LEAD SCORING CASE STUDY

Vishnu M Menon
menon.vishnum@gmail.com,
mob# 97312-88115

Dharmesh Bhuta
dharmeshbhuta85@gmail.com,
mob# 98197-38519

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.

X Education has appointed you to help them select the most promising leads, i.e. the leads that are most likely to convert into paying customers. The company requires you to build a model wherein you need to assign a lead score to each of the leads such that the customers with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance. The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%.

STEPS TO PERFORM CASE STUDY ANALYSIS

- Import Data
- Inspecting the Dataframe
- Data Preparation
- Test-Train Split
- Feature Scaling
- Looking for Correlation
- Model Building
- Feature Scaling using RFE
- Plotting the ROC Curve
- Finding Optimal Cutoff Point
- Making Predictions to the Test set

IMPORT DATA

Imported “Leads.csv”

The raw data contains 9240 rows

The raw data contains 37 columns (combination of Original + Sales info)

```
In [3]: raw_data = pd.read_csv("Leads.csv")
        pd.options.display.max_columns = None
        raw_data.head()
```

Out[3]:

	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	Last Activity	Country	Specialization	How did you hear about X Education	c
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.00	Page Visited on Website	NaN	Select	Select	U
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.50	Email Opened	India	Select	Select	U
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.00	Email Opened	India	Business Administration	Select	
	0cc2df48-7cf4-		Landing	Direct									Media and		

```
In [4]: raw_data.shape
Out[4]: (9240, 37)
```

INSPECTING THE DATAFRAME

Checking on the data types of the variables

```
In [5]: raw_data.dtypes

Out[5]: Prospect ID                                        object
        Lead Number                                         int64
        Lead Origin                                        object
        Lead Source                                        object
        Do Not Email                                       object
        Do Not Call                                        object
        Converted                                           int64
        TotalVisits                                       float64
        Total Time Spent on Website                         int64
        Page Views Per Visit                              float64
        Last Activity                                      object
        Country                                            object
        Specialization                                     object
        How did you hear about X Education                 object
        What is your current occupation                    object
        What matters most to you in choosing a course      object
        Search                                             object
        Magazine                                           object
        Newspaper Article                                  object
        X Education Forums                                 object
        Newspaper                                          object
        Digital Advertisement                              object
        Through Recommendations                            object
        Receive More Updates About Our Courses             object
        Tags                                               object
        Lead Quality                                       object
        Update me on Supply Chain Content                  object
        Get updates on DM Content                          object
        Lead Profile                                       object
        City                                               object
        Asymmetrique Activity Index                        object
        Asymmetrique Profile Index                         object
        Asymmetrique Activity Score                       float64
        Asymmetrique Profile Score                        float64
        I agree to pay the amount through cheque           object
        A free copy of Mastering The Interview             object
        Last Notable Activity                              object
        dtype: object
```

Checking on the null value percent

```
In [7]: raw_data.isnull().sum()/len(raw_data) * 100

Out[7]: Prospect ID                                         0.000000
        Lead Number                                         0.000000
        Lead Origin                                         0.000000
        Lead Source                                         0.389610
        Do Not Email                                        0.000000
        Do Not Call                                         0.000000
        Converted                                           0.000000
        TotalVisits                                         1.482684
        Total Time Spent on Website                         0.000000
        Page Views Per Visit                                1.482684
        Last Activity                                       1.114719
        Country                                            26.634199
        Specialization                                     15.562771
        How did you hear about X Education                 23.885281
        What is your current occupation                    29.112554
        What matters most to you in choosing a course      29.318182
        Search                                              0.000000
        Magazine                                            0.000000
        Newspaper Article                                   0.000000
        X Education Forums                                  0.000000
        Newspaper                                           0.000000
        Digital Advertisement                               0.000000
        Through Recommendations                             0.000000
        Receive More Updates About Our Courses              0.000000
        Tags                                               36.287879
        Lead Quality                                       51.590909
        Update me on Supply Chain Content                   0.000000
        Get updates on DM Content                           0.000000
        Lead Profile                                       29.318182
        City                                               15.367965
        Asymmetrique Activity Index                        45.649351
        Asymmetrique Profile Index                         45.649351
        Asymmetrique Activity Score                        45.649351
        Asymmetrique Profile Score                         45.649351
        I agree to pay the amount through cheque            0.000000
        A free copy of Mastering The Interview              0.000000
        Last Notable Activity                               0.000000
        dtype: float64
```

```
In [14]: raw_data["Specialization"].value_counts() #18

Out[14]: Select                                1942
         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 [10]: raw_data["Do Not Email"].value_counts()

Out[10]: No     8506
         Yes     734
         Name: Do Not Email, dtype: int64
```

```
In [11]: raw_data["Do Not Call"].value_counts()

Out[11]: No     9238
         Yes       2
         Name: Do Not Call, dtype: int64
```

```
In [8]: raw_data["Lead Origin"].value_counts()

Out[8]: Landing Page Submission    4886
        API                        3580
        Lead Add Form               718
        Lead Import                  55
        Quick Add Form                1
        Name: Lead Origin, dtype: int64
```

