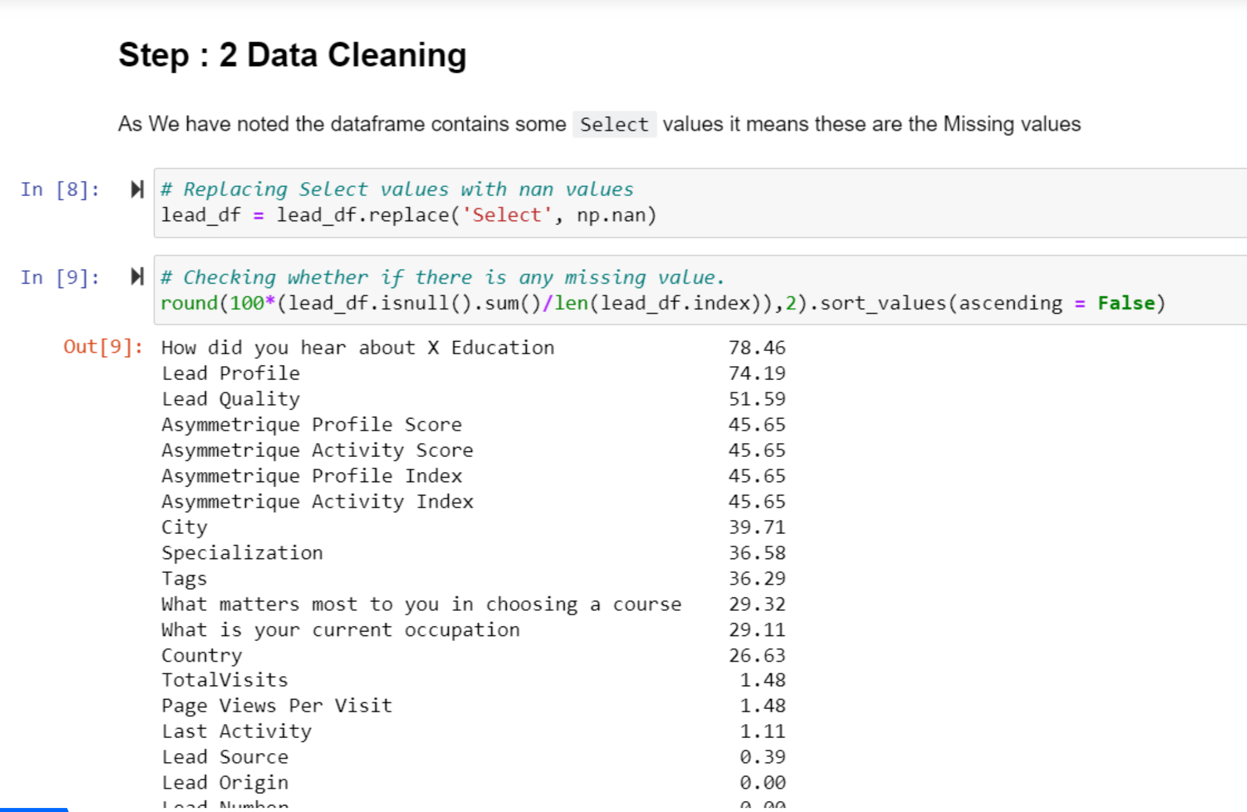
Build a logistic regression model to assign a lead score between 0 and 100 to each of the leads which can be used by the company to target potential leads. A higher score would mean that the lead is hot, i.e. is most likely to convert whereas a lower score would mean that the lead is cold and will mostly not get converted.

Solution Methodology

Let’s approach the solution by following the below steps:

1. Data cleaning and data manipulation.
    1. Check and handle duplicate data. 2.
    2. Check and handle NA values and missing values. 3.
    3. Drop columns, if it contains large amount of missing values
    4. 4. Imputation of the values, if necessary.
    5. 5. Check and handle outliers in data. □
2. EDA
    1. Univariate data analysis: value count, distribution of variable etc.
    2. Bivariate data analysis: correlation coefficients and pattern between the variables etc.
        1. Feature Scaling & Dummy Variables and encoding of the data.
        2. Classification technique: logistic regression used for the model making and prediction.
        3. Validation of the model.
        4. Model presentation. □
3. Conclusions

Data Manipulation:



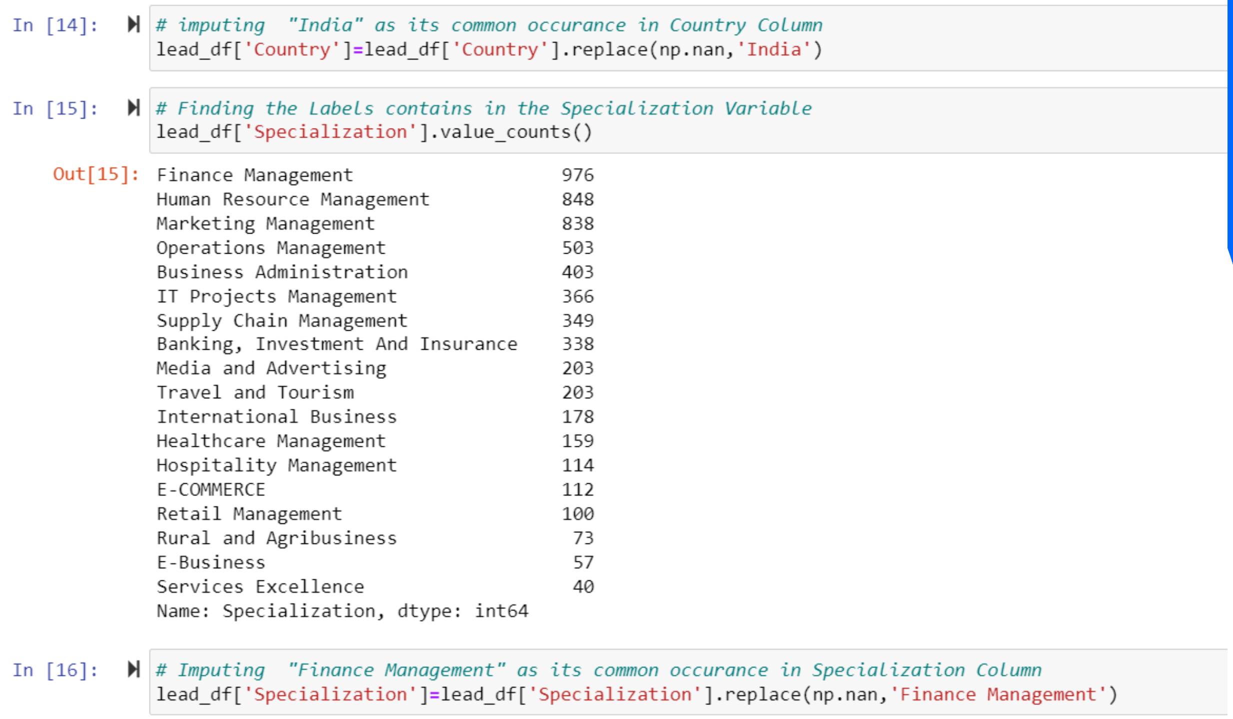
Step : 2 Data Cleaning

As We have noted the dataframe contains some `Select` values it means these are the Missing values

```
In [8]:  # Replacing Select values with nan values
         lead_df = lead_df.replace('Select', np.nan)
```

```
In [9]:  # Checking whether if there is any missing value.
         round(100*(lead_df.isnull().sum()/len(lead_df.index)),2).sort_values(ascending = False)
```

```
Out[9]: How did you hear about X Education              78.46
        Lead Profile                                    74.19
        Lead Quality                                    51.59
        Asymmetrique Profile Score                      45.65
        Asymmetrique Activity Score                     45.65
        Asymmetrique Profile Index                      45.65
        Asymmetrique Activity Index                     45.65
        City                                            39.71
        Specialization                                  36.58
        Tags                                            36.29
        What matters most to you in choosing a course   29.32
        What is your current occupation                 29.11
        Country                                         26.63
        TotalVisits                                      1.48
        Page Views Per Visit                             1.48
        Last Activity                                    1.11
        Lead Source                                      0.39
        Lead Origin                                      0.00
        Lead Number                                      0.00
```



