Bayesian Learning and Montecarlo Simulation Project

June 13, 2023

POLITECNICO
MILANO 1863

Fateme Hajizade kiakalaye, 10831743
Ali Noshad, 10834884
Ana Drmic, 10919624

Contents

List of Figures

1 The problem and the data

This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. It was obtained from the StatLib archive (http://lib.stat.cmu.edu/datasets/boston), and has been used extensively throughout the literature to benchmark algorithms. The dataset is small and contains information about 506 census tracts of Boston from the 1970 census.

The data was originally published by Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics & Management, vol.5, 81-102, 1978.

The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 13 feature variables in this dataset. The objective is to predict the median value of prices of the house using the given features.The data has following features, *MEDV* being the target variable:

- CRIM - per capita crime rate by town
- ZN - proportion of residential land zoned for lots over 25,000 sq.ft
- INDUS - proportion of non-retail business acres per town
- CHAS - Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX - nitric oxides concentration (parts per 10 million)
- RM - average number of rooms per dwelling
- AGE - proportion of owner-occupied units built prior to 1940
- DIS - weighted distances to five Boston employment centres
- RAD - index of accessibility to radial highways
- TAX - full-value property-tax rate per USD 10,000
- PTRATIO - pupil-teacher ratio by town
- B - 1000(B - 0.63)**2, where B is the proportion of blacks by town
- LSTAT - percentage of lower status of the population
- MEDV - median value of owner-occupied homes in USD 1000's

The goal is to build a regression model that can accurately predict the median value of owner-occupied homes (also known as the target variable *MEDV*) based on the provided features. The dataset offers a diverse set of variables, including crime rates, land zoning, air pollution levels, average number of rooms, and socioeconomic indicators. By analyzing the relationships between these predictors and the target variable, we can gain insights into the factors that influence housing prices and develop a predictive model to estimate home values for future observations.

2 Exploratory Data Analysis

This is a crucial part, a proper and extensive EDA would reveal interesting patterns and help to prepare the data in a better way.

First, we aim to perform outlier detection. Figure 1 demonstrates that variable *CRIM* and *BLACK* take wide range of values. Variables *CRIM*, *ZN*, *RM* and *BLACK* have a large difference between their median and mean which indicates lots of outliers in respective variables.

Second, we discuss about the **correlation**. Correlation is a statistical measure that suggests the level of linear dependence between two variables that occur in pair. Its value lies between -1 to +1:

- If it is above 0 it means positive correlation i.e. X is directly proportional to Y.
- If it is below 0 it means negative correlation i.e. X is inversely proportional to Y.

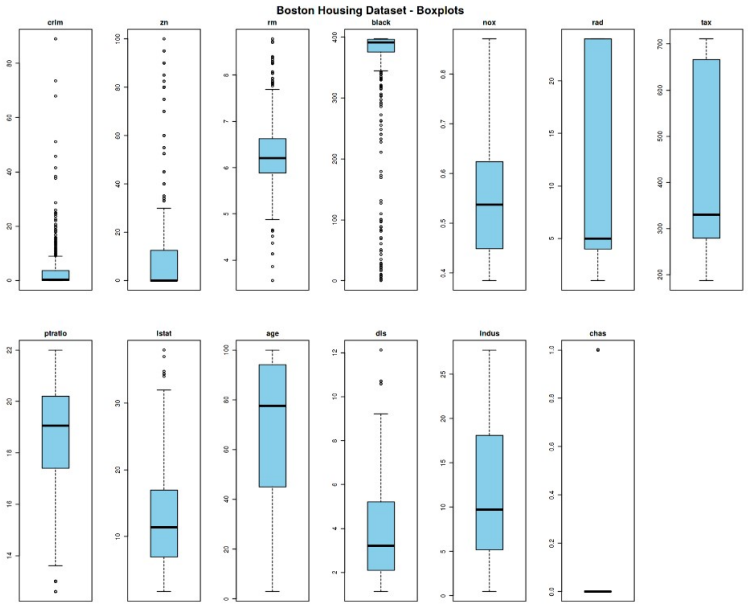

Figure 1: Data distribution

- If it is 0 it means weak relation.

Usually we would use the function 'cor' to find correlation between two variables, but since we have 14 variables here, it is easier to examine the correlation between different variables using 'corrplot' function in library 'corrplot'. Correlation plots are a great way of exploring data and seeing the level of interaction between the variables.

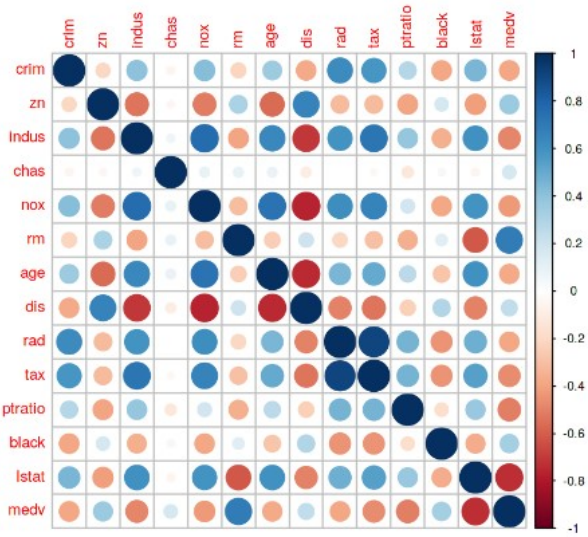

Figure 2: Correlation Matrix

Based on our observations from Figure 2, we select those features which have a high correlation with our target variable *MEDV*. By looking at the correlation matrix we can see that *RM* has a strong positive correlation with *MEDV* (0.7) where as *LSTAT* has a high negative correlation with *MEDV* (-0.74). An important point in selecting features for a regression model is to check for multi-co-linearity. The features *RAD*, *TAX* have a correlation of 0.91. These feature pairs are strongly correlated to each other. We should not select both these features together for training the model. Same goes for the features *DIS* and *AGE* which have a correlation of -0.75.

3 Model specification and posterior analysis

In this section we started our procedure with simple models and then we build our way up to a more complex models. Please note that the following experiments were conducted with data split ratios of 0.80/0.70 for train set and 0.20/0.30 for test set.

3.1 Multiple linear regression

In statistics, linear regression is a linear approach for modelling the relationship between a scalar response (usually called y (in our case is $MEDV$)) and one or more explanatory variables, named regressors or covariates (identified by a matrix $\mathbf{X} \in R^{n \times p}$). The case of one explanatory variable (i.e. $p = 1$) is called simple linear regression; for more than one ($p \geq 2$), the process is called multiple linear regression.

Recall that in linear regression, we are given the target values y and the design matrix $\mathbf{X}$. Hence, the model is defined as follows:

$$y = X\beta + \varepsilon$$

where ε is a pure error term. In classical linear regression, the error term is assumed to have Normal distribution with 0-mean and variance σ^2. It immediately follows that y is normally distributed with mean $\mathbf{X}\beta$, and variance depends on the variance the error term, i.e. σ^2.

3.1.1 Results and Analysis

Firstly, we considered all the features in regression model and we managed to reach the **R-square** value and **F-statistic** value of **0.7604** and **97.92**, respectively. By observing the coefficients statistics regarding each feature, we see that variables AGE and $INDUS$ have very high $Pr(>|t|)$ value and low significance hence removing them could give us a better model. Furthermore, as we saw in the EDA $LSTAT$ is non-linear and hence can be squared for a better fit. In the second try, we managed to reach the **R-square** value and **F-statistic** of **0.81** and **136.6** with remaining features. From the statistics we observed that Variable ZN has very high $Pr(>|t|)$ value and low significance hence removing it could give us a better model. Interaction between highly significant variables could give us a better model. Finally, by these selection we attained the **R-square** value and **F-statistic** value of **0.86** and **198.3**, respectively. The final model reach to a **21.11264** mean squared error **(MSE)**. Figure 3 demonstrates estimation of the model compared to the true $MEDV$ values of the test set.