DATA PREPARATION

Mode for one column

In [16]:
```
# What matters most to you in choosing a course
raw_data["What matters most to you in choosing a course"].value_counts()
```

Out[16]:
```
Better Career Prospects      6528
Flexibility & Convenience       2
Other                           1
Name: What matters most to you in choosing a course, dtype: int64
```

In [17]:
```
raw_data["What matters most to you in choosing a course"].isnull().sum()
```

Out[17]:
```
2709
```

In [18]:
```
raw_data["What matters most to you in choosing a course"].fillna(raw_data["What matters most to you in choosing a course"].mode()
```

handling "Select" values

In [24]:
```
raw_data = raw_data.replace("Select", np.NaN)
```

Remove null values

In [25]:
```
# What is your current occupation
raw_data["What is your current occupation"].value_counts()
```

Out[25]:
```
Unemployed              5600
Working Professional     706
Student                  210
Other                     16
Housewife                 10
Businessman                8
Name: What is your current occupation, dtype: int64
```

In [26]:
```
raw_data["What is your current occupation"].isnull().sum()/len(raw_data) * 100
```

Out[26]:
```
29.11255411255411
```

In [27]:
```
raw_data = raw_data.dropna(axis=0, subset=["What is your current occupation"])
raw_data["What is your current occupation"].isnull().sum()/len(raw_data) * 100
```

Out[27]:
```
0.0
```

Setting up dummy variables

Dummy var1

In [63]:
```
raw_data.shape
```
Out[63]:
```
(6420, 26)
```

In [64]:
```
raw_data["Lead Origin"].value_counts()
```
Out[64]:
```
Landing Page Submission    3644
API                        2140
Others                      636
Name: Lead Origin, dtype: int64
```

In [65]:
```
# Creating dummy variables for the remaining categorical variables and dropping the level with big names.
# Creating dummy variables for the variable 'MultipleLines'
ml = pd.get_dummies(raw_data["Lead Origin"], prefix="Lead Origin")

# Dropping MultipleLines_No phone service column
ml1 = ml.drop(["Lead Origin_Others"], 1)

#Adding the results to the master dataframe
raw_data = pd.concat([raw_data,ml1], axis=1)
```

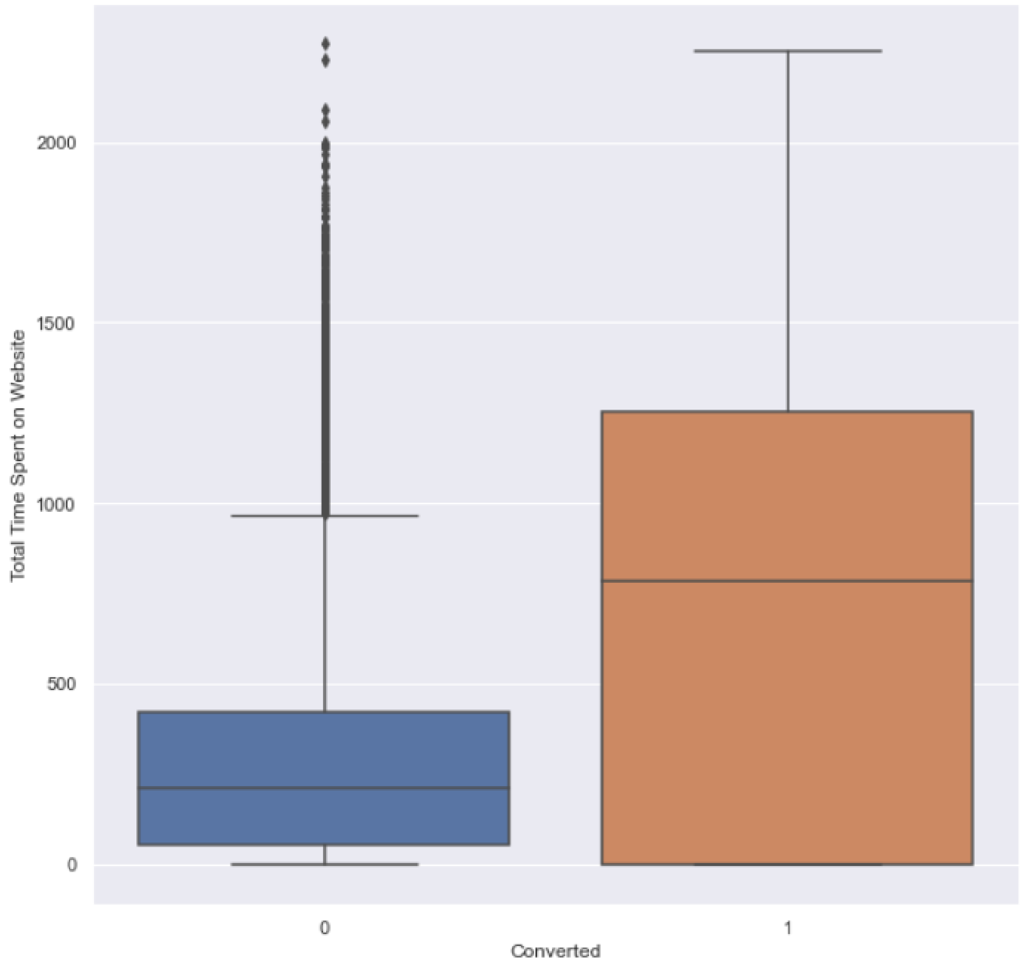
In [66]:
```
raw_data.shape
```
Out[66]:
```
(6420, 28)
```