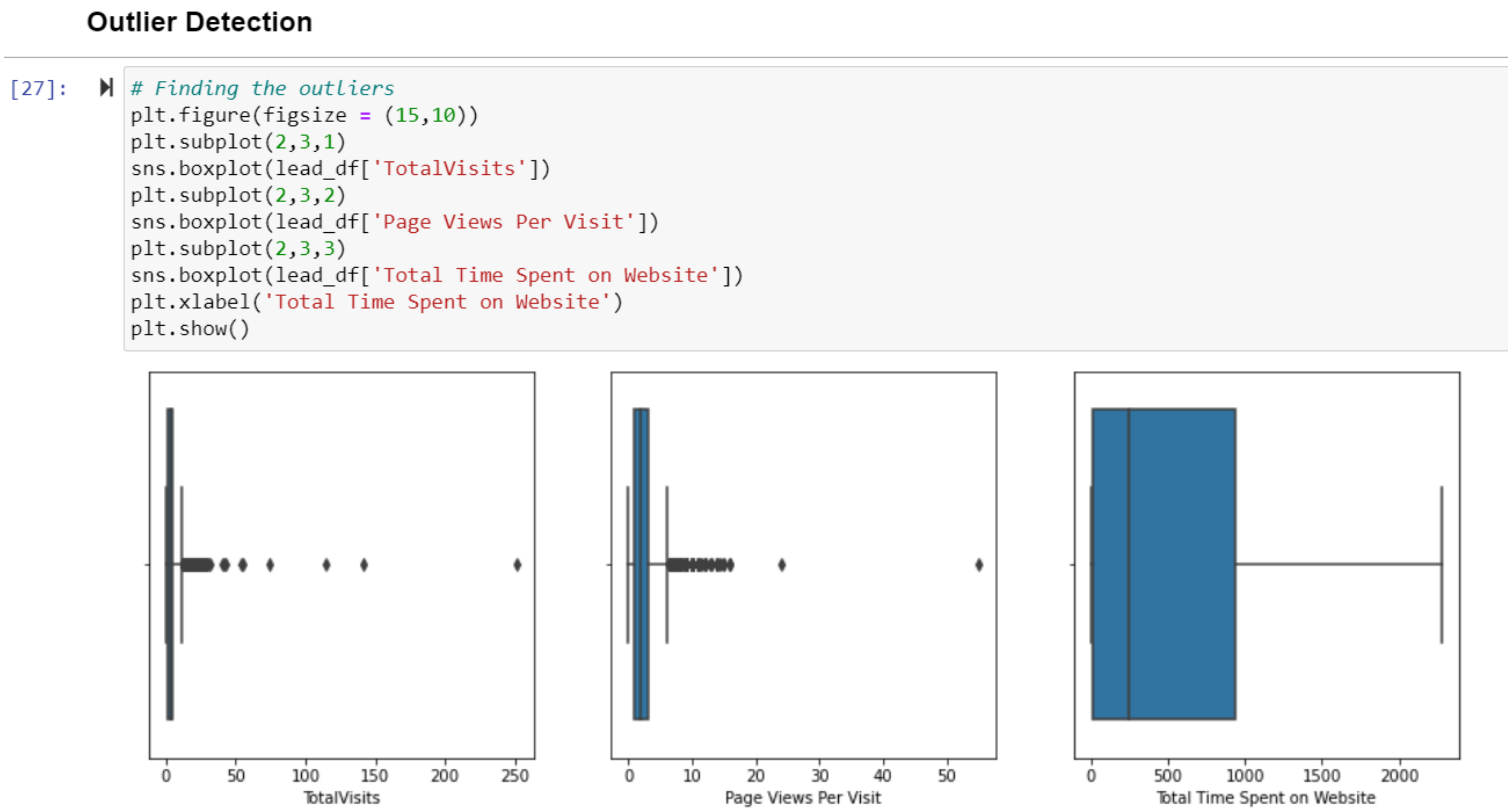
```
In [14]:  # imputing  "India" as its common occurance in Country Column
          lead_df['Country']=lead_df['Country'].replace(np.nan,'India')
```

```
In [15]:  # Finding the Labels contains in the Specialization Variable
          lead_df['Specialization'].value_counts()
```

```
Out[15]: Finance Management                   976
         Human Resource Management            848
         Marketing Management                 838
         Operations Management                503
         Business Administration              403
         IT Projects Management               366
         Supply Chain Management              349
         Banking, Investment And Insurance    338
         Media and Advertising                203
         Travel and Tourism                   203
         International Business               178
         Healthcare Management                159
         Hospitality Management               114
         E-COMMERCE                           112
         Retail Management                    100
         Rural and Agribusiness                73
         E-Business                            57
         Services Excellence                   40
         Name: Specialization, dtype: int64
```

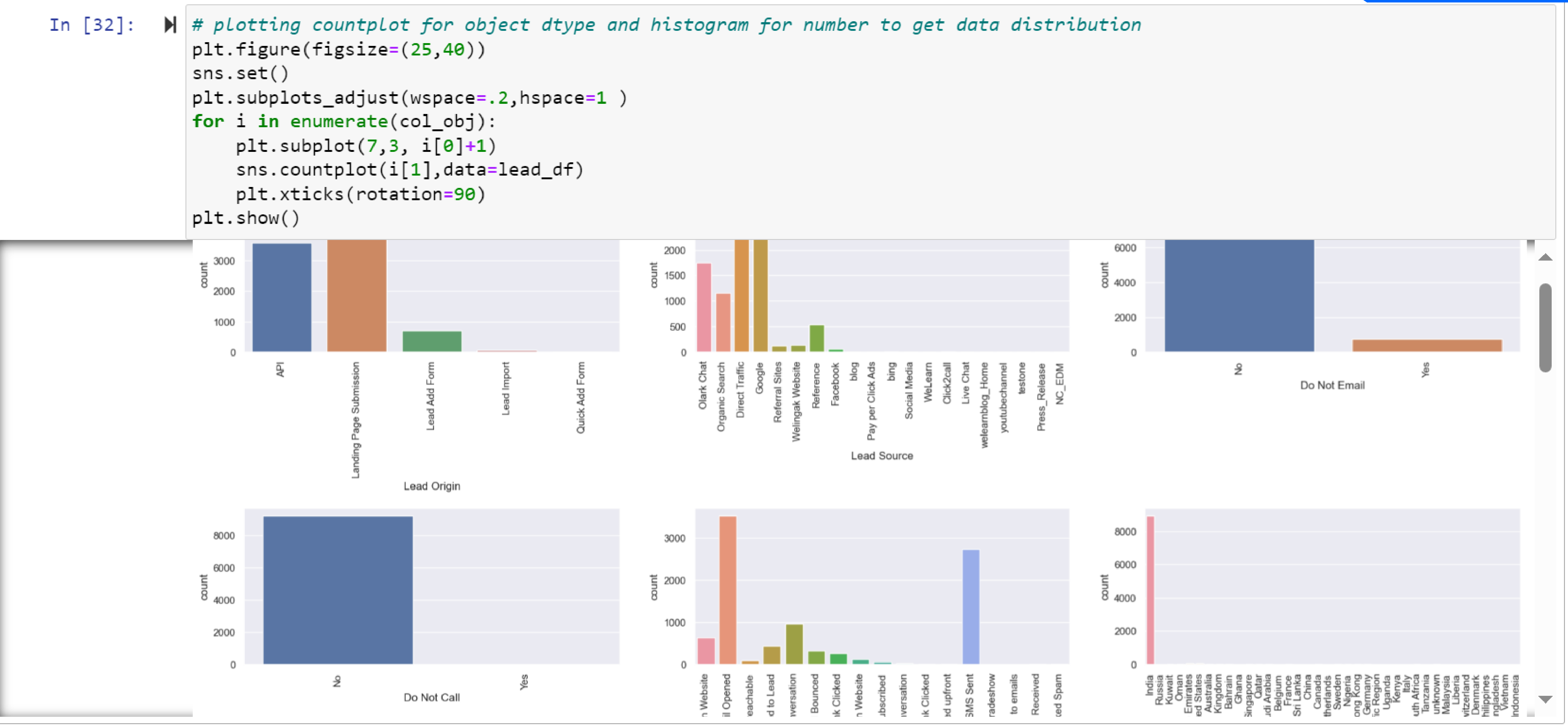
```
In [16]:  # Imputing  "Finance Management" as its common occurance in Specialization Column
          lead_df['Specialization']=lead_df['Specialization'].replace(np.nan,'Finance Management')
```

