Project 2 Report: Predicting house prices.

Data Preparation:

1. Identified shape, size, and basic information of dataset.
2. Checked for duplicates, missing values, and unique values of dataset.
3. Found the statistical information of the dataset, including mean, standard deviation, minimum, maximum, median, and 25%, 50%, 75% quartiles.
4. Performed univariate analysis using histograms for distribution analysis, boxplots for outlier detection, correlation matrix, and scatter plots to explore relationships.
5. Data Splitting: dataset was split into a test size of 20%, and a random state of 42 to ensure reproducibility.
6. Data Standardization: the features were standardized using StandardScaler, to ensure all features have a mean of 0 and a standard of 1, for model optimization.

Insights from data preparation:

- Median Income (MedInc): the mean is higher than median, indicating right-skewed distribution. There are significant outliers of high-income values that contribute to this result.
- House Age (HouseAge): the distribution is relatively uniform, with some concentration around between about 18-36 years. Not any significant outliers or anomalies.
- Average Rooms (AveRooms): the average number of rooms per block has a right skewed distribution, because of some outliers with very high number of rooms like 26 or 34 rooms.
- Population: the population has a right-skewed distribution with few high values, possible outliers.
- Latitude and Longitude: These features are uniformly distributed within the range of this dataset.
- Price Above Median (price_above_median): Roughly balanced, dataset is well sampled since this is a binary feature.
- MedInc seems to have a positive correlation with price_above_median, which makes sense because higher income neighborhoods tend to have higher house prices.
- AveRooms and AveBedrms are highly correlated, as expected, because more average rooms within a block means more average number of bedrooms within a block.

Model Training Procedure:

1. Model Initialization: Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, and AdaBoost classification models were initialized.
2. Model Training: Each model was trained on the standardized training data with fit method.
3. Hyperparameter Tuning: Grid Search with cross-validation (GridSearchCV) was used to find the best hyperparameters for each model. Defined a grid of possible hyperparameters and searching for the combination that resulted in the best performance based on cross-validation accuracy, reducing risk of overfitting.

Model Performance:

Model Performance was measured on accuracy, precision, recall and f1 scores. Model was first measured without hyperparameters, and then with hyperparameters to help solve some overfitting problems and higher performance. See scores in Jupyter Notebook for reference.

- Random Forest model performed best with highest testing accuracy of 89.77% and strong precision, recall, and f1-scores of 90%. Model generalizes well while maintaining robustness.
- AdaBoost model performed second best with 87.74% testing accuracy, and strong precision, recall, and f1-scores of 87-88%. It has strong predictability but just performs slightly lower than Random Performance.
- K-Nearest Neighbors had a training accuracy of 100%, which might indicate high chances of overfitting, because its testing accuracy was 83.55%.
- Decision Tree showed small signs of overfitting, with a training accuracy of 90.96% but the testing accuracy was 85.63%.
- Logistic Regression had a balanced performance of testing accuracy of 83.81% but it did not perform better than Random Forest or AdaBoost models.

Best Model:

The best model that performed is Random Forest Model. The Random Forest Classifier model is recommended for this dataset due to its high accuracy and f1 score.

Most Important Metric:

F1 score is the most important metric because it balances both precision and recall, is the best metric for this dataset as it is crucial for classification problems and model performance.