Empirical Analysis of Credit Rating Changes in S&P 500 Component Stocks: Factors and Predictions
Group 1

林祥恩、李香儀、許詠婷、李明祐、陳碩川

May 29, 2024

Contents

1 **Summary** **2**

2 **Data and Sample** **3**

2.1 Sample selection 3

2.2 Data Processing Process 4

2.3 Dependent Variables 4

2.4 Independent Variables 4

3 **Methodology** **7**

3.1 Ordinary least squares 7

3.2 Ridge Regression 8

3.3 Lasso Regression 8

3.4 ENet Regression 9

3.5 Ordered Logistic Regression 9

3.6 Ordered Probit Regression 9

3.7 Decision Tree 10

3.8 Random Forest 11

4 **Empirical result** **12**

5 **Conclusion** **18**

6 **Reference** **19**

7 **Introduction of each team member** **20**

7.1 陳碩川 20

7.2 林祥恩 20

7.3 李香儀 21

7.4 許詠婷 21

7.5 李明祐 22

Appendix **23**

1 Summary

This report aims to understand the factors influencing changes in credit ratings. We focused on the historical credit rating changes of companies included in the S&P 500 index, utilizing various financial statement data as explanatory variables. Multiple models, including Ordinary Least Squares (OLS), ordered probit model, ordered logistic model, decision tree, random forest, ridge regression, Lasso regression, and ENet regression, were employed to establish predictive models. The comparison among these models involved evaluating the selected significant variables, their explanatory power, and predictive capabilities.

Across the models, we observed a substantial overlap in the variables identified as significant. Notably, variables such as the change in several types of financial ratio, including total-debt-to-total-assets ratio, interest-to-average-total-debt ratio, price-to-sales ratio, interest coverage ratio, gross profit margin, and gross-profit-to-total-assets ratio demonstrated the highest consistency among the selected significant variables. Additionally, some models identified the receivables turnover ratio as a significant variable. The majority of these variables fall under the categories of financial soundness, solvency, and profitability, with valuation and efficiency being a secondary consideration.

2 Data and Sample

2.1 Sample selection

We used companies included in the S&P 500 as our sample, selecting the period from January 31, 2010, to December 31, 2022, with a monthly frequency. The credit ratings, financial ratios, and fundamental data were sourced from WRDS, with the credit ratings being the result of S&P. In building our model, we referenced some relevant studies to choose the factors included.

The study on credit ratings in the Australian market conducted by Grey et al.(2006) [1]provides insight into certain critical financial ratios that significantly influence credit ratings. Grey et al.(2006)[1] use three types of financial ratios to capture the firm' s financial characteristics, which are interest coverage, profitability, and leverage, note that 'interest coverage and leverage ratio have the most pronounced effect on credit ratings'(p. 333). Grey et al.(2006)[1]also mentioned that profitability is an important factor in credit ratings. Therefore, we referred to the two categories of financial ratios commonly used by researchers, as publicly disclosed by the WRDS research team: financial soundness and profitability. Moreover, Grey et al.(2006)[1] pointed out that their model might lack some variables, resulting in the inability to clearly distinguish between AA-rated and A-rated companies. Therefore, we referenced factors discussed in other credit rating studies and incorporated them into our initial model as alternative independent variables.

The study on companies operating in the United States of Sih (2006)[3], shows that there exist significant impacts on cash and market value were found. However, these two factors were highly correlated with many financial ratios already included in the initial model. Considering the issue of multicollinearity, we chose to abandon them and add capitalization and liquidity.

In the research on credit ratings in the Brazilian market by Murcia et al.(2014)[2], it was discovered that the growth of companies had statistically significant explanatory power for credit ratings. However, this factor was highly correlated with profitability. Therefore, we continued to focus on financial ratios as the primary variables. Given that Murcia et al. (2014)[2] found the market-to-book ratio to be statistically significant when used as a proxy variable for financial market performance, we also included it in our initial model.

When conducting credit rating research, the inclusion of valuation and efficiency is less common. We have also incorporated them into the model, hoping to examine whether they have a significant impact on the changes in credit ratings.

2.2 Data Processing Process

We use 20% missing values as a benchmark to determine whether to delete the independent variable or not. If the missing values' proportion is less than 20%, then we use the linear interpolating method to fill the blank space.

Since we model the changes in credit ratings rather than the credit ratings themselves, in constructing the model, the dependent variable corresponds to the change in credit ratings, and for the independent variables, we utilized the variations in financial ratios, specifically the differences between consecutive periods. After removing data with no credit rating changes, there are a total of 193 remaining records.

2.3 Dependent Variables

Since there are many different credit rating levels, we assign different numbers to different credit ratings.

Table 1: Credit Ratings

Rating	Numeric Value	Rating	Numeric Value
D	0	BBB-	12
CC	1	BBB	14
CCC	2	BBB+	16
CCC+	3	A-	19
B-	5	A	22
B	6	A+	25
B+	7	AA-	28
BB-	8	AA	32
BB	9	AA+	36
BB+	10	AAA	40

2.4 Independent Variables

In the regression models that we construct, we can assess the impact of various variables on credit rating changes based on the sign of the corresponding coefficients and whether they are statistically significant. The following are our assumptions regarding the effects of each variable on credit rating changes and the variables selected from each category for inclusion in the initial model.

Financial Soundness / Solvency

H_1: The greater financial soundness improves, the greater credit ratings increase. We utilized the following variables:

Table 2: Selected Variables with Explanations

Variable	Explanation
Cash Flow/Total Debt	Ratio indicating the proportion of cash flow to the total debt
Cash Flow Margin	Percentage representing the cash flow as a margin of total revenue
Interest/Average LTD	Ratio of interest payments to the average long-term debt
Interest/Average Total Debt	Ratio of interest payments to the average total debt
Cash Balance/Total Liabilities	Ratio of cash balance to total liabilities
Free Cash Flow/Operating Cash Flow	Ratio measuring the proportion of free cash flow to operating cash flow
Total Liabilities/Total Tangible Assets	Ratio indicating the proportion of total liabilities to total tangible assets

Table 3: Selected Variables and Explanations

Variable	Explanation
Total Debt/Total Assets	The ratio of total debt to total assets, indicating financial leverage.
Total Debt/Capital	The ratio of total debt to capital, measuring the financial leverage of a company.
Total Debt/Equity	The ratio of total debt to equity, showing the proportion of debt in the capital structure.
After-tax Interest Coverage	The ability of a company to cover interest expenses after-tax payments.
Interest Coverage Ratio	The ratio of earnings before interest and taxes to interest expenses, assessing debt servicing ability.

Profitability

H_2: The greater profitability improves, the greater credit ratings increase.

Efficiency

H_3: Efficiency's change has no impact on credit rating.

Table 4: Selected Profitabilities Variables and Explanations

Variable	Explanation
operating profit margin before depreciation	The percentage of operating profit relative to total revenue before depreciation.
return on equity	The measure of profitability representing the return generated on shareholders' equity.
gross profit margin	The percentage of gross profit relative to total revenue, indicating profitability.
after-tax return on average common equity	The return on common equity after accounting for taxes.
after-tax return on average stockholder's equity	The return on stockholders' equity after accounting for taxes.
gross profit/total assets	The ratio of gross profit to total assets, assessing operational efficiency.

Table 5: Selected Efficiency Variables and Explanations

Variable	Explanation
asset turnover	The efficiency ratio measuring how efficiently a company utilizes its assets to generate sales.
receivables turnover	The number of times a company collects its average accounts receivable during a specific period.
payables turnover	The ratio indicating how quickly a company pays its suppliers.
sales invested capital	The efficiency of capital utilization in generating sales revenue.
sales stockholders equity	The ratio of sales to stockholders' equity, indicating the efficiency of equity in generating sales.

Valuation

H_3: Valuation's change has no impact on credit rating.

Table 6: Selected Financial Ratios and Explanations

Variable	**Explanation**
price/book	The ratio of a company's stock price to its book value per share, indicating valuation.
shillers cyclically adjusted P/E ratio	The price-to-earnings ratio adjusted for inflation and business cycle variations.
enterprise value multiple	The ratio of enterprise value to a financial metric, providing a measure of a company's overall value.
price/operating earnings (basic, excl. EI)	The ratio of stock price to basic operating earnings per share, excluding extraordinary items.
price/operating earnings (diluted, excl. EI)	The ratio of stock price to diluted operating earnings per share, excluding extraordinary items.
P/E (diluted, excl. EI)	The price-to-earnings ratio based on diluted earnings per share, excluding extraordinary items.
P/E (diluted, incl. EI)	The price-to-earnings ratio based on diluted earnings per share, including extraordinary items.
price/sales	The ratio of a company's stock price to its revenue per share, indicating valuation relative to sales.
price/cash flow	The ratio of a company's stock price to its operating cash flow per share, indicating valuation relative to cash flow.

3 Methodology

In the paper, we used eight methods, including ordinary least squares, ridge regression, lasso regression, ENet regression, ordered logistic regression, ordered probabilistic regression, decision trees, and random forests. Each method is tailored to address different challenges in linear regression and ordered categorical dependent variables. We use multiple methods to model changes in credit rates and derive the factors that best explain changes in ratings. In addition, in the machine learning models, we also try to divide the samples into training sets and test sets to measure the performance of different models on changes in credit ratings.

3.1 Ordinary least squares

$a.$ Purpose

Mainly used for linear regression analysis, typically employed to estimate and predict the linear relationship between a dependent variable and one or more independent variables.

b. Features

Finds the best-fitting line by minimizing the sum of squared residuals.

c. Objective function

$$\min_{\beta_0,\beta_1,\ldots,\beta_k} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}))^2 \tag{1}$$

3.2 Ridge Regression

a. Purpose

Used to address multicollinearity (high correlation among independent variables) issues in linear regression.

b. Features

Adds a regularization term to OLS to constrain the size of parameters, aiding in stabilizing estimates and reducing overfitting.

c. Objective function

$$\min_{\beta_0,\beta_1,\ldots,\beta_k} \left\{ \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \beta\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{2}$$

3.3 Lasso Regression

a. Purpose

Also used to address multicollinearity in linear regression, with the added feature of performing feature selection.

b. Features

Similar to Ridge but uses L1 regularization, tends to shrink some unimportant feature weights to zero, achieving feature selection.

c. Objective function

$$\min_{\beta_0,\beta_1,\ldots,\beta_k} \left\{ \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \beta\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{3}$$

3.4 ENet Regression

a. Purpose

Utilizes both L1 and L2 regularization, combining the advantages of Ridge and Lasso, suitable for high-dimensional datasets.

b. Features

Addresses multicollinearity while incorporating feature selection effects.

c. Objective function

$$\min_{\beta_0,\beta_1,\ldots,\beta_k} \left\{ \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \beta\right)^2 + \lambda \left[\frac{1}{2}(1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right] \right\} \tag{4}$$

3.5 Ordered Logistic Regression

a. Purpose

Used for regression problems with an ordered categorical dependent variable, where the dependent variable has an inherent order.

b. Features

Typically applied in scenarios predicting ordered categories, such as consumer satisfaction ratings.

c. Objective function

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} \sum_{k=1}^{K-1} w_{ik} \log \left[\frac{P(Y_i \le k)}{P(Y_i \le k-1)} \right] \\ \text{subject to} \quad & P(Y_i \le K) = 1 \\ & P(Y_i \le 0) = 0 \\ & \sum_{k=1}^{K} P(Y_i = k) = 1 \end{aligned} \tag{5}$$

3.6 Ordered Probit Regression

a. Purpose

Similar to Ordered Logistic Regression, used for ordered categorical dependent variables.

b. Features

Utilizes a normal distribution of latent variables, distinguishing it from Logistic Regression.

$c.$ Objective function

$$\begin{aligned} &\max \quad \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{I}(Y_i = k) \cdot \left[\Phi(\alpha_k - \mathbf{x}_i^T \beta) - \Phi(\alpha_{k-1} - \mathbf{x}_i^T \beta)\right] \\ &\text{subject to} \quad \alpha_0 = -\infty, \quad \alpha_K = \infty, \quad \alpha_1 < \alpha_2 < \ldots < \alpha_{K-1}, \end{aligned} \tag{6}$$

3.7 Decision Tree

$a.$ Purpose

Used for both classification and regression problems, predicting outcomes by recursively splitting the dataset.

$b.$ Features

Strong interpretability, effective in modeling non-linear relationships and interactions, but prone to overfitting.

Usually an objective function (criterion) is used to measure the effect of tree splitting. The goal of Decision Tree is to make the samples in each child node more pure (similar) by splitting the nodes.

There are two common Decision Tree objective functions:

$i.$ Gini Impurity for Classification Tree

Gini impurity measures how mixed the samples within a node are. The lower the Gini impurity, the purer the samples within the node.

$$Gini(t) = 1 - \sum_{i=1}^{c} p(i|t)^2 \tag{7}$$

p(i|t) is the proportion of samples belonging to class i at node t .

$ii.$ Mean Squared Error for Regression Tree

MSE measures the squared average prediction error for samples within a node. The lower the MSE, the more accurate the sample prediction within the node is.

$$MSE(t) = \frac{1}{|t|} \sum_{i \in t} (y_i - \bar{y}_t)^2 \tag{8}$$

MSE(t) is the mean square error on node t .

|t| is the number of samples on node t .

3.8 Random Forest

a. Purpose

An ensemble algorithm based on multiple decision trees, used for classification and regression tasks.

b. Features

Improves model accuracy and generalization by combining multiple weak learners (decision trees), mitigating the risk of overfitting. Furthermore, Random Forest integrates the predictions of multiple decision trees by voting or averaging, depending on whether it is a classification or regression problem.

For classification problems, Random Forest performs voting and selects the class with the most votes as the final prediction.

For regression problems, Random Forest averages the predictions of multiple decision trees as the final prediction value.

4 Empirical result

After running three different regression models, we found that the change of **Total Debt/Total Assets, Interest/Average Total Debt, Price/Sales, Interest Coverage Ratio , Gross profit Margin**, **Gross Profit/Total Assets and Receivables/Turnover** are factors with significant explanatory power in different models.

In Table 7, we use credit ratings and financial ratio information collected from WRDS. After first cleaning the data, the regression model was then established using the OLS method (we know that the OLS method may not be suitable for such data types) and backward selection to find significant factors that affect credit rating analysis.

Surprisingly, these factors with explanatory power are all related to solvency, profitability, leverage, etc., as we mentioned in the literature review in Chapter 2. Most of them have high economic intuition to explain their impact on rating changes.

Table 7: Ordinary Least Squares

	Dependent variable:
	Rating Change
Total Debt / Total Assets (% change)	-1.893^{**}
	(0.911)
Interest / Average Total Debt (% change)	-2.037^{**}
	(0.883)
Price / Sales (% change)	0.651^{**}
	(0.287)
Interest Coverage Ratio (% change)	0.114^{**}
	(0.048)
Gross profit Margin (% change)	-1.906^{**}
	(0.812)
Gross Profit / Total Assets (% change)	1.896^{***}
	(0.683)
Constant	-0.304
	(0.207)
Observations	193
R^2	0.106
Adjusted R^2	0.078
Residual Std. Error	2.580 (df = 186)
F Statistic	3.694^{***} (df = 6; 186)
Note:	*p<0.1; **p<0.05; ***p<0.01

On the other hand, we endeavored to implement regularization methods through cross-validation. Given that the initial dataset comprises numerous similar financial variables within the same category, the issue of multicollinearity emerged, potentially leading to a significant increase in the variance of the regression model. Consequently, we applied **ridge regression (L2)**, **lasso (L1)**, and **elastic net** to assess whether the significance of independent variables changed after undergoing regularization.

Initially, Ridge Regression penalizes each explanatory variable simultaneously, resulting in regression coefficients for almost every variable becoming smaller. In this scenario, we can no longer directly measure the significance of variables by observing their t-values. Instead, we opt for variables with larger coefficients, indicating relatively stronger explanatory power. Here we show below in Table 8:

Table 8: Ridge Regression

	Independent variable:
	Coefficient
Total Debt / Total Assets (% change)	−0.01552
Interest / Average Total Debt (% change)	−0.01334
Constant	−0.1700360

Now, only **Total Debt/Total Assets** and **Interest/Average Total Debt**, along with the intercept, retained their influence on credit rating changes. The advantage of this approach is that, even under the constraints of regularization, it still identifies features that have a strong impact on the target variable. By selecting variables with larger regression coefficients, we retain explanatory variables with a greater influence on the target variable, refining the process of feature selection in our model. It is noteworthy that even when we randomly split the sample into a 60 percent training set and a 40 percent test set, the training model consistently retained the variable **Total Debt/Total Assets**. This observation suggests that this liquidity variable holds significant importance in evaluating credit ratings, as it continues to play a crucial role in the model's predictive capacity across different subsets of the data.

Regarding lasso, it is distinguished by its variable selection functionality. Initially, our expectation was that it might eliminate variables with weak explanatory power while retaining the important ones. However, the outcome of lasso reduced all variables to zero, leaving only the intercept term, which was -0.1709845. Essentially, this value represented the average of all credit rating changes in the sample. This might mean that under the screening mechanism of lasso, all explanatory variables in the sample we selected are highly correlated.

As for Elastic Net, which combines the strengths of both Ridge Regression and Lasso, it ultimately retained two variables, aligning with what we have mentioned. The coefficients of these variables are recorded in the following table:

Table 9: Elastic Net

	Independent variable: Coefficient
Total Debt / Total Assets (% change)	−0.03826
Interest / Average Total Debt (% change)	0.10034
Constant	−0.18984

Next, in Tables 10 and 11, we used ordered logit model and ordered probit model respectively. The difference from the former (OLS) is that when we want to use backward selection, we notice that the model cannot be estimated because the starting value is difficult to determine. Therefore, we switched to the forward selection method and slowly added it from one factor to the next. Finally, we found that this method only added one significant factor compared to the OLS method for backward selection. However, the coefficient estimation result of this factor is contrary to economic intuition. Specifically, this extra factor is the **accounts receivable turnover**. Before our study, we believed that the higher the accounts receivable turnover rate, it means that the company's debt repayment ability is relatively good. However, the estimated coefficient result is negative, which implies that the lower the accounts receivable turnover rate will help improve the credit rating.

We later proposed several explanations, such as firms' tendency to build long-term relationships with customers. Partnerships mean more stable operating income. Or, it is just a coincidence that this variable is significant, and we believe that more samples or experiments are needed to see whether similar situations will occur in companies in other countries.

Table 10: Ordered Logit Model

	Dependent variable:
	Rating Change
Total Debt / Total Assets (% change)	-1.789^{***}
	(0.620)
Interest / Average Total Debt (% change)	-1.792^{***}
	(0.618)
Price / Sales (% change)	0.387^{**}
	(0.181)
Interest Coverage Ratio (% change)	0.099^{***}
	(0.032)
Gross Profit Margin (% change)	-1.576^{***}
	(0.581)
Gross Profit / Total Assets (% change)	1.607^{***}
	(0.515)
Receivables Turnover (% change)	-0.461^{*}
	(0.255)
Observations	193
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 11: Ordered Probit Model

	Dependent variable:
	Rating Change
Total Debt / Total Assets (% change)	-0.876^{**}
	(0.366)
Interest / Average Total Debt (% change)	-0.919^{***}
	(0.352)
Price / Sales (% change)	0.205^{*}
	(0.115)
Interest Coverage Ratio (% change)	0.058^{***}
	(0.020)
Gross Profit Margin (% change)	-0.859^{**}
	(0.350)
Gross Profit / Total Assets (% change)	0.903^{***}
	(0.311)
Receivables Turnover (% change)	-0.285^{*}
	(0.166)
Observations	193
Note:	*p<0.1; **p<0.05; ***p<0.01

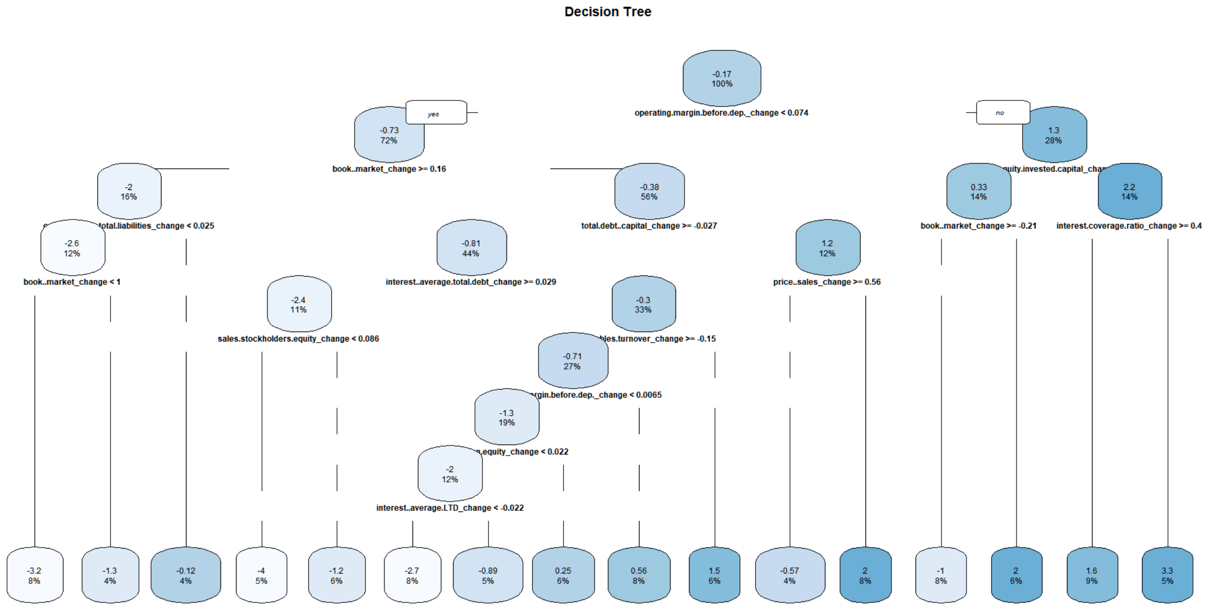


Figure 1: Decision Tree

At last, we applied the tree models, including **decision tree** and **random forest**.

Because our main goal was to find financial variables with explanatory power, we were not making predictions, so we did not consider the problem of overfitting due to many decision-making branch points temporarily. The above Figure 1 is a schematic diagram of the results of the decision tree.

From the above we can observe that the decision tree chose the **operating margin before depreciation** as the most important variable, which is relate with the profitability of a company. Furthermore, we also printed the importance of the variables by the "value.importance" function in the rpart package. Here we listed the top five important variables below:

Table 12: Important Variables in Decision Tree

	Independent variable:
	variable.importance
Operating Margin Before Depreciation	233.3553
Book / Market	161.3987
Price / Book	143.9431
Total Debt / Capital	126.0136
Interest Coverage Ratio	124.9567

Likewisely, as we conducted the random forest model, we also painted the **variable importance plot** in order to determine which variable the model considered most important. Here we show at Figure 2.

The Figure 2 represents the "IncNodePurity" among variables in the model, which is a

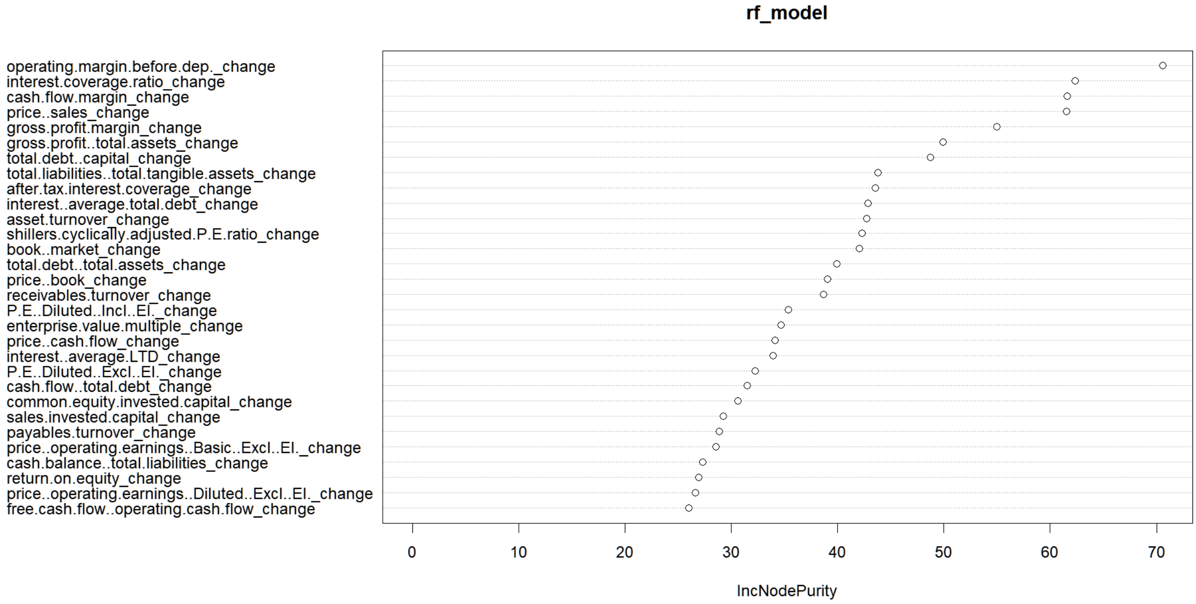


Figure 2: Importance of Variables in Random Forest

measure used to assess the improvement in node purity achieved by splitting a node during the construction of decision trees within the ensemble. It plays a vital role in enhancing the predictive capability of the overall Random Forest model. Here we further list the first five important variables as follows:

Table 13: Important Variables in Random Forest

	Independent variable:
	IncNodePurity
Operating Margin Before Depreciation	70.28877
Price / Sales	64.18365
Gross Profit Margin	59.26180
Interest Coverage Ratio	58.85127
Gross Profit / Total Assets	52.10683

The observation that the variables identified as relatively important in the Random Forest model align with the significant variables verified in the regression models is consistent with the notion of feature importance across different modeling techniques. This consistency reinforces the relevance and influence of those particular variables on the credit rating, providing additional confidence in their significance. This alignment between different modeling techniques contributes to a more robust understanding of the factors driving the outcome of interest.

In conclusion, we divide the factors with significant explanatory power into three parts for discussion: leverage, solvency, and profitability.

Let's first look at the part related to leverage: **Total Debt / Total Assets** represents the change in leverage usage, which is statistically negatively correlated with the change in ratings, which is consistent with our intuition. **Interest / Average Total Debt** represents the change in the cost of leveraging, which is statistically negatively correlated with the change in ratings, which is also in line with our intuition and expectation.