Outlier Detection



Outlier Detection

```
[27]: # Finding the outliers
      plt.figure(figsize = (15,10))
      plt.subplot(2,3,1)
      sns.boxplot(lead_df['TotalVisits'])
      plt.subplot(2,3,2)
      sns.boxplot(lead_df['Page Views Per Visit'])
      plt.subplot(2,3,3)
      sns.boxplot(lead_df['Total Time Spent on Website'])
      plt.xlabel('Total Time Spent on Website')
      plt.show()
```



```
In [32]:
# plotting countplot for object dtype and histogram for number to get data distribution
plt.figure(figsize=(25,40))
sns.set()
plt.subplots_adjust(wspace=.2,hspace=1 )
for i in enumerate(col_obj):
    plt.subplot(7,3, i[0]+1)
    sns.countplot(i[1],data=lead_df)
    plt.xticks(rotation=90)
plt.show()
```

EDA

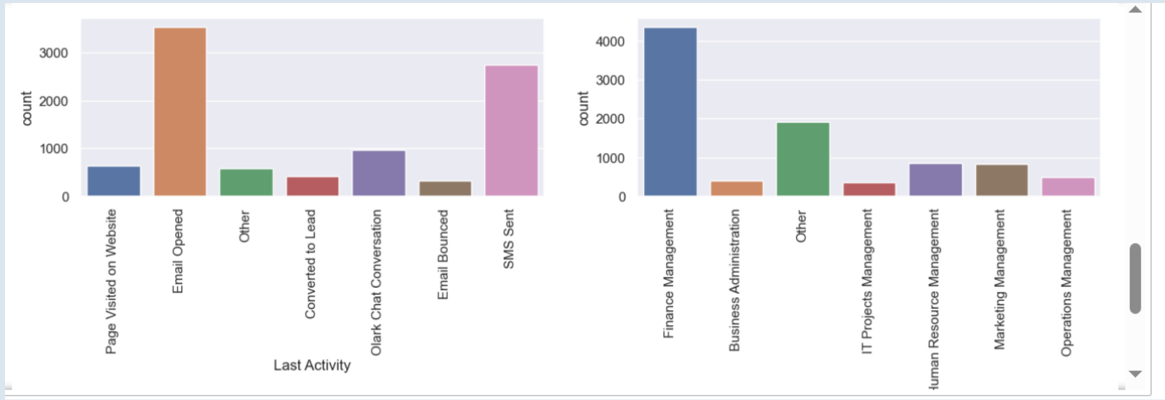
Univariate Analysis for Categorical Values

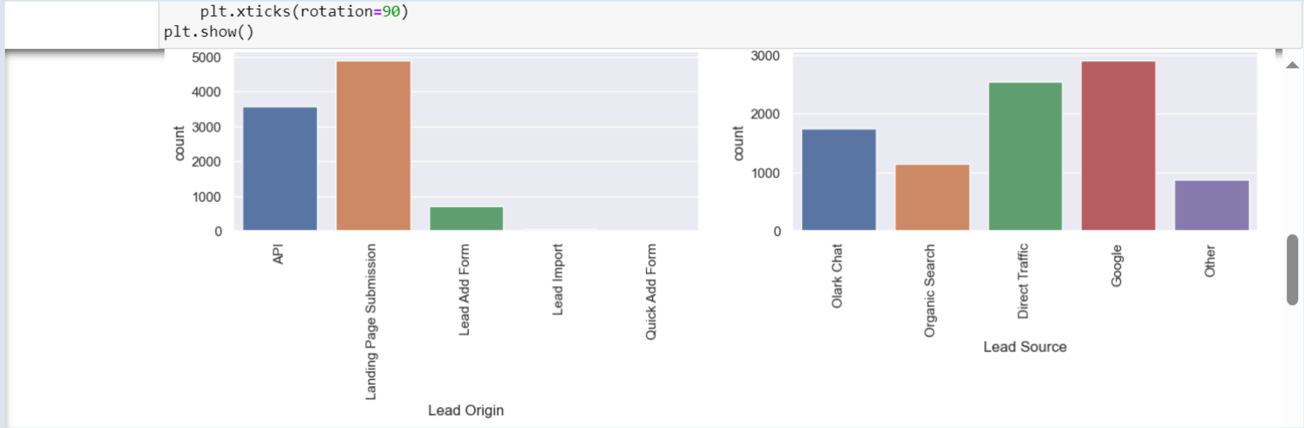
- In Lead Source Direct Traffic and Google are the two main source for Leads

- The Number of values is High in Email Opened and SMS Sent in Last Activity

- Most of the people chooses Finance Management Specialization rather than other Specialization

- The IT Project management have very lees so that most of the People not prefered this Specialization



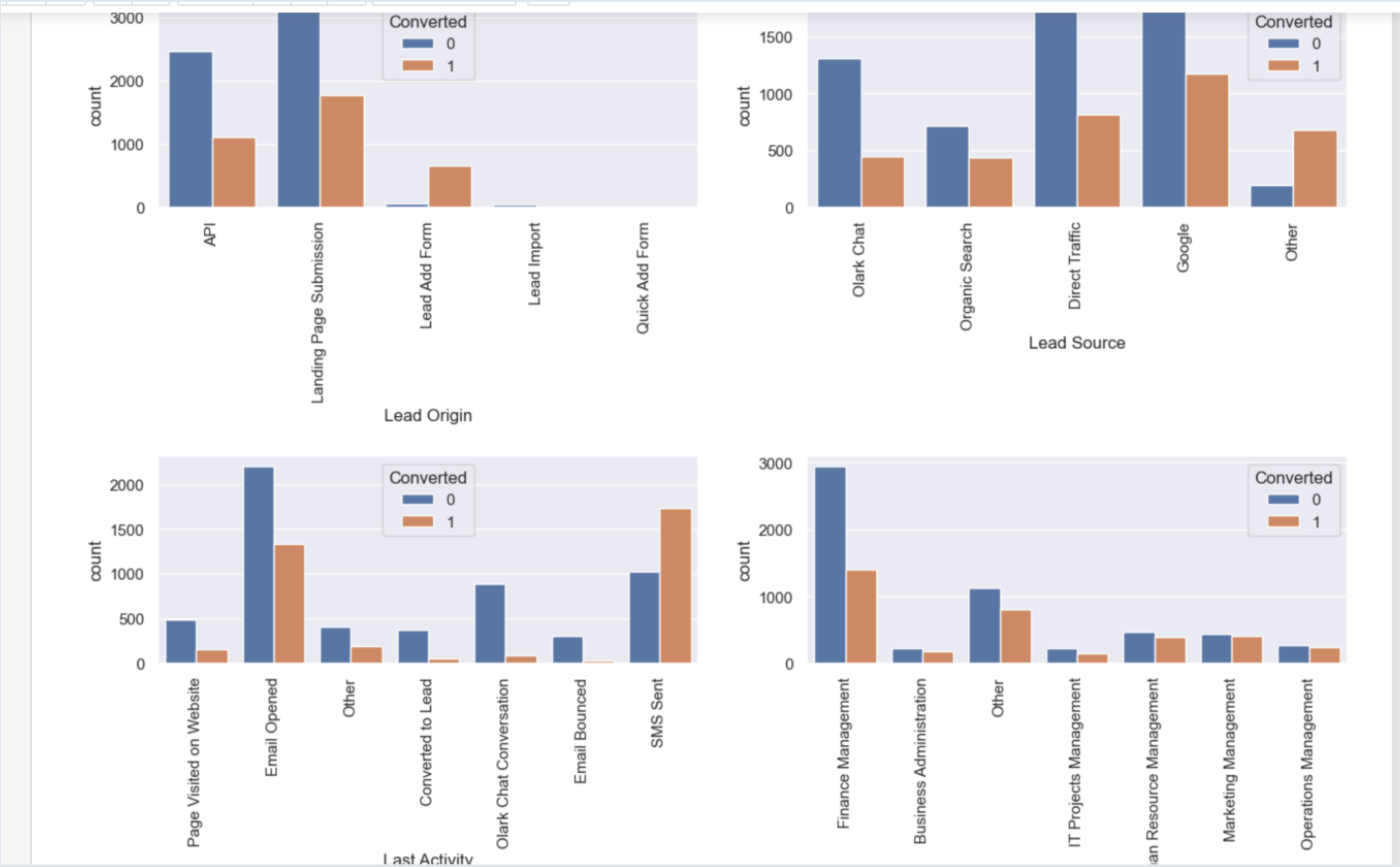


```
    plt.xticks(rotation=90)
plt.show()
```

