LUT School of Business and Management
A220A0752 Analytics for Business
Prof. Mostafa Goudarzi

Analytics for Business
Group Assignment Group
Ich liebe richtige Entscheidungen

What are the most critical factors influencing house prices?

19.05.2024
Saleh Butt
Sunjida Haque
Nicolas Pauli
Arttu Rytkönen

Table of contents

1 Introduction, Background, and Problem Definition

By taking a look at the development of property prices over the last few decades, it becomes clear that they have risen continuously in almost every metropolitan area (E. Glaeser, 2005).This development not only affects the social fabric, but also the economic stability of many societies (Eisen, 2021). In order to overcome the associated challenges, a thorough understanding of the underlying factors is required. Carefully analysing these dynamics should help to provide decision-makers with the necessary information to develop effective policies. These measures should aim to improve the accessibility and affordability of housing, especially for lower income groups.

The relevance of this issue is particularly noticeable in urban areas, where the dynamics of property prices have a direct impact on quality of life: High housing costs can lead to less money being available for other important areas of life such as health, education and leisure activities, which affects overall life satisfaction (Dunn, 2020) (A. Sardina, 2021).In addition, the rise in property prices is exacerbating social inequality, as property ownership is increasingly becoming a privilege that only certain social classes can afford (Eisen, 2021). This is leading to increased segregation in the cities, with high-income and low-income population groups increasingly living in separate areas (A. Sardina, 2021). In addition, price trends on the property market have a significant impact on urban development. Rising land prices, for example, can lead to less space being available for public facilities and green spaces, as these generate less profit than commercial or residential property projects. This can further reduce the quality of life in densely populated urban centres. (Zhang Biao, 2012) Analysing property price trends is therefore a key component for planning future urban development projects and for designing social and economic policies aimed at creating a sustainable and equitable urban environment. The knowledge gained can help to design precise interventions that promote more balanced socio-economic development and thus contribute to strengthening overall social stability.

To make the question more tangible, it is helpful to identify different factors and understand how they affect property prices. For example, the question "Influence of crime rate and environmental factors on property prices" aims to understand the extent to which safety and the natural environment are decisive factors for buyers: By analysing how crime rates (CRIM) and proximity to the Charles River (CHAS) influence property prices (MEDV), we can assess how subjective and environmental perceptions of quality determine market values. These insights are essential for urban planners and developers to create attractive and safe neighbourhoods that are also economically viable (M. Topcu, 2009)

The question "Impact of building characteristics and environmental quality on property value" looks at the average number of rooms per dwelling (RM), the age of buildings (AGE) and air quality as measured by nitrogen oxide (NOX) concentrations, allowing us to understand the physical and environmental aspects that affect property value. This analysis helps to formulate guidelines for building standards and environmental protection measures that improve the quality of living and minimise environmental impact. (Jian-gu, 2012)

The question on "Socio-economic and infrastructural factors influencing property prices" analyzes the socio-economic and infrastructural factors, such as the percentage of the population with lower social status (LSTAT), the pupil-teacher ratio (PTRATIO) and access to the road network (RAD), thus providing deeper insights into the social determinants of property values. These findings are of great importance for the development of policies aimed at social equity and improved educational opportunities while stabilizing real estate prices.

By answering these specific questions, based on careful data analysis and review of existing theories, evidence-based recommendations can be developed for policies that promote sustainable and equitable urban development. (Jian-gu, 2012)

2. Data Understanding and Data Preparation

In the given data we have 506 observation against 14 variables. All having numeric values and no missing values. Out of these 14 variables the dependent variable is MEDV (Median value of owner-occupied homes) which in other words is the price of the house in a specific city. The 13 other variables are explanatory variables and are as follows;

- crim: Per capita crime rate by town
- zn: the percentage of residential floor area for large lots
- indus: Proportion of non-retail business acres per town
- NOX: Nitric oxide concentration (parts per 10 million)
- chas: proximity to the Charles River
- rm: Average number of rooms per dwelling
- AGE: Proportion of owner-occupied units built prior to 1940
- DIS: Weighted distances to five Boston employment centers
- RAD: Index of accessibility to radial highways
- TAX: Full-value property tax rate per $10,000
- ptratio: the pupil-teacher ratio
- b: the proportion of blacks in the population
- lstat: the lower status of the population

These characteristics provide a comprehensive overview of various aspects of the residential property market in a given geographic area, from environmental conditions to infrastructure and socioeconomic indicators. This data can be used for a variety of analyses, such as evaluating the quality of housing, examining market trends or assessing the influence of environmental factors on property prices.

2.1 Boxplots

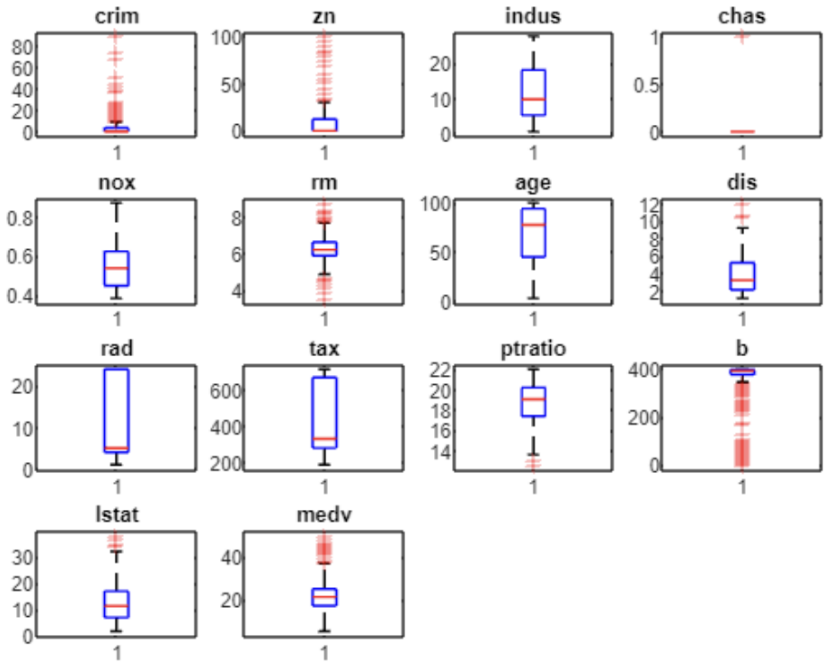


Figure 1. Boxplots

The boxplot for Crim and zn shows a right-skewed distribution with many outliers above the upper whisker and the Crim shows a minimum of 0.0063 and a maximum of 88.97. This is very widespread, which indicates that some areas are significantly safer than others. Outliers in this variable, which account for 13.04% of the data, could indicate particularly problematic neighborhoods or data collection errors. The Zn (proportion of residential plots for large properties) in turn varies from 0% to 100%, suggesting that some areas are designated exclusively for large residential plots, while others have no such plots at all. 13.44% of observations are considered outliers, indicating different development strategies in different neighborhoods. Indus (percentage of commercial non-retail space) ranges from 0.74% to 27.74%, with no outliers, indicating the more even distribution of commercial space in the region. The Chas (Charles River dummy variable), which indicates whether a property is located on the river or not, naturally only has values of 0 or 1. Approximately 6.92% of the data points are outliers, reflecting the smaller number of river locations. Nox (nitrogen oxide concentration) has a range of 0.385 to 0.871 parts per million. This variable shows no outliers, indicating a relatively even distribution of air quality conditions in the analyzed region. Rm (average number of rooms per dwelling) varies from 3.561 to 8.78, with 5.93% of the values categorized as outliers. These outliers could represent dwellings that are unusually small or large. Age (proportion of owner-occupied units built before 1940) shows that in some areas all units were built before 1940, while other areas have very recent properties. There are no outliers in this variable. Dis (Weighted distances to five Boston employment centres), Rad (Accessibility to radial highways), and Tax (Full value property tax rate per $10,000) show similar patterns with no outliers, suggesting uniformity in the collection of this infrastructure and tax data. Ptratio (student-teacher ratio) and B (1000(Bk - 0.63)^2, where Bk is the proportion of blacks in the city), with outlier proportions of 2.96% and 15.02%, respectively, indicate varying educational and demographic conditions. Lstat (lower status population) and Medv (median value of owner-occupied homes), with outliers of 1.19% and 7.31% respectively, reflect socio-economic differences and the varying property market in different neighborhoods. This analysis illustrates that the dataset is a rich source for understanding urban dynamics, with the outliers providing important clues to specific local circumstances or potential data quality issues.

