North South University

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ASSIGNMENT REPORT

Predictive Modeling of Housing Prices in Boston Using Machine Learning Techniques

Course Information
Machine Learning
CSE445 (Section 4)
Summer 2025

Submitted by
Saif Mohammed 2121913042

Submitted to
Dr. Mohammad Abdul Qayum [MAQm]
Assistant Professor
Department of Electrical and Computer Engineering
North South University

Submission Date
August 4, 2025

Contents

1 Abstract

This report presents a comprehensive machine learning assignment conducted on the Boston Housing dataset. The analysis encompasses data preprocessing, exploratory data analysis (EDA), regression modeling, hyperparameter tuning, classification, ensemble learning, and feature importance interpretation. The results indicate that the Random Forest Regressor performs best in predicting continuous house prices, while Logistic Regression achieves the highest accuracy in categorical classification.

2 Objective

The purpose of this report is to explore the Boston Housing dataset and develop predictive models for housing prices. This report includes:

- Data loading and preprocessing
- Exploratory Data Analysis (EDA)
- Regression modeling and evaluation
- Hyperparameter tuning
- Classification modeling and evaluation metrics
- Ensemble learning methods and performance assessment
- Feature importance analysis and model comparison

3 Dataset Description

The Boston Housing dataset consists of 506 observations and 14 columns (13 features + 1 target). The target variable is **MEDV**

Feature	**Description**
CRIM	Per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	Nitric oxides concentration (parts per 10 million)
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centres
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil-teacher ratio by town
B	$1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
LSTAT	% Lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

4 Data Preprocessing

The data preprocessing phase involved several steps to ensure the dataset was clean, consistent, and suitable for modeling. The key steps are outlined below:

- The dataset was loaded from a hosted URL using `pandas.read_csv()`. Additional transformations were performed using functions such as `numpy.hstack()`, `numpy.column_stack()`, and `pandas.DataFrame()` to appropriately structure the data.
- Duplicate records were checked using `drop_duplicates()`, and no duplicates were found in the dataset.
- Missing values were assessed using `info()` and `isnull()`; no missing data were detected.
- Feature scaling was applied using `StandardScaler()` to standardize the numerical features, ensuring each had a mean of zero and a standard deviation of one. This step is particularly important for algorithms sensitive to feature scale.
- The dataset was split into training and test sets using an 80:20 ratio to evaluate model performance on unseen data.

5 Exploratory Data Analysis (EDA)

5.1 Statistical Overview

Summary statistics were generated using `.describe()` and `.nunique()` value counts of the features in the Boston Housing dataset.

- **Dataset Overview**:
  - Total observations: **506**
  - Total variables: **14** (13 features + 1 target: `MEDV`)
- **Central Tendency (Mean / Median)**:
  - Median home value (`MEDV`): $**21.2k**; Mean: $**22.5k**
  - Average number of rooms per dwelling (`RM`): **6.28**
  - Average percentage of lower status population (`LSTAT`): **12.65%**
- **Dispersion and Range**:
  - `CRIM` (crime rate): Std = **8.60**, Max = **88.98**
  - `ZN` (zoned land): Std = **23.32**, Max = **100.00**
  - `TAX` (property tax): Range = **187** – **711**
  - `AGE` (pre-1940 buildings): Range = **2.9%** – **100%**
- **Skewed Distributions (Right-skewed)**:

- Features with large outliers: `CRIM`, `ZN`, `B`, `LSTAT`
- `MEDV` has a max of **50**, indicating a possible upper limit or cap

- **Categorical Feature**:
  - `CHAS` (adjacent to Charles River): Only **6.9%** of homes have river proximity
- **Location and Accessibility**:
  - `DIS` (distance to employment centers): Mean = **3.80**, Max = **12.13**
  - `RAD` (highway access index): Mean = **9.55**, Max = **24**