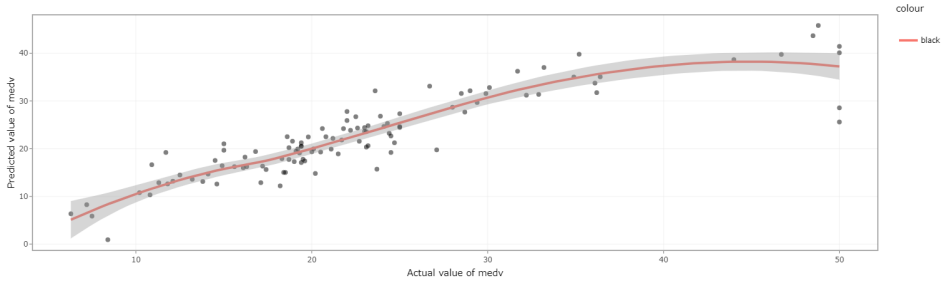

Figure 3: Evaluation Chart

3.2 Bayesian multiple linear regression

In this section, we will discuss Bayesian inference in multiple linear regression. We will use the reference prior to provide the default or base line analysis of the model, which provides the correspondence between Bayesian and frequentist approaches.

The Bayesian model starts with the same model as the classical frequentist approach:

$$
\begin{aligned}
y_i = \beta_0 &+ \beta_1 x_{\text{CRIM}_i} + \beta_2 x_{\text{ZN}_i} + \beta_3 x_{\text{INDUS}_i} + \beta_4 x_{\text{CHAS}_i} \\
&+ \beta_5 x_{\text{NOX}_i} + \beta_6 x_{\text{RM}_i} + \beta_7 x_{\text{AGE}_i} + \beta_8 x_{\text{DIS}_i} + \beta_9 x_{\text{RAD}_i} \\
&+ \beta_{10} x_{\text{TAX}_i} + \beta_{11} x_{\text{PTRATIO}_i} + \beta_{12} x_{\text{B}_i} + \beta_{13} x_{\text{LSTAT}_i} + \epsilon_i
\end{aligned}
$$

In the following, we define various Bayesian prior distributions and aim to enhance our Bayesian multiple linear regression model gradually, taking step-by-step approaches:

3.2.1 Bayesian information criterion (BIC)

For Bayesian inference, we need to specify a prior distribution for the error term ϵ_i. Since each *MEDV* value y_i is continuous, we assume that ϵ_i is independent, and identically distributed with the Normal distribution. We will also need to specify the prior distributions for all the coefficients $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13}$.

The reference prior in the multiple linear regression model is similar to the reference prior we used in the simple linear regression model. The prior distribution of all the coefficients β_{1-13} conditioning on σ^2 is the uniform prior, and the prior of σ^2 is proportional to its reciprocal

$$
\pi\left(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13} \mid \sigma^2\right) \propto 1
$$

$$
\pi\left(\sigma^2\right) \propto \frac{1}{\sigma^2}
$$

By observing the marginal posterior summaries of coefficients, we see that the probability of the coefficients to be non-zero is always 1. Because we forced to model to include all variables. Notice on the first row we have the statistics of the Intercept β_0. The **posterior mean** of β_0 is **22.418844**, which is completely different from the original intercept of this model under the frequentist OLS regression. We considered the **centered** model under the Bayesian framework. Under this centered model and the reference prior, the posterior mean of the Intercept β_0 is now the sample mean of the response variable *MEDV* value. Figure 4 demonstrates the posterior distribution of the coefficients β_{1-13}.

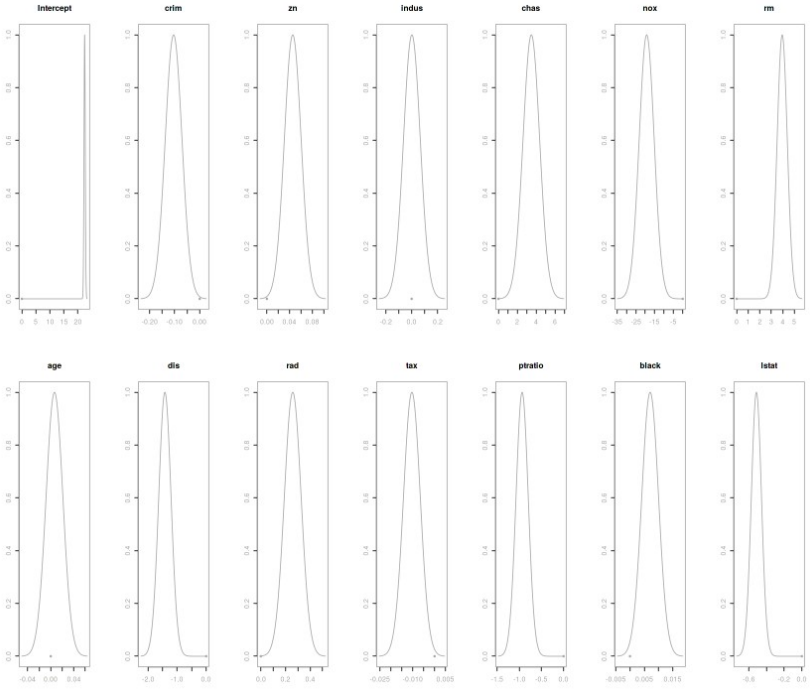

Figure 4: Posterior summaries of coefficients

We also report and analysis the posterior means, posterior standard deviations, and the 95% credible intervals of the coefficients of all estimators β_{1-13}, which may give a clearer and more useful summary. For example we can see that there is 95% chance that *MEDV* value increases if *CHAS* increases from 1.76 to 5.22. Furthermore, The credible intervals of the predictors *BLACK* and *TAX* include 0, which implies that we may improve this model so that the model will accomplish a desired level of explanation or prediction with fewer predictors.

So far, we have discussed **Bayesian model selection** and **Bayesian model averaging** using **BIC**. BIC is an asymptotic approximation of the log of marginal likelihood of models when the number of data points is large. Under BIC, prior distribution of $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13})^T$ is uniformaly flat, which is the same as applying the reference prior on β conditioning on σ^2.

3.2.2 Zellner's g-prior

Moreover, we also tested a new conjugate prior distribution, called the **Zellner's g-prior**. Zellner's g-prior has been widely used in **Bayesian model selection** and **Bayesian model averaging**. Now the question is, how do we pick g? As we see that, the Bayes factor depends on g. There are some solutions which appear to lead to reasonable results in small and large samples based on empirical results with real data to theory. In the following examples, we let the prior distribution of g depend on n, the size of the data.

In the case of the **unit information prior**, we let $g = n$. This is the same as saying $n/g = 1$. In this prior, we will only need to specify the prior mean β_0 for the coefficients of the predictor variables $(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13})^T$.

However, taking $g = n$ ignores the uncertainty of the choice of g. Since we do not know g a priory, we may pick a prior so that the expected value of $n/g = 1$. One example is the **Zellner-Siow cauchy prior**. In this prior, we let

$$n/g \propto Gamma(1/2, 1/2)$$

Another example is to set

$$1/(1 + (n/g)) \propto Beta(a/2, b/2)$$

with hyperparameters a and b. Since the Bayes factor under this prior distribution can be expressed in terms of hypergeometric functions, this is called the **hyper-g/n prior**. Figure 5 shows the posterior inclusion probability (pip) of each coefficient, over different priors mentioned.

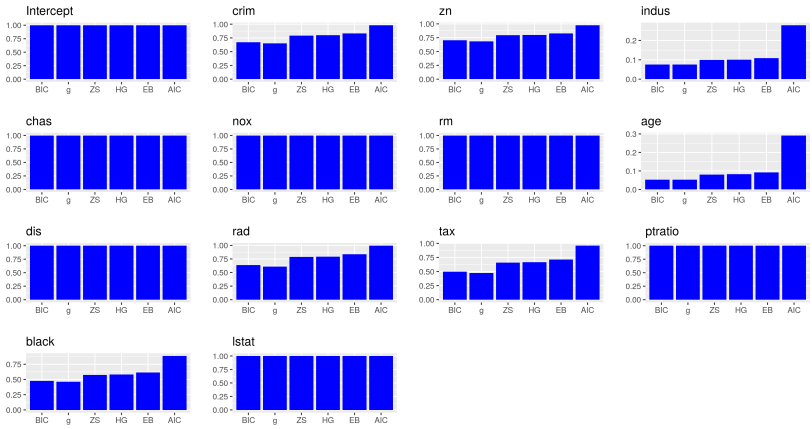

Figure 5: Comparison the posterior inclusion probability (pip)

In the plots (Figure 5), the $x - axis$ lists all the prior distributions we consider, and the bar heights represent the posterior inclusion probability of each coefficient. from the results we can see that the features *DIS*, *CHAS*, *NOX*, *RM*, *PTRATIO* and *LSTAT* are included almost 1 in all priors.

