Presentation on Lead score case study

Nikhil Bhati

Tanudeep BM

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%. There are a lot of leads generated in the initial stage (top) but only a few of them come out as paying customers from the bottom. In the middle stage, you need to nurture the potential leads well (i.e. educating the leads about the product, constantly communicating etc.) in order to get a higher lead conversion. 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 be followed for model building

1. Data cleaning – remove unwanted data
2. Univariate/Bivariate/multivariate analysis- check the correlation and outliers of clean data
3. Create dummies for EDA processed data
4. Split the data into train and test dataset
5. Fit dataset and perform rfe method and plot ROC curve to find cut off point. Calculate the precision and recall
6. Perform same steps on test dataset

EDA analysis on the data set

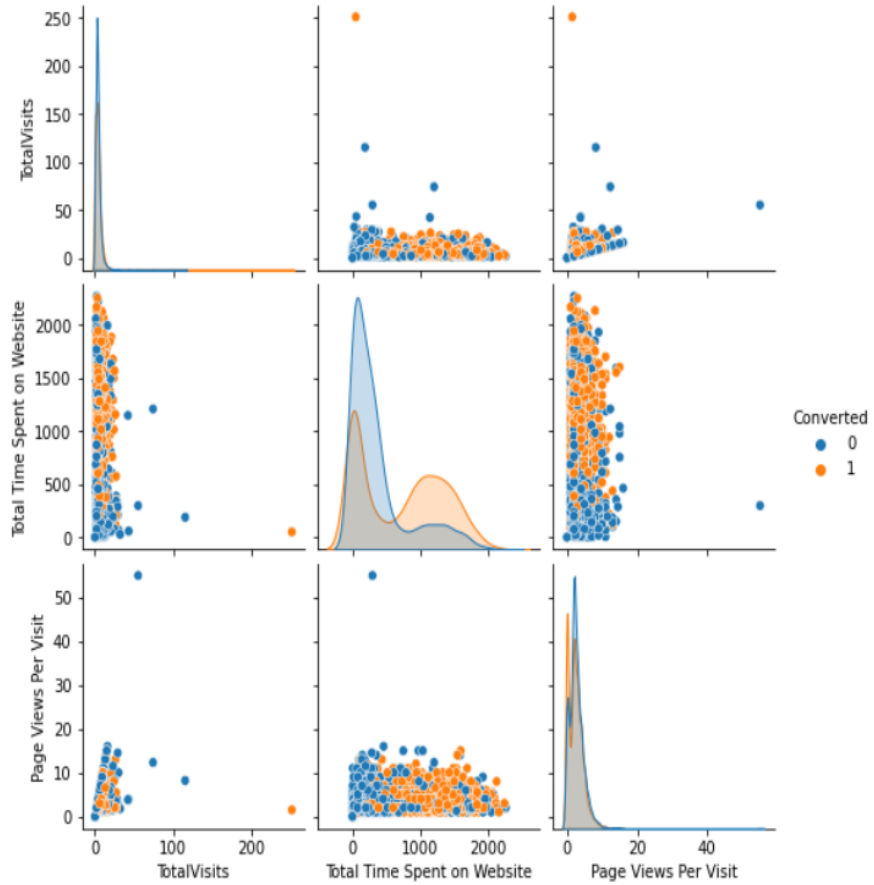
```
In [20]: # Lets check if we have any Null values left again
         leads.isnull().sum()

Out[20]: Prospect ID                                 0
         Lead Number                                 0
         Lead Origin                                 0
         Lead Source                                 0
         Do Not Email                                0
         Converted                                   0
         TotalVisits                                 0
         Total Time Spent on Website                 0
         Page Views Per Visit                        0
         Last Activity                               0
         Specialization                              0
         What is your current occupation             0
         A free copy of Mastering The Interview      0
         Last Notable Activity                       0
         dtype: int64
```