5.2 Visualizations

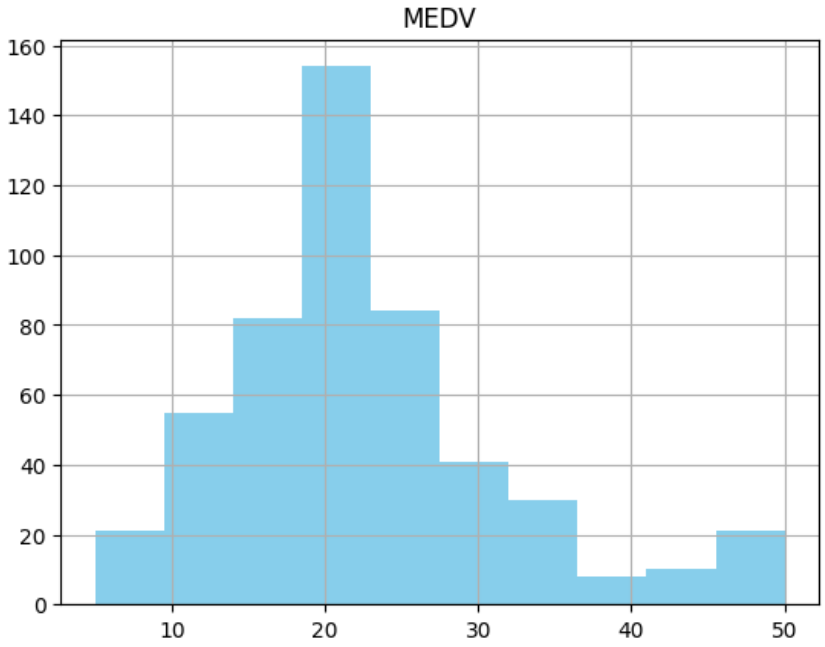


Figure 1: Distribution of target variable MEDV

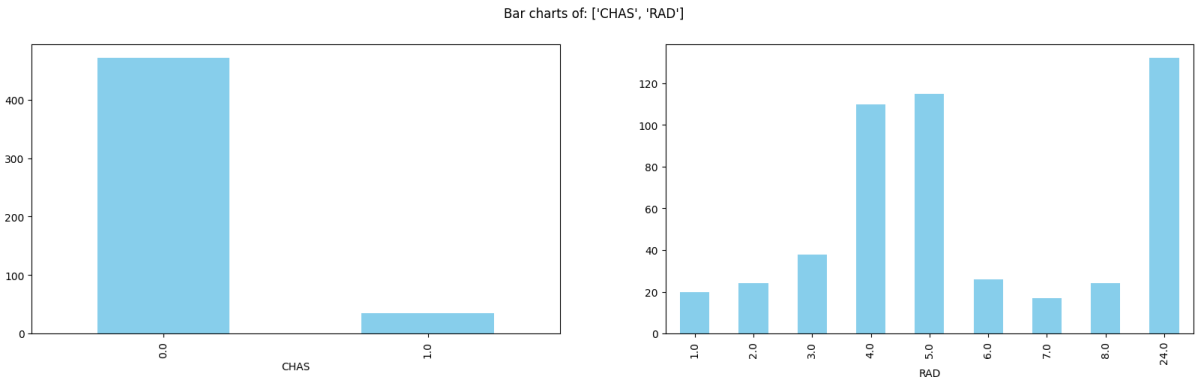


Figure 2: Distribution of categorical variables CHAS and RAD

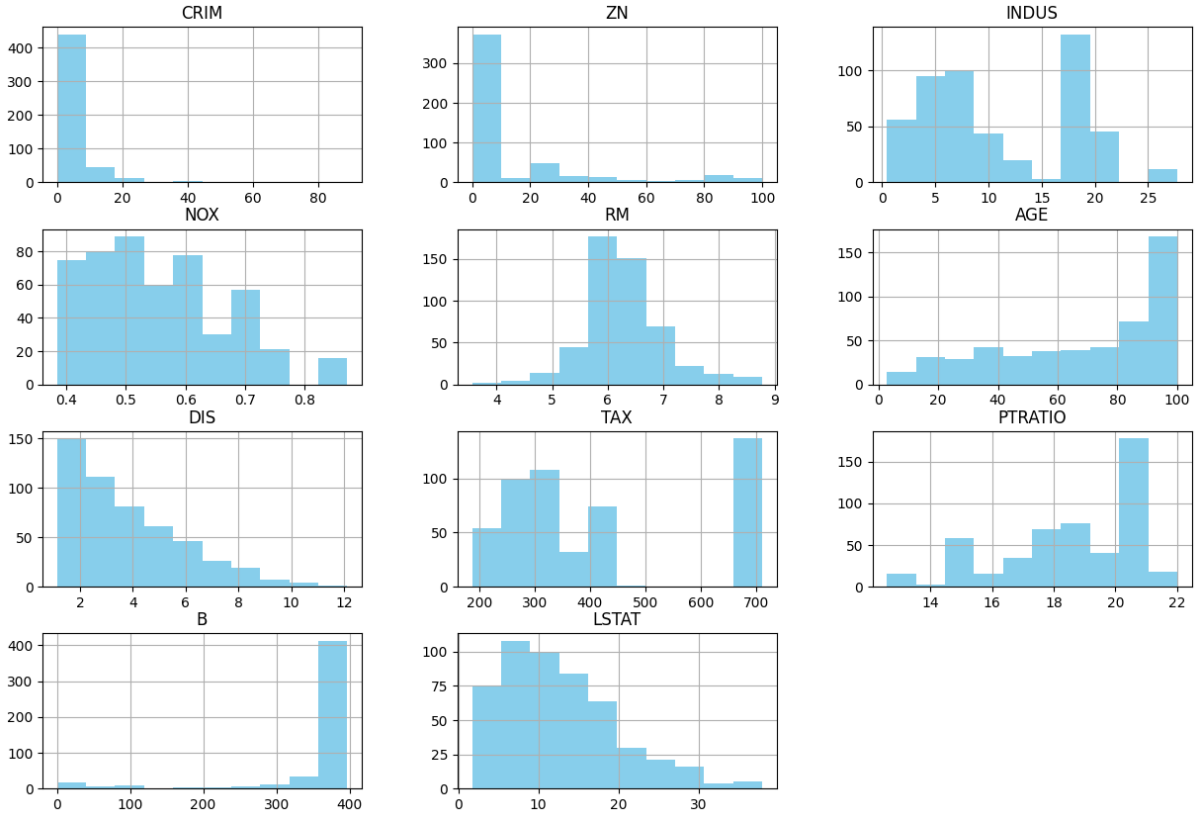


Figure 3: Distribution of feature variables CRIM, ZN, INDUS, NOX, RM, AGE, DIS, TAX, PTRATIO, B, LSTAT

