Assignment 6 – Report

Logistic Regression Model

For the Logistic regression model, I print a classification report for the test set of the model. To check if there is an overfitting problem in the model, I also generate a report for the training set and compare the values. The accuracy value of the model is 0.85, which shows that the model correctly classified 85% of the test data. When we look at the class differences, the model is correct in 88% of its predictions for the income class of 50k and below (precision: 0.88). In addition, it really predicted 94% of the income class of 50k and below (recall: 0.94). On the other hand, the model is correct in 74% of its predictions for the income class of over 50k (precision: 0.74). Also, it predicted 59% of the income class of over 50k (recall: 0.59). The low precision and recall values for the class above 50k caused the F1 score of this class (.066) to be lower than the class below 50k (0.91). Finally, the very small difference in training accuracy (85.21%) and test accuracy (85.16%) shows that there is no overfitting problem in the model.

After optimizing the C hyperparameter of logistic regression, for 50k and above, precision (0.75) increases slightly, while recall (0.58) and F1 score (0.65) decrease slightly. The optimization seems to have not worked very well :(

Decision Tree Classifier

I print a classification report for both the training and test sets of the Decision tree model to evaluate its performance and detect potential overfitting. The training set accuracy is 1.00, indicating that the model perfectly classified 100% of the training data. The precision, recall, and F1-score values for both classes (50k and below and over 50K) are also 1.00, meaning the model memorized the training data entirely. On the test set, the accuracy drops to 0.82, meaning the model correctly classified 82% of the test data. When we examine the class-wise performance we see that income class of 50k and below, the precision, recall, and F1-score values are 0.88, indicating consistent performance in correctly identifying this majority class. For the income class of over 50k, the precision is 0.62, meaning the model is correct in 62% of its predictions. The recall is 0.61, showing that it correctly identified 61% of the actual over 50k. These relatively low precision and recall values for the minority class resulted in an F1-score of 0.62, significantly lower than the F1-score for the 50k and below (0.88). The difference between the training accuracy (100%) and test accuracy (82%) highlights that the model is overfitting the training data.

After applying hyperparameter tuning to the Decision tree model, I checked the result if the tuning worked. The optimized model shows clear improvements in terms of generalization and reduced overfitting. The training set accuracy of the optimized model is 0.85, compared to the previous overfitted model's perfect score of 1.00. This indicates that the model no longer memorizes the training data but instead learns more general patterns. For the income class 50k and below, the model achieves a precision of 0.86, a recall of 0.96, and an F1 score of 0.91, showing strong and consistent performance. However, for the income class of over 50k, the model's performance drops, with a precision of 0.79, a recall of 0.51, and an F1 score of 0.62. On the test set, the optimized model achieves an accuracy of 0.84, only slightly lower than its training accuracy, indicating that the model generalizes well. For the 50k and below class, the precision (0.85), recall (0.96), and F1 score (0.90) values remain high, confirming the model's ability to consistently classify the majority class. For the over 50k class, the precision is 0.78, the recall is 0.49, and the F1 score is 0.60, which aligns closely with the training results and confirms that overfitting has been reduced.

Random Forest

The Random forest model achieves again perfect performance on the training set, with an accuracy of 1.00 and precision, recall, and F1 scores of 1.00 for both classes. On the test set, the accuracy drops to 0.86, showing that the model generalizes better than the unoptimized Decision tree. For the income class of 50k and below, the model performs well, with an F1 score of 0.91. However, for the over 50k class, the precision is 0.74, recall is 0.62, and F1 score is 0.68, reflecting a moderate imbalance in the model's performance across classes. The gap between training and test performance shows overfitting.

After optimization, the Random Forest model shows improved generalization, with the training accuracy dropping to 0.92 (from 1.00) and the test accuracy remaining at 0.86, indicating reduced overfitting. The performance for the over 50k class has slightly improved, with an F1 score of 0.68, and class balance is better reflected. The optimized hyperparameters (max_depth:30, n_estimators:50) effectively balance the model's complexity, making it more generalizable.

Conclusion

Among the three models tested, the **optimized Random Forest** is the best choice. It achieves the highest test accuracy (0.86) while maintaining a better balance between training and test performance, demonstrating reduced overfitting after optimization. Unlike the Decision Tree, which heavily overfits the training data, the Random Forest generalizes well due to its ensemble nature. Additionally, its F1-score for the over 50k class (0.68) is higher than that of Logistic Regression (0.66), making it better suited for handling the class imbalance. Therefore, the optimized Random forest provides the best trade-off between accuracy, class balance, and generalization, making it the best option for this data.