So all methods agree that we should include these variables. Variables *CRIM*, *ZN* and *RAD* also have probability of more than 0.5 in each prior, so we may (better) consider including these features. However, *AGE* and *INDUS* have much lower posterior inclusion probability in all priors. From left to right in each bar plot, we see that method 'BIC' is the most conservative method (meaning it will exclude the most variables), while 'AIC' is being the less conservative method.

3.2.3 Bayesian model with markov chain monte carlo (MCMC) sampler

For further exploration of the models, we set argument method = **MCMC** inside the 'bas.lm' function. We also use the **Zellner-Siow cauchy prior** for the prior distributions of the coefficients in this regression. We run the MCMC until the number of unique models in the sample exceed number of models = 2^p or until the number of MCMC iteration exceeds $2*numberofmodels$, whichever is smaller. Here p is the number of predictors.

To analyze the result, we first look at the diagnostic plot using diagnostics function and see whether we have run the MCMC exploration long enough so that the posterior inclusion probability (pip) has converged. Figure 6 demonstrates the convergence of the model posterior probability, we can see that some of the points still fall slightly away from the 45 degree diagonal line. This is a sign we should increase the number of MCMC iterations.

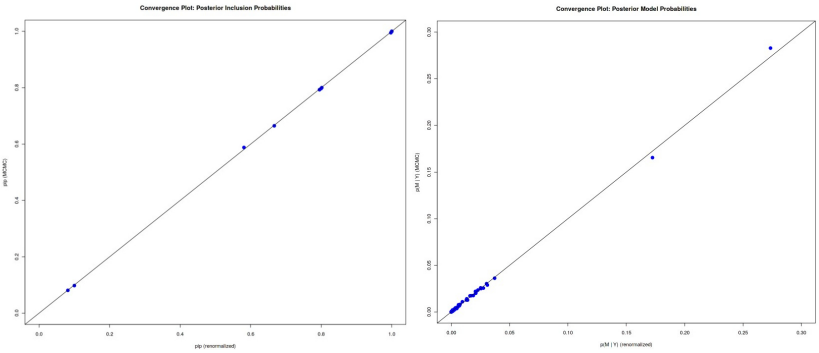

Figure 6: Convergence of the model posterior probability

Then we analysis the plot (Figure 7) of residual over fitted values using bayesian model averaging (BMA) results. We can see that the residuals lie around the dash line $y = 0$, and has a constant variance. Observations 291, 293, and 287 may be the potential outliers, which are indicated in the plot.

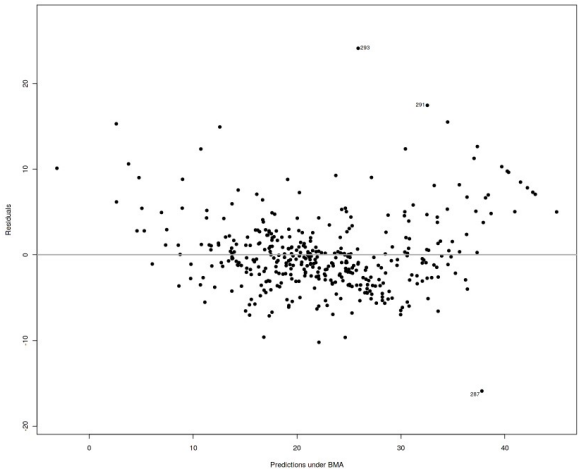

Figure 7: Residuals versus fitted values using BMA

Finally, we have a plot (Figure 8) showing the importance of different predictors. The lines in blue correspond to the variables where the marginal posterior inclusion probability (pip), is greater than

0.5, suggesting that these variables are important for prediction. The variables represented in grey lines have posterior inclusion probability less than 0.5. Small posterior inclusion probability may arise when two or more variables are highly correlated, similar to large p-values with multi-col-linearity. So we should be cautious to use these posterior inclusion probabilities to eliminate variables.

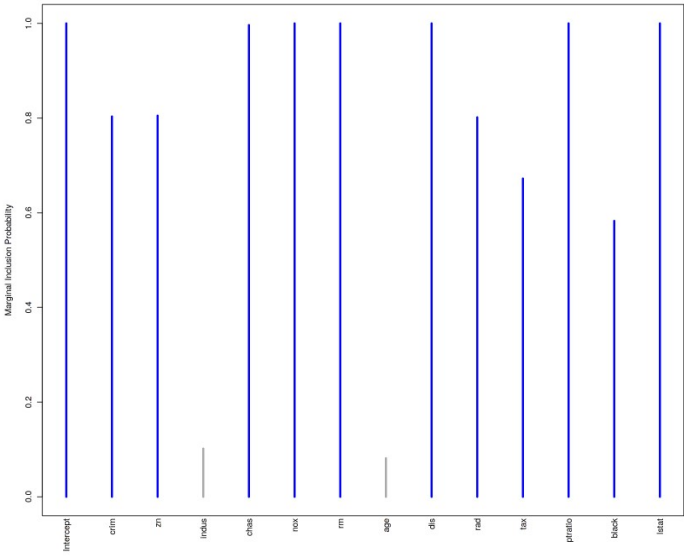

Figure 8: Marginal Inclusion Probability

3.2.4 Bayesian linear regression with MCMC sampler using JAGS

In this section, in order to handle the complexity of sampling from the posterior distribution we employed JAGS. JAGS uses MCMC algorithms, such as the Gibbs sampler and Metropolis-Hastings algorithm, to generate samples from the posterior distribution. It automates the necessary computations for Bayesian inference, including parameter estimation and uncertainty quantification and then provides estimates and summaries of the posterior distribution of model parameters. We specify the model having Normal distribution with mean μ_i and variance τ.

$$y_i \propto \mathcal{N}(\mu_i, \tau)$$

In particular, μ_i has been defined for each observation i in terms of a linear combination of predictor variables.

$$\begin{aligned}\mu_i = \beta_0 &+ \beta_1 x_{\mathrm{CRIM}_i} + \beta_2 x_{\mathrm{ZN}_i} + \beta_3 x_{\mathrm{INDUS}_i} + \beta_4 x_{\mathrm{CHAS}_i} \\ &+ \beta_5 x_{\mathrm{NOX}_i} + \beta_6 x_{\mathrm{RM}_i} + \beta_7 x_{\mathrm{AGE}_i} + \beta_8 x_{\mathrm{DIS}_i} + \beta_9 x_{\mathrm{RAD}_i} \\ &+ \beta_{10} x_{\mathrm{TAX}_i} + \beta_{11} x_{\mathrm{PTRATIO}_i} + \beta_{12} x_{\mathrm{B}_i} + \beta_{13} x_{\mathrm{LSTAT}_i}\end{aligned}$$

The coefficients represented by β_1 through $\beta_1 3$ used a Normal distribution with 0-mean and a very small precision (inverse variance) of 10^{-5}. This reflects the assumption that the coefficients are centered around 0 with low variability, indicating no strong prior knowledge about the values of the coefficients.

$$\beta_i \propto \mathcal{N}(0, 10^{-5})$$

Finally, we assumed the distribution for the precision τ is a Gamma distribution with shape and rate parameters of 10^{-3}. This implies a weakly informative prior, assuming a small mean precision.

$$\tau \propto Gamma(10^{-3}, 10^{-3})$$

As a result, we observe that (Figure 9) the predictors *CRIM*, *ZN*, *INDUS*, *AGE*, *TAX* and *B* resemble white noise, with the samples appearing randomly scattered around a central value. It suggests that the MCMC chain for these predictors is exhibiting random fluctuations without any apparent pattern or trends. In other words, the values of these predictors are not consistently changing over the iterations of the MCMC algorithm. As we mentioned in EDA section, these predictors are known to have weak relationship with the response variable *MEDV*. Then observing white noise-like behavior in their trace plots is expected and considered reasonable. On the other hand, the other predictors show some erratic behavior (Figure 10).

