Lead Score Case Study

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%.

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%.

Solution Approach

The above problem statement can be solved or the best approach towards solution is Logistic regression. As logistic regression is a statistical **model** that in its basic form uses a **logistic** function to **model** a binary dependent variable, although many more complex extensions exist. In **regression** analysis, **logistic regression** (or **logit regression**) is estimating the parameters of a **logistic model** (a form of binary **regression**).

Below are the key steps or points to be used in the solution process.

- Data collection and preprocessing
- EDA (Exploratory Data Analysis)
- Outlier analysis

- Dummy variable creation
- Rescaling and Model building
- Model Evaluation

Analysis

Performing basic checks on the data set.

```
print("Data set has {0} rows and {1} columns".format(lead_data.shape[0],lead_data.shape[1]))
```

```
Data set has 9240 rows and 37 columns
```

```
#checking the conversion rate of the data

print("The conversion rate is {}%".format(round(len(lead_data[lead_data.Converted==0])/lead_dat
```

```
The conversion rate is 61.46%
```

```
lead_data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Lead Number	9240.0	617188.435606	23405.995698	579533.0	596484.5	615479.0	637387.25	660737.0
Converted	9240.0	0.385390	0.486714	0.0	0.0	0.0	1.00	1.0
TotalVisits	9103.0	3.445238	4.854853	0.0	1.0	3.0	5.00	251.0
Total Time Spent on Website	9240.0	487.698268	548.021466	0.0	12.0	248.0	936.00	2272.0
Page Views Per Visit	9103.0	2.362820	2.161418	0.0	1.0	2.0	3.00	55.0
Asymmetrique Activity Score	5022.0	14.306252	1.386694	7.0	14.0	14.0	15.00	18.0
Asymmetrique Profile Score	5022.0	16.344883	1.811395	11.0	15.0	16.0	18.00	20.0

Converting the data fields with ‘select’ to np.nan

Analysis

```
#Converting 'Select' values to NaN.

lead_data = lead_data.replace('Select', np.nan)
```

Checking for missing values :

```
#Checking which all variables are missing and the percentage missing
def missing_percentage(df):
    """This function takes a DataFrame(df) as input and returns two columns, total missing values and total miss
    total = df.isnull().sum().sort_values(ascending = False)[df.isnull().sum().sort_values(ascending = False) !=
    percent = round(df.isnull().sum().sort_values(ascending = False)/len(df)*100,2)[round(df.isnull().sum().sort
    return pd.concat([total, percent], axis=1, keys=['Total','Percent Missing'])

missing_percentage(lead_data)
```

	Total	Percent Missing
How did you hear about X Education	7250	78.46
Lead Profile	6855	74.19
Lead Quality	4767	51.59
Asymmetrique Profile Score	4218	45.65
Asymmetrique Activity Score	4218	45.65
Asymmetrique Profile Index	4218	45.65
Asymmetrique Activity Index	4218	45.65
City	3669	39.71
Specialization	3380	36.58
Tags	3353	36.29
What matters most to you in choosing a course	2709	29.32
What is your current occupation	2690	29.11
Country	2461	26.63
TotalVisits	137	1.48
Page Views Per Visit	137	1.48
Last Activity	103	1.11
Lead Source	36	0.39

Dropping columns with missing values more than 45%

Analysis

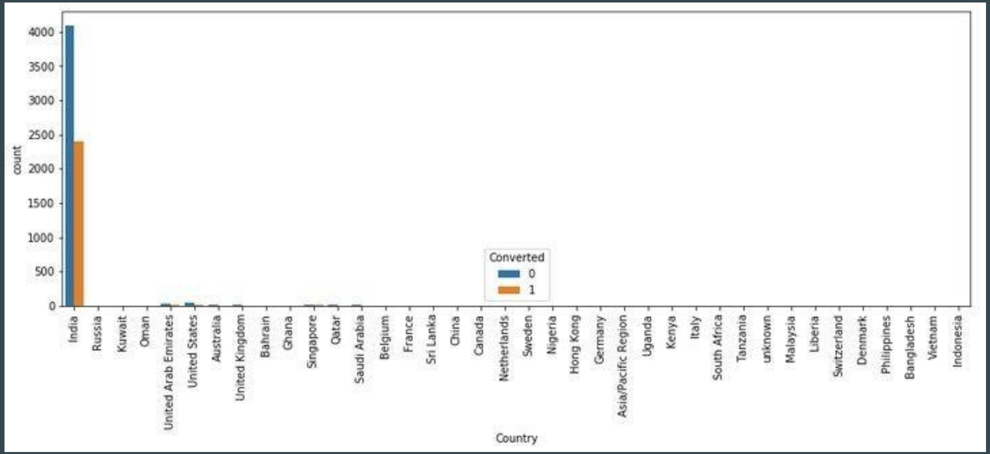
```
cols=lead_data.columns