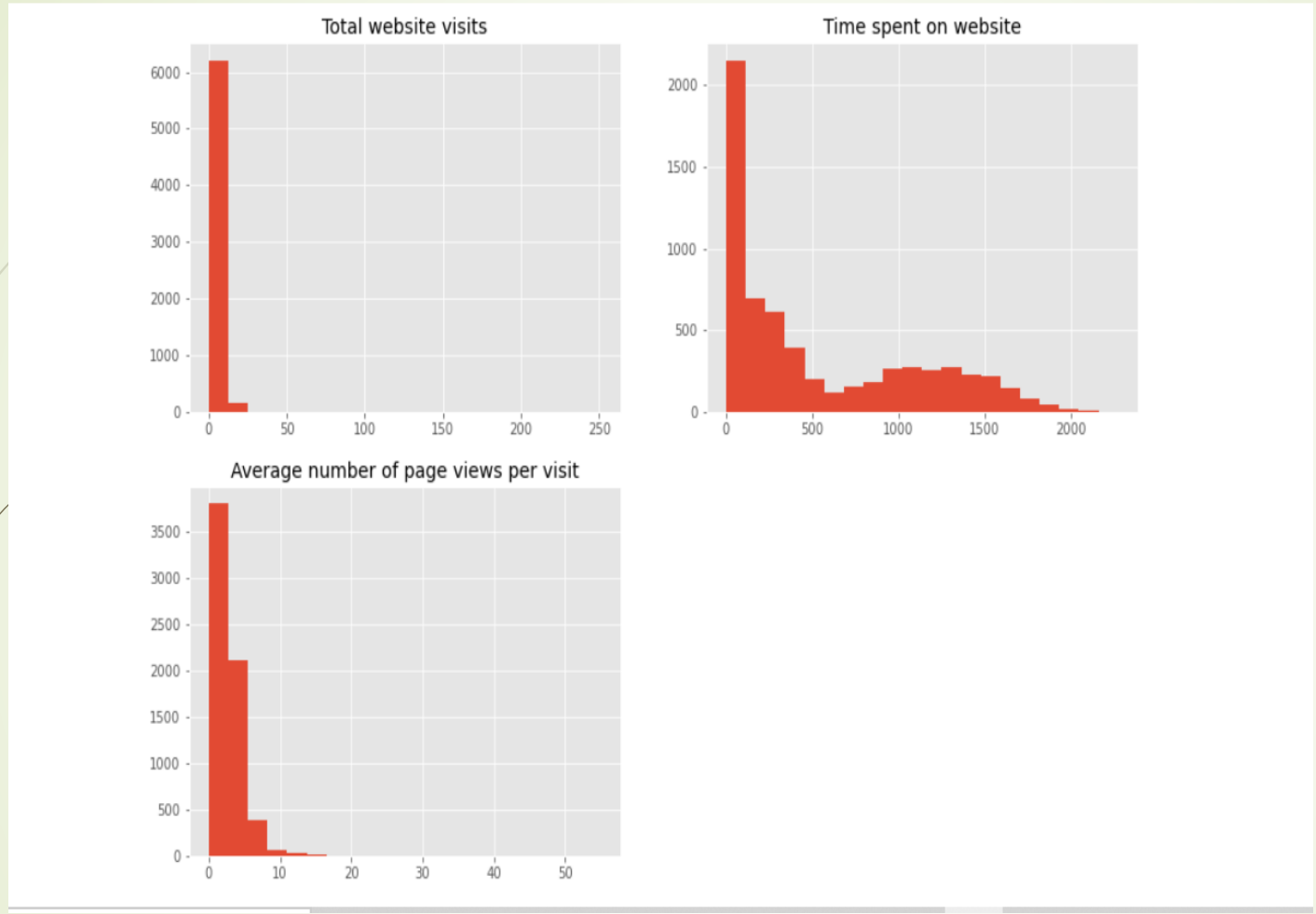
All the Null values are dropped

The initial step of EDA is data cleaning . we process the given dataset by removing those rows/columns that have null values