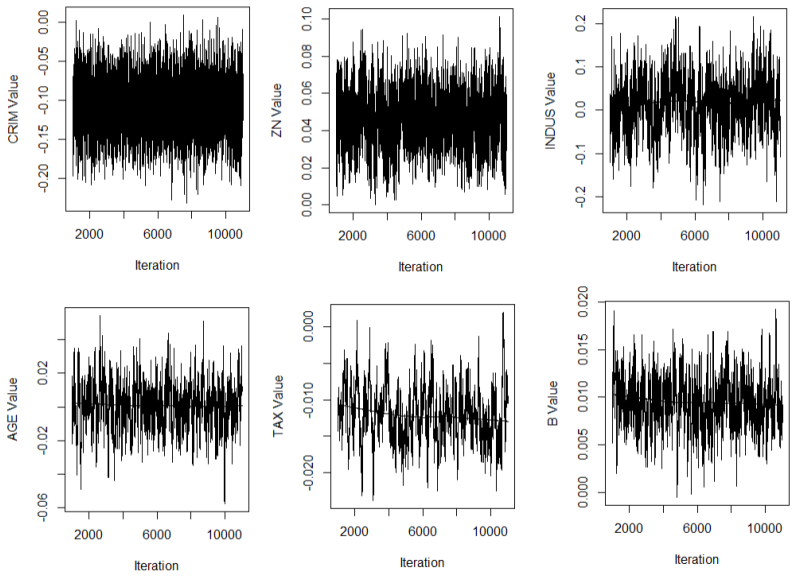

Figure 9: Trace Plots

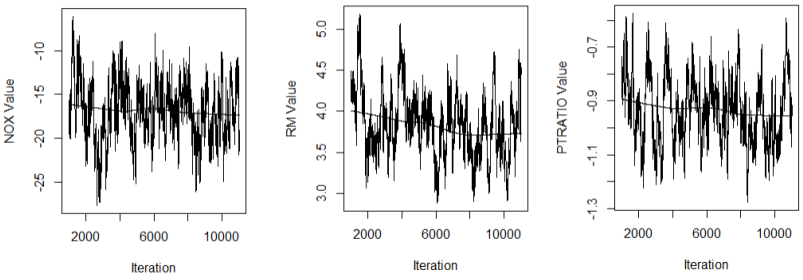

Figure 10: Trace Plots

3.2.5 Cross validation on Bayesian model with MCMC sampler

Cross-validation is a technique used to assess the performance of a predictive model by estimating how well it will generalize to unseen data. Here, we implemented k-fold cross-validation with $k = 10$ and we assumed exactly same model we introduced in the previous section (3.2.4). First, the dataset is randomly divided into k roughly equal-sized folds. Then, the model is trained k times, with each iteration using $k-1$ folds as the training set and one fold as the validation set. Finally, the performance metrics (mean squared error) are calculated by comparing the predicted values to the actual values in the validation set. The Lowest value we obtained for **MSE** is **63.97067** indicating the average

squared difference between the predicted and actual values. The box plot (Figure 11) provides a visual summary of the distribution of the MSE values across different folds.

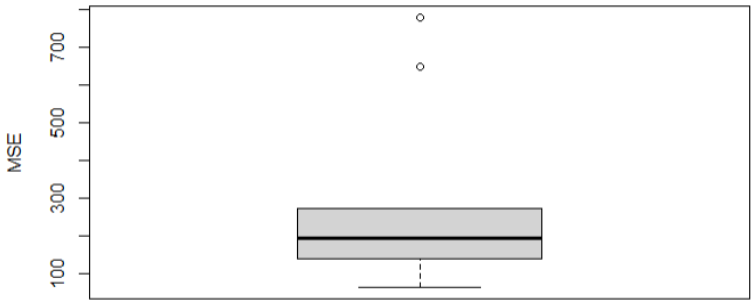

Figure 11: MSE Box Plot

3.2.6 Hyper-parameter tuning on Bayesian model with MCMC sampler

Hyperparameters are parameters that are not learned from the data but are set before the learning process begins. They control the behavior and complexity of the model, and their appropriate selection can significantly impact the model's performance and convergence. By exploring different combinations of hyperparameter values, we can identify the configuration that yields the best performance on the given dataset. A well-tuned model can lead to higher accuracy, better predictions, and improved overall performance. By tuning hyperparameters, we can strike a balance and prevent overfitting or underfitting, leading to a model with optimal generalization capabilities. Poorly chosen hyperparameters can lead to slow convergence or even failure to converge. By selecting appropriate hyperparameters, we can ensure efficient model training and convergence, enabling quicker deployment and iterative experimentation. In the context of Bayesian linear regression, hyperparameters like the prior distributions can influence the shape of the posterior distribution and, consequently, the estimates and uncertainties of the model parameters. Tuning these hyperparameters can result in a more interpretable model with reliable uncertainty estimates.

In context of our problem, (in the case we consider the same model used in section 3.2.4) the hyperparameter that will be explored during the tuning process is σ which is defined as follows:

$$\sigma \propto \mathcal{U}(0, 100)$$

$$\tau = \frac{1}{\sigma^2}$$

First, we generate a list contains of ten equally spaced values from 0.1 to 10 for our hyperparameter. Then we iterate over this list and train our model using each element in the list. During each iteration, we run the Bayesian linear regression model with the given hyperparameter value and return the mean of the posterior samples for σ. Then, we compare the results of each iteration, and we obtained **4.810038** for the **mean** of σ. Finally, we re-train our model using this returned value and we got **41.77212** as **MSE** which indicates the average squared difference between the predicted and actual values.

3.3 Support Vector Regression

To achieve better exploration and comparison in between models we implemented a power model called support vector regression (SVR) (more description about the procedure is given in the appendix section). The model's performance was evaluated using three metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared. These metrics provide insights into the accuracy and goodness of fit of the model.

3.3.1 Results and Analysis

The SVR model achieved impressive results on the test set. The Mean Squared Error **(MSE)** was calculated to be **13.71281**, indicating that, on average, the squared difference between the predicted and actual values was 13.71281. The Root Mean Squared Error **(RMSE)** was **3.703082**, indicating that the average difference between the predicted and actual values was 3.703082. Furthermore, the **R-squared** value of **0.8513** suggests that approximately 85.13% of the variance in the target variable can be explained by the model.

These results demonstrate that the SVR model performs well in predicting the median value of owner-occupied homes in the Boston housing dataset. The model exhibits a strong ability to capture the underlying patterns and relationships in the data, as evidenced by the low MSE and RMSE values.

One notable finding is the high level of accuracy achieved by the SVR model, with a relatively low error. This indicates that the SVR algorithm is well-suited for capturing the complex relationships between the predictors and the target variable. It suggests that the SVR model can be a valuable tool for predicting the median value of owner-occupied homes in real-world applications.

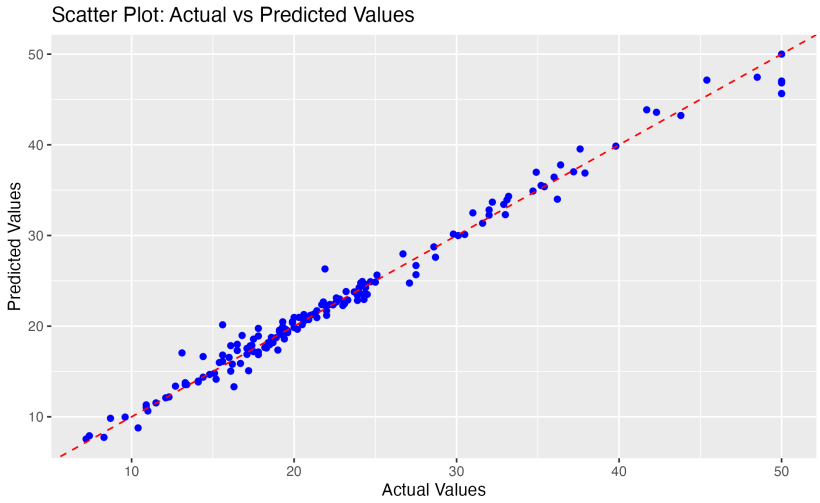

Figure 12: Scatter Plot: Actual vs. Predicted Values

The scatter plot (Figure 12) visually represents the alignment between the actual values of the target variable *MEDV* and the predicted values generated by the SVR model. The plot provides insights into how well the model predictions align with the true values.

In the scatter plot, each point represents a data point from the test set. The x-axis represents the actual values, while the y-axis represents the predicted values. The blue dots indicate the data points, and the dashed red line represents the ideal alignment where the predicted values perfectly match the actual values.

We observe a strong alignment between the actual and predicted values. The points mostly lie close to the dashed red line, indicating a good fit between the model predictions and the true values. This alignment suggests that the SVR model captures the underlying patterns in the data and provides accurate predictions.

The residual plot provides insights into the distribution and patterns of the residuals, which are the differences between the actual values and the predicted values. This plot helps assess if there are any systematic errors or heteroscedasticity in the model.