for i in cols:
    if((100*(lead_data[i].isnull().sum()/len(lead_data.index))) >= 45):
        lead_data.drop(i, 1, inplace = True)
```

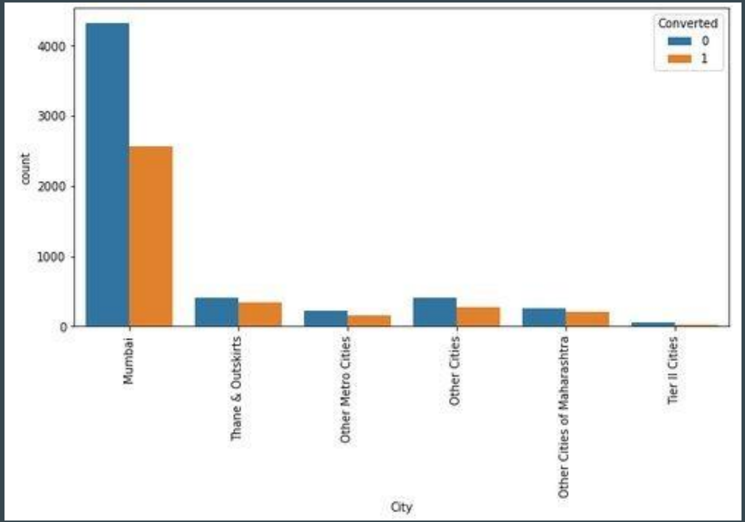
```
#Re-Checking which all variables are missing and the percentage missing