DATA PREPARATION

UNIVARIATE ANALYSIS

Total Time Spent on Website

```
In [160]: plt.figure(figsize = [10,10])
          sns.boxplot(data=raw_data, x="Converted", y="Total Time Spent on Website")
          plt.show()
```

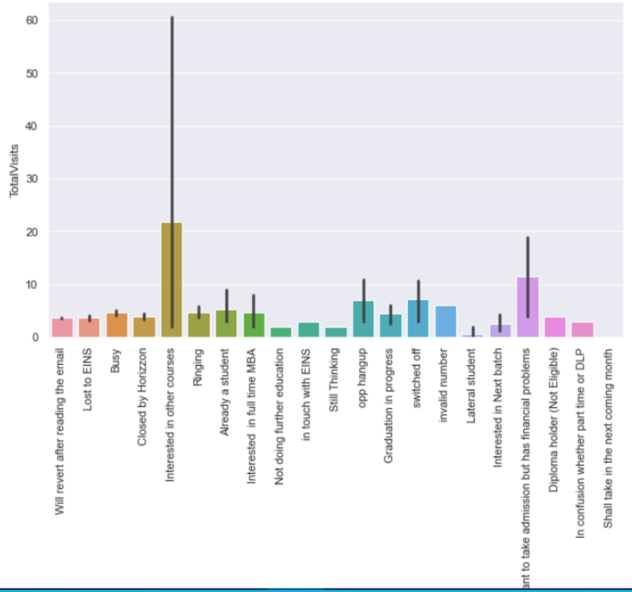


<u>Inference:</u>

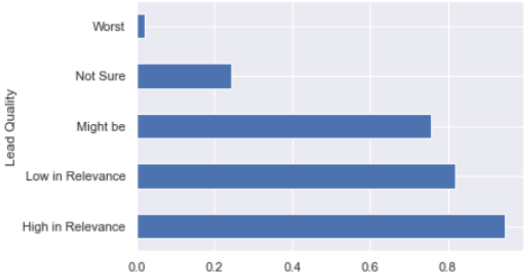
As we can clearly see from above plot those who have been converted to take up the course have spent considerable amount of time on the website as compared to those who have not.

As we can see, apart from those whose field of specialization is Unknown, there are 3 primary fields of people who are usually getting converted to the course and hence they should be prioritized for the leads.

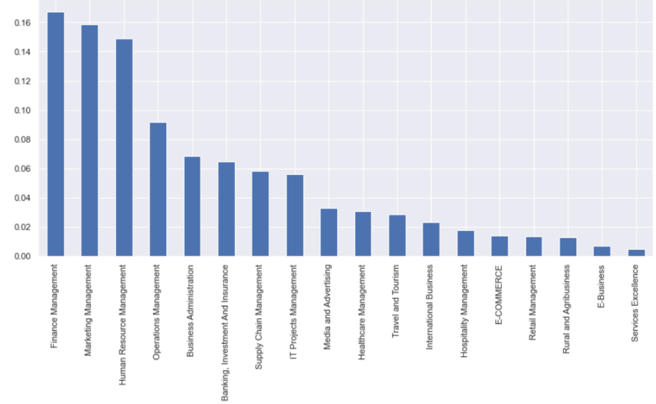
In [33]:
```
## Tags

plt.figure(figsize = [10,6])
sns.barplot(data=TARGET1, x = "Tags", y = 'TotalVisits')
plt.xticks(rotation=90)
plt.show()
```



In [30]:
```
#plot the bar graph of Lead Quality with average of Converted Lead.
raw_data.groupby("Lead Quality")["Converted"].mean().plot.barh()
plt.show()
```



In [23]:
```
plt.figure(figsize=[14,6])
TARGET1[~(TARGET1.Specialization=='Others')].Specialization.value_counts(normalize=True).plot.bar()
plt.show()
```



As we can see from above those with tags category as "Interested in other courses" and probably those with some finanacial need are keen to take up courses and hence making more visits to the website should also be pursued more as they have better chances of being converted.

