**Assignment Summary**

X Education sells online courses to industry professionals. The aim of the assignment is to identify leads i.e. potential customers whom the sales team could reach out to.

The company requires us to build a model to identify leads with higher conversion rate. To achieve this, we need to assign lead-score to each of the leads. The customers with high lead-score have a higher conversion chance and the customers with lower lead-score have a lower conversion rate. This enables the sales team to contact leads with higher probability/lead-scores instead of the entire customer dataset.

The process followed:

1. Load the data and perform the following EDA
   1. Execute df.describe(include='all'). This allows us to view the statistics of numeric and non-numeric datatypes
   2. df.isnull().sum() displays number of missing values.

We dropped the columns which over 35% missing rows i.e. > 3000 rows missing (e.g. of columns dropped Asymmetrique Profile Index, Score, Activity Index etc)

There were few columns with values = 'Select' which basically means that the student had not selected the option for that particular column which is why it shows 'Select'. These values are as good as missing values and should be dropped if present in large proportion (e.g. Lead Profile, How did you hear about X Education etc)

We can observe several columns with no data variation (e.g. City, Magazine', 'Newspaper Article' etc). They can be dropped as we can’t derive any insights from skewed data.

Drop the rows with null values that can’t be imputed and small in numbers. (e.g. Lead Source', TotalVisits)

The variables `Prospect ID` and `Lead Number` won't be of any use in the analysis as they are auto generated incremental numbers, so it's best that we drop these two variables.

Club categorical columns like ‘Lead Source’ which have 1 or 2 values or alternatively drop them after get\_dummies.

1. After plotting sns.pairplot, we can see the numerical features are not normally distributed. We will use PowerTransformer to normalise the same.
2. Create dummy variables using the 'get\_dummies' command (e.g. Lead Origin', 'Lead Source', 'Do Not Email) on categorical columns. Drop the original columns and insignificant ones (less number of records, or values with ‘Select’).
3. sns.heatmap helps us to identify the top features & correlation. Since the number of columns are high tabular form is more useful.
4. Perform train\_test\_split and move the target variable to Y.
5. Perform scaling (e.g. StandardScaler) to ensure no feature overshadows the other.
6. Perform model building using LogisticRegression. There are many variables present in the dataset which we cannot deal with. So, the best approach would be to use RFE iteratively to arrive at a smaller subset of features.
7. After all the variables selected by RFE have p-values < 0.05 and VIFs < 5, we can finalise the feature list and make predictions using them.
8. We need to find the optimal cut-off as the values > cut-off will be 1 and less than cut-off will be 0. But in order to get good results, we need to optimise the threshold. To achieve this, plot a ROC curve and find the optimal cut-off between three metrics (accuracy, sensitivity, and specificity)
9. ROC curve plots the true positive rate against the false positive rate and shows the trade-off between them. If the ROC curve is more towards the top left corner of the graph, the model is deemed to be more accurate. Since the area under the curve of the ROC is 0.87 the model is deemed **good**.
10. Create columns with different probability cutoffs and create dataframe to see the values of accuracy, sensitivity, and specificity for each probability cutoff. Since the graph intersects at 0.42 it is chosen as the cut-off i.e. optimal value of the three metrics.
11. Another view is the Precision and Recall trade-off i.e. two curves intersecting threshold value is 0.42. Calculate metrics on train set and if okay proceed to test data.
12. Using the cut-off as 0.42, perform predictions on the test dataset and calculate metrics on Sensitivity, Specificity, Accuracy, Precision to ensure the values are decent and inline with training dataset.
13. The Sensitivity= 0.78, Specificity = 0.79, Accuracy=0.79
14. Precision i.e. probability that a predicted 'Yes' is actually a 'Yes' =0.77 and Recall i.e. Probability that an actual 'Yes' case is predicted correctly = 0.78. The model is good to go