EDA

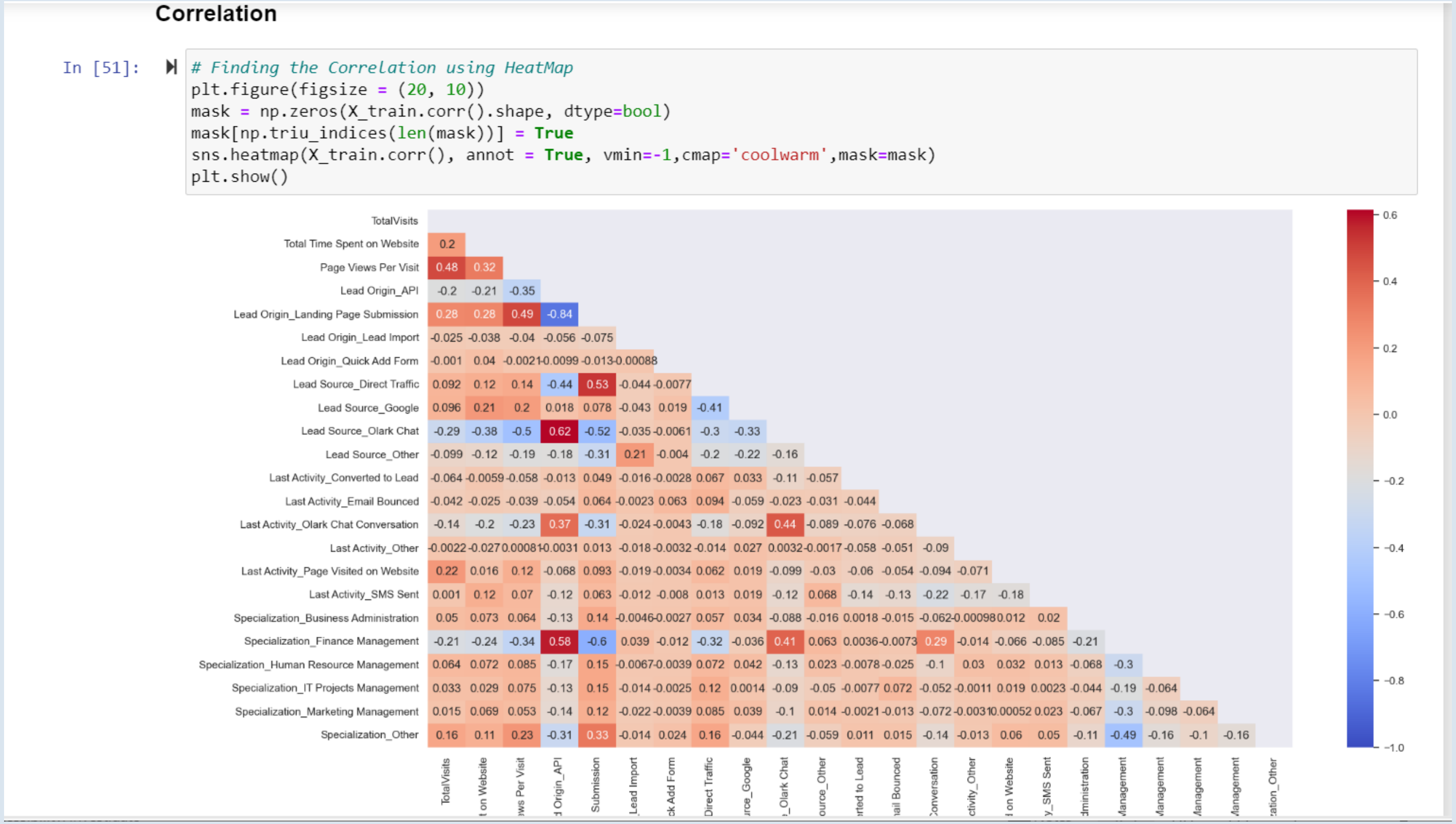
Bivariate Analysis with respect to Target Columns

- •In Lead Source The number of Hot leads is higher in Direct Traffic and Google less in Other Category
- •In Last Activity the number of Hot leads is higher in SMS and in EMAIL cold leads is higher than hot leads.
- •In Last Notable Activity it's mostly same as Last Activity.
- •In Specialization most of the leads are comes from Finance management but here Hot leads are lesser than Cold lead



Model Creation

- Split the dataset into Train and Test Data set
- Identify the correlation in the train data set
- FIT the Regression model with 20 cols.
- Building a Logistic Regression using statsmodel, for the detailed statistic



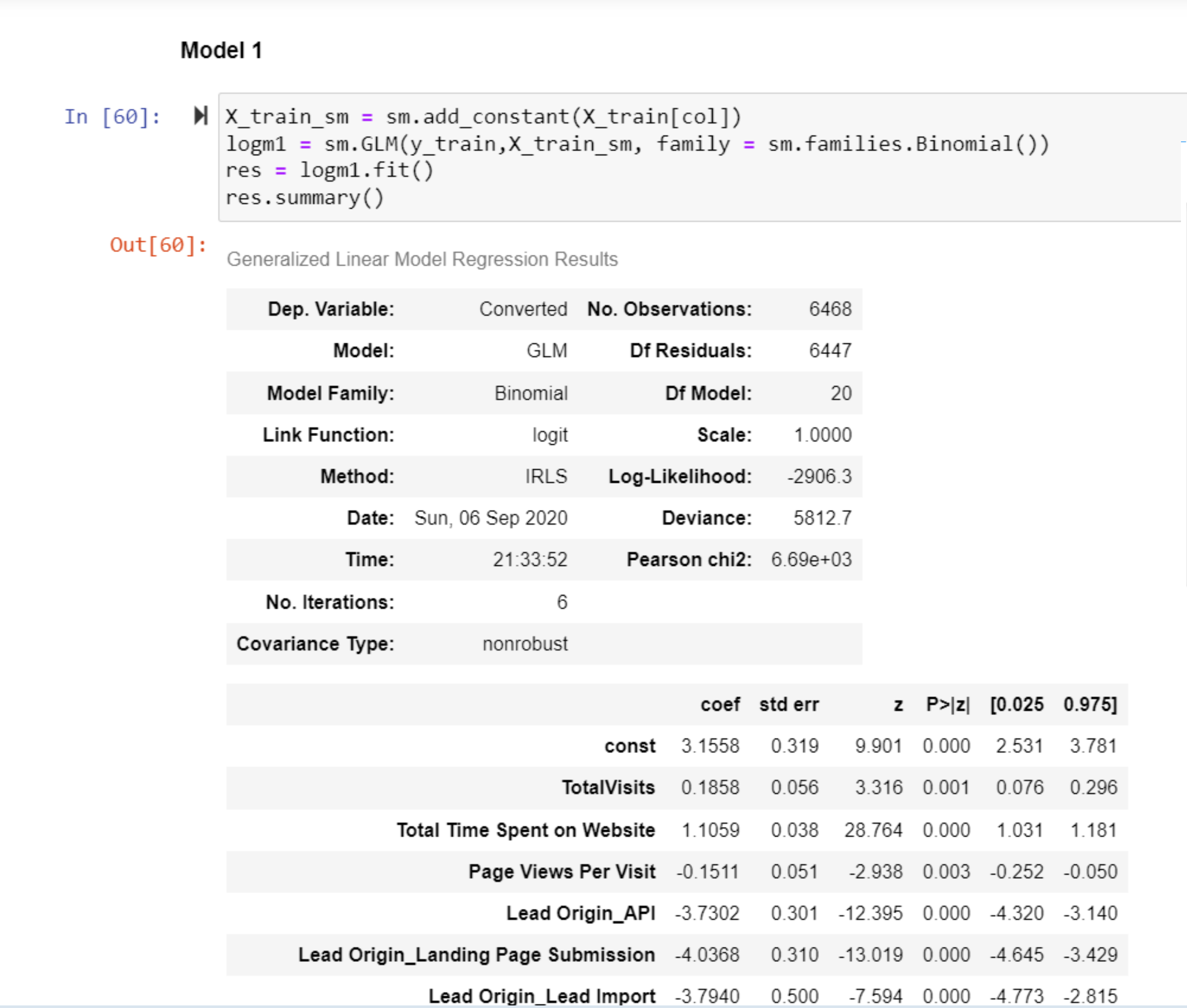
Correlation

In [51]:

```
# Finding the Correlation using HeatMap
plt.figure(figsize = (20, 10))
mask = np.zeros(X_train.corr().shape, dtype=bool)
mask[np.triu_indices(len(mask))] = True
sns.heatmap(X_train.corr(), annot = True, vmin=-1,cmap='coolwarm',mask=mask)
plt.show()
```