BI-VARIATE ANALYSIS

TEST-TRAIN SPLIT

The dependent variable to be predicted is "Converted"

Train set size = 70% of the raw_data

Test set size = 30% of the raw_data

```
In [81]: from sklearn.model_selection import train_test_split
```

```
In [82]: # Putting feature variable to X
         X = raw_data.drop(["Converted"], axis=1)

         X.head()
```

Out[82]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Search	Magazine	Newspaper Article	X Education Forums	Newspaper	Digital Advertisement	Through Recommendations	Receive More Updates About Our Courses	Update me on Supply Chain Content	Ge update on DI Conten
0	0	0	0.0	0	0.0	0	0	0	0	0	0	0	0	0	
1	0	0	5.0	674	2.5	0	0	0	0	0	0	0	0	0	
2	0	0	2.0	1532	2.0	0	0	0	0	0	0	0	0	0	
3	0	0	1.0	305	1.0	0	0	0	0	0	0	0	0	0	
4	0	0	2.0	1428	1.0	0	0	0	0	0	0	0	0	0	

```
In [83]: # Putting response variable to y
         y = raw_data["Converted"]

         y.head()
```

```
Out[83]: 0    0
         1    0
         2    1
         3    0
         4    1
         Name: Converted, dtype: int64
```

```
In [84]: # Splitting the data into train and test
         X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3, random_state=100)
```

FEATURE SCALING

Performing feature scaling using StandardScaler

Post Data Preparation, we are left with 3 columns for scaling:

- TotalVisits

- Time Spent on Website

- Page Views Per Visit

We have almost 48-49% conversion rate

In [85]:
```
from sklearn.preprocessing import StandardScaler
```

In [86]:
```
scaler = StandardScaler()

X_train[["TotalVisits","Total Time Spent on Website","Page Views Per Visit"]] = scaler.fit_transform(X_train[["TotalVisits","Tota
X_train.head()
```

Out[86]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Search	Magazine	Newspaper Article	X Education Forums	Newspaper	Digital Advertisement	Through Recommendations	Receive More Updates About Our Courses	Update me on Supply Chain Content
6630	0	0	-0.446432	-0.601041	-0.232526	0	0	0	0	0	0	0	0	0
8176	0	0	-0.446432	-0.320349	-0.719960	0	0	0	0	0	0	0	0	0
4978	0	0	0.117760	1.056467	0.742343	0	0	0	0	0	0	0	0	0
7457	0	0	0.399855	-0.595712	1.229777	0	0	0	0	0	0	0	0	0
4383	0	0	0.681951	0.001205	1.717211	0	0	0	0	0	0	0	0	0

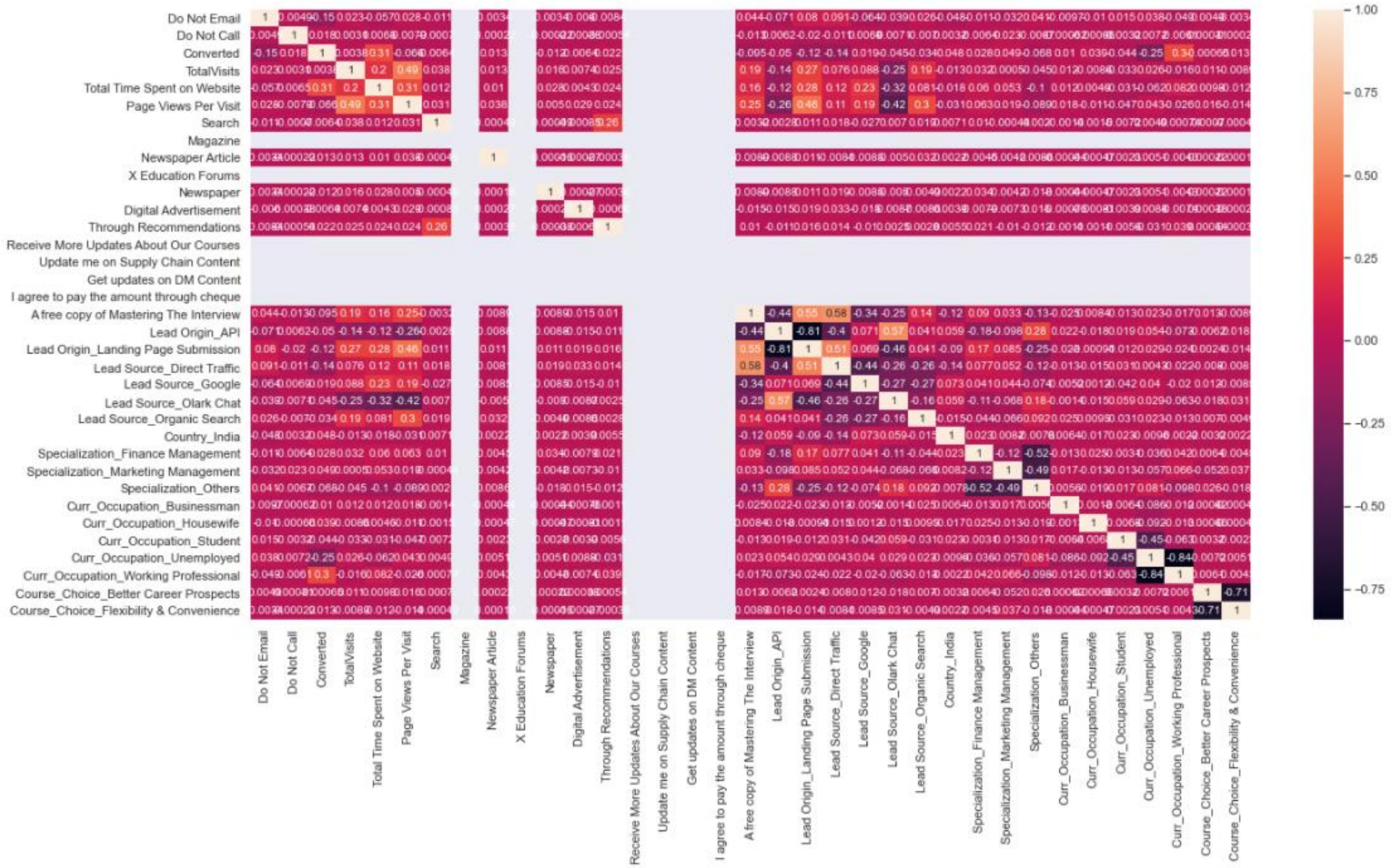
In [87]:
```
### Checking the converted Rate
converted = (sum(raw_data["Converted"])/len(raw_data["Converted"].index))*100
converted
```

