Boston House-Price Report

Jan Vincent G. Elleazar

Department of Computer Science, College of Information and Computing Sciences
University of Santo Tomas

janvincent.elleazar.cics@ust.edu.ph

I. Introduction

The Boston house-price problem is one of the most significant machine learning problems in the field of computer science. This problem tackles different ways to address data analysis and machine learning modelling as it is composed of a linear dataset mostly filled with float values and some integer values which is ideal for regression models. The dataset is composed of input features and a target variable.

There are 13 input features and 1 target variable which are the following:

Input Features

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: Calculated as 1000(Bk−0.63)21000(Bk - 0.63)^21000(Bk−0.63)2, where BkBkBk is the proportion of Black residents by town.
13. LSTAT: Percentage of lower-status population.

Target Variable

1. MEDV: Median value of owner-occupied homes in $1000's.

The objective of the Boston house-price problem is to train a machine learning regression model to predict the median value of owner-occupied homes based on the input features.

II. Methodology

This section discusses the detailed approach of using data analysis techniques and machine learning modelling techniques to address the problem. This section is segmented into three (3) subsections; Exploratory Data Analysis, Data Preprocessing, and Model Implementation.

A. Exploratory Data Analysis

Exploratory Data Analysis (EDA) was used to inspect the dataset for gathering insights on the structure of the data and the correlation between the input features and the target variable. Domain knowledge based on personal experience was also used as part of the EDA. The following steps explain the flow of EDA.

1) *Data Inspection:* The dataset was first loaded into a variable. The head and tails of the dataset were viewed and noticed that the CHAR values are 0. The initial insight for this information was that CHAR values might not have any correlation to MEDV. The whole dataset was then viewed to check any patterns on the values. While checking the values, background knowledge on the video game called Cities: Skylines was used as the game has a land value feature that is dependent on how the city was built by a player. With that at hand, TAX, RAD, INDUS, and CRIM are the input features that could have correlation to MEDV. The dataset was sorted based from lowest to highest MEDV. It showed that TAX, CRIM, and INDUS

have potential relation to MEDV due to visible patterns while RAD was not considered since its high values are scattered in low and high MEDV. With further checking on other input features, LSTAT, DIS, PTRATIO, RN, and ZN also have potential due to visible correlation patterns while AGE, B, NOX, and CHAS were not considered due to scattered values in high and low MEDV.

2) *Data Visualization*: The dataset was described to analyze the basic statistics of each attribute, especially the standard deviation.

TABLE I
STANDARD DEVIATION OF FEATURES AND TARGET VARIABLE

Standard Deviation of features and target variable	
Features	**Standard Deviation**
CRIM	8.601545
ZN	23.322453
INDUS	6.860353
CHAS	0.253994
NOX	0.115878
RM	0.702617
AGE	28.148861
DIS	2.105710
RAD	8.707259
TAX	168.537116
PTRATIO	2.164946
B	91.294864
LSTAT	7.141062
MEDV	9.197104

It showed that CRIM, INDUS, CHAS, NOX, RM, DIS, RAD, PTRATIO, LSTAT have standard deviations less than 10. This means that the data of the said features are clustered around the mean which is beneficial for feature scaling thus would be good for the accuracy of the machine learning models. CRIM, INDUS, RM, DIS, PTRATIO, and LSTAT were chosen as initial features for further analysis but CHAS, NOX, and RAD were not included due to the previous analysis during data inspection.