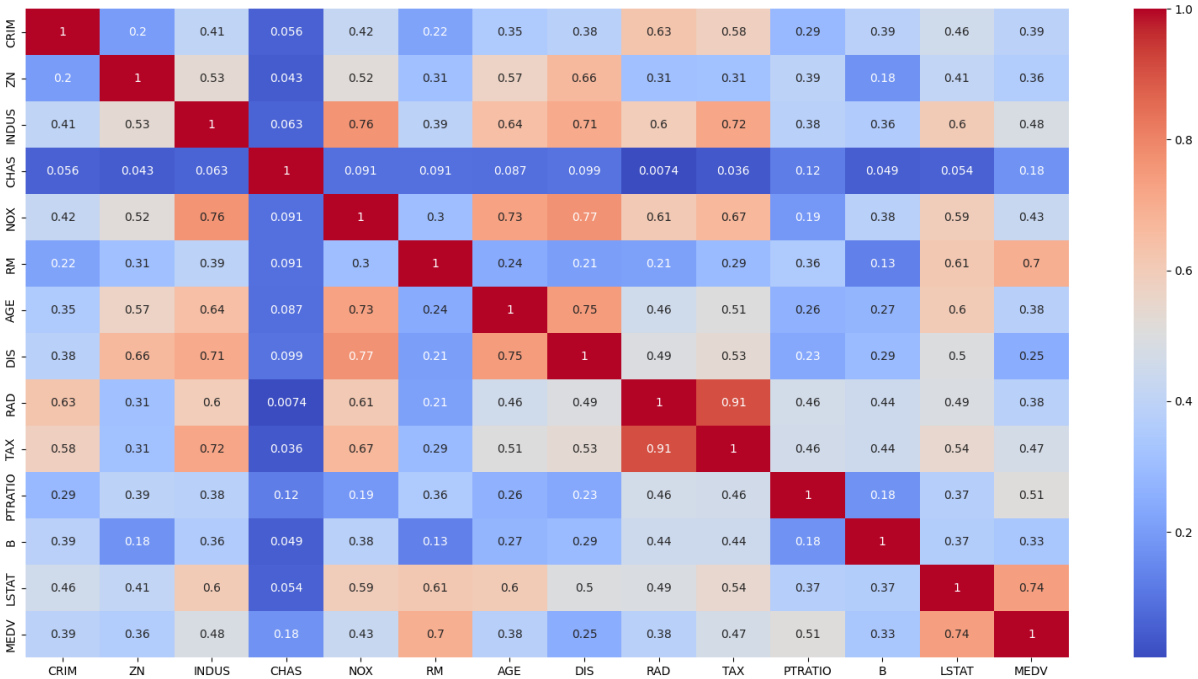


Figure 4: Correlation heatmap of features

Explanation: The correlation matrix reveals several notable relationships between the features:

- **RAD** and **TAX** show a strong positive correlation, indicating that areas with greater highway accessibility tend to have higher property taxes.
- **DIS** and **AGE** exhibit a positive correlation. This suggests that older neighborhoods tend to be located farther from central employment areas.

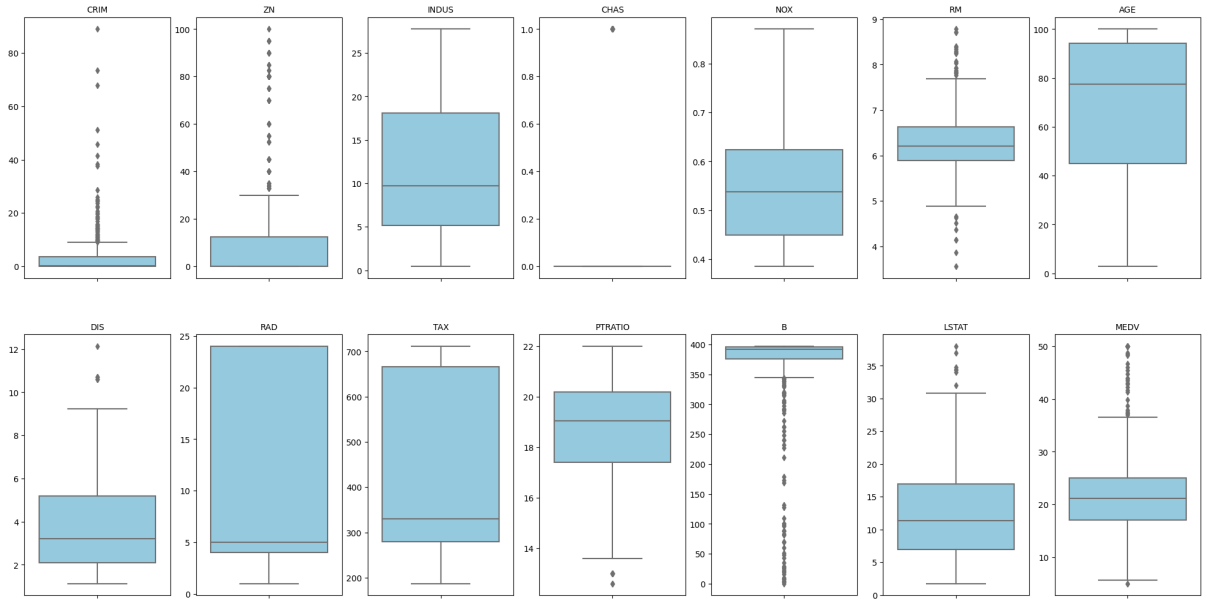


Figure 5: Boxplot Outliers of Feature and Target variable

5.3 Outliers Analysis

An outlier analysis was conducted to identify features with extreme values that may influence model performance or indicate data quality issues. The following key insights were observed:

- **CHAS** exhibited 100% outliers due to its binary nature (0 or 1), which commonly triggers statistical outlier detection techniques that are not well-suited for categorical variables.
- **CRIM**,**ZN** and **B** showed relatively high outlier rates of **13.04%**, **13.44%**, and **15.22%**, respectively, indicating significant skewness or extreme values in these features.
- **RM** and **MEDV** had moderate outlier rates of **5.93%** and **7.91%**, suggesting the presence of unusually large homes or expensive properties.
- **PTRATIO**, **LSTAT**, and **DIS** contained a small proportion of outliers under **3%**.
- Features such as **INDUS**, **NOX**, **AGE**, **RAD**, and **TAX** had **no detected outliers**, indicating relatively uniform distributions within typical value ranges.

