CSP 571 Data Preparation and Analysis

Project Report

On

Boston Housing Prices

By

Name of Student	**CWID**
Mohammed Sahil	A20536680
Mohammed Abrar Ahmed	A20540340
Mohammad Anas Hussain	A20548229

Under The Supervision

of

Prof. Oleksandr Narykov

Illinois Institute of Technology

Chicago, Illinois

November 2024

CONTENT:

I. Overview:

In this project, we delved into the complexities of housing prices in Boston by working with the Boston Housing Dataset from Kaggle [1]. Our objective was to accurately predict the median house value based on the input features provided in the data. We successfully addressed missing values, mitigated skewness in feature distributions, and conducted correlation analysis to guide feature selection. Dimensionality reduction strategies were applied for better dataset understanding.

The model training phase began with a linear regression model, highlighting the need for a more sophisticated approach due to feature size concerns. A decision tree regressor initially outperformed the linear model but faced overfitting issues. We optimized model performance through grid search, feature selection, and hyperparameter tuning. Despite achieving a simpler yet effective decision tree model, dataset limitations, including biases and a small size, were recognized.

To address overfitting, we adopted a random forest regressor, with the default model proving highly accurate. The iterative process of evaluation and improvement, including cross-validation and feature analysis, led to a robust predictive model. However, we remained mindful of significant dataset limitations, underscoring the importance of critical evaluation for ethical and representational concerns in model development.

II. Data Description:

The Boston Housing Dataset, compiled in 1978 by Harrison, D., and Rubinfeld, D.L., comprises 506 records representing distinct neighborhoods in Boston. It includes 13 features and 1 target variable, a mix of numerical and categorical types, necessitating meticulous data preprocessing. Key features like 'rm' (average number of rooms) and 'lstat' (percentage of lower status population) influence Boston housing prices, with the target variable 'medv' (median value of owner-occupied homes in $1000s) being the focal point for prediction.

The Boston Housing Dataset is derived from information collected by the U.S. Census Service concerning housing in the area of Boston MA. There are 14 attributes in each case of the dataset. They are:

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940

8. DIS - weighted distances to five Boston employment centers
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per $10,000
11. PTRATIO - pupil-teacher ratio by town
12. B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT - % lower status of the population
14. MEDV - Median value of owner-occupied homes in $1000's

The dataset poses challenges for data exploration and modeling due to its different anomalies. One example we observed was that some columns had Null values. However, it has been consistently used in different research and project studies which proves its potential in the domain of machine learning.

III. Data Preprocessing:

Handling Missing Values

The initial step in data preprocessing involved a meticulous examination of missing values within the dataset. This critical analysis aimed to identify the extent of missing data and determine the most appropriate strategy for handling it. Several methods were considered, such as imputation or removal of rows/columns, based on the nature and significance of the missing values. The 'rm' column had 5 missing values, identified at indices 10, 35, 63, 96, and 135. We addressed this by replacing the missing values with the median of the 'rm' column, as demonstrated in our associated Python notebook.

Exploring Feature Distribution

We performed EDA to understand the distribution of each feature and the target variable. As seen in Figure 3 of the associated python notebook, some features, such as 'crim,' 'zn,' and 'b,' exhibited high skewness. The 'chas' column was identified as categorical.

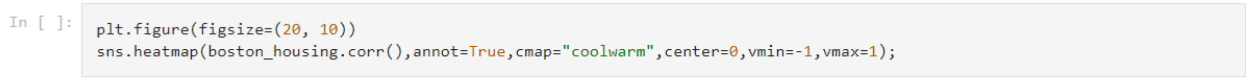
Understanding the distribution of features is essential for gaining insights into the dataset's characteristics. Visualization techniques, including histograms, were employed to analyze feature distributions. This exploration provided valuable information on the spread, central tendency, and potential outliers in each feature.