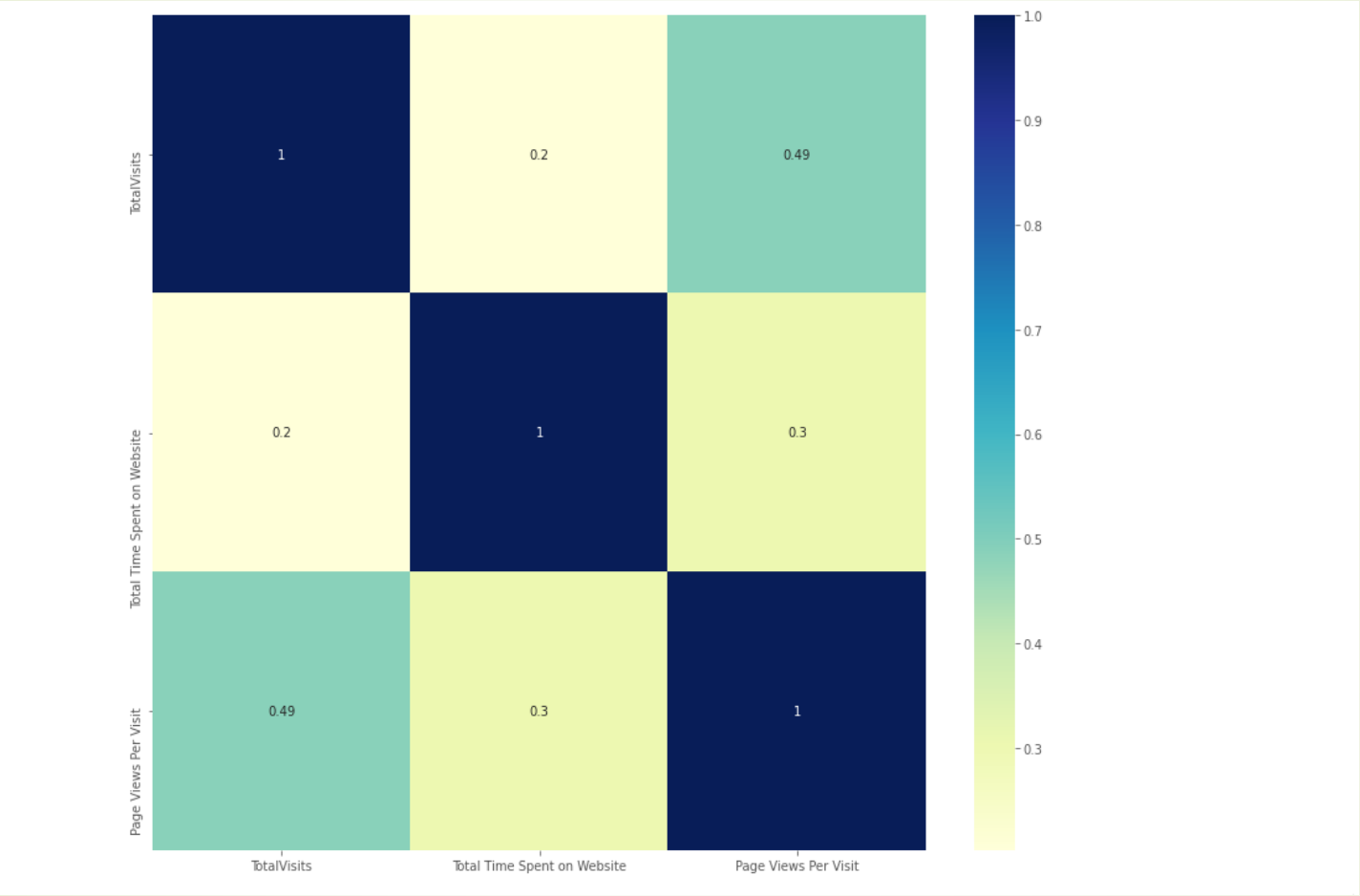
```
In [24]: sns.pairplot(leads,diag_kind='kde',hue='Converted')
         plt.show()
```