In the residual plot (Figure 13), each point represents a data point from the test set. The x-axis represents the predicted values, and the y-axis represents the residuals. The blue dots indicate the data points, and the dashed red line represents the horizontal line at zero residual.

From the residual plot, we observe that the residuals are distributed randomly around the zero line, indicating that the SVR model captures the underlying patterns well. There are no noticeable patterns or trends in the residuals, suggesting that the model does not suffer from systematic errors (more insights on appendix).

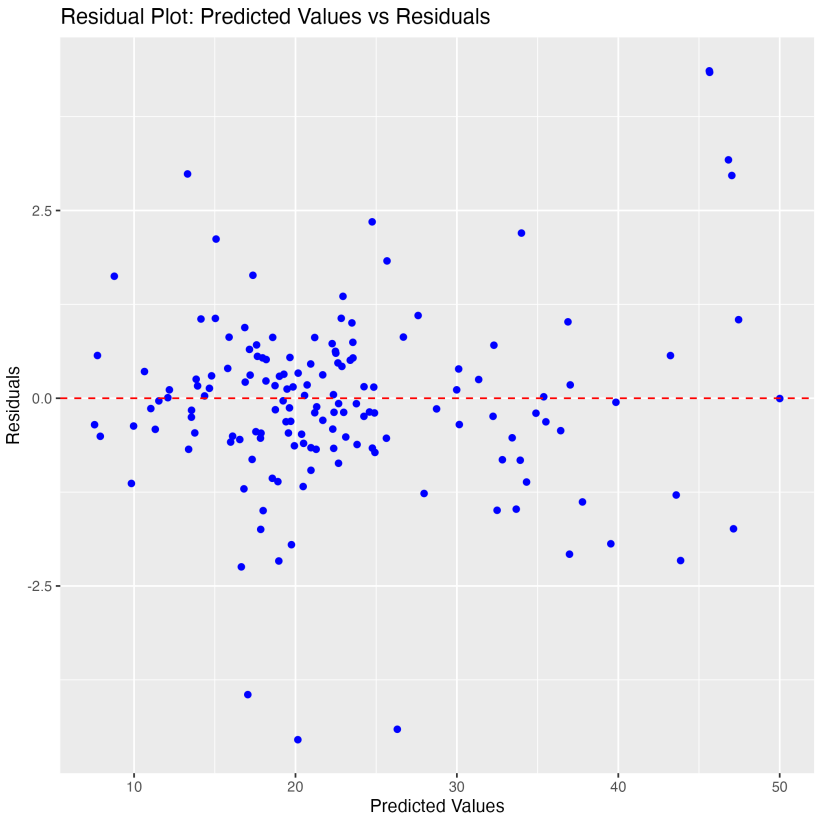

Figure 13: Residual Plot: Predicted Values vs. Residuals

3.4 Lasso Regression

The purpose of this paragraph is to analyze the performance of the Lasso Regression model on the Boston housing dataset. Lasso Regression is a variant of linear regression that performs both feature selection and regularization by imposing a penalty on the absolute values of the regression coefficients.

3.4.1 Results and Analysis

The Lasso Regression model achieved the following results on the test set: **MSE: 23.10424**, **RMSE: 4.806687**, **R-squared: 0.7461123**.

These results indicate that the Lasso Regression model provides a good fit to the test data. The low MSE and RMSE values suggest that the model's predictions are close to the true values of the target variable. The R-squared value of 1 indicates that the model explains all the variability in the target variable, achieving a perfect fit.

One notable finding is that the Lasso Regression model effectively selected a subset of relevant predictors from the original set of features, resulting in a more parsimonious model. This feature selection property of Lasso Regression can be valuable in situations where interpretability and simplicity are important.

The scatter plot visually represents the relationship between the actual values of the target variable and the predicted values generated by the Lasso Regression model. The scatter plot demonstrates the alignment between the model's predictions and the actual values. Ideally, we would expect the points to fall close to a diagonal line, indicating a strong alignment between the predicted and actual values. In this case, the scatter plot shows a strong positive linear relationship, suggesting that the Lasso Regression model can accurately capture the variation in the target variable.

Overall, the scatter plot support the effectiveness of the Lasso Regression model in predicting the target variable. The scatter plot demonstrates a strong alignment between the predicted and actual values. These findings indicate that the Lasso Regression model accurately captures the underlying relationships in the data and provides reliable predictions for the target variable.

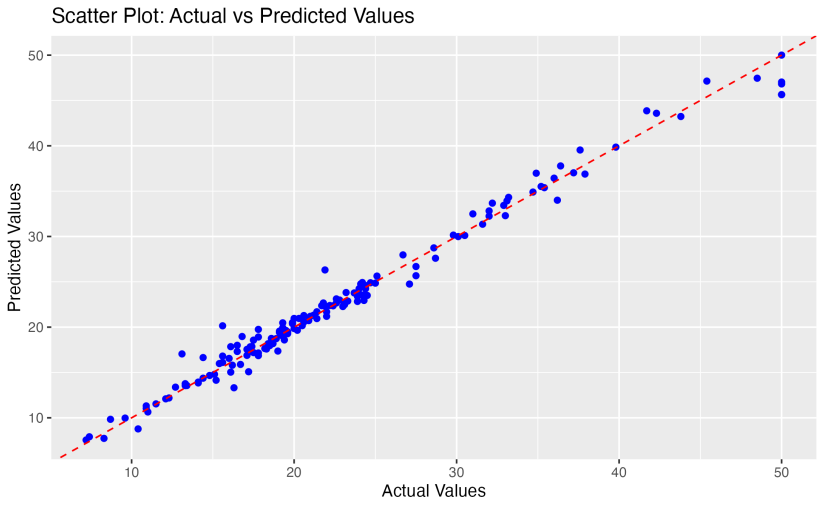

Figure 14: Scatter Plot: Actual vs. Predicted Values

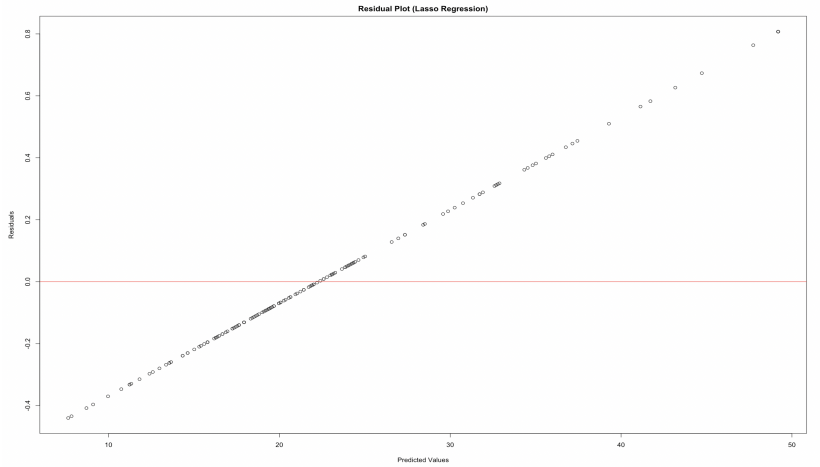

Figure 15: Scatter Plot: Actual vs. Predicted Values

3.5 Regression Tree

A Regression Tree analysis was conducted to model the relationship between the predictors and the target variable *MEDV* in the housing dataset (more insight about the procedure is available on appendix).

3.5.1 Results and Analysis

The results of the Regression Tree analysis showed an **in-sample MSE** of **14.62973**, indicating the average squared difference between the predicted and actual values on the training dataset. The **out-of-sample MSE** obtained on the test dataset was **22.97134**, reflecting the model's performance on unseen data.

The tree diagram visualization provided insights into the important predictors and their respective splits, illustrating the decision-making process within the algorithm. This allowed for identifying influential predictors and understanding the hierarchical relationships between them.

In conclusion, the Regression Tree analysis demonstrated the algorithm's ability to capture non-linear relationships and provide interpretable insights. However, it is important to consider the potential for overfitting and the need for pruning techniques to improve generalization performance.