Out[87]: 48.14641744548286

LOOKING FOR CORRELATION

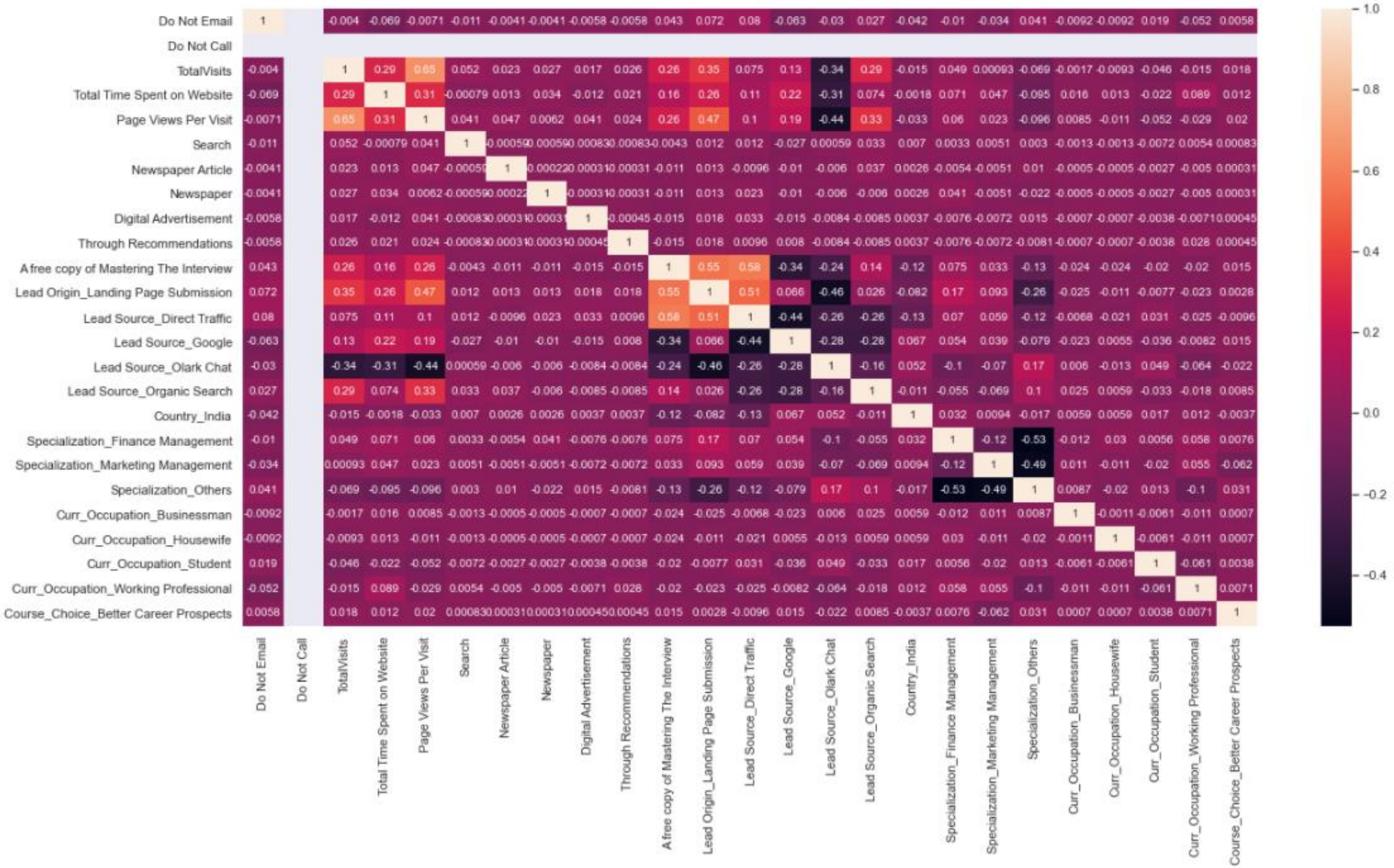
On performing correlation on the 35 variables

We found few variables with no correlation value and decided to get rid of them



CORRELATION CONT.