TotalVisits
Total Time Spent on Website 0.2
Page Views Per Visit 0.48 0.32
Lead Origin_API -0.2 -0.21 -0.35
Lead Origin_Landing Page Submission 0.28 0.28 0.49 -0.84
Lead Origin_Lead Import -0.025 -0.038 -0.04 -0.056 -0.075
Lead Origin_Quick Add Form -0.001 0.04 -0.0021 -0.0099 -0.013 -0.00088
Lead Source_Direct Traffic 0.092 0.12 0.14 -0.44 0.53 -0.044 -0.0077
Lead Source_Google 0.096 0.21 0.2 0.018 0.078 -0.043 0.019 -0.41
Lead Source_Olark Chat -0.29 -0.38 -0.5 0.62 -0.52 -0.035 -0.0061 -0.3 -0.33
Lead Source_Other -0.099 -0.12 -0.19 -0.18 -0.31 0.21 -0.004 -0.2 -0.22 -0.16
Last Activity_Converted to Lead -0.064 -0.0059 -0.058 -0.013 0.049 -0.016 -0.0028 0.067 0.033 -0.11 -0.057
Last Activity_Email Bounced -0.042 -0.025 -0.039 -0.054 0.064 -0.0023 0.063 0.094 -0.059 -0.023 -0.031 -0.044
Last Activity_Olark Chat Conversation -0.14 -0.2 -0.23 0.37 -0.31 -0.024 -0.0043 -0.18 -0.092 0.44 -0.089 -0.076 -0.068
Last Activity_Other -0.0022 -0.027 0.00081 -0.0031 0.013 -0.018 -0.0032 -0.014 0.027 0.0032 -0.0017 -0.058 -0.051 -0.09
Last Activity_Page Visited on Website 0.22 0.016 0.12 -0.068 0.093 -0.019 -0.0034 0.062 0.019 -0.099 -0.03 -0.06 -0.054 -0.094 -0.071
Last Activity_SMS Sent 0.001 0.12 0.07 -0.12 0.063 -0.012 -0.008 0.013 0.019 -0.12 0.068 -0.14 -0.13 -0.22 -0.17 -0.18
Specialization_Business Administration 0.05 0.073 0.064 -0.13 0.14 -0.0046 -0.0027 0.057 0.034 -0.088 -0.016 0.0018 -0.015 -0.062 -0.00098 0.012 0.02
Specialization_Finance Management -0.21 -0.24 -0.34 0.58 -0.6 0.039 -0.012 -0.32 -0.036 0.41 0.063 0.0036 -0.0073 0.29 -0.014 -0.066 -0.085 -0.21
Specialization_Human Resource Management 0.064 0.072 0.085 -0.17 0.15 -0.0067 -0.0039 0.072 0.042 -0.13 0.023 -0.0078 -0.025 -0.1 0.03 0.032 0.013 -0.068 -0.3
Specialization_IT Projects Management 0.033 0.029 0.075 -0.13 0.15 -0.014 -0.0025 0.12 0.0014 -0.09 -0.05 -0.0077 0.072 -0.052 -0.0011 0.019 0.0023 -0.044 -0.19 -0.064
Specialization_Marketing Management 0.015 0.069 0.053 -0.14 0.12 -0.022 -0.0039 0.085 0.039 -0.1 0.014 -0.0021 -0.013 -0.072 -0.0031 0.00052 0.023 -0.067 -0.3 -0.098 -0.064
Specialization_Other 0.16 0.11 0.23 -0.31 0.33 -0.014 0.024 0.16 -0.044 -0.21 -0.059 0.011 0.015 -0.14 -0.013 0.06 0.05 -0.11 -0.49 -0.16 -0.1 -0.16
TotalVisits
t on Website
ws Per Visit
d Origin_API
Submission
Lead Import
ck Add Form
Direct Traffic
rce_Google
_Olark Chat
ource_Other
rted to Lead
ail Bounced
onversation
ctivity_Other
l on Website
y_SMS Sent
dministration
Management
Management
Management
Management
ation_Other
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0

Model Creation



Model 1

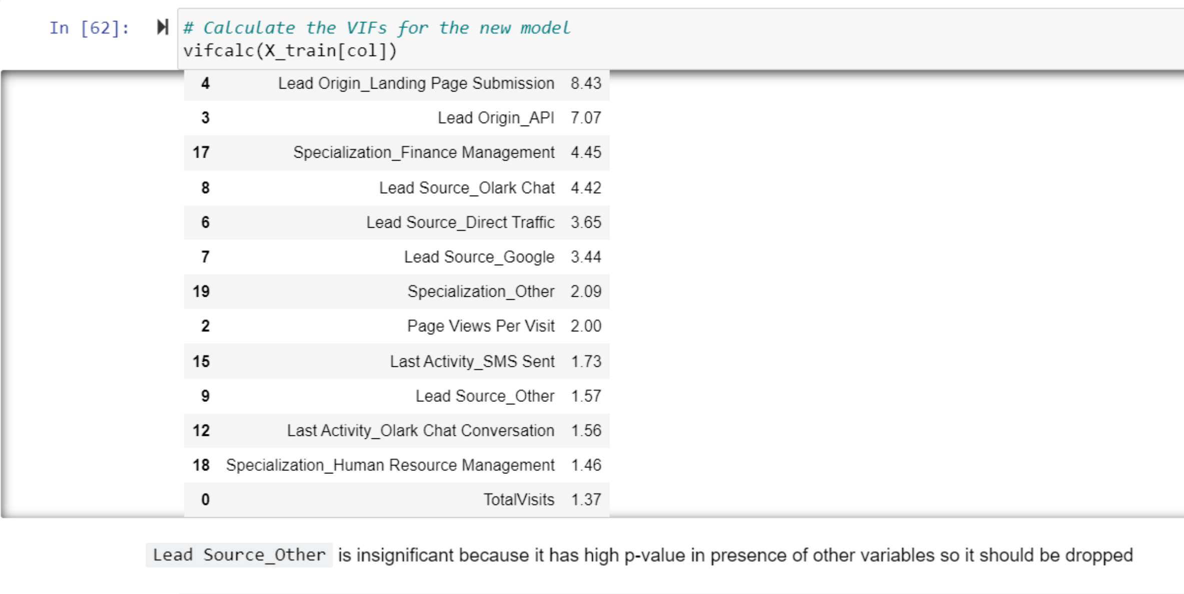
```
In [60]: X_train_sm = sm.add_constant(X_train[col])
         logm1 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
         res = logm1.fit()
         res.summary()
```

Out[60]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6468
Model:	GLM	**Df Residuals:**	6447
Model Family:	Binomial	**Df Model:**	20
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2906.3
Date:	Sun, 06 Sep 2020	**Deviance:**	5812.7
Time:	21:33:52	**Pearson chi2:**	6.69e+03
No. Iterations:	6		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	3.1558	0.319	9.901	0.000	2.531	3.781
TotalVisits	0.1858	0.056	3.316	0.001	0.076	0.296
Total Time Spent on Website	1.1059	0.038	28.764	0.000	1.031	1.181
Page Views Per Visit	-0.1511	0.051	-2.938	0.003	-0.252	-0.050
Lead Origin_API	-3.7302	0.301	-12.395	0.000	-4.320	-3.140
Lead Origin_Landing Page Submission	-4.0368	0.310	-13.019	0.000	-4.645	-3.429
Lead Origin_Lead Import	-3.7940	0.500	-7.594	0.000	-4.773	-2.815