missing_percentage(lead_data)
```

	Total	Percent Missing
City	3669	39.71
Specialization	3380	36.58
Tags	3353	36.29
What matters most to you in choosing a course	2709	29.32
What is your current occupation	2690	29.11
Country	2461	26.63
TotalVisits	137	1.48
Page Views Per Visit	137	1.48
Last Activity	103	1.11
Lead Source	36	0.39

Bi-variate analysis with respect to target column ‘converted’

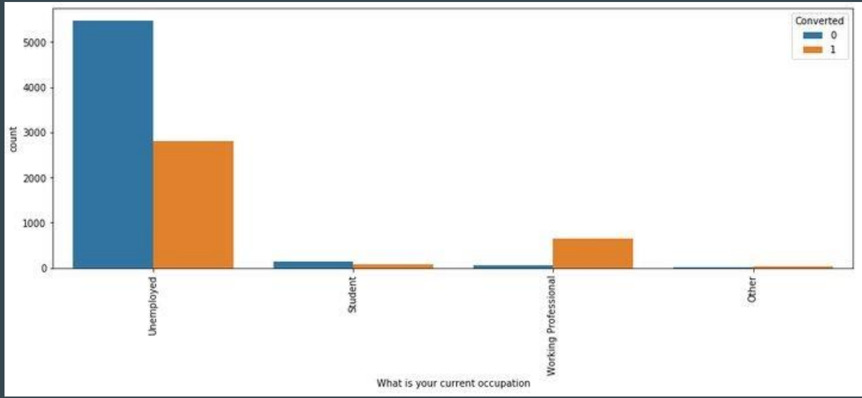
Analysis

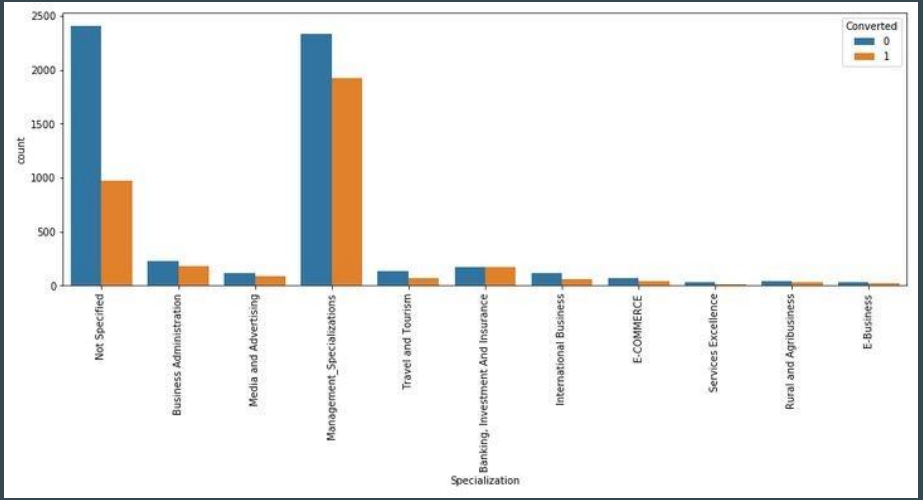




Bi-variate analysis with respect to target column ‘converted’

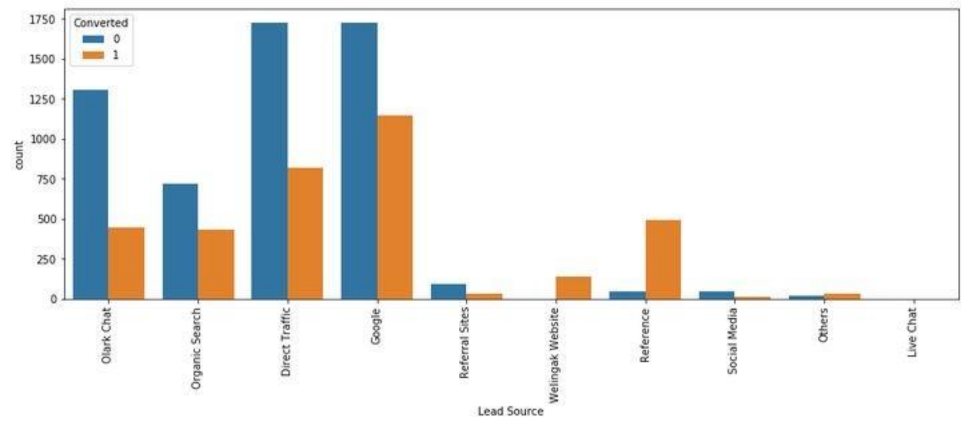
Analysis

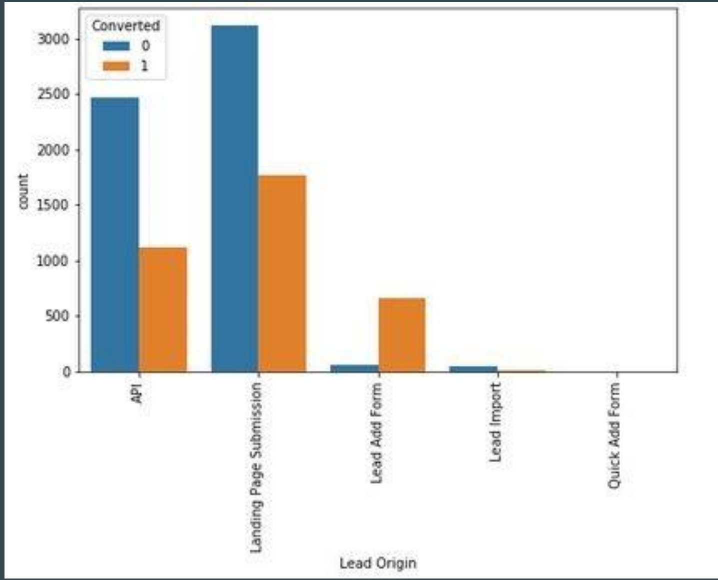




Bi-variate analysis with respect to target column ‘converted’

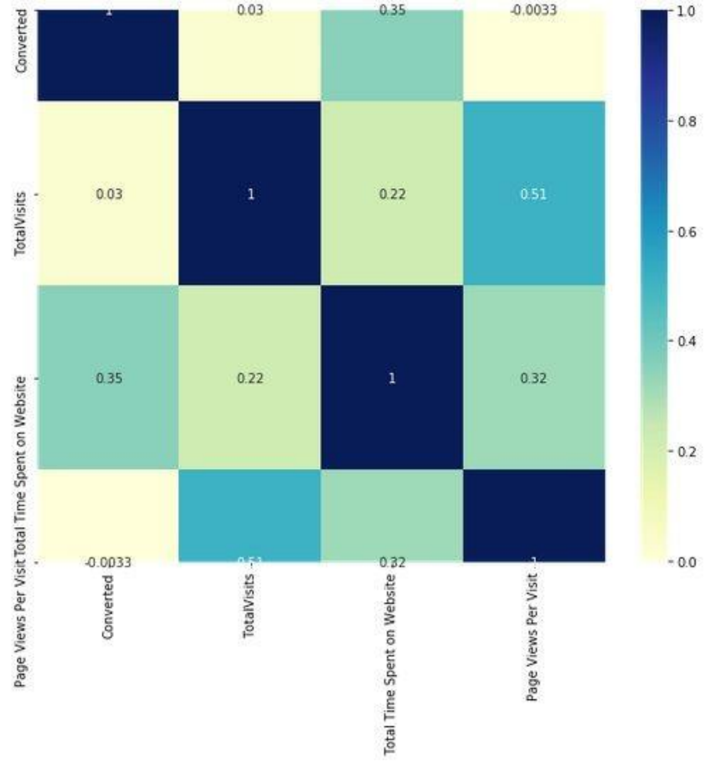
Analysis





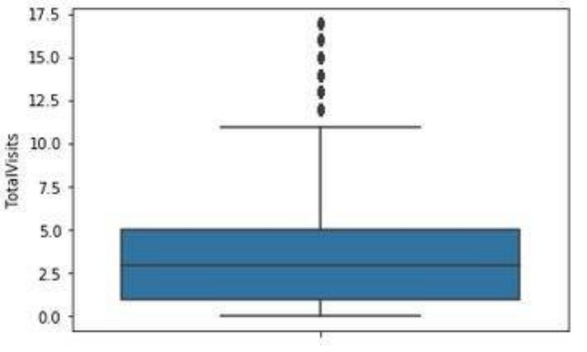
Heat map of numerical variables using correlation matrix

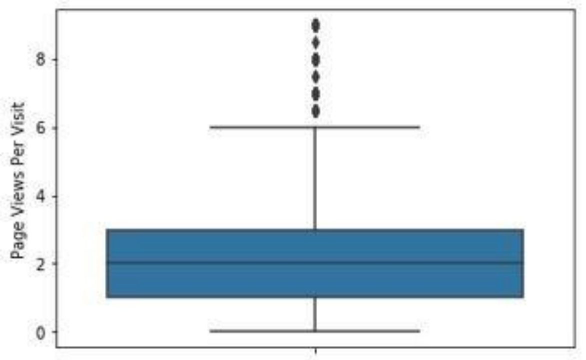
Analysis

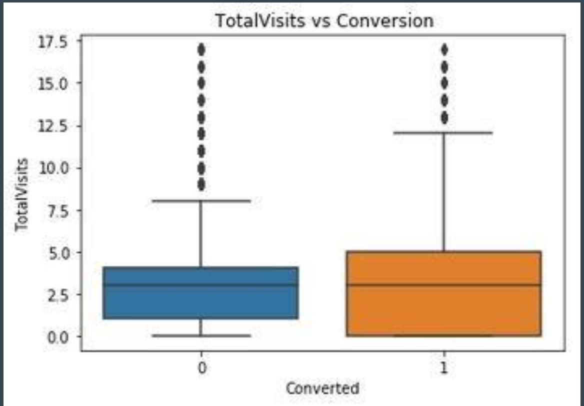


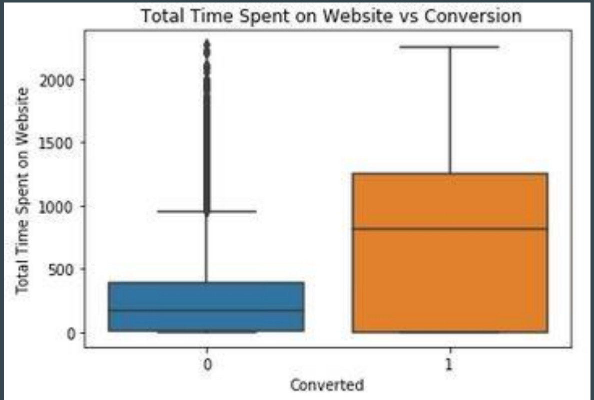
Outlier analysis

Analysis


TotalVisits
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Page Views Per Visit
0
2
4
6
8


TotalVisits vs Conversion
TotalVisits
Converted
0
1


