Lead Score Case Study

To Build a Logistic Regression Model to predict whether a lead for Online Courses for an education company named X Education would be successfully converted or not.

GROUP MEMBERS:

ADITYA RANJAN BEHERA

ABHIJEET BEBARTA

Business Objectives

- To help X education to select the most promising leads (Hot Leads), i.e. the leads that are most likely to convert into paying customer.
- To build a logistic regression model to assign a lead score value between 0 to 100 each of the leads which can be used by the company to target potential leads.

The objective is thus classified into sub-goals

Sub Goal-1

Create a logistic regression model to predict the lead conversion probabilities for each lead

Sub Goal-2

Decide on a probability threshold value above which a lead will be predicted as converted, whereas not converted if it is below it.

Sub Goal-3

Multiply the lead conversion probability to arrive at the lead score value for each end

Problem Solving Methodology

The approach for this project has been to divide the entire case study into various checkpoints to meet each of the sub-goals. The checkpoints are represented in a sequential flow as below.

1. Understanding the data set and data preparation
2. Applying Recursive feature elimination to identify the best performing subset of features for building the model.
3. Building the model with features selected by RFE. Eliminate all features with high p-values and VIF values and finalize the model
4. Perform model evaluation with various metrics like sensitivity, specificity, precision, recall, etc.
5. Decide on the probability threshold value based on Optimal cutoff point and predict the dependent variable for the training data.
6. Use the model for prediction on the test dataset and perform model evaluation for the test set.

Data Preparation and Processing

The following data preparation processes were applied to make the data dependable so that it can provide significant business value by improving Decision Making Process:

where a particular column has high missing values	**Columns were identified and deleted as they cannot be responsible for predicting a successful lead case.**
Removing of rows where a particular column has high missing values	**Lead source is an important column for analysis. Hence all rows having null values were dropped.**
Filling missing categorical values with random choice form that column	**Assigning a unique category to fill the null values.**

Data Preparation and Processing

Outlier Treatment

The outliers present in the columns were removed based on interquartile range analysis.

Prospect ID and Last Notable Activity were dropped in order to proceed with analysis

Prospect ID: It had unique values, hence it is dropped.

Last Notable Activity: This column was similar to last activity.

Data Preparation and Processing

Perfom binary map on columns having Boolean values.

Convert the boolean string values "Yes" and "No" to 1 and 0 by using binary map function

Create dummy variables for categorical columns by using Label encoder

The columns on which create dummy variables 'Do Not Email', 'Magazine', 'Newspaper', 'A free copy of Mastering The Interview'.

Data Preparation and Processing

Correlation matrix is a symmetric matrix so represent the upper matrix with features having higher correlations and lower correlation.

```
d Import                          Lead Source_Facebook                                   0.98
d Add Form                        Lead Source_Reference                                  0.86
rrent occupation_Unemployed       What is your current occupation_Working Professional   0.84
                                  Page Views Per Visit                                   0.73
                                  Last Activity_Email Bounced                            0.62
ding Page Submission              A free copy of Mastering The Interview_1               0.56
                                  Lead Source_Olark Chat                                 0.52
Visit                             Lead Source_Olark Chat                                 0.52
mail Opened                       Last Activity_SMS Sent                                 0.51
Visit                             Lead Origin_Landing Page Submission                    0.50
                                  Tags_Will revert after reading the email               0.48
                                  Lead Source_Olark Chat                                 0.47
                                  Country_United States                                  0.47
d Add Form                        Lead Source_Welingak Website                           0.45
                                  Country_Other_Country                                  0.45
```

```
otalVisits                                   Tags_Will revert after reading the email    0.
pecialization_Business Administration        Tags_Not doing further education            0.
hat is your current occupation_Student       City_Other Cities                           0.
hat is your current occupation_Unemployed    Country_United Kingdom                      0.
ountry_Qatar                                 City_Other Cities of Maharashtra            0.
pecialization_Supply Chain Management        City_Tier II Cities                         0.
otalVisits                                   Country_Bahrain                             0.
pecialization_E-Business                     Tags_Not doing further education            0.
ead Source_Facebook                          What is your current occupation_Student     0.
ead Source_Welingak Website                  City_Other Metro Cities                     0.
                                             Specialization_Operations Management        0.
ast Activity_SMS Sent                        Country_France                              0.
ast Activity_Unreachable                     Tags_Busy                                   0.
ead Source_Reference                         Specialization_Hospitality Management       0.
ead Origin_Lead Import                       Tags_Busy                                   0.
type: float64
```