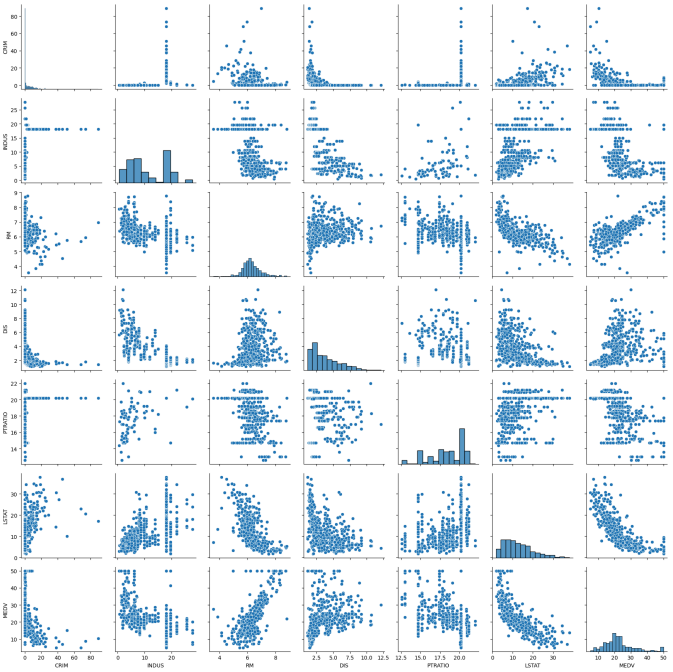

Fig. 1 Pair plot of the initial features CRIM, INDUS, RM, DIS, PTRATIO, LSTAT, and the target variable MEDV

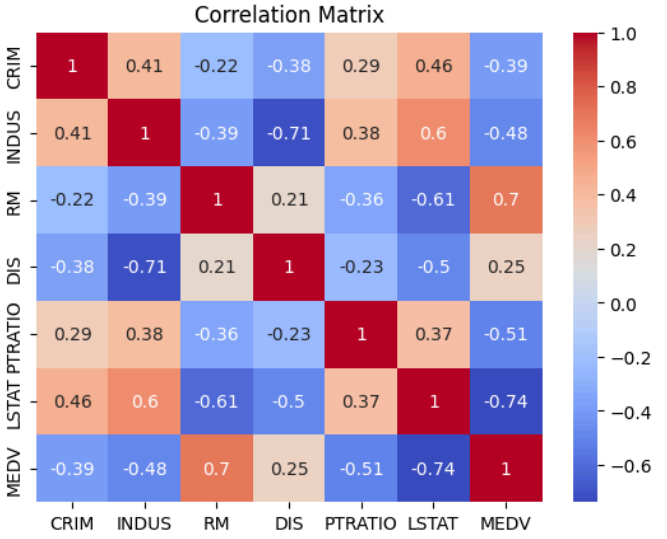

Fig. 2 Correlation Matrix of the initial features CRIM, INDUS, RM, DIS, PTRATIO, LSTAT, and the target variable MEDV

Pair Plot and Correlation Matrix was used next to visualize the distribution of data and its correlation between the initial features and the target variable. It showed that RM and LSTAT have a visible pattern of plotted points in the Pair Plot in relation

to MEDV. RM showed a positive pattern while LSTAT showed a negative pattern. On the other hand of the Correlation Matrix, RM showed strong positive correlation (0.7), LSTAT showed strong negative correlation (-0.74), and PTRATIO showed moderate negative correlation (-0.51) in relation to MEDV. INDUS showed weak but almost moderate negative correlation (-0.48).

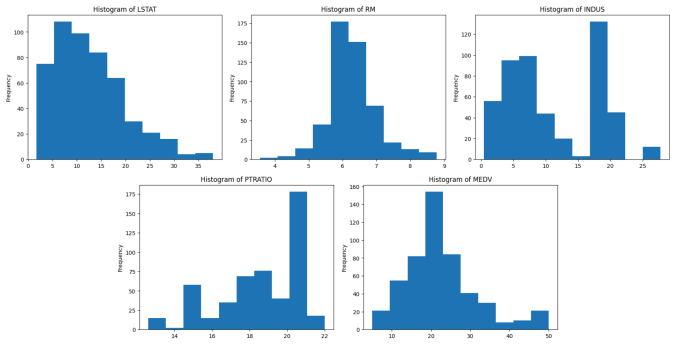

Fig. 3 Histogram of LSTAT, RM, INDUS, PTRATIO, and MEDV

Histogram was then used to visualize the data distribution of each initial feature and the target variable. INDUS have negative skewness but showed potential outliers to the right. LSTAT has positive skewness with potential outliers to the left. RM has zero skewness with potential outliers on both sides and MEDV has almost zero skewness with potential outliers to the right.