2.2 Histograms

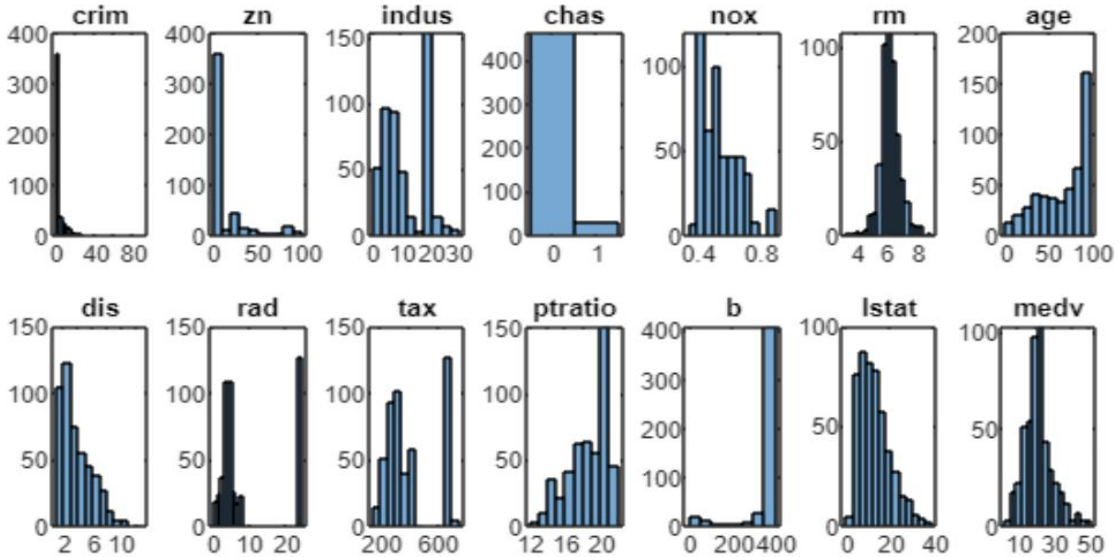


Figure 2. Histograms

The observation of the individual variables in the form of histograms largely supports the assumptions that could already be made in the boxplots. The histogram for the variable Crim (crime rate per capita) shows a strong skew to the right, with most areas showing a very low crime rate, but a few showing extreme values. This repeatedly indicates that some neighborhoods have significantly higher crime rates, although the majority of areas can be considered safe. In the case of Zn (proportion of residential plots for large properties), it is noticeable that many of the observations are 0, which indicates that large properties are rare in many areas. However, there are some higher values that indicate certain areas where larger properties are common. The Indus (proportion of non-retail commercial floorspace) distribution is broad and does not show a pronounced skew, indicating a diverse industrial use in the region. This indicates that industrial use is widely dispersed across the area analyzed. For Chas (Charles River dummy variable), most of the data points are at 0, which means that most of the properties are not located directly on the Charles River. There are only a few 1s, which emphasizes the rarity of river locations. In terms of Nox (nitrogen oxide concentration), the distribution appears to be unimodal and centred, indicating a relatively even distribution of air pollution in the region. The concentrations do not vary greatly, suggesting similar environmental conditions in different neighborhoods. For Rm (average number of rooms per dwelling), the histogram also shows an approximately normal distribution around a central mean value. This indicates standardisation in the size of dwellings, with an average number of rooms typical of most dwellings. The Age (proportion of units built before 1940) histogram shows that many units were built before 1940, as indicated by a peak at high percentages in the histogram. This shows that many of the properties in the area are older. The Dis (Weighted Distances to Five Boston Employment Centres) shows a right skewed distribution. This indicates that many properties are close to employment centres, with a few exceptions further away. The Rad (Accessibility to radial highways) histogram shows a concentration of middle values, indicating generally good accessibility to highways, with some urban areas being particularly well connected. The Tax (Full Value Property Tax Rate per $10,000) data shows a wide distribution, indicating different tax rates in different neighbourhoods. The variation could reflect different municipal tax policies. In terms of Ptratio (student-teacher ratio), the histogram shows a concentration at higher values, indicating larger class sizes in many schools. This could be an indicator of overcrowding or limited educational resources. Regarding the B (1000(Bk - 0.63)^2) distribution, it can be assumed that in many areas the proportion of black residents is low, but there are some outliers with high values, indicating specific communities with higher black populations. The Lstat (Lower Status of Population distribution is right skewed, indicating a concentration of higher socioeconomic status population in many parts of the region, with some areas showing higher proportions of poorer households. The Medv (median value of owner-occupied homes) is a broad distribution, showing that a variety of property values exist, with a clear peak in the middle price range and some more expensive properties as outliers. This indicates a heterogeneous property market structure.

2.3 Correlation Analysis

We checked for correlation in the data. The results are shown below in a correlation matrix heatmap. We can clearly see that there are some strong positive correlations, for example (rm) average number of rooms per dwelling and (medv) median value of owner-occupied homes. This indicates that homes with more rooms tend to have higher values. Another strong positive correlation is between (nox) nitrogen oxide concentration and (indus) the proportion of non-retail business acres per town. This indicates that areas with more industrial activity have higher levels of nitrogen oxide pollution. Another notable strong positive correlation is between (tax) full-value property tax rate per $10,000 and (rad) index of accessibility to radial highways. This indicates that areas with higher property tax rates have better accessibility to radial highways.

On the contrary, there were some strong negative correlations, for example (nox) nitrogen oxide concentration and (dis) weighted distances to five Boston employment centers. This suggests that areas closer to employment centers have higher nitrogen oxide levels. Another strong negative correlation is between (lstat) the percentage of lower status population and (medv) the median value of owner-occupied homes. This finding suggests that areas with a higher percentage of lower-status population tend to have lower home values. In addition, it seems that (chas) proximity to the Charles River is not strongly correlated with any other variable.

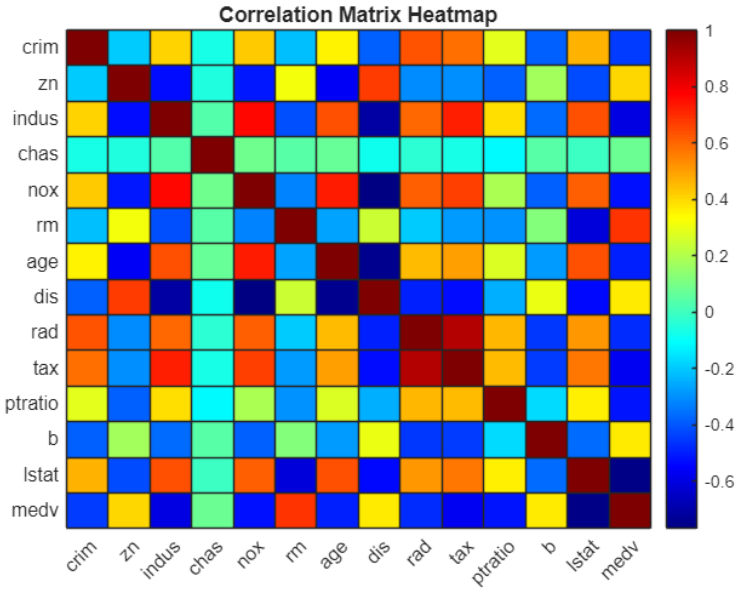


Figure 3. Correlation Matrix Heatmap

3. Model Development

To ensure the dataset was ready for model training, we performed a couple of preprocessing steps. These steps included handling missing values and removing outliers. We checked for any missing values in the dataset and confirmed there were none, ensuring the integrity of our data. We removed outliers by excluding rows where the median value (medv) was less than 50.0, as these extreme values could skew the results. In addition, we standardized the dataset which is important for models which are sensitive to the scale of the input data. By doing the splitting the data the data into training (70%), validation 15%), and test sets (15%), we ensured that models are trained on one set of data and validated on another, preventing overfitting, and ensuring the model's generalizability.

Our goal was to assess the impact of each explanatory variable on the dependent variable directly. Therefore, we did not conduct PCA. PCA transforms the original variables into principal components, which are combinations of the original variables. This transformation makes it difficult to interpret the direct impact of individual explanatory variables on Y. We employed three different regression models to predict the median value of owner-occupied homes (medv) and compared their performances: linear regression, decision tree regression, and ridge regression. We trained the linear regression model using the training set and extracted and interpreted coefficients to understand the impact of each explanatory variable on the dependent variable.

For decision tree, we conducted a grid search to find the optimal parameters (MaxNumSplits and MinLeafSize). Once the optimal parameters were found, we trained the decision tree model with those parameters and extracted feature importance to identify which variables most

significantly impact the target variable. We implemented ridge regression to address multicollinearity by adding a penalty term to the regression model. We conducted cross-validation to find the best lambda (regularization parameter) and extracted and interpreted coefficients from the ridge regression model to compare with those from the linear regression.

In addition, we evaluated and compared the performance of the models using the Mean Squared Error (MSE) on the validation and test sets. The analysis indicated that while linear regression and decision tree regression performed similarly on the test set, ridge regression's performance was less favorable. This discrepancy highlights the importance of model selection and tuning to achieve optimal performance. By following these methodologies, we ensured each model's design was well-suited to the dataset and problem at hand. This comprehensive approach allowed for a robust comparison of methodologies, ensuring the best possible recommendations for predicting property values.

4. Results and Discussion

Our study aimed to analyze various factors affecting property prices (medv) in the Boston Housing dataset. We applied multiple regression models, including linear regression, ridge regression, and decision tree models, to evaluate and compare the significance of these factors. The models provided insights into the relationships between the explanatory variables and property prices, as well as their relative importance.

4.1 Coefficients and variable importance results

The results of all models are summarized at the table below to help with comparing and visualizing results. The ridge regression model, which helps mitigate multicollinearity, showed similar trends to the linear regression model but with slightly different coefficient magnitudes due to regularization. It is worth noting that the scores of decision tree feature importance are always non-negative and do not convey the direction of the relationship with the target variable. It only indicates the relative importance of each feature in making predictions. The larger the importance value, the more significant the explanatory variable is in predicting the target variable.

Variable	Linear regression	Ridge regression	Decision Tree
crim	-0.017	-0.001	0.026
zn	0.03	0.003	0
indus	-0.014	-0.001	0
chas	0.415	1.0221e-05	0
nox	-11.69	-4.6384e-06	0.034
rm	3.49	0.0004	0.52
age	-0.009	-0.003	0
dis	-1.11	0.0003	0.014
rad	0.27	-0.0008	0
tax	-0.015	-0.018	0.048
ptratio	-0.87	0.0001	0.032
b	0.012	0.021	0.011
lstat	-0.41	-0.0017	1.78

Table 1. Comparative summary of the coefficients and variable importance values from the DT model

The variable crim (crime rate) exhibits a negative relationship with property values in both linear regression (-0.017) and ridge regression (-0.001), suggesting that an increase in the crime rate tends to decrease property values. The decision tree model assigns an importance value of 0.026. Although crime rate is a moderately important feature in predicting property values, its effect is notably diminished in the ridge regression due to regularization. For zn (proportion of residential land zoned for large lots), both linear regression (0.030) and ridge regression (0.003) show a positive relationship, indicating that higher zoning for large lots correlates with increased property values. The decision tree model reflects this with a low importance value of 0, suggesting that while zoning plays a role, its predictive power is minimal in a non-linear model. The indus variable (proportion of non-retail business acres) reveals a negative relationship in linear regression (-0.014) and ridge regression (-0.001), indicating that more industrial land tends to decrease property values. The decision tree model assigns an importance value of 0, suggesting that it is not a significant predictor in the non-linear model. Regarding chas (proximity to Charles River), linear regression (0.415) shows a positive impact, while ridge regression (1.0221e-05) and the decision tree model (0) indicate a negligible effect. This discrepancy highlights the linear model's sensitivity to outliers and the river's minimal role in predicting property values in more complex models. The nox variable (nitrogen oxide concentration) has a strong negative relationship with property values, as shown by linear regression (-11.69), suggesting that higher pollution levels significantly decrease property values. Ridge regression (-4.6384e-06) almost nullifies this effect due to regularization, while the decision tree model (0.034) assigns moderate importance, underscoring the role of air quality in property value predictions. The average number of rooms (rm) shows a strong positive relationship with property values in linear regression (3.49), suggesting that more rooms significantly increase property values. Ridge regression (0.0004) greatly reduces this impact, but still indicates a positive effect. The decision tree model (0.52) aligns with these findings, indicating high importance, consistent with the variable's strong predictive power. The age variable (proportion of older homes) has a slight negative relationship in linear regression (-0.009) and ridge regression (-0.003), suggesting that older homes slightly decrease property values. The decision tree model assigns a low importance value of 0, reflecting its minor role in property value predictions. For dis (distance to employment centers), linear regression (-1.11) shows a negative relationship, indicating that greater distances reduce property values. Ridge regression (0.0003) greatly reduces or nullifies this impact and converts to slightly positive. The decision tree model assigns a low importance value of 0.014, indicating that while distance to employment centers matters, its predictive power is limited. The variable rad (accessibility to radial highways) shows a positive relationship with property values in linear regression (0.27) and slightly negative value in ridge regression (-0.0008), suggesting that better accessibility increases property values. However, the decision tree model assigns no importance to this variable, indicating its lesser role in the non-linear model. For tax (property tax rate), linear regression (-0.015) and ridge regression (-0.018) both show a negative relationship, suggesting that higher taxes decrease property values. The decision tree model assigns a low importance value of 0.048, indicating that while taxes matter, their impact is relatively minor. The ptratio variable (pupil-teacher ratio) has a negative relationship with property values in linear regression (-0.87) and almost zero relationship in ridge regression (0.0001), suggesting that higher ratios decrease property values. The decision tree model (0.032) assigns moderate importance, indicating its role in predicting property values. Interestingly, for b (proportion of black population), linear regression (0.012) and ridge regression (0.021) both show a slight positive relationship, suggesting that a higher black population slightly increases property values. The decision tree model (0.011) assigns moderate importance, indicating its role in property value predictions. Finally, lstat (percentage of lower status population) has a negative relationship with property values in linear regression (-0.41) and ridge regression (-0.0017),

suggesting that higher percentages of lower status population in the area decrease property values. The decision tree model (1.78) assigns high importance, highlighting its strong predictive power and it is the most important predictor of median property values according to the decision tree model.

