 # Lead-Scoring-Logistic-Regression

X Education needs a machine learning model which will increase their lead conversion beyond 80%. To solve this problem logistic regression model is developed. This will predict a lead conversion and assigns a lead score to each Lead to determine the probability of its conversion.

 # The steps followed in developing this model is as follows: -

1. Importing necessary libraries

We have used NumPy, Pandas for data frame, Matplotlib and Seaborn for visualization and SKLearn for Train Test Split, Scaling and Model evaluation, Stats model, api for building the LRM.

2. Importing and Describing

After importing the dataset, we replace all the 'Select' values in the Data frame with NULL values. We drop the columns which has more than 60% Null values. After the imputations are complete, we remove those rows which have less than 2% null value in their columns. Finally, we have a Dataset with zero null values.

3. Handling anomalies and outliers

Columns such as Page Views Per Visit and Total Visits has outliers, thus we cap the datapoints beyond 95 percentiles to the 95th percentile. Also, we drop those columns which are highly imbalanced and does not play as part in deciding the outcome of the converted column.

4. EDA Steps

We compare all the columns with respect to 'Converted' column, to analyse which attributes play a part in deciding the outcome of a Lead Conversion. We decide which column to use in the Model building. Countplot has been used predominantly for this purpose. We plot the Heatmap (with relavance >+/- 0.7) to identify those attributes, which are corelated and dropped columns accordingly.

5. Splitting the data into Train and Test Data

The final dataset is separated randomly into a 70-30% split dataset for Training and Testing purpose. This is done using SKLearn's train_test_split function.

6. Scaling the data

As the dataset has data of different dimensions, we need to scale the data to make it suitable for a regression model. Since there are several dummy variables created from the categorical values, we opt for Normalization or MinMax scaling so that the 1 and the

0s remain as it is. For this we use SKLearn's MinMaxScaler function. We fit and transform the train dataset only.

7. Building the model
Using the GLM method, we build a regression model in the train dataset. We use RFE function to identify the Top 15 features which we can use in building the model. We drop the rest of the columns from the train dataset and build another model using stats model. We continue to drop variables if the P- value of that variable is > 0.05. We have repeated this process until the P- value is under desired range. After that, We checked the VIF of the remaining variables and find out that all the VIFs are less than 2. To check the VIF we use variance_inflation_factor method.

8. Evaluating the model
We determine the Confusion Matrix and the parameters like Sensitivity, Specificity, etc. We plot the ROC curve, the Accuracy, Sensitivity and Specificity plot, we determine the optimal cut-off at 0.37 and got an
accuracy of 81%.

9. Making predictions on the test dataset
We scale the test dataset with only transform. Then predict the probabilities using the final model. On the test dataset we use the optimal cut-off of 0.37 and get an accuracy of 80.5%.

10. Generating Lead Scores for the sales team for the full dataset
We provide that lead score in a range of 1-100 based on the probability determined by the final model. We have also provided 'Hot Leads' for which the lead score is above 90%. We have extracted the details using Lead number for this.

Final Learnings
In case of X Education, the sales and marketing team must target leads who have,
- Spent more time on their website (Has 100% Feature importance)
- Visited their website a greater number of times
- Their Last activity was SMS or Email
- Placed into Hot Leads Category

Following these traits would increase the lead conversions to a higher percentage.