COE 379I

Project 2: California Housing Classification

By: Jeet Patel EID: jcp4345

Techniques to Train and Optimize Models

The dataset was trained by separating features from the target variable (price_above_median) and splitting it into training and testing sets using a 70:30 ratio while maintaining class balance through stratified sampling. Feature scaling with StandardScaler ensured that all features had a mean of 0 and a standard deviation of 1, which was important for distance-based algorithms like K-Nearest Neighbors (KNN) and tree-based models like Random Forest and AdaBoost. To optimize performance, feature scaling was applied to standardize the dataset, and GridSearchCV was used for hyperparameter tuning. The optimal number of neighbors (n_neighbors), weighting strategy (weights), and distance metric (metric) were determined for KNN, while Random Forest underwent fine-tuning of parameters like the number of estimators (n_estimators), tree depth (max_depth), minimum samples for splitting (min_samples_split), minimum samples of leaves (min_sample_leaves) and Adaboost using learning rate (learning_rate) and number of estimators (n_estimators). 5-fold cross-validation was incorporated to ensure robust evaluation and prevent overfitting for Random Forest and K nearest neighbor and 3-fold cross-validation for Adaboost.

The best Hyperparameters for each method are listed below:

KNN: Best Hyperparameters: {'metric': 'manhattan', 'n_neighbors': 21, 'weights': 'distance'}

Random Forest: Best Hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 93}

AdaBoost: Best Hyperparameters: {'learning_rate': 0.1, 'n_estimators': 50}

Model Performance Comparison

Models	Precision (test)	Recall (test)	F1-score (test)	Accuracy (test)	Precision (train)	Recall (train)	F1-score (train)	Accuracy (train)
K-nearest neighbor	0.85	0.87	0.86	0.86	1	1	1	1
Decision Trees	0.83	0.84	0.83	0.83	1	1	1	1

Random Forests	0.89	0.9	0.9	0.9	1	1	1	1
AdaBoost	0.89	0.91	0.9	0.9	0.95	0.96	0.95	0.95

Comparing the models first time wise to run, from fastest to slowest, our fastest method was Decision Trees which makes sense as it only has two options to classify, next was K-nearest neighbor, followed by AdaBoost in which AdaBoost took slightly longer than K-nearest neighbor to run, and finally we have Random Forests which took significantly longer than the other models to run which makes sense due to the amount of decision trees there are in hand with the combination of all the independent variables in relation to the dependent variable causing for a longer time to find the best accuracy for our model. In terms of accuracy of the model from best to worse: we had a tie of AdaBoost and Random Forest but with Adaboost having a slightly higher recall it edges out our Random Forest model, next we had K nearest neighbor with an accuracy of 0.86, with the worst performing being Decision trees with accuracy 0.83 for the TEST data. In all metrics the same order follows as in the accuracy metric showing the ranking of our models for this dataset are as follows: Adaboost (1), Random Forest (2), KNN (3), and Decision Trees (4).

Recommendation for Model

I recommend using the AdaBoost model for this dataset. While Random Forest similar accuracy as AdaBoost produced of 90% the recall of Adaboost was very slightly higher at 0.91 over the 0.9. However, AdaBoost ran significantly faster, making it a more efficient model for achieving the same accuracy while reducing computation time. As compared to the other two models while they were faster they were less accurate by a large portion in which it would not be worth it to give the accuracy for a slight speed increase. This balance of speed and performance makes AdaBoost the preferred choice for this problem.

Most Important Metric

Accuracy is the most important metric for this model because any misclassification can lead to significant financial consequences for homeowners, investors, and the real estate market. If the model overvalues a property, buyers may overpay, leading to inflated prices and potential financial losses when they attempt to resell. Investors who rely on these

valuations could make poor investment decisions, ultimately losing money. Conversely, if the model undervalues a property, homeowners may see their assets devalued, limiting their ability to sell at a fair price or secure favorable loan terms. This misrepresentation can also negatively impact neighborhoods by artificially lowering property values. Since both overvaluation and undervaluation have direct financial implications, ensuring the highest possible accuracy in predictions is crucial for maintaining fairness and stability in the market.

ChatGPT Uses

I used ChatGPT on a few parts of this assignment. The 1st use was having it help me format a table for the statistical information, when I was doing it the values would be off center. I also utilized ChatGPT to fix a warning when I was running k nearest neighbor where the feature names wouldn't be properly labeled after the Standard Scalar and it gave me the solution to make it back into a dataframe. Those were my uses of ChatGPT in this assignment.