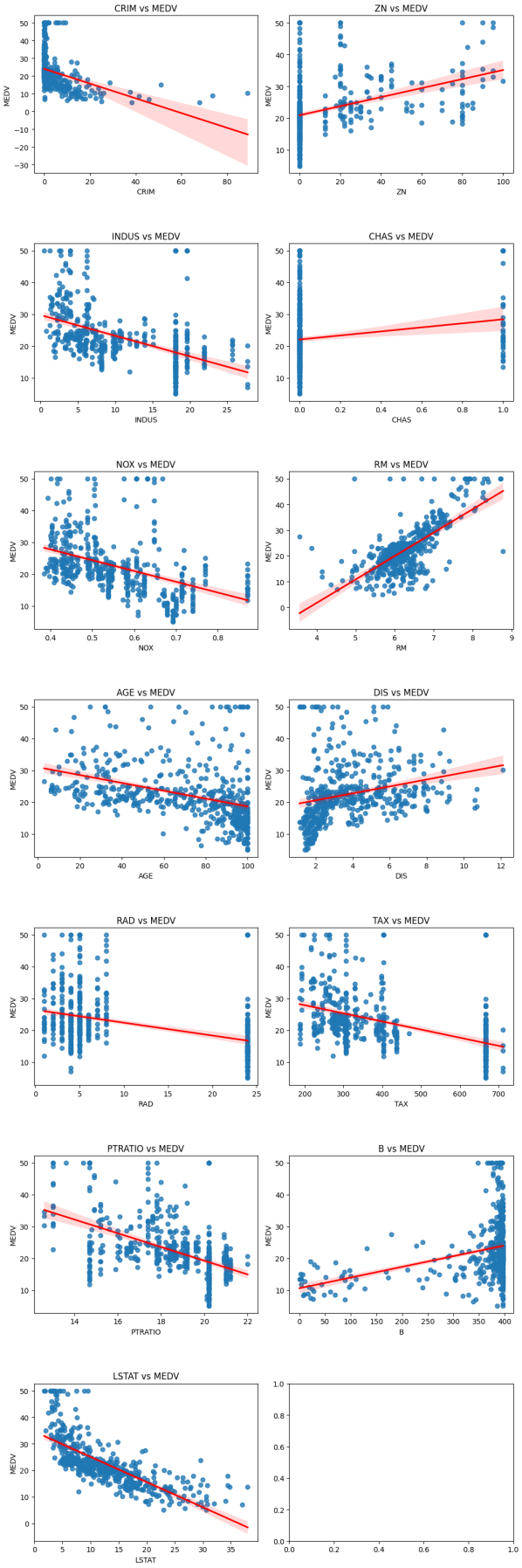


Figure 6: Scatter and Regression Plot of Target vs. Feature variables

6 Regression Modeling

6.1 Evaluation

Metrics used: Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 Score.

Model	MSE	MAE	R^2 Score
Linear Regression	24.29	3.18	0.66
Decision Tree	10.41	2.39	0.85
Random Forest	**7.91**	**2.04**	**0.89**
Support Vector Regressor	25.66	2.73	0.65

Table 2: Regression model performance on test data

Explanation:

- **Random Forest Regressor** demonstrated the best performance among all models, achieving the lowest MSE (**7.91**) and MAE (**2.04**), along with the highest R^2 score (**0.89**), indicating excellent predictive accuracy.
- **Decision Tree Regressor** also performed well, with an R^2 score of **0.85**. Although slightly less accurate than Random Forest, it showed solid results and benefited from its simplicity and interpretability.
- **Linear Regression** showed moderate performance with an R^2 of **0.66**, indicating that it could not capture the underlying complexity in the data as effectively as tree-based models.
- **Support Vector Regressor** had the highest MSE (**25.66**) and the lowest R^2 score (**0.65**), suggesting it was the least effective model in this regression task without further tuning or kernel optimization.

7 Hyperparameter Tuning

7.1 Grid Search

Grid search was performed on Random Forest, Decision Tree, and SVM to optimize parameters such as maximum depth, number of estimators, and regularization constants.

Model	Best Parameters
Random Forest	max_depth: None, min_samples_split: 2, n_estimators: 100
Support Vector Regressor	C: 100, gamma: 'auto', kernel: 'rbf'
Decision Tree	max_depth: None, min_samples_leaf: 2, min_samples_split: 10

Table 3: Best hyperparameters for each regression model using GridSearchCV

Model	MSE	MAE	R^2 Score
Random Forest	**7.91**	**2.04**	**0.89**
Support Vector Machine	11.88	2.06	0.83
Decision Tree	19.46	2.68	0.73

Table 4: Updated regression model performance on test data using GridSearchCV

Explanation:

- **Random Forest** achieved the best performance overall, with an optimized configuration of `max_depth=None`, `min_samples_split=2`, and `n_estimators=100`. It produced the lowest MSE (**7.91**) and MAE (**2.04**), and the highest R^2 score of **0.89**, indicating strong predictive power and generalization.
- **Support Vector Regressor (SVR)** performed reasonably well with parameters `C=100`, `gamma='auto'`, and `kernel='rbf'`, achieving an R^2 score of **0.83**. This suggests SVR can capture nonlinear patterns in the data effectively when properly tuned.
- **Decision Tree** demonstrated lower predictive performance compared to the other two models, despite hyperparameter tuning (`max_depth=None`, `min_samples_leaf=2`, `min_samples_split=10`). It resulted in a higher MSE (**19.46**) and a lower R^2 score of **0.73**.