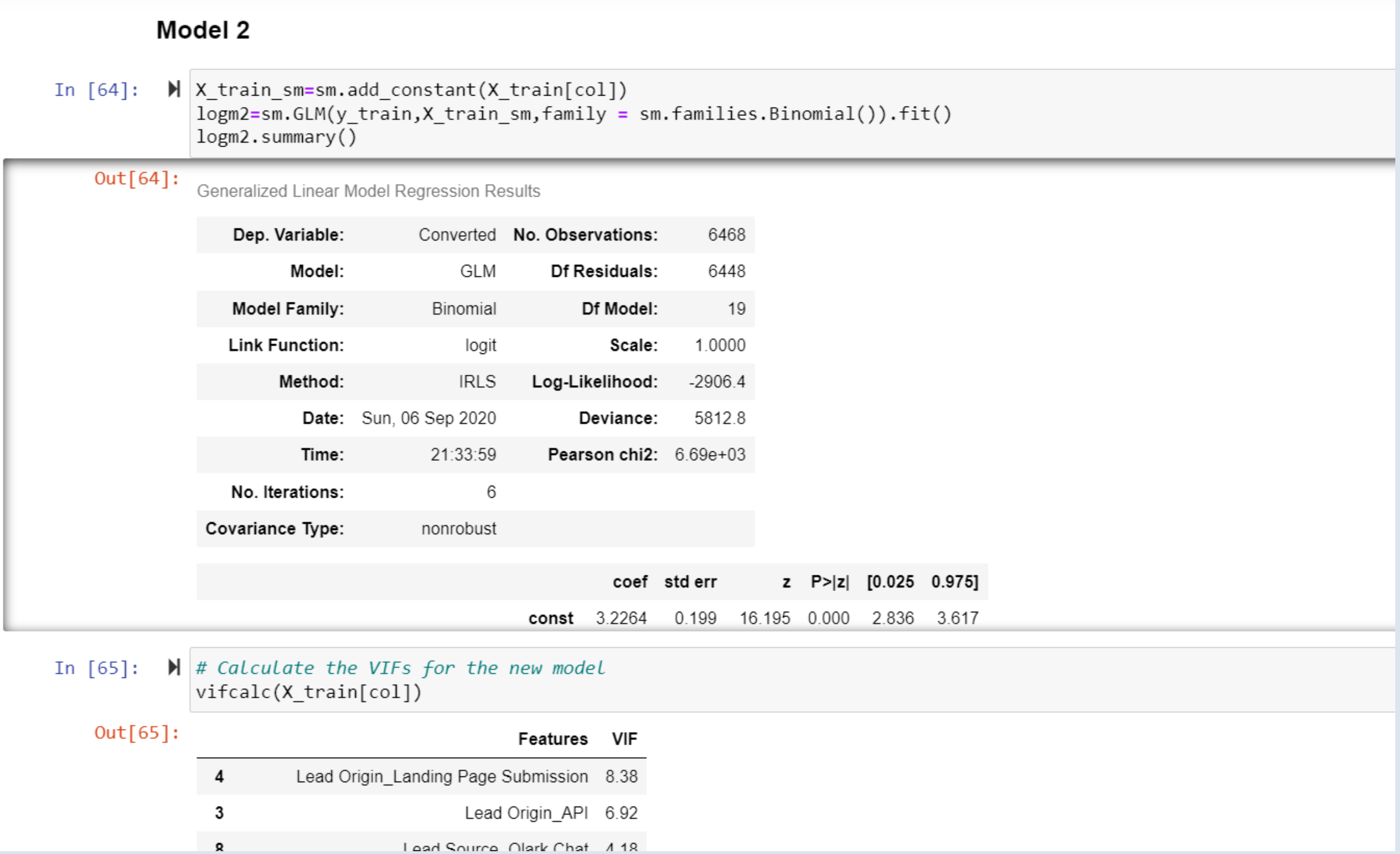


```
In [62]: # Calculate the VIFs for the new model
         vifcalc(X_train[col])
```

4	Lead Origin_Landing Page Submission	8.43
3	Lead Origin_API	7.07
17	Specialization_Finance Management	4.45
8	Lead Source_Olark Chat	4.42
6	Lead Source_Direct Traffic	3.65
7	Lead Source_Google	3.44
19	Specialization_Other	2.09
2	Page Views Per Visit	2.00
15	Last Activity_SMS Sent	1.73
9	Lead Source_Other	1.57
12	Last Activity_Olark Chat Conversation	1.56
18	Specialization_Human Resource Management	1.46
0	TotalVisits	1.37

`Lead Source_Other` is insignificant because it has high p-value in presence of other variables so it should be dropped

Model Creation



Model 2

In [64]:
```
X_train_sm=sm.add_constant(X_train[col])
logm2=sm.GLM(y_train,X_train_sm,family = sm.families.Binomial()).fit()
logm2.summary()
```

Out[64]:

Generalized Linear Model Regression Results

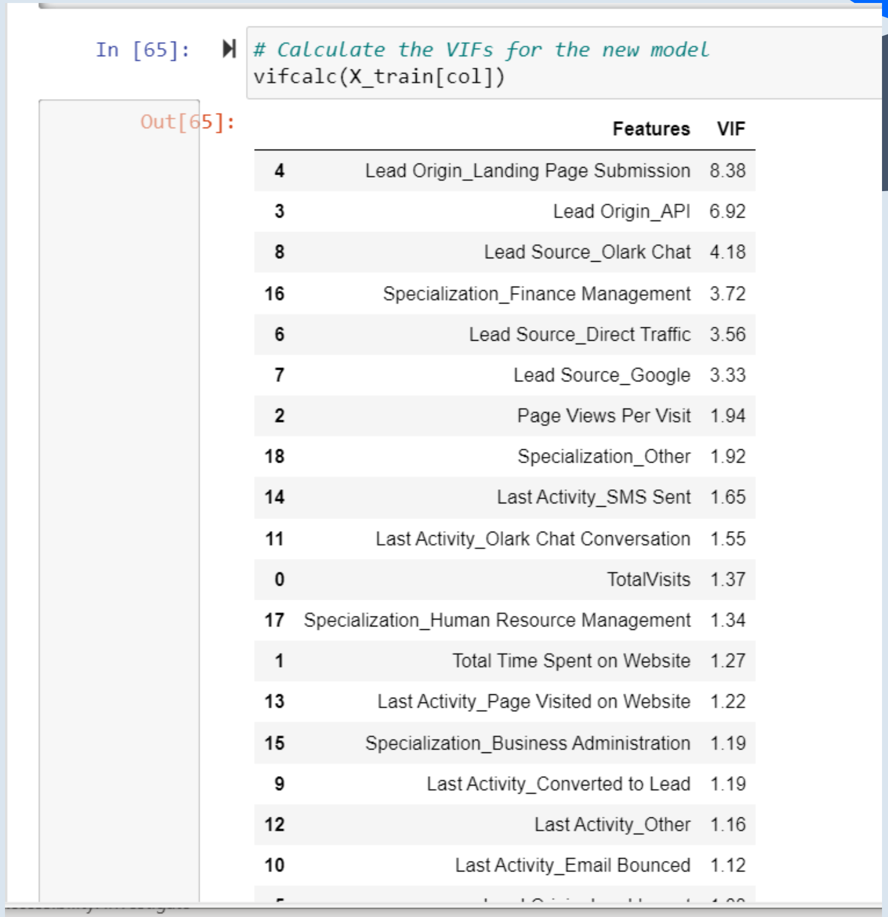
Dep. Variable:	Converted	**No. Observations:**	6468
Model:	GLM	**Df Residuals:**	6448
Model Family:	Binomial	**Df Model:**	19
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2906.4
Date:	Sun, 06 Sep 2020	**Deviance:**	5812.8
Time:	21:33:59	**Pearson chi2:**	6.69e+03
No. Iterations:	6		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	3.2264	0.199	16.195	0.000	2.836	3.617

In [65]:
```
# Calculate the VIFs for the new model
vifcalc(X_train[col])
```

Out[65]:

	Features	VIF
4	Lead Origin_Landing Page Submission	8.38
3	Lead Origin_API	6.92
8	Lead Source_Olark Chat	4.18