Next, look at debt repayment ability: **Interest Coverage Ratio** represents the improvement of debt repayment ability, which is statistically positively correlated with rating changes, in line with expectations.

Finally, it is related to profits: the statistical results of **Gross Profit / Total Assets** and **Price / Sales** all imply an increase in asset usage efficiency, which will improve the rating.

5 Conclusion

Based on the previous model estimation results, we observed a significant convergence in the variables identified as meaningful across the various models. Notably, variables such as changes in several types of financial ratios, including the total-debt-to-total-assets ratio, interest-to-average-total-debt ratio, price-to-sales ratio, interest coverage ratio, gross profit margin, and gross-profit-to-total-assets ratio, exhibited the highest consistency among the selected significant variables. Additionally, certain models highlighted the receivables turnover ratio as a significant variable. The majority of these variables are categorized under financial soundness, solvency, and profitability, with valuation and efficiency considered as a secondary factor.

To further extend our research, most importantly, we should attempt to increase the number of data entries. Due to a significant portion of the data not meeting the selection criteria, approximately 99.6% of the data has been eliminated (originally 50,000 entries, reduced to 200 after cleaning). Next, we propose considering situations where credit ratings remain unchanged in the sample. This approach addresses the issue of a limited sample size and incorporates information on stable credit ratings into the model. With an increased sample size, we can leverage additional machine learning methods such as XGBoost and Support Vector Machines (SVM) to provide more accurate assessments of credit rating changes. This expanded analysis will contribute to a more comprehensive understanding of the factors influencing credit ratings, particularly by incorporating cases where creditworthiness remains constant.

6 Reference

[1] Stephen Gray, Alexsander Mirkovic, and Vanitha Ragunathan. "The determinants of credit ratings: Australian evidence". In: *Australian Journal of Management* 31.2 (2006), pp. 333–354.

[2] Flávia Cruz de Souza MurciaI et al. "The determinants of credit rating: Brazilian evidence". In: *BAR-Brazilian Administration Review* 11 (2014), pp. 188–209.

[3] Andre Sih. "Predição do grau de ratings corporativos". In: *Programa de Pos-Graduacao do Departamento de Engenharia Eletrica, PontifÃcia Universidade Catolica do Rio de Janeiro, Rio de Janeiro, RJ, Brazil. Master's Thesis* (2006).

7 Introduction of each team member

7.1 陳碩川

Graduated from NCCU PF, responsible for Methodology of ML.

1. Research interests

   a. Macroeconomics Research In Developing Countries

   b. Quantitative Financial Analysis by ML Methods

   c. Time Series Analysis In Macroeconomics Variables

2. Related Experiences

   a. Equity and Industry Research Intern, IPR Advisors

   b. Researcher of China Real Estate Market, NCCU MF Club

   c. Reasercher of Macro News Group, NTU iBank Club

   d. Teaching Assistant of ETP Economics, NCCU IB

   e. Intern of Fund Administrative Department, Eastspring

7.2 林祥恩

Graduated from SCU FEAM, responsible for Data Cleaning and Methodology of Regression.

1. Research interests

   a. Constructing Trading Strategy

   b. Quantitative Financial Analysis

2. Related Experiences

   a. Quantitative Trader, Pergolas Investing

   b. Lecturer of Financial Management, GET

   c. Lecturer of Economics, GET

   d. Teaching Assistant of Financial Management, SCU

7.3 李香儀

Graduated from UT Math, responsible for Data and Samples and Methodology of Regression.

1. Research interests
   - *a.* Computational Finance
   - *b.* Derivatives and Risk Management
   - *c.* Applications of Machine Learning in Finance
   - *d.* Mathematical Modeling
2. Related Experiences
   - *a.* US Dollar Exchange Rate Analysis
   - *b.* Course TA of Probability
   - *c.* Course TA of Calculus

7.4 許詠婷

Graduated from NCKU STAT, responsible for Summary&Conclusion.

1. Research interests
   - *a.* Statistical Modeling & Factorial Analysis
   - *b.* Application of Time Series Model Building
   - *c.* Analysis & Application of Risk Management
2. Related Experiences
   - *a.* Predicting Crude Oil Market Trends
   - *b.* Analysis of Potential Factors Influencing Facial Anxiety
   - *c.* Time Series Analysis - Correlation between Economic Fluctuations and the Number of Drug Seizure Cases.

7.5 李明祐

Graduated from NTUST BA, responsible for Methodology Introduce.

1. Research interests
   - a. Corporate Governance and Royalty Calculation
   - b. Financial innovation and digital transformation
   - c. Discussion on the International General Economy
2. Related Experiences
   - a. Internship at TEJ Digital Transformation Project
   - b. Financial assistant at Jiefeng management consulting firm
   - c. NFT community management and marketing research
   - d. Financial evaluation of Greater Taipei BOT cases
   - e. The trend of the British pound expectations

Empirical Analysis of Credit Rating Changes in S&P500 Component Stocks: Factors and Predictions

Group 1

```r
rm(list=ls())
library(MASS)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

OLS

```r
# full sample
data <- read.csv('C:/Users/user/Desktop/      /Project 1/New Data/SP500_change_V7Final.csv')
data <- data[, -1]
factor_summary <- summary(data[1:35])
factor_summary
```

```
##   rating_diff      after.tax.interest.coverage_change
##  Min.   :-11.000   Min.   :-7.2370
##  1st Qu.: -2.000   1st Qu.:-0.3830
##  Median : -1.000   Median :-0.0250
##  Mean   : -0.171   Mean   : 0.2326
##  3rd Qu.:  2.000   3rd Qu.: 0.3080
##  Max.   :  6.000   Max.   :33.6130
##  interest.coverage.ratio_change cash.flow..total.debt_change
##  Min.   :-13.2550               Min.   :-3.17600
##  1st Qu.: -0.3630               1st Qu.:-0.26500
```

```
##  Median : -0.0110               Median : 0.00000
##  Mean   :  0.1875               Mean   :-0.02884
##  3rd Qu.:  0.2640               3rd Qu.: 0.08700
##  Max.   : 56.9970               Max.   : 8.30000
##  operating.margin.before.dep._change return.on.equity_change
##  Min.   :-10.67000                   Min.   :-32.2000
##  1st Qu.: -0.14800                   1st Qu.: -0.4600
##  Median :  0.00000                   Median : -0.0480
##  Mean   : -0.02581                   Mean   :  0.6636
##  3rd Qu.:  0.08300                   3rd Qu.:  0.3700
##  Max.   : 18.50000                   Max.   : 82.9640
##  total.debt..total.assets_change book..market_change
##  Min.   :-0.32100                Min.   :-0.9860
##  1st Qu.:-0.02400                1st Qu.:-0.2610
##  Median : 0.00800                Median :-0.0790
##  Mean   : 0.04941                Mean   : 0.1614
##  3rd Qu.: 0.08100                3rd Qu.: 0.0710
##  Max.   : 2.06700                Max.   :23.4290
##  interest..average.LTD_change interest..average.total.debt_change
##  Min.   :-0.85700             Min.   :-0.86200
##  1st Qu.:-0.16100             1st Qu.:-0.17200
##  Median :-0.03400             Median :-0.04700
##  Mean   :-0.01954             Mean   :-0.05958
##  3rd Qu.: 0.03900             3rd Qu.: 0.03600
##  Max.   : 5.05400             Max.   : 1.33300
##  cash.balance..total.liabilities_change
##  Min.   :-0.9520
##  1st Qu.:-0.2710
##  Median :-0.0360
##  Mean   : 0.3645
##  3rd Qu.: 0.2010
##  Max.   :19.6670
##  free.cash.flow..operating.cash.flow_change
##  Min.   :-55.5560
##  1st Qu.: -0.2670
##  Median : -0.0410
##  Mean   : -0.6797
##  3rd Qu.:  0.0310
##  Max.   : 22.0600
##  total.liabilities..total.tangible.assets_change total.debt..capital_change
##  Min.   :-0.6060                                 Min.   :-0.4570
##  1st Qu.:-0.0600                                 1st Qu.:-0.0380
##  Median : 0.0110                                 Median : 0.0120
##  Mean   : 0.1044                                 Mean   : 0.1108
##  3rd Qu.: 0.1450                                 3rd Qu.: 0.1260
##  Max.   : 4.9050                                 Max.   : 4.2880
##  total.debt..equity_change asset.turnover_change receivables.turnover_change
##  Min.   :-172.2750         Min.   :-0.83400      Min.   :-0.60700
##  1st Qu.:  -0.1100         1st Qu.:-0.10600      1st Qu.:-0.11800
##  Median :   0.0140         Median :-0.00800      Median :-0.01600
##  Mean   :  -0.6311         Mean   :-0.01718      Mean   : 0.04384
##  3rd Qu.:   0.2150         3rd Qu.: 0.04800      3rd Qu.: 0.07800
##  Max.   :  34.5590         Max.   : 1.73700      Max.   : 6.39300
##  payables.turnover_change sales.invested.capital_change
```

```
##  Min.   :-1.34800         Min.   :-0.86500
##  1st Qu.:-0.11100         1st Qu.:-0.12700
##  Median :-0.00900         Median : 0.00000
##  Mean   : 0.04643         Mean   : 0.03226
##  3rd Qu.: 0.05900         3rd Qu.: 0.08400
##  Max.   : 4.20700         Max.   : 4.94300
##  sales.stockholders.equity_change price..book_change
##  Min.   :-0.9500                  Min.   :-0.9590
##  1st Qu.:-0.1080                  1st Qu.:-0.1470
##  Median : 0.0090                  Median : 0.0950
##  Mean   : 0.5416                  Mean   : 0.5529
##  3rd Qu.: 0.1580                  3rd Qu.: 0.3730
##  Max.   :32.0330                  Max.   :25.6740
##  shillers.cyclically.adjusted.P.E.ratio_change enterprise.value.multiple_change
##  Min.   :-20.5110                              Min.   :-29.0740
##  1st Qu.: -0.5750                              1st Qu.: -0.1070
##  Median : -0.0370                              Median :  0.0770
##  Mean   : -0.3528                              Mean   : -0.1701
##  3rd Qu.:  0.3130                              3rd Qu.:  0.2800
##  Max.   :  3.8670                              Max.   :  7.1450
##  price..operating.earnings..Basic..Excl..EI._change
##  Min.   :-18.761
##  1st Qu.: -0.420
##  Median :  0.015
##  Mean   :  3.921
##  3rd Qu.:  0.333
##  Max.   :727.017
##  price..operating.earnings..Diluted..Excl..EI._change
##  Min.   :-18.761
##  1st Qu.: -0.417
##  Median :  0.009
##  Mean   :  3.897
##  3rd Qu.:  0.338
##  Max.   :727.017
##  P.E..Diluted..Excl..EI._change P.E..Diluted..Incl..EI._change
##  Min.   : -5.744                Min.   :-12.931
##  1st Qu.: -0.494                1st Qu.: -0.524
##  Median :  0.019                Median :  0.024
##  Mean   :  4.204                Mean   :  4.262
##  3rd Qu.:  0.597                3rd Qu.:  0.597
##  Max.   :746.319                Max.   :746.319
##  price..sales_change price..cash.flow_change gross.profit.margin_change
##  Min.   :-0.9140     Min.   :-36.52400       Min.   :-9.718
##  1st Qu.:-0.1660     1st Qu.: -0.24500       1st Qu.:-0.062
##  Median : 0.0640     Median :  0.00800       Median : 0.000
##  Mean   : 0.2068     Mean   : -0.08246       Mean   :-0.018
##  3rd Qu.: 0.3640     3rd Qu.:  0.39300       3rd Qu.: 0.046
##  Max.   : 3.9480     Max.   : 19.50600       Max.   : 6.571
##  after.tax.return.on.average.common.equity_change
##  Min.   : -33.200
##  1st Qu.:  -0.466
##  Median :  -0.053
##  Mean   :   6.619
##  3rd Qu.:   0.342
```

```
##  Max.   :1154.729
##  after.tax.return.on.average.stockholders..equity_change
##  Min.   : -33.200
##  1st Qu.:  -0.463
##  Median :  -0.052
##  Mean   :   6.712
##  3rd Qu.:   0.339
##  Max.   :1154.729
##  gross.profit..total.assets_change common.equity.invested.capital_change
##  Min.   :-13.03200                 Min.   :-9.5000
##  1st Qu.: -0.15100                 1st Qu.:-0.1230
##  Median : -0.01900                 Median :-0.0130
##  Mean   : -0.04508                 Mean   : 0.2562
##  3rd Qu.:  0.09100                 3rd Qu.: 0.0560
##  Max.   :  6.47100                 Max.   :24.5000
##  cash.flow.margin_change
##  Min.   :-33.500
##  1st Qu.: -0.286
##  Median : -0.042
##  Mean   :  1.054
##  3rd Qu.:  0.112
##  Max.   :302.000
```

```r
# ols_1 <- stargazer(factor_summary, type = "latex")
factor_cov <- cov(data[1:35])

# Train and test data
data_train_df70 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/train_df70.csv")
data_train_df70 <- data_train_df70[, -1]
data_test_df30 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/test_df30.csv")
data_test_df30 <- data_test_df30[, -1]

model_ols <- lm(rating_diff ~ .,
                data = data)
summary(model_ols)
```

```
##
## Call:
## lm(formula = rating_diff ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2666  -1.5185  -0.0922   1.8230   6.3207
##
## Coefficients:
##                                                           Estimate Std. Error
## (Intercept)                                              -0.296241   0.255960
## after.tax.interest.coverage_change                       -0.007279   0.087874
## interest.coverage.ratio_change                            0.128463   0.080337
## cash.flow..total.debt_change                              0.169666   0.284642
## operating.margin.before.dep._change                      -0.021491   0.180636
## return.on.equity_change                                   0.016153   0.039831
## total.debt..total.assets_change                          -4.706216   3.743905
## book..market_change                                      -0.122624   0.143426
```

```
## interest..average.LTD_change                           0.228683   1.011795
## interest..average.total.debt_change                   -2.028868   1.556061
## cash.balance..total.liabilities_change                -0.038724   0.098157
## free.cash.flow..operating.cash.flow_change            -0.013377   0.059079
## total.liabilities..total.tangible.assets_change        0.536974   1.008106
## total.debt..capital_change                             1.681743   1.729715
## total.debt..equity_change                              0.016069   0.065049
## asset.turnover_change                                  1.282533   2.165425
## receivables.turnover_change                           -0.848560   0.530768
## payables.turnover_change                               0.363878   0.496052
## sales.invested.capital_change                         -0.608110   0.665468
## sales.stockholders.equity_change                       0.030921   0.162050
## price..book_change                                    -0.167040   0.197196
## shillers.cyclically.adjusted.P.E.ratio_change         -0.043580   0.114551
## enterprise.value.multiple_change                      -0.017912   0.105600
## price..operating.earnings..Basic..Excl..EI._change    -0.142749   0.501874
## price..operating.earnings..Diluted..Excl..EI._change   0.144695   0.540458
## P.E..Diluted..Excl..EI._change                        -0.026905   0.122329
## P.E..Diluted..Incl..EI._change                         0.021042   0.103722
## price..sales_change                                    1.031496   0.428051
## price..cash.flow_change                               -0.048333   0.060207
## gross.profit.margin_change                            -2.472924   1.648566
## after.tax.return.on.average.common.equity_change       0.209777   0.290931
## after.tax.return.on.average.stockholders..equity_change -0.207166   0.290366
## gross.profit..total.assets_change                      2.581863   1.506721
## common.equity.invested.capital_change                  0.036793   0.116770
## cash.flow.margin_change                               -0.008675   0.009514
##                                                       t value Pr(>|t|)
## (Intercept)                                            -1.157   0.2489
## after.tax.interest.coverage_change                     -0.083   0.9341
## interest.coverage.ratio_change                          1.599   0.1118
## cash.flow..total.debt_change                            0.596   0.5520
## operating.margin.before.dep._change                    -0.119   0.9054
## return.on.equity_change                                 0.406   0.6856
## total.debt..total.assets_change                        -1.257   0.2106
## book..market_change                                    -0.855   0.3939
## interest..average.LTD_change                            0.226   0.8215
## interest..average.total.debt_change                    -1.304   0.1942
## cash.balance..total.liabilities_change                 -0.395   0.6937
## free.cash.flow..operating.cash.flow_change             -0.226   0.8212
## total.liabilities..total.tangible.assets_change         0.533   0.5950
## total.debt..capital_change                              0.972   0.3324
## total.debt..equity_change                               0.247   0.8052
## asset.turnover_change                                   0.592   0.5545
## receivables.turnover_change                            -1.599   0.1119
## payables.turnover_change                                0.734   0.4643
## sales.invested.capital_change                          -0.914   0.3622
## sales.stockholders.equity_change                        0.191   0.8489
## price..book_change                                     -0.847   0.3982
## shillers.cyclically.adjusted.P.E.ratio_change          -0.380   0.7041
## enterprise.value.multiple_change                       -0.170   0.8655
## price..operating.earnings..Basic..Excl..EI._change     -0.284   0.7765
## price..operating.earnings..Diluted..Excl..EI._change    0.268   0.7893
## P.E..Diluted..Excl..EI._change                         -0.220   0.8262
```

```
## P.E..Diluted..Incl..EI._change                               0.203   0.8395
## price..sales_change                                          2.410   0.0171 *
## price..cash.flow_change                                     -0.803   0.4233
## gross.profit.margin_change                                  -1.500   0.1356
## after.tax.return.on.average.common.equity_change             0.721   0.4719
## after.tax.return.on.average.stockholders..equity_change     -0.713   0.4766
## gross.profit..total.assets_change                            1.714   0.0886 .
## common.equity.invested.capital_change                        0.315   0.7531
## cash.flow.margin_change                                     -0.912   0.3633
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.71 on 158 degrees of freedom
## Multiple R-squared:  0.1624, Adjusted R-squared:  -0.01784
## F-statistic: 0.901 on 34 and 158 DF,  p-value: 0.628
```

```r
# Backward model selection
model_backward <- step(model_ols,direction="backward", test="F")
summary(model_backward)
```

```r
# According to Backward model selection...
model_ols2 <- lm(rating_diff ~ total.debt..total.assets_change +
                 interest..average.total.debt_change +
                 price..sales_change +
                 interest.coverage.ratio_change +
                 gross.profit.margin_change +
                 gross.profit..total.assets_change,
                 data = data)
summary(model_ols2)
```

```
##
## Call:
## lm(formula = rating_diff ~ total.debt..total.assets_change +
##     interest..average.total.debt_change + price..sales_change +
##     interest.coverage.ratio_change + gross.profit.margin_change +
##     gross.profit..total.assets_change, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6702  -1.6358  -0.0452   2.0430   6.7069
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -0.30367    0.20684  -1.468   0.1438
## total.debt..total.assets_change     -1.89274    0.91123  -2.077   0.0392 *
## interest..average.total.debt_change -2.03662    0.88307  -2.306   0.0222 *
## price..sales_change                  0.65109    0.28726   2.266   0.0246 *
## interest.coverage.ratio_change       0.11432    0.04814   2.375   0.0186 *
## gross.profit.margin_change          -1.90552    0.81190  -2.347   0.0200 *
## gross.profit..total.assets_change    1.89587    0.68340   2.774   0.0061 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.58 on 186 degrees of freedom
## Multiple R-squared:  0.1065, Adjusted R-squared:  0.07766
## F-statistic: 3.694 on 6 and 186 DF,  p-value: 0.001715
```

```r
# train 70-30
model_ols_70 <- lm(rating_diff ~ total.debt..total.assets_change +
                   interest..average.total.debt_change +
                   price..sales_change +
                   interest.coverage.ratio_change +
                   gross.profit.margin_change +
                   gross.profit..total.assets_change,
                   data = data_train_df70, type = 'response')
summary(model_ols_70)
```

```
##
## Call:
## lm(formula = rating_diff ~ total.debt..total.assets_change +
##     interest..average.total.debt_change + price..sales_change +
##     interest.coverage.ratio_change + gross.profit.margin_change +
##     gross.profit..total.assets_change, data = data_train_df70,
##     type = "response")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8313  -1.6690  -0.0981   1.9730   6.6634
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          -0.2306     0.2535  -0.910   0.3647
## total.debt..total.assets_change      -2.0494     1.0645  -1.925   0.0564 .
## interest..average.total.debt_change  -2.2632     1.1623  -1.947   0.0537 .
## price..sales_change                   0.5958     0.3457   1.723   0.0872 .
## interest.coverage.ratio_change        0.1163     0.0534   2.177   0.0313 *
## gross.profit.margin_change           -1.9553     1.0462  -1.869   0.0639 .
## gross.profit..total.assets_change     1.9486     0.8819   2.210   0.0289 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.669 on 128 degrees of freedom
## Multiple R-squared:  0.1041, Adjusted R-squared:  0.06214
## F-statistic:  2.48 on 6 and 128 DF,  p-value: 0.02653
```

```r
required_columns <- c('total.debt..total.assets_change',
                      'interest..average.total.debt_change',
                      'price..sales_change',
                      'interest.coverage.ratio_change',
                      'gross.profit.margin_change',
                      'gross.profit..total.assets_change')
test_df30 <- data_test_df30[required_columns]
predictors_30 <- predict(model_ols_70, type='response', newdata=test_df30)
print(predictors_30-data_test_df30$rating_diff)
```

```
##          1          2          3          4          5          6          7
```

```
## -3.0894690 -2.6878290  0.9125864  2.3224191 -1.8508558  3.0627068 -0.6171860
##          8          9         10         11         12         13         14
## -2.1376147 -3.0157658  1.2019071 -5.5184977 -1.5032964 -3.5685912 -3.3123091
##         15         16         17         18         19         20         21
##  0.2903099  0.8877763  0.4716697 -2.8989890 -3.0187917  0.0981005 -0.5213755
##         22         23         24         25         26         27         28
## -1.6188512  0.0401696 -0.3364337  0.5277415 -0.8568544 -2.4864215  1.4713005
##         29         30         31         32         33         34         35
##  0.8430904  2.3893620  3.1756259  3.4334821  2.2521330  2.5500550  3.8593747
##         36         37         38         39         40         41         42
##  1.7927411  4.8023837 -1.9867946 -1.3902506 -1.6571742 -2.5800300 -2.8988297
##         43         44         45         46         47         48         49
## -2.6380174  2.4771486 -2.2121934 -2.8679926  3.3384914  4.8217224 -0.2739838
##         50         51         52         53         54         55         56
##  1.8302782 -0.4458668  0.6315437  0.7608952 -2.9966988 -3.1350646  2.8029581
##         57         58
## -1.1729214  1.6934809
```

Order Logit Model

```
# full sample
data <- read.csv('C:/Users/user/Desktop/      /Project 1/New Data/SP500_change_V7Final.csv')
data <- data[, -1]
factor_summary <- summary(data[1:35])
factor_cov <- cov(data[1:35])

# Train and test data
data_train_df70 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/train_df70.csv")
data_train_df70 <- data_train_df70[, -1]
data_train_df70$rating_diff <- factor(data_train_df70$rating_diff, ordered = TRUE)
data_test_df30 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/test_df30.csv")
data_test_df30 <- data_test_df30[, -1]
column_names <- colnames(data)

data$rating_diff <- factor(data$rating_diff, ordered = TRUE)

# According to Backward model selection with OLS...
model_logit <- polr(rating_diff ~ total.debt..total.assets_change +
                    interest..average.total.debt_change +
                    price..sales_change +
                    interest.coverage.ratio_change +
                    gross.profit.margin_change +
                    gross.profit..total.assets_change,
                    data=data, Hess = TRUE)
summary(model_logit)
```

```
## Call:
## polr(formula = rating_diff ~ total.debt..total.assets_change +
##     interest..average.total.debt_change + price..sales_change +
##     interest.coverage.ratio_change + gross.profit.margin_change +
##     gross.profit..total.assets_change, data = data, Hess = TRUE)
```

```
##
## Coefficients:
##                                         Value Std. Error t value
## total.debt..total.assets_change      -1.59497    0.59098  -2.699
## interest..average.total.debt_change -1.73352    0.60158  -2.882
## price..sales_change                   0.34176    0.18033   1.895
## interest.coverage.ratio_change        0.07923    0.03011   2.632
## gross.profit.margin_change           -1.18456    0.53487  -2.215
## gross.profit..total.assets_change     1.18823    0.45914   2.588
##
## Intercepts:
##        Value   Std. Error t value
## -11|-6 -5.5560  1.0242    -5.4245
## -6|-5  -4.1592  0.5478    -7.5927
## -5|-4  -3.5566  0.4307    -8.2580
## -4|-3  -2.8080  0.3165    -8.8711
## -3|-2  -1.3160  0.1875    -7.0189
## -2|-1  -0.4505  0.1600    -2.8160
## -1|1    0.1372  0.1574     0.8717
## 1|2     0.6595  0.1647     4.0034
## 2|3     1.8147  0.2104     8.6244
## 3|4     3.4965  0.3947     8.8593
## 4|5     4.0734  0.5126     7.9459
## 5|6     4.7778  0.7161     6.6723
##
## Residual Deviance: 790.6027
## AIC: 826.6027
```

```r
# find factors forward
data <- read.csv('C:/Users/user/Desktop/      /Project 1/New Data/SP500_change_V7Final.csv')
# data <- data[, !(names(data) %in% "asset.turnover_change")]
data <- data[, -1]
data$rating_diff <- factor(data$rating_diff, ordered = TRUE)
column_names <- colnames(data)

# Stepwise selection
stepwise_model_selection <- function(data, column_names, base_formula) {
  best_aic <- Inf
  best_factor <- NA
  best_model <- NULL

  for (i in column_names) {
    # Update formula
    formula <- as.formula(paste(base_formula, i, sep = " + "))
    model_logit_ <- polr(formula, data = data, Hess = TRUE)
    aic <- AIC(model_logit_)

    if (aic < best_aic) {
      best_aic <- aic
      best_factor <- i
      best_model <- model_logit_
    }
  }
  return(list(model = best_model, aic = best_aic, factor = best_factor))
```

```r
}

# Initialization
base_formula <- "rating_diff ~"

# Initial set of column names
initial_column_names <- column_names[2:35]

# Iterate
for (step in 1:10) {
  result <- stepwise_model_selection(data, initial_column_names, base_formula)

  base_formula <- paste(base_formula, result$factor, sep = " + ")
  initial_column_names <- setdiff(initial_column_names, result$factor)

  print(paste("step", step, ": the smallest AIC:", result$aic, "factor:", result$factor))
}
```

```
## [1] "step 1 : the smallest AIC: 834.470144657707 factor: total.debt..total.assets_change"
## [1] "step 2 : the smallest AIC: 831.683404446406 factor: interest..average.total.debt_change"
## [1] "step 3 : the smallest AIC: 830.19722164248 factor: asset.turnover_change"
## [1] "step 4 : the smallest AIC: 828.751647036692 factor: price..sales_change"
## [1] "step 5 : the smallest AIC: 828.615769199632 factor: interest.coverage.ratio_change"
## [1] "step 6 : the smallest AIC: 828.818322688303 factor: gross.profit..total.assets_change"
## [1] "step 7 : the smallest AIC: 828.578125229504 factor: gross.profit.margin_change"
## [1] "step 8 : the smallest AIC: 827.566811163104 factor: receivables.turnover_change"
## [1] "step 9 : the smallest AIC: 827.966905067553 factor: cash.flow.margin_change"
## [1] "step 10 : the smallest AIC: 828.440806104208 factor: sales.invested.capital_change"
```

According to Backward model selection with AIC Selection (w/o asset.turnover_change, cause we find out that 'asset.turnover_change' will lead to a bigger AIC)...

