SUMMARY REPORT

Problem Approach and Analysis of the Data

X Education offers online courses to potential industry professionals. They get lots of leads from various sources such as past referrals, several websites and search engine such as Google. After the lead is identified the sales team will follow up with potential candidates and convert them as paying customers. But due to high volumes of potential leads they want to efficiently follow only the Hot Leads instead of all the leads. So our target should be to identify those hot leads which can be concentrated and converted to a paying customer thus increasing the overall revenue.

Analysis Approach:

To decide on the hot leads from the X education data provided. We will perform all the following steps.

1. Read the data from the dataset provided.
2. We need to handle the Null values and outliers from the dataset.
3. Do some initial EDA for getting the insights of the data more.
4. Create dummy columns for categorical variables.
5. Feature scaling for continuous variables.
6. Divide the data into Train and Test sets.
7. Use the RFE to decide the variables based on RFE score and eliminating the features based on p values and VIFs
8. Perform Model evaluation on Test data.
9. Perform Model Validation.

Data Preparation:

1. Import required libraries.
2. Use info, describe and shape from pandas dataset to have insights of the data
3. Check for Nulls\Select in the dataset and dropping columns with more than 30% null values.
4. Impute Nulls with Unknown.
5. Data binning of the numerical variables
6. Converting binary values for Yes/No values.
7. Create dummy variables for categorical variables.

8. Standardizing scales of continues variables like total visits , Total Time Spent on Website, Page views per visit.

Columns based on RFE scores

```
'Total Time Spent on Website', 'Lead Source_Direct Traffic',
'Lead Source_Organic Search', 'Lead Source_Reference',
'Lead Source_Unknown', 'Lead Source_Welingak Website',
'Last Activity_Email Bounced', 'Last Activity_Had a Phone Conversation',
'Last Activity_SMS Sent',
'Last Notable Activity_Had a Phone Conversation',
'Last Notable Activity_Modified',
'Last Notable Activity_Olark Chat Conversation',
'Last Notable Activity_Unreachable'],
```

RFE (Recursive Feature Elimination):

We will be Using Recursive Feature Elimination to eliminate few of the variables. We first start with 13 variables and then based on the p-value and VIF. We arrive at final list of variables. In the final list we have about 11 variables used for prediction.

The idea was to have as less variables as possible to keep the model simple yet effective

Dep. Variable:	Converted	**No. Observations:**	6299
Model:	GLM	**Df Residuals:**	6287
Model Family:	Binomial	**Df Model:**	11
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2858.4
Date:	Mon, 10 Jun 2019	**Deviance:**	5716.8
Time:	00:10:19	**Pearson chi2:**	6.41e+03
No. Iterations:	7	**Covariance Type:**	nonrobust

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.6679	0.057	-11.738	0.000	-0.779	-0.556
Total Time Spent on Website	1.0108	0.034	29.381	0.000	0.943	1.078
Lead Source_Direct Traffic	-0.5765	0.078	-7.387	0.000	-0.729	-0.424
Lead Source_Organic Search	-0.3409	0.102	-3.358	0.001	-0.540	-0.142
Lead Source_Reference	3.5941	0.201	17.857	0.000	3.200	3.989
Lead Source_Welingak Website	5.8393	1.011	5.775	0.000	3.858	7.821
Last Activity_Email Bounced	-1.0946	0.283	-3.866	0.000	-1.650	-0.540
Last Activity_Had a Phone Conversation	1.6646	0.555	2.999	0.003	0.577	2.752
Last Activity_SMS Sent	1.2830	0.071	18.085	0.000	1.144	1.422
Last Notable Activity_Modified	-0.9472	0.074	-12.860	0.000	-1.092	-0.803
Last Notable Activity_Olark Chat Conversation	-1.4055	0.319	-4.401	0.000	-2.032	-0.780
Last Notable Activity_Unreachable	1.4338	0.472	3.035	0.002	0.508	2.360