The strong predictors across models (rm, lstat, and crim) consistently show significant impacts across linear regression, ridge regression, and the decision tree model, highlighting their importance in predicting property values. Ridge regression generally reduces the magnitude of coefficients, indicating the presence of multicollinearity, which regularization helps address by shrinking the coefficients. The decision tree model provides additional insights by showing the importance of variables like lstat and rm, confirming their strong predictive power, while variables like rad show no importance, indicating their lesser role in the non-linear model. These findings enhance our understanding of the relative importance of different factors in predicting property values, thereby informing urban planning and real estate development decisions.

In conclusion, our analysis reveals that crime rate, environmental quality, building characteristics, and socio-economic factors significantly influence property prices. The findings suggest that urban planners and policymakers should prioritize improving environmental quality, enhancing socio-economic conditions, and considering the crime rate's impact when developing policies to create attractive, safe, and economically viable neighborhoods. These insights align with existing theories and provide evidence-based recommendations for sustainable and equitable urban development.

4.2 Model Evaluation and Comparison

To assess the effectiveness of the models used in predicting property prices (medv) in the Boston Housing dataset, we evaluated three different models: linear regression, decision tree, and ridge regression. The performance of these models was compared using Mean Squared Error (MSE) for both validation and test sets. The results are summarized in Table 2 and visualized in the bar graphs below.

Model	Validation MSE	Test MSE
Linear Regression	14.424	17.281
Ridge Regression	45.825	44.11
Decision Tree	8.9238	11.769

Table 2. MSE results.

Validation MSE indicates the model's performance on the validation set. Lower values suggest better model performance during the validation phase. Test MSE indicates the model's performance on the test set. Lower values suggest better generalization to unseen data.

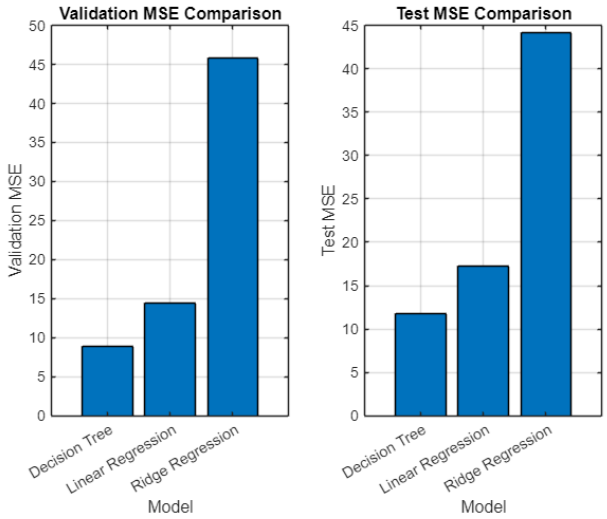


Figure 3. Bar graph of MSE results.

The comparison of the three models reveals distinct differences in their performance based on the Mean Squared Error (MSE) metrics for both validation and test datasets. The linear regression model demonstrates a strong performance with a validation MSE of 14.424 and a test MSE of 17.281. These relatively low MSE values indicate that the linear regression model achieves a good balance between fitting the training data and generalizing to unseen data. This suggests that linear regression is a reliable model for predicting property prices in the Boston Housing dataset, capturing the relationship between the features and the target variable effectively. The decision tree model shows the lowest MSE values among the three models, with a validation MSE of 8.9238 and a test MSE of 11.769. This suggests that the decision tree model is highly effective in fitting the training data and generalizing to the test data. The low MSE values indicate that the decision tree can capture complex patterns in the data that linear regression might miss. However, the performance gap between validation and test MSE implies a risk of overfitting, where the model might perform well on the validation set but slightly less effectively on new, unseen data. In contrast, the ridge regression model underperforms significantly with the highest validation MSE of 45.825 and test MSE of 44.11. Despite ridge regression's strength in handling multicollinearity by applying regularization, its high MSE values suggest that the chosen regularization parameter or the overall model complexity may not be optimal. This poor performance highlights the need for further tuning of hyperparameters or exploring more suitable regularization techniques to improve the model's efficacy. These insights indicate that for the Boston Housing dataset, the decision tree model provides the most accurate predictions among the three models, although caution is required to avoid overfitting. Linear regression also proves to be a reliable model, balancing simplicity and predictive power. Ridge regression, while generally robust against multicollinearity, requires significant adjustments to enhance its performance for this specific dataset.

5. References

A. Sardina, S. T. (2021). A Preliminary Study of the Correlates of Leisure Interests and Constraints Among Adults Residing in Public Housing. *Journal of Aging and Environment.*, 113-135.

Dunn, J. (2020). Housing and Health. *International Encyclopedia of Human Geography*, 75-78.

E. Glaeser, J. G. (2005). Why have housing prices gone up? *Urban Economics & Regional Studies eJournal*, 329-333.

Eisen, M. P. (2021). Rising Inequality and Spatial Social Segregation due to Urbanization and Increasing Housing Prices.
Jian-gu, C. (2012). The Empirical Analysis of House Price Influencing Factors in Our Country. *Journal of Zhangzhou Normal University*.

M. Topcu, A. S. (2009). The Analysis of Urban Features that Affect Land Values in Residential Areas. *7th International Space Syntax Symposium.* Stockolm.

Zhang Biao, X. G. (2012). The Effects of Public Green Spaces on Residential Property Value in Beijing.

6. Contribution of Group Members

Nicolas Pauli

- Preparation
- Introduction, Background and Problem Definition
- Data Understanding and Data Preparation
- References
- Preparing presentation slides

Arttu Rytkönen

- Introduction, Background, and Problem Definition check
- Data Understanding and Data Preparation (missing values check, Data splits, correlation and heatmap matrix)
- Model Development Matlab + document (Linear Regression, Decision Tree Regression, Ridge Regression, Model Evaluation and Comparison)
- Results and discussion Matlab + document

Sunjida Haque

- Interpretation Data Understanding and Data Preparation to Result and Discussion

Saleh Shahbaz

- Dataset selection
- Coding for Data understanding and Data Protection (Descriptive Analysis, Plots, correlation, Outliner removal etc.)
- Final Formatting of Report
- Finalizing Presentation slides

7 Appendix: Matlab code

Clearing Matlab

```
% Clearing MATLAB
clear all; clc; close all;
```

Setting Up Data

```
% Importing dataset
% Import data from text file
BostonHousing = readtable("C:\Users\ajryt\OneDrive\Documents\MATLAB\Business Analytics\BostonHousing.csv");

% Display results
BostonHousing
```

BostonHousing = 506×14 table

	crim	zn	indus	chas	nox	rm	age
1	0.0063	18	2.3100	0	0.5380	6.5750	65.20
2	0.0273	0	7.0700	0	0.4690	6.4210	78.90
3	0.0273	0	7.0700	0	0.4690	7.1850	61.10
4	0.0324	0	2.1800	0	0.4580	6.9980	45.80
5	0.0691	0	2.1800	0	0.4580	7.1470	54.20
6	0.0299	0	2.1800	0	0.4580	6.4300	58.70
7	0.0883	12.5000	7.8700	0	0.5240	6.0120	66.60
8	0.1446	12.5000	7.8700	0	0.5240	6.1720	96.10
9	0.2112	12.5000	7.8700	0	0.5240	5.6310	1
10	0.1700	12.5000	7.8700	0	0.5240	6.0040	85.90
11	0.2249	12.5000	7.8700	0	0.5240	6.3770	94.30
12	0.1175	12.5000	7.8700	0	0.5240	6.0090	82.90
13	0.0938	12.5000	7.8700	0	0.5240	5.8890	
14	0.6298	0	8.1400	0	0.5380	5.9490	61.80
15	0.6380	0	8.1400	0	0.5380	6.0960	84.50
16	0.6274	0	8.1400	0	0.5380	5.8340	56.50
17	1.0539	0	8.1400	0	0.5380	5.9350	29.30
18	0.7842	0	8.1400	0	0.5380	5.9900	81.70
19	0.8027	0	8.1400	0	0.5380	5.4560	36.60
20	0.7258	0	8.1400	0	0.5380	5.7270	69.50
21	1.2518	0	8.1400	0	0.5380	5.5700	98.10
22	0.8520	0	8.1400	0	0.5380	5.9650	89.20
23	1.2325	0	8.1400	0	0.5380	6.1420	91.70
24	0.9884	0	8.1400	0	0.5380	5.8130	1
25	0.7503	0	8.1400	0	0.5380	5.9240	94.10
26	0.8405	0	8.1400	0	0.5380	5.5990	85.70
27	0.6719	0	8.1400	0	0.5380	5.8130	90.30
28	0.9558	0	8.1400	0	0.5380	6.0470	88.80
29	0.7730	0	8.1400	0	0.5380	6.4950	94.40
30	1.0025	0	8.1400	0	0.5380	6.6740	87.30
31	1.1308	0	8.1400	0	0.5380	5.7130	94.10
32	1.3547	0	8.1400	0	0.5380	6.0720	1
33	1.3880	0	8.1400	0	0.5380	5.9500	
34	1.1517	0	8.1400	0	0.5380	5.7010	
35	1.6128	0	8.1400	0	0.5380	6.0960	96.90
36	0.0642	0	5.9600	0	0.4990	5.9330	68.20
37	0.0974	0	5.9600	0	0.4990	5.8410	61.40
38	0.0801	0	5.9600	0	0.4990	5.8500	41.50
39	0.1751	0	5.9600	0	0.4990	5.9660	30.20
40	0.0276	75	2.9500	0	0.4280	6.5950	21.80
41	0.0336	75	2.9500	0	0.4280	7.0240	15.80
42	0.1274	0	6.9100	0	0.4480	6.7700	2.90
43	0.1415	0	6.9100	0	0.4480	6.1690	6.60
44	0.1594	0	6.9100	0	0.4480	6.2110	6.50
45	0.1227	0	6.9100	0	0.4480	6.0690	
46	0.1714	0	6.9100	0	0.4480	5.6820	33.80
47	0.1884	0	6.9100	0	0.4480	5.7860	33.30
48	0.2293	0	6.9100	0	0.4480	6.0300	85.50
49	0.2539	0	6.9100	0	0.4480	5.3990	95.30
50	0.2198	0	6.9100	0	0.4480	5.6020	
51	0.0887	21	5.6400	0	0.4390	5.9630	45.70
52	0.0434	21	5.6400	0	0.4390	6.1150	
53	0.0536	21	5.6400	0	0.4390	6.5110	21.10
54	0.0498	21	5.6400	0	0.4390	5.9980	21.40
55	0.0136	75	4	0	0.4100	5.8880	47.60
56	0.0131	90	1.2200	0	0.4030	7.2490	21.90
57	0.0205	85	0.7400	0	0.4100	6.3830	35.70
58	0.0143	100	1.3200	0	0.4110	6.8160	40.50