```r
model_logit <- polr(rating_diff ~ total.debt..total.assets_change +
                    interest..average.total.debt_change +
                    price..sales_change +
                    interest.coverage.ratio_change +
                    gross.profit.margin_change +
                    gross.profit..total.assets_change +
                    receivables.turnover_change,
                    data=data, Hess = TRUE)
summary(model_logit)
```

```
## Call:
## polr(formula = rating_diff ~ total.debt..total.assets_change +
##     interest..average.total.debt_change + price..sales_change +
##     interest.coverage.ratio_change + gross.profit.margin_change +
##     gross.profit..total.assets_change + receivables.turnover_change,
##     data = data, Hess = TRUE)
##
## Coefficients:
##                                         Value Std. Error t value
## total.debt..total.assets_change     -1.78902    0.61970  -2.887
```

```
## interest..average.total.debt_change -1.79166    0.61805  -2.899
## price..sales_change                   0.38670    0.18131   2.133
## interest.coverage.ratio_change        0.09933    0.03244   3.062
## gross.profit.margin_change           -1.57562    0.58105  -2.712
## gross.profit..total.assets_change     1.60698    0.51484   3.121
## receivables.turnover_change          -0.46134    0.25505  -1.809
##
## Intercepts:
##        Value   Std. Error t value
## -11|-6 -5.6035  1.0240    -5.4721
## -6|-5  -4.2067  0.5479    -7.6773
## -5|-4  -3.6089  0.4331    -8.3328
## -4|-3  -2.8599  0.3197    -8.9447
## -3|-2  -1.3587  0.1902    -7.1437
## -2|-1  -0.4847  0.1621    -2.9912
## -1|1    0.1084  0.1591     0.6813
## 1|2     0.6360  0.1663     3.8249
## 2|3     1.8080  0.2118     8.5371
## 3|4     3.4963  0.3953     8.8444
## 4|5     4.0739  0.5131     7.9390
## 5|6     4.7791  0.7164     6.6713
##
## Residual Deviance: 787.5671
## AIC: 825.5671
```

```r
# train 70-30
model_logit_70 <- polr(rating_diff ~ total.debt..total.assets_change +
                       interest..average.total.debt_change +
                       price..sales_change +
                       interest.coverage.ratio_change +
                       gross.profit.margin_change +
                       gross.profit..total.assets_change +
                       receivables.turnover_change,
                       data = data_train_df70, Hess = TRUE)
summary(model_logit_70)
```

```
## Call:
## polr(formula = rating_diff ~ total.debt..total.assets_change +
##     interest..average.total.debt_change + price..sales_change +
##     interest.coverage.ratio_change + gross.profit.margin_change +
##     gross.profit..total.assets_change + receivables.turnover_change,
##     data = data_train_df70, Hess = TRUE)
##
## Coefficients:
##                                       Value Std. Error t value
## total.debt..total.assets_change     -2.211    0.69058  -3.201
## interest..average.total.debt_change -2.436    0.81078  -3.004
## price..sales_change                   0.353    0.21072   1.675
## interest.coverage.ratio_change        0.120    0.03803   3.155
## gross.profit.margin_change          -2.127    0.78947  -2.694
## gross.profit..total.assets_change     2.123    0.70504   3.012
## receivables.turnover_change         -0.614    0.27902  -2.201
##
## Intercepts:
```

```
##         Value   Std. Error t value
## -11|-6 -5.2605  1.0356    -5.0796
## -6|-5  -4.1532  0.6384    -6.5051
## -5|-4  -3.5753  0.5046    -7.0850
## -4|-3  -3.0044  0.3968    -7.5718
## -3|-2  -1.4476  0.2318    -6.2441
## -2|-1  -0.6221  0.1975    -3.1506
## -1|1    0.1131  0.1917     0.5898
## 1|2     0.5712  0.1990     2.8707
## 2|3     1.8426  0.2561     7.1944
## 3|4     3.2021  0.4072     7.8637
## 4|5     3.7887  0.5224     7.2519
## 5|6     4.5021  0.7235     6.2231
##
## Residual Deviance: 550.8166
## AIC: 588.8166
```

```r
# test 70-30
required_columns <- c("interest.coverage.ratio_change",
                      "total.debt..total.assets_change",
                      "interest..average.total.debt_change",
                      "price..sales_change",
                      "gross.profit.margin_change",
                      "gross.profit..total.assets_change",
                      "receivables.turnover_change")
test_df30 <- data_test_df30[required_columns]
predictors_30 <- predict(model_logit_70, type='class', newdata=test_df30)
temp_df <- data.frame(predict = predictors_30, actual = data_test_df30$rating_diff, stringsAsFactors =
temp_df$predict <- as.numeric(temp_df$predict)
temp_df$actual <- as.numeric(temp_df$actual)
print(temp_df$predict - temp_df$actual)
```

```
##  [1]  6  3  7 11  7  8  8  7  6 11  0  7  2  6 10 10 10  6  6  6  8  8  8  7 10
## [26]  8  7 11  7 11 12 12  8 12 13  8 14  7  7  5  6  6  6  8  4  5 12 13  6 11
## [51]  6 10  7  3  6 12  7 11
```

```r
# Totally inaccurate
```

Order Probit Model

```r
# full sample
data <- read.csv('C:/Users/user/Desktop/      /Project 1/New Data/SP500_change_V7Final.csv')
data <- data[, -1]
factor_summary <- summary(data[1:35])
factor_cov <- cov(data[1:35])

# Train and test data
data_train_df70 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/train_df70.csv")
data_train_df70 <- data_train_df70[, -1]
data_train_df70$rating_diff <- factor(data_train_df70$rating_diff, ordered = TRUE)
```

```r
data_test_df30 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/test_df30.csv")
data_test_df30 <- data_test_df30[, -1]
column_names <- colnames(data)

data$rating_diff <- factor(data$rating_diff, ordered = TRUE)

# According to Backward model selection with OLS...
model_probit <- polr(rating_diff ~ total.debt..total.assets_change +
                     interest..average.total.debt_change +
                     price..sales_change +
                     interest.coverage.ratio_change +
                     gross.profit.margin_change +
                     gross.profit..total.assets_change,
                     method = 'probit', data=data, Hess = TRUE)
summary(model_probit)
```

```
## Call:
## polr(formula = rating_diff ~ total.debt..total.assets_change +
##     interest..average.total.debt_change + price..sales_change +
##     interest.coverage.ratio_change + gross.profit.margin_change +
##     gross.profit..total.assets_change, data = data, Hess = TRUE,
##     method = "probit")
##
## Coefficients:
##                                        Value Std. Error t value
## total.debt..total.assets_change     -0.76761     0.3604  -2.130
## interest..average.total.debt_change -0.90466     0.3517  -2.572
## price..sales_change                  0.17482     0.1138   1.537
## interest.coverage.ratio_change       0.04559     0.0191   2.386
## gross.profit.margin_change          -0.61970     0.3212  -1.930
## gross.profit..total.assets_change    0.64154     0.2710   2.367
##
## Intercepts:
##        Value    Std. Error t value
## -11|-6  -2.6038   0.3325    -7.8317
## -6|-5   -2.1240   0.2170    -9.7898
## -5|-4   -1.8780   0.1839   -10.2123
## -4|-3   -1.5484   0.1493   -10.3699
## -3|-2   -0.7833   0.1085    -7.2207
## -2|-1   -0.2731   0.0992    -2.7532
## -1|1     0.0834   0.0986     0.8464
## 1|2      0.3994   0.1016     3.9323
## 2|3      1.0735   0.1179     9.1021
## 3|4      1.9089   0.1765    10.8130
## 4|5      2.1521   0.2110    10.1977
## 5|6      2.4261   0.2691     9.0144
##
## Residual Deviance: 795.4636
## AIC: 831.4636
```

```r
# find factors forward
data <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/SP500_change_V7Final.csv")
# data <- data[, !(names(data) %in% "asset.turnover_change")]
```

```r
data <- data[, -1]
data$rating_diff <- factor(data$rating_diff, ordered = TRUE)
column_names <- colnames(data)
```

It turns out that the probit model will fail to converge. Since the stepwise selection process is roughly similar to logit, the explanatory variables selected by logit are directly used in the follow-up.

```r
# Stepwise selection
stepwise_model_selection <- function(data, column_names, base_formula) {
  best_aic <- Inf
  best_factor <- NA
  best_model <- NULL

  for (i in column_names) {
    # Update formula
    formula <- as.formula(paste(base_formula, i, sep = " + "))
    model_probit_ <- polr(formula, method = 'probit', data = data, Hess = TRUE,
                          control = list(maxit = 50, reltol = 1e-3))
    aic <- AIC(model_probit_)

    if (aic < best_aic) {
      best_aic <- aic
      best_factor <- i
      best_model <- model_probit_
    }
  }
  return(list(model = best_model, aic = best_aic, factor = best_factor))
}

# Initialization
base_formula <- "rating_diff ~"

# Initial set of column names
initial_column_names <- column_names[2:35]

# Iterate
# for (step in 1:10) {
#   result <- stepwise_model_selection(data, initial_column_names, base_formula)

#   base_formula <- paste(base_formula, result$factor, sep = " + ")
#   initial_column_names <- setdiff(initial_column_names, result$factor)

#   print(paste("step", step, ": the smallest AIC:", result$aic, "factor:", result$factor))
# }
```

According to Backward model selection with AIC Selection(w/o asset.turnover_change, cause we find out that 'asset.turnover_change' will lead to a bigger AIC)...

```r
model_probit <- polr(rating_diff ~ total.debt..total.assets_change +
                     interest..average.total.debt_change +
                     price..sales_change +
                     interest.coverage.ratio_change +
                     gross.profit.margin_change +
```

```r
                     gross.profit..total.assets_change +
                     receivables.turnover_change,
                     method = 'probit', data=data, Hess = TRUE)
summary(model_probit)
```

```
## Call:
## polr(formula = rating_diff ~ total.debt..total.assets_change +
##     interest..average.total.debt_change + price..sales_change +
##     interest.coverage.ratio_change + gross.profit.margin_change +
##     gross.profit..total.assets_change + receivables.turnover_change,
##     data = data, Hess = TRUE, method = "probit")
##
## Coefficients:
##                                        Value Std. Error t value
## total.debt..total.assets_change     -0.87644    0.36605  -2.394
## interest..average.total.debt_change -0.91899    0.35196  -2.611
## price..sales_change                  0.20475    0.11514   1.778
## interest.coverage.ratio_change       0.05807    0.02046   2.838
## gross.profit.margin_change          -0.85863    0.35036  -2.451
## gross.profit..total.assets_change    0.90323    0.31138   2.901
## receivables.turnover_change         -0.28463    0.16648  -1.710
##
## Intercepts:
##        Value    Std. Error t value
## -11|-6  -2.6323   0.3345    -7.8701
## -6|-5   -2.1488   0.2174    -9.8862
## -5|-4   -1.9039   0.1847   -10.3064
## -4|-3   -1.5744   0.1505   -10.4645
## -3|-2   -0.8066   0.1096    -7.3623
## -2|-1   -0.2936   0.1001    -2.9320
## -1|1     0.0650   0.0994     0.6544
## 1|2      0.3838   0.1023     3.7522
## 2|3      1.0679   0.1186     9.0046
## 3|4      1.9096   0.1772    10.7743
## 4|5      2.1535   0.2117    10.1708
## 5|6      2.4314   0.2712     8.9649
##
## Residual Deviance: 792.5395
## AIC: 830.5395
```

```r
# train 70-30
model_probit_70 <- polr(rating_diff ~ total.debt..total.assets_change +
                        interest..average.total.debt_change +
                        price..sales_change +
                        interest.coverage.ratio_change +
                        gross.profit.margin_change +
                        gross.profit..total.assets_change +
                        receivables.turnover_change,
                        method = 'probit', data = data_train_df70, Hess = TRUE)
summary(model_probit_70)
```

```
## Call:
## polr(formula = rating_diff ~ total.debt..total.assets_change +
```

```
##     interest..average.total.debt_change + price..sales_change +
##     interest.coverage.ratio_change + gross.profit.margin_change +
##     gross.profit..total.assets_change + receivables.turnover_change,
##     data = data_train_df70, Hess = TRUE, method = "probit")
##
## Coefficients:
##                                        Value Std. Error t value
## total.debt..total.assets_change     -1.04339    0.42306  -2.466
## interest..average.total.debt_change -1.09043    0.45227  -2.411
## price..sales_change                  0.17824    0.13315   1.339
## interest.coverage.ratio_change       0.06509    0.02271   2.866
## gross.profit.margin_change          -1.08408    0.45265  -2.395
## gross.profit..total.assets_change    1.11424    0.40238   2.769
## receivables.turnover_change         -0.34763    0.17761  -1.957
##
## Intercepts:
##        Value   Std. Error t value
## -11|-6 -2.5036  0.3492    -7.1685
## -6|-5  -2.1188  0.2541    -8.3382
## -5|-4  -1.8877  0.2171    -8.6941
## -4|-3  -1.6412  0.1844    -8.8990
## -3|-2  -0.8554  0.1320    -6.4825
## -2|-1  -0.3786  0.1203    -3.1460
## -1|1    0.0602  0.1186     0.5072
## 1|2     0.3327  0.1214     2.7404
## 2|3     1.0663  0.1418     7.5213
## 3|4     1.7582  0.1909     9.2110
## 4|5     2.0167  0.2250     8.9644
## 5|6     2.3112  0.2860     8.0809
##
## Residual Deviance: 555.9103
## AIC: 593.9103
```

```r
# test 70-30
required_columns <- c("interest.coverage.ratio_change",
                      "total.debt..total.assets_change",
                      "interest..average.total.debt_change",
                      "price..sales_change",
                      "gross.profit.margin_change",
                      "gross.profit..total.assets_change",
                      "receivables.turnover_change")
test_df30 <- data_test_df30[required_columns]
predictors_30 <- predict(model_probit_70, type='class', newdata=test_df30)
temp_df <- data.frame(predict = predictors_30, actual = data_test_df30$rating_diff, stringsAsFactors = 
temp_df$predict <- as.numeric(temp_df$predict)
temp_df$actual <- as.numeric(temp_df$actual)
print(temp_df$predict - temp_df$actual)
```

```
##  [1]  6  3  7 11  7  8  8  7  6 11  0  7  6  6 10 10 10  6  6  6  8  8  8  7 10
## [26]  8  7 11  7 11 12 12  8 12 13  8 14  7  7  5  6  6  6  8  4  5 12 13  6 11
## [51]  6 10  7  3  6 12  7 11
```

```r
# totally inaccurate

# stargazer(model_probit, type = "latex")
```

Machine Learning Models

```r
library(MASS)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-7
```

```r
library(ggplot2)
data <- read.csv('C:/Users/user/Desktop/      /Project 1/New Data/SP500_change_V7Final.csv')
data <- data[, -1]
# Train and test data
data_train_60 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/train_df60.csv")
data_train_70 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/train_df70.csv")
data_test_40 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/test_df40.csv")
data_test_30 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/test_df30.csv")
```

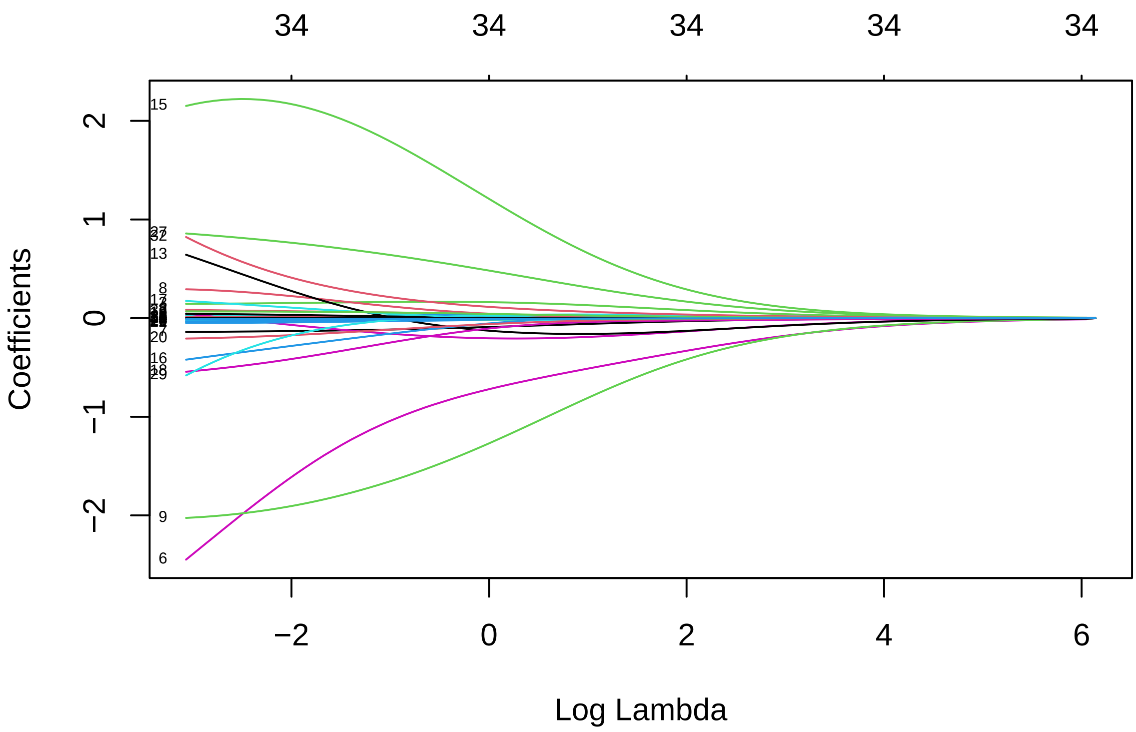
Ridge Regression

Full sample

```r
predictors <- data[, -1] %>% as.matrix()
ridge_model <- glmnet(predictors, data$rating_diff, alpha = 0)
summary(ridge_model)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      3400   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         4   -none-    call
## nobs         1   -none-    numeric
```

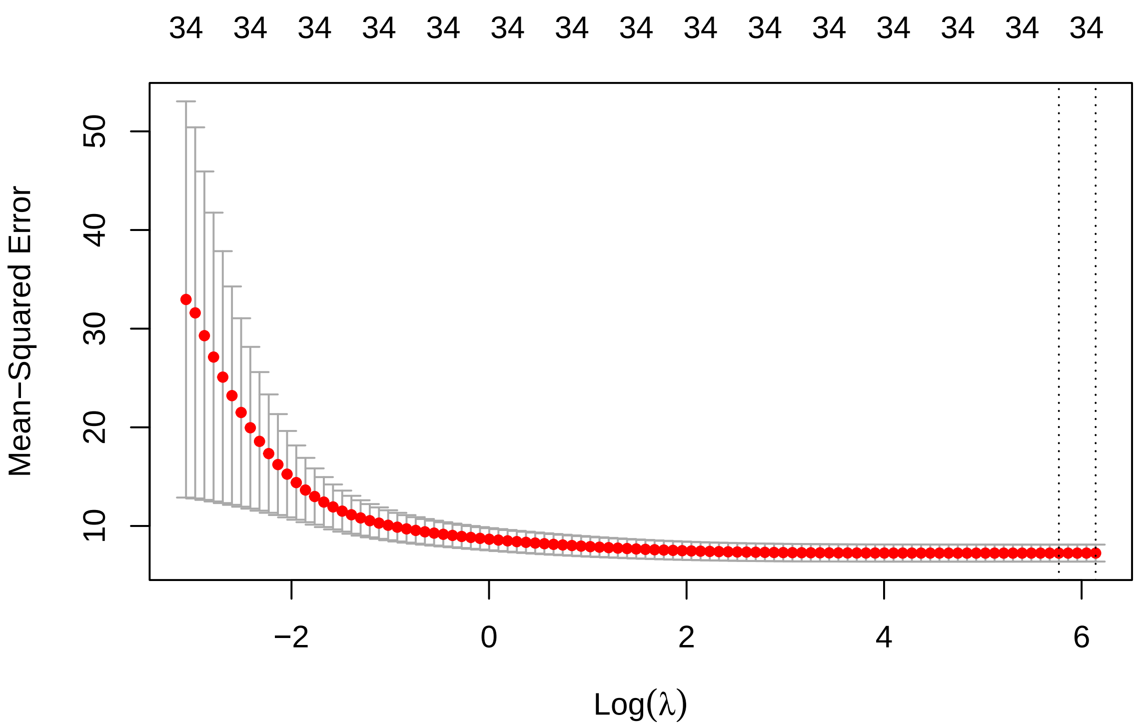
```r
plot(ridge_model, xvar = "lambda", label = TRUE)
```




```r
    # With a cross-validation
cv.ridge <- cv.glmnet(predictors, data$rating_diff, alpha = 0)
bestlam_rr <- cv.ridge$lambda.min
bestlam_rr
```

```
## [1] 320.5601
```

```r
plot(cv.ridge)
```




```r
cv_ridge <- glmnet(predictors, data$rating_diff, alpha = 0, lambda = bestlam_rr)
coef(cv_ridge)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                          s0
## (Intercept)                                   -1.700361e-01
## after.tax.interest.coverage_change             2.010891e-04
## interest.coverage.ratio_change                 2.138301e-04
## cash.flow..total.debt_change                   3.121036e-03
## operating.margin.before.dep._change            7.858672e-04
## return.on.equity_change                        8.201290e-05
## total.debt..total.assets_change               -1.551752e-02
## book..market_change                           -9.703193e-04
## interest..average.LTD_change                  -9.237409e-05
## interest..average.total.debt_change           -1.334395e-02
## cash.balance..total.liabilities_change         2.517161e-04
## free.cash.flow..operating.cash.flow_change    -5.235882e-05
## total.liabilities..total.tangible.assets_change -6.393789e-03
## total.debt..capital_change                    -6.893189e-03
## total.debt..equity_change                      4.379105e-05
## asset.turnover_change                          6.403389e-03
```

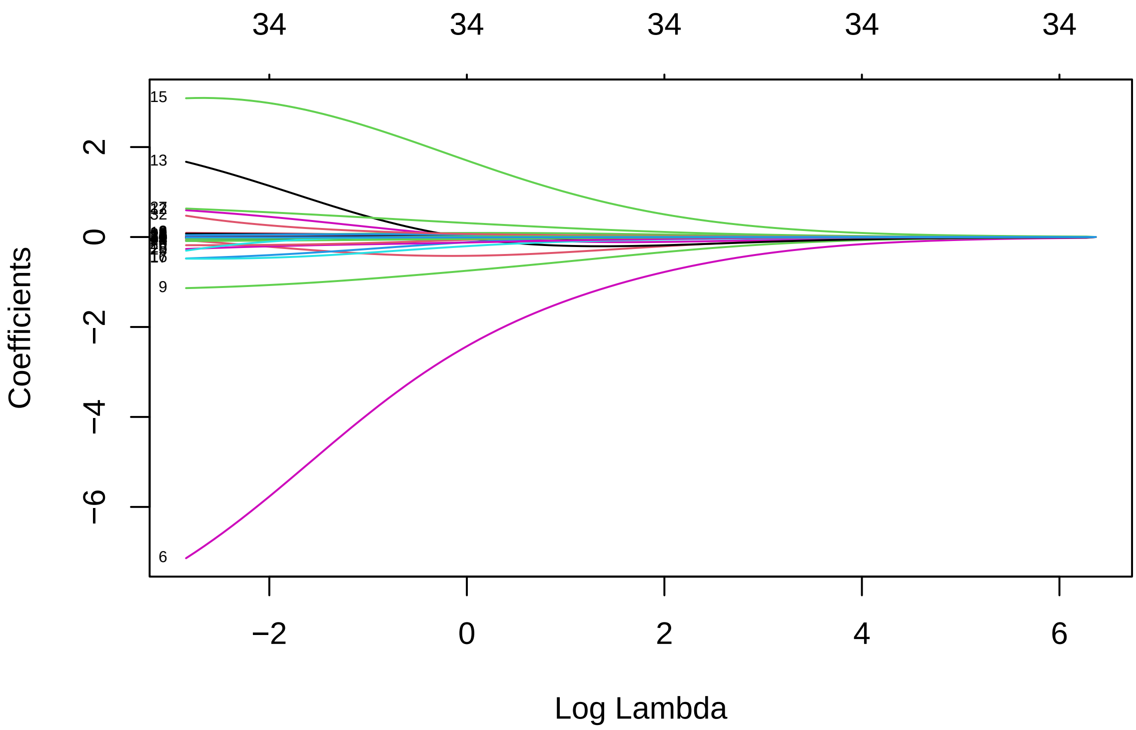
```
## receivables.turnover_change                                   1.705922e-03
## payables.turnover_change                                     -1.345853e-03
## sales.invested.capital_change                                -1.794903e-03
## sales.stockholders.equity_change                             -1.580487e-04
## price..book_change                                           -3.420780e-04
## shillers.cyclically.adjusted.P.E.ratio_change                -1.107359e-04
## enterprise.value.multiple_change                              9.693978e-05
## price..operating.earnings..Basic..Excl..EI._change           -1.024881e-05
## price..operating.earnings..Diluted..Excl..EI._change         -1.012674e-05
## P.E..Diluted..Excl..EI._change                               -8.157970e-06
## P.E..Diluted..Incl..EI._change                               -9.556681e-06
## price..sales_change                                           5.641634e-03
## price..cash.flow_change                                       7.740261e-05
## gross.profit.margin_change                                    1.658132e-03
## after.tax.return.on.average.common.equity_change             -4.960637e-06
## after.tax.return.on.average.stockholders..equity_change -5.558821e-06
## gross.profit..total.assets_change                             1.570693e-03
## common.equity.invested.capital_change                         4.713646e-04
## cash.flow.margin_change                                      -6.517698e-05
```

60-40