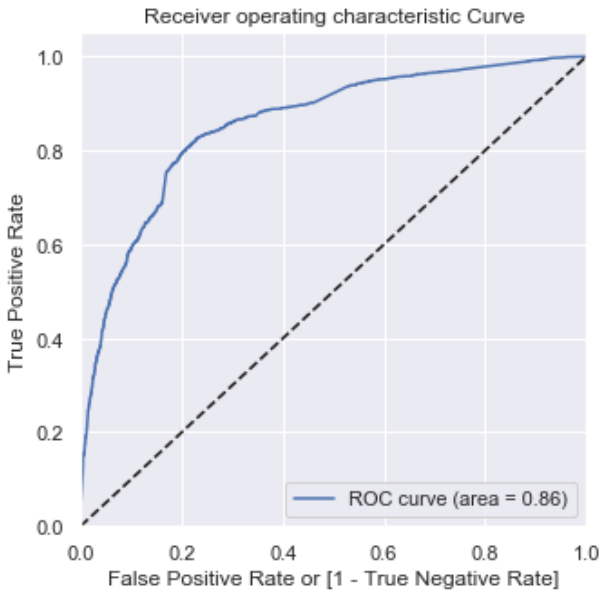
	Features	vif
8	Last Notable Activity_Modified	1.27
1	Lead Source_Direct Traffic	1.26
9	Last Notable Activity_Olark Chat Conversation	1.26
2	Lead Source_Organic Search	1.13
6	Last Activity_Had a Phone Conversation	1.09
3	Lead Source_Reference	1.08
0	Total Time Spent on Website	1.07
5	Last Activity_Email Bounced	1.03
4	Lead Source_Welingak Website	1.01
7	Last Activity_SMS Sent	1.01
10	Last Notable Activity_Unreachable	1.00

This also ensured that the selected variables have least correlation among them yet they are effective to predict the outcome well.

Model Evaluation:

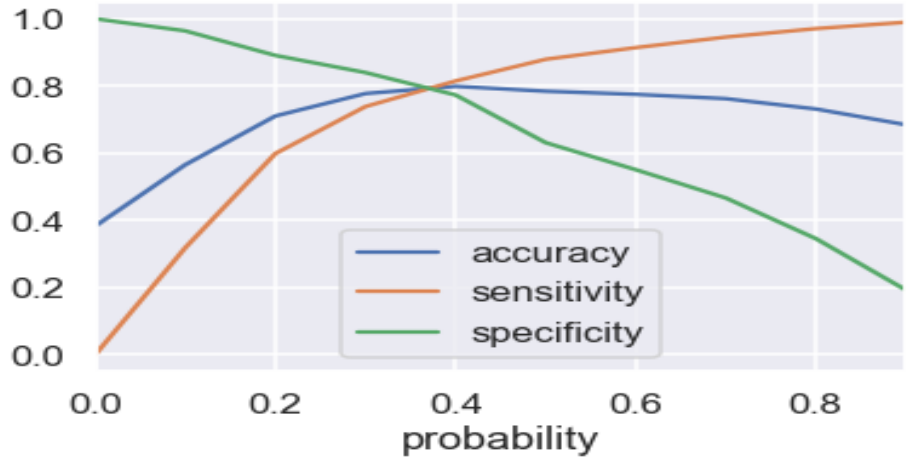
ROC Curve:

We used ROC curve to show the trade-off between True Positive Rate and False Positive Rate which essentially can also be viewed as a tradeoff between Sensitivity and Specificity. As the Sensitivity and the Specificity are inversely proportional, we have to have as much area as we can as AUC.



