**Solution: DS Test**

# **Objective: Binary Classification**

**Data Information:** Data set consists of 43 variables, 8 of which are categorical.

# **Preprocessing**

1. Removed the columns that had more than 30% of the nan value.
2. Rows that had more than 1 Nan value were also removed.
3. Rest of the Nan values were replaced by mean of their respective column mean.
4. The percentage of Data retained was 86.03%

# **Working on Outliers**

1. Any value outside the range of [mean-3\*std, mean+3\*std] was considered as an outlier and was replaced by their respective column mean.

# **Model Selection**

A significant class imbalance was seen in the data with {0 : 20201, 1 : 8333}. Class weights are calculated accordingly.

A function named; *train\_models\_and\_evaluate*, was created that took input variables and output targets, scaled the numerical features, and fed into four classification models i.e., Support Vector Machines, Random Forest Classifier, Gradient Boosting Classifier, XGBoost Classifier. The feature selection and engineering was done in the following four ways and their performance were noted by passing it to the *train\_models\_and\_evaluate* function.

* All features with non-zero variance were selected.
* The top 10 features were selected using Chi Square feature selection.
* The top 10 features were selected using Mutual Information Gain for classification.
* Principal component analysis was done bringing features to 10 dimensions.

The best performance model, Gradient Boosting Classifier, with all features, performs best with Test ROC-AUC score of 0.73.

# **Hyper-parameter Tuning**

The model hyper-parameters were further trained to find the best hyper-parameters using GridSearchCV. The tuned parameters were:

* subsample: The proportion of samples to use for creating individual base learners.
* n\_estimators: The number of boosting stages to perform.

# **Training with best parameters and predicting.**

Using the function named *training\_GBC* the model was trained with the best parameters we got from hyper-parameter tuning. The training and testing ROC-AUC score for tuned model was 0.78 and 0.75 respectively.

The function then named *predicting* was used to get the final outcomes. The file named ‘Final Submission.csv’ holds the data in the required format.

**P.S : Some rows in** ‘Final Submission.csv’ **has NaN values, Its because their respective features had NaN values in them.**