```
predictors_60 <- cbind.data.frame(data_train_60[, 3:36]) %>% as.matrix()
ridge_60 <- glmnet(predictors_60, data_train_60$rating_diff, alpha = 0)
summary(ridge_60)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      3400   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         4   -none-    call
## nobs         1   -none-    numeric
```

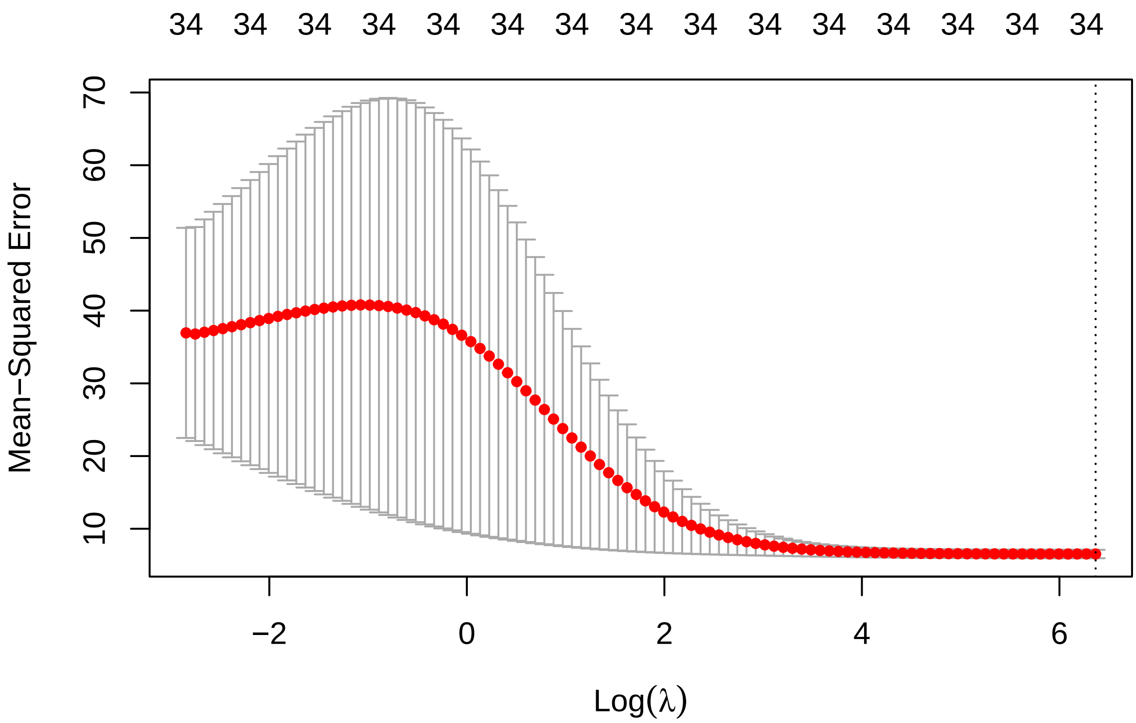
```
plot(ridge_60, xvar = "lambda", label = TRUE)
```




```r
    # With a cross-validation
cv.ridge_60 <- cv.glmnet(predictors_60, data_train_60$rating_diff, alpha = 0)
bestlam_rr_60 <- cv.ridge_60$lambda.min
bestlam_rr_60
```

```
## [1] 582.1793
```

```r
plot(cv.ridge_60)
```




```r
cv_ridge_60 <- glmnet(predictors_60, data_train_60$rating_diff, alpha = 0, lambda = bestlam_rr_60)
coef(cv_ridge_60)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                              s0
## (Intercept)                                       -3.445787e-01
## after.tax.interest.coverage_change                 2.085953e-05
## interest.coverage.ratio_change                    -2.965919e-06
## cash.flow..total.debt_change                       1.241256e-03
## operating.margin.before.dep._change                3.090257e-04
## return.on.equity_change                            1.352303e-04
## total.debt..total.assets_change                   -1.638612e-02
## book..market_change                               -2.876909e-04
## interest..average.LTD_change                      -4.235512e-03
## interest..average.total.debt_change               -7.305369e-03
## cash.balance..total.liabilities_change             2.110322e-04
## free.cash.flow..operating.cash.flow_change        -6.710577e-05
## total.liabilities..total.tangible.assets_change   -3.567137e-03
## total.debt..capital_change                        -5.462355e-03
## total.debt..equity_change                          6.355823e-05
## asset.turnover_change                              8.946196e-03
## receivables.turnover_change                        1.044486e-03
## payables.turnover_change                          -1.556949e-03
## sales.invested.capital_change                     -9.189477e-04
## sales.stockholders.equity_change                   1.630947e-04
```

```
## price..book_change                                       3.500175e-04
## shillers.cyclically.adjusted.P.E.ratio_change           -2.145823e-04
## enterprise.value.multiple_change                         1.174292e-04
## price..operating.earnings..Basic..Excl..EI._change      -4.228504e-06
## price..operating.earnings..Diluted..Excl..EI._change    -4.211026e-06
## P.E..Diluted..Excl..EI._change                          -3.113813e-06
## P.E..Diluted..Incl..EI._change                          -3.801510e-06
## price..sales_change                                      2.277711e-03
## price..cash.flow_change                                  3.144412e-04
## gross.profit.margin_change                               7.658946e-04
## after.tax.return.on.average.common.equity_change         1.009089e-04
## after.tax.return.on.average.stockholders..equity_change  1.004139e-04
## gross.profit..total.assets_change                        7.662566e-04
## common.equity.invested.capital_change                    2.231510e-04
## cash.flow.margin_change                                 -3.237365e-05
```

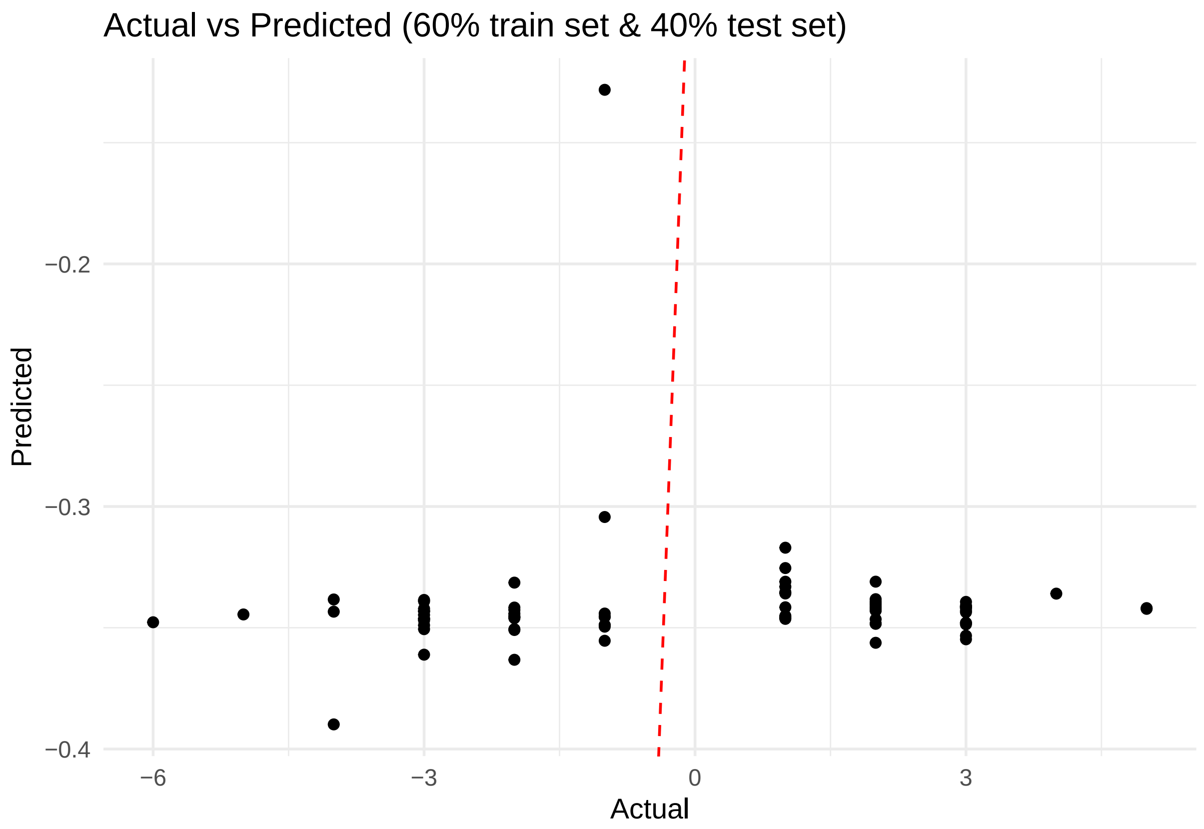
```
    # Prediction
test_60 <- cbind.data.frame(data_test_40[, 3:36]) %>% as.matrix()
rr_60_pred <- predict(cv_ridge_60, s = bestlam_rr_60, newx = test_60)
print(rr_60_pred)
```

```
##               s1
##  [1,] -0.3452111
##  [2,] -0.3420559
##  [3,] -0.3419112
##  [4,] -0.3464755
##  [5,] -0.3416695
##  [6,] -0.3462430
##  [7,] -0.3391037
##  [8,] -0.3422019
##  [9,] -0.3393275
## [10,] -0.3456910
## [11,] -0.3385336
## [12,] -0.3457629
## [13,] -0.3450970
## [14,] -0.3429676
## [15,] -0.3532896
## [16,] -0.3477299
## [17,] -0.3505743
## [18,] -0.3330883
## [19,] -0.3455303
## [20,] -0.3480240
## [21,] -0.3562011
## [22,] -0.3505731
## [23,] -0.3426374
## [24,] -0.3414605
## [25,] -0.3253916
## [26,] -0.3310152
## [27,] -0.3428087
## [28,] -0.3435878
## [29,] -0.3456517
## [30,] -0.3418979
## [31,] -0.3405523
## [32,] -0.3485856
```

```
## [33,] -0.3430486
## [34,] -0.3495466
## [35,] -0.3043442
## [36,] -0.3489001
## [37,] -0.3412717
## [38,] -0.3547558
## [39,] -0.3553651
## [40,] -0.3169985
## [41,] -0.3463461
## [42,] -0.3353214
## [43,] -0.3309984
## [44,] -0.3441366
## [45,] -0.3358600
## [46,] -0.3482702
## [47,] -0.3456187
## [48,] -0.3632052
## [49,] -0.3313906
## [50,] -0.3421442
## [51,] -0.3432024
## [52,] -0.3489792
## [53,] -0.3469931
## [54,] -0.3433622
## [55,] -0.3610949
## [56,] -0.3445061
## [57,] -0.3392504
## [58,] -0.3398042
## [59,] -0.3898401
## [60,] -0.3408574
## [61,] -0.3414849
## [62,] -0.3414805
## [63,] -0.3447070
## [64,] -0.3415546
## [65,] -0.3359250
## [66,] -0.3431579
## [67,] -0.3383413
## [68,] -0.3488727
## [69,] -0.3442081
## [70,] -0.1282169
## [71,] -0.3454595
## [72,] -0.3509242
## [73,] -0.3483779
## [74,] -0.3479712
## [75,] -0.3463095
## [76,] -0.3382154
## [77,] -0.3460480
```

```
    # Plot
plot_rr_60 <- data.frame(Actual = data_test_40$rating_diff, Predicted = as.vector(rr_60_pred))
ggplot(plot_rr_60, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted (60% train set & 40% test set)",
       x = "Actual",
       y = "Predicted") +
```

```r
      theme_minimal()
```




```r
    # Evaluation
pe_rr_60<- plot_rr_60$Predicted - plot_rr_60$Actual
rmse_rr_60 <- sqrt(mean(pe_rr_60^2))
cat("Ridge (60-40) RMSE:", rmse_rr_60, "\n")
```

```
## Ridge (60-40) RMSE: 2.612516
```

```r
table_rr_60 <- data.frame(Model = "Ridge Regression (60-40)",
                          RMSE = rmse_rr_60)
print(table_rr_60)
```

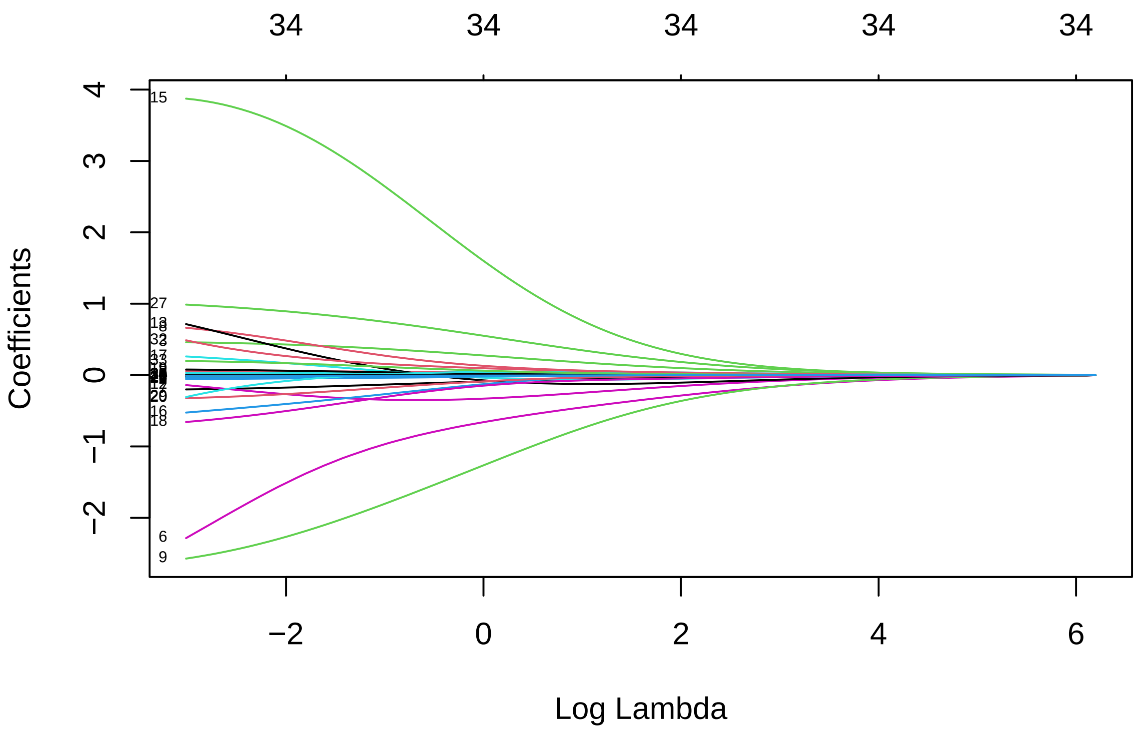
```
##                      Model     RMSE
## 1 Ridge Regression (60-40) 2.612516
```

70-30

```r
predictors_70 <- cbind.data.frame(data_train_70[, 3:36]) %>% as.matrix()
ridge_70 <- glmnet(predictors_70, data_train_70$rating_diff, alpha = 0)
summary(ridge_70)
```

```
##             Length Class     Mode
## a0          100    -none-    numeric
## beta        3400   dgCMatrix S4
## df          100    -none-    numeric
## dim           2    -none-    numeric
## lambda      100    -none-    numeric
## dev.ratio   100    -none-    numeric
## nulldev       1    -none-    numeric
## npasses       1    -none-    numeric
## jerr          1    -none-    numeric
## offset        1    -none-    logical
## call          4    -none-    call
## nobs          1    -none-    numeric
```

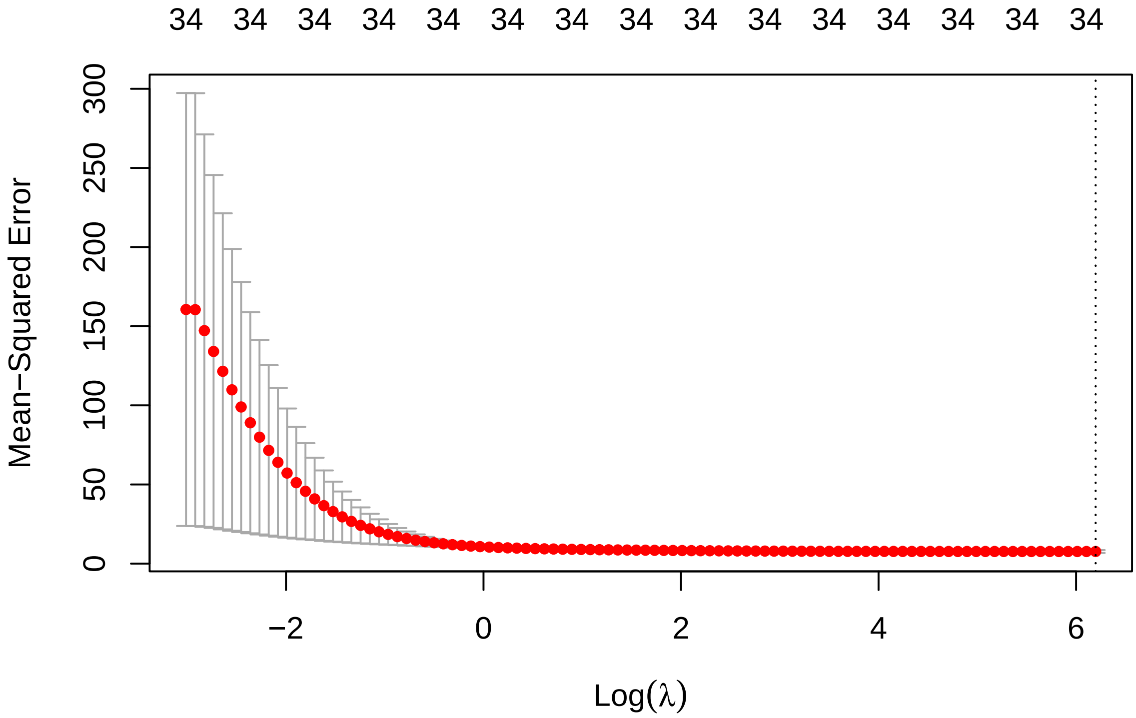
```r
plot(ridge_70, xvar = "lambda", label = TRUE)
```




```r
    # With a cross-validation
cv.ridge_70 <- cv.glmnet(predictors_70, data_train_70$rating_diff, alpha = 0)
bestlam_rr_70 <- cv.ridge_70$lambda.min
bestlam_rr_70
```

```
## [1] 491.8733
```

```r
plot(cv.ridge_70)
```




```r
cv_ridge_70 <- glmnet(predictors_70, data_train_70$rating_diff, alpha = 0, lambda = bestlam_rr_70)
coef(cv_ridge_70)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                           s0
## (Intercept)                                    -7.996963e-02
## after.tax.interest.coverage_change              7.241562e-05
## interest.coverage.ratio_change                  1.159779e-04
## cash.flow..total.debt_change                    2.442943e-03
## operating.margin.before.dep._change             5.104786e-04
## return.on.equity_change                         1.304144e-04
## total.debt..total.assets_change                -9.121911e-03
## book..market_change                            -6.441985e-04
## interest..average.LTD_change                    4.203256e-04
## interest..average.total.debt_change            -7.214086e-03
## cash.balance..total.liabilities_change          1.200824e-04
## free.cash.flow..operating.cash.flow_change      6.845061e-05
## total.liabilities..total.tangible.assets_change -4.483235e-03
## total.debt..capital_change                     -3.967314e-03
## total.debt..equity_change                       3.380183e-05
## asset.turnover_change                           3.386997e-03
## receivables.turnover_change                     8.104641e-04
## payables.turnover_change                       -1.725527e-03
```

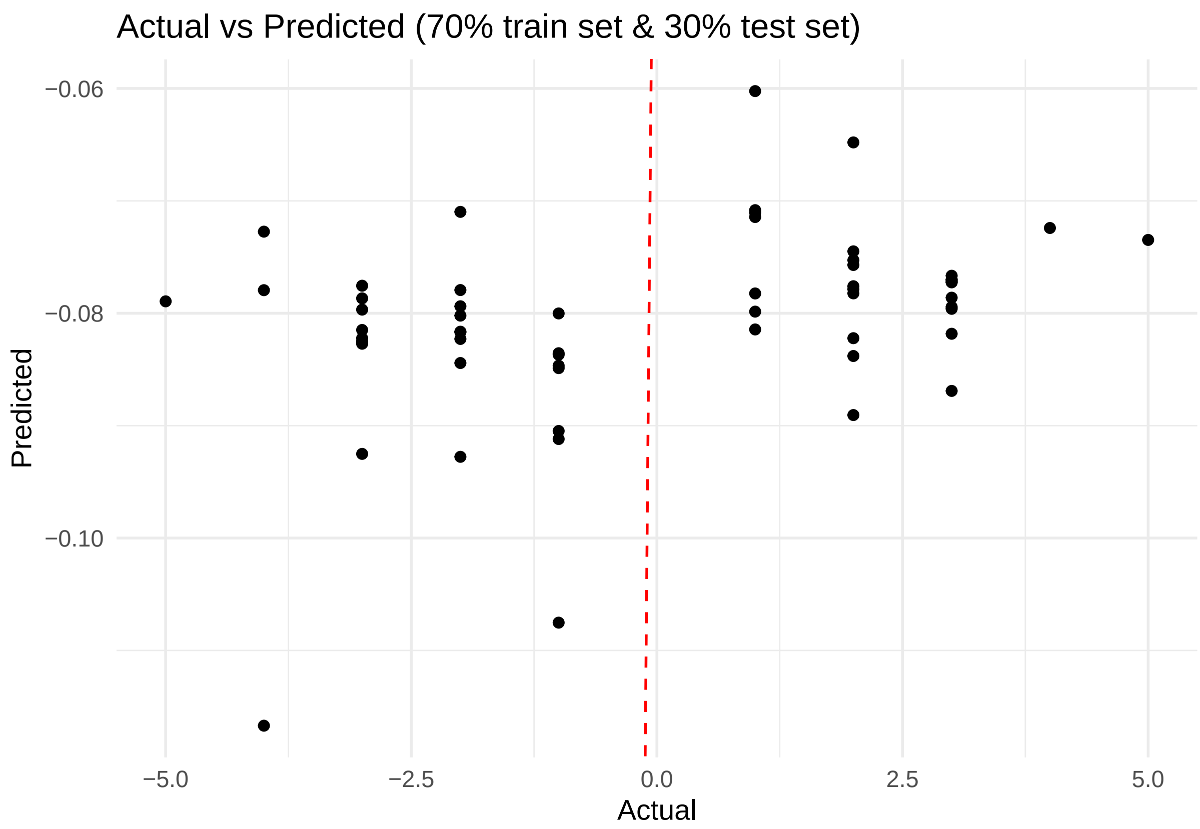
```
## sales.invested.capital_change                           -1.767295e-03
## sales.stockholders.equity_change                        -1.078417e-04
## price..book_change                                      -3.789025e-04
## shillers.cyclically.adjusted.P.E.ratio_change           -4.213292e-05
## enterprise.value.multiple_change                         6.902442e-05
## price..operating.earnings..Basic..Excl..EI._change      -7.416351e-06
## price..operating.earnings..Diluted..Excl..EI._change    -7.315965e-06
## P.E..Diluted..Excl..EI._change                          -6.105718e-06
## P.E..Diluted..Incl..EI._change                          -6.710715e-06
## price..sales_change                                      3.932516e-03
## price..cash.flow_change                                  7.411687e-06
## gross.profit.margin_change                               1.241050e-03
## after.tax.return.on.average.common.equity_change        -3.762490e-06
## after.tax.return.on.average.stockholders..equity_change -4.190315e-06
## gross.profit..total.assets_change                        1.059170e-03
## common.equity.invested.capital_change                    2.779346e-04
## cash.flow.margin_change                                 -4.911373e-05
```

```
    # Prediction
test_70 <- cbind.data.frame(data_test_30[, 3:36]) %>% as.matrix()
rr_70_pred <- predict(cv_ridge_70, s = bestlam_rr_70, newx = test_70)
print(rr_70_pred)
```

```
##                s1
##   [1,] -0.08182272
##   [2,] -0.08905626
##   [3,] -0.08442623
##   [4,] -0.07793432
##   [5,] -0.07822988
##   [6,] -0.06023422
##   [7,] -0.07143948
##   [8,] -0.07759568
##   [9,] -0.07943404
##  [10,] -0.08020965
##  [11,] -0.07347582
##  [12,] -0.07449043
##  [13,] -0.07960484
##  [14,] -0.07721637
##  [15,] -0.08488205
##  [16,] -0.08355643
##  [17,] -0.09118956
##  [18,] -0.07666111
##  [19,] -0.08690918
##  [20,] -0.09047370
##  [21,] -0.07082686
##  [22,] -0.08143952
##  [23,] -0.07104091
##  [24,] -0.06479936
##  [25,] -0.08466421
##  [26,] -0.07985271
##  [27,] -0.08380190
##  [28,] -0.08167265
##  [29,] -0.09277132
##  [30,] -0.07097985
```

```
## [31,] -0.07754829
## [32,] -0.07967676
## [33,] -0.08250690
## [34,] -0.08271056
## [35,] -0.07794613
## [36,] -0.09251063
## [37,] -0.07893883
## [38,] -0.07570357
## [39,] -0.07785836
## [40,] -0.11669459
## [41,] -0.07705079
## [42,] -0.07725223
## [43,] -0.07699332
## [44,] -0.08148561
## [45,] -0.07823680
## [46,] -0.07241357
## [47,] -0.07867271
## [48,] -0.07273971
## [49,] -0.08368542
## [50,] -0.07937825
## [51,] -0.10752716
## [52,] -0.08001472
## [53,] -0.08164104
## [54,] -0.08221879
## [55,] -0.07861265
## [56,] -0.08222343
## [57,] -0.07527708
## [58,] -0.08228637
```

```
    # Plot
plot_rr_70 <- data.frame(Actual = data_test_30$rating_diff, Predicted = as.vector(rr_70_pred))
ggplot(plot_rr_70, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted (70% train set & 30% test set)",
       x = "Actual",
       y = "Predicted") +
       theme_minimal()
```




```
# Evaluation
pe_rr_70<- plot_rr_70$Predicted - plot_rr_70$Actual
rmse_rr_70 <- sqrt(mean(pe_rr_70^2))
cat("Ridge (70-30) RMSE:", rmse_rr_70, "\n")
```

```
## Ridge (70-30) RMSE: 2.513331
```

```
table_rr_70 <- data.frame(Model = "Ridge Regression (70-30)",
                          RMSE = rmse_rr_70)
print(table_rr_70)
```