Post removing 10 variables with highest and no correlation, we ended up with 24 independent variables to go ahead with our regression model building



MODEL BUILDING

```
In [96]: import statsmodels.api as sm
```

```
In [97]: # Logistic Regression Model
X_train_sm = sm.add_constant(X_train)
log_res_m1 = sm.GLM(y_train, X_train_sm, family=sm.families.Binomial())
log_res_m1.fit().summary()
```

Out[97]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	4494
Model:	GLM	Df Residuals:	4469
Model Family:	Binomial	Df Model:	24
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-2301.4
Date:	Tue, 11 Jan 2022	Deviance:	4602.8
Time:	13:05:16	Pearson chi2:	4.75e+03
No. Iterations:	21		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	2.5180	1.473	1.709	0.087	-0.369	5.405

We started to build our first model using StatsModels as the dependent variable is 0s and 1s

Using train set, after adding a constant

Family as Binomial because our outcome is binary (zeros and ones), proportions of "successes" and "failures" (values between 0 and 1), or their counts, we can use Binomial distribution

FEATURE SCALING USING RFE

Performed feature scaling using LogisticRegression library

Ran RFE with 20 variables as output

Split the columns into two sets

```
In [98]: from sklearn.linear_model import LogisticRegression

In [99]: log_reg_m1 = LogisticRegression()

In [100]: from sklearn.feature_selection import RFE
          rfe = RFE(log_reg_m1, 20)        # running RFE with 20 variables as output out of 25
          rfe = rfe.fit(X_train, y_train)

In [101]: rfe.support_

Out[101]: array([ True, False,  True,  True,  True, False,  True,  True,  True,
                  True, False,  True,  True,  True,  True,  True,  True,  True,
                  True, False, False,  True,  True,  True,  True])
```

```
In [102]: list(zip(X_train.columns, rfe.support_, rfe.ranking_))

Out[102]: [('Do Not Email', True, 1),
 ('Do Not Call', False, 6),
 ('TotalVisits', True, 1),
 ('Total Time Spent on Website', True, 1),
 ('Page Views Per Visit', True, 1),
 ('Search', False, 4),
 ('Newspaper Article', True, 1),
 ('Newspaper', True, 1),
 ('Digital Advertisement', True, 1),
 ('Through Recommendations', True, 1),
 ('A free copy of Mastering The Interview', False, 5),
 ('Lead Origin_Landing Page Submission', True, 1),
 ('Lead Source_Direct Traffic', True, 1),
 ('Lead Source_Google', True, 1),
 ('Lead Source_Olark Chat', True, 1),
 ('Lead Source_Organic Search', True, 1),
 ('Country_India', True, 1),
 ('Specialization_Finance Management', True, 1),
 ('Specialization_Marketing Management', True, 1),
 ('Specialization_Others', False, 3),
 ('Curr_Occupation_Businessman', False, 2),
 ('Curr_Occupation_Housewife', True, 1),
 ('Curr_Occupation_Student', True, 1),
 ('Curr_Occupation_Working Professional', True, 1),
 ('Course_Choice_Better Career Prospects', True, 1)]
```

```
In [103]: col = X_train.columns[rfe.support_]

In [104]: col

Out[104]: Index(['Do Not Email', 'TotalVisits', 'Total Time Spent on Website',
        'Page Views Per Visit', 'Newspaper Article', 'Newspaper',
        'Digital Advertisement', 'Through Recommendations',
        'Lead Origin_Landing Page Submission', 'Lead Source_Direct Traffic',
        'Lead Source_Google', 'Lead Source_Olark Chat',
        'Lead Source_Organic Search', 'Country_India',
        'Specialization_Finance Management',
        'Specialization_Marketing Management', 'Curr_Occupation_Housewife',
        'Curr_Occupation_Student', 'Curr_Occupation_Working Professional',
        'Course_Choice_Better Career Prospects'],
       dtype='object')

In [105]: X_train.columns[~rfe.support_]

Out[105]: Index(['Do Not Call', 'Search', 'A free copy of Mastering The Interview',
        'Specialization_Others', 'Curr_Occupation_Businessman'],
       dtype='object')
```

RE-BUILDING THE MODEL

Now we re-built the mode using 20 variables

Assessing the model with StatsModels

In [106]:
```
# Logistic Regression Model
X_train_sm2 = sm.add_constant(X_train[col])
log_res_m2 = sm.GLM(y_train, X_train_sm2, family=sm.families.Binomial())
res = log_res_m2.fit()
res.summary()
```