Correlation Analysis and Feature Selection

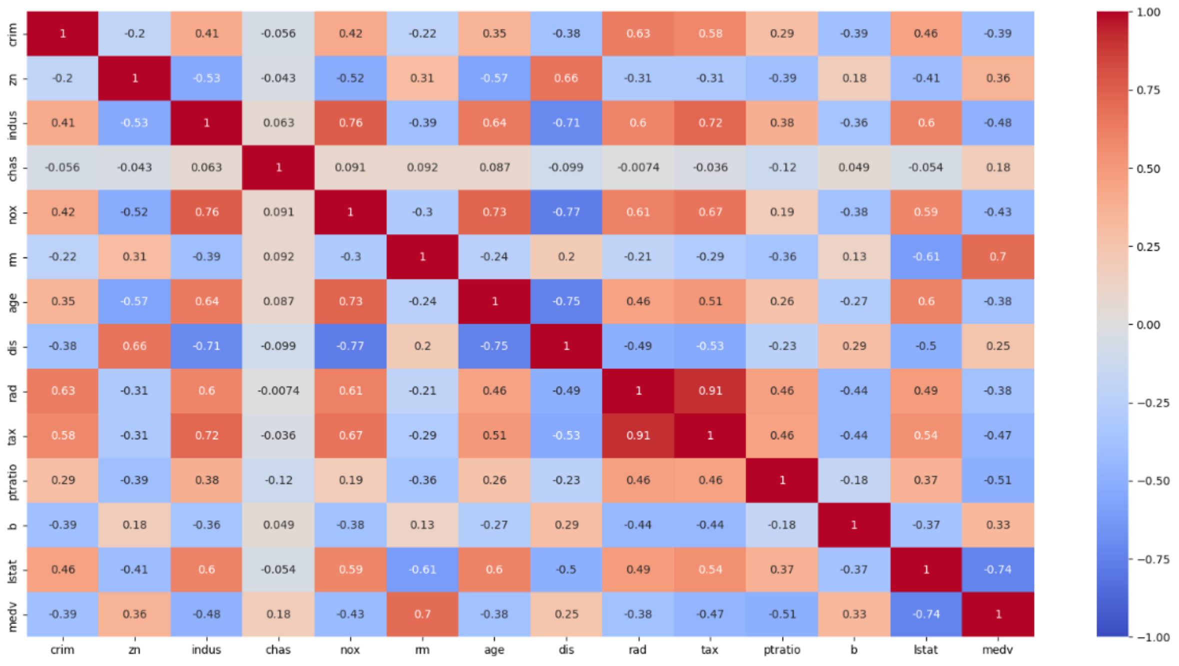
In exploring feature relationships, the 'Chas' feature showed limited correlation with others, indicating weak linear association. Due to its categorical nature and lack of strong correlations, we decided to drop 'Chas' during preprocessing. Conversely, a significant correlation between 'Tax' and 'Rad' raised concerns about multicollinearity, prompting us to drop the 'Rad' feature.

This step aimed to mitigate multicollinearity and enhance the dataset for more effective model training. The considered removal of 'Chas' and 'Rad' played a crucial role in refining the dataset for predictive modeling, emphasizing the importance of thoughtful feature correlation analysis during preprocessing.

The correlation analysis heatmap below highlights the importance of thoughtful feature consideration in developing accurate and robust predictive models. This visual representation guides informed decisions in data preprocessing, contributing to the dataset's refinement for improved model training.



```
In [ ]:  plt.figure(figsize=(20, 10))
         sns.heatmap(boston_housing.corr(),annot=True,cmap="coolwarm",center=0,vmin=-1,vmax=1);
```



'chas' doesnt seem to be correlated to anything (could possibly drop) however 'tax' and 'rad' seem to have very high correlation.

Handling Categorical Variables - 'chas'

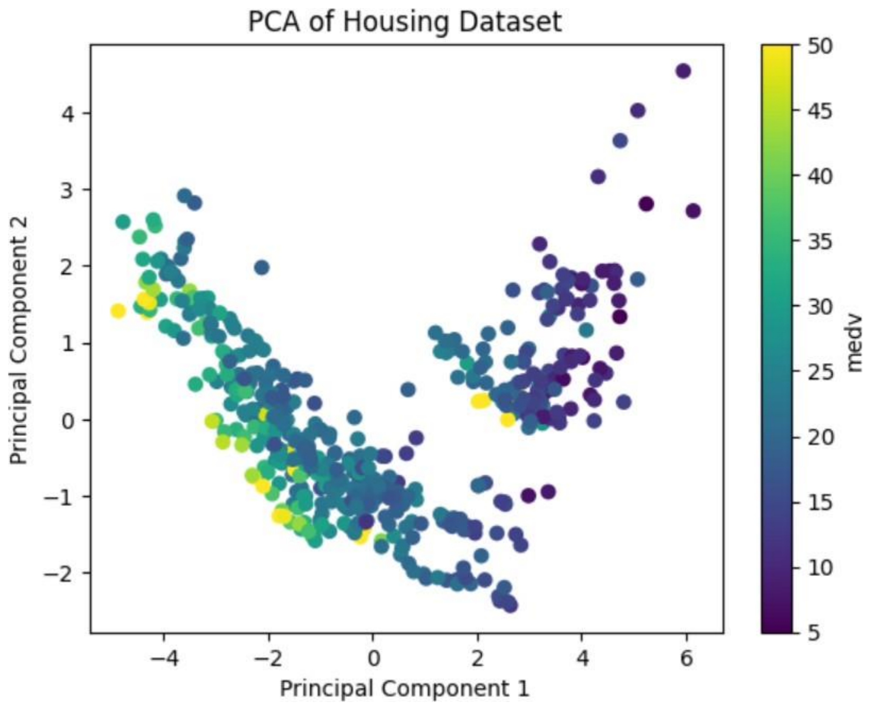
The presence of categorical variables requires special attention during preprocessing. In this project, the 'chas' column, representing a categorical variable, was scrutinized. Due to low variance, a decision was made to consider its removal from the dataset. This decision was based on the understanding that low-variance categorical features may not contribute significantly to the predictive power of the model.