```
##                      Model     RMSE
## 1 Ridge Regression (70-30) 2.513331
```

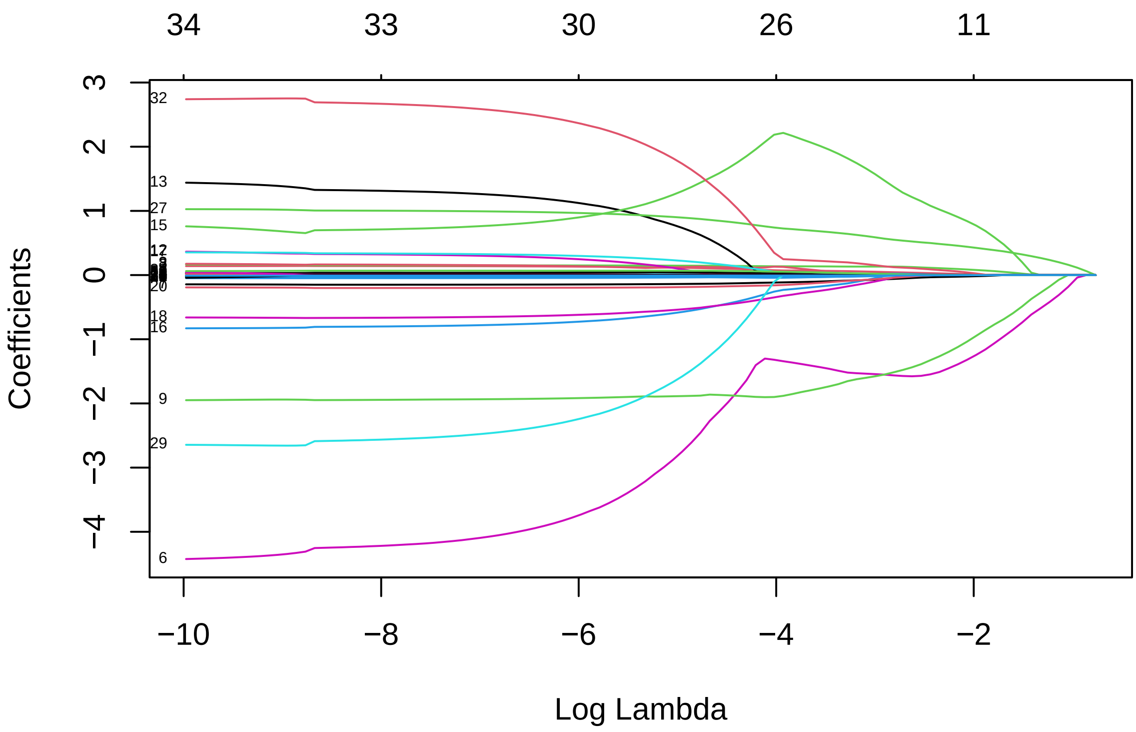
Lasso

Full sample

```
lasso_model <- glmnet(predictors, data$rating_diff, alpha = 1)
summary(lasso_model)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      3400   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         4   -none-    call
## nobs         1   -none-    numeric
```

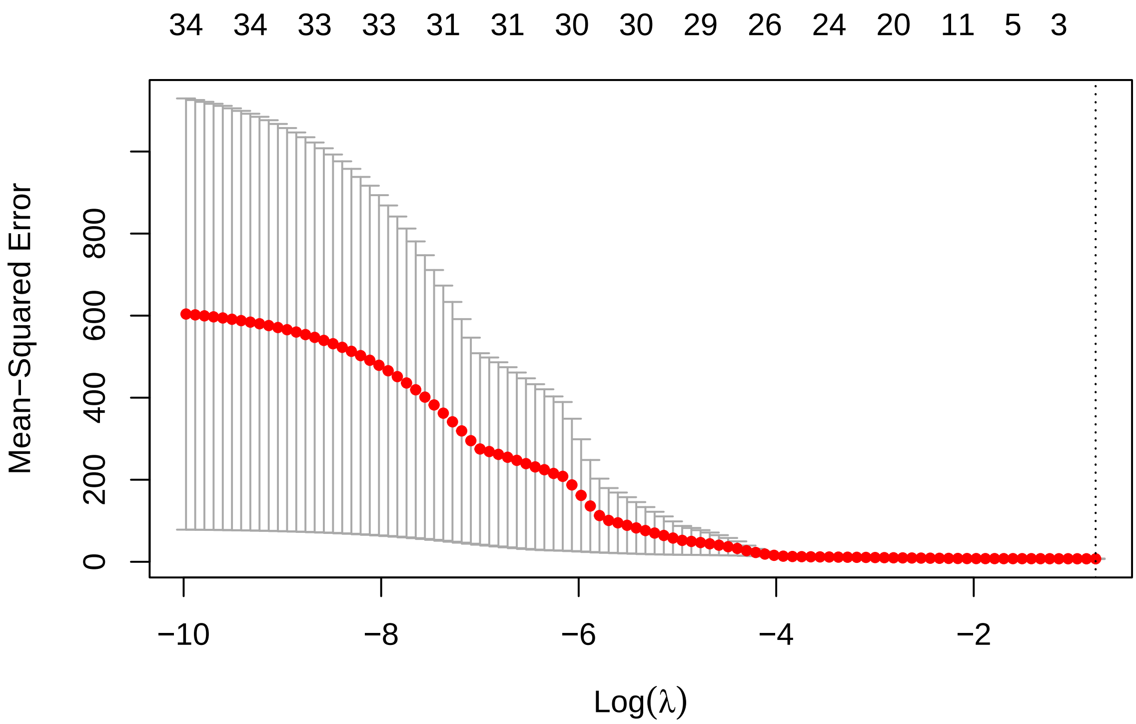
```
plot(lasso_model, xvar = "lambda", label = TRUE)
```




```
    # With a cross-validation
cv.lasso <- cv.glmnet(predictors, data$rating_diff, alpha = 1)
bestlam_la <- cv.lasso$lambda.min
bestlam_la
```

```
## [1] 0.4650778
```

```r
plot(cv.lasso)
```




```r
cv_lasso <- glmnet(predictors, data$rating_diff, alpha = 1, lambda = bestlam_la)
coef(cv_lasso)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                             s0
## (Intercept)                                         -0.1709845
## after.tax.interest.coverage_change                   0.0000000
## interest.coverage.ratio_change                       .
## cash.flow..total.debt_change                         .
## operating.margin.before.dep._change                  .
## return.on.equity_change                              .
## total.debt..total.assets_change                      .
## book..market_change                                  .
## interest..average.LTD_change                         .
## interest..average.total.debt_change                  .
## cash.balance..total.liabilities_change               .
## free.cash.flow..operating.cash.flow_change           .
## total.liabilities..total.tangible.assets_change      .
## total.debt..capital_change                           .
## total.debt..equity_change                            .
## asset.turnover_change                                .
## receivables.turnover_change                          .
## payables.turnover_change                             .
```

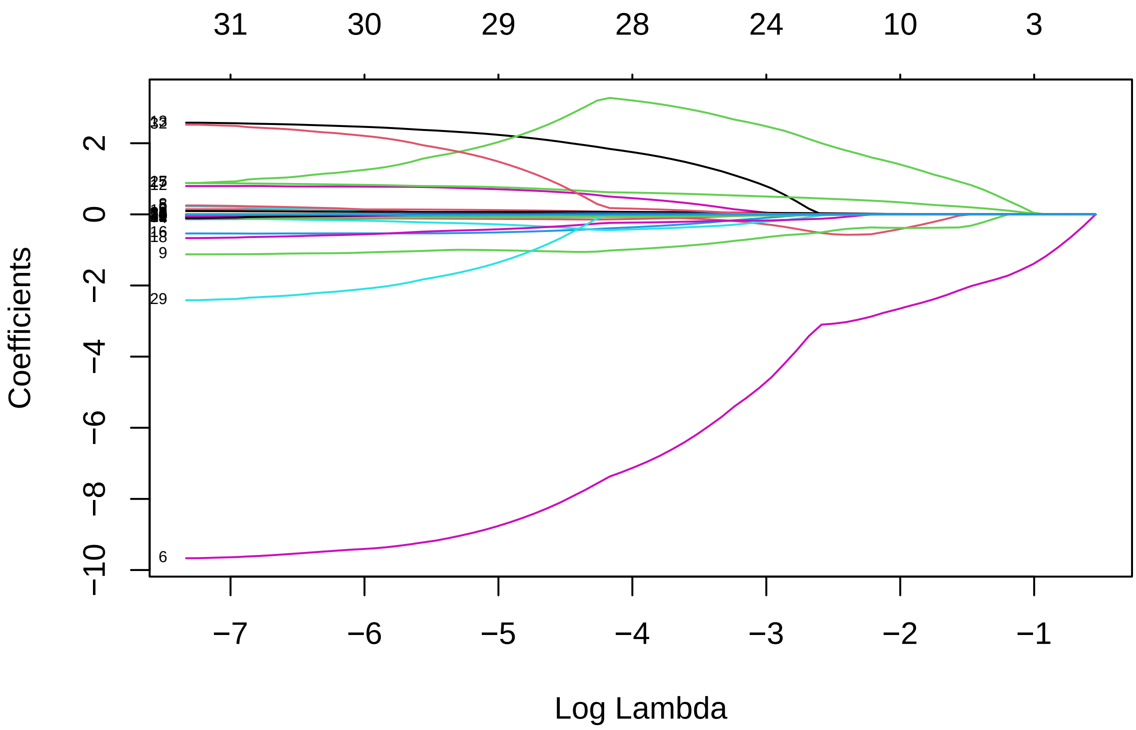
```
## sales.invested.capital_change                               .
## sales.stockholders.equity_change                            .
## price..book_change                                          .
## shillers.cyclically.adjusted.P.E.ratio_change               .
## enterprise.value.multiple_change                            .
## price..operating.earnings..Basic..Excl..EI._change          .
## price..operating.earnings..Diluted..Excl..EI._change        .
## P.E..Diluted..Excl..EI._change                              .
## P.E..Diluted..Incl..EI._change                              .
## price..sales_change                                         .
## price..cash.flow_change                                     .
## gross.profit.margin_change                                  .
## after.tax.return.on.average.common.equity_change            .
## after.tax.return.on.average.stockholders..equity_change     .
## gross.profit..total.assets_change                           .
## common.equity.invested.capital_change                       .
## cash.flow.margin_change                                     .
```

60-40

```
lasso_60 <- glmnet(predictors_60, data_train_60$rating_diff, alpha = 1)
summary(lasso_60)
```

```
##           Length Class     Mode
## a0          74   -none-    numeric
## beta      2516   dgCMatrix S4
## df          74   -none-    numeric
## dim          2   -none-    numeric
## lambda      74   -none-    numeric
## dev.ratio   74   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         4   -none-    call
## nobs         1   -none-    numeric
```

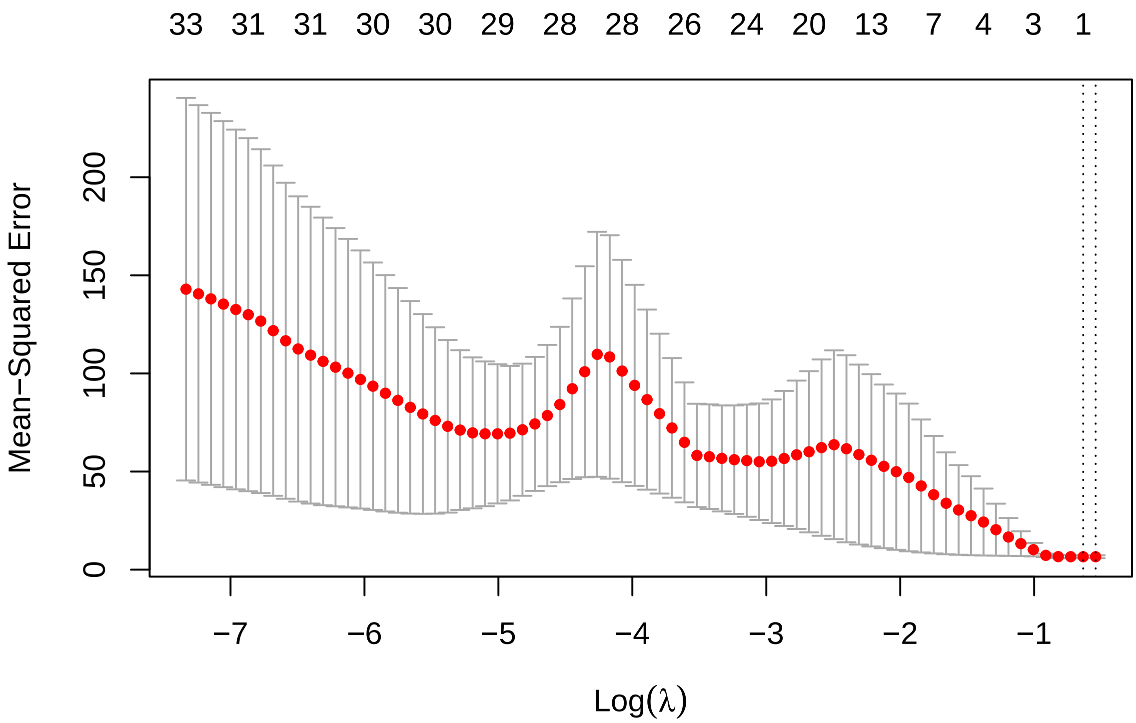
```
plot(lasso_60, xvar = "lambda", label = TRUE)
```




```r
# With a cross-validation
cv.lasso_60 <- cv.glmnet(predictors_60, data_train_60$rating_diff, alpha = 1)
bestlam_la_60 <- cv.lasso_60$lambda.min
bestlam_la_60
```

```
## [1] 0.5304601
```

```r
plot(cv.lasso_60)
```




```
cv_lasso_60 <- glmnet(predictors_60, data_train_60$rating_diff, alpha = 1, lambda = bestlam_la_60)
coef(cv_lasso_60)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                        s0
## (Intercept)                                    -0.3339939
## after.tax.interest.coverage_change              .
## interest.coverage.ratio_change                  .
## cash.flow..total.debt_change                    .
## operating.margin.before.dep._change             .
## return.on.equity_change                         .
## total.debt..total.assets_change                -0.3368301
## book..market_change                             .
## interest..average.LTD_change                    .
## interest..average.total.debt_change             .
## cash.balance..total.liabilities_change          .
## free.cash.flow..operating.cash.flow_change      .
## total.liabilities..total.tangible.assets_change .
## total.debt..capital_change                      .
## total.debt..equity_change                       .
## asset.turnover_change                           .
## receivables.turnover_change                     .
## payables.turnover_change                        .
## sales.invested.capital_change                   .
## sales.stockholders.equity_change                .
```

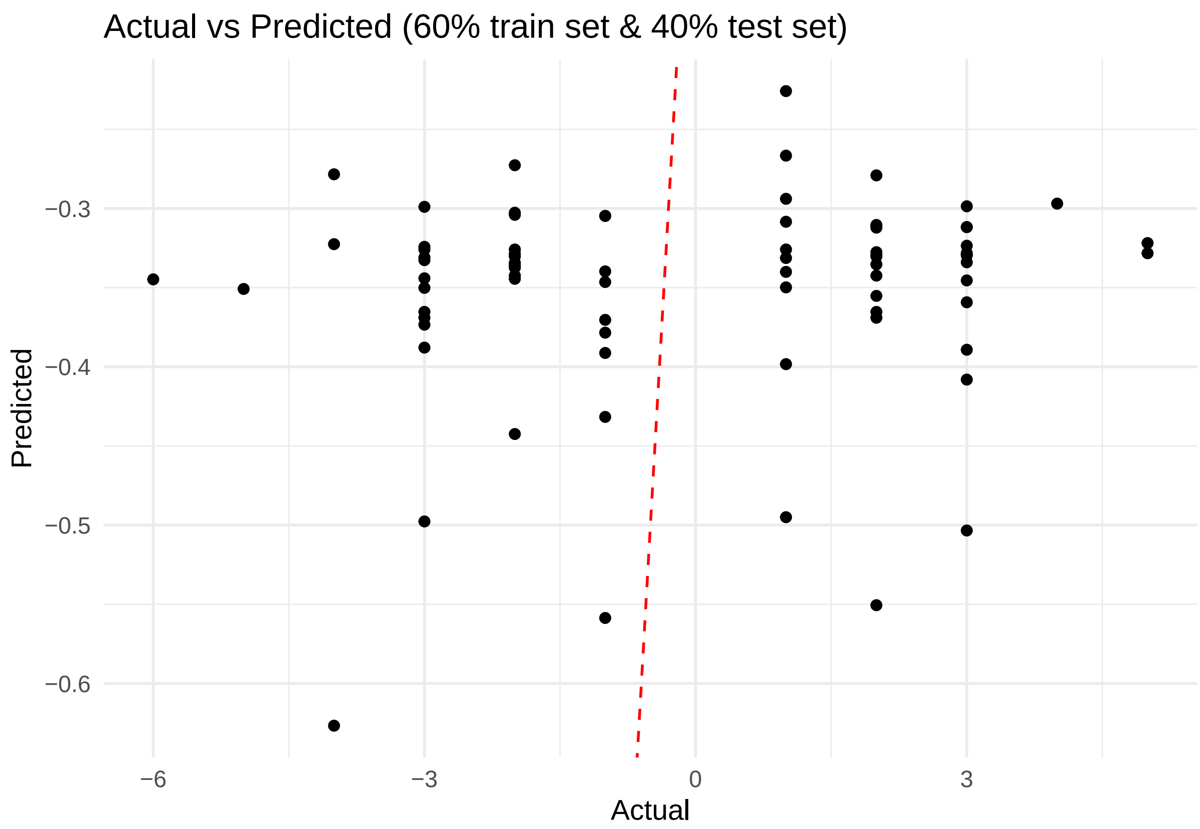
```
## price..book_change                                                .
## shillers.cyclically.adjusted.P.E.ratio_change                     .
## enterprise.value.multiple_change                                  .
## price..operating.earnings..Basic..Excl..EI._change                .
## price..operating.earnings..Diluted..Excl..EI._change              .
## P.E..Diluted..Excl..EI._change                                    .
## P.E..Diluted..Incl..EI._change                                    .
## price..sales_change                                               .
## price..cash.flow_change                                           .
## gross.profit.margin_change                                        .
## after.tax.return.on.average.common.equity_change                  .
## after.tax.return.on.average.stockholders..equity_change           .
## gross.profit..total.assets_change                                 .
## common.equity.invested.capital_change                             .
## cash.flow.margin_change                                           .
```

```r
    # Prediction
la_60_pred <- predict(cv_lasso_60, s = bestlam_la_60, newx = test_60)
print(la_60_pred)
```

```
##               s1
##  [1,] -0.3373622
##  [2,] -0.3592561
##  [3,] -0.3552141
##  [4,] -0.3653191
##  [5,] -0.3360148
##  [6,] -0.3690242
##  [7,] -0.3501617
##  [8,] -0.3282677
##  [9,] -0.3454461
## [10,] -0.2726908
## [11,] -0.3309624
## [12,] -0.3259099
## [13,] -0.4949986
## [14,] -0.3424146
## [15,] -0.4080965
## [16,] -0.3447724
## [17,] -0.2989635
## [18,] -0.2666278
## [19,] -0.3464566
## [20,] -0.3892340
## [21,] -0.5505756
## [22,] -0.3302887
## [23,] -0.3259099
## [24,] -0.3296151
## [25,] -0.3400568
## [26,] -0.2258714
## [27,] -0.3353412
## [28,] -0.3282677
## [29,] -0.3424146
## [30,] -0.3218680
## [31,] -0.3302887
## [32,] -0.3117631
## [33,] -0.3117631
```

```
## [34,] -0.3784554
## [35,] -0.3912550
## [36,] -0.3046896
## [37,] -0.2986267
## [38,] -0.5034194
## [39,] -0.4316746
## [40,] -0.3983284
## [41,] -0.3083948
## [42,] -0.2939111
## [43,] -0.3104157
## [44,] -0.3703715
## [45,] -0.3498249
## [46,] -0.3690242
## [47,] -0.3286046
## [48,] -0.4424531
## [49,] -0.3040160
## [50,] -0.3242258
## [51,] -0.3653191
## [52,] -0.3440988
## [53,] -0.3878867
## [54,] -0.3225416
## [55,] -0.4976933
## [56,] -0.3508354
## [57,] -0.3120999
## [58,] -0.3117631
## [59,] -0.6266992
## [60,] -0.3339939
## [61,] -0.3296151
## [62,] -0.3292782
## [63,] -0.3326465
## [64,] -0.3312992
## [65,] -0.2969425
## [66,] -0.3259099
## [67,] -0.2784169
## [68,] -0.3046896
## [69,] -0.3343307
## [70,] -0.5586595
## [71,] -0.3397200
## [72,] -0.3026687
## [73,] -0.3275941
## [74,] -0.3235521
## [75,] -0.3734030
## [76,] -0.2790905
## [77,] -0.3444356
```

```
    # Plot
plot_la_60 <- data.frame(Actual = data_test_40$rating_diff, Predicted = as.vector(la_60_pred))
ggplot(plot_la_60, aes(x = Actual, y = Predicted)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Actual vs Predicted (60% train set & 40% test set)",
       x = "Actual",
       y = "Predicted") +
  theme_minimal()
```




```r
    # Evaluation
pe_la_60<- plot_la_60$Predicted - plot_la_60$Actual
rmse_la_60 <- sqrt(mean(pe_la_60^2))
cat("Lasso (60-40) RMSE:", rmse_la_60, "\n")
```

```
## Lasso (60-40) RMSE: 2.608052
```

```r
table_la_60 <- data.frame(Model = "Lasso (60-40)",
                          RMSE = rmse_la_60)
print(table_la_60)
```

```
##           Model     RMSE
## 1 Lasso (60-40) 2.608052
```

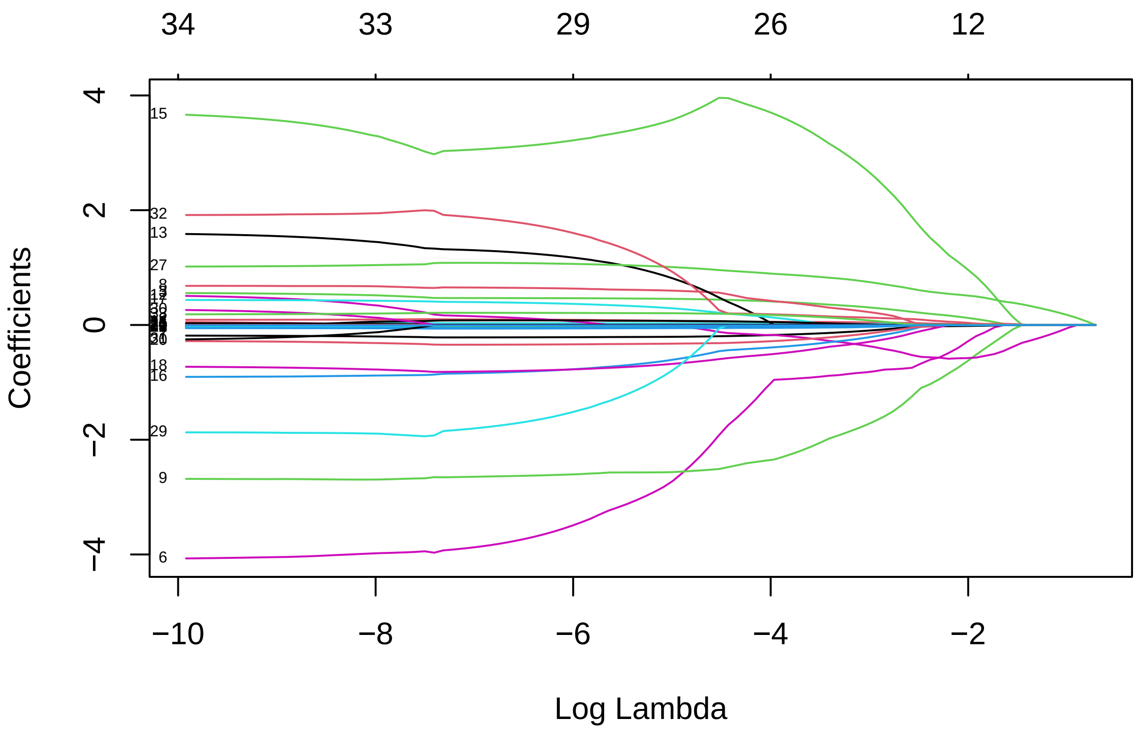
70-30

```r
lasso_70 <- glmnet(predictors_70, data_train_70$rating_diff, alpha = 1)
summary(lasso_70)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      3400   dgCMatrix S4
```

```
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         4   -none-    call
## nobs         1   -none-    numeric
```

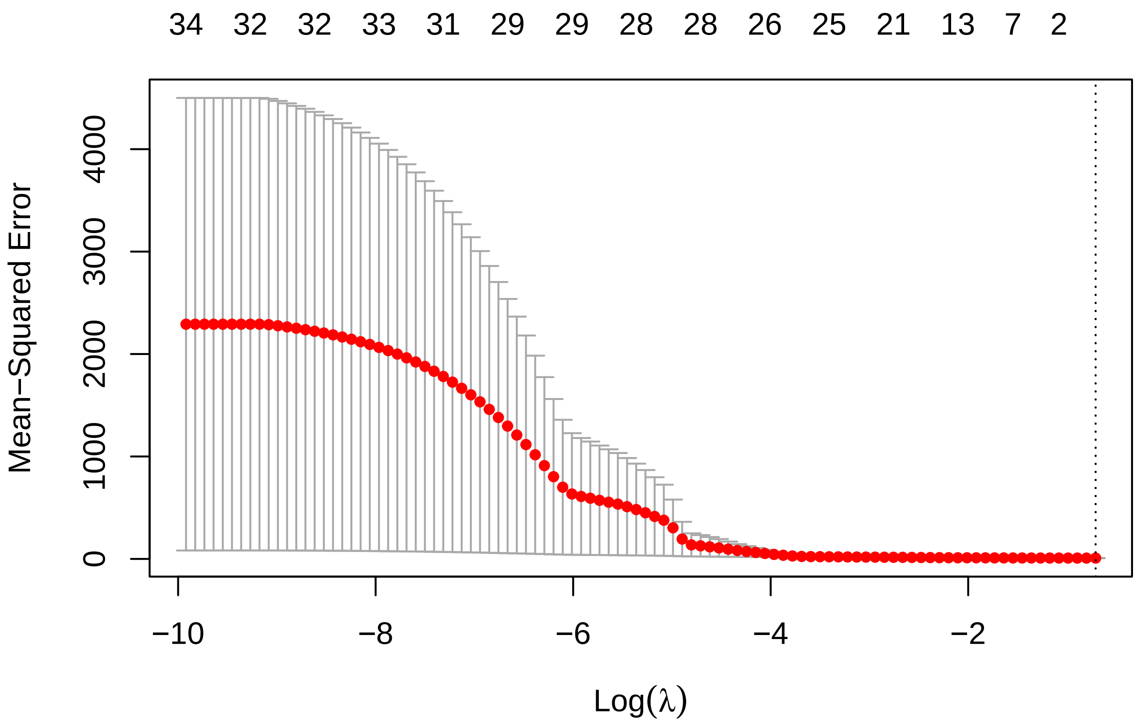
```
plot(lasso_70, xvar = "lambda", label = TRUE)
```