7.2 Randomized Search

Model	Best Parameters
Random Forest	max_depth: 25, max_features: 'log2', min_samples_leaf: 1, min_samples_split: 2, n_estimators: 178
Support Vector Machine	C: 98.42, gamma: 'scale', kernel: 'rbf'
Decision Tree	max_depth: 6, max_features: 'auto', min_samples_leaf: 3, min_samples_split: 9

Table 5: Best hyperparameters for each regression model suing RandomizedSearchCV

Model	MSE	MAE	R^2 Score
Decision Tree	**9.68**	2.50	**0.86**
Random Forest	10.07	**1.99**	0.86
Support Vector Machine	11.87	2.06	0.83

Table 6: Updated regression model performance on test data using RandomizedSearchCV

Explanation:

- **Decision Tree** showed significant improvement compared to previous tuning efforts. With optimized parameters — `max_depth=6`, `max_features='auto'`, `min_samples_leaf=3`, and `min_samples_split=9` — it achieved an R^2 score of **0.86**, MSE of **9.68**, and MAE of **2.50**.
- **Random Forest** continued to demonstrate strong and reliable performance. The model used `max_depth=25`, `max_features='log2'`, `min_samples_leaf=1`, `min_samples_split=2`, and `n_estimators=178`, achieving an R^2 of **0.86**, MSE of **10.07**, and the lowest MAE of **1.99** among all models.
- **Support Vector Machine (SVM)** achieved an R^2 score of **0.83**, with MSE of **11.87** and MAE of **2.06**. It was tuned with `C=98.42`, `gamma='scale'`, and `kernel='rbf'`, and while it performed slightly lower than the tree-based models, it still showed robust accuracy with non-linear data.

8 Classification Modeling

8.1 Target Categorization

The continuous target variable **MEDV** was discretized into three categories to enable classification modeling using `LabelEncoder()` from `scikit-learn`.

- Low: Less than 33rd percentile
- Medium: In between 33rd–66th percentile
- High: Greater than 66th percentile

8.2 Classifiers Performance

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	**0.804**	**0.80**	**0.80**	**0.80**
Decision Tree	0.686	0.68	0.68	0.68
Random Forest	0.735	0.73	0.73	0.73
Support Vector Machine	0.725	0.73	0.72	0.73

Table 7: Classification performance (macro average metrics) of models

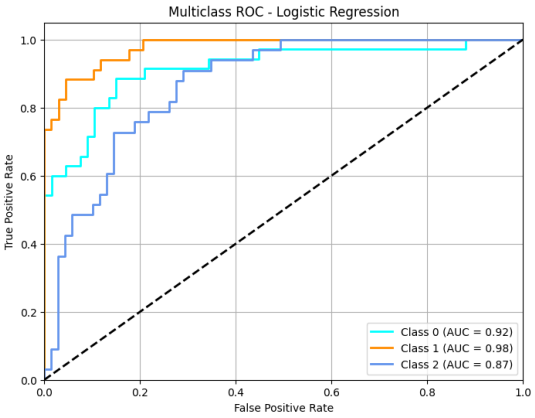
Explanation:

- **Logistic Regression** achieved the best overall performance with an accuracy of **0.804**, and macro-averaged precision, recall, and F1-score all at **0.80**. This indicates that despite its simplicity, the model effectively captures the underlying class distribution.
- **Decision Tree** performed the weakest among the models, with an accuracy of **0.686** and all macro-average metrics at **0.68**. This suggests potential overfitting or insufficient depth to capture class boundaries in this multiclass setting.

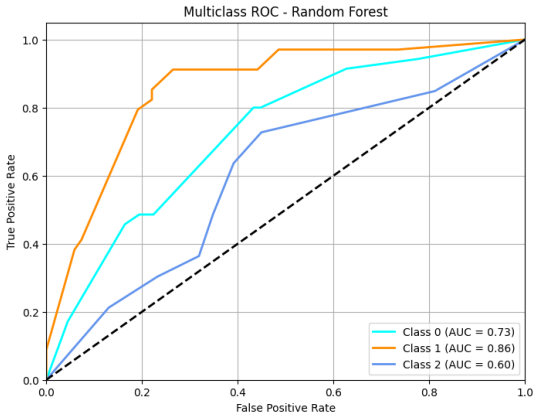
- **Random Forest** improved upon the Decision Tree by leveraging ensemble learning. It achieved an accuracy of **0.735** and consistent macro-average precision, recall, and F1-score of **0.73**, indicating good generalization and balanced class predictions.
- **Support Vector Machine (SVM)** produced competitive results, with an accuracy of **0.725** and a macro F1-score of **0.73**. It slightly underperformed compared to Random Forest and Logistic Regression but maintained robust class separation, particularly for non-linear boundaries.

ROC curves and confusion matrices were plotted for each classifier.

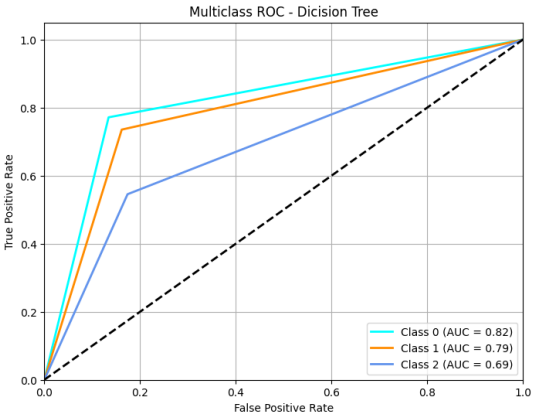
8.3 Multiclass ROC-AUC



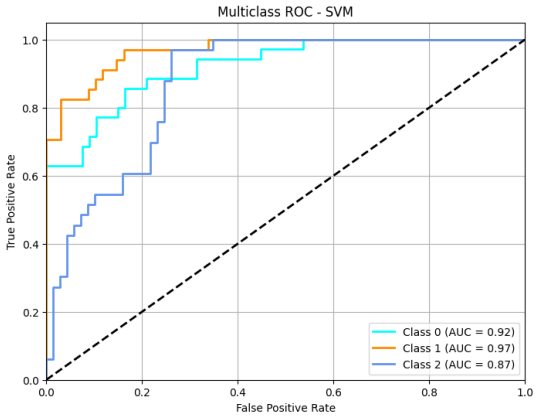
(a) Logistic Regression ROC Curve



(b) Random Forest ROC Curve



(c) Decision Tree ROC Curve



(d) SVM ROC Curve

Figure 7: Multiclass ROC Curves for Classification Models

Explanation:

- **Logistic Regression** demonstrated the highest overall AUC performance, with scores of **0.92**, **0.98**, and **0.87** for classes 0, 1, and 2 respectively. This highlights its strong ability to distinguish between all three classes.