In [65]:
```
# Calculate the VIFs for the new model
vifcalc(X_train[col])
```

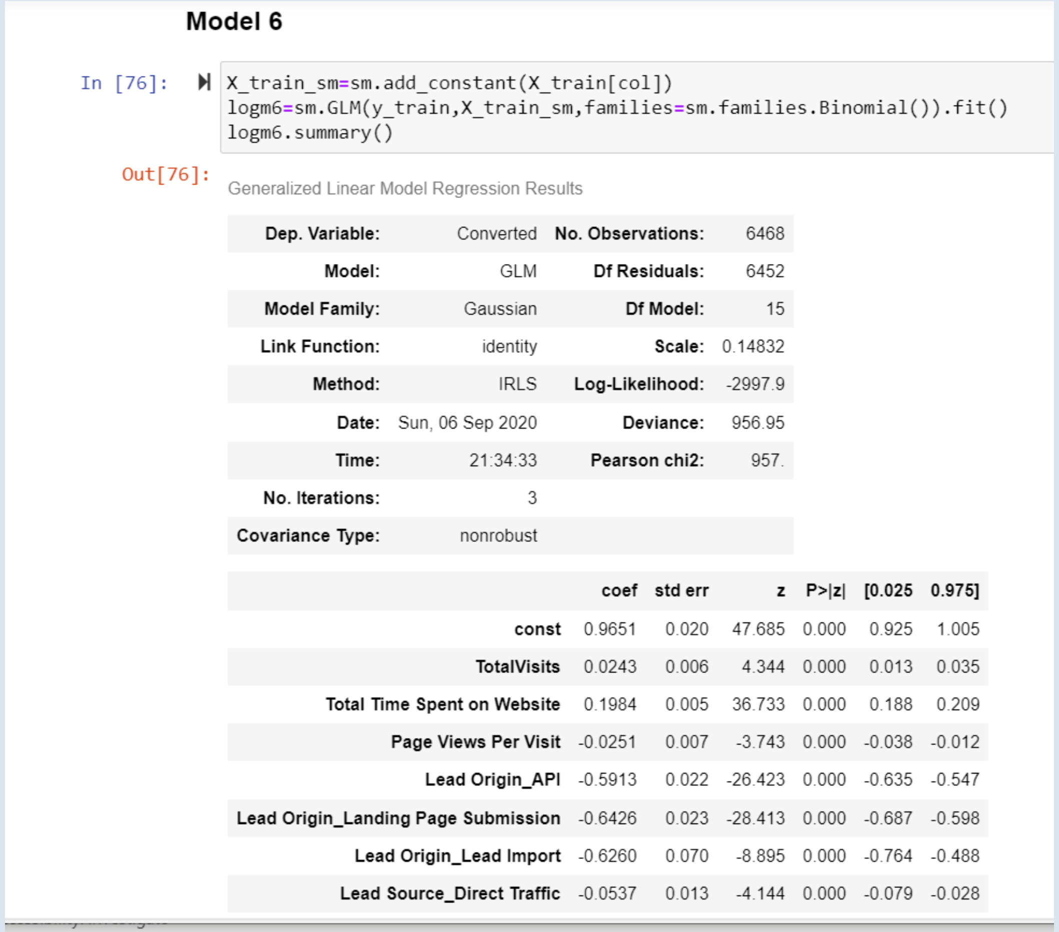
Out[65]:

	Features	VIF
4	Lead Origin_Landing Page Submission	8.38
3	Lead Origin_API	6.92
8	Lead Source_Olark Chat	4.18
16	Specialization_Finance Management	3.72
6	Lead Source_Direct Traffic	3.56
7	Lead Source_Google	3.33
2	Page Views Per Visit	1.94
18	Specialization_Other	1.92
14	Last Activity_SMS Sent	1.65
11	Last Activity_Olark Chat Conversation	1.55
0	TotalVisits	1.37
17	Specialization_Human Resource Management	1.34
1	Total Time Spent on Website	1.27
13	Last Activity_Page Visited on Website	1.22
15	Specialization_Business Administration	1.19
9	Last Activity_Converted to Lead	1.19
12	Last Activity_Other	1.16
10	Last Activity_Email Bounced	1.12

Similar to this models, we have explored 6 different model with various attributes and concluded with the final model that had VIF values less than 5.

Model Creation

- Final Model for validation



Model 6

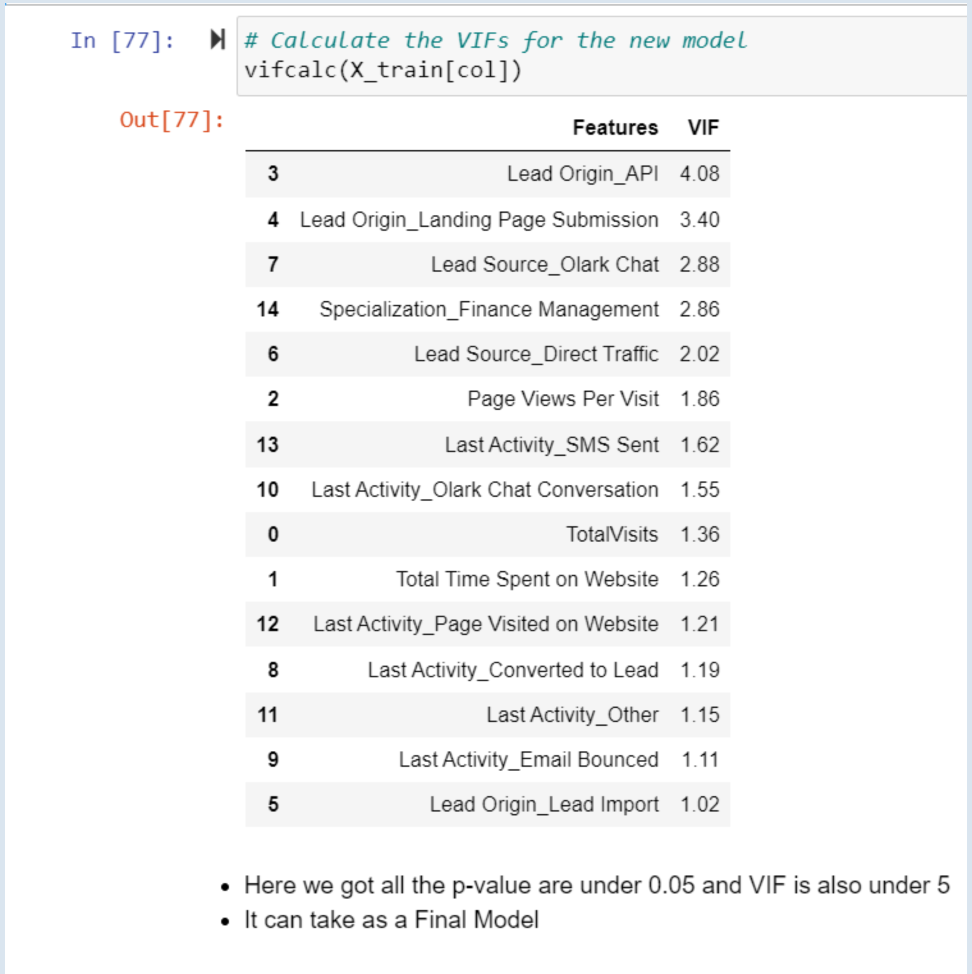
In [76]:
```
X_train_sm=sm.add_constant(X_train[col])
logm6=sm.GLM(y_train,X_train_sm,families=sm.families.Binomial()).fit()
logm6.summary()
```

Out[76]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6468
Model:	GLM	**Df Residuals:**	6452
Model Family:	Gaussian	**Df Model:**	15
Link Function:	identity	**Scale:**	0.14832
Method:	IRLS	**Log-Likelihood:**	-2997.9
Date:	Sun, 06 Sep 2020	**Deviance:**	956.95
Time:	21:34:33	**Pearson chi2:**	957.
No. Iterations:	3		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.9651	0.020	47.685	0.000	0.925	1.005
TotalVisits	0.0243	0.006	4.344	0.000	0.013	0.035
Total Time Spent on Website	0.1984	0.005	36.733	0.000	0.188	0.209
Page Views Per Visit	-0.0251	0.007	-3.743	0.000	-0.038	-0.012
Lead Origin_API	-0.5913	0.022	-26.423	0.000	-0.635	-0.547
Lead Origin_Landing Page Submission	-0.6426	0.023	-28.413	0.000	-0.687	-0.598
Lead Origin_Lead Import	-0.6260	0.070	-8.895	0.000	-0.764	-0.488
Lead Source_Direct Traffic	-0.0537	0.013	-4.144	0.000	-0.079	-0.028