```
    # With a cross-validation
cv.lasso_70 <- cv.glmnet(predictors_70, data_train_70$rating_diff, alpha = 1)
bestlam_la_70 <- cv.lasso_70$lambda.min
bestlam_la_70
```

```
## [1] 0.4918733
```

```
plot(cv.lasso_70)
```




```r
cv_lasso_70 <- glmnet(predictors_70, data_train_70$rating_diff, alpha = 1, lambda = bestlam_la_70)
coef(cv_lasso_70)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                         s0
## (Intercept)                                    -0.08148148
## after.tax.interest.coverage_change              0.00000000
## interest.coverage.ratio_change                  .
## cash.flow..total.debt_change                    .
## operating.margin.before.dep._change             .
## return.on.equity_change                         .
## total.debt..total.assets_change                 .
## book..market_change                             .
## interest..average.LTD_change                    .
## interest..average.total.debt_change             .
## cash.balance..total.liabilities_change          .
## free.cash.flow..operating.cash.flow_change      .
## total.liabilities..total.tangible.assets_change .
## total.debt..capital_change                      .
## total.debt..equity_change                       .
## asset.turnover_change                           .
## receivables.turnover_change                     .
## payables.turnover_change                        .
## sales.invested.capital_change                   .
## sales.stockholders.equity_change                .
```

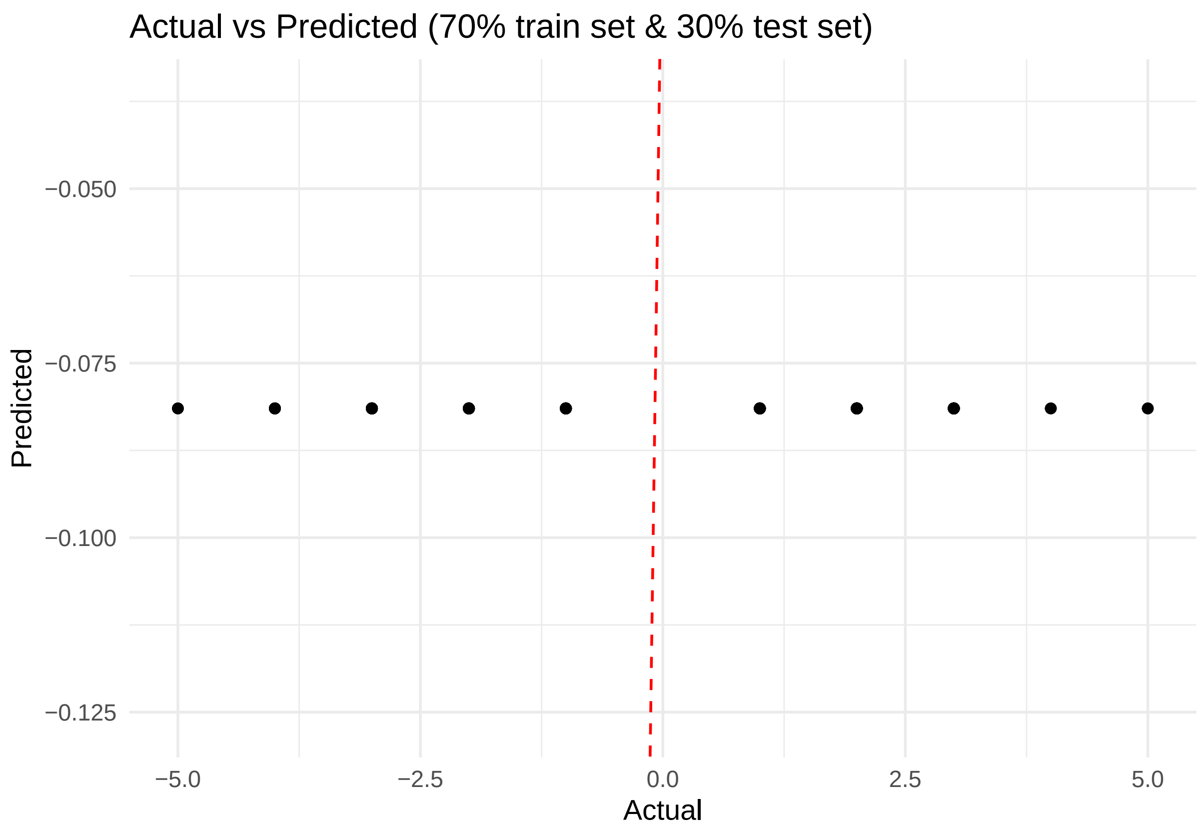
```
## price..book_change                                           .
## shillers.cyclically.adjusted.P.E.ratio_change                .
## enterprise.value.multiple_change                             .
## price..operating.earnings..Basic..Excl..EI._change           .
## price..operating.earnings..Diluted..Excl..EI._change         .
## P.E..Diluted..Excl..EI._change                               .
## P.E..Diluted..Incl..EI._change                               .
## price..sales_change                                          .
## price..cash.flow_change                                      .
## gross.profit.margin_change                                   .
## after.tax.return.on.average.common.equity_change             .
## after.tax.return.on.average.stockholders..equity_change      .
## gross.profit..total.assets_change                            .
## common.equity.invested.capital_change                        .
## cash.flow.margin_change                                      .
```

```
    # Prediction
la_70_pred <- predict(cv_lasso_70, s = bestlam_la_70, newx = test_70)
print(la_70_pred)
```

```
##                s1
##  [1,] -0.08148148
##  [2,] -0.08148148
##  [3,] -0.08148148
##  [4,] -0.08148148
##  [5,] -0.08148148
##  [6,] -0.08148148
##  [7,] -0.08148148
##  [8,] -0.08148148
##  [9,] -0.08148148
## [10,] -0.08148148
## [11,] -0.08148148
## [12,] -0.08148148
## [13,] -0.08148148
## [14,] -0.08148148
## [15,] -0.08148148
## [16,] -0.08148148
## [17,] -0.08148148
## [18,] -0.08148148
## [19,] -0.08148148
## [20,] -0.08148148
## [21,] -0.08148148
## [22,] -0.08148148
## [23,] -0.08148148
## [24,] -0.08148148
## [25,] -0.08148148
## [26,] -0.08148148
## [27,] -0.08148148
## [28,] -0.08148148
## [29,] -0.08148148
## [30,] -0.08148148
## [31,] -0.08148148
## [32,] -0.08148148
## [33,] -0.08148148
```

```
## [34,] -0.08148148
## [35,] -0.08148148
## [36,] -0.08148148
## [37,] -0.08148148
## [38,] -0.08148148
## [39,] -0.08148148
## [40,] -0.08148148
## [41,] -0.08148148
## [42,] -0.08148148
## [43,] -0.08148148
## [44,] -0.08148148
## [45,] -0.08148148
## [46,] -0.08148148
## [47,] -0.08148148
## [48,] -0.08148148
## [49,] -0.08148148
## [50,] -0.08148148
## [51,] -0.08148148
## [52,] -0.08148148
## [53,] -0.08148148
## [54,] -0.08148148
## [55,] -0.08148148
## [56,] -0.08148148
## [57,] -0.08148148
## [58,] -0.08148148
```

```
    # Plot
plot_la_70 <- data.frame(Actual = data_test_30$rating_diff, Predicted = as.vector(la_70_pred))
ggplot(plot_la_70, aes(x = Actual, y = Predicted)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Actual vs Predicted (70% train set & 30% test set)",
       x = "Actual",
       y = "Predicted") +
  theme_minimal()
```




```r
    # Evaluation
pe_la_70<- plot_la_70$Predicted - plot_la_70$Actual
rmse_la_70 <- sqrt(mean(pe_la_70^2))
cat("Lasso (70-30) RMSE:", rmse_la_70, "\n")
```

```
## Lasso (70-30) RMSE: 2.51623
```

```r
table_la_70 <- data.frame(Model = "Lasso (70-30)",
                          RMSE = rmse_la_70)
print(table_la_70)
```

```
##           Model    RMSE
## 1 Lasso (70-30) 2.51623
```

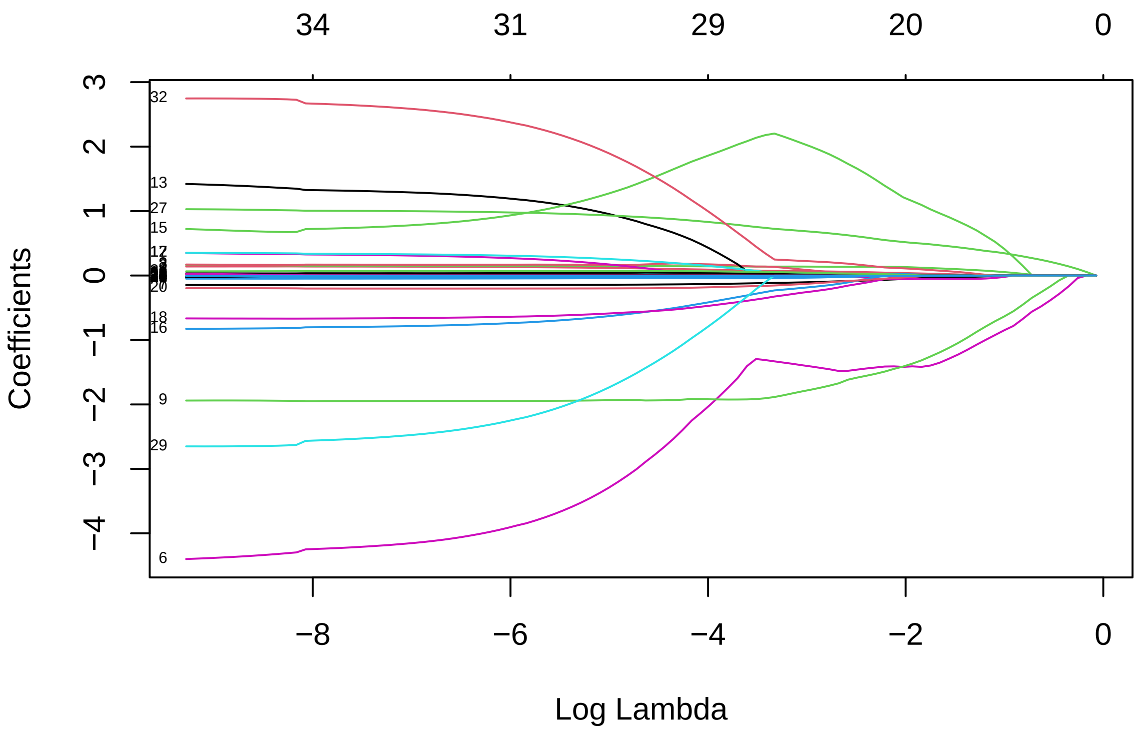
Elsatic Net

Full sample

```r
enet_model <- glmnet(predictors, data$rating_diff, alpha = 0.5)
summary(enet_model)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      3400   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         4   -none-    call
## nobs         1   -none-    numeric
```

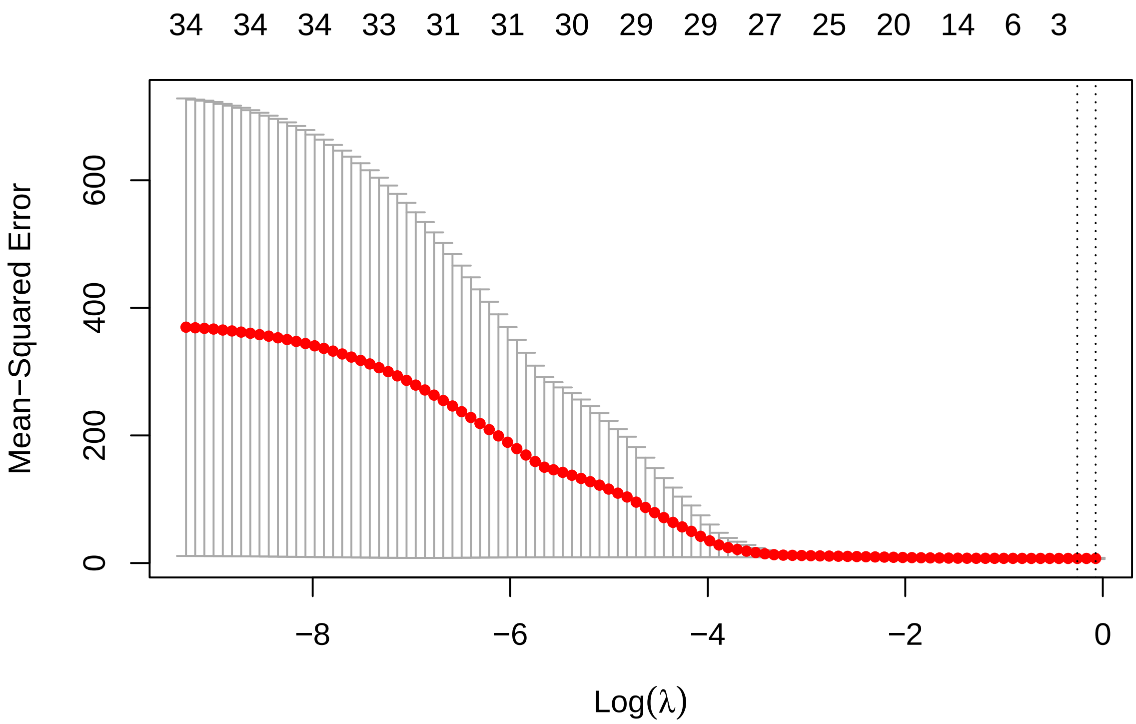
```r
plot(enet_model, xvar = "lambda", label = TRUE)
```




```r
    # With a cross-validation
cv.enet <- cv.glmnet(predictors, data$rating_diff, alpha = 0.5)
bestlam_en <- cv.enet$lambda.min
bestlam_en
```

```
## [1] 0.7722315
```

```r
plot(cv.enet)
```




```r
cv_enet <- glmnet(predictors, data$rating_diff, alpha = 0.5, lambda = bestlam_en)
coef(cv_enet)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                     s0
## (Intercept)                                -0.18983978
## after.tax.interest.coverage_change           .
## interest.coverage.ratio_change               .
## cash.flow..total.debt_change                 .
## operating.margin.before.dep._change          .
## return.on.equity_change                      .
## total.debt..total.assets_change            -0.03825809
## book..market_change                          .
## interest..average.LTD_change                 .
## interest..average.total.debt_change          .
## cash.balance..total.liabilities_change       .
## free.cash.flow..operating.cash.flow_change   .
## total.liabilities..total.tangible.assets_change .
## total.debt..capital_change                   .
## total.debt..equity_change                    .
## asset.turnover_change                        .
## receivables.turnover_change                  .
## payables.turnover_change                     .
```

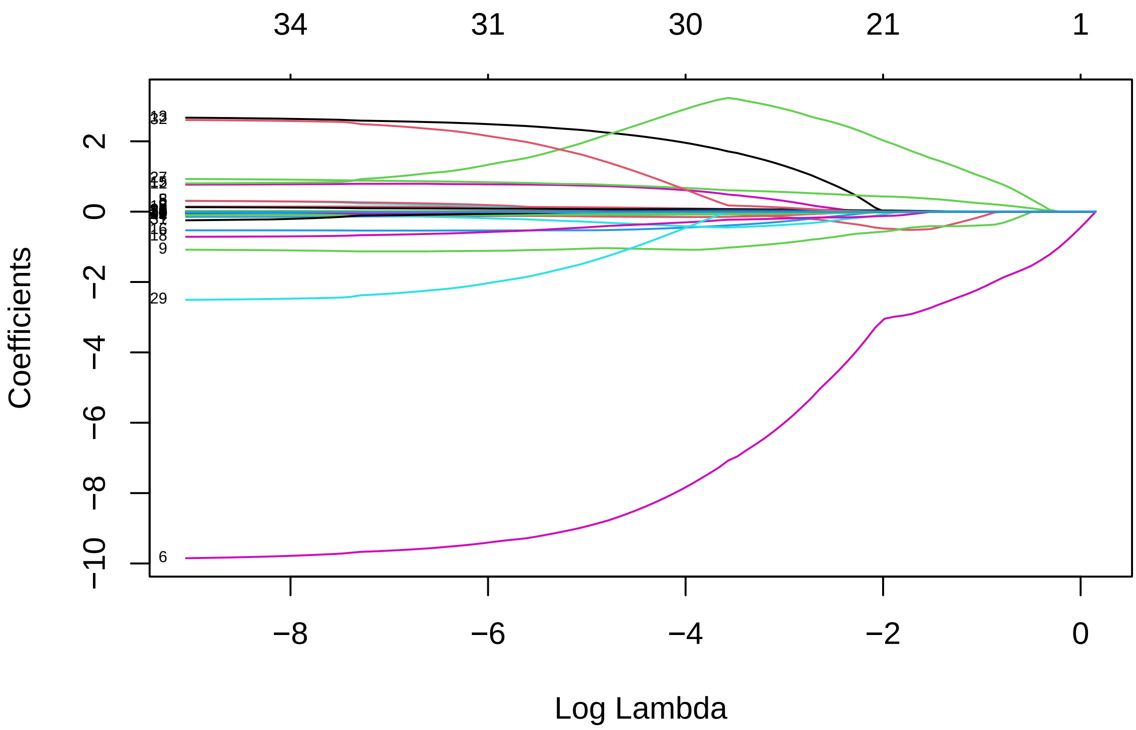
```
## sales.invested.capital_change                             .
## sales.stockholders.equity_change                          .
## price..book_change                                        .
## shillers.cyclically.adjusted.P.E.ratio_change             .
## enterprise.value.multiple_change                          .
## price..operating.earnings..Basic..Excl..EI._change        .
## price..operating.earnings..Diluted..Excl..EI._change      .
## P.E..Diluted..Excl..EI._change                            .
## P.E..Diluted..Incl..EI._change                            .
## price..sales_change                                       0.10033944
## price..cash.flow_change                                   .
## gross.profit.margin_change                                .
## after.tax.return.on.average.common.equity_change          .
## after.tax.return.on.average.stockholders..equity_change   .
## gross.profit..total.assets_change                         .
## common.equity.invested.capital_change                     .
## cash.flow.margin_change                                   .
```

60-40

```r
enet_60 <- glmnet(predictors_60, data_train_60$rating_diff, alpha = 0.5)
summary(enet_60)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      3400   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         4   -none-    call
## nobs         1   -none-    numeric
```

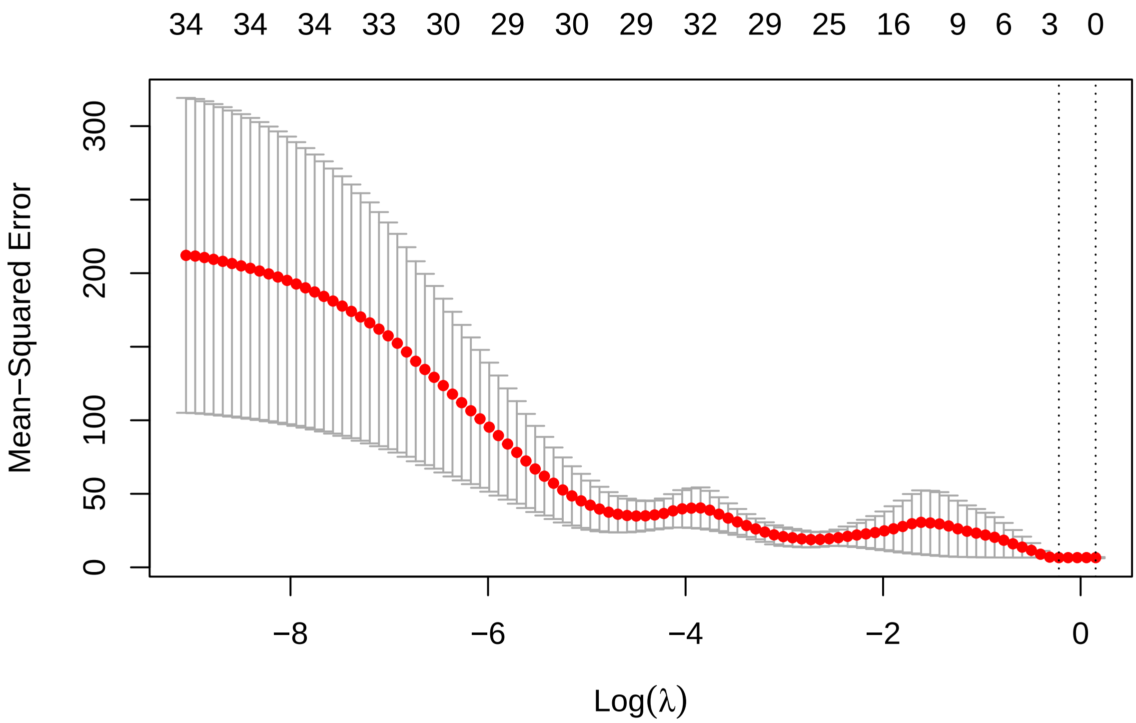
```r
plot(enet_60, xvar = "lambda", label = TRUE)
```




```
# With a cross-validation, set lambda
cv.enet_60 <- cv.glmnet(predictors_60, data_train_60$rating_diff, alpha = 0.5)
bestlam_en_60 <- cv.enet_60$lambda.min
bestlam_en_60
```

```
## [1] 0.8025472
```

```
plot(cv.enet_60)
```




```r
cv_enet_60 <- glmnet(predictors_60, data_train_60$rating_diff, alpha = 0.5, lambda = bestlam_en_60)
coef(cv_enet_60)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                              s0
## (Intercept)                                          -0.3120974
## after.tax.interest.coverage_change                    .
## interest.coverage.ratio_change                        .
## cash.flow..total.debt_change                          .
## operating.margin.before.dep._change                   .
## return.on.equity_change                               .
## total.debt..total.assets_change                      -1.0176104
## book..market_change                                   .
## interest..average.LTD_change                          .
## interest..average.total.debt_change                   .
## cash.balance..total.liabilities_change                .
## free.cash.flow..operating.cash.flow_change            .
## total.liabilities..total.tangible.assets_change       .
## total.debt..capital_change                            .
## total.debt..equity_change                             .
## asset.turnover_change                                 .
## receivables.turnover_change                           .
## payables.turnover_change                              .
## sales.invested.capital_change                         .
## sales.stockholders.equity_change                      .
```

```
## price..book_change                                        .
## shillers.cyclically.adjusted.P.E.ratio_change             .
## enterprise.value.multiple_change                          .
## price..operating.earnings..Basic..Excl..EI._change        .
## price..operating.earnings..Diluted..Excl..EI._change      .
## P.E..Diluted..Excl..EI._change                            .
## P.E..Diluted..Incl..EI._change                            .
## price..sales_change                                       .
## price..cash.flow_change                                   .
## gross.profit.margin_change                                .
## after.tax.return.on.average.common.equity_change          .
## after.tax.return.on.average.stockholders..equity_change   .
## gross.profit..total.assets_change                         .
## common.equity.invested.capital_change                     .
## cash.flow.margin_change                                   .
```

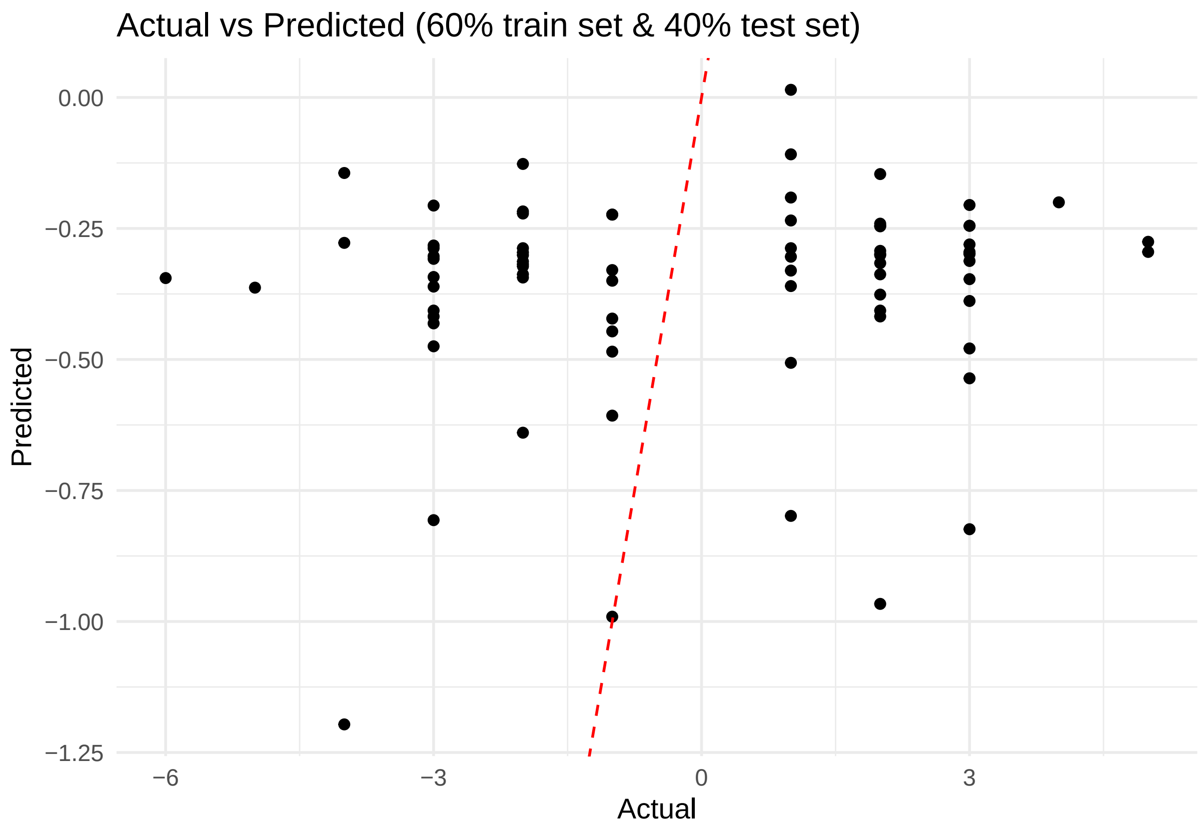
```r
    # Prediction
en_60_pred <- predict(cv_enet_60, s = bestlam_en_60, newx = test_60)
print(en_60_pred)
```