- **Support Vector Machine (SVM)** achieved comparable AUC scores to Logistic Regression, particularly excelling in class 1 (**0.97**) and class 0 (**0.92**), indicating excellent class separation capabilities for these categories.
- **Decision Tree** showed moderate discriminatory power, with AUC scores ranging from **0.69** (class 2) to **0.82** (class 0). The lower AUC for class 2 suggests weaker performance in identifying that class correctly.
- **Random Forest**, while generally a strong classifier, had the most uneven AUC distribution in this case, with a high score of **0.86** for class 1, but a relatively poor score of **0.60** for class 2, indicating difficulty in modeling that class effectively.

9 Ensemble Learning

Model	Accuracy	Precision	Recall	F1-Score
Bagging	0.765	0.76	0.76	0.76
AdaBoost	0.735	0.76	0.73	0.74
Gradient Boosting	0.735	0.74	0.73	0.73
LightGBM	0.735	0.73	0.73	0.73
Stacking	0.735	0.74	0.73	0.74

Table 8: Classification performance (macro average metrics) of ensemble models

Explanation:

- **Bagging** achieved the highest overall performance, with an accuracy, precision, recall, and F1-score of **0.76**. Its approach of averaging predictions from multiple base learners helped reduce variance and improved model robustness.
- **AdaBoost** performed comparably in terms of accuracy (**0.735**), and stood out for its higher precision (**0.76**), indicating strong performance on correctly predicting positive class labels, though recall was slightly lower.
- **Gradient Boosting** delivered balanced metrics with an accuracy of **0.735** and F1-score of **0.73**. Its iterative error correction approach contributed to stable predictions across all classes.
- **LightGBM**, while optimized for speed and efficiency, matched other boosting models with an accuracy and F1-score of **0.735** and **0.73** respectively, demonstrating its reliability on medium-sized datasets.
- **Stacking** combined the strengths of multiple base learners and a meta-model, achieving an accuracy of **0.735** and an F1-score of **0.74**. This slight edge in F1-score suggests that stacking benefited from the diversity of its constituent models, improving overall generalization.

10 Feature Importance

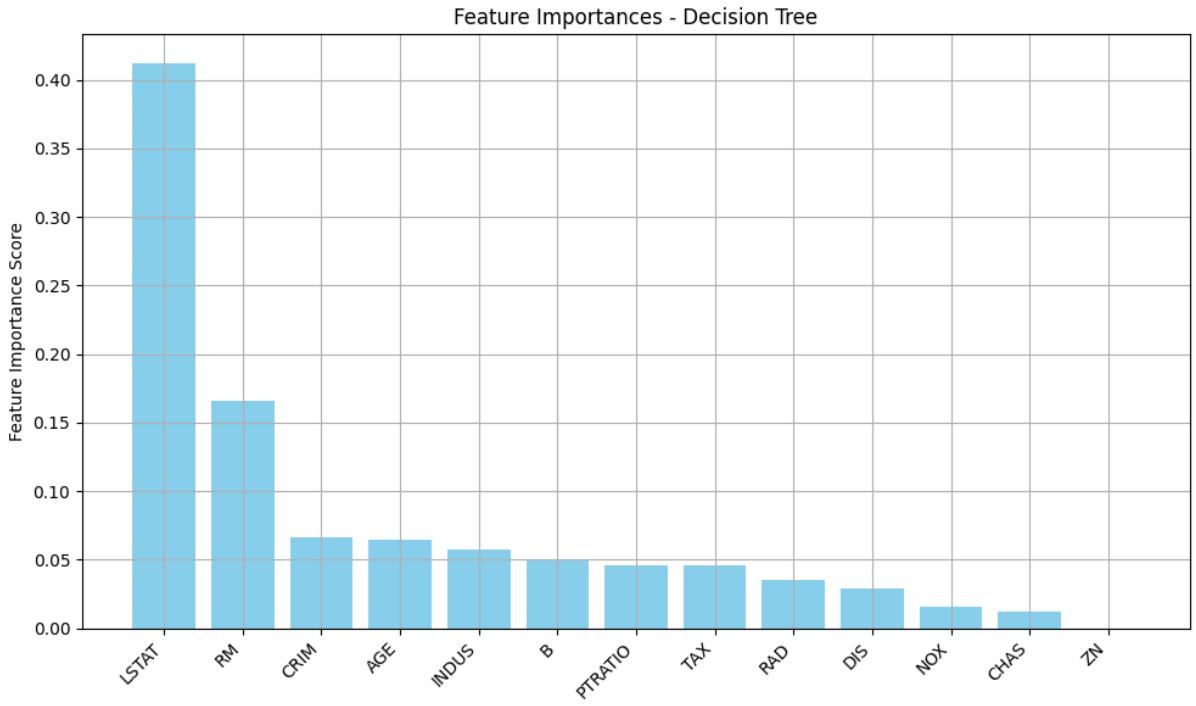


Figure 8: Feature importance plot (Random Forest)