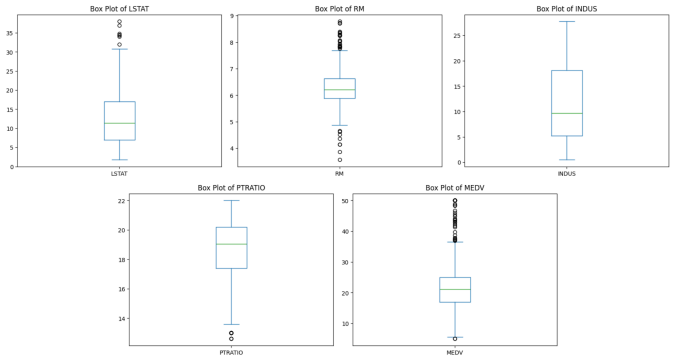

Fig. 4 Box Plot of LSTAT, RM, INDUS, PTRATIO, and MEDV

Box Plot was then used to visualize the potential outliers of each initial feature and the target variable. INDUS showed no outliers. LSTAT showed potential outliers above the maximum value. PTRATIO showed potential outliers below the minimum value. RM showed potential outliers above and below the maximum and minimum values respectively. MEDV showed potential outliers above the maximum value and one below the minimum value

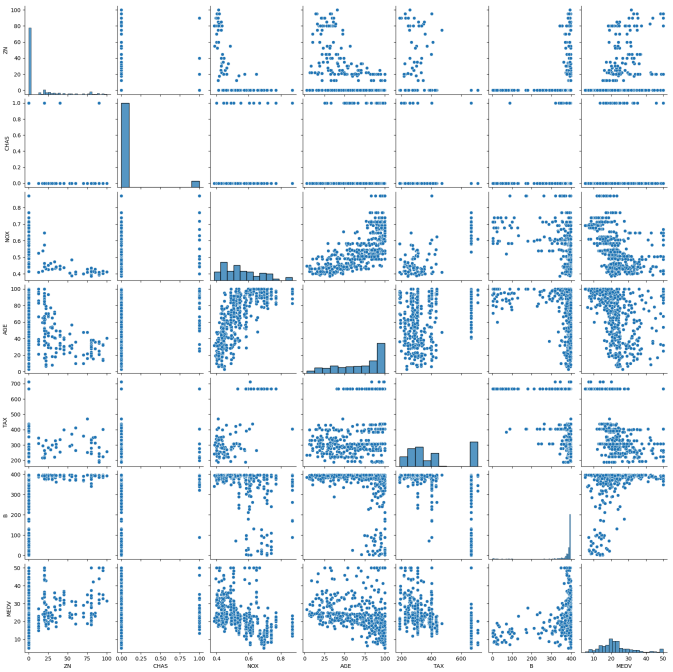

Fig. 5 Pair Plot of other features ZN, CHAS, NOX, AGE, TAX, B, and target variable MEDV.

