Team : Anna Moy & Natalie Kalukeerthie

After conducting the exploratory data in the previous assignment I conducted pre-processing of the data to prepare it for the models. We proceeded in running the different model and look at the different performance.

Model 1: Baseline Decision Tree

The goal of this experiment was to establish a baseline using a Decision Tree classifier with default settings. Starting with this simple and interpretable model allows us to better understand the initial performance before applying any complexity or tuning.

We did not modify any parameters except for setting the minimum samples split to 2, keeping the tree depth at its default value. The data was cleaned beforehand to address any class imbalance, ensuring a fair model training process. We used the entire dataset and all available features, without removing any data, to maintain a true baseline. The dataset was split into training and testing sets with an 80/20 ratio, and this split was consistently used across all models in subsequent experiments.

For evaluation, we selected accuracy, precision, recall, and F1-score. Using multiple metrics provides a more comprehensive understanding of model performance, as relying solely on accuracy can be misleading. Together, these metrics give a clearer picture of how well the model handles different aspects of the data.

Given the simplicity of the baseline Decision Tree, we expect it to exhibit high bias since it likely cannot capture complex patterns within the data. On the other hand, the variance is expected to be low because the default settings limit the tree’s depth and complexity.

For comparison on the performance of the model a cross validation with 5 folds was used to see how the performance would be instead of relying on the 80/20 split from training. The performance result was better than the baseline decision tree. This model provides low variance and bias because it is consistently using 5 folds across the data. The accuracy at 88% and precision at 92% and f1-score at 93% which shows the balance between false positives and false negatives.

Model 2: Decision Tree, depth 10 and split 4

The goal for the second decision tree is to increase the depth and split of the model which would increase performance and capture more complex patterns in the data. We increased the depth to 10 and increased the split at 4 which should increase the performance of the decision model. We choose to have a depth of 10 and a split of 4 to add more complexity and to prevent it from overfitting.

The evaluation will remain the same using accuracy, precision, recall and f1-score. The bias in the model would have been reduced since we are increasing the model to look into the patterns but this could cause the model to overfit. The variance in the model increased since I added more splits to the data.

As a result, the accuracy was slightly higher in the baseline model than the tuned decision tree. But the tuned model was better in predicting the positive case and capturing the positive case as the results were higher in precision and recall. I would say the tuned model is better than the baseline model since the accuracy, precision, recall and f1-scores are better than the baseline decision tree.

With decision trees it can easily overfit especially when we are increasing the depth and mini split on the decision tree . But in this case, the results between training and test sets shows there’s no sign of overfitting. Both models seem to generalize well to the data. In our coding we also showed visually the difference between the two decision trees for comparison and easy interpretation.

Metrics	Baseline Decision Tree	Cross Validation, 5 fold	Depth 10, 4 split Decision Tree
accuracy	.8993487	.8821609	.9160378
precision	.8634032	.9202256	.8897624
recall	.9488055	.9488056	.9497448
f1-score	.9040920	.9343947	.9187757

Model 3 - Random Forest, 100 trees

The objective of this experiment was to evaluate the performance of a Random Forest model with a limited number of 100 trees. This serves as a benchmark to see how well a Random Forest performs compared to the Decision Tree baseline, while keeping the number of trees relatively low due to computational constraints.

Instead of using the default number of trees I manually set the number of trees to 100 because the model was complicated and would take a long time to run. All other parameters remained at their default settings. The dataset and features remained unchanged for consistency.

I consistently used accuracy, precision, recall, and F1-score as metrics because they are easy to interpret and allow for clear comparison across models at the very end to evaluate which model is the best out of all of them.

In this model there is high bias because I am only including 100 trees and not allowing it to run the default parameter. The variance is low since I am including all the features in the model.

The model achieved an accuracy of 89.47%, with precision at 90.42%, recall at 98.51%, and an F1-score of 94.29%. These results indicate strong performance and with the high recall, which suggests the model is effective at identifying positive cases. Compared to the Decision Tree models, this Random Forest showed a clear improvement in all key metrics.

Model 4 - Random Forest, 500 trees and 4 features

The objective is to find out if increasing the number of trees and limiting the number of features considered at each split would lead to better performance. This tests the hypothesis that more trees provide better generalization, and that feature reduction may enhance model efficiency. We increased the number of trees to increase stability and reduce variance

I increased the number of trees from 100 to 500 and restricted the number of features considered at each split to 4. This change increases ensemble diversity and potentially reduces overfitting by limiting feature overlap.

