Brief info

Previously, we have achieved about 75% accuracy by KNN model because we have used all features to establish a very naive baseline accuracy. But, we can achieve more accuracy. Now, we have only used the features with proper comparison with mutual information scores calculated via mutual info regression. Also, we handled the outliers and dropped some unnecessary features from the dataset. Also, we have encoded and scaled only required features and not like previously when we have scaled all features to achieve poor accuracy.

By only doing this, we are able to achieve 82 % testing accuracy with 90% + ROC/ AUC Curve and 83% F1-score by XgBoost model. But we can get 90% plus accuracy by balancing the data. Due to class imbalance, very good models like Extra tree classifier and XgBoost models are stuck to 80% accuracy. Now, we have balanced the classes by using SMOTE.

This is a type of data augmentation for the minority class and is referred to as the **Synthetic Minority Oversampling Technique**, or **SMOTE** for short.

Working of SMOTE

SMOTE simply duplicates examples from the minority class in the training dataset prior to fitting a model. SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. SMOTE first selects a minority class instance 'a' at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random and connecting a and b to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances a and b.

Insights from dataset (EDA)

In this new notebook, we have done some more EDA and gained some insights which can be observed from the notebook.

Preprocessing the data

We normalize non-categorical continuous data using mean and standard deviations. Also, we encode categorical features using ordinal encoder. Previously, we have normalized all of the dataframe which results in poor accuracy.

Algorithms used

Here, we have tried 4 ML models, Extra Tree Classifier, XgBoost (Extreme Gradient Boosting), random Forest classifier and logistic regression. All the accuracies and metrics can be seen from the notebook.

Extra tree classifier(ETC) proved to be the best model and competition between XgBoost and Random Forest classifier (RFC) models. But XgBoost is a light weight model with size less than 26 MB while ETC has size 9GB+ and RFC has size of 333 MB. So, for deployment purposes, we will use the XgBoost model. All the sizes are with respect to the parameters which can be viewed from the notebook. We have used the pickle library of python to save all the objects.

****