Feature Elimination using RFE

Feature Elimation By Using Recursive Feature Elimination (RFE) Method

```
# Selecting features using RFE
log_model = LogisticRegression()

rfe = RFE(log_model, 15)
rfe = rfe.fit(X_train, y_train)
```

```
# listing the columns
sorted(list(zip(rfe.ranking_,X_train.columns, rfe.support_)))
```

```
cols = X_train.columns[rfe.support_]
temp= X_train.columns[~rfe.support_]
```

Model Building

Build the first logistic regression model by using GLM(Generalised Linear Model)

```
X_train_sm = sm.add_constant(X_train[cols])
lg_model1 = sm.GLM(y_train, X_train_sm, family = sm.families.Binomial())
res1 = lg_model1.fit()
res1.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6293
Model:	GLM	**Df Residuals:**	6277
Model Family:	Binomial	**Df Model:**	15
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2045.3
Date:	Sun, 19 Apr 2020	**Deviance:**	4090.7
Time:	21:54:20	**Pearson chi2:**	6.47e+03
No. Iterations:	21		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.9682	0.090	-33.156	0.000	-3.144	-2.793
Total Time Spent on Website	1.0689	0.046	23.289	0.000	0.979	1.159
Lead Origin_Lead Add Form	3.4339	0.275	12.495	0.000	2.895	3.973
Lead Source_Olark Chat	1.0310	0.118	8.724	0.000	0.799	1.263
Lead Source_Welingak Website	3.4100	1.059	3.219	0.001	1.334	5.486
Do Not Email_1	-1.6270	0.186	-8.754	0.000	-1.991	-1.263
Last Activity_Olark Chat Conversation	-1.5058	0.174	-8.661	0.000	-1.847	-1.165
Last Activity_Other_Activity	1.1949	0.505	2.368	0.018	0.206	2.184
Last Activity_SMS Sent	1.6515	0.091	18.151	0.000	1.473	1.830
What is your current occupation_Working Professional	0.9193	0.137	6.707	0.000	0.651	1.188
Country_Qatar	-22.1357	1.27e+04	-0.002	0.999	-2.49e+04	2.49e+04
Tags_Busy	1.7875	0.186	9.632	0.000	1.424	2.151
Tags_Closed by Horizzon	2.8630	0.175	16.353	0.000	2.520	3.206
Tags_Lost to EINS	3.5391	0.219	16.183	0.000	3.110	3.968
Tags_Will revert after reading the email	2.8170	0.094	30.056	0.000	2.633	3.001
Tags_switched off	-0.9227	0.283	-3.261	0.001	-1.477	-0.368

Model Building

Build the second logistic regression model by using GLM(Generalised Linear Model) after dropping the feature having high p- value

Dropping column with High P-Value

```
X_train_sm.drop('Country_Qatar', axis = 1, inplace  = True)
# building another model after dropping variable