The metrics used are accuracy, precision, recall and f1-score across the model to have an apple to apples comparison in the two Random Forests and able to compare it with the other models.

With 500 trees, bias is expected to be lower due to greater model complexity and ensemble averaging. Variance remains low as we are still including 4 features.

The model achieved an accuracy of 89.49%, precision of 90.39%, recall of 98.58%, and an F1-score of 94.31%. While precision was slightly lower than the Random Forest 100 model, the recall and F1-score improved. This suggests that increasing the number of trees helped the model capture more complex patterns, while limiting features did not significantly hurt performance.

Metrics	Random Forest, 100 trees	500 tree, 4 feature, Random Forest
accuracy	.8947252	.8949464
precision	.9042419	.9038925
recall	.9850971	.9858485
f1-score	.9429393	.9430933

Model 5 - Adaboost, 1000 sample size, 50 decision trees

The goal of this experiment was to establish a reference point for AdaBoost performance using a smaller subset of the dataset due to computational limitations. Since we couldn’t run AdaBoost on the full dataset, we randomly sampled 1000 rows and trained the model with 50 decision trees. This acts as a starting point for tuning and comparison.

We consistently used accuracy, precision, recall, and F1-score to ensure comparability across all models. These metrics allow us to assess both the overall performance and how well the model handles false positives and false negatives.

There is high bias due to the limited sample size we are using for the model. The model has low variance and less likely to overfit because we are limiting the trees at 50.

This model performed the worst among all models tested so far in terms of accuracy and recall, likely due to the limited sample size and the low number of estimators. While precision was high, indicating that the model was accurate when it predicted a positive class, it missed many true positives, as shown by the low recall.

Model 6 - Adaboost, 2000 sample size, 150 trees

The objective is to improve the previous AdaBoost model by increasing both the sample size and the number of decision trees. The hypothesis was that more data and more boosting rounds would help capture more patterns and improve model performance. Increasing the sample size and the decision tree numbers will help me reduce bias.

We changed the sample size from 1000 to 2000 and the iteration from 50 to 500 for the model and we kept the factor as 5 for both models.

Metric used are accuracy, precision, recall and f1-score for consistent comparison at the end.

All four metrics improved slightly but were not as significant as expected by increasing sample size and decision trees. We may have to tune other features to increase the performance significantly.

Metrics	1st Adaboost	2nd Adaboost
accuracy	.6608579	.6897047
precision	.9374632	.9418188
recall	.6599503	.6912962
f1-score	.7746011	.7973422

After systematically experimenting with the multiple models (two Decision Trees, two Random Forests and two Adaboost) the tuned Random Forest has the best performance out of all the models.

The Random Forest model achieved the highest balance of accuracy (89%), recall (98%), and F1-score (94%) among all experiments. The Random Forest model (500 trees and 4 features) has the highest recall which is important when picking a model because we want to minimize the number of false negatives. Through the experimentation process, we learned that increasing the number of trees and features in the Random Forest improved its ability to generalize, reduced bias, and captured more complex decision boundaries than the simpler Decision Trees or the baseline Random Forest. Compared to AdaBoost and Decision Trees, Random Forest demonstrated the most consistent and reliable performance across training and test sets, indicating it is the most stable and effective model for this classification task.

From the experiments, it is clear that systematically adjusting model parameters helps reveal important trade-offs between bias and variance. The Decision Tree models were fast and easy to interpret, but their performance plateaued even when the tree depth was increased, showing limited improvement. In contrast, the Random Forest models performed better overall because combining multiple trees improved generalization and minimized the risk of overfitting. The AdaBoost models showed potential for higher recall but were more sensitive to parameter tuning, requiring more careful adjustments to achieve consistent results.

For future work, additional hyperparameter tuning and feature selection could further improve performance, but within the scope of this experimentation process, the Random Forest clearly provided the most reliable and well-balanced results.

Metrics side by side comparison

Metrics	Accuracy	Precision	Recall	F1-score
Baseline Decision Tree	.8993487	.8634032	.9488055	.9040920
Depth 10, 4 split Decision Tree	.9160378	.8897624	.9497448	.9187757
Random Forest, 100 tree	.8937300	.9054491	.9822167	.9422719
500 tree, 4 feature, Random Forest	.8949464	.9038925	.9858485	.9430933
1000 sample, 50 decision tree, Adaboost	.6608579	.9374632	.6599503	.7746011
2000 sample, 150 decision tree, Adaboost	.6897047	.9418188	.6912962	.7973422