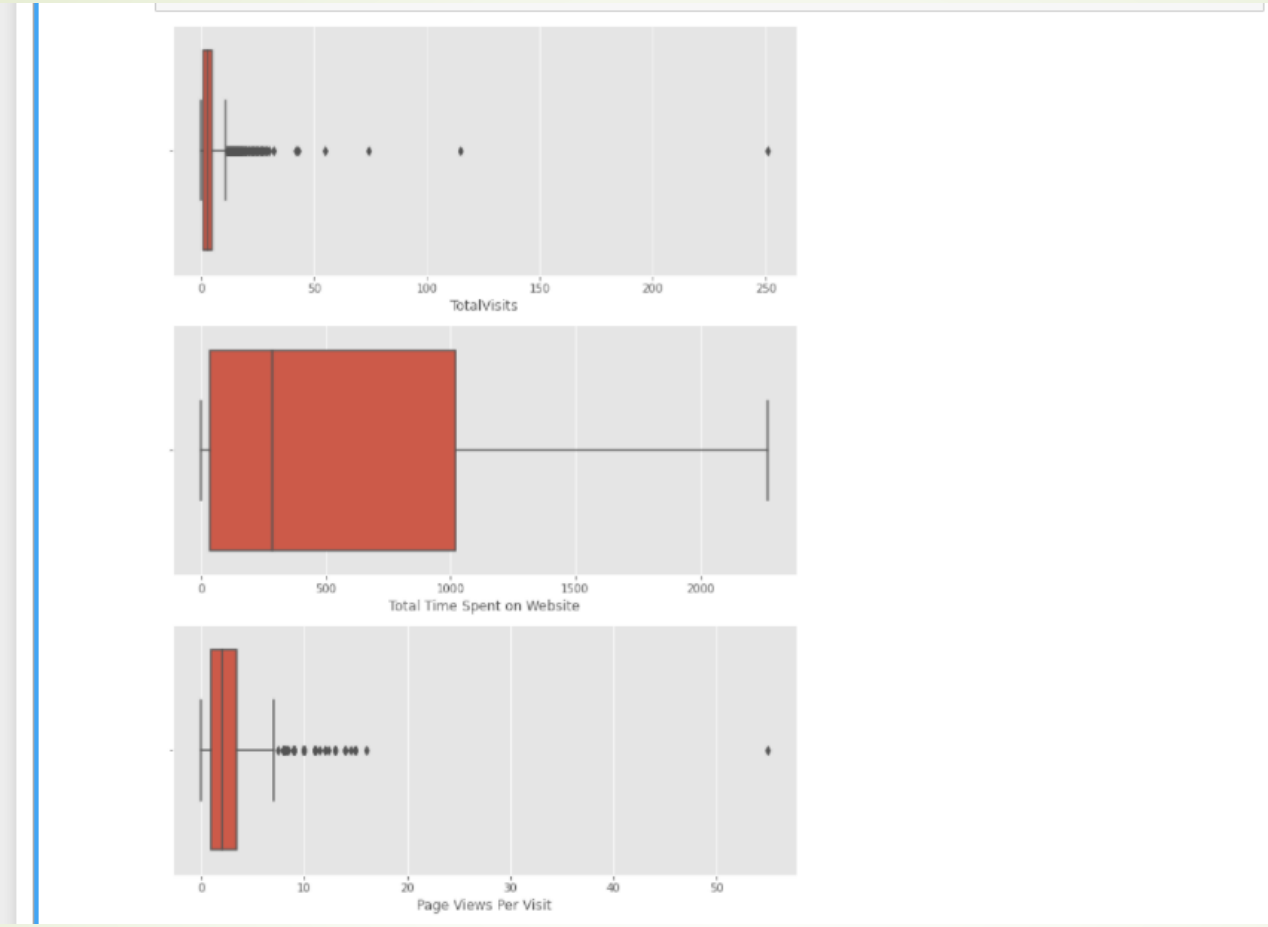
After processing the data cleaning our data set will undergo the various analysis to find the correlation in dataset In above figure we have bi variate analysis using pairplot and we can infer fro above analysis that leads who visited the website on high no and spend maximum time are mostly converted leads