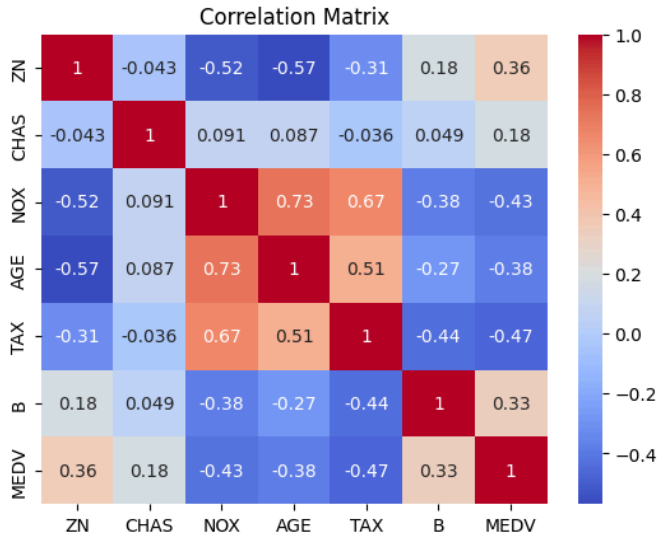

Fig. 6 Correlation Matrix of other features ZN, CHAS, NOX, AGE, TAX, B, and target variable MEDV.

In order to maximize the analysis of the feature for feature selection, the other features (ZN, CHAS, NOX, AGE, TAX, B) were also used for analysis using Pair Plot and Correlation Matrix. It showed that TAX has the only value with nearest to moderate negative correlation (-0.47).

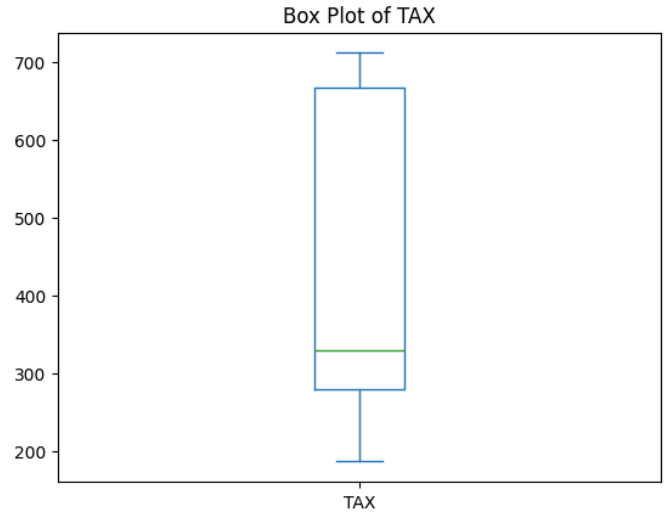

Fig. 7 Box Plot of tax

It also showed in Box Plot that TAX showed no outliers. Therefore, RM, LSTAT, PTRATIO, TAX, and INDUS were chosen for feature selection based on data inspection, applied background knowledge, standard deviation, plot box pattern recognition, and data correlation.

B. Data Preprocessing

Data Preprocessing was used prior to model implementation to make sure that the dataset to be used for the training of the models is precise and correct for the accuracy of the models.

The first process was to check for null values and duplicates in the dataset. It showed that there are no null values and duplicates. Encoding was also considered to be implemented in handling categorical data. After checking for the type of data in each feature however, all datatypes of the dataset are floats and integers so there is no need for encoding. Next was feature selection based on the EDA.

RM, LSTAT, PTRATIO, TAX, and INDUS were initialized into the X variable while MEDV was initialized into the y variable. The selected features and the target variable were then splitted into training and testing sets with training size of 80%, testing size of 20%, and a random state of 42. The 80:20 ratio was chosen as a standard ratio for the models to have ample training data given the size of the dataset.

To handle outliers, the training set was winsorized. Winsorization is the process of replacing the extreme values of each attribute in a dataset in order to limit the effects of the outliers on the predicted results of the models. The top and bottom extreme values were replaced by the values at the top 99th percentile and the bottom 1st percentile respectively. Therefore, the training set was winsorized by 2% so the training data will not be trimmed down by a large amount. On the other hand, the testing set was not winsorized because it should not be modified as it will be beneficial to compare the predicted winsorized data to the actual data for evaluation of the models whether the models became robust in handling outliers.

Lastly, the training set and testing set was normalized. Fit transform was applied to the training set while transform was applied to the testing set. This was done to prevent data leakage when training the model. It is also the same measure taken in winsorization for accurate model evaluation.