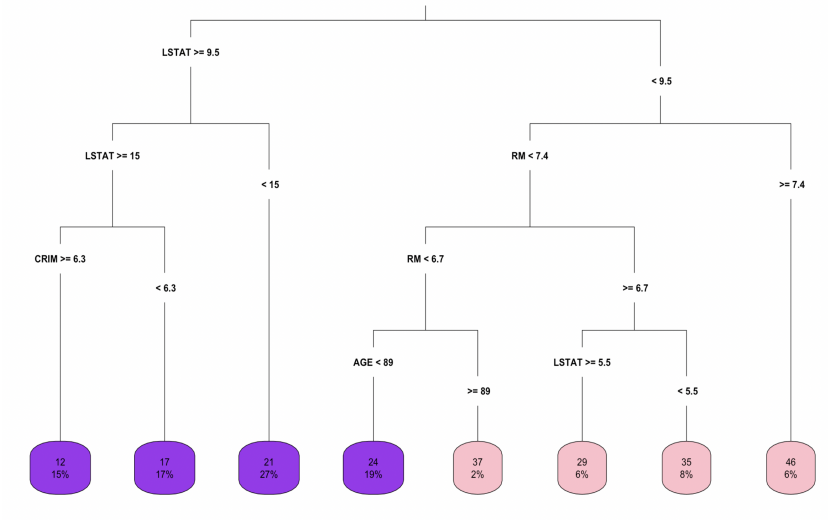

Figure 16: Regression Tree

3.6 Model choice

In this section, we will talk about different methods for selecting models and decision making for posterior distributions and predictions. For Bayesian model choice, we start with the full model, which includes all the predictors. The uncertainty of selecting variables, or model uncertainty that we have been discussing, arises when we believe that some of the explanatory variables may be unrelated to the response variable. This corresponds to setting a regression coefficient β_j to be exactly zero. We specify prior distributions that reflect our uncertainty about the importance of variables. We then update the model based on the data we obtained, resulting in posterior distributions over all models and the coefficients and variances within each model.

3.6.1 Bayesian model averaging (BMA)

We do have a single model, the one that is obtained by averaging all models using their posterior probabilities, the Bayesian model averaging model, or BMA. This is referred to as a hierarchical model and it is composed of many simpler models as building blocks. This represents the full posterior uncertainty after seeing the data.

3.6.2 Highest probability model (HPM)

If our objective is to learn what is the most likely model to have generated the data using a 0-1 loss L0, then the highest probability model (HPM) is optimal. We see that, except the intercept, which is always in any models, the highest probability model also includes *CRIM*, *ZN*, *CHAS*, *NOX*, *RM*, *DIS*, *RAD*, *TAX*, *PTRATIO*, *BLACK*, and *LSTAT*.

3.6.3 Median probability model (MPM)

Another model that is frequently reported, is the median probability model (MPM). This model includes all predictors whose marginal posterior inclusion probabilities are greater than 0.5. If the variables are all uncorrelated, this will be the same as the highest posterior probability model. For a sequence of nested models such as polynomial regression with increasing powers, the median probability model is the best single model for prediction. As we see, this model only includes 11 variables, *CRIM*, *ZN*, *CHAS*, *NOX*, *RM*, *DIS*, *RAD*, *RAD*, *PTRATIO*, *BLACK*, and *LSTAT*.

3.6.4 Best predictive model

If our objective is prediction from a single model, the best choice is to find the model whose predictions are closest to those given by BMA. "Closest" could be based on squared error loss for predictions, or be

based on any other loss functions. Unfortunately, there is no nice expression for this model. However, we can still calculate the loss for each of our sampled models to try to identify this best predictive model, or BPM. The best predictive model includes 12 variables: *CRIM*, *ZN*, *INDUS*, *CHAS*, *NOX*, *RM*, *DIS*, *RAD*, *PTRATIO*, *BLACK*, and *LSTAT*.

3.6.5 Comparison of models

After discussing all 4 different models, let us compare their prediction results.

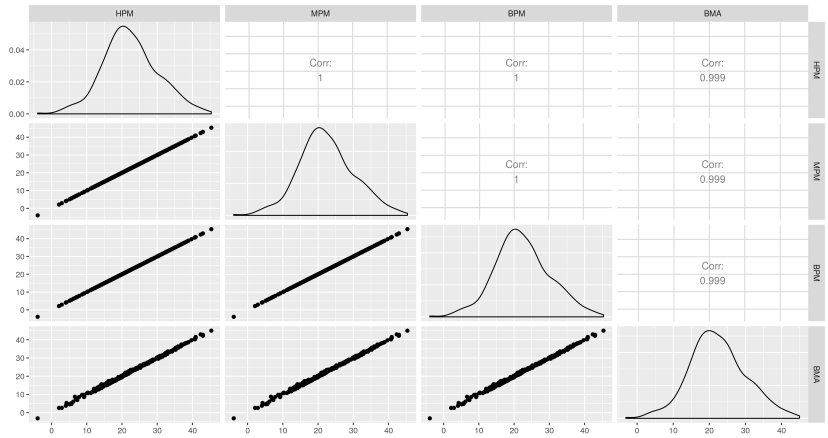

Figure 17: Comparison of models

From the above paired correlation plots (Figure 17), we see that the correlations among them are extremely high. As expected, the single best predictive model (BPM) has the highest correlation with MPM, with a correlation of 1. However, the highest posterior model (HPM) and the Bayesian predictive model (BPM) are equally as good. The same situation also applies to HPM and MPM.

4 Final comments & conclusions

In this project, we tried to solve the problem of regression on Boston housing data set. We approached this problem with different methods and techniques. We applied Bayesian models using different priors and methods, moreover we have tried hyper-parameter tuning and techniques such as PCA and feature selection to further improve our models. For further exploration and more comprehensive comparison we also applied models and techniques such as SVR, regression tree, lasso regression, and also regularization techniques such as lasso, ridge, and elastic-net. However, we discovered that the regularization was not so much effective based on the obtained results, but maybe an effective lasso regularization can provide improvement because it can exclude some features. Based on the obtained results we reach to an conclusion that best suited models for this specific problem were SVR and multiple linear regression. Bayesian models are also provide good estimate and analysis of the problem at hand. Furthermore, we discovered that selecting relevant features is considered very effective in this problem, in all the approaches applied to the problem and enhanced the performance.

5 Appendix

In this section we provide more detailed information regarding the models, model's posterior distribution, analysis criteria and the obtained results.

5.1 Multiple linear regression model with regularization

We also explored the regularization techniques such as lasso, ridge, and elastic-net regularization with multiple linear regression.

5.2 Multiple linear regression model with principal component analysis

Principal component analysis (PCA) is a mathematical procedure, which takes a few linearly correlated features and returns few uncorrelated features. It is often used in dimensionality reduction for reducing complexity of learning models or to visualize the multidimensional data into 2D or 3D data, making to easy to visualize. However, we do not need it for dimesionality reduction of course, as our model is not that complex, We need to remove the multicollinearity problem in our data. By selecting the first 4 principal components we managed to reach the **MSE** of **12.2466** on multiple linear regression model. Further more we also tested PCA with bayesian models using all the mentioned prior above. Figure 18 shows the posterior probability based on 4 principal components used in Bayesian models. From the plot we can note that all the components have the probability of 1 in all the priors which indicates that all the methods consider that these components provide prominent information.

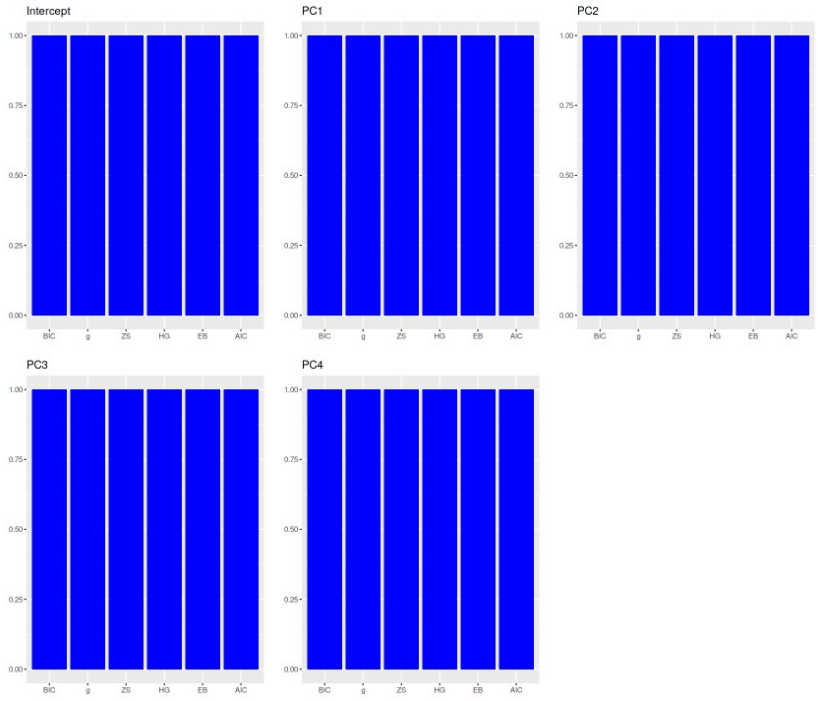

Figure 18: Comparison of the posterior probability using PCA

5.3 Support vector regression

Support Vector Regression (SVR) is a machine learning algorithm used for regression tasks. It aims to find a hyperplane that best fits the data points, while also maximizing the margin between the hyperplane and the data points. In SVR, the goal is to minimize the error within a certain tolerance level, rather than fitting the data exactly. The code for SVR was implemented using the 'e1071' package in R. The Boston housing dataset was divided into predictors (X) and the target variable (y). To ensure that all variables have a comparable scale, the predictors were scaled using the 'scale()'

