# Readme file

## What we have tried out -

We have tried to tune the model using various hyperparameters where we varied the hyperparameters such as class_weight criterion, Splitter, min samples split, max depth, min samples leaf, max features (sqrt/log2), and max leaf nodes. Here we varied all these parameters and studied how we could increase the model. But unfortunately, changing the values of parameters did not result in better accuracy.

So then we studied all the models and tried them out (SVM, KNeighbors, AdaBoost, SGDC, Random forest and many more). This could not result in the accuracy to increase more than 60.3%. We also used pfeature to convert the initial data into required format. We tried multiple models by changing our number of columns in training data from 17 to around 340 using pfeature but our performance was around 60% only.

So then we tried combining all these models into a stack classifier(we used RandomForest,KNN,Decision Tree and Logistic Regression for stacking classifiers). Here we have tried out various different combinations and note down the results for the same. But again we could not increase the accuracy to more than 59%.Then we tried ensembling + bagging + boosting + xgboost(from XGBClassifier).This helped in increasing the accuracy to 61.5.Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. We applied this with bagging and boosting techniques, but no drastic change occured.

Then we studied about 1-hot encoding and tried to apply this technique to our dataset to note changes. And here we discovered a slight increase in our accuracy. Then we tried going to Random forest.So using **1-hot encoding** and **random forest** as a classifier our performance increased a little bit. Then, we used the hint posted in the classroom and instead of predicting the labels 0s and 1s for submission file labels we used the predicted probability values. Hence for predicting the labels we used '**predict_proba**' instead of 'predict' function. This results in a high increase in our performance from around 60%-61% to 66%.

## Technique we used -

1) Ensembling - Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model.
2) Bagging - It is a method where random data samples are chosen with replacement in order to reduce variance present in data.
3) Boosting - It is a method where random data samples are chosen without replacement and in each successive round we increase the weight of non selected data samples.

4) XGBoost - It is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.
5) 1-hot Encoding - In this method we convert each categorical value into a new categorical column and assign a binary value of 1 or 0 to those columns. Each integer value is represented as a binary vector.
6) predict_proba - It gives us the probabilities for the target (0 and 1 in our case) instead of 0 or 1 in array form.
7) Random forest - It is an ensemble method for decision trees where we combine multiple decision trees for predicting a answer.