C. Model Implementation

Four (4) regression models were implemented and evaluated. These models were Linear Regression, Ridge Regression, Lasso Regression, and Elastic Net.

Linear Regression is used to find the relationship between variables. Ridge Regression is used for minimizing the impact of less important variables by using an L2 penalty, which adds the sum of the squared coefficient values multiplied by a tuning parameter. Lasso Regression is used for setting some coefficients to zero which is useful for feature selection by using an L1 penalty, which sums the absolute values of the coefficients multiplied by lambda. Elastic Net combines both the penalties of Lasso and Ridge regression methods to minimize overfitting.

In Linear Regression, no hyperparameters were tuned and the model was trained as it is. In Ridge Regression and Lasso Regression, the strength of regularization was tuned to handle overfitting. In Elastic Net, both strength of regularization and the

balance between L1 and L2 were tuned. The said balance was tuned to determine how Elastic Net would behave more effectively in feature selection or coefficient shrinkage.

The other parameters such as fit intercept, tolerance, solver, and selection were not tuned as these parameters have little to no impact or improvement to the regression models.

III. Experiments

As stated above, only the parameters of Ridge, Lasso and Elastic Net Regression models were tuned since these models have alpha parameters and L1 ratio for Elastic Net. Refer to the Jupyter Notebook file for the code blocks of each experiment.

In implementing the Linear Regression model, the testing set and the training set was used for the prediction of the model. This was done to determine if the model has overfitting issues. The evaluation metrics were then implemented and then printed the result. Scatter plot was used to visualize the predicted data versus the actual data.

In implementing the Ridge Regression model, the alpha hyperparameter was tuned to determine the optimal regularization strength of the model. The goal was to iterate through each given alpha value and it will return the alpha value of the lowest Mean Squared Error (MSE) on the testing set. The program will also return the lowest MSE on the testing and training set, the highest R-Squared (R2) on the testing set, and the lowest Root Mean Squared Error (RMSE) on the testing set. The alpha values that were set were 0.001, 0.01, 0.1, 1, 10, and 100. The basis of the said values is by using logarithmic scale to cover a broad range of regularization strength. Scatter plot was also utilized to visualize the distribution of predicted and actual data on each iteration.

In implementing the Lasso Regression model, the alpha hyperparameter was tuned to determine the optimal regularization strength of the model. It will also iterate through each given alpha value and it will return the alpha value of the lowest MSE on the testing set. The program will also return the lowest MSE on the testing and training set, the highest R2 on the testing set, and the lowest RMSE on the testing set. The alpha values that were set were 0.001, 0.01, 0.1, 1, 10, and 100.

In implementing the Elastic Net Regression model. The alpha and L1 ratio hyperparameters were tuned to determine the optimal regularization strength and to control the balance between L1 and L2 regularization strength. It will iterate through each given alpha and L1 ratio values and it will return the alpha value and the L1 ratio value of the lowest MSE on the testing set. It will also return the lowest MSE on the testing and training set, the highest R2 on the testing set, and the lowest Root Mean Squared Error RMSE on the testing set. The alpha values that were set were 0.001, 0.01, 0.1, 1, 10, and 100 while the L1 ratio values that were set were 0.1, 0.3, 0.5, 0.7, 0.9. The basis of the L1 ratio values were to provide an ample range from 0 (Ridge-like behavior) to 1 (Lasso-like behavior) and to experiment with the balance between L1 and L2 regularization.