Total Time Spent on Website vs Conversion
Total Time Spent on Website
0
500
1000
1500
2000
Converted
0
1

Dummy Variables

Creating dummy variables on lead origin, specialization, lead source, last activity, Last notable activity, Tags.

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

dummy = pd.get_dummies(lead_data['Specialization'], prefix  = 'Specialization')
dummy = dummy.drop(['Specialization_Not Specified'], 1)
lead_data = pd.concat([lead_data, dummy], axis = 1)

dummy = pd.get_dummies(lead_data['Lead Source'], prefix  = 'Lead Source')
dummy = dummy.drop(['Lead Source_Others'], 1)
lead_data = pd.concat([lead_data, dummy], axis = 1)

dummy = pd.get_dummies(lead_data['Last Activity'], prefix  = 'Last Activity')
dummy = dummy.drop(['Last Activity_Others'], 1)
lead_data = pd.concat([lead_data, dummy], axis = 1)

dummy = pd.get_dummies(lead_data['Last Notable Activity'], prefix  = 'Last Notable Activity')
dummy = dummy.drop(['Last Notable Activity_Other_Notable_activity'], 1)
lead_data = pd.concat([lead_data, dummy], axis = 1)

dummy = pd.get_dummies(lead_data['Tags'], prefix  = 'Tags')
dummy = dummy.drop(['Tags_Not Specified'], 1)
lead_data = pd.concat([lead_data, dummy], axis = 1)
```

