Lead Score Case Study Summary

Problem Statement: -

An education company "X Education Company" sells online course to various kinds of customers which includes working professionals, unemployed, students & housewife etc. These customers visit the website and browse for courses.

This company is advertising its portfolio of courses & achievements on various platforms like Google, social media & print media etc.

Upon browsing the catalogs of courses present on the website if customers find it relevant, intuitive syllabus and aligned to its goals then it fills the form for contact by providing its email-id, phone number then they are classified as **LEADS.** The company also receives leads from various other sources which includes referrals, email advertisements etc.

But the X Education is facing problem of very low conversion of leads which is 30%, as it is not making much of the profit to the company & this company wants to increase its leads conversion rate to more than 80% so that it makes considerable profit and which could help in expansion of services or portfolio. For this purpose, it wants to identify the "*hot leads*".

Solution Approach: -

Data Cleaning, Inspection & Exploration: -

#) The dataset provided is loaded and its dimension in terms of rows & columns, total number of rows & datatypes of its columns are identified.

#) Upon loading of the dataset, the dataframe is checked for number of missing values (in numeric & percentage) in the columns.

#) Length of converted & non converted leads are identified and if imbalances is found then it is balanced by dropping the few rows of those leads which are high in number so as to bridge the gap.

#) Once the imbalance in the dataset is narrowed find all those cells in which are missing or which are intentionally left blank or by either not selecting the options available in the drop down.

#) Again, inspect the dataframe for the total missing values present in the dataframe.

#) Drop all those columns which have greater extent of missing values (>40%).

#) Analyze all the categorical columns for EDA, while trying to impute the missing values in those columns which have low missing values & drop all those columns which are highly skewed.

#) Analyze all the numerical columns for outliers, if any present then try to cap it, so as not deviate the outcome towards the odd values present in the columns.

#) Upon completion of analyze of both the type of columns (i.e. numerical & categorical) & imputation of missing values, convert all the columns which have 'yes & no' into binary variable of '0s & 1s'.

Data Preparation for Model: -

#) Create the dummy variables of all the categorical columns & drop the original columns once the dummy variables are created.

#) Split the original dataframe in train & test dataframes in the ratio of 70 & 30 respectively.

#) Do the feature scaling of the train data frame.

Model Building & Evaluation: -

#) Build the logistic regression model by selecting most suitable feature using RFE.

#) Once the features are selected then access the model using statsmodel api of sklearn.

#) Drop the variable manually by checking its p-value & vif (variance inflation factor).

#) while refining the model, keep calculating the accuracy, sensitivity & specificity on the train data by making a random guess of the probability.

#) Once the refined model is obtained plot the ROC (Receiver Operating Characteristic) curve to understand the fit of the model.

#) obtain the optimal probability cut-off by plotting the curve of accuracy, sensitivity & specificity for different probabilities & where they intersect that will be the optimal cut-off.

#) Once the final optimal cut-off is obtained calculate the metrics parameters like accuracy, sensitivity, specificity etc.

#) plot the precision recall trade-off curve.

Predictions on Final Test data: -

#) Making the predictions on the test data and calculate its final accuracy, sensitivity & specificity.

Learnings from the Case Study: -

- Observe categorical columns carefully as it may contain default value like select etc.
- Balance the dataframe.
- Identification of columns created by the employees for their tracking purpose.
- Use RFE in case number of features are too large as manual elimination is not possible as it is time consuming process.
- Finding the optimal probability cut-off by plotting the graph of accuracy, sensitivity, specificity.