lg_ml2 = sm.GLM(y_train, X_train_sm, family = sm.families.Binomial())
res2 = lg_ml2.fit()
res2.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6293
Model:	GLM	**Df Residuals:**	6278
Model Family:	Binomial	**Df Model:**	14
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2049.0
Date:	Mon, 20 Apr 2020	**Deviance:**	4098.0
Time:	06:45:31	**Pearson chi2:**	6.47e+03
No. Iterations:	8		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.9702	0.089	-33.199	0.000	-3.146	-2.795
Total Time Spent on Website	1.0672	0.046	23.304	0.000	0.977	1.157
Lead Origin_Lead Add Form	3.4356	0.275	12.504	0.000	2.897	3.974
Lead Source_Olark Chat	1.0326	0.118	8.741	0.000	0.801	1.264
Lead Source_Welingak Website	3.4089	1.059	3.218	0.001	1.333	5.485
Do Not Email_1	-1.6234	0.186	-8.740	0.000	-1.987	-1.259
Last Activity_Olark Chat Conversation	-1.5086	0.174	-8.680	0.000	-1.849	-1.168
Last Activity_Other_Activity	1.1972	0.505	2.373	0.018	0.208	2.186
Last Activity_SMS Sent	1.6486	0.091	18.143	0.000	1.471	1.827
What is your current occupation_Working Professional	0.9231	0.137	6.738	0.000	0.655	1.192
Tags_Busy	1.7907	0.185	9.654	0.000	1.427	2.154
Tags_Closed by Horizzon	2.8642	0.175	16.365	0.000	2.521	3.207
Tags_Lost to EINS	3.5406	0.219	16.195	0.000	3.112	3.969
Tags_Will revert after reading the email	2.8140	0.094	30.060	0.000	2.631	2.998
Tags_switched off	-0.9196	0.283	-3.252	0.001	-1.474	-0.365

Model Prediction

Predict the train dataset

Predict The Model

```
y_train_pred = res2.predict(X_train_sm)
```

```
y_train_pred = y_train_pred.values.reshape(-1)
```

```
# Makeing a dataframe contains y_train, and prediction
y_train_pred_final = pd.DataFrame({"converted": y_train.values, "pred_prob": y_train_pred})
y_train_pred_final.head()
```

	converted	pred_prob
0	1	0.962016
1	0	0.009591
2	1	0.689834
3	1	0.829806
4	0	0.483910

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

```
y_train_pred_final
```

Building Confusion Matrix

Build the first confusion matrix from the prediction of train dataset.

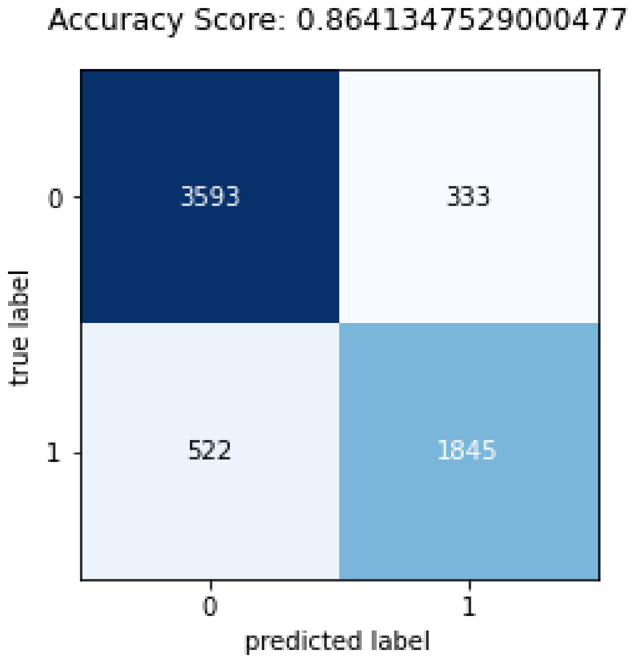
Creating Confusion Matrix

```
CM1  = confusion_matrix(y_train_pred_final.converted, y_train_pred_final.predicted)
print(CM1)
```

```
[[3593  333]
 [ 522 1845]]
```

```
accu_score = accuracy_score(y_train_pred_final.predicted, y_train_pred_final.converted)
```

```
fig, ax = plot_confusion_matrix(conf_mat=CM1)
all_sample_title = 'Accuracy Score: {0}'.format(accu_score)
plt.title(all_sample_title, size = 12);
plt.tight_layout()
plt.show()
```



ROC curve

Calculate the accuracy, sensitivity and specificity.

Plot the ROC(Receiver Operating Characteristic curve.