59	0.1545	25	5.1300	0	0.4530	6.1450	29.20
60	0.1033	25	5.1300	0	0.4530	5.9270	47.20
61	0.1493	25	5.1300	0	0.4530	5.7410	66.20
62	0.1717	25	5.1300	0	0.4530	5.9660	93.40
63	0.1103	25	5.1300	0	0.4530	6.4560	67.80
64	0.1265	25	5.1300	0	0.4530	6.7620	43.40
65	0.0195	17.5000	1.3800	0	0.4161	7.1040	59.50
66	0.0358	80	3.3700	0	0.3980	6.2900	17.80
67	0.0438	80	3.3700	0	0.3980	5.7870	31.10
68	0.0579	12.5000	6.0700	0	0.4090	5.8780	21.40
69	0.1355	12.5000	6.0700	0	0.4090	5.5940	36.80
70	0.1282	12.5000	6.0700	0	0.4090	5.8850	
71	0.0883	0	10.8100	0	0.4130	6.4170	6.60
72	0.1588	0	10.8100	0	0.4130	5.9610	17.50
73	0.0916	0	10.8100	0	0.4130	6.0650	7.80
74	0.1954	0	10.8100	0	0.4130	6.2450	6.20
75	0.0790	0	12.8300	0	0.4370	6.2730	
76	0.0951	0	12.8300	0	0.4370	6.2860	
77	0.1015	0	12.8300	0	0.4370	6.2790	74.50
78	0.0871	0	12.8300	0	0.4370	6.1400	45.80
79	0.0565	0	12.8300	0	0.4370	6.2320	53.70
80	0.0839	0	12.8300	0	0.4370	5.8740	36.60
81	0.0411	25	4.8600	0	0.4260	6.7270	33.50
82	0.0446	25	4.8600	0	0.4260	6.6190	70.40
83	0.0366	25	4.8600	0	0.4260	6.3020	32.20
84	0.0355	25	4.8600	0	0.4260	6.1670	46.70
85	0.0506	0	4.4900	0	0.4490	6.3890	
86	0.0573	0	4.4900	0	0.4490	6.6300	56.10
87	0.0519	0	4.4900	0	0.4490	6.0150	45.10
88	0.0715	0	4.4900	0	0.4490	6.1210	56.80
89	0.0566	0	3.4100	0	0.4890	7.0070	86.30
90	0.0530	0	3.4100	0	0.4890	7.0790	63.10
91	0.0468	0	3.4100	0	0.4890	6.4170	66.10
92	0.0393	0	3.4100	0	0.4890	6.4050	73.90
93	0.0420	28	15.0400	0	0.4640	6.4420	53.60
94	0.0288	28	15.0400	0	0.4640	6.2110	28.90
95	0.0429	28	15.0400	0	0.4640	6.2490	77.30
96	0.1220	0	2.8900	0	0.4450	6.6250	57.80
97	0.1150	0	2.8900	0	0.4450	6.1630	69.60
98	0.1208	0	2.8900	0	0.4450	8.0690	
99	0.0819	0	2.8900	0	0.4450	7.8200	36.90
100	0.0686	0	2.8900	0	0.4450	7.4160	62.50

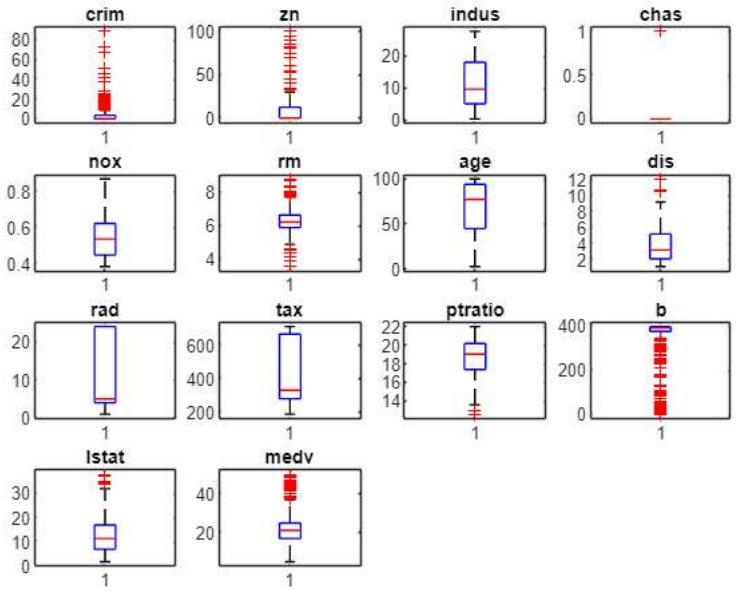
```
data = table2array(BostonHousing);

 [Observations,Variables] = size(BostonHousing)
```

```
Observations = 506
Variables = 14
```

```
 % Set the random number generator seed for reproducibility
seed = 42; % You can choose any integer value
rng(seed);
```

2.2 Data Understanding and Data Preparation

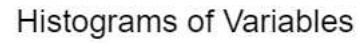
```
% Creating boxplot
figure;

% Loop through each variable
for i = 1:Variables
    subplot(4, 4, i); % Create subplots in a 4x4 grid
    boxplot(data(:,i));
    title(BostonHousing.Properties.VariableNames(i));
end
```

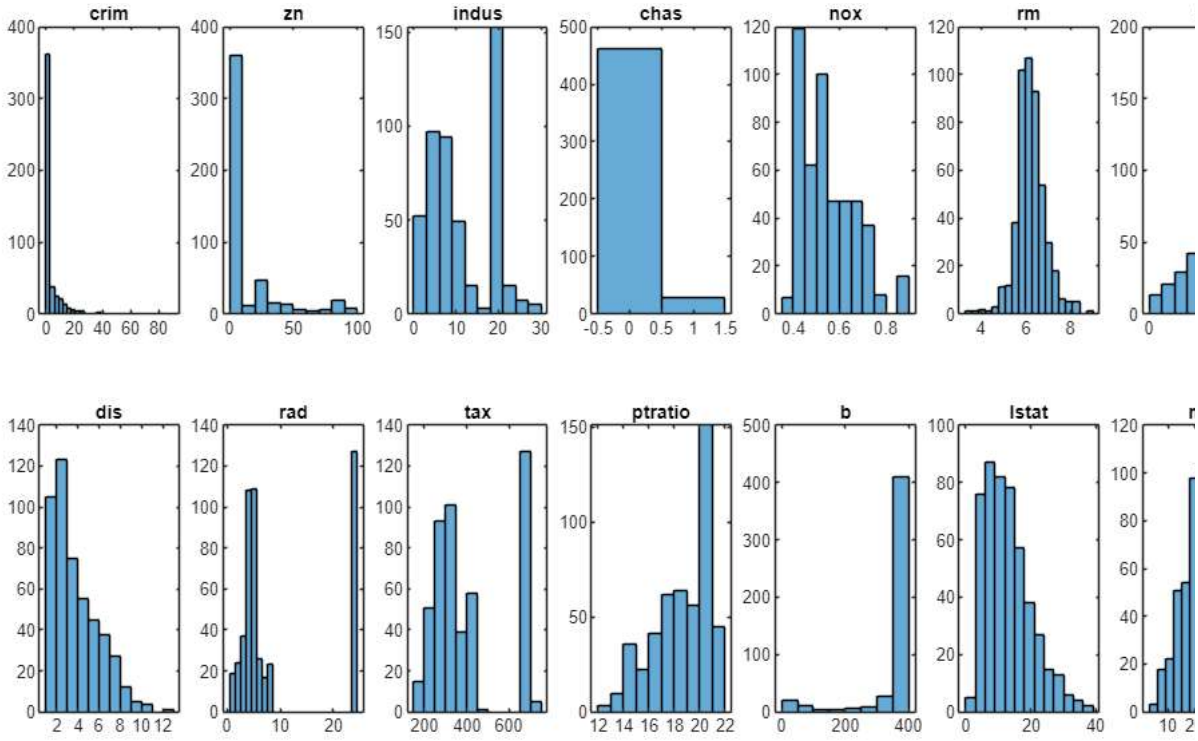



```
for i = 1:Variables
    q1 = quantile(data(:,i), 0.25);
    q3 = quantile(data(:,i), 0.75);
    irq = q3 - q1;
    % Find outliers
    outliers = data(:,i) < (q1 - 1.5 * irq) | data(:,i) > (q3 + 1.5 * irq);
    perc = sum(outliers) * 100.0 / numel(data(:,i));
    fprintf('Variable %s outliers = %.2f%%\n', BostonHousing.Properties.VariableNames{i}, perc);
end
```

```
Variable crim outliers = 13.04%
Variable zn outliers = 13.44%
Variable indus outliers = 0.00%
Variable chas outliers = 6.92%
Variable nox outliers = 0.00%
Variable rm outliers = 5.93%
Variable age outliers = 0.00%
Variable dis outliers = 0.99%
Variable rad outliers = 0.00%
Variable tax outliers = 0.00%
Variable ptratio outliers = 2.96%
Variable b outliers = 15.02%
Variable lstat outliers = 1.19%
Variable medv outliers = 7.31%
```

```
% Find rows where 'MEDV' column value is less than 50.0 to remove outliners
data = data(data(:,end) < 50.0, :);

% Plotting distribution of data'
fig = figure;
set(fig, 'Position', [0, 0, 1200, 600]); % Set figure size

num_cols = 7;
num_rows = ceil(size(data, 2) / num_cols);

for i = 1:Variables
    subplot(num_rows, num_cols, i);
    histogram(data(:, i));
    title(BostonHousing.Properties.VariableNames(i));
end

sgtitle('Histograms of Variables');

% Adjust layout
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
```






```
% Compute mean, median, minimum, maximum, and quartiles for each variable
means = mean(data)
```

```
means = 1×14
    3.6432   11.1122   11.1131    0.0592    0.5543    6.2455   68.2790    3.8345    9.5143  408.(
```

```
medians = median(data)
```

```
medians = 1×14
    0.2475         0    9.6900         0    0.5380    6.1850   76.8000    3.2759    5.0000  330.(
```

```
mins = min(data)
```

```
mins = 1×14
    0.0063         0    0.7400         0    0.3850    3.5610    2.9000    1.1370    1.0000  187.(
```

```
maxs = max(data)
```

```
maxs = 1×14
   88.9762  100.0000   27.7400    1.0000    0.8710    8.7800  100.0000   12.1265   24.0000  711.(
```

```
q1 = quantile(data, 0.25)
```

```
q1 = 1×14
    0.0820         0    5.1900         0    0.4490    5.8800   44.4000    2.1107    4.0000  280.(
```

```
q3 = quantile(data, 0.75)
```