Out[106]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	4494
Model:	GLM	**Df Residuals:**	4473
Model Family:	Binomial	**Df Model:**	20
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2301.5
Date:	Tue, 11 Jan 2022	**Deviance:**	4603.0
Time:	13:05:16	**Pearson chi2:**	4.75e+03
No. Iterations:	21		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	2.4828	1.467	1.692	0.091	-0.393	5.359
Do Not Email	-1.2376	0.170	-7.278	0.000	-1.571	-0.904
TotalVisits	0.1747	0.049	3.602	0.000	0.080	0.270
Total Time Spent on Website	1.0770	0.044	24.228	0.000	0.990	1.164
Page Views Per Visit	-0.1828	0.055	-3.296	0.001	-0.291	-0.074
Newspaper Article	22.6095	4.82e+04	0.000	1.000	-9.44e+04	9.45e+04
Newspaper	-24.6263	4.82e+04	-0.001	1.000	-9.45e+04	9.44e+04
Digital Advertisement	-20.8031	3.4e+04	-0.001	1.000	-6.66e+04	6.66e+04
Through Recommendations	21.2192	3.24e+04	0.001	0.999	-6.34e+04	6.35e+04
Lead Origin_Landing Page Submission	-0.0916	0.106	-0.860	0.390	-0.300	0.117
Lead Source_Direct Traffic	-3.2210	0.188	-17.118	0.000	-3.590	-2.852
Lead Source_Google	-2.8123	0.173	-16.236	0.000	-3.152	-2.473
Lead Source_Olark Chat	-1.8660	0.160	-11.655	0.000	-2.180	-1.552
Lead Source_Organic Search	-2.9108	0.196	-14.849	0.000	-3.295	-2.527
Country_India	0.4769	0.235	2.029	0.042	0.016	0.937

REMOVING VARIABLES WITH HIGH P-VALUE

[107]:
```
# Removing variables with high p-value
col = col.drop(["Newspaper Article","Newspaper","Digital Advertisement","Curr_Occupation_Housewife",
                "Through Recommendations"], 1)
col
```

[107]:
```
Index(['Do Not Email', 'TotalVisits', 'Total Time Spent on Website',
       'Page Views Per Visit', 'Lead Origin_Landing Page Submission',
       'Lead Source_Direct Traffic', 'Lead Source_Google',
       'Lead Source_Olark Chat', 'Lead Source_Organic Search', 'Country_India',
       'Specialization_Finance Management',
       'Specialization_Marketing Management', 'Curr_Occupation_Student',
       'Curr_Occupation_Working Professional',
       'Course_Choice_Better Career Prospects'],
      dtype='object')
```

CHECKING THE VIF VALUES

Now the model variables all look good enough

All the variable VIF value is less than 5

Model is also giving accuracy % within the expected range

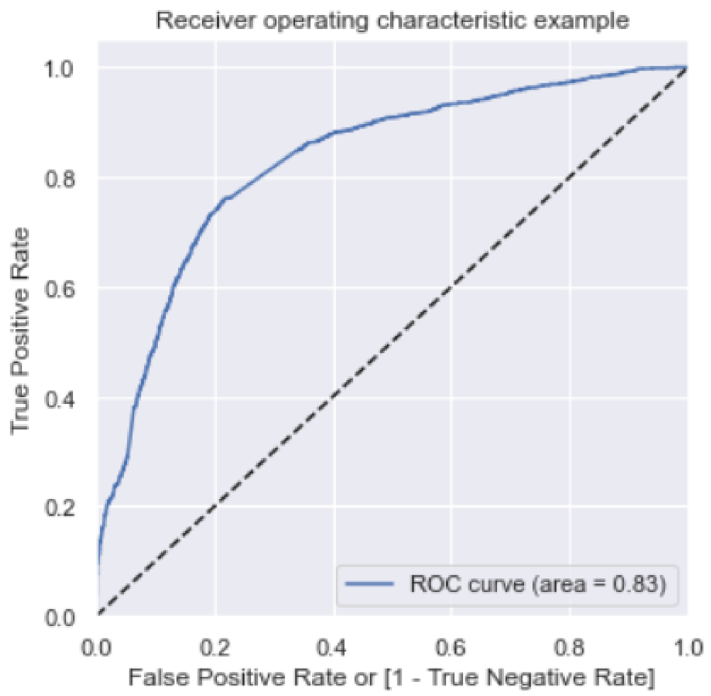
In [133]:
```
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[133]:

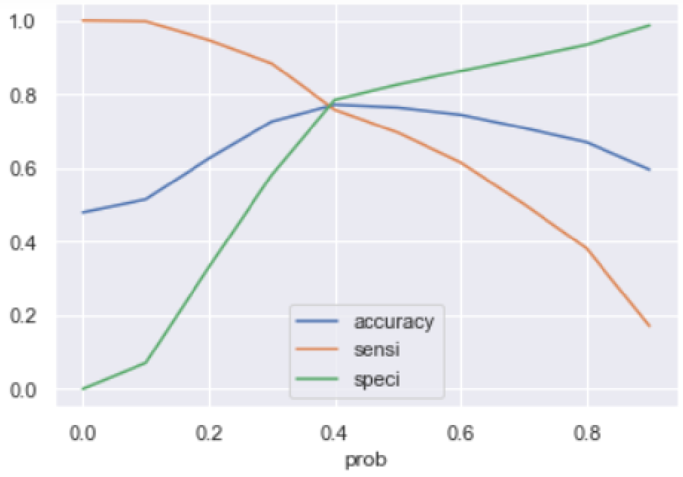
	Features	VIF
4	Lead Origin_Landing Page Submission	4.90
5	Lead Source_Direct Traffic	3.26
3	Page Views Per Visit	2.14
6	Lead Source_Google	2.08
1	TotalVisits	1.79
8	Lead Source_Organic Search	1.50
7	Lead Source_Olark Chat	1.31
2	Total Time Spent on Website	1.22
9	Specialization_Finance Management	1.19
10	Specialization_Marketing Management	1.15
12	Curr_Occupation_Working Professional	1.11
0	Do Not Email	1.10
11	Curr_Occupation_Student	1.04

PLOTTING THE ROC CURVE

