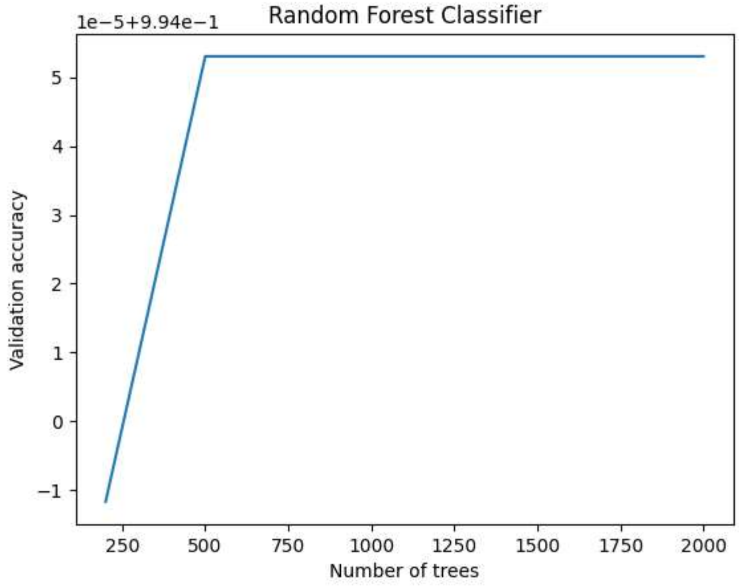
4 Base models with 21 setting was tested in this week's project. Validation accuracy are used as the score of how good the model is and F1 score are used to act as a supportive evidence.

First base model: Random Forest Classifier:

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

A set of n_estimators(the number of trees in the forest) is tested. Lower number of trees seem will affect the accuracy. But since random forest has a good over- fitting control too many trees seem won't affect the validation accuracy that much.

Here is the graph of n_estimators vs Validation accuracy:

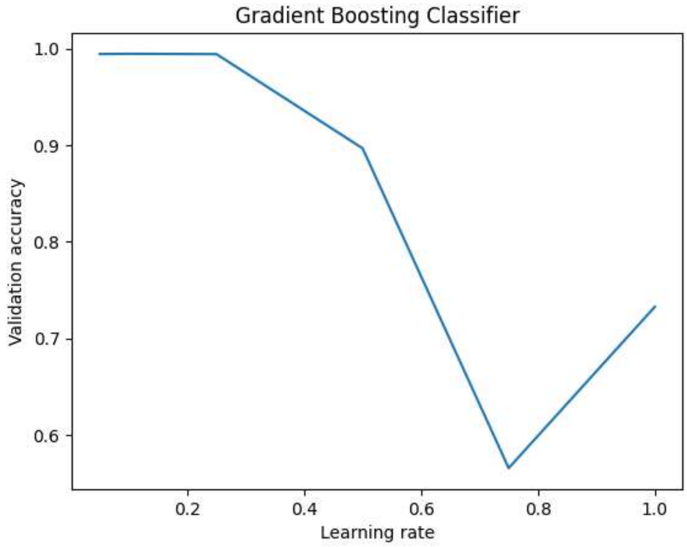


Second base model: Gradient Boosting Classifier:

Gradient boosting is a machine learning technique based on boosting in a functional space, where the target is pseudo-residuals rather than the typical residuals used in traditional boosting. It gives a prediction model in the form of an ensemble of weak prediction models.

A set of learning rate is tested for GB:

The learning rate is a hyperparameter that needs to be carefully tuned, often using techniques like cross-validation. The optimal learning rate is typically dataset-specific and can greatly affect the performance of a Gradient Boosting model. It's common to use a lower learning rate in conjunction with a higher number of boosting rounds (trees), which can lead to better model performance but at the cost of increased computational resources and training time.