```
##                s1
##  [1,] -0.32227348
##  [2,] -0.38841816
##  [3,] -0.37620683
##  [4,] -0.40673514
##  [5,] -0.31820304
##  [6,] -0.41792886
##  [7,] -0.36094268
##  [8,] -0.29479800
##  [9,] -0.34669613
## [10,] -0.12689228
## [11,] -0.30293888
## [12,] -0.28767473
## [13,] -0.79851515
## [14,] -0.33753764
## [15,] -0.53597166
## [16,] -0.34466091
## [17,] -0.20626589
## [18,] -0.10857529
## [19,] -0.34974896
## [20,] -0.47898548
## [21,] -0.96642087
## [22,] -0.30090366
## [23,] -0.28767473
## [24,] -0.29886844
## [25,] -0.33041436
## [26,]  0.01455556
## [27,] -0.31616782
## [28,] -0.29479800
## [29,] -0.33753764
## [30,] -0.27546340
## [31,] -0.30090366
## [32,] -0.24493509
## [33,] -0.24493509
```

```
## [34,] -0.44642195
## [35,] -0.48509114
## [36,] -0.22356527
## [37,] -0.20524828
## [38,] -0.82395541
## [39,] -0.60720439
## [40,] -0.50646096
## [41,] -0.23475898
## [42,] -0.19100174
## [43,] -0.24086465
## [44,] -0.42199930
## [45,] -0.35992506
## [46,] -0.41792886
## [47,] -0.29581561
## [48,] -0.63976793
## [49,] -0.22153005
## [50,] -0.28258667
## [51,] -0.40673514
## [52,] -0.34262569
## [53,] -0.47491504
## [54,] -0.27749862
## [55,] -0.80665603
## [56,] -0.36297790
## [57,] -0.24595270
## [58,] -0.24493509
## [59,] -1.19640082
## [60,] -0.31209738
## [61,] -0.29886844
## [62,] -0.29785083
## [63,] -0.30802693
## [64,] -0.30395649
## [65,] -0.20016023
## [66,] -0.28767473
## [67,] -0.14419166
## [68,] -0.22356527
## [69,] -0.31311499
## [70,] -0.99084352
## [71,] -0.32939675
## [72,] -0.21745961
## [73,] -0.29276278
## [74,] -0.28055145
## [75,] -0.43115779
## [76,] -0.14622688
## [77,] -0.34364330
```

```
    # Plot and Evaluation
plot_en_60 <- data.frame(Actual = data_test_40$rating_diff,
                         Predicted = as.vector(en_60_pred))
ggplot(plot_en_60, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted (60% train set & 40% test set)",
       x = "Actual",
       y = "Predicted") +
```

```
      theme_minimal()
```




```
    # Evaluation
pe_en_60<- plot_en_60$Predicted - plot_en_60$Actual
rmse_en_60 <- sqrt(mean(pe_en_60^2))
cat("ENet (60-40) RMSE:", rmse_en_60, "\n")
```

```
## ENet (60-40) RMSE: 2.601397
```

```
table_en_60 <- data.frame(Model = "ENet (60-40)",
                          RMSE = rmse_en_60)
print(table_en_60)
```

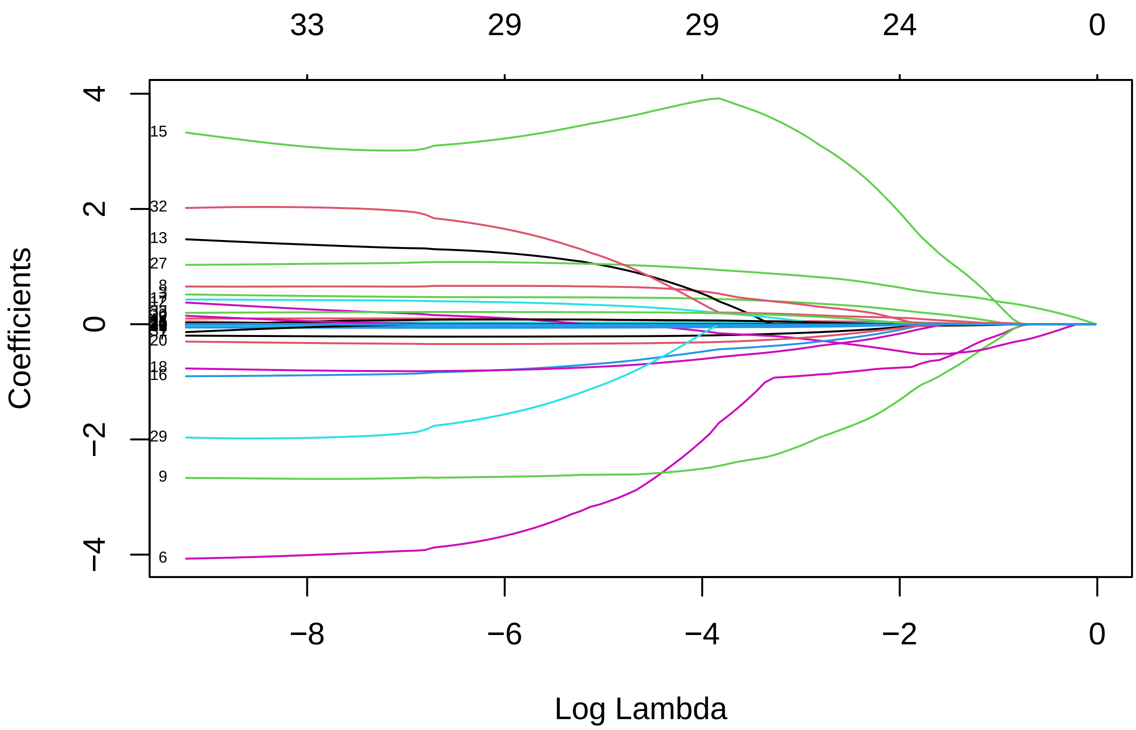
```
##          Model     RMSE
## 1 ENet (60-40) 2.601397
```

70-30

```
enet_70 <- glmnet(predictors_70, data_train_70$rating_diff, alpha = 0.5)
summary(enet_70)
```

```
##            Length Class     Mode
## a0          100   -none-    numeric
## beta       3400   dgCMatrix S4
## df          100   -none-    numeric
## dim           2   -none-    numeric
## lambda      100   -none-    numeric
## dev.ratio   100   -none-    numeric
## nulldev       1   -none-    numeric
## npasses       1   -none-    numeric
## jerr          1   -none-    numeric
## offset        1   -none-    logical
## call          4   -none-    call
## nobs          1   -none-    numeric
```

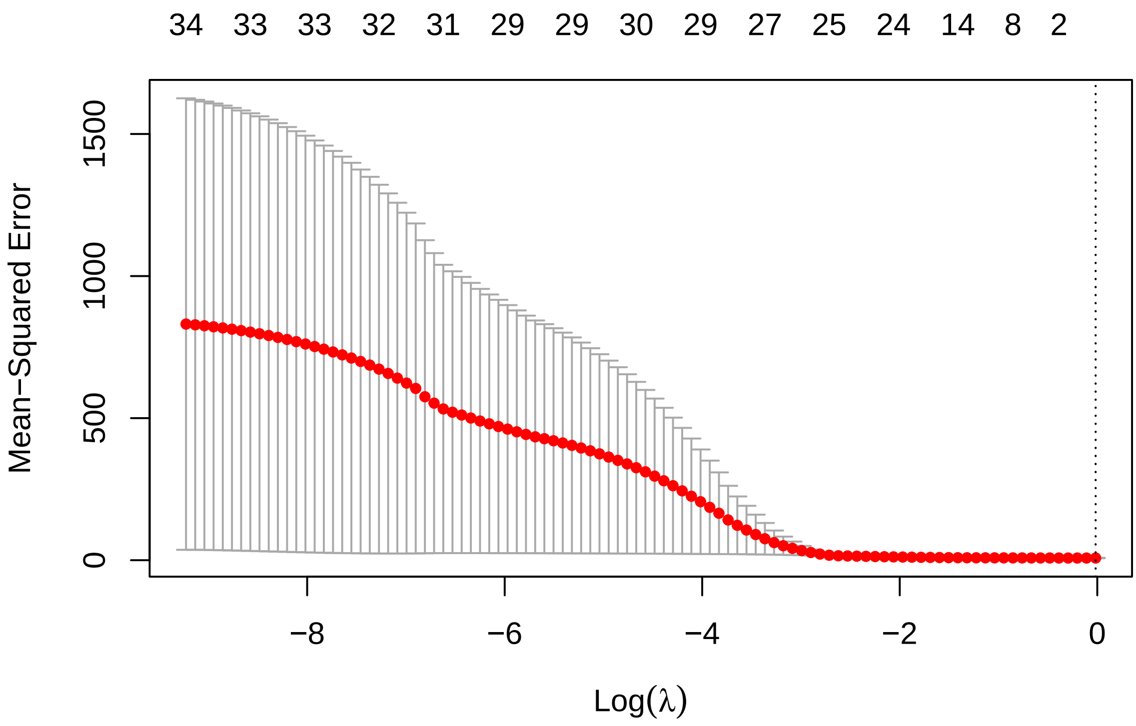
```
plot(enet_70, xvar = "lambda", label = TRUE)
```




```
    # With a cross-validation, set lambda
cv.enet_70 <- cv.glmnet(predictors_70, data_train_70$rating_diff, alpha = 0.5)
bestlam_en_70 <- cv.enet_70$lambda.min
bestlam_en_70
```

```
## [1] 0.9837466
```

```r
plot(cv.enet_70)
```




```r
cv_enet_70 <- glmnet(predictors_70, data_train_70$rating_diff, alpha = 0.5, lambda = bestlam_en_70)
coef(cv_enet_70)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                                                       s0
## (Intercept)                                  -0.08148148
## after.tax.interest.coverage_change            0.00000000
## interest.coverage.ratio_change                .
## cash.flow..total.debt_change                  .
## operating.margin.before.dep._change           .
## return.on.equity_change                       .
## total.debt..total.assets_change               .
## book..market_change                           .
## interest..average.LTD_change                  .
## interest..average.total.debt_change           .
## cash.balance..total.liabilities_change        .
## free.cash.flow..operating.cash.flow_change    .
## total.liabilities..total.tangible.assets_change .
## total.debt..capital_change                    .
## total.debt..equity_change                     .
## asset.turnover_change                         .
## receivables.turnover_change                   .
## payables.turnover_change                      .
```

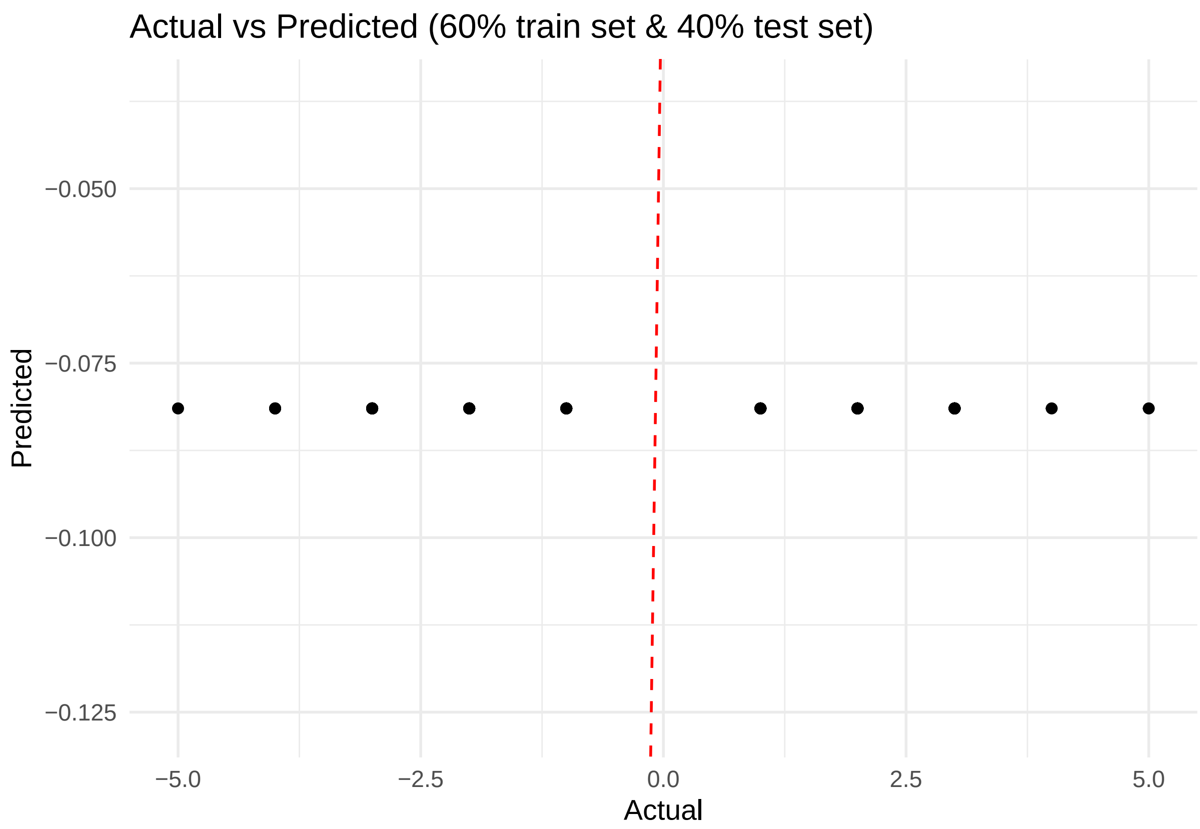
```
## sales.invested.capital_change                              .
## sales.stockholders.equity_change                           .
## price..book_change                                         .
## shillers.cyclically.adjusted.P.E.ratio_change              .
## enterprise.value.multiple_change                           .
## price..operating.earnings..Basic..Excl..EI._change         .
## price..operating.earnings..Diluted..Excl..EI._change       .
## P.E..Diluted..Excl..EI._change                             .
## P.E..Diluted..Incl..EI._change                             .
## price..sales_change                                        .
## price..cash.flow_change                                    .
## gross.profit.margin_change                                 .
## after.tax.return.on.average.common.equity_change           .
## after.tax.return.on.average.stockholders..equity_change    .
## gross.profit..total.assets_change                          .
## common.equity.invested.capital_change                      .
## cash.flow.margin_change                                    .
```

```
    # Prediction
en_70_pred <- predict(cv_enet_70, s = bestlam_en_70, newx = test_70)
print(en_70_pred)
```

```
##                s1
##  [1,] -0.08148148
##  [2,] -0.08148148
##  [3,] -0.08148148
##  [4,] -0.08148148
##  [5,] -0.08148148
##  [6,] -0.08148148
##  [7,] -0.08148148
##  [8,] -0.08148148
##  [9,] -0.08148148
## [10,] -0.08148148
## [11,] -0.08148148
## [12,] -0.08148148
## [13,] -0.08148148
## [14,] -0.08148148
## [15,] -0.08148148
## [16,] -0.08148148
## [17,] -0.08148148
## [18,] -0.08148148
## [19,] -0.08148148
## [20,] -0.08148148
## [21,] -0.08148148
## [22,] -0.08148148
## [23,] -0.08148148
## [24,] -0.08148148
## [25,] -0.08148148
## [26,] -0.08148148
## [27,] -0.08148148
## [28,] -0.08148148
## [29,] -0.08148148
## [30,] -0.08148148
## [31,] -0.08148148
```

```
## [32,] -0.08148148
## [33,] -0.08148148
## [34,] -0.08148148
## [35,] -0.08148148
## [36,] -0.08148148
## [37,] -0.08148148
## [38,] -0.08148148
## [39,] -0.08148148
## [40,] -0.08148148
## [41,] -0.08148148
## [42,] -0.08148148
## [43,] -0.08148148
## [44,] -0.08148148
## [45,] -0.08148148
## [46,] -0.08148148
## [47,] -0.08148148
## [48,] -0.08148148
## [49,] -0.08148148
## [50,] -0.08148148
## [51,] -0.08148148
## [52,] -0.08148148
## [53,] -0.08148148
## [54,] -0.08148148
## [55,] -0.08148148
## [56,] -0.08148148
## [57,] -0.08148148
## [58,] -0.08148148
```

```
    # Plot
plot_en_70 <- data.frame(Actual = data_test_30$rating_diff,
                         Predicted = as.vector(en_70_pred))
ggplot(plot_en_70, aes(x = Actual, y = Predicted)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Actual vs Predicted (60% train set & 40% test set)",
       x = "Actual",
       y = "Predicted") +
  theme_minimal()
```




```r
    # Evaluation
pe_en_70<- plot_en_70$Predicted - plot_en_70$Actual
rmse_en_70 <- sqrt(mean(pe_en_70^2))
cat("ENet (70-30) RMSE:", rmse_en_70, "\n")
```

```
## ENet (70-30) RMSE: 2.51623
```

```r
table_en_70 <- data.frame(Model = "ENet (70-30)",
                          RMSE = rmse_en_70)
print(table_en_70)
```

```
##          Model    RMSE
## 1 ENet (70-30) 2.51623
```

Summary

```r
result_table <- rbind.data.frame(table_rr_60, table_rr_70,
                                 table_la_60, table_la_70,
                                 table_en_60, table_en_70)
result_table
```

```
##                       Model     RMSE
## 1 Ridge Regression (60-40) 2.612516
## 2 Ridge Regression (70-30) 2.513331
## 3            Lasso (60-40) 2.608052
## 4            Lasso (70-30) 2.516230
## 5             ENet (60-40) 2.601397
## 6             ENet (70-30) 2.516230
```

```r
rm(list = ls())
library(MASS)
library(dplyr)
library(rpart)
library(rpart.plot)
library(randomForest)
```

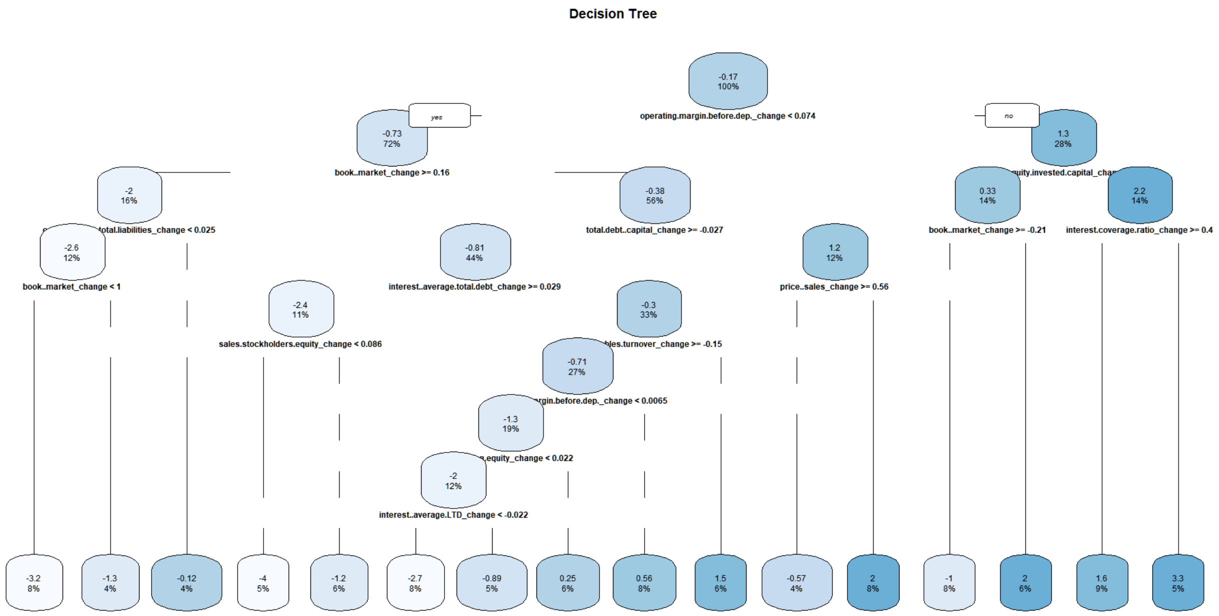
```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(ggplot2)

# full sample
data <- read.csv('C:/Users/user/Desktop/      /Project 1/New Data/SP500_change_V7Final.csv')
# Train set and test set
data_train_60 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/train_df60.csv")
data_train_70 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/train_df70.csv")
data_test_40 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/test_df40.csv")
data_test_30 <- read.csv("C:/Users/user/Desktop/      /Project 1/New Data/test_df30.csv")

data <- data[, -1]
data_train_60 <- data_train_60[, -1]
data_train_70 <- data_train_70[, -1]
data_test_40 <- data_test_40[, -1]
data_test_30 <- data_test_30[, -1]
```

Decision Tree

Full sample

```r
tree_model <- rpart(rating_diff ~ ., data = data, xval = 5)
# rpart.plot(tree_model, main = "Decision Tree",
#            cex = 0.5, tweak = 1.0, compress = TRUE)
```



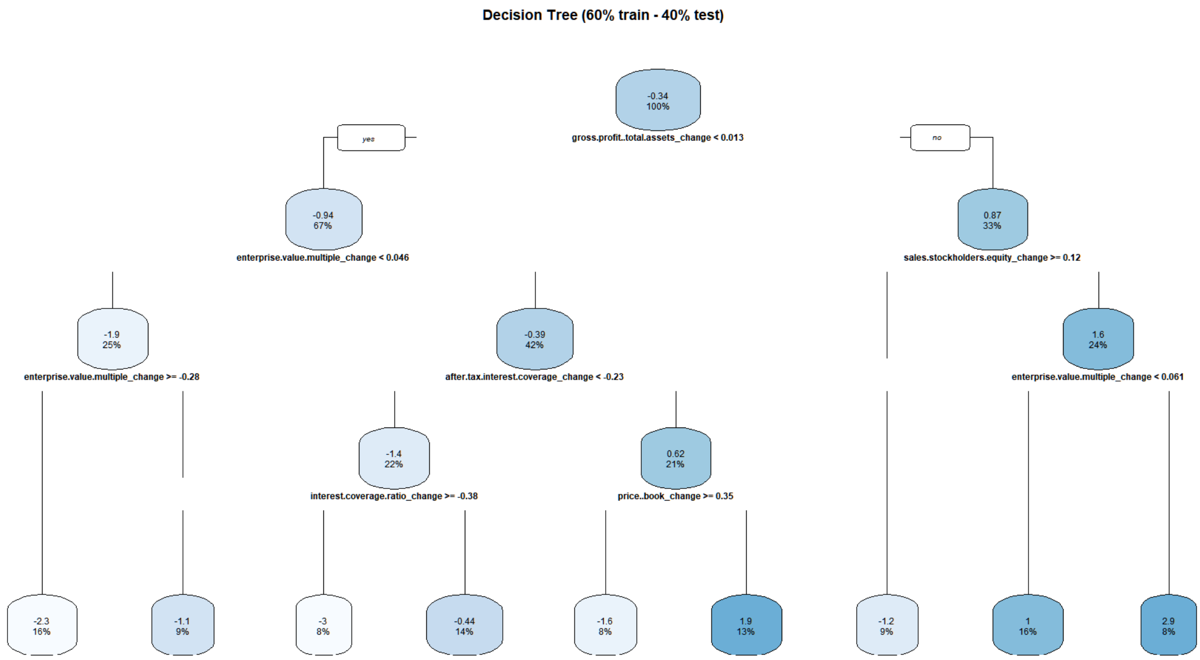

```r
tree_model$variable.importance
```

```
##                     operating.margin.before.dep._change
##                                              233.355287
##                                     book..market_change
##                                              161.398717
##                                      price..book_change
##                                              143.943108
##                              total.debt..capital_change
##                                              126.013624
##                           interest.coverage.ratio_change
##                                              124.956701
##                                     price..sales_change
##                                              105.381487
##                                 cash.flow.margin_change
##                                              100.488701
##                               gross.profit.margin_change
##                                               94.528911
##                        sales.stockholders.equity_change
##                                               93.739478
##                     interest..average.total.debt_change
##                                               93.634072
##                         total.debt..total.assets_change
##                                               78.971166
##                              receivables.turnover_change
##                                               67.309755
##               free.cash.flow..operating.cash.flow_change
```

```
##                                                     64.427362
##                      after.tax.interest.coverage_change
##                                                     63.216371
##                            interest..average.LTD_change
##                                                     63.057221
##                   common.equity.invested.capital_change
##                                                     61.129085
##                                 return.on.equity_change
##                                                     58.066908
##                               total.debt..equity_change
##                                                     53.899853
## after.tax.return.on.average.stockholders..equity_change
##                                                     49.393497
##        after.tax.return.on.average.common.equity_change
##                                                     45.892339
##                  cash.balance..total.liabilities_change
##                                                     45.119419
##                       gross.profit..total.assets_change
##                                                     39.091237
##                           sales.invested.capital_change
##                                                     38.860132
##                       enterprise.value.multiple_change
##                                                     36.103873
##      price..operating.earnings..Basic..Excl..EI._change
##                                                     35.000000
##    price..operating.earnings..Diluted..Excl..EI._change
##                                                     35.000000
##                            cash.flow..total.debt_change
##                                                     30.967624
##                                 price..cash.flow_change
##                                                     27.598935
##         total.liabilities..total.tangible.assets_change
##                                                     25.049053
##           shillers.cyclically.adjusted.P.E.ratio_change
##                                                     18.399290
##                                   asset.turnover_change
##                                                     16.831843
##                                payables.turnover_change
##                                                      7.033754
```

60-40