```
In [144]: draw_roc(y_train_pred_final.Converted, y_train_pred_final.Converted_Prob)
```



FINDING THE OPTIMAL CUTOFF



From the curve above, 0.4 is the optimum point to take it as a cutoff probability.

In [165]:
```
y_train_pred_final['final_predicted'] = y_train_pred_final.Converted_Prob.map( lambda x: 1 if x > 0.4 else 0)

y_train_pred_final.head()
```

Out[165]:

	Converted	Converted_Prob	Lead Number	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
0	0	0.158933	6630	0	1	1	0	0	0	0	0	0	0	0	0
1	0	0.298823	8176	0	1	1	1	0	0	0	0	0	0	0	0
2	0	0.617596	4978	1	1	1	1	1	1	1	1	0	0	0	1
3	0	0.205180	7457	0	1	1	1	0	0	0	0	0	0	0	0
4	0	0.324083	4383	0	1	1	1	1	0	0	0	0	0	0	0

CHECKING OVERALL ACCURACY

Creating new column 'predicted' with 1 if Churn_Prob > 0.5 else 0

```
y_train_pred_final["predicted"] = y_train_pred_final.Converted_Prob.map(lambda x: 1 if x > 0.5 else 0)

# Let's see the head
y_train_pred_final.head(10)
```

	Converted	Converted_Prob	Lead Number	predicted
0	0	0.162972	6630	0
1	0	0.302033	8176	0
2	0	0.619870	4978	1
3	0	0.207763	7457	0
4	0	0.326288	4383	0
5	0	0.384336	4152	0
6	1	0.769680	406	1
7	0	0.256225	8992	0
8	0	0.203254	1504	0
9	1	0.572031	66	1

```
In [150]: y_train_pred_final['final_predicted'] = y_train_pred_final.Converted_Prob.map( lambda x: 1 if x > 0.4 else 0)

          y_train_pred_final.head()
```

Out[150]:

	Converted	Converted_Prob	Lead Number	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
0	0	0.158933	6630	0	1	1	0	0	0	0	0	0	0	0	0
1	0	0.298823	8176	0	1	1	1	0	0	0	0	0	0	0	0
2	0	0.617596	4978	1	1	1	1	1	1	1	1	0	0	0	1
3	0	0.205180	7457	0	1	1	1	0	0	0	0	0	0	0	0
4	0	0.324083	4383	0	1	1	1	1	0	0	0	0	0	0	0

```
In [151]: # Let's check the overall accuracy.
          metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
```

Out[151]: 0.7719181130396083

Creating new column 'predicted' with 1 if Churn_Prob > 0.5 else 0

Accuracy = 77.1%

METRICS BEYOND ACCURACY

Metrices	Value
Sensitivity	75.8
Specificity	78.4
False Positive Rate	21.5
Positive Predictive Value	76.4
Negative Predictive value	77.9

```
In [153]: 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 [154]: # Let's see the sensitivity of our logistic regression model
          TP / float(TP+FN)

Out[154]: 0.7580120761727822

In [155]: # Let us calculate specificity
          TN / float(TN+FP)

Out[155]: 0.7847073900042717

In [156]: # Calculate false postive rate - predicting churn when customer does not have churned
          print(FP/ float(TN+FP))

          0.21529260999572833

In [157]: # Positive predictive value
          print (TP / float(TP+FP))

          0.7640449438202247

In [158]: # Negative predictive value
          print (TN / float(TN+ FN))

          0.779050042408821
```

MAKING PREDICTIONS ON THE TEST SET

Metrices	Value
Overall Accuracy	76.1
Sensitivity	77.5
Specificity	74.9

```
In [219]: TP = confusion2[1,1] # true positive
          TN = confusion2[0,0] # true negatives
          FP = confusion2[0,1] # false positives
          FN = confusion2[1,0] # false negatives
```