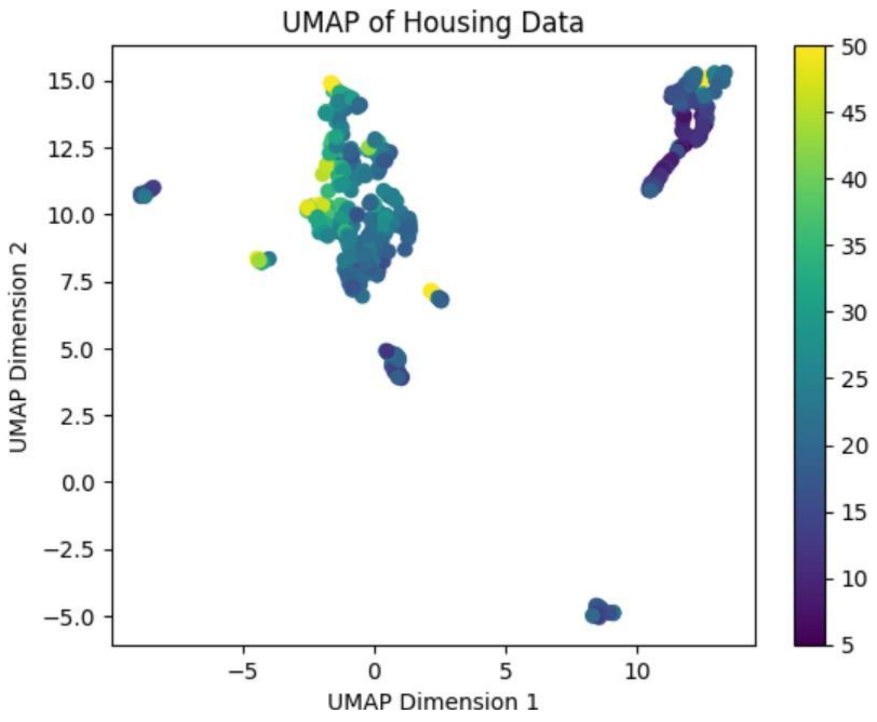
Dimensionality Reduction

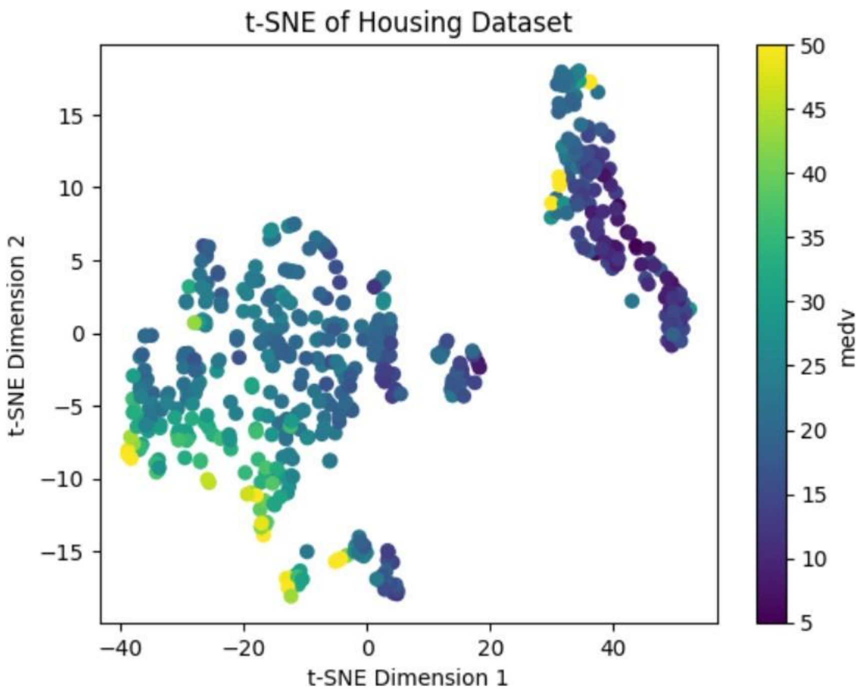
Given the high dimensionality, strategies for dimensionality reduction, including Principal Component Analysis (PCA), feature selection, Uniform Manifold Approximation and Projection (UMAP), and t-distributed Stochastic Neighbor Embedding (t-SNE), were explored [3]. The impact of reduction, evaluated with and without the 'chas' feature, aimed to enhance model efficiency and interpretability. These visualization techniques provided a comprehensive understanding of the dataset's characteristics, guiding preprocessing and improving overall data quality for model training.