In [77]:
```
# Calculate the VIFs for the new model
vifcalc(X_train[col])
```

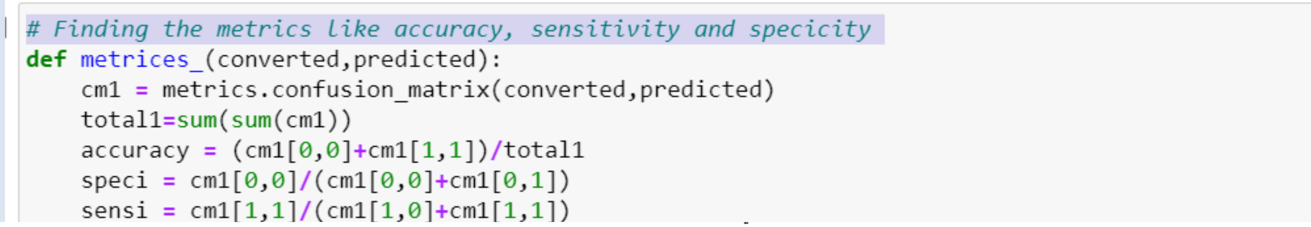
Out[77]:

	Features	VIF
3	Lead Origin_API	4.08
4	Lead Origin_Landing Page Submission	3.40
7	Lead Source_Olark Chat	2.88
14	Specialization_Finance Management	2.86
6	Lead Source_Direct Traffic	2.02
2	Page Views Per Visit	1.86
13	Last Activity_SMS Sent	1.62
10	Last Activity_Olark Chat Conversation	1.55
0	TotalVisits	1.36
1	Total Time Spent on Website	1.26
12	Last Activity_Page Visited on Website	1.21
8	Last Activity_Converted to Lead	1.19
11	Last Activity_Other	1.15
9	Last Activity_Email Bounced	1.11
5	Lead Origin_Lead Import	1.02

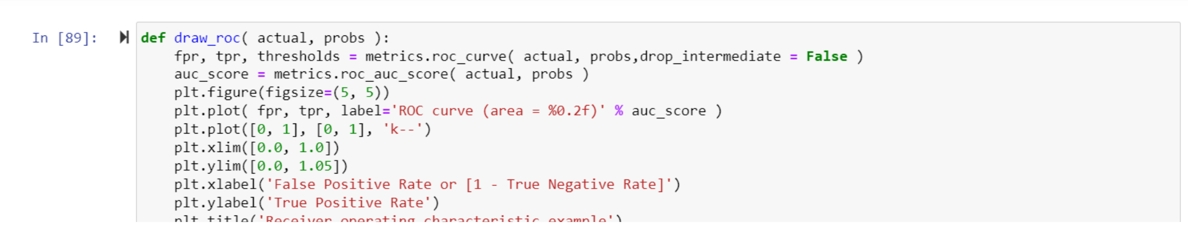
- Here we got all the p-value are under 0.05 and VIF is also under 5
- It can take as a Final Model

Model Evaluation

- Finding the metrics like accuracy, sensitivity and specificity



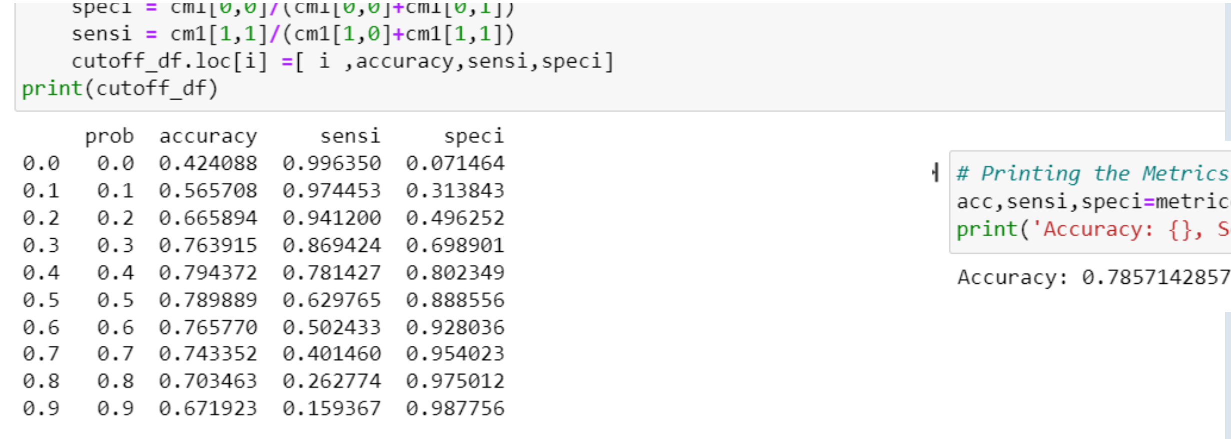
```
# Finding the metrics like accuracy, sensitivity and specicity
def metrices_(converted,predicted):
    cm1 = metrics.confusion_matrix(converted,predicted)
    total1=sum(sum(cm1))
    accuracy = (cm1[0,0]+cm1[1,1])/total1
    speci = cm1[0,0]/(cm1[0,0]+cm1[0,1])
    sensi = cm1[1,1]/(cm1[1,0]+cm1[1,1])
```



```
In [89]:  def draw_roc( actual, probs ):
              fpr, tpr, thresholds = metrics.roc_curve( actual, probs,drop_intermediate = False )
              auc_score = metrics.roc_auc_score( actual, probs )
              plt.figure(figsize=(5, 5))
              plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score )
              plt.plot([0, 1], [0, 1], 'k--')
              plt.xlim([0.0, 1.0])
              plt.ylim([0.0, 1.05])
              plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
              plt.ylabel('True Positive Rate')
              plt.title('Receiver operating characteristic example')
```

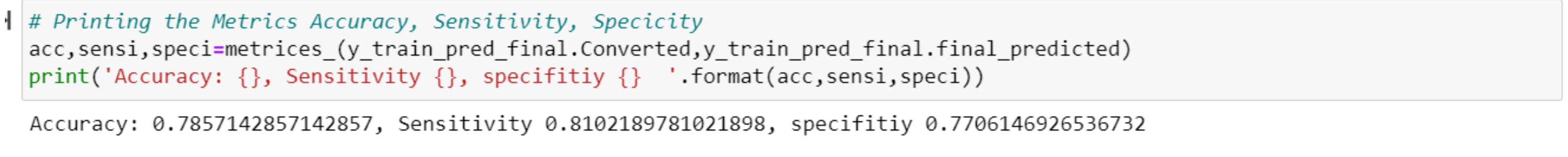
Conclusion :-

- We have noted that the variables that important the most in the potential buyers are:
  - The total time spend on the Website.
  - Total number of visits.
  - When the lead source was: a. Google b. Direct traffic c. Organic search d. Olark Chat
  - When the last activity was: a. SMS b. Olark chat conversation
  - When the lead origin is Lead add format.



```
    speci = cm1[0,0]/(cm1[0,0]+cm1[0,1])
    sensi = cm1[1,1]/(cm1[1,0]+cm1[1,1])
    cutoff_df.loc[i] =[ i ,accuracy,sensi,speci]
print(cutoff_df)

     prob  accuracy     sensi     speci
0.0   0.0  0.424088  0.996350  0.071464
0.1   0.1  0.565708  0.974453  0.313843
0.2   0.2  0.665894  0.941200  0.496252
0.3   0.3  0.763915  0.869424  0.698901
0.4   0.4  0.794372  0.781427  0.802349
0.5   0.5  0.789889  0.629765  0.888556
0.6   0.6  0.765770  0.502433  0.928036
0.7   0.7  0.743352  0.401460  0.954023
0.8   0.8  0.703463  0.262774  0.975012
0.9   0.9  0.671923  0.159367  0.987756
```



```
# Printing the Metrics Accuracy, Sensitivity, Specicity
acc,sensi,speci=metrices_(y_train_pred_final.Converted,y_train_pred_final.final_predicted)
print('Accuracy: {}, Sensitivity {}, specifitiy {}  '.format(acc,sensi,speci))

Accuracy: 0.7857142857142857, Sensitivity 0.8102189781021898, specifitiy 0.7706146926536732
```

Conclusion

The variables that mattered the most in the potential buyers are (In descending order) :

1. The total time spend on the Website.
2. Total number of visits.
3. When the lead source is a. Google b. Direct traffic c. Organic search d. Olark website
4. When the last activity is a. SMS b. Olark chat conversation
5. When the lead origin is Lead add format.

Keeping these in mind the X Education can flourish as they have a very high chance to get almost all the potential buyers to change their mind and buy their courses

Thank you