**PROBLEM STATEMENT: To train an ML model on the given TRAIN dataset, make predictions against the TEST dataset, and make predictions for the target column, the “DIAGNOSIS” column. This is a BINARY CLASSIFICATION task.**

**PREPROCESSING THE TRAIN DATASET**

* The features from (feature\_1 to feature\_10 and the patient\_id column) are numerical columns so no **“ENCODING”** techniques were used to convert categorical data to numerical data.
* The diagnosis column (the target column) is categorical as it contains 0’s and 1’s.
* All the columns have 4000 numerical and continuous data entries.
* There are **no MISSING VALUES** in any column.
* There are **no OUTLIERS** in any column except the ‘DIAGNOSIS’ column.
* All columns had almost the **SAME RANGE** and all the values were scaled between 0&1.
* The standard deviation of all the features was around **0.28 indicating a narrow spread**.
* I used **POINT-BISERIAL COEFFICIENT** to calculate the correlation between the dichotomous variables and the numerical features.
* There was **NO STRONG CORRELATION** between the target column and the independent features.

**HANDLING CLASS IMBALANCE**

As discussed above the only **column having outliers** is the target column or the **‘DIAGNOSIS’** column. The number of outliers in that column equals the number of features classified as ‘class-1’. This indicates that the target column has **“CLASS IMBALANCE”.**

**CLASS IMBALANCE:** It refers to a situation in a classification problem where the number of instances of one class outnumbers the instances of one or more classes. This can lead to a bias in the model's predictions, as it may tend to favour the majority class(class-0) over the minority class(class-1).

I tried implementing various methods like **SMOTE, ADASYN, and SMOTE+UNDERSAMPLING** trained the Random Forest model and evaluated its **AUC-ROC, ACCURACY, PRECISION,** and **RECALL.**

The **best results** were produced when I used the **BORDERLINE-SMOTE**. It gave better results than the latter.

**FINDING THE BEST HYPERPARAMETERS FOR RANDOM FOREST**

I used **‘GRIDSEARCH-CV’** and **initialized** them with **5 hyperparameters** to know which set of hyperparameters would optimize the model's performance on various metrics.

The best set of hyperparameters are {'bootstrap': False, 'max\_depth': None, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 300}

The corresponding metric scores for the above hyperparameters for a random forest model is as follows

ROC-AUC Score with Best RandomForestClassifier: 1.0

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

F1 Score: 1.0

Confusion Matrix:

[[766 0]

[ 0 756]]

This is **a very good confusion matrix** as the random forest model with the optimal parameters has **no false positives and false negatives**.

**THE CROSS-VALIDATION**

**To ensure that these metrics aren’t occurring due to overfitting of the random forest model I used “STRATIFIED CROSS VALIDATION” to evaluate the best fit random forest model on unseen data**.

Cross-Validation Results:

Accuracy: 0.9753 ± 0.0046

Precision: 0.9809 ± 0.0059

Recall: 0.9695 ± 0.0040

F1 Score: 0.9752 ± 0.0046

ROC-AUC: 0.9753 ± 0.0046

The cross-validation results indicated that the best-fit random forest model was still **performing exceptionally well on unseen data** so I decided to **proceed to train** **the random forest model using the optimal parameters and the resampled data from handling class imbalance and make it ready for predictions.**

**THE VALIDATION USING TEST DATASET**

I generated predictions on the best-fit model using the predict function on the test dataset against the features provided in the test dataset and saved them against the patient\_id in a predictions CSV file.

The class-0 points appeared 981 times in the prediction column of the predictions CSV file

The class-1 points appeared 19 times in the prediction column of the predictions CSV file

**CHALLENGES FACED**

The major challenges I faced in this task were handling the class imbalance and finding out which model to use such that the evaluating parameter AUC-ROC score is maximized.

I used SMOTE, ADASYN, SMOTE+UNDERSAMPLING, and BORDERLINE SMOTE on Random Forest and Gradient Boosting and tried out all 8 possibilities to find out which model is giving the maximum AUC-ROC score on which sampling techniques.

I then chose Random Forest with Borderline SMOTE.

The next challenge was to identify the optimal hyperparameters for the model which was time-consuming as it took an ample amount of time to execute.