IV. Results and Analysis

TABLE 2
Best hyperparameter values of Ridge, Lasso, and Elastic Net Regression models

Model	Alpha	L1 Ratio
Ridge Regression	100	-
Lasso Regression	1	-
Elastic Net Regression	1	0.9

TABLE 3
Best evaluation metrics of Linear, RIdge, Lasso, and Elastic Net Regression models

Model	Best MSE on testing set	Best MSE on training set	Best R2	Best RMSE
Linear	28.830618187903 07	25.869030866332416	0.6068576180656966	5.369415069437552
Ridge	26.966059002403597	25.86903086655789	0.6322832691796336	5.1928854216517815
Lasso	27.517127497870945	25.869040767921682	0.6247687448810222	5.24567702950448
Elastic	27.453997881209208	25.869042066687975	0.6256295980095 69	5.239656275101375

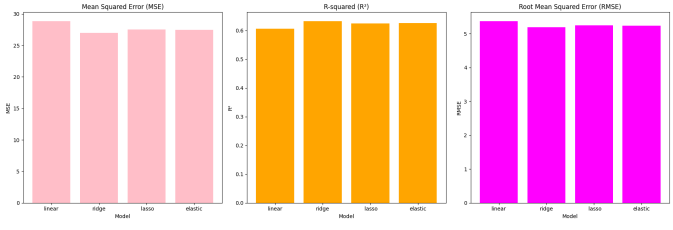

Fig. 8 Bar graph of Mean Squared Error (MSE), R-Squared (R2), and Root Mean Squared Error (RMSE) on each regression model

The best evaluation metrics of each regression model was measured on the corresponding best hyperparameter values in Ridge, Lasso and Elastic Regression models.

Ridge Regression performs very well with regularization strength of 100, which shows how strong the model should be in order to perform well. With regularization strength of 100, the model is less likely to overfit the data as seen on Table 3 where there is only a small difference between the MSE on testing set versus the MSE on training set. However, this also could show that the influence of many features is greatly reduced as Ridge Regression always keeps all features in the data which is valuable in Feature Importance. But overall, Ridge Regression performs the best among all the models in which it explains 63% of the variance in house prices.

The second best performing model is Elastic Net Regression with regularization strength of 1 and an L1 Ratio of 0.9. This means that it has strong regularization and is heavily dependent on the regularization strength of Lasso Regression. With the strong regularization strength, there is a possibility of complexity in the data and the data also might have irrelevant features. Overall, Elastic Net explains 62.4% of the variance in house prices.

The next better performing model is Lasso Regression with a regularization strength of 1. This means that the model performs with a moderate regularization strength. It also means that it performs feature selection, wherein it sets some coefficients to zero. This could also mean that some features are irrelevant and the regularization is assisting to prevent overfitting. Overall, Lasso Regression explains 62.5% of the variance in house prices

The least performing model is Linear Regression. The model has the highest Mean Squared Error compared to the other regression models. When Linear Regression performs the least among the other regression models, it means that the data optimally needs regularization where Ridge, Lasso, and Elastic Net are necessarily used for. Overall, Linear Regression explains 60% of the variance in the house prices.

