Which techniques did you use to train the models?

I used four supervised learning models: K-Nearest Neighbors, Decision Tree, Random Forest, and AdaBoost. Before I trained the model, I cleaned the dataset by checking and removing the duplicates and graphing the data to ensure it was correct. In order to train the model, I first split the dataset into training and testing sets using a stratified split to maintain the proportion of the target variable. In order to improve the model, I then standardized features using a StandardScaler. Finally, I fit each model to the training data using their respective fit() methods.

Explain any techniques used to optimize model performance?

To optimize performance, I standardized features using StandardScaler so that distance-based algorithms like KNN perform optimally. In addition, I tested different test/train splits to find the best split to ensure that the class distribution of the target variable is preserved in both the training and testing sets, reducing sampling bias.

Compare the performance of all models to predict the dependent variable?

KNN: This model generally performed well when features were standardized, but its performance can be sensitive to the choice of neighbors and local data variations. In addition, I noticed that because I tested so many neighbors (1-21), this model took a noticeably larger amount of time to run.
Decision Tree: This model provided an easily interpretable model. However, it can be prone to overfitting if not properly pruned or tuned. I noticed that despite the high performance, this model had the most false positives and false negatives.
Random Forest: This model performed the best overall. Random forest had a great performance by averaging multiple trees, which reduced overfitting and was able to deliver better performance.
AdaBoost: This model also performed pretty well by sequentially correcting misclassifications, though it can be sensitive to noisy data. I think this performed well on this dataset because this dataset was pretty clean.

Overall, ensemble methods (Random Forest and AdaBoost) tended to show a better balance between precision and recall, with Random Forest often edging ahead in overall performance.

Which model would you recommend to be used for this dataset?

For this dataset, I would recommend using the Random Forest classifier. Its ensemble nature provides robust, high-performing results by reducing variance and handling complex, non-linear relationships effectively. It consistently achieved higher scores across key evaluation metrics in our experiments.

For this dataset, which metric is more important, why?

For this dataset, I think recall is one of the more important metrics because the goal is to predict houses priced above the median. Recall is more important because missing a high-value house (i.e., a false negative) could be more costly or critical from a business perspective than a false positive. In addition, I need to consider the balance. If both false positives and false negatives are of concern, the F1-score, which balances precision and recall, would be an excellent metric. However, in this context, ensuring that most high-value houses are correctly identified, meaning high recall, is more important.