Calculating Accuracy, Sensitivity and Specificity

```
sensitivity = tp/(tp+fn)
specificity = tn/(tn+fp)
Accuracy = (tn+tp)/(tn+tp+fp+fn)
print("sensitivity is ", sensitivity, ", specificity is", specificity)
print("\n")
print("Accuracy  is", Accuracy)
```

```
sensitivity is  0.779467680608365 , specificity is 0.9151808456444218


Accuracy  is 0.8641347529000477
```

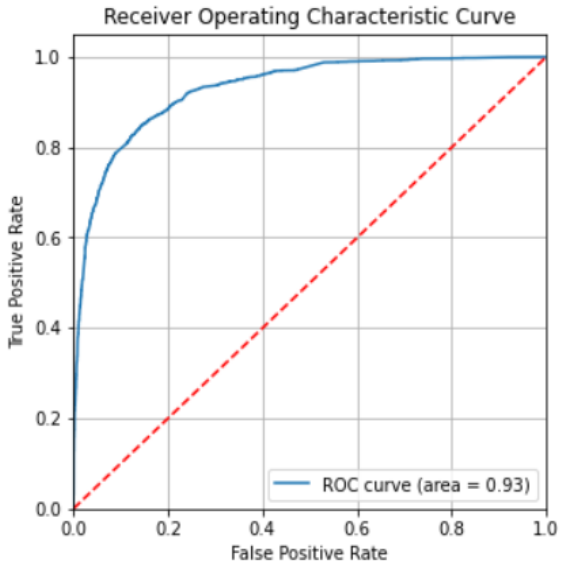
Plotting the ROC curve

```
# Function For Plotting ROC Curve
def draw_roc( actual, probs ):
    fpr, tpr, thresholds = roc_curve( actual, probs,
                                              drop_intermediate = False )
    auc_score = 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], 'r--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic Curve')
    plt.legend(loc="lower right")
    plt.grid()
    plt.show()

    return None

fpr, tpr, thresholds = roc_curve( y_train_pred_final.converted, y_train_pred_final.pred_prob, drop_intermediate = False

draw_roc( y_train_pred_final.converted, y_train_pred_final.pred_prob)
```



Finding Optimal Cut-off Point

Finding the optimal cut off point by calculating Accuracy , Sensitivity, Specificity for the probability of (0.1,0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9).

As we can the graph above, when the probability thresholds are very low, the sensitivity is very high.

Similarly for larger probability thresholds, the sensitivity values are very low but the speicificity

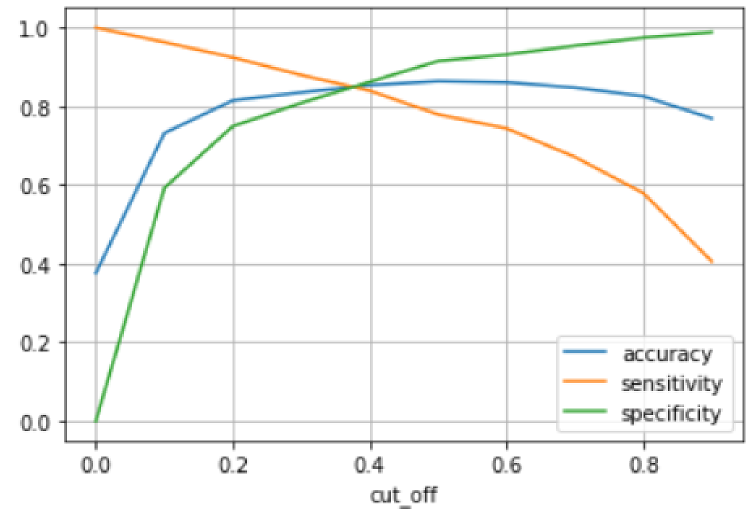
Values are very high. And at about 0.39m the three metrics seem to be almost equal with decent

values and hence we choose 0.39 as the optimal cut-off point.