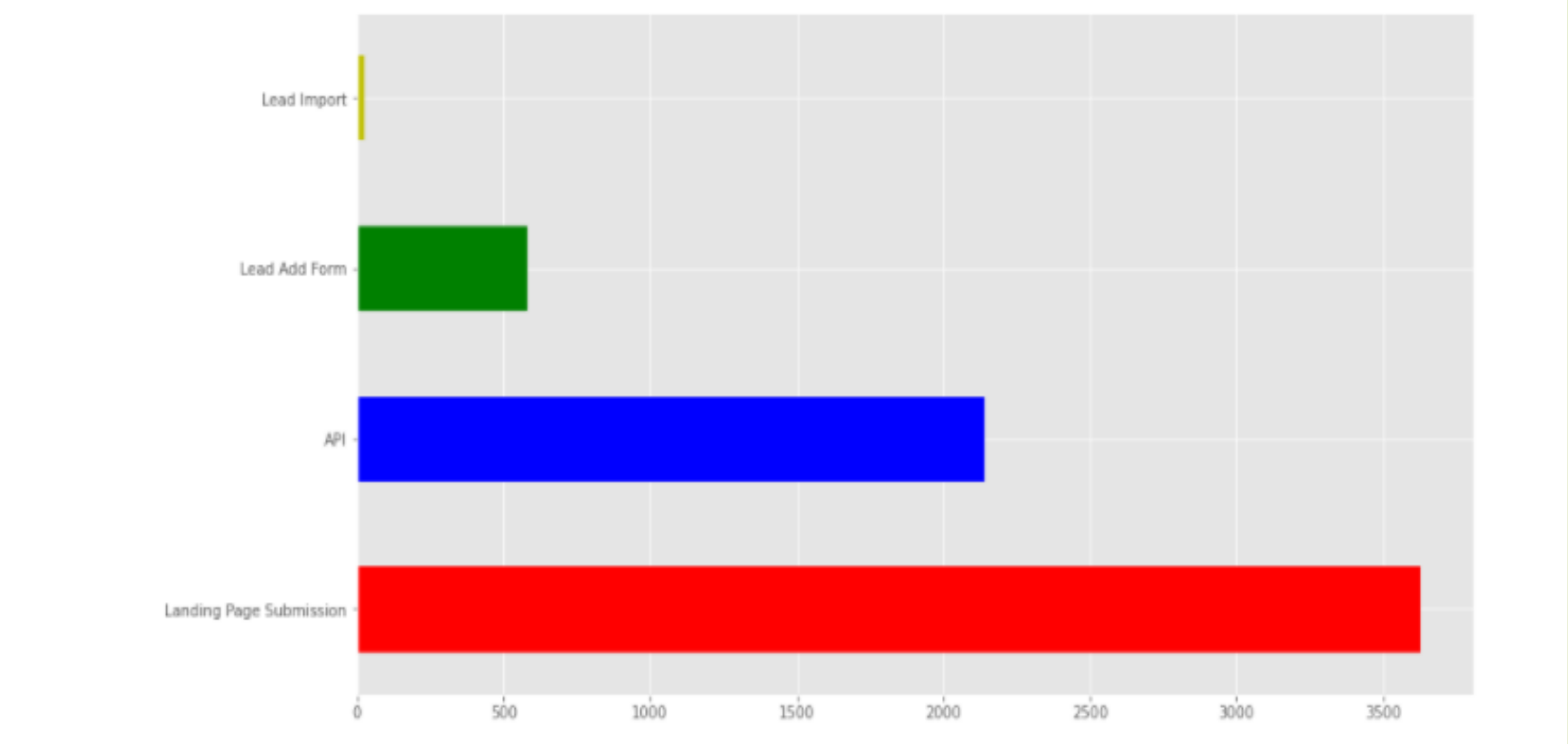
High peaks and skewed data. There might be a possibility of outliers



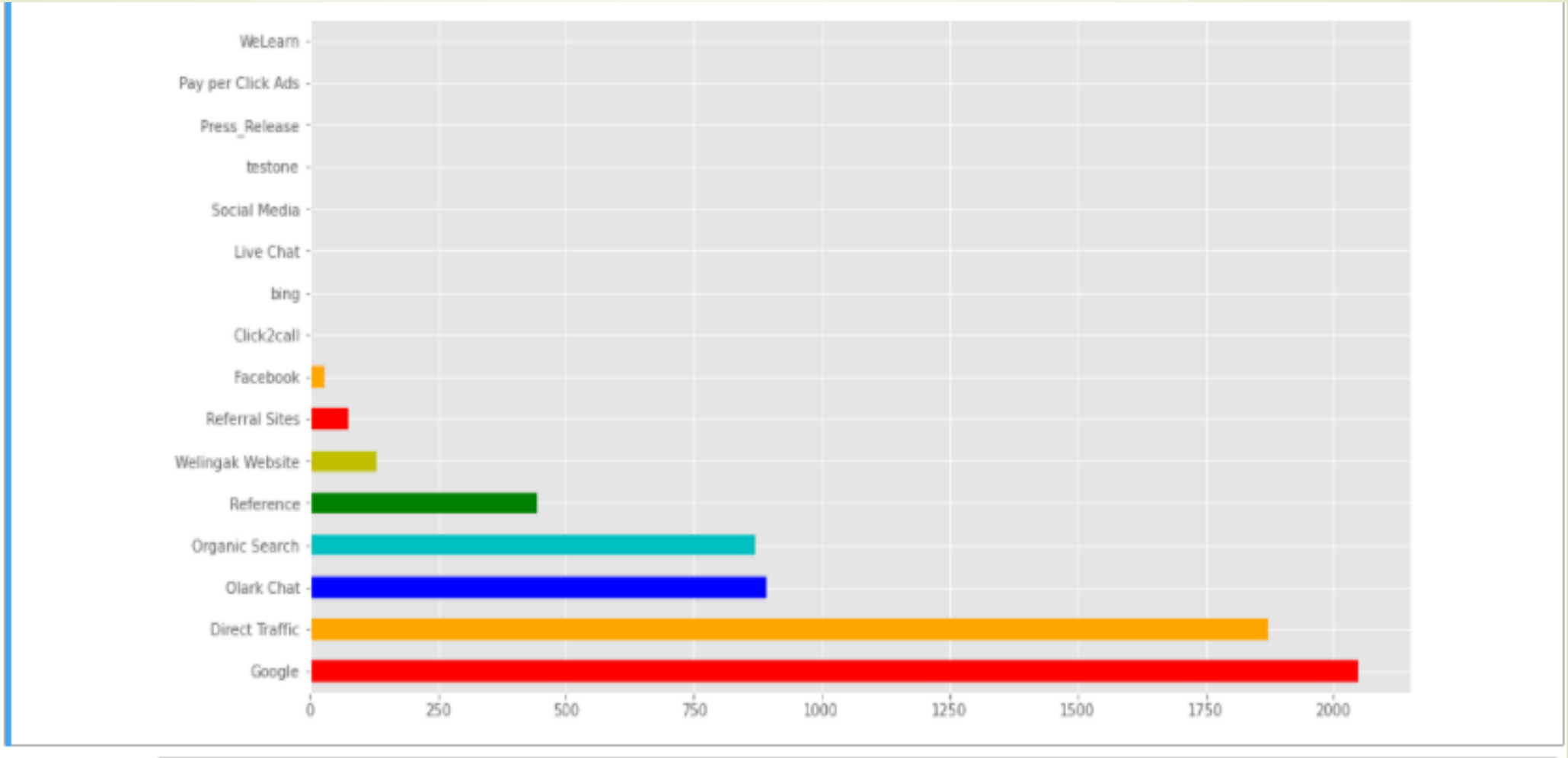
No significant correlation such that columns can be dropped