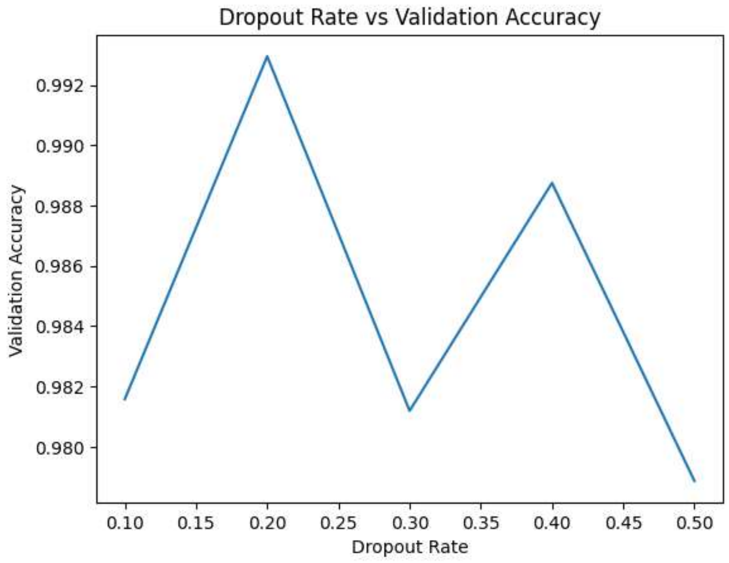


Base model 3: CNN model using TensorFlow.

Convolutional neural network (CNN) is a regularized type of feed-forward neural network that learns feature engineering by itself via filters (or kernel) optimization. Vanishing gradients and exploding gradients, seen during backpropagation in earlier neural networks, are prevented by using regularized weights over fewer connections.

A set of drop out rate is tested for CNN:

A higher dropout rate means that a larger proportion of neurons are randomly ignored during training at each iteration. This can significantly reduce overfitting since it forces the network to learn more robust features. By not relying on any single neuron, the network has to find more general patterns that are present in the data, which typically leads to a better generalization on unseen data. However, if the dropout rate is too high, it can lead to underfitting, as the network might not be able to learn the training data sufficiently.

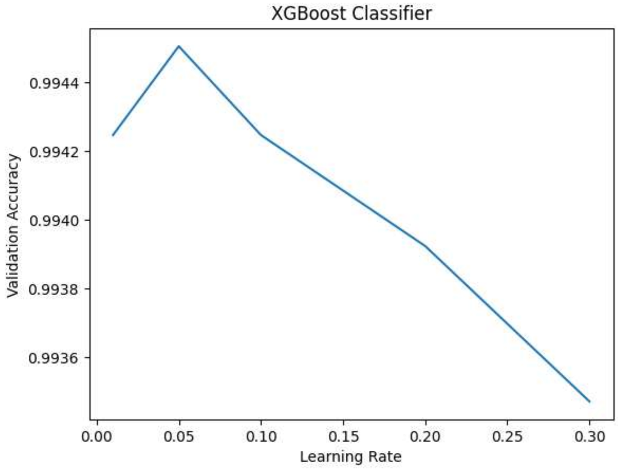


Base model 4: XGB (Newly added this week for testing.

XGB is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

A set of learning rate is tested for XGB:

The learning rate is one of the most important hyperparameters in XGBoost and affects both the speed of convergence and the quality of the final model. It requires careful tuning to strike the right balance between learning too quickly and potentially missing the optimal solution, and learning too slowly, resulting in long training times or getting stuck in suboptimal solutions. The optimal learning rate can vary significantly depending on the dataset and the specific characteristics of the problem being solved.



Random forest is the only model that can reach about 99.6% validation accuracy so a random forest model is choose to be the winner of all 4 models and will be used for this project later. Other models may also be able to reach this score or be even better than Random Forest, but they will require more tuning and more hyperparameters needs to be tested.

The hyperparameters for random forest will be n_estimators = 1000 and max depth of 10. This is tested to be the best for random forest on this problem.