Finding Optimal Cut-off Point

```
numbers = [float(x)/10 for x in range(10)]
for i in numbers:
    y_train_pred_final[i]= y_train_pred_final.pred_prob.map(lambda x: 1 if x > i else 0)
y_train_pred_final.head()
```

	cut_off	accuracy	sensitivity	specificity
0.00	0.00	0.38	1.00	0.00
0.10	0.10	0.73	0.96	0.59
0.20	0.20	0.82	0.92	0.75
0.30	0.30	0.84	0.88	0.81
0.40	0.40	0.85	0.84	0.86
0.50	0.50	0.86	0.78	0.92
0.60	0.60	0.86	0.74	0.93
0.70	0.70	0.85	0.67	0.95
0.80	0.80	0.83	0.58	0.97
0.90	0.90	0.77	0.41	0.99



Finding Optimal Cut-off Point

Find the optimal cutoff point from the graph is 0.4 and built the 2nd confusion matrix

From the above Analysis We can Consider the optimal cuttoff point is 0.4

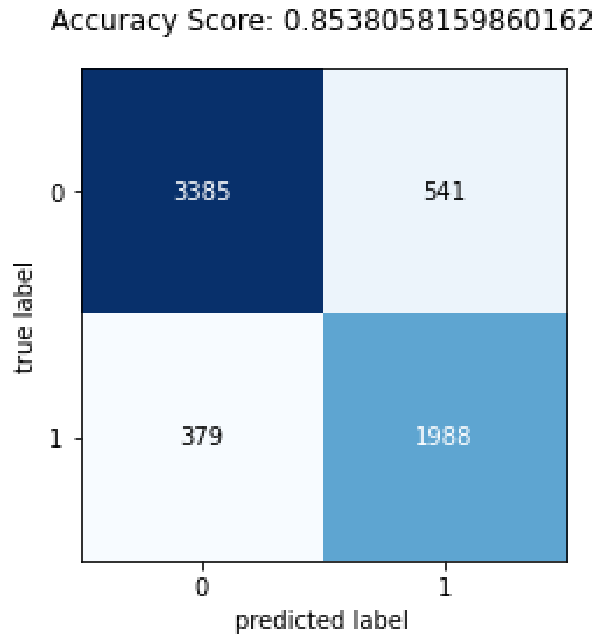
```
y_train_pred_final['predicted_values'] = y_train_pred_final.pred_prob.map(lambda x:[redacted]else 0)
y_train_pred_final.head()
```

	converted	pred_prob	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	predicted_values
0	1	0.962016	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0.009591	0	1	0	0	0	0	0	0	0	0	0	0
2	1	0.689834	1	1	1	1	1	1	1	1	0	0	0	1
3	1	0.829806	1	1	1	1	1	1	1	1	1	1	0	1
4	0	0.483910	0	1	1	1	1	1	0	0	0	0	0	1

```
accu_score = accuracy_score(y_train_pred_final.converted, y_train_pred_final.predicted_values)
```

```
CM2 = confusion_matrix(y_train_pred_final.converted, y_train_pred_final.predicted_values)
CM2
```

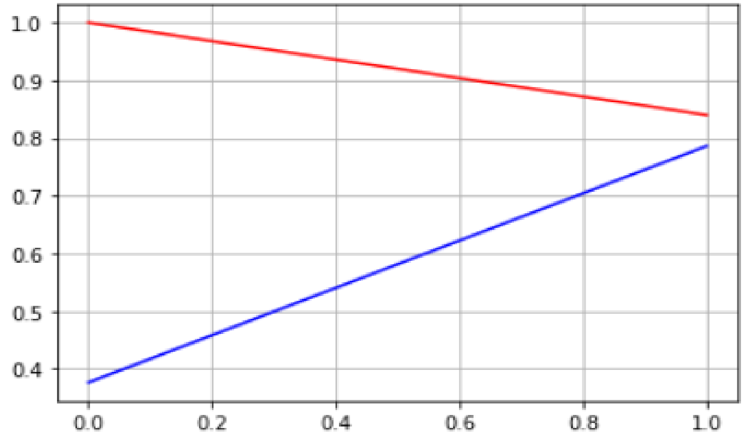
```
array([[3385,  541],
       [ 379, 1988]], dtype=int64)
```