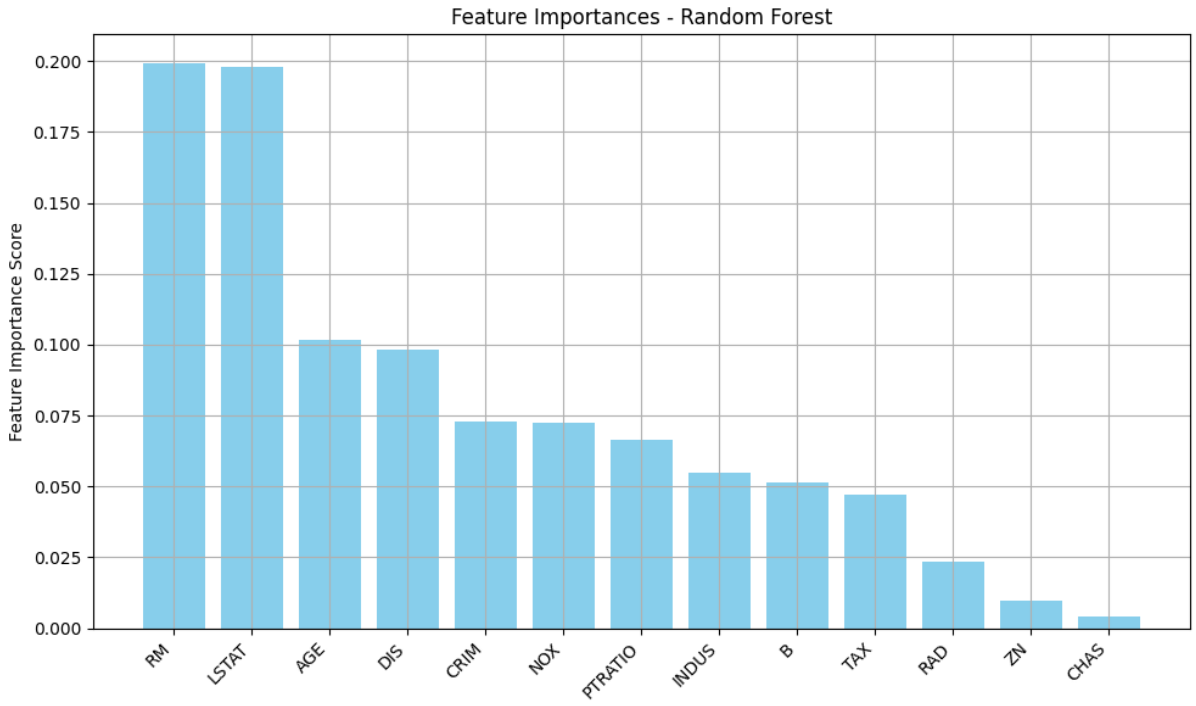


Figure 9: Feature importance plot (Random Forest)

Explanation:: Tree-based models, namely **Random Forest** and **Decision Tree**, identified **LSTAT** and **RM** as the most influential features in predicting housing prices.

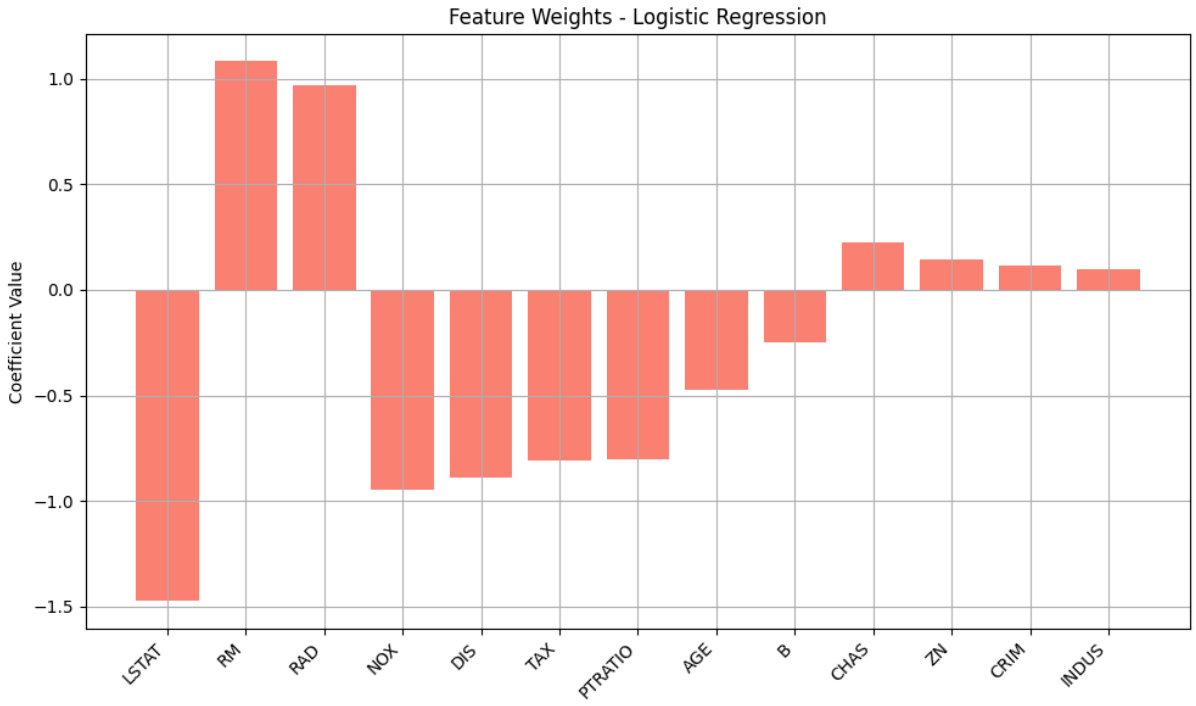


Figure 10: Feature importance plot (Logistic Regression)