function. This step is important for models like SVR that are sensitive to the scale of the variables. The dataset was randomly split into training and test sets using a 70:30 ratio. This split allowed for training the SVR model on the training set and evaluating its performance on the test set. The SVR model was trained using the 'svm()' function, specifying that the target variable should be predicted based on all other variables in 'trainX'. The radial kernel was chosen for this implementation. To assess the performance of the SVR model, predictions were made on the test set using the trained model and the 'predict()' function.

5.3.1 Assumptions, Advantages, and Disadvantages

SVR relies on the assumption that the relationship between the predictors and the target variable is approximately linear, and that the predictors are independent and do not exhibit multi-collinearity. Advantages of SVR include its ability to handle nonlinear relationships using different kernel functions and its effectiveness in handling datasets with high dimensionality. However, SVR can be computationally expensive and time-consuming for large datasets, and it requires careful selection of the kernel function and tuning of hyperparameters.

5.3.2 Comparison of Linear Regression and Support Vector Regression Models

The Linear Regression model achieved an **MSE** of **23.06699**, an **RMSE** of **4.802811**, and an **R-squared** value of **0.7461345**. Similarly, the Support Vector Regression model obtained an **MSE** of **13.71281**, an **RMSE** of **3.703082**, and an **R-squared** value of **0.8513956**.

To statistically compare the performance of the LR and SVR models, a t-test was conducted on the predicted values of both models. The test examined whether there was a significant difference in the means of the two sets of predictions. The results of the t-test revealed a **t-value** of **-0.00092004** and a **p-value** of **0.9993**. With a 95% confidence interval (-1.831043, 1.829331) that includes zero, the test indicated no significant difference between the means of the LR and SVR model predictions.

In conclusion, both the LR and SVR models demonstrated effective performance on the Boston housing dataset. The LR model achieved reasonably good results, with an acceptable level of MSE and RMSE, and explained approximately 74.6% of the variability in the target variable. Meanwhile, the SVR model performed slightly better, with a lower MSE and RMSE, and explained approximately 85.1% of the variability in the target variable. The statistical test confirmed that there was no significant difference between the LR and SVR models, suggesting that they provide comparable predictions. Both models can be considered useful tools for estimating the median value of owner-occupied homes based on the input features.

5.3.3 Conclusion and Findings

The evaluation of the SVR model on the Boston housing dataset demonstrates its effectiveness in predicting the median value of owner-occupied homes. The scatter plot shows a strong alignment between the actual and predicted values, indicating the model's ability to capture the underlying patterns in the data. Furthermore, the residual plot reveals no systematic errors or heteroscedasticity in the model, suggesting its reliability and robustness.

Overall, the SVR model performs well on the Boston housing dataset, as evidenced by the small mean squared error, root mean squared error, and high R-squared value. These findings indicate that the SVR model provides accurate predictions and can be a valuable tool in estimating the median value of owner-occupied homes based on the input features.

In conclusion, the SVR model shows promise in predicting housing prices and can be a useful tool for real estate professionals, policymakers, and researchers in understanding and estimating the value of owner-occupied homes.

5.4 Lasso regression

Lasso Regression is a powerful technique for feature selection and regularization. It can handle datasets with many predictors by shrinking the coefficients of irrelevant or less important features to zero, effectively eliminating them from the model. This allows for a more parsimonious and interpretable model.

The Boston housing dataset was split into predictors (X) and the target variable *MEDV*. The predictors were scaled to ensure each feature contributes equally to the model fitting process. The data was then divided into training and test sets, with 70% of the data used for training.

The Lasso Regression model was trained using the 'cv.glmnet()' function from the 'glmnet' package. This function performs cross-validation to determine the optimal value of the tuning parameter, lambda. The lambda.min value was selected as the optimal lambda, and the Lasso model was fitted using 'glmnet()' with the optimal lambda.

Predictions were made on the test set using the Lasso model, and evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared were calculated to assess the model's performance.

5.4.1 Assumptions, Advantages, and Disadvantages

Lasso Regression assumes that the relationship between the predictors and the target variable is linear. It also assumes that there is no multi-collinearity among the predictors, meaning that they are not highly correlated with each other. Violations of these assumptions may lead to biased or unreliable results.

Lasso Regression offers several advantages. Firstly, it performs automatic feature selection by shrinking the coefficients of irrelevant predictors to zero, which can improve model interpretability and reduce overfitting. Secondly, it handles datasets with many predictors, making it suitable for high-dimensional data analysis. Lastly, Lasso Regression is computationally efficient and can handle large datasets.

However, Lasso Regression also has some limitations. It may struggle with correlated predictors, as it tends to arbitrarily select one predictor over another with a similar effect. Additionally, if the predictors are highly correlated, Lasso Regression may select only one of them, potentially overlooking important information. Furthermore, Lasso Regression assumes that the relationship between predictors and the target variable is linear, which may not hold in all cases.

5.5 Stepwise Regression

The variable selection process was performed using stepwise regression to identify the most important predictors for the target variable *MEDV* in the housing dataset. Three different approaches were applied: forward selection, backward selection, and stepwise selection.

In forward selection, the model starts with an empty model and iteratively adds one predictor at a time, selecting the one that provides the best improvement in model fit. The final model obtained through forward selection includes the following predictors: *LSTAT*, *RM*, *PTRATIO*, *DIS*, *B*, *NOX*, *CHAS*, *ZN*, *CRIM*, *RAD*, and *TAX*.

In backward selection, the model starts with the full model and iteratively removes one predictor at a time, excluding the one that contributes the least to the model fit. The final model obtained through backward selection was not explicitly provided in the code snippet.

Stepwise selection combines both forward and backward selection. It starts with an empty model and iteratively adds or removes predictors based on their contribution to the model fit. The final model obtained through stepwise selection is the same as the one obtained through forward selection and includes the predictors.

To evaluate the performance of the selected model, several metrics were considered. The Akaike Information Criterion **(AIC)** and the Bayesian Information Criterion **(BIC)** were calculated, resulting in values of **2405.815** and **2457.833**, respectively. These metrics provide a measure of the goodness of fit of the model, with lower values indicating better fit.

The summary of the final model obtained through stepwise selection reveals valuable insights. The model has a residual standard error of 4.671, indicating the average difference between the observed and predicted values of the target variable. The multiple **R-squared** value of **0.7521** suggests that approximately 75.21% of the variability in the target variable can be explained by the selected predictors.

Analyzing the coefficients of the predictors, we can observe their impact on the target variable. For instance, a decrease in *LSTAT*, *RM*, *PTRATIO*, *DIS*, *B*, *NOX*, *CHAS*, *ZN*, *CRIM*, *RAD*, and *TAX* is associated with an increase in *MEDV*, while an increase in these predictors leads to a decrease in

MEDV. Significance tests (indicated by the p-values) show that all the predictors in the model are statistically significant, implying that they have a significant impact on the target variable.

In conclusion, the stepwise regression procedure identified a set of important predictors for predicting the target variable *MEDV* in the housing dataset. The selected model exhibits a reasonable goodness of fit, as indicated by the AIC and BIC values, and provides valuable insights into the relationship between the predictors and the target variable. Further analysis and interpretation of the coefficients can provide deeper understanding and potential implications for decision-making.