Precision and recall tradeoff for calculating model accuracy

```
p, r, thresholds = precision_recall_curve(y_train_pred_final.converted, y_train_pred_final.predicted_values)
```

```
plt.plot(thresholds, p[:-1], "b-")
plt.plot(thresholds, r[:-1], "r-")
plt.grid()
plt.show()
```



Calculate The Model Accuracy

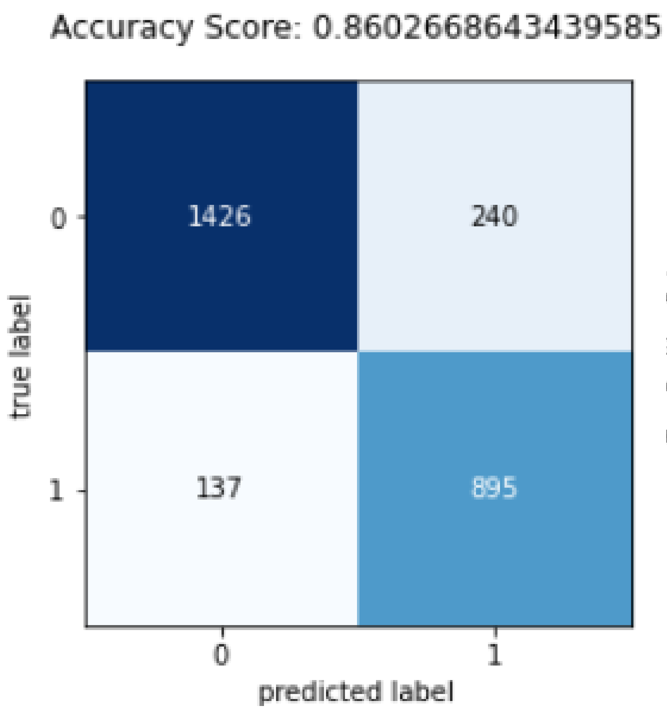
Predict The Model

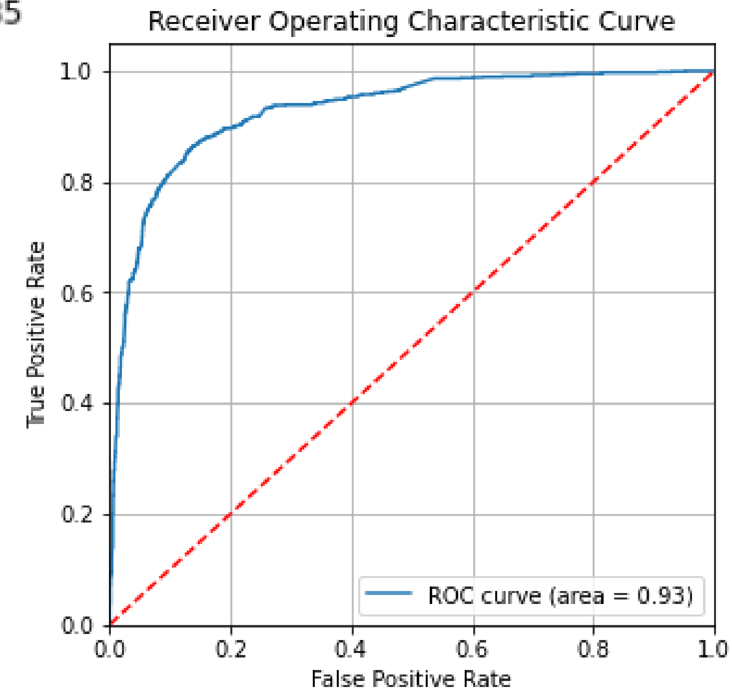
Fit the trained predicted model on test data set. Then build the confusion matrix of predicted test data set.