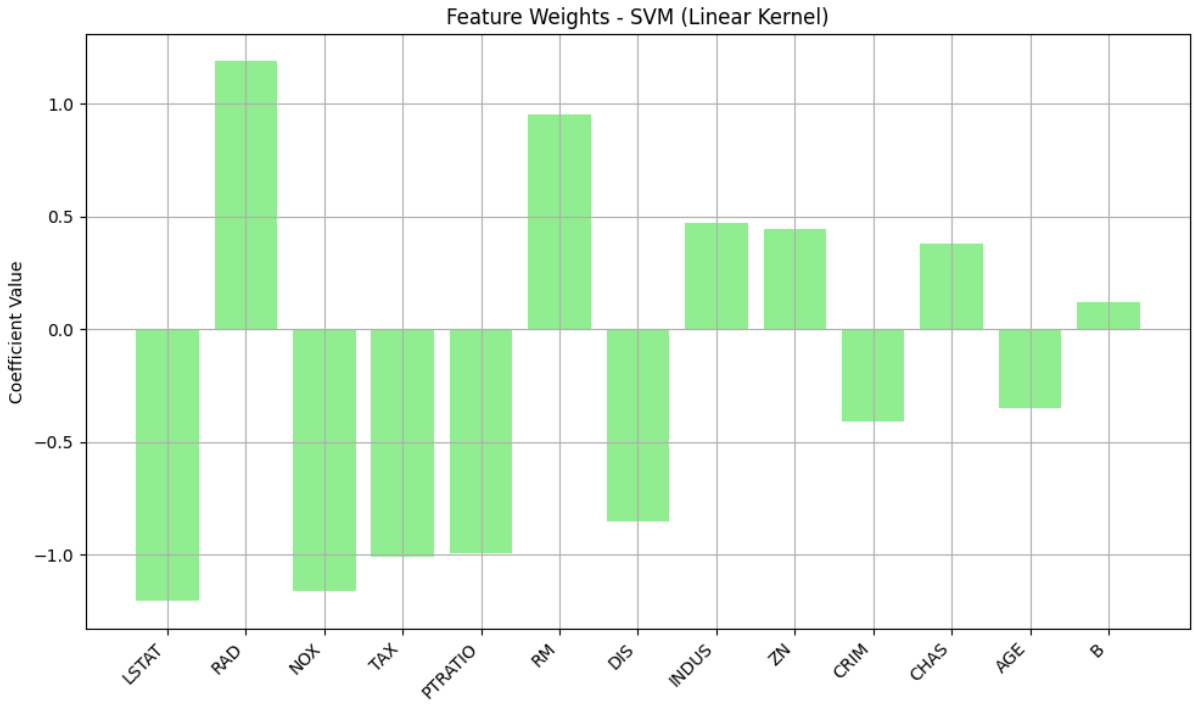


Figure 11: Feature importance plot (SVM)

Observation: Linear models, namely **Logistic Regression** and **SVM**, identified **RM**, **RAD** as the most influential features in predicting housing prices.

11 Comparative Analysis

.

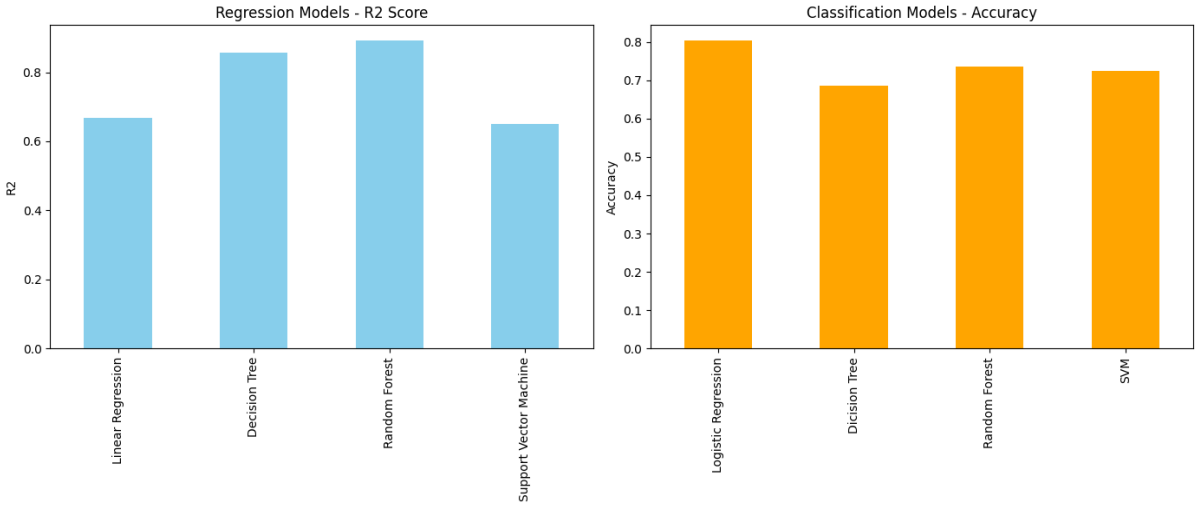


Figure 12: Comparison of regression and classification model performance

Explanation:

- **Regression:** Among all regression models, the **Random Forest Regressor** demonstrated the best performance with the lowest Mean Squared Error (MSE) of **7.91** and the highest R^2 score of **0.89**, indicating strong predictive accuracy and model fit.
- **Classification:** In the classification task, **Logistic Regression** achieved the highest accuracy of **0.804**, making it the most effective model for correctly predicting the categorical target variable.

12 Conclusion

In this assignment we have applied a range of regression and classification models to analyze the Boston Housing dataset, aiming to predict housing prices and categorize them into meaningful classes. Through rigorous preprocessing, outlier detection, feature scaling, and model tuning via both GridSearchCV and RandomizedSearchCV, we evaluated the performance of various machine learning algorithms. For the regression task, the **Random Forest Regressor** consistently outperformed other models, achieving the highest R^2 score (**0.89**) and the lowest MSE, indicating strong predictive capabilities. Feature importance analysis further highlighted **LSTAT** and **RM** as the most influential predictors of housing prices. In the classification task, **Logistic Regression** delivered the best performance with an accuracy of **0.804**, followed closely by ensemble methods such as Bagging and Random Forest. ROC-AUC analysis confirmed the robust class discrimination capability of Logistic Regression and SVM. Overall, ensemble techniques proved effective in both regression and classification settings, and hyperparameter tuning significantly enhanced model performance. These results demonstrate the value of combining robust preprocessing with model optimization to derive accurate, interpretable insights from real-world housing data.