Accuracy, Specificity ad sensitivity optimal point:

We can see the optimum threshold value is 0.35



1. **Accuracy for the predicted data**: The prediction accuracy achieved by the model is

```
# Check the accuracy score | Start
accuracy4 = metrics.accuracy_score(y_train_pred_final['Converted'], y_train_pred_final['predicte
# Check the accuracy score | End

round(accuracy4,2)
```

```
0.79
```

2. Sensitivity: Sensitivity of a model with train data is the proportion of positives correctly predicted by it as positives. here from the model we see the 81% are correctly predicted as Yes.

```
# Let's see the sensitivity of our logistic regression model
round(TP / float(TP+FN),2)
```

```
0.81
```

3. Specificity: Specificity of a model is proportion of negatives correctly predicted by the model as negatives. Here from the model we see the 79% is the specificity of the model.

```
# Let us calculate specificity
round(TN / float(TN+FP),2)
```

```
0.79
```

Model have high accuracy (~79%) with a Sensitivity of 81% and Specificity of 79 %

Apply the model on the Test Data:

Accuracy Score of the test data

```
round(metrics.accuracy_score(y_pred_final['Converted'], y_pred_final['predicted_final']),2)
```

```
0.78
```

We got accuracy score of - 0.7871090649309415 in the 3rd run and the final accuracy was - 0.7944118113986347 and accuracy of prediction in the test data is 0.7818518518518518 which is similar without much change

```
## Create the confusion matrix | Start
confusion4 = metrics.confusion_matrix(y_pred_final['Converted'], y_pred_final['predicted_final'
confusion4
## Create the confusion matrix | End
```

```
array([[1294,  383],
       [ 206,  817]])
```

```
TP = confusion4[1,1] # true positive
TN = confusion4[0,0] # true negatives
FP = confusion4[0,1] # false positives
FN = confusion4[1,0] # false negatives
```

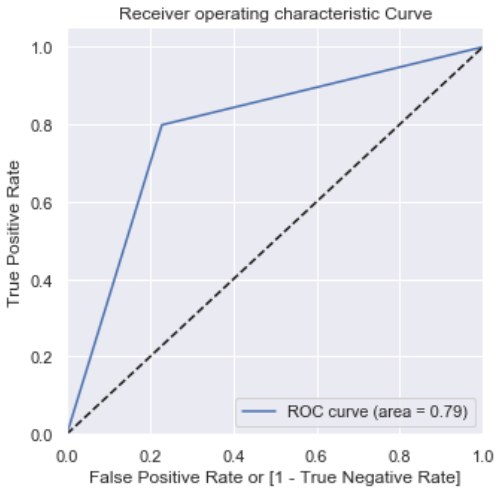
```
# Sensitivity of our logistic regression model
round(TP / float(TP+FN),2)
```

```
0.8
```

```
# Specificity of our logistic regression model
round(TN / float(TN+FP),2)
```

```
0.77
```

ROC Curve of Test Data:



The AUC of the curve is 0.79

Classification Report and Cross Validation Scores:

```
from sklearn.metrics import classification_report
print(classification_report(y_pred_final['Converted'], y_pred_final['predicted_final']))
```

```
              precision    recall  f1-score   support

           0       0.86      0.77      0.81      1677
           1       0.68      0.80      0.74      1023

   micro avg       0.78      0.78      0.78      2700
   macro avg       0.77      0.79      0.77      2700
weighted avg       0.79      0.78      0.78      2700
```

To avoid overfitting, let us calculate the Cross Validation Score to see how our model performs

```
from sklearn.model_selection import cross_val_score

lr = LogisticRegression(solver = 'lbfgs')
scores = cross_val_score(lr, X, y, cv=10)
scores.sort()
accuracy = scores.mean()

print(scores)
print(accuracy)
```

```
[0.76692564 0.76974416 0.79023307 0.79087875 0.79888889 0.80645161
 0.81465039 0.81535039 0.83018868 0.83092325]
0.8014234833760426
```

Our model performs good here with an **mean accuracy score of 0.8014234833760426**

Depending on the business requirement, we can increase or decrease the probability threshold value with in turn will decrease or increase the Sensitivity and increase or decrease the Specificity of the model.

High Sensitivity will ensure that almost all leads who are likely to Convert are correctly predicted where as high Specificity will ensure that leads that are on the brink of the probability of getting Converted or not are not selected.