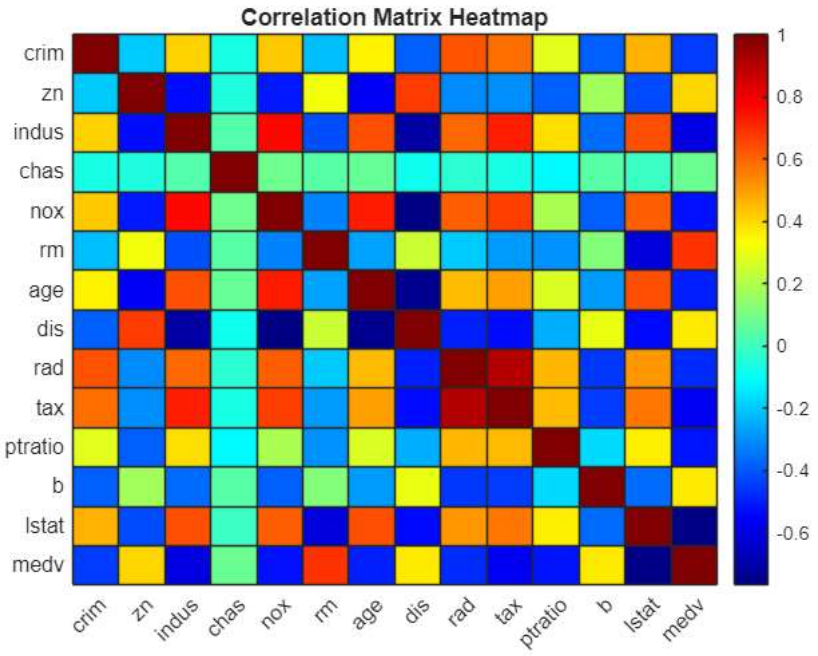
```
q3 = 1×14
    3.6737   12.5000   18.1000         0    0.6240    6.5790   93.9000    5.2146   24.0000  666.(
```

```
% Calculate and display the correlation matrix
corrMatrix = corr(data);
disp(corrMatrix);
```

```
    1.0000   -0.1991    0.4081   -0.0642    0.4205   -0.2193    0.3538   -0.3822    0.6274
   -0.1991    1.0000   -0.5271   -0.0539   -0.5121    0.3105   -0.5632    0.6732   -0.3077
    0.4081   -0.5271    1.0000    0.0358    0.7652   -0.4124    0.6380   -0.7103    0.5961
   -0.0642   -0.0539    0.0358    1.0000    0.0856    0.0450    0.0712   -0.0777   -0.0328
    0.4205   -0.5121    0.7652    0.0856    1.0000   -0.3226    0.7277   -0.7681    0.6122
   -0.2193    0.3105   -0.4124    0.0450   -0.3226    1.0000   -0.2685    0.2458   -0.1958
    0.3538   -0.5632    0.6380    0.0712    0.7277   -0.2685    1.0000   -0.7430    0.4519
   -0.3822    0.6732   -0.7103   -0.0777   -0.7681    0.2458   -0.7430    1.0000   -0.4919
    0.6274   -0.3077    0.5961   -0.0328    0.6122   -0.1958    0.4519   -0.4919    1.0000
    0.5837   -0.3029    0.7177   -0.0677    0.6674   -0.2820    0.4997   -0.5320    0.9090
    0.2871   -0.3818    0.3877   -0.1168    0.1884   -0.2933    0.2685   -0.2468    0.4560
   -0.3845    0.1761   -0.3634    0.0417   -0.3831    0.1192   -0.2790    0.2994   -0.4515
    0.4618   -0.4221    0.6365   -0.0065    0.6124   -0.6104    0.6379   -0.5365    0.5102
   -0.4501    0.4046   -0.6000    0.0748   -0.5245    0.6866   -0.4929    0.3688   -0.4763
```

```
% Create the heatmap
figure;
h = heatmap(BostonHousing.Properties.VariableNames(1:end), ... % x-axis
            BostonHousing.Properties.VariableNames(1:end), ... % y-axis
```

```
            corrMatrix, ... % Correlation matrix data
            'Colormap', jet, ...
            'ColorbarVisible', 'on');
title('Correlation Matrix Heatmap');
```



Model Development

```
BostonHousing = BostonHousing(BostonHousing.medv < 50.0, :);
% Check for missing values in the table
missingValues = ismissing(BostonHousing);
% Display missing values summary
missingSummary = sum(missingValues);
for i = 1:length(missingSummary)
    fprintf('%s: %d\n', BostonHousing.Properties.VariableNames{i}, missingSummary(i));
end
```

```
crim: 0
zn: 0
indus: 0
chas: 0
nox: 0
rm: 0
age: 0
dis: 0
rad: 0
tax: 0
ptratio: 0
b: 0
lstat: 0
medv: 0
```

```
% Data standardization
X = BostonHousing(:, 1:end-1); % Explanatory variables
X = table2array(X);
Y = BostonHousing.medv; % Dependent variable
% Split data into 70% training, 15% validation, and 15% test sets
cv = cvpartition(size(X, 1), 'HoldOut', 0.3);
idx = cv.test;
% Separate into training and test data
XTrain = X(~idx, :);
YTrain = Y(~idx);
XTest = X(idx, :);
YTest = Y(idx);
% Further split the training data into training and validation sets
cv2 = cvpartition(size(XTrain, 1), 'HoldOut', 0.1765); % 15% of 70% is 10.5%
idx2 = cv2.test;
XValidation = XTrain(idx2, :);
YValidation = YTrain(idx2);
XTrain = XTrain(~idx2, :);
YTrain = YTrain(~idx2);
```

Linear Regression

```
%Train Linear Regression Model
linearModel = fitlm(XTrain, YTrain);

% Validate Linear Regression Model
YValidationPredLinear = predict(linearModel, XValidation);
validation_mse_linear = mean((YValidation - YValidationPredLinear).^2);

% Test Linear Regression Model
YTestPredLinear = predict(linearModel, XTest);
test_mse_linear = mean((YTest - YTestPredLinear).^2);
% Extract coefficients
coefficients = linearModel.Coefficients.Estimate;
% Extract variable names (including intercept)
variableNames = ['(Intercept)'; BostonHousing.Properties.VariableNames(1:end-1)'];
coeffTable = table(variableNames, coefficients, 'VariableNames', {'Variable', 'Coefficient'});
disp(coeffTable);
```

Variable	Coefficient
{'(Intercept)'}	31.596
{'crim' }	-0.016629
{'zn' }	0.030146
{'indus' }	-0.014289
{'chas' }	0.41452
{'nox' }	-11.699
{'rm' }	3.4906
{'age' }	-0.0087401
{'dis' }	-1.1091
{'rad' }	0.27224
{'tax' }	-0.015216
{'ptratio' }	-0.86599

```
%Let's evaluate the model's performance from test and validation sets
disp(['Test MSE for Linear Regression: ', num2str(test_mse_linear)]);
```

```
Test MSE for Linear Regression: 17.2812
```

```
disp(['Validation MSE for Linear Regression: ', num2str(validation_mse_linear)]);
```

```
Validation MSE for Linear Regression: 14.4236
```

Decision Tree Regression

```
% Define parameter grid
maxNumSplits_values = [5, 10, 20, 50, 100];
minLeafSize_values = [1, 5, 10, 20];
best_maxNumSplits = 0;
best_minLeafSize = 0;
best_validation_mse_tree = inf;

% Manual Grid Search for Decision Tree
for maxNumSplits = maxNumSplits_values
    for minLeafSize = minLeafSize_values
        % Train Decision Tree Model
        tree_model = fitrtree(XTrain, YTrain, 'MaxNumSplits', maxNumSplits, 'MinLeafSize', minLeafSize);

        % Validate Decision Tree Model
        YValidationPred = predict(tree_model, XValidation);
        validation_mse = mean((YValidation - YValidationPred).^2);

        % Update best parameters if current setting has lower validation MSE
        if validation_mse < best_validation_mse_tree
            best_validation_mse_tree = validation_mse;
            best_maxNumSplits = maxNumSplits;
            best_minLeafSize = minLeafSize;
        end
    end
end

% Train final Decision Tree model with best parameters
final_tree_model = fitrtree(XTrain, YTrain, 'MaxNumSplits', best_maxNumSplits, 'MinLeafSize', best_minLeafSize);

% Test Decision Tree Model
YTestPred = predict(final_tree_model, XTest);
test_mse_tree = mean((YTest - YTestPred).^2);

disp(['Best MaxNumSplits: ', num2str(best_maxNumSplits)]);
```

```
Best MaxNumSplits: 20
```

```
disp(['Best MinLeafSize: ', num2str(best_minLeafSize)]);
```

```
Best MinLeafSize: 5
```

```
disp(['Validation MSE for best Decision Tree model: ', num2str(best_validation_mse_tree)]);
```

```
Validation MSE for best Decision Tree model: 8.9238
```

```
disp(['Test MSE for best Decision Tree model: ', num2str(test_mse_tree)]);
```

```
Test MSE for best Decision Tree model: 11.7688
```

```
% Extract variable importance
variableImportance = predictorImportance(final_tree_model);
% Extract variable names (excluding the dependent variable 'medv')
variableNames = BostonHousing.Properties.VariableNames(1:end-1)';
%Let's create a table of variable importance
variableImportanceTable = table(variableNames, variableImportance', ...
        'VariableNames', {'Variable', 'Importance'});
disp(variableImportanceTable);
```

```
      Variable       Importance
    _____________    __________

    {'crim'   }      0.026141
    {'zn'     }            0
    {'indus'  }            0
    {'chas'   }            0
    {'nox'    }      0.033737
    {'rm'     }       0.52146
    {'age'    }            0
    {'dis'    }       0.01375
    {'rad'    }            0
    {'tax'    }      0.048159
    {'ptratio'}      0.031913
    {'b'      }      0.010883
```

Ridge Regression

```
% Range of lambda values for Ridge Regression
lambda_values = logspace(-4, 4, 50); % 50 values between 10^-4 and 10^4
best_lambda = 0;
best_validation_mse_ridge = inf;

% Manual Grid Search for Ridge Regression
for lambda = lambda_values
    % Train Ridge Regression Model
    ridge_model = fitrlinear(XTrain, YTrain, 'Learner', 'leastsquares', 'Regularization', 'ridge', 'Lambda', lambda);

    % Validate Ridge Regression Model
    YValidationPred = predict(ridge_model, XValidation);
    validation_mse = mean((YValidation - YValidationPred).^2);

    % Update best lambda if current lambda has lower validation MSE
    if validation_mse < best_validation_mse_ridge
        best_validation_mse_ridge = validation_mse;
        best_lambda = lambda;
    end
end

% Train final Ridge Regression model with best lambda
final_ridge_model = fitrlinear(XTrain, YTrain, 'Learner', 'leastsquares', 'Regularization', 'ridge', 'Lambda', best_lambda);

% Test Ridge Regression Model
YTestPred = predict(final_ridge_model, XTest);
test_mse_ridge = mean((YTest - YTestPred).^2);

disp(['Best Lambda: ', num2str(best_lambda)]);
```

```
Best Lambda: 0.0001
```

```
disp(['Validation MSE for Ridge Regression model: ', num2str(best_validation_mse_ridge)]);
```

```
Validation MSE for Ridge Regression model: 45.8249
```

```
disp(['Test MSE for Ridge Regression model: ', num2str(test_mse_ridge)]);
```

```
Test MSE for Ridge Regression model: 44.1101
```

```
% Extract coefficients from Ridge Regression model
coefficients_ridge = final_ridge_model.Beta;
intercept_ridge = final_ridge_model.Bias;

% Combine the intercept and coefficients into one array
coefficients_ridge = [intercept_ridge; coefficients_ridge];

% Extract feature names including intercept
VariableNames = [{'Intercept'}, BostonHousing.Properties.VariableNames(1:end-1)];

% Create a table for better readability
coeffTable_ridge = table(VariableNames(:), coefficients_ridge, ...
    'VariableNames', {'Variable', 'Coefficient'});
disp(coeffTable_ridge);
```