Tried each dimensionality reduction technique, both including and excluding the 'chas' feature, as a visualization tool. Notably, the first dimension (on the x-axis) consistently demonstrated effective separation between lower and higher median value houses. Despite the challenges in interpretation associated with dimensionality reduction techniques, this observation suggests the presence of crucial features influencing house prices. Looking ahead, the dataset will no longer include the 'chas' feature due to its overall low variance (info value). Additionally, 'rad' will be excluded due to its high correlation with 'tax,' addressing multicollinearity concerns.

Across methodologies, it became clear that the first component (PCA) and first dimension (UMAP, t-SNE) effectively grouped more expensive houses on the left and cheaper houses on the right of the x-axis. While interpretability decreases after dimensionality reduction, there is a promising indication that certain features play a crucial role in determining house prices.







The above dimensionality reduction graphs for PCA, UMAP, t-SNE illustrate how observations with higher house prices are concentrated below x=0

Cross-Validation Strategy - Permutation of k-Fold:

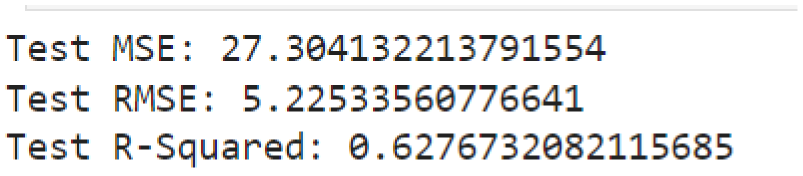
For model evaluation and training, a robust cross-validation strategy is crucial. The permutation of k-fold cross-validation was chosen for this project. The decision to use k-fold cross-validation was driven by the regression nature of the task, the absence of stratification needs, and the close-to-normal distribution of the target feature, 'medv.' This approach aimed to provide a comprehensive evaluation of the model across different subsets of the dataset.

IV. Model Training:

The model training phase commenced with the implementation of a simple linear regression model from scikit-learn, utilizing default parameters. To ensure an ample number of observations for model training, the dataset was split into a training set (80% of the data) and a testing set (20%). This larger split was deemed necessary, given the complexity introduced by the multitude of features in the dataset as well as a relatively small number of observations.

Baseline Linear Regression Model

The linear regression model served as the initial baseline for the regression task, aiming to predict housing prices in Boston. However, preliminary results indicated the need for a more robust model. The coefficient of determination score on the test set suggested that the current feature set lacked the predictive power needed for accurate housing price predictions.



```
Test MSE: 27.304132213791554
Test RMSE: 5.22533560776641
Test R-Squared: 0.6276732082115685
```

The figure illustrates the test-set results of the linear model

Feature Size Concerns

The main concern at this juncture was the size and relevance of the features. The suboptimal performance of the linear model prompted a realization that a more sophisticated model was necessary to enhance predictive accuracy.

Cross-Validation Strategy

Before delving into more advanced models, a discussion on cross-validation and the selection of an appropriate strategy was paramount. Cross-validation serves to measure model performance and efficiency across all provided data points, offering insights into how these data points influence model performance [2]. Among the plethora of cross-validation strategies available, the group opted for k-Fold cross-validation. With k=10, as seen in figure 4, the models that would then be trained would be given essentially 10 different datasets to help evaluate the performance of the models.

