Project 2 Report

Techniques Used to Train the Models

This paper entails a study of California housing data to predict whether house prices are above the given median. Various classification techniques were attempted and contrasted to determine the most optimal one. The following supervised learning techniques were used for classifying house prices: K-Nearest Neighbors (KNN), Decision Tree Classifier, Random Forest Classifier, and AdaBoost Classifier. These models use a number of methods such as distance-based classification, tree-based decision, ensemble learning, and boosting algorithms. All of these models were trained with stratified train-test split in order to preserve the same class distribution. Training was performed by fitting the models to the training dataset and subsequently using the learned parameters to predict on the test dataset.

Techniques Used to Optimize Model Performance

Different optimization techniques were utilized to enhance model performance. StandardScaler was mainly used to normalize features and standardize data to enhance model performance in distance-based models like K-Nearest Neighbors, which rely on Euclidean distances. Grid search and empirical tuning were used for hyperparameter tuning to optimize parameters for different models. This would help optimize the results. For example, neighbors in KNN were optimized to find the accuracy and realistic value, while max depth and number of estimators were tuned and refined in Decision Tree and Random Forest classifiers. Ensemble learning was also employed in Random Forest and AdaBoost to maximize accuracy and avoid overfitting and data leaks by combining a set of weak classifiers.

Comparison of Model Performance

Performance of all models was evaluated in terms of accuracy, precision, recall, and F1-score. K-Nearest Neighbors (KKN) performed reasonably well but was not effective for large datasets since it is computationally expensive and sensitive to the value of k. Decision Tree overfitted a bit and had a bit of data leak, leading to high training set accuracy but poor generalization in the test set. Random Forest was also able to achieve decent performance through ensemble learning to reduce overfitting and improve accuracy through the aggregation of numerous decision trees. AdaBoost was best achieved through the aggregation of weak classifiers and iterative adjustment of

misclassified points. The performance of AdaBoost, however, is based on the choice of the base learner and may be affected by noisy data.

Model Performance Summary

Model	Precision	Recall	F1-score	Accuracy
K-Nearest Neighbors (KNN)	0.82	0.82	0.82	82%
Decision Tree Classifier	0.84	0.84	0.84	84%
Random Forest Classifier	0.89	0.89	0.89	89%
AdaBoost Classifier	0.84	0.84	0.84	84%

Recommended Model for This Dataset

By performance criteria, the best model for this data is the Random Forest Classifier. It is a compromise between accuracy, readability, efficiency and overfitting resistance that makes it ideal for this classification task. Decisions Trees are susceptible to overfitting but Random Forest prevents this because it averages out many decision trees, leading it to generalize more. It also provides feature importance scores that make it simple to understand which features most drive the predictions.

Most Important Metric and Its Significance

The best metric for this problem is the F1-score, since it will strike a balance between precision and recall. Since whether or not house prices are above the median has both buyer and seller implications, having as few false positives as false negatives is necessary. If precision is made too low, then many lower-priced houses would be labeled as being expensive and mislead potential buyers. On the other hand, if there is too little recall, many costly houses will be misclassified as affordable and impact market predictions. Thus the F1-score achieves a balance between these two attributes and provides the best measure for this problem.

Conclusion

Through exploratory data analysis and model comparisons determined the Random Forest Classifier provides the optimum accuracy and reliability balance. F1-score was

used to guarantee good classification while minimizing misclassification error. Preprocessing techniques such as feature scaling and hyperparameter tuning also helped significantly in terms of model performance. Future improvements could involve exploring deep learning approaches such as neural networks or incorporating additional features to enhance model robustness. Furthermore, fine-tuning hyperparameters using advanced techniques like Bayesian optimization could lead to further improvements in model performance.