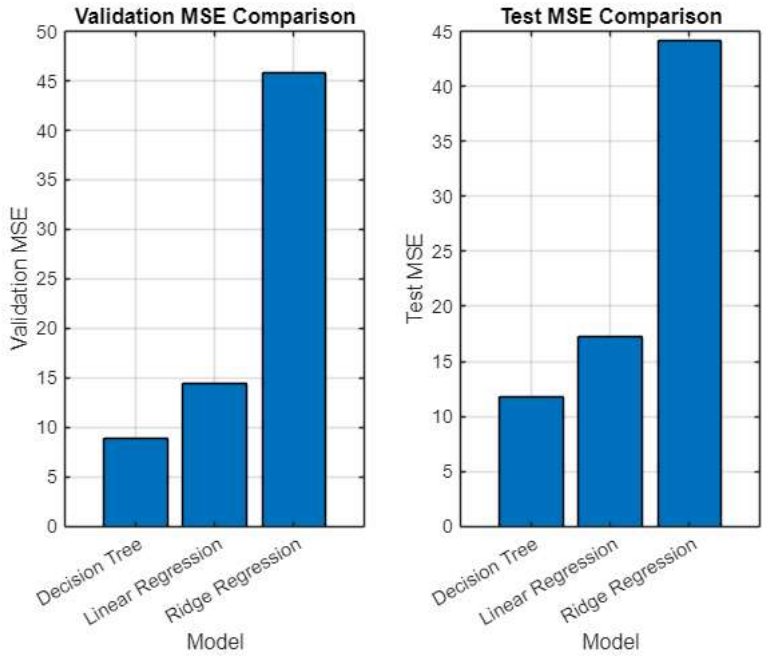
Variable	Coefficient
{'Intercept'}	21.371
{'crim'}	-0.0010078
{'zn'}	0.0030171
{'indus'}	-0.0011388
{'chas'}	1.0221e-05
{'nox'}	-4.6384e-06
{'rm'}	0.00036369
{'age'}	-0.0025564
{'dis'}	0.00034368
{'rad'}	-0.00084003
{'tax'}	-0.01822
{'ptratio'}	0.00011561

Model Evaluation and Comparison

```
% Collect performance metrics for comparison
models = {'Linear Regression', 'Decision Tree', 'Ridge Regression'};
validation_mse_values = [validation_mse_linear, best_validation_mse_tree, best_validation_mse_ridge];
test_mse_values = [test_mse_linear, test_mse_tree, test_mse_ridge];

% Display the results in a table
T = table(models', validation_mse_values', test_mse_values', 'VariableNames', {'Model', 'Validation MSE', 'Test MSE'});
disp(T);
```

Model	Validation MSE	Test MSE
{'Linear Regression'}	14.424	17.281
{'Decision Tree' }	8.9238	11.769
{'Ridge Regression' }	45.825	44.11

```
% Create the bar graph for comparison
figure;
subplot(1, 2, 1);
bar(categorical(models), validation_mse_values);
xlabel('Model');
ylabel('Validation MSE');
title('Validation MSE Comparison');
grid on;

subplot(1, 2, 2);
bar(categorical(models), test_mse_values);
xlabel('Model');
ylabel('Test MSE');
title('Test MSE Comparison');
grid on;
```



Clearing Matlab

```
% Clearing MATLAB
clear all; clc; close all;
```

Setting Up Data

```
% Importing dataset
% Import data from text file
BostonHousing = readtable("C:\Users\ajryt\OneDrive\Documents\MATLAB\Business Analytics\BostonHousing.csv");

% Display results
BostonHousing
```

BostonHousing = 506×14 table

	crim	zn	indus	chas	nox	rm	age
1	0.0063	18	2.3100	0	0.5380	6.5750	65.20
2	0.0273	0	7.0700	0	0.4690	6.4210	78.90
3	0.0273	0	7.0700	0	0.4690	7.1850	61.10
4	0.0324	0	2.1800	0	0.4580	6.9980	45.80
5	0.0691	0	2.1800	0	0.4580	7.1470	54.20
6	0.0299	0	2.1800	0	0.4580	6.4300	58.70
7	0.0883	12.5000	7.8700	0	0.5240	6.0120	66.60
8	0.1446	12.5000	7.8700	0	0.5240	6.1720	96.10
9	0.2112	12.5000	7.8700	0	0.5240	5.6310	1
10	0.1700	12.5000	7.8700	0	0.5240	6.0040	85.90
11	0.2249	12.5000	7.8700	0	0.5240	6.3770	94.30
12	0.1175	12.5000	7.8700	0	0.5240	6.0090	82.90
13	0.0938	12.5000	7.8700	0	0.5240	5.8890	
14	0.6298	0	8.1400	0	0.5380	5.9490	61.80
15	0.6380	0	8.1400	0	0.5380	6.0960	84.50
16	0.6274	0	8.1400	0	0.5380	5.8340	56.50
17	1.0539	0	8.1400	0	0.5380	5.9350	29.30
18	0.7842	0	8.1400	0	0.5380	5.9900	81.70
19	0.8027	0	8.1400	0	0.5380	5.4560	36.60
20	0.7258	0	8.1400	0	0.5380	5.7270	69.50
21	1.2518	0	8.1400	0	0.5380	5.5700	98.10
22	0.8520	0	8.1400	0	0.5380	5.9650	89.20
23	1.2325	0	8.1400	0	0.5380	6.1420	91.70
24	0.9884	0	8.1400	0	0.5380	5.8130	1
25	0.7503	0	8.1400	0	0.5380	5.9240	94.10
26	0.8405	0	8.1400	0	0.5380	5.5990	85.70
27	0.6719	0	8.1400	0	0.5380	5.8130	90.30
28	0.9558	0	8.1400	0	0.5380	6.0470	88.80
29	0.7730	0	8.1400	0	0.5380	6.4950	94.40
30	1.0025	0	8.1400	0	0.5380	6.6740	87.30
31	1.1308	0	8.1400	0	0.5380	5.7130	94.10
32	1.3547	0	8.1400	0	0.5380	6.0720	1
33	1.3880	0	8.1400	0	0.5380	5.9500	
34	1.1517	0	8.1400	0	0.5380	5.7010	
35	1.6128	0	8.1400	0	0.5380	6.0960	96.90
36	0.0642	0	5.9600	0	0.4990	5.9330	68.20
37	0.0974	0	5.9600	0	0.4990	5.8410	61.40
38	0.0801	0	5.9600	0	0.4990	5.8500	41.50
39	0.1751	0	5.9600	0	0.4990	5.9660	30.20
40	0.0276	75	2.9500	0	0.4280	6.5950	21.80
41	0.0336	75	2.9500	0	0.4280	7.0240	15.80
42	0.1274	0	6.9100	0	0.4480	6.7700	2.90
43	0.1415	0	6.9100	0	0.4480	6.1690	6.60
44	0.1594	0	6.9100	0	0.4480	6.2110	6.50
45	0.1227	0	6.9100	0	0.4480	6.0690	
46	0.1714	0	6.9100	0	0.4480	5.6820	33.80
47	0.1884	0	6.9100	0	0.4480	5.7860	33.30
48	0.2293	0	6.9100	0	0.4480	6.0300	85.50
49	0.2539	0	6.9100	0	0.4480	5.3990	95.30
50	0.2198	0	6.9100	0	0.4480	5.6020	
51	0.0887	21	5.6400	0	0.4390	5.9630	45.70
52	0.0434	21	5.6400	0	0.4390	6.1150	
53	0.0536	21	5.6400	0	0.4390	6.5110	21.10
54	0.0498	21	5.6400	0	0.4390	5.9980	21.40
55	0.0136	75	4	0	0.4100	5.8880	47.60
56	0.0131	90	1.2200	0	0.4030	7.2490	21.90
57	0.0205	85	0.7400	0	0.4100	6.3830	35.70
58	0.0143	100	1.3200	0	0.4110	6.8160	40.50

59	0.1545	25	5.1300	0	0.4530	6.1450	29.20
60	0.1033	25	5.1300	0	0.4530	5.9270	47.20
61	0.1493	25	5.1300	0	0.4530	5.7410	66.20
62	0.1717	25	5.1300	0	0.4530	5.9660	93.40
63	0.1103	25	5.1300	0	0.4530	6.4560	67.80
64	0.1265	25	5.1300	0	0.4530	6.7620	43.40
65	0.0195	17.5000	1.3800	0	0.4161	7.1040	59.50
66	0.0358	80	3.3700	0	0.3980	6.2900	17.80
67	0.0438	80	3.3700	0	0.3980	5.7870	31.10
68	0.0579	12.5000	6.0700	0	0.4090	5.8780	21.40
69	0.1355	12.5000	6.0700	0	0.4090	5.5940	36.80
70	0.1282	12.5000	6.0700	0	0.4090	5.8850	
71	0.0883	0	10.8100	0	0.4130	6.4170	6.60
72	0.1588	0	10.8100	0	0.4130	5.9610	17.50
73	0.0916	0	10.8100	0	0.4130	6.0650	7.80
74	0.1954	0	10.8100	0	0.4130	6.2450	6.20
75	0.0790	0	12.8300	0	0.4370	6.2730	
76	0.0951	0	12.8300	0	0.4370	6.2860	
77	0.1015	0	12.8300	0	0.4370	6.2790	74.50
78	0.0871	0	12.8300	0	0.4370	6.1400	45.80
79	0.0565	0	12.8300	0	0.4370	6.2320	53.70
80	0.0839	0	12.8300	0	0.4370	5.8740	36.60
81	0.0411	25	4.8600	0	0.4260	6.7270	33.50
82	0.0446	25	4.8600	0	0.4260	6.6190	70.40
83	0.0366	25	4.8600	0	0.4260	6.3020	32.20
84	0.0355	25	4.8600	0	0.4260	6.1670	46.70
85	0.0506	0	4.4900	0	0.4490	6.3890	
86	0.0573	0	4.4900	0	0.4490	6.6300	56.10
87	0.0519	0	4.4900	0	0.4490	6.0150	45.10
88	0.0715	0	4.4900	0	0.4490	6.1210	56.80
89	0.0566	0	3.4100	0	0.4890	7.0070	86.30
90	0.0530	0	3.4100	0	0.4890	7.0790	63.10
91	0.0468	0	3.4100	0	0.4890	6.4170	66.10
92	0.0393	0	3.4100	0	0.4890	6.4050	73.90
93	0.0420	28	15.0400	0	0.4640	6.4420	53.60
94	0.0288	28	15.0400	0	0.4640	6.2110	28.90
95	0.0429	28	15.0400	0	0.4640	6.2490	77.30
96	0.1220	0	2.8900	0	0.4450	6.6250	57.80
97	0.1150	0	2.8900	0	0.4450	6.1630	69.60
98	0.1208	0	2.8900	0	0.4450	8.0690	
99	0.0819	0	2.8900	0	0.4450	7.8200	36.90
100	0.0686	0	2.8900	0	0.4450	7.4160	62.50