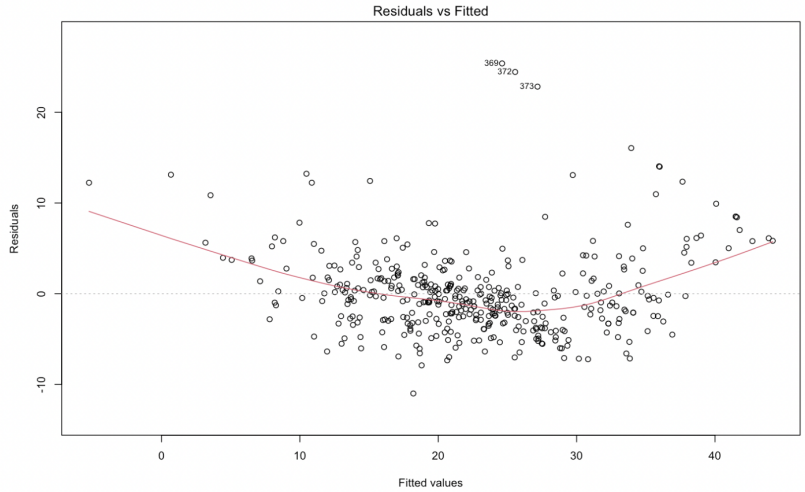

Figure 19: Residuals vs. Fitted Plot

This plot examines the relationship between the residuals (the differences between the observed and predicted values of the target variable) and the fitted values (the predicted values themselves). It helps us assess whether there is a pattern in the residuals, indicating if the model captures the underlying structure of the data adequately. In this plot, we look for a random scatter of points around the horizontal line at zero, indicating that the residuals are randomly distributed and unbiased. Any systematic patterns, such as a curved line or funnel shape, may suggest that the model is not adequately capturing the data's structure.

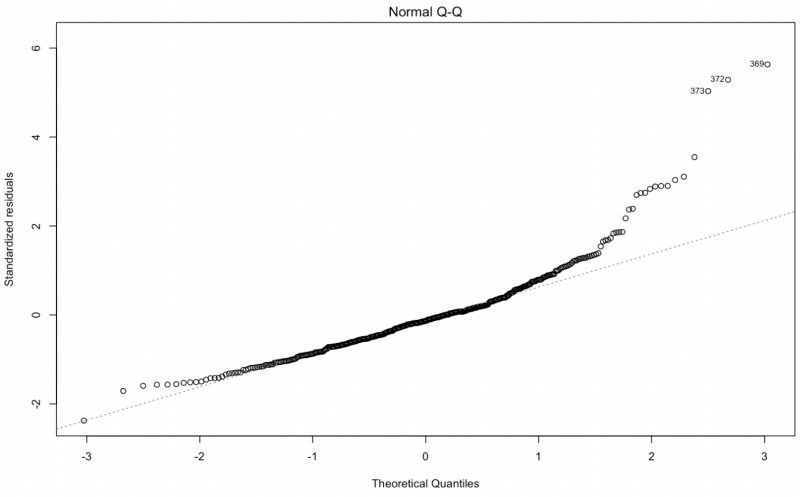

Figure 20: The normal quantile-quantile (Q-Q) plot

The normal quantile-quantile (Q-Q) plot assesses whether the residuals follow a normal distribution. It compares the observed quantiles of the residuals against the quantiles of a standard normal distribution. In an ideal scenario, the points should fall along a straight line, indicating that the

residuals are normally distributed. Departures from a straight line suggest deviations from normality. For instance, if the points deviate from the line in the tails, it indicates heavy tails or outliers in the residuals.

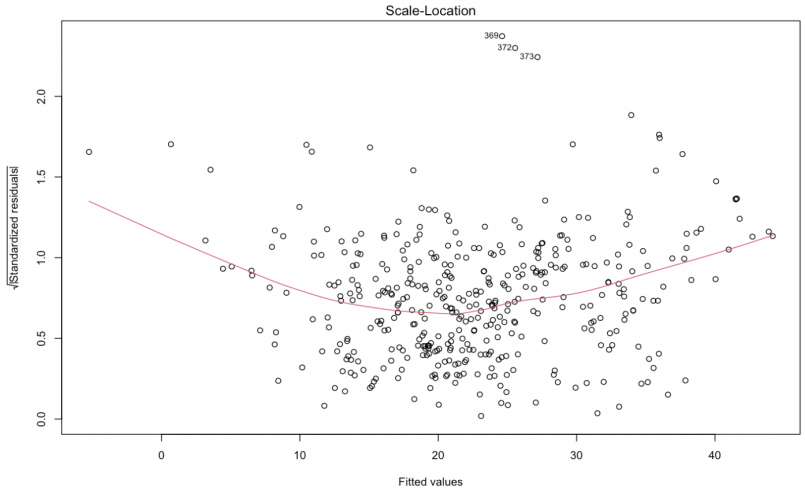

Figure 21: The scale-location plot

The scale-location plot (Figure 21) examines the relationship between the square root of the absolute residuals and the fitted values. This plot helps assess the assumption of constant variance, also known as homoscedasticity. In a well-fitted model, we expect to see a random scatter of points with an even spread across the range of fitted values. If the spread of points appears to change systematically, indicating a cone-like or fan-like shape, it suggests heteroscedasticity, which violates the assumption of constant variance.

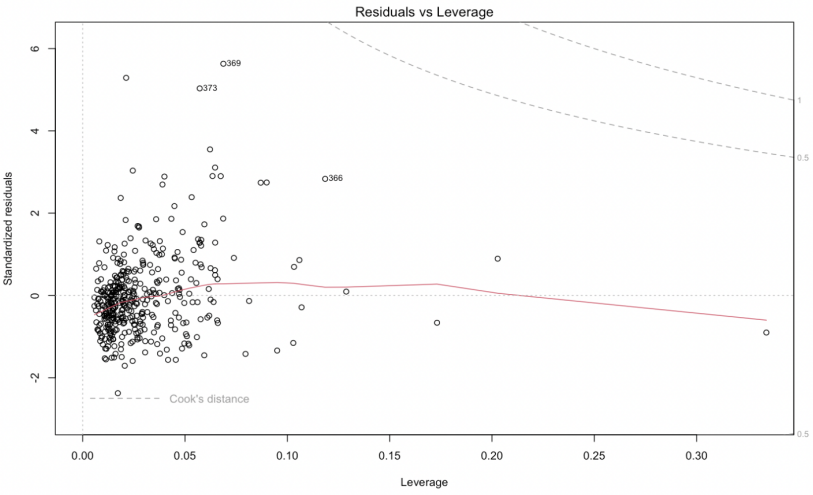

Figure 22: Residuals vs. Leverage Plot

Figure 22 assesses the influence of each observation on the regression model. It combines the standardized residuals (residuals divided by their standard deviation) and the leverage of each observation. Leverage measures how far an observation's predictor values are from the average predictor values. Points with high leverage have extreme predictor values, potentially exerting a strong influence on the regression model. In this plot, we look for points that have both high leverage and large standardized residuals, as they can significantly impact the model's parameters. These influential points may include outliers or observations with extreme predictor values.

5.6 Regression Tree

A Regression Tree analysis was conducted to model the relationship between the predictors and the target variable *MEDV* in the housing dataset. The Regression Tree algorithm was applied using the 'rpart' and 'rpart.plot' packages in R.

The Regression Tree was built using the 'rpart' function, specifying the formula 'MEDV .' to predict the *MEDV* variable using all available predictors. The resulting tree was visualized using the 'rpart.plot' function, providing insights into the splits and decisions made by the algorithm based on the predictor variables.

To assess the model's complexity and determine an optimal tree, we examined the complexity parameter ('cp') values and their corresponding cross-validated error rates using the 'printcp' function.

The performance of the Regression Tree model was evaluated in terms of in-sample mean squared error (MSE) and out-of-sample performance. The in-sample MSE was calculated by comparing the predicted values from the model with the actual target variable values in the training dataset. This metric assesses the model's performance on the data used for training. The out-of-sample MSE was calculated to evaluate the model's predictive performance on unseen data, providing insights into how well the model generalizes.

5.6.1 Assumptions, Advantages, and Disadvantages

The Regression Tree algorithm assumes that the relationship between predictors and the target variable can be effectively captured using a hierarchical structure of decision rules. It is advantageous due to its interpretability, as the resulting tree structure is intuitive and easily understandable. Regression Trees can handle non-linear relationships and are robust to outliers and missing data.

However, Regression Trees can be prone to overfitting, leading to poor generalization performance on unseen data. Additionally, the model's stability may be affected by small changes in the data. Regression Trees may also struggle to effectively utilize continuous variables without appropriate splits.