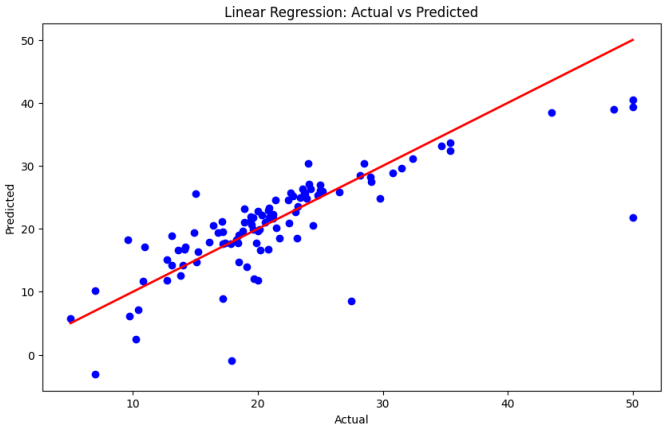

Fig. 9 Scatter Plot of actual versus predicted data of Linear Regression model

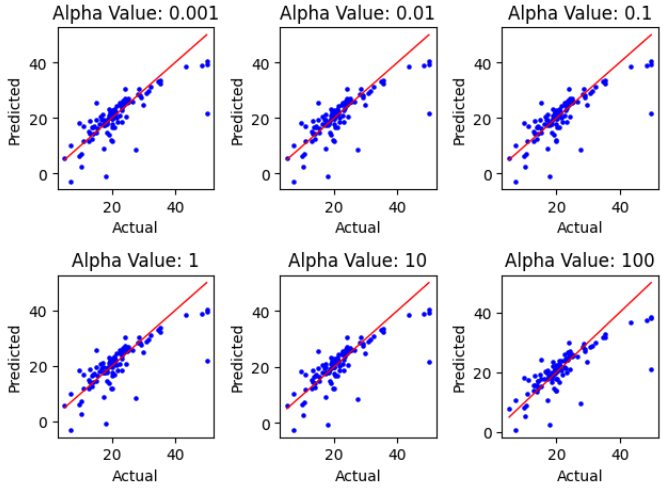

Fig. 10 Scatter Plot of actual versus predicted data of Ridge Regression model

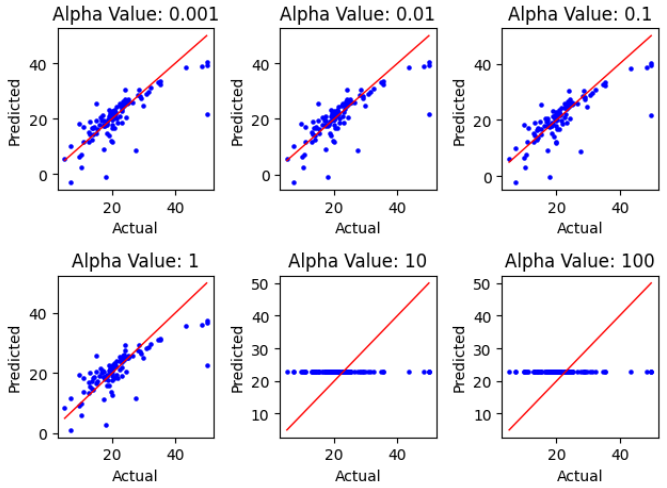

Fig. 11 Scatter Plot of actual versus predicted data of Lasso Regression model

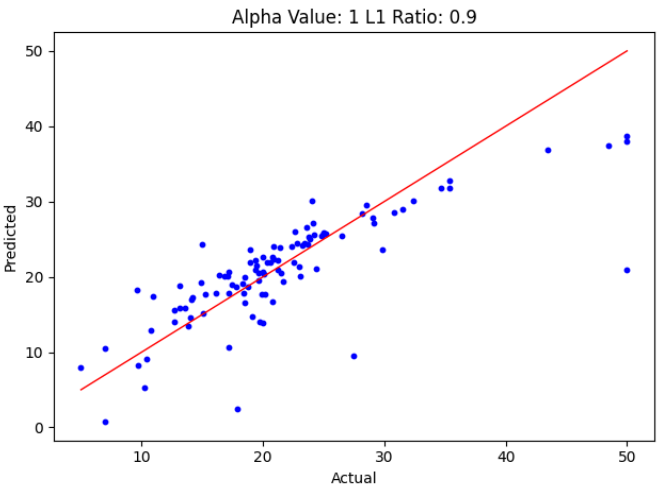

Fig. 12 Scatter Plot of actual versus predicted data of Elastic Net model

Figures 9, 10, 11, and 2 show the scatter plot of each regression model. The scattered dots in each figure are identical but there is a significant change in Figure 11 wherein the dots form a straight line on the x axis of the plane. This simply explains that the Lasso Regression model is over-regularized when the regularization strength is over 10. The features are being ignored by the model thus the model becoming over-simplified.