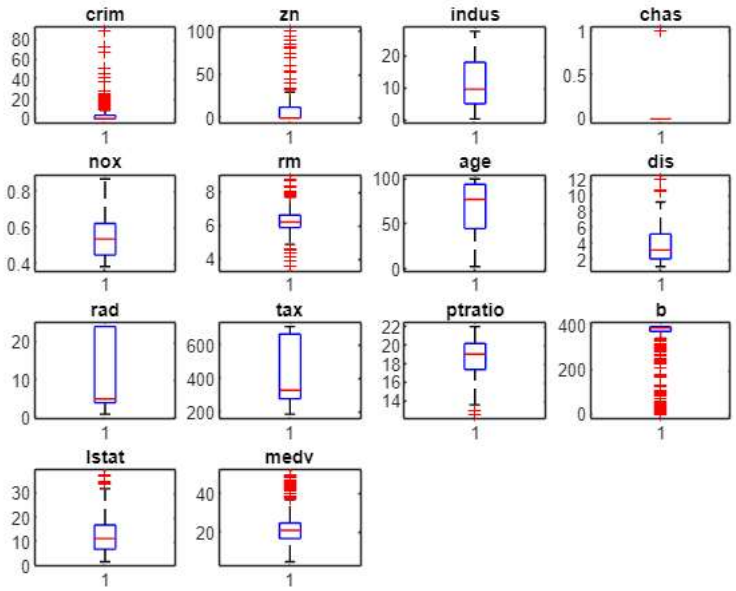
```
data = table2array(BostonHousing);

 [Observations,Variables] = size(BostonHousing)
```

```
Observations = 506
Variables = 14
```

```
 % Set the random number generator seed for reproducibility
seed = 42; % You can choose any integer value
rng(seed);
```

2.2 Data Understanding and Data Preparation

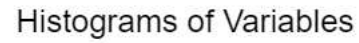
```
% Creating boxplot
figure;

% Loop through each variable
for i = 1:Variables
    subplot(4, 4, i); % Create subplots in a 4x4 grid
    boxplot(data(:,i));
    title(BostonHousing.Properties.VariableNames(i));
end
```

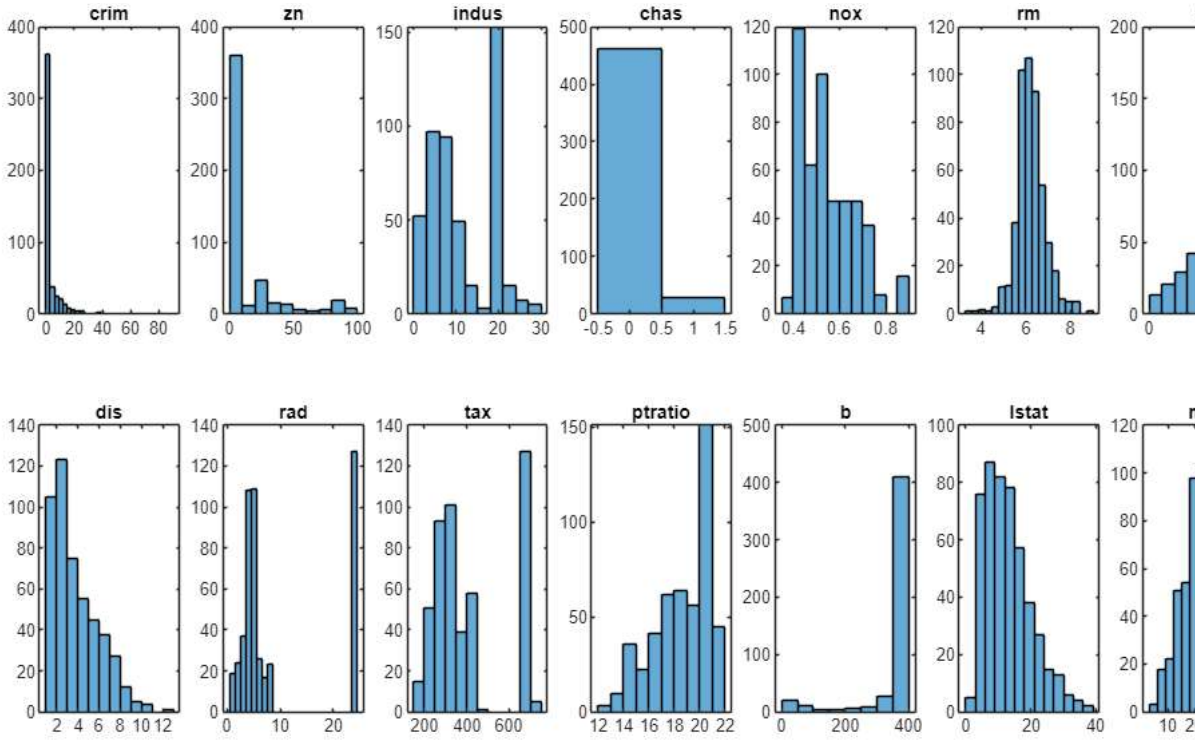



```
for i = 1:Variables
    q1 = quantile(data(:,i), 0.25);
    q3 = quantile(data(:,i), 0.75);
    irq = q3 - q1;
    % Find outliers
    outliers = data(:,i) < (q1 - 1.5 * irq) | data(:,i) > (q3 + 1.5 * irq);
    perc = sum(outliers) * 100.0 / numel(data(:,i));
    fprintf('Variable %s outliers = %.2f%%\n', BostonHousing.Properties.VariableNames{i}, perc);
end
```

```
Variable crim outliers = 13.04%
Variable zn outliers = 13.44%
Variable indus outliers = 0.00%
Variable chas outliers = 6.92%
Variable nox outliers = 0.00%
Variable rm outliers = 5.93%
Variable age outliers = 0.00%
Variable dis outliers = 0.99%
Variable rad outliers = 0.00%
Variable tax outliers = 0.00%
Variable ptratio outliers = 2.96%
Variable b outliers = 15.02%
Variable lstat outliers = 1.19%
Variable medv outliers = 7.31%
```

```
% Find rows where 'MEDV' column value is less than 50.0 to remove outliners
data = data(data(:,end) < 50.0, :);

% Plotting distribution of data'
fig = figure;
set(fig, 'Position', [0, 0, 1200, 600]); % Set figure size

num_cols = 7;
num_rows = ceil(size(data, 2) / num_cols);

for i = 1:Variables
    subplot(num_rows, num_cols, i);
    histogram(data(:, i));
    title(BostonHousing.Properties.VariableNames(i));
end

sgtitle('Histograms of Variables');

% Adjust layout
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
```






```
% Compute mean, median, minimum, maximum, and quartiles for each variable
means = mean(data)
```

```
means = 1×14
    3.6432   11.1122   11.1131    0.0592    0.5543    6.2455   68.2790    3.8345    9.5143  408.(
```

```
medians = median(data)
```

```
medians = 1×14
    0.2475         0    9.6900         0    0.5380    6.1850   76.8000    3.2759    5.0000  330.(
```

```
mins = min(data)
```

```
mins = 1×14
    0.0063         0    0.7400         0    0.3850    3.5610    2.9000    1.1370    1.0000  187.(
```

```
maxs = max(data)
```

```
maxs = 1×14
   88.9762  100.0000   27.7400    1.0000    0.8710    8.7800  100.0000   12.1265   24.0000  711.(
```

```
q1 = quantile(data, 0.25)
```

```
q1 = 1×14
    0.0820         0    5.1900         0    0.4490    5.8800   44.4000    2.1107    4.0000  280.(
```

```
q3 = quantile(data, 0.75)
```

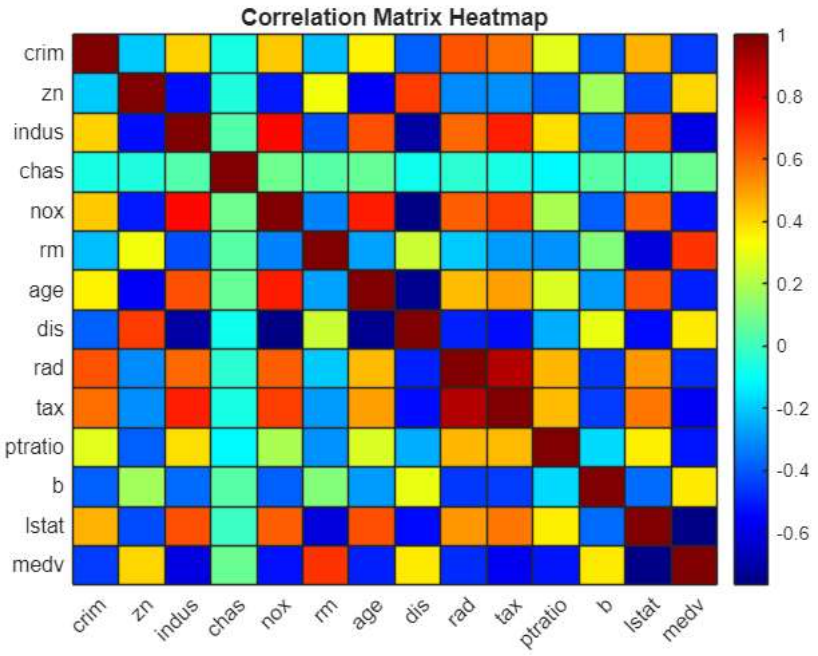
```
q3 = 1×14
    3.6737   12.5000   18.1000         0    0.6240    6.5790   93.9000    5.2146   24.0000  666.(
```

```
% Calculate and display the correlation matrix
corrMatrix = corr(data);
disp(corrMatrix);
```

```
    1.0000   -0.1991    0.4081   -0.0642    0.4205   -0.2193    0.3538   -0.3822    0.6274
   -0.1991    1.0000   -0.5271   -0.0539   -0.5121    0.3105   -0.5632    0.6732   -0.3077
    0.4081   -0.5271    1.0000    0.0358    0.7652   -0.4124    0.6380   -0.7103    0.5961
   -0.0642   -0.0539    0.0358    1.0000    0.0856    0.0450    0.0712   -0.0777   -0.0328
    0.4205   -0.5121    0.7652    0.0856    1.0000   -0.3226    0.7277   -0.7681    0.6122
   -0.2193    0.3105   -0.4124    0.0450   -0.3226    1.0000   -0.2685    0.2458   -0.1958
    0.3538   -0.5632    0.6380    0.0712    0.7277   -0.2685    1.0000   -0.7430    0.4519
   -0.3822    0.6732   -0.7103   -0.0777   -0.7681    0.2458   -0.7430    1.0000   -0.4919
    0.6274   -0.3077    0.5961   -0.0328    0.6122   -0.1958    0.4519   -0.4919    1.0000
    0.5837   -0.3029    0.7177   -0.0677    0.6674   -0.2820    0.4997   -0.5320    0.9090
    0.2871   -0.3818    0.3877   -0.1168    0.1884   -0.2933    0.2685   -0.2468    0.4560
   -0.3845    0.1761   -0.3634    0.0417   -0.3831    0.1192   -0.2790    0.2994   -0.4515
    0.4618   -0.4221    0.6365   -0.0065    0.6124   -0.6104    0.6379   -0.5365    0.5102
   -0.4501    0.4046   -0.6000    0.0748   -0.5245    0.6866   -0.4929    0.3688   -0.4763
```

```
% Create the heatmap
figure;
h = heatmap(BostonHousing.Properties.VariableNames(1:end), ... % x-axis
            BostonHousing.Properties.VariableNames(1:end), ... % y-axis
```

```
            corrMatrix, ... % Correlation matrix data
            'Colormap', jet, ...
            'ColorbarVisible', 'on');
title('Correlation Matrix Heatmap');
```