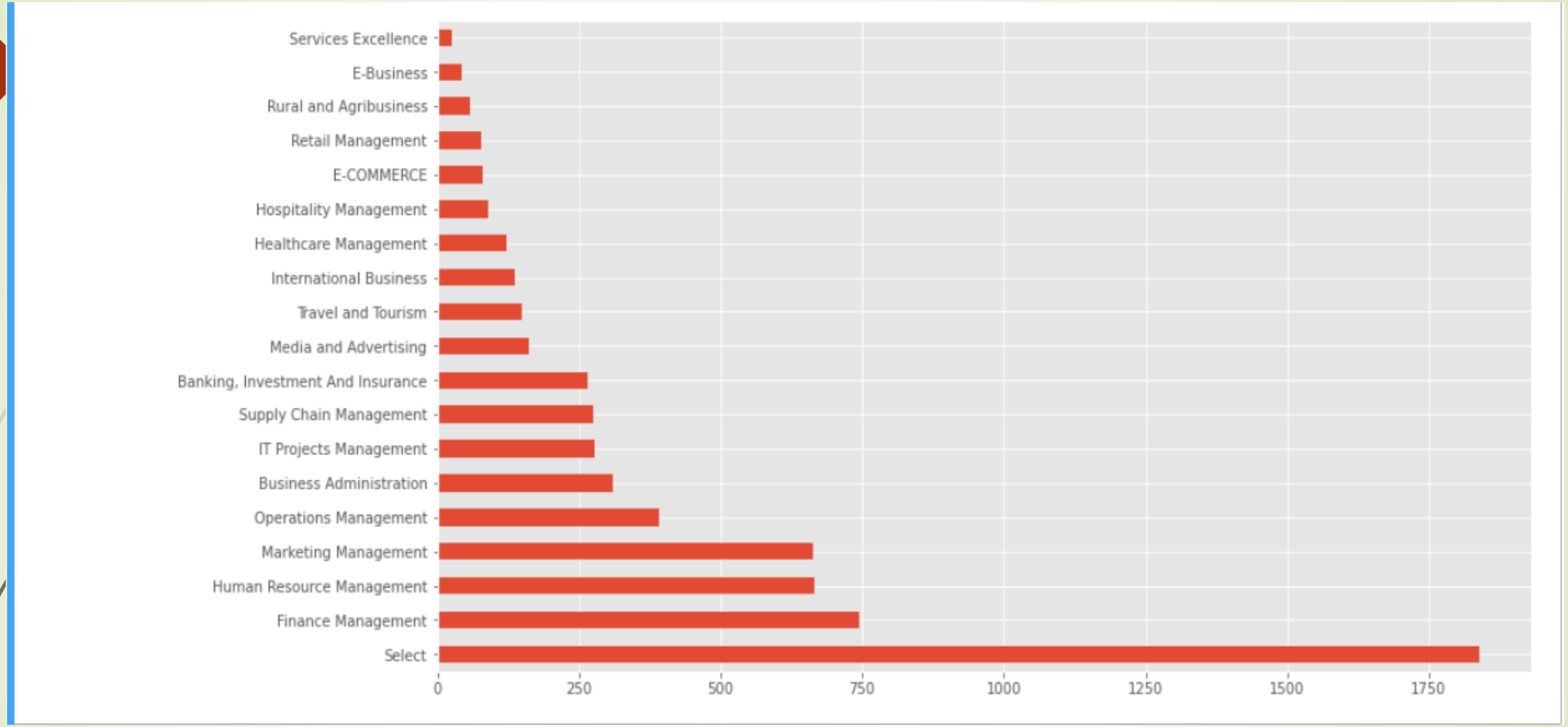
Looking at both the box plots and the statistics, there are upper bound outliers in both TotalVisits and Page Views Per Visit columns. We can also see that the data can be capped at 99 percentile