```
tree_60 <- rpart(data_train_60$rating_diff ~ .,
                 data = data_train_60, xval = 5)
# rpart.plot(tree_60, main = "Decision Tree (60% train - 40% test)",
#            cex = 0.5, tweak = 1.0, compress = TRUE)
```

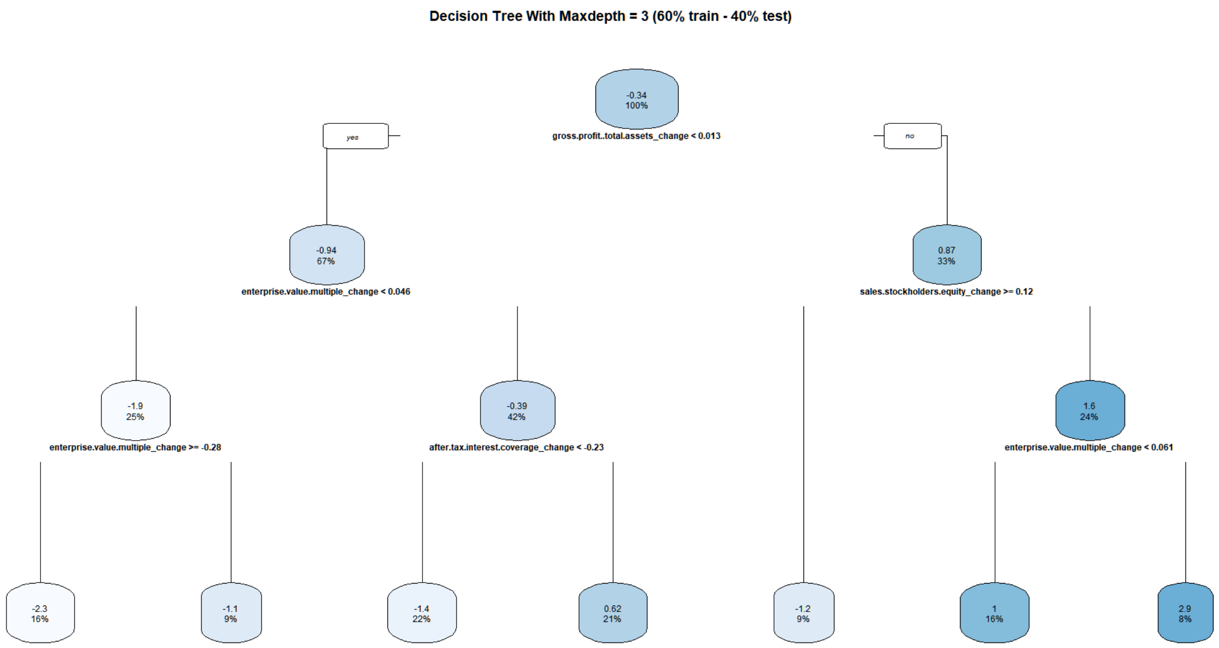



```
tree_60$variable.importance
```

```
##                          interest.coverage.ratio_change
##                                              115.243356
##                       gross.profit..total.assets_change
##                                               91.808274
##                                total.debt..equity_change
##                                               85.484201
##                          total.debt..total.assets_change
##                                               78.073994
##                        operating.margin.before.dep._change
##                                               76.476688
##                         enterprise.value.multiple_change
##                                               71.927426
##                       after.tax.interest.coverage_change
##                                               69.260153
##                                       price..book_change
##                                               68.469444
##                               total.debt..capital_change
##                                               66.263778
##                         sales.stockholders.equity_change
##                                               58.063534
##                            sales.invested.capital_change
##                                               53.111723
##                                      book..market_change
##                                               38.038580
##                                    asset.turnover_change
##                                               37.214478
##      price..operating.earnings..Diluted..Excl..EI._change
##                                               35.536133
##                                  return.on.equity_change
```

```
##                                                     30.430864
##                                          cash.flow.margin_change
##                                                     30.154783
##                                              price..sales_change
##                                                     29.031767
##         after.tax.return.on.average.common.equity_change
##                                                     26.134145
## after.tax.return.on.average.stockholders..equity_change
##                                                     24.123827
##       price..operating.earnings..Basic..Excl..EI._change
##                                                     23.654285
##                                     receivables.turnover_change
##                                                     21.890870
##                                      gross.profit.margin_change
##                                                     20.504817
##                                  P.E..Diluted..Excl..EI._change
##                                                     15.020143
##                                    cash.flow..total.debt_change
##                                                     12.607500
##                                  P.E..Diluted..Incl..EI._change
##                                                     10.923740
##                                    interest..average.LTD_change
##                                                      4.842152
##                             interest..average.total.debt_change
##                                                      4.842152
```

```r
    # Hyperparameter tuning
tree_maxdepth_60 <- rpart(data_train_60$rating_diff ~ ., data = data_train_60, maxdepth = 3, xval = 5)
# rpart.plot(tree_maxdepth_60,
#            main = "Decision Tree With Maxdepth = 3 (60% train - 40% test)",
#            cex = 0.5, tweak = 1.0, compress = TRUE)
```



```r
tree_maxdepth_60$variable.importance
```

```
##                            gross.profit..total.assets_change
##                                                    91.808274
##                               interest.coverage.ratio_change
##                                                    77.420856
##                            operating.margin.before.dep._change
##                                                    76.476688
##                             enterprise.value.multiple_change
##                                                    71.927426
##                             sales.stockholders.equity_change
##                                                    58.063534
##                                sales.invested.capital_change
##                                                    53.111723
##                            after.tax.interest.coverage_change
##                                                    48.247653
##                                        asset.turnover_change
##                                                    37.214478
##      price..operating.earnings..Diluted..Excl..EI._change
##                                                    35.536133
##                                    total.debt..equity_change
##                                                    34.838120
##                                      cash.flow.margin_change
##                                                    30.154783
##                                         price..sales_change
##                                                    29.031767
##          after.tax.return.on.average.common.equity_change
##                                                    26.134145
## after.tax.return.on.average.stockholders..equity_change
##                                                    24.123827
##        price..operating.earnings..Basic..Excl..EI._change
##                                                    23.654285
##                                   total.debt..capital_change
##                                                    23.225414
##                              total.debt..total.assets_change
##                                                    23.225414
##                                 receivables.turnover_change
##                                                    21.890870
##                                   gross.profit.margin_change
##                                                    20.504817
##                               P.E..Diluted..Excl..EI._change
##                                                    15.020143
##                               P.E..Diluted..Incl..EI._change
##                                                    10.923740
##                                 interest..average.LTD_change
##                                                     4.842152
##                           interest..average.total.debt_change
##                                                     4.842152
```

```r
    # Prediction
tree_pred_60 <- predict(tree_60, newdata = data_test_40)
tree_pred_60
```

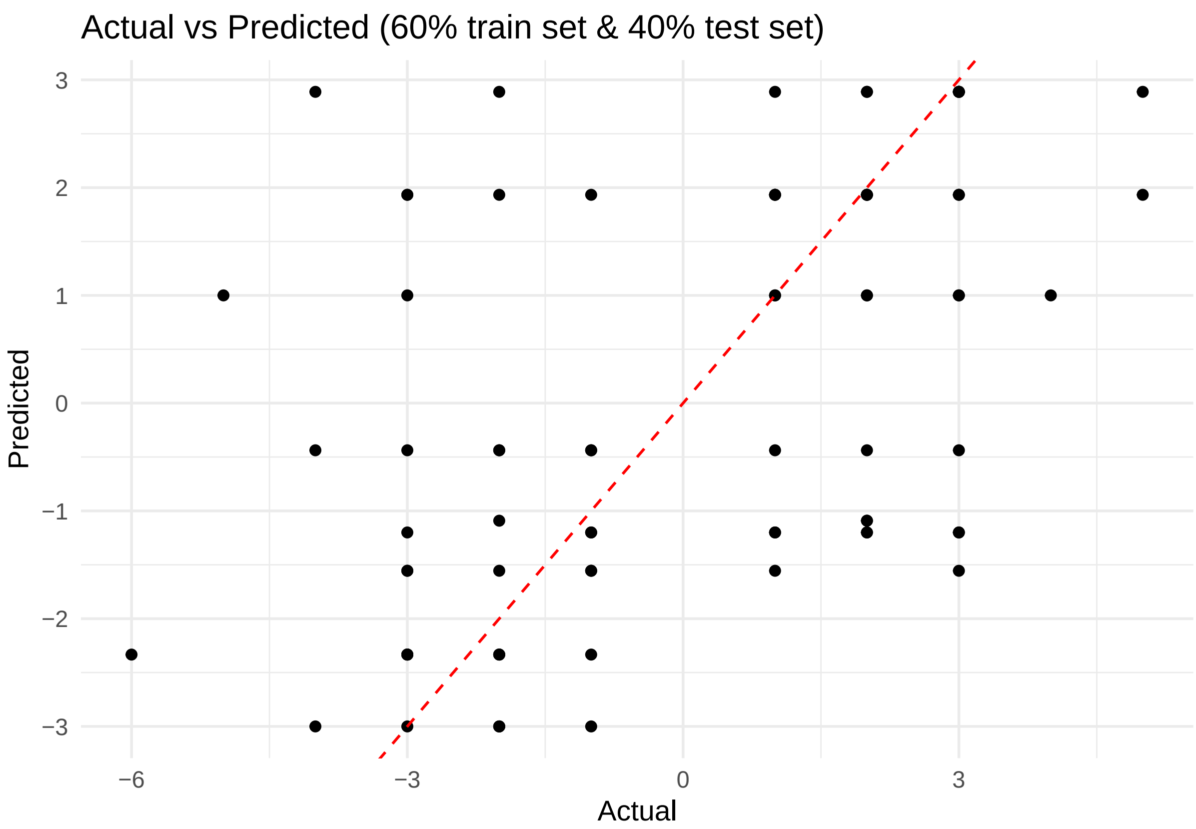
```
##         1         2         3         4         5         6         7         8
## -3.000000 -1.555556 -1.200000  1.933333  1.933333 -2.333333 -1.555556  2.888889
##         9        10        11        12        13        14        15        16
## -1.200000 -0.437500 -2.333333  1.933333 -1.555556  2.888889 -0.437500 -2.333333
##        17        18        19        20        21        22        23        24
## -1.555556  1.933333 -2.333333  1.933333 -1.200000 -2.333333  2.888889  1.933333
##        25        26        27        28        29        30        31        32
## -1.200000  1.000000  1.000000  1.000000 -2.333333  1.933333  1.933333  2.888889
##        33        34        35        36        37        38        39        40
##  2.888889 -1.200000 -1.200000 -3.000000  2.888889 -1.200000 -0.437500 -1.200000
##        41        42        43        44        45        46        47        48
##  1.933333  2.888889  2.888889 -1.555556 -0.437500 -1.090909 -2.333333 -1.090909
##        49        50        51        52        53        54        55        56
## -1.555556  1.933333  1.933333  1.000000 -3.000000  2.888889 -2.333333  1.000000
##        57        58        59        60        61        62        63        64
##  1.000000  1.933333 -0.437500  2.888889  2.888889  1.000000 -0.437500  1.000000
##        65        66        67        68        69        70        71        72
##  1.000000 -1.200000 -3.000000  1.933333 -3.000000 -1.555556 -0.437500 -0.437500
##        73        74        75        76        77
## -0.437500  1.933333 -2.333333 -1.090909 -3.000000
```

```r
tree_pred_md_60 <- predict(tree_maxdepth_60, newdata = data_test_40)
tree_pred_md_60
```

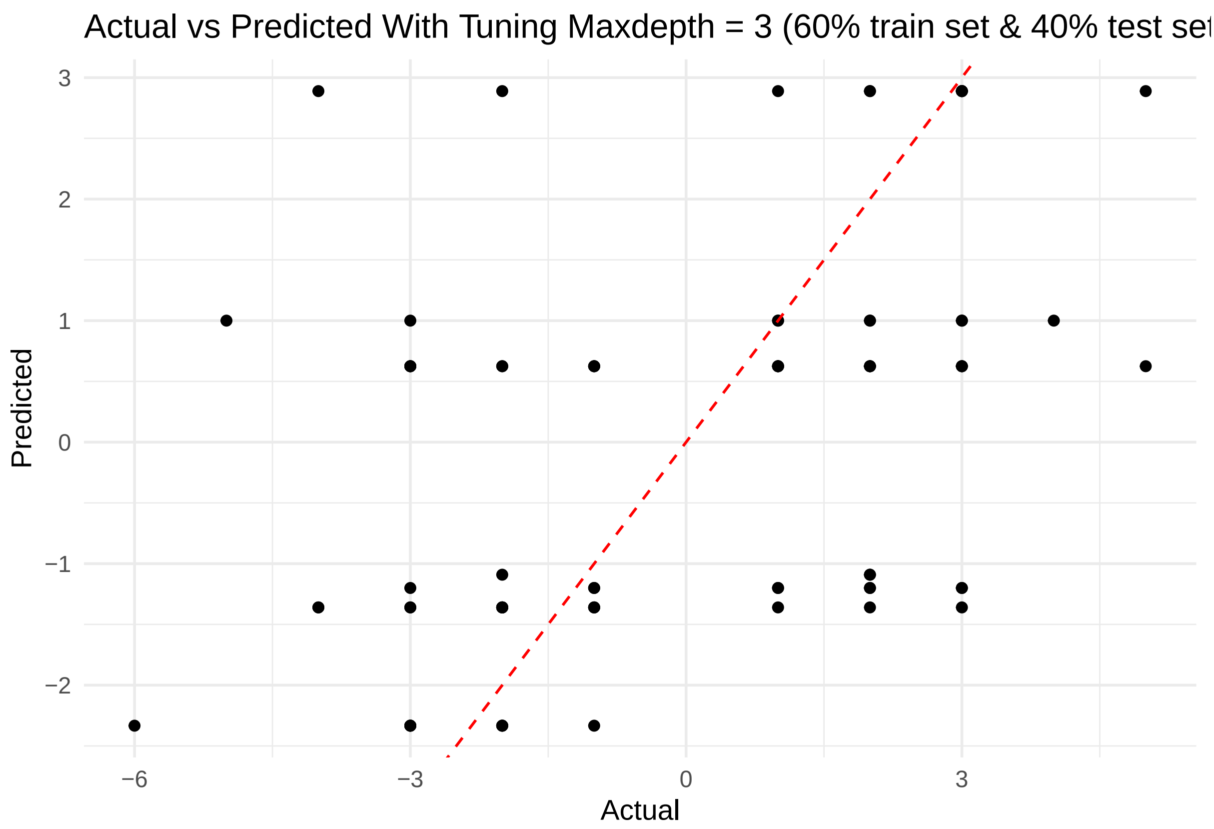
```
##         1         2         3         4         5         6         7         8
## -1.360000  0.625000 -1.200000  0.625000  0.625000 -2.333333  0.625000  2.888889
##         9        10        11        12        13        14        15        16
## -1.200000 -1.360000 -2.333333  0.625000  0.625000  2.888889 -1.360000 -2.333333
##        17        18        19        20        21        22        23        24
##  0.625000  0.625000 -2.333333  0.625000 -1.200000 -2.333333  2.888889  0.625000
##        25        26        27        28        29        30        31        32
## -1.200000  1.000000  1.000000  1.000000 -2.333333  0.625000  0.625000  2.888889
##        33        34        35        36        37        38        39        40
##  2.888889 -1.200000 -1.200000 -1.360000  2.888889 -1.200000 -1.360000 -1.200000
##        41        42        43        44        45        46        47        48
##  0.625000  2.888889  2.888889  0.625000 -1.360000 -1.090909 -2.333333 -1.090909
##        49        50        51        52        53        54        55        56
##  0.625000  0.625000  0.625000  1.000000 -1.360000  2.888889 -2.333333  1.000000
##        57        58        59        60        61        62        63        64
##  1.000000  0.625000 -1.360000  2.888889  2.888889  1.000000 -1.360000  1.000000
##        65        66        67        68        69        70        71        72
##  1.000000 -1.200000 -1.360000  0.625000 -1.360000  0.625000 -1.360000 -1.360000
##        73        74        75        76        77
## -1.360000  0.625000 -2.333333 -1.090909 -1.360000
```

```r
    # Plot
plot_tree_60 <- data.frame(Actual = data_test_40$rating_diff,
                           Predicted = as.vector(tree_pred_60))
ggplot(plot_tree_60, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted (60% train set & 40% test set)",
       x = "Actual",
```

```
       y = "Predicted") +
       theme_minimal()
```




```
plot_tree_md_60 <- data.frame(Actual = data_test_40$rating_diff,
                              Predicted = as.vector(tree_pred_md_60))
ggplot(plot_tree_md_60, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted With Tuning Maxdepth = 3 (60% train set & 40% test set)",
       x = "Actual",
       y = "Predicted") +
       theme_minimal()
```




```r
    # Evaluation
pe_tr_60 <- plot_tree_60$Predicted - plot_tree_60$Actual
rmse_tr_60 <- sqrt(mean(pe_tr_60^2))
cat("Decision Tree (60-40) RMSE:", rmse_tr_60, "\n")
```

```
## Decision Tree (60-40) RMSE: 2.363091
```

```r
table_tr_60 <- data.frame(Model = "Decision Tree (60-40)",
                          RMSE = rmse_tr_60)
print(table_tr_60)
```

```
##                     Model     RMSE
## 1 Decision Tree (60-40) 2.363091
```

```r
pe_tr_md_60 <- plot_tree_md_60$Predicted - plot_tree_md_60$Actual
rmse_tr_md__60 <- sqrt(mean(pe_tr_md_60^2))
cat("Decision Tree With Maxdepth = 3 (60-40) RMSE:", rmse_tr_md__60, "\n")
```

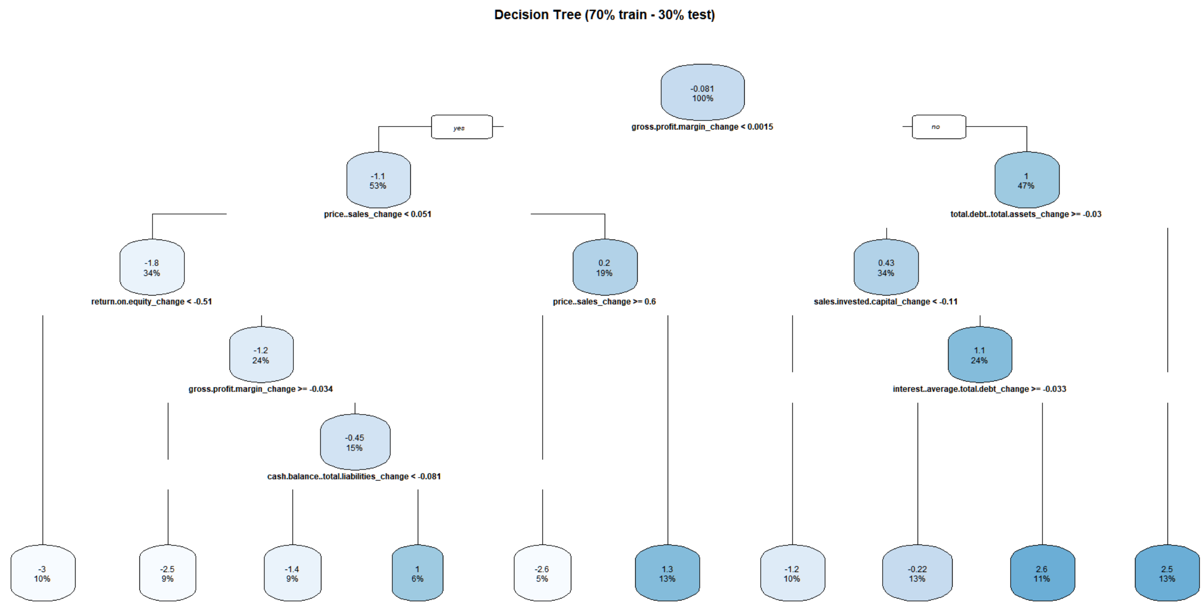
```
## Decision Tree With Maxdepth = 3 (60-40) RMSE: 2.374456
```

```r
table_tr_md_60 <- data.frame(Model = "Decision Tree With Maxdepth = 3 (60-40)",
                             RMSE = rmse_tr_md__60)
print(table_tr_md_60)
```

```
##                                     Model     RMSE
## 1 Decision Tree With Maxdepth = 3 (60-40) 2.374456
```

70-30

```r
tree_70 <- rpart(data_train_70$rating_diff ~ .,
                 data = data_train_70, xval = 5)
# rpart.plot(tree_70,
#            main = "Decision Tree (70% train - 30% test)",
#            cex = 0.5, tweak = 1.0, compress = TRUE)
```

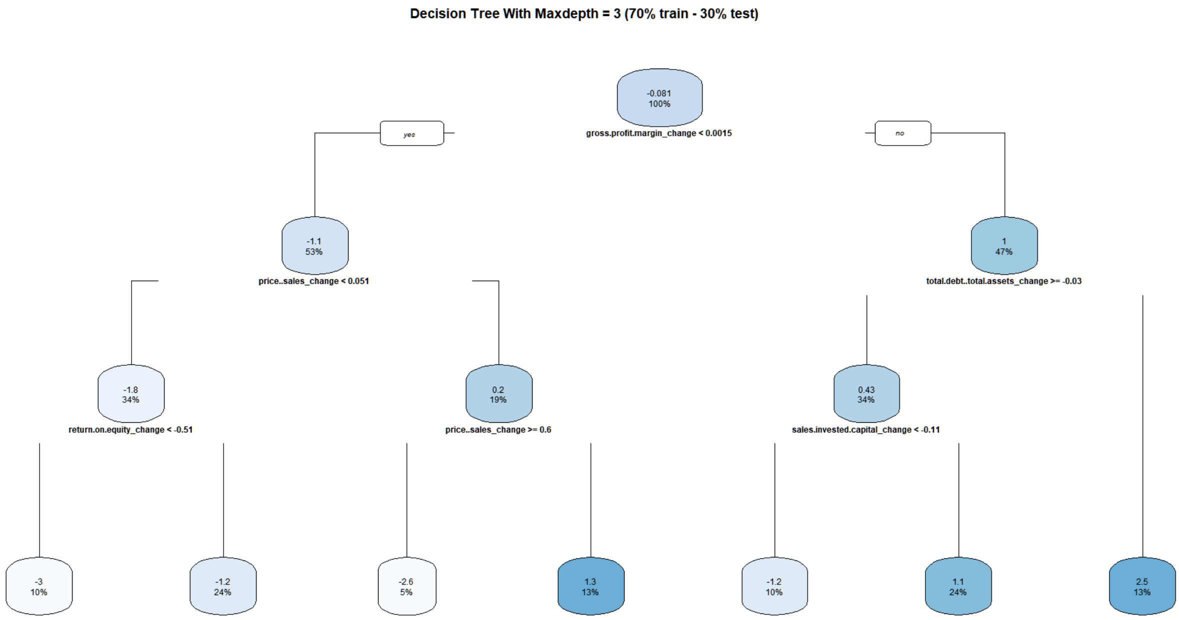



```r
tree_70$variable.importance
```

```
##                               gross.profit.margin_change
##                                                177.99019
##                                     price..sales_change
##                                                136.95293
##                          sales.invested.capital_change
##                                                121.18177
##                      gross.profit..total.assets_change
##                                                120.60900
##                     operating.margin.before.dep._change
##                                                111.20537
##                           interest.coverage.ratio_change
##                                                103.87772
##                              total.debt..equity_change
##                                                 84.55494
##                        total.debt..total.assets_change
##                                                 72.70086
##                    interest..average.total.debt_change
##                                                 65.16768
##         shillers.cyclically.adjusted.P.E.ratio_change
##                                                 65.09140
##                       sales.stockholders.equity_change
##                                                 58.33740
```

```
##                                  interest..average.LTD_change
##                                                      56.47865
##                                       cash.flow.margin_change
##                                                      52.63817
##                                    total.debt..capital_change
##                                                      52.55154
##                                P.E..Diluted..Excl..EI._change
##                                                      51.11381
##                              enterprise.value.multiple_change
##                                                      48.82786
##                                         asset.turnover_change
##                                                      47.17473
##                                       price..cash.flow_change
##                                                      42.67120
##                        cash.balance..total.liabilities_change
##                                                      37.23000
##                                           book..market_change
##                                                      36.61786
##                                            price..book_change
##                                                      32.38473
##                                       return.on.equity_change
##                                                      30.90082
##          after.tax.return.on.average.common.equity_change
##                                                      28.69361
## after.tax.return.on.average.stockholders..equity_change
##                                                      28.69361
##        price..operating.earnings..Basic..Excl..EI._change
##                                                      27.40246
##      price..operating.earnings..Diluted..Excl..EI._change
##                                                      27.40246
##                                P.E..Diluted..Incl..EI._change
##                                                      26.48641
##                    free.cash.flow..operating.cash.flow_change
##                                                      26.06707
##                              after.tax.interest.coverage_change
##                                                      19.86481
##                                   cash.flow..total.debt_change
##                                                      17.52083
##                         common.equity.invested.capital_change
##                                                      15.32779
```

```
    # Hyperparameter tuning
tree_maxdepth_70 <- rpart(data_train_70$rating_diff ~ .,
                          data = data_train_70, maxdepth = 3, xval = 5)
# rpart.plot(tree_maxdepth_70,
#            main = "Decision Tree With Maxdepth = 3 (70% train - 30% test)",
#            cex = 0.5, tweak = 1.0, compress = TRUE)
```




```
tree_maxdepth_70$variable.importance
```

```
##                            gross.profit.margin_change
##                                            146.471441
##                                    price..sales_change
##                                            136.952925
##                   operating.margin.before.dep._change
##                                            100.699116
##                        gross.profit..total.assets_change
##                                             94.343376
##                         sales.invested.capital_change
##                                             77.736654
##                        interest.coverage.ratio_change
##                                             72.346467
##            shillers.cyclically.adjusted.P.E.ratio_change
##                                             65.091398
##                       sales.stockholders.equity_change
##                                             58.337400
##                              total.debt..equity_change
##                                             56.527858
##                        total.debt..total.assets_change
##                                             55.180027
##                                cash.flow.margin_change
##                                             52.638174
##                          P.E..Diluted..Excl..EI._change
##                                             51.113810
##                                price..cash.flow_change
##                                             42.671202
##                                    book..market_change
##                                             36.617860
##                                     price..book_change
##                                             32.384727
```

```
##                                  return.on.equity_change
##                                                30.900815
##                after.tax.return.on.average.common.equity_change
##                                                28.693614
## after.tax.return.on.average.stockholders..equity_change
##                                                28.693614
##          price..operating.earnings..Basic..Excl..EI._change
##                                                27.402462
##        price..operating.earnings..Diluted..Excl..EI._change
##                                                27.402462
##                          P.E..Diluted..Incl..EI._change
##                                                26.486413
##                             total.debt..capital_change
##                                                24.524457
##                                  asset.turnover_change
##                                                21.107655
##                    after.tax.interest.coverage_change
##                                                19.864810
##                    common.equity.invested.capital_change
##                                                15.327785
##                       enterprise.value.multiple_change
##                                                14.071770
##                    cash.balance..total.liabilities_change
##                                                 9.196671
```

```
    # Prediction
tree_pred_70 <- predict(tree_70, newdata = data_test_30)
tree_pred_70
```