Model Evaluation

Model Building

Final model after removing columns for no mulit-collinearity and correct VIF values

Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	6267
Model:	GLM	Df Residuals:	6253
Model Family:	Binomial	Df Model:	13
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-1263.3
Date:	Sat, 05 Dec 2020	Deviance:	2526.6
Time:	23:23:26	Pearson chi2:	8.51e+03
No. Iterations:	8		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.1179	0.084	-13.382	0.000	-1.282	-0.954
Total Time Spent on Website	0.8896	0.053	16.907	0.000	0.786	0.993
Lead Origin_Lead Add Form	1.6630	0.455	3.657	0.000	0.772	2.554
Lead Source_Direct Traffic	-0.8212	0.127	-6.471	0.000	-1.070	-0.572
Lead Source_Welingak Website	3.8845	1.114	3.488	0.000	1.701	6.068
Last Activity_SMS Sent	1.9981	0.113	17.718	0.000	1.777	2.219
Last Notable Activity_Modified	-1.6525	0.124	-13.279	0.000	-1.896	-1.409
Last Notable Activity_Olark Chat Conversation	-1.8023	0.491	-3.669	0.000	-2.765	-0.839
Tags_Closed by Horizzon	7.1955	1.020	7.053	0.000	5.196	9.195
Tags_Interested in other courses	-2.1318	0.406	-5.253	0.000	-2.927	-1.336
Tags_Lost to EINS	5.9177	0.611	9.689	0.000	4.721	7.115
Tags_Other_Tags	-2.3737	0.206	-11.507	0.000	-2.778	-1.969
Tags_Ringing	-3.4531	0.238	-14.532	0.000	-3.919	-2.987
Tags_Will revert after reading the email	4.5070	0.188	24.002	0.000	4.139	4.875