In the above graph we can infer that leads who are converted has lead orgin that landing page submission



In above fig we can observe the most leads visited the eduation website through google search and very less amount approach the website through Facebook as lead source



From above bar graph we can max no of leads are still didn't chose any specialization they chose select option as specialization

In [32]:
```
# List of variables to map

varlist =  ['A free copy of Mastering The Interview','Do Not Email']

# Defining the map function
def binary_map(x):
    return x.map({'Yes': 1, "No": 0})

# Applying the function to the housing list
leads[varlist] = leads[varlist].apply(binary_map)
```

In [33]:
```
#getting dummies and dropping the first column and adding the results to the master dataframe
dummy = pd.get_dummies(leads[['Lead Origin','What is your current occupation']], drop_first=True)
leads = pd.concat([leads,dummy],1)
```

In [34]:
```
dummy = pd.get_dummies(leads['Specialization'], prefix  = 'Specialization')
leads = pd.concat([leads, dummy], axis = 1)
```

In [35]:
```
dummy = pd.get_dummies(leads['Lead Source'], prefix  = 'Lead Source')
leads = pd.concat([leads, dummy], axis = 1)
```

In [36]:
```
dummy = pd.get_dummies(leads['Last Activity'], prefix  = 'Last Activity')
leads = pd.concat([leads, dummy], axis = 1)
```

In [37]:
```
dummy = pd.get_dummies(leads['Last Notable Activity'], prefix  = 'Last Notable Activity')
leads = pd.concat([leads, dummy], axis = 1)
```

In [38]:
```
#dropping the original columns after dummy variable creation
leads.drop(cat_cols,1,inplace = True)
```

In [39]:
```
leads.head()
```

Out[39]:

Converted	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 occupation_

After procession EDA we can proced our data for logistic regision model building . We took categorical variable and created dummies using one_hot encoding method

In [50]:
```
#BUILDING MODEL #1

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[50]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	4461
Model:	GLM	**Df Residuals:**	4445
Model Family:	Binomial	**Df Model:**	15
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2069.9
Date:	Mon, 06 Dec 2021	**Deviance:**	4139.8
Time:	22:31:00	**Pearson chi2:**	4.99e+03
No. Iterations:	22		
Covariance Type:	nonrobust		

There are quite a few variables with p-value higher than 0.05. We will be dropping them. Since 'All' the p-values are less we can check the Variance Inflation Factor to see if there is any correlation between the variables

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.0549	0.605	1.745	0.081	-0.130	2.240
Total Time Spent on Website	1.0921	0.046	23.778	0.000	1.002	1.182
Lead Origin_Landing Page Submission	-0.9905	0.136	-7.265	0.000	-1.258	-0.723
Lead Origin_Lead Add Form	3.4528	0.266	12.976	0.000	2.931	3.974
What is your current occupation_Housewife	22.6721	2.39e+04	0.001	0.999	-4.68e+04	4.68e+04
What is your current occupation_Student	-1.0664	0.635	-1.678	0.093	-2.312	0.179
What is your current occupation_Unemployed	-1.2500	0.599	-2.087	0.037	-2.424	-0.076
What is your current occupation_Working Professional	1.2238	0.628	1.949	0.051	-0.007	2.455
Specialization_Select	-1.0185	0.134	-7.588	0.000	-1.282	-0.755
Lead Source_Olark Chat	1.1515	0.135	8.500	0.000	0.886	1.417
Lead Source_Welingak Website	2.6403	1.039	2.542	0.011	0.604	4.676
Last Activity_Email Bounced	-1.6914	0.371	-4.557	0.000	-2.419	-0.964
Last Activity_Had a Phone Conversation	1.1933	0.980	1.218	0.223	-0.727	3.114
Last Activity_SMS Sent	1.1268	0.082	13.707	0.000	0.966	1.288
Last Notable Activity_Had a Phone Conversation	23.1228	2.07e+04	0.001	0.999	-4.05e+04	4.06e+04
Last Notable Activity_Unreachable	2.7018	0.809	3.342	0.001	1.117	4.287

So the Values all seem to be in order so now, Moving on to derive the Probabilities, Lead Score, Predictions on Train Data