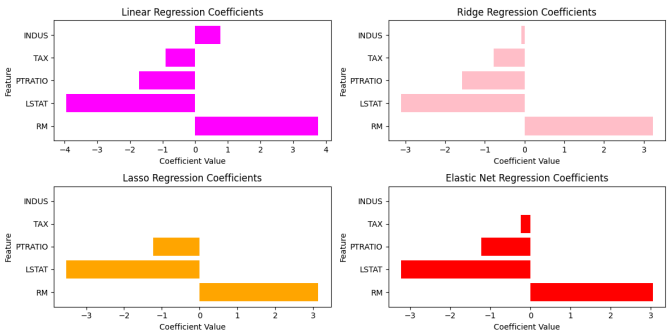

Fig. 13 Horizontal bar graph showing the coefficients of each regression model on their corresponding features

Figure 13 shows how each selected feature relates to the target variable in each regression model using its coefficients. RM has strong positive correlation with MEDV on all regression models while LSTAT has strong negative correlation with MEDV on all regression models as well. PTRATIO has moderate negative correlation with MEDV on all regression models. Tax has weak negative correlation to MEDV on Linear, Ridge, and Elastic Net Regression models while no linear correlation on

Lasso Regression. Lastly, INDUS has mixed correlation to MEDV on all regression models; no linear correlation on Elastic Net and Lasso Regression, weak positive correlation on Linear Regression, and weak negative correlation on Ridge Regression. This supports the finding that the Ridge, Lasso, and Elastic Net Regressions are over-regularized in which the models achieved their best performance.

V. Conclusions

Among all the regression models trained, Ridge Regression performed the best with 63% explained variance in house prices in Boston. This means that some selected features for the model have high correlation to the target variable as shown in Figure 13. Overfitting is also not an issue in this model since the difference between the best MSE of the testing set versus the best MSE of the training set is small as seen on Table 3. However, the regularization strength where the model performed the best is 100 which suggests that there is a risk of underfitting since the model shrinks the coefficients significantly.

Lasso and Elastic Net Regression sets the coefficient of INDUS to zero which means that the feature is not relevant. Elastic Net also performed best in strong regularization strength while Lasso performed the best in moderate regularization strength.

Linear Regression performs the least well among the other regression models. This suggests that the selected features require strong regularization for the models to perform well, thus the findings of strong regularization strength of Ridge and Elastic Net.

There is still a large amount of improvement that needs to be done on Exploratory Data Analysis since the findings suggest that the models performed the best with strong regularization strength suggesting that there are unnecessary features included such as TAX and INDUS as seen on Figure 13. The evaluation metrics of the regression models could improve with changes on feature selection.

Overall, this activity emphasizes the significance of Exploratory Data Analysis and hyperparameter tuning of the models to deliver the best performance and results of the Regression Models.

VI. References

[1] GeeksforGeeks. (2021, May 30). *Winsorization*. GeeksforGeeks. https://www.geeksforgeeks.org/winsorization/

[2] GeeksforGeeks. (2024, June 5). *What is Elasticnet in Sklearn?* GeeksforGeeks. https://www.geeksforgeeks.org/what-is-elasticnet-in-sklearn/

[3] Jain, A. (2025, January 6). *Ridge and lasso regression in Python*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/

[4] Shreayan98c. (2019, March 5). *Boston house price prediction*. Kaggle. https://www.kaggle.com/code/shreayan98c/boston-house-price-prediction

[5] Verma, S., & Chaturvedi, T. (2024, July 17). *LASSO Regression: A Comprehensive Guide by Pickl.AI*. Pickl.AI. https://www.pickl.ai/blog/lasso-regression/

[6] *W3Schools.com*. (n.d.). https://www.w3schools.com/python/python_ml_linear_regression.asp

[7] *What is Overfitting? - Overfitting in Machine Learning Explained - AWS*. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/what-is/overfitting/#:~:text=The%20model%20trains%20for%20too,single%20sample%20set%20of%20data.&The%20model%20complexity%20is%20high,noise%20within%20the%20training%20data.