	Features	VIF
1	Lead Origin_Lead Add Form	1.82
12	Tags_Will revert after reading the email	1.56
4	Last Activity_SMS Sent	1.46
5	Last Notable Activity_Modified	1.40
2	Lead Source_Direct Traffic	1.38
3	Lead Source_Welingak Website	1.34
10	Tags_Other_Tags	1.25
0	Total Time Spent on Website	1.22
7	Tags_Closed by Horizzon	1.21
11	Tags_Ringing	1.16
8	Tags_Interested in other courses	1.12
9	Tags_Lost to EINS	1.06
6	Last Notable Activity_Olark Chat Conversation	1.01

Model is stable with no multicollinearity

```
# Getting the Predicted values on the train set
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

```
9196    0.283149
4696    0.031440
3274    0.576636
2164    0.006433
1667    0.989105
7024    0.130813
8018    0.024219
778     0.205594
6942    0.002678
4440    0.096716
dtype: float64
```

Accuracy:

Model Evaluation

```
# Let's check the overall accuracy.
print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Predicted))
```

```
0.9250039891495133
```

Sensitivity:

```
TP / float(TP+FN)
```

```
0.8821802935010482
```

Specificity:

```
TN / float(TN+FP)
```

```
0.9513137557959814
```

False-positive:

```
print(FP/ float(TN+FP))
```

```
0.04868624420401855
```

Positive predictive:

```
print (TP / float(TP+FP))
```

```
0.9175752289576974
```

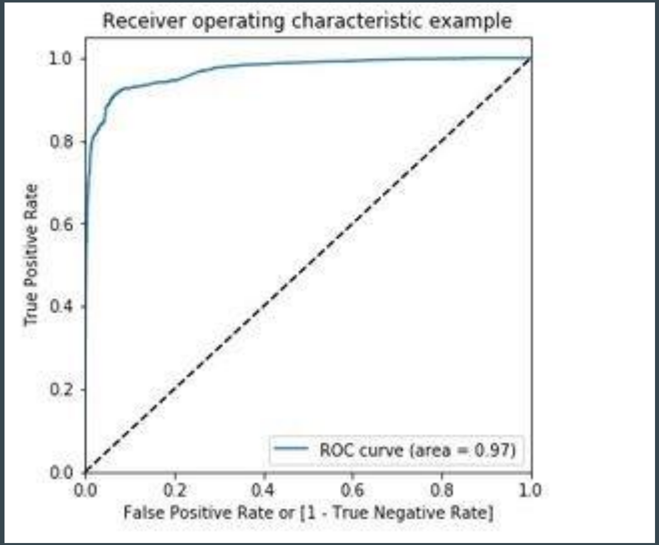
Negative predictive:

```
print (TN / float(TN+ FN))
```

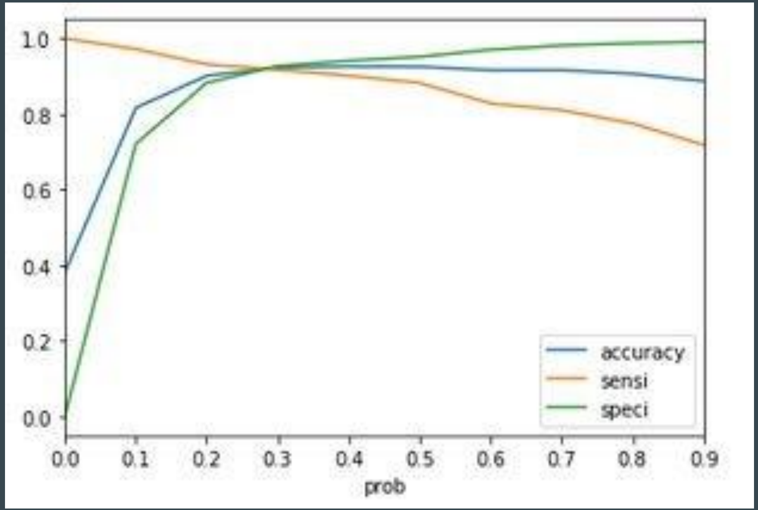
```
0.9292903875188727
```

Model Evaluation

ROC Curve (Receiver Operating Characteristic)



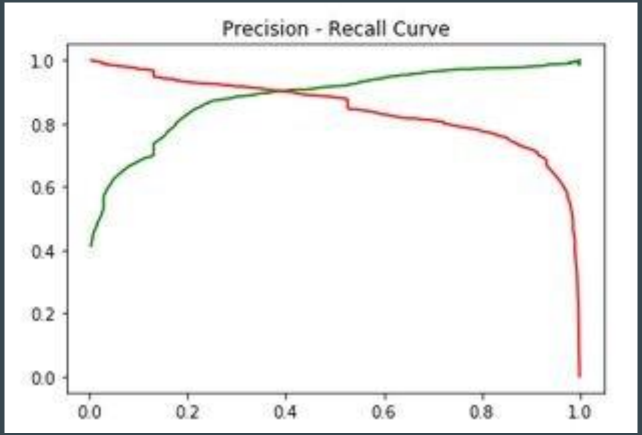
Plot for accuracy, sensitivity and specificity



Precision-Recall curve

Final Comparison of Train and Test set

Model Evaluation




```
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%
```

Inferences

- For the given Lead score problem statement we used the Logistic regression algorithm to proceed with analysis.

- In the EDA process, ‘select’ has been replaced with np.nan and then calculate the percentile of missing value in the data set. Columns with more than 45% missing values.
- Maximum number of leads are coming through Google and Direct traffic.
- Conversion Rate of leads through reference and through visiting website is high.
- Improvement of overall lead conversion rate could be achieved by focusing on olark chat, leads coming through organic search, direct traffic, and google leads .Larger focus should be done on giving proper incentives to references and improving visiting website for the coming traffic.
- API and Landing Page Submission seems to be bringing lots of lead and converting too
- Lead Add form seems to be having very good conversion rate but substantially less leads are coming with this medium.
- Lead Import and Quick Add Form get very few leads. API and Landing Page Submission should be focussed to improve the conversion .Lead Add form should be targeted to bring in more leads.
- Dropping the ‘Do Not Call’ as it is not important and not adding value to the model.

Inferences

- Majority of user are commiting activities like "Modified" or "Email Opened" .
- Users which have been receiving SMS seems more likely to getting converted which being to the forefront concept of personalization
- Dropping the rows with NaN values as rows that are being dropped are 2 % which will not impact the analysis.
- As ‘total visits’ and ‘page views per visit’ have outliers and there was a sudden increase after 90th percentile, so removing top and bottom 1% of the column outlier values.
- There seems to be a strong possibility of conversion with the time spent on the website. May be a effort be made to make the website more engaging and user friendly.
- 15 features have been selected using RFE and dropping ‘Lead Source_Referral’ as it has p-value greater than 0.05 in Model 1 and dropping ‘Last Notable Activity_SMS Sent’ as it has greater p-value than 0.05 in Model 2.

- Model 3, considered to be the stable model as it has p-values less than 0.05 and no multicollinearity has been observed with 92.29% accuracy and 92.66% specificity, when the ROC curve getting 0.97 value and 0.3 optimal cut-off. And test set having 92.78% accuracy and 93.26% specificity.