```
]: # Create a dataframe that will contain the names of all the feature variables and their respective VIFs
   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
```

]:

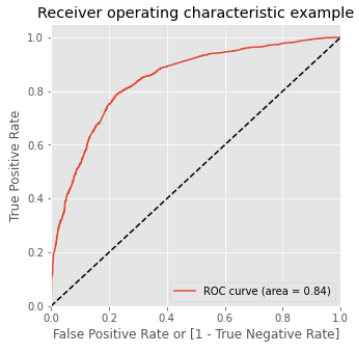
	Features	VIF
3	Specialization_Select	1.61
4	Lead Source_Olark Chat	1.54
7	Last Activity_SMS Sent	1.54
2	Lead Origin_Lead Add Form	1.51
1	Lead Origin_Landing Page Submission	1.35
5	Lead Source_Welingak Website	1.34
0	Total Time Spent on Website	1.24
6	Last Activity_Email Bounced	1.05
8	Last Notable Activity_Unreachable	1.00

In [69]:
```
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')
    plt.legend(loc="lower right")
    plt.show()

    return None
```

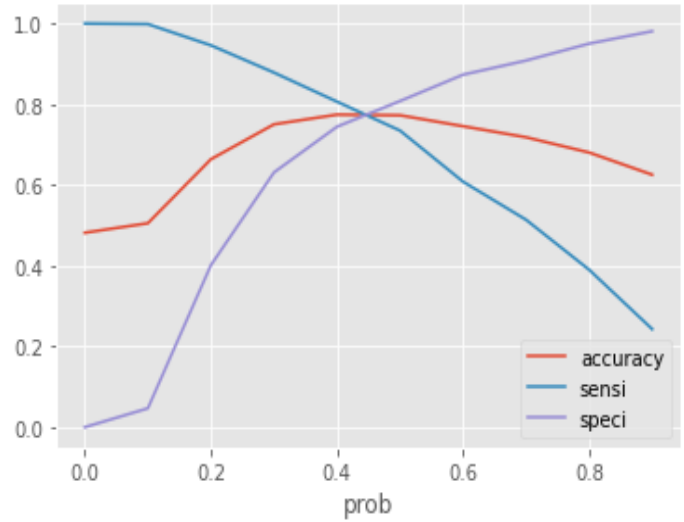
In [70]:
```
fpr, tpr, thresholds = metrics.roc_curve( y_train_pred_final.Converted, y_train_pred_final.Converted_prob, drop_intermediate = Fa
```

In [71]:
```
draw_roc(y_train_pred_final.Converted, y_train_pred_final.Converted_prob)
```



The ROC Curve should be a value close to 1. We are getting a good value of 0.84 indicating a good predictive model.

```
: # Let's plot accuracy sensitivity and specificity for various probabilities.
cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'])
plt.show()
```



From the curve above, 0.45 is the optimum point to take it as a cutoff probability.

- **Final Observation:¶**
- Let us compare the values obtained for Train & Test:
- Train Data:
- Accuracy : 77.85%
- Sensitivity : 77.33%
- Specificity : 78.33%
- Test Data:
- Accuracy : 76.52%
- Sensitivity : 75.44%
- Specificity : 77.51%
- The Model seems to predict the Conversion Rate pretty decent and we should be able to give the CEO confidence in making pretty much of good calls based on this model

Conclusion

- After understanding the problem and skim through the dataset followed series of step to build model
- The intial step was to import the necessary libraries we required and for building a model
- Then we performed data cleaning and remove the unwanted to data that is not needed
- Then we performed various analysis like univariate , bivariate and multivariate to extract the correlation and checking the outlier through graphical representation
- Once the EDA process is performed on dataset further we create dummies for categorical column nd using sklearn we split the processed dataset into Train and test dataset fit it using scalar
- Further perform the rfe elimination and using vif
- After above steps we plot roc curve and find threshold then calculate the accuracy, specificity for train data
- Follow similar steps to test dataset and calculate accuracy specificity