Model Development

```
BostonHousing = BostonHousing(BostonHousing.medv < 50.0, :);
% Check for missing values in the table
missingValues = ismissing(BostonHousing);
% Display missing values summary
missingSummary = sum(missingValues);
for i = 1:length(missingSummary)
    fprintf('%s: %d\n', BostonHousing.Properties.VariableNames{i}, missingSummary(i));
end
```

```
crim: 0
zn: 0
indus: 0
chas: 0
nox: 0
rm: 0
age: 0
dis: 0
rad: 0
tax: 0
ptratio: 0
b: 0
lstat: 0
medv: 0
```

```
% Data standardization
X = BostonHousing(:, 1:end-1); % Explanatory variables
X = table2array(X);
Y = BostonHousing.medv; % Dependent variable
% Split data into 70% training, 15% validation, and 15% test sets
cv = cvpartition(size(X, 1), 'HoldOut', 0.3);
idx = cv.test;
% Separate into training and test data
XTrain = X(~idx, :);
YTrain = Y(~idx);
XTest = X(idx, :);
YTest = Y(idx);
% Further split the training data into training and validation sets
cv2 = cvpartition(size(XTrain, 1), 'HoldOut', 0.1765); % 15% of 70% is 10.5%
idx2 = cv2.test;
XValidation = XTrain(idx2, :);
YValidation = YTrain(idx2);
XTrain = XTrain(~idx2, :);
YTrain = YTrain(~idx2);
```

Linear Regression

```
%Train Linear Regression Model
linearModel = fitlm(XTrain, YTrain);

% Validate Linear Regression Model
YValidationPredLinear = predict(linearModel, XValidation);
validation_mse_linear = mean((YValidation - YValidationPredLinear).^2);

% Test Linear Regression Model
YTestPredLinear = predict(linearModel, XTest);
test_mse_linear = mean((YTest - YTestPredLinear).^2);
% Extract coefficients
coefficients = linearModel.Coefficients.Estimate;
% Extract variable names (including intercept)
variableNames = ['(Intercept)'; BostonHousing.Properties.VariableNames(1:end-1)'];
coeffTable = table(variableNames, coefficients, 'VariableNames', {'Variable', 'Coefficient'});
disp(coeffTable);
```

Variable	Coefficient
{'(Intercept)'}	31.596
{'crim' }	-0.016629
{'zn' }	0.030146
{'indus' }	-0.014289
{'chas' }	0.41452
{'nox' }	-11.699
{'rm' }	3.4906
{'age' }	-0.0087401
{'dis' }	-1.1091
{'rad' }	0.27224
{'tax' }	-0.015216
{'ptratio' }	-0.86599

```
%Let's evaluate the model's performance from test and validation sets
disp(['Test MSE for Linear Regression: ', num2str(test_mse_linear)]);
```

```
Test MSE for Linear Regression: 17.2812
```

```
disp(['Validation MSE for Linear Regression: ', num2str(validation_mse_linear)]);
```

```
Validation MSE for Linear Regression: 14.4236
```

Decision Tree Regression

```
% Define parameter grid
maxNumSplits_values = [5, 10, 20, 50, 100];
minLeafSize_values = [1, 5, 10, 20];
best_maxNumSplits = 0;
best_minLeafSize = 0;
best_validation_mse_tree = inf;

% Manual Grid Search for Decision Tree
for maxNumSplits = maxNumSplits_values
    for minLeafSize = minLeafSize_values
        % Train Decision Tree Model
        tree_model = fitrtree(XTrain, YTrain, 'MaxNumSplits', maxNumSplits, 'MinLeafSize', minLeafSize);

        % Validate Decision Tree Model
        YValidationPred = predict(tree_model, XValidation);
        validation_mse = mean((YValidation - YValidationPred).^2);

        % Update best parameters if current setting has lower validation MSE
        if validation_mse < best_validation_mse_tree
            best_validation_mse_tree = validation_mse;
            best_maxNumSplits = maxNumSplits;
            best_minLeafSize = minLeafSize;
        end
    end
end

% Train final Decision Tree model with best parameters
final_tree_model = fitrtree(XTrain, YTrain, 'MaxNumSplits', best_maxNumSplits, 'MinLeafSize', best_minLeafSize);

% Test Decision Tree Model
YTestPred = predict(final_tree_model, XTest);
test_mse_tree = mean((YTest - YTestPred).^2);

disp(['Best MaxNumSplits: ', num2str(best_maxNumSplits)]);
```

```
Best MaxNumSplits: 20
```

```
disp(['Best MinLeafSize: ', num2str(best_minLeafSize)]);
```

```
Best MinLeafSize: 5
```

```
disp(['Validation MSE for best Decision Tree model: ', num2str(best_validation_mse_tree)]);
```

```
Validation MSE for best Decision Tree model: 8.9238
```

```
disp(['Test MSE for best Decision Tree model: ', num2str(test_mse_tree)]);
```

```
Test MSE for best Decision Tree model: 11.7688
```

```
% Extract variable importance
variableImportance = predictorImportance(final_tree_model);
% Extract variable names (excluding the dependent variable 'medv')
variableNames = BostonHousing.Properties.VariableNames(1:end-1)';
%Let's create a table of variable importance
variableImportanceTable = table(variableNames, variableImportance', ...
        'VariableNames', {'Variable', 'Importance'});
disp(variableImportanceTable);
```

```
      Variable       Importance
    _____________    __________

    {'crim'   }      0.026141
    {'zn'     }            0
    {'indus'  }            0
    {'chas'   }            0
    {'nox'    }      0.033737
    {'rm'     }       0.52146
    {'age'    }            0
    {'dis'    }       0.01375
    {'rad'    }            0
    {'tax'    }      0.048159
    {'ptratio'}      0.031913
    {'b'      }      0.010883
```

Ridge Regression

```
% Range of lambda values for Ridge Regression
lambda_values = logspace(-4, 4, 50); % 50 values between 10^-4 and 10^4
best_lambda = 0;
best_validation_mse_ridge = inf;

% Manual Grid Search for Ridge Regression
for lambda = lambda_values
    % Train Ridge Regression Model
    ridge_model = fitrlinear(XTrain, YTrain, 'Learner', 'leastsquares', 'Regularization', 'ridge', 'Lambda', lambda);

    % Validate Ridge Regression Model
    YValidationPred = predict(ridge_model, XValidation);
    validation_mse = mean((YValidation - YValidationPred).^2);

    % Update best lambda if current lambda has lower validation MSE
    if validation_mse < best_validation_mse_ridge
        best_validation_mse_ridge = validation_mse;
        best_lambda = lambda;
    end
end

% Train final Ridge Regression model with best lambda
final_ridge_model = fitrlinear(XTrain, YTrain, 'Learner', 'leastsquares', 'Regularization', 'ridge', 'Lambda', best_lambda);

% Test Ridge Regression Model
YTestPred = predict(final_ridge_model, XTest);
test_mse_ridge = mean((YTest - YTestPred).^2);

disp(['Best Lambda: ', num2str(best_lambda)]);
```

```
Best Lambda: 0.0001
```

```
disp(['Validation MSE for Ridge Regression model: ', num2str(best_validation_mse_ridge)]);
```

```
Validation MSE for Ridge Regression model: 45.8249
```

```
disp(['Test MSE for Ridge Regression model: ', num2str(test_mse_ridge)]);
```

```
Test MSE for Ridge Regression model: 44.1101
```

```
% Extract coefficients from Ridge Regression model
coefficients_ridge = final_ridge_model.Beta;
intercept_ridge = final_ridge_model.Bias;

% Combine the intercept and coefficients into one array
coefficients_ridge = [intercept_ridge; coefficients_ridge];

% Extract feature names including intercept
VariableNames = [{'Intercept'}, BostonHousing.Properties.VariableNames(1:end-1)];

% Create a table for better readability
coeffTable_ridge = table(VariableNames(:), coefficients_ridge, ...
    'VariableNames', {'Variable', 'Coefficient'});
disp(coeffTable_ridge);
```

Variable	Coefficient
{'Intercept'}	21.371
{'crim'}	-0.0010078
{'zn'}	0.0030171
{'indus'}	-0.0011388
{'chas'}	1.0221e-05
{'nox'}	-4.6384e-06
{'rm'}	0.00036369
{'age'}	-0.0025564
{'dis'}	0.00034368
{'rad'}	-0.00084003
{'tax'}	-0.01822
{'ptratio'}	0.00011561

Model Evaluation and Comparison

```
% Collect performance metrics for comparison
models = {'Linear Regression', 'Decision Tree', 'Ridge Regression'};
validation_mse_values = [validation_mse_linear, best_validation_mse_tree, best_validation_mse_ridge];
test_mse_values = [test_mse_linear, test_mse_tree, test_mse_ridge];

% Display the results in a table
T = table(models', validation_mse_values', test_mse_values', 'VariableNames', {'Model', 'Validation MSE', 'Test MSE'});
disp(T);
```

Model	Validation MSE	Test MSE
{'Linear Regression'}	14.424	17.281
{'Decision Tree' }	8.9238	11.769
{'Ridge Regression' }	45.825	44.11

```
% Create the bar graph for comparison
figure;
subplot(1, 2, 1);
bar(categorical(models), validation_mse_values);
xlabel('Model');
ylabel('Validation MSE');
title('Validation MSE Comparison');
grid on;

subplot(1, 2, 2);
bar(categorical(models), test_mse_values);
xlabel('Model');
ylabel('Test MSE');
title('Test MSE Comparison');
grid on;
```