Simple Model Training: Linear Regression, Decision Tree Regressor

Following the initial linear regression model, a decision tree regressor was introduced. The linear regression model exhibited poor performance, prompting the exploration of more complex models. The decision tree regressor surpassed the linear model, achieving higher R-squared scores on both training and test sets [4]. This was a hopeful sign indicating that the dataset may have value after all.

The decision tree model was then also subject to a k=10 k-fold cross-validation (Figure 5). These results were shocking as the cross-validation MSE scores were worse than the linear model when it was subject to cross-validation.

Some pitfalls for the linear model were its assumption that features are scaled and distributed evenly. This was not the case in our dataset, and it makes sense as to why the linear model performed as poor as it did. The pitfall for decision tree models can be that they severely overfit training data. This is why it was apparent that during cross-validation, the majority of the folds had a worse MSE compared to the cross-validation MSE of the linear model. For this reason, an ensemble method was chosen to strike a balance between performance and complexity.

Grid Search CV, Feature Selection, Hyper-parameter Tuning

Numerous steps were taken after the initial simple linear and decision tree model training to improve model performance. The first was to implement a grid search cross validation in order to find the best hyper-parameters for the decision tree model in order to decrease the complexity. As found in the analysis, the default decision tree parameter was very deep, with a depth of 19, and wide, with 382 leaf nodes (Figure 6). It would make sense as to why on cross-validation, the model was poor since there was an obvious over-fitting to training data. Hyper-parameter selection through grid search led to a much smaller tree with depth of 7 and 30 leaf nodes (Figure 7)! This smaller model, however, did not perform on the test set as well as the default parameter tree, yet we see that a much smaller tree still only resulted in a small drop in performance (Figure 8).

Next, it was determined to analyze the feature importance of the decision tree (Figure 9), and upon the findings two features were heavily favored, “rm” and “lstat”. These were important to the model so they were kept and instead the worst feature was dropped, “ptratio”. The default decision tree with the dropped feature, performed better than the default tree with previous features on the test set (Figure 10). Now to decrease complexity another grid search was performed, and while a less complex tree was found, its performance drop was extreme and was deemed to be a model not worth the trade-off between performance and complexity (Figure 11).

The final steps were to create an ensemble model. The random forest regressor was chosen to solve the problem of over-fitting because ensemble methods increase accuracy by combining prediction power of many uncorrelated models [5].

The random forest regressor, with default parameters, was by far the best model. The cross-validation scores looked low and more balanced than for previous models (Figure 12). The test R-squared was .969 and the test MSE was 2.247 (Figure 13). There was an attempt to prune the model which resulted in a model that performed better than the decision tree with feature selection, but still not as good as the default random forest model (Figure 14).

V. Results:

The decision tree model, after feature selection (excluding 'ptratio'), showed improved performance with a higher R-squared score on the test set and reduced complexity. The grid search, however, provided a simpler model with slightly worse metrics, emphasizing the trade-off between model complexity and performance. The random forest model was the best model.

In context outside of model building and experimentation, the more important conclusion the group came to was that the dataset included a lot of bias, as well as, being too small to be able to be a good representation of house prices in Boston. Features like ‘b’, were severely skewed and also were inherently problematic as they indicated the proportion of black people in towns. There is no indication of which towns were exactly surveyed, and certain populations may live in areas where houses are priced drastically differently than other areas.

VI. Conclusion:

The project successfully explored, preprocessed, and modeled the Boston Housing dataset. The decision tree regressor, after feature selection, proved to be a promising model. Further improvements could involve fine-tuning hyperparameters or exploring other advanced models. In addition to an opportunity to explore machine learning techniques, the group also understood that data can be problematic itself. The most advanced techniques can be used to make a perfect model to represent a certain sample, but when the model is exposed to an entire population that doesn’t fit the data collected, it is appropriate to consider data collection techniques and perspectives. The report is accompanied by a publicly accessible Git repository containing the code and additional details.

Github Link: https://github.com/Sahil-337/Boston_Housing_Project

References:

[1]. https://www.kaggle.com/datasets/arunjangir245/boston-housing-dataset

[2]. https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right

[3]. https://aurigait.com/blog/blog-easy-explanation-of-dimensionality-reduction-and-techniques/

[4]. https://medium.com/@vk.viswa/unveiling-decision-tree-regression-exploring-its-principles-implementation-beb882d756c6

[5]. https://www.ibm.com/topics/random-forest