REPORT

Hackathon

(a) Pre-processing the dataset

- Checked for missing values, and replaced null value of column with its mode.
- Performed Label encoding to columns before applying Algorithm.
- Applied logistic regression to find out top 29 attributes that the **effort for training the model is reduced by more than half.**

(b) Apply gradient boosting using the function sklearn.ensemble import Gradient Boosting Classifier for training the model.

- **Train accuracy** is achieved by using **n_estimators = 633** in **`Gradient Boosting Classifier`** `i.e. = 0.837308215187415`

(c) Apply gradient boosting using the function from xgboost import XGBClassifier for training the model.

- **Best test accuracy** is achieved by using **n_estimators = 560** in XGBCLassifier i.e. = **0.86606**

We preferred using XGBoost as it become a widely used ,highly flexible and versatile tool .It is a more regularized model formalization to control over-fitting, which gives it better performance. As well as it is a ensemble classifier which create strong classifier .In this context, weak and strong refer to a measure of how correlated are the learners to the actual target variable. By adding models on top of each other iteratively, the errors of the previous model are corrected by the next predictor, until the training data is accurately predicted or reproduced by the model.