Plot the ROC curve

Visualizing Confusion Matrix Of Test Data Set

```
fig, ax = plot_confusion_matrix(conf_mat=cm_test)
all_sample_title = 'Accuracy Score: {0}'.format(accu_score
plt.title(all_sample_title, size = 12);
plt.tight_layout()
plt.show()
```





Classification Report

Calculate the accuracy, Precision, Recall and F1 score for the predicted model.

Classification Report

```
print(classification_report(y_pred_final.Converted, y_pred_final.final_predicted))
```

```
              precision    recall  f1-score   support

           0       0.91      0.86      0.88      1666
           1       0.79      0.87      0.83      1032

    accuracy                           0.86      2698
   macro avg       0.85      0.86      0.85      2698
weighted avg       0.86      0.86      0.86      2698
```

Visualizing Conversion Ratio

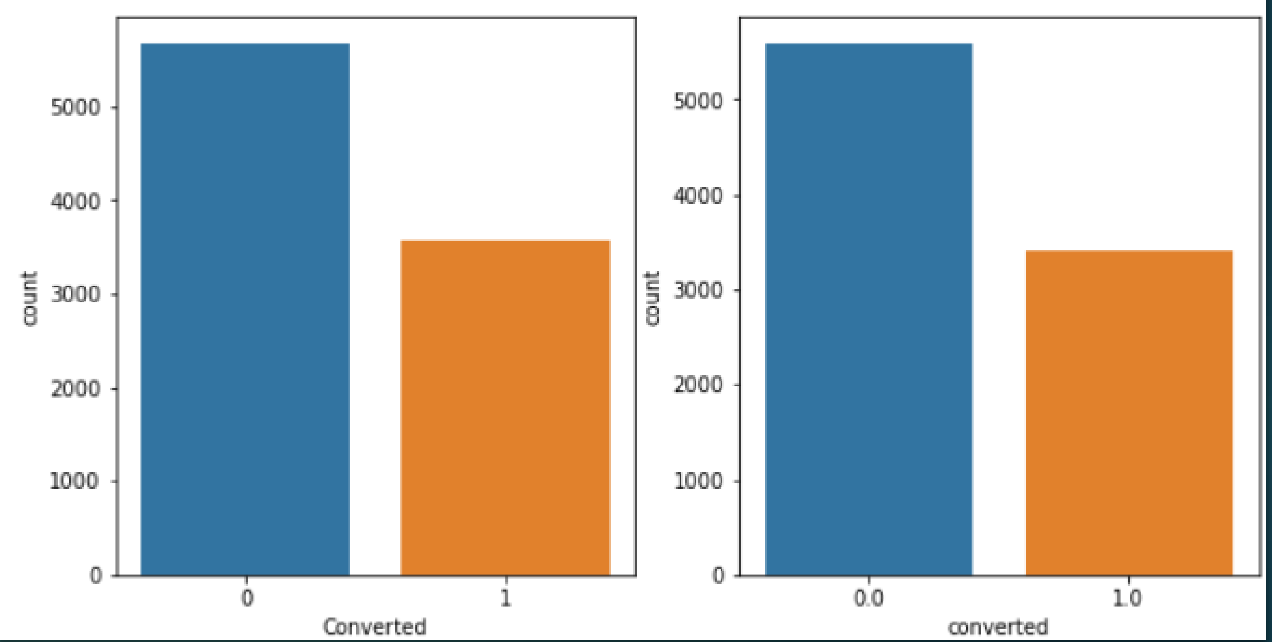
Visualize the difference between conversion ratio of actual data and predicted data.

From this visualization we can consider that the model is correctly predicted and accuracy having more than 80%.

For getting the more precise model we can increase the optimal threshold value.

Visualizing Dataset To visualizing distribution between Actual Data and Predicted Data

```
plt.figure(figsize = (10,5))
plt.subplot(1,2,1)
sns.countplot(lead_score_final["Converted"])
plt.subplot(1,2,2)
sns.countplot(lead_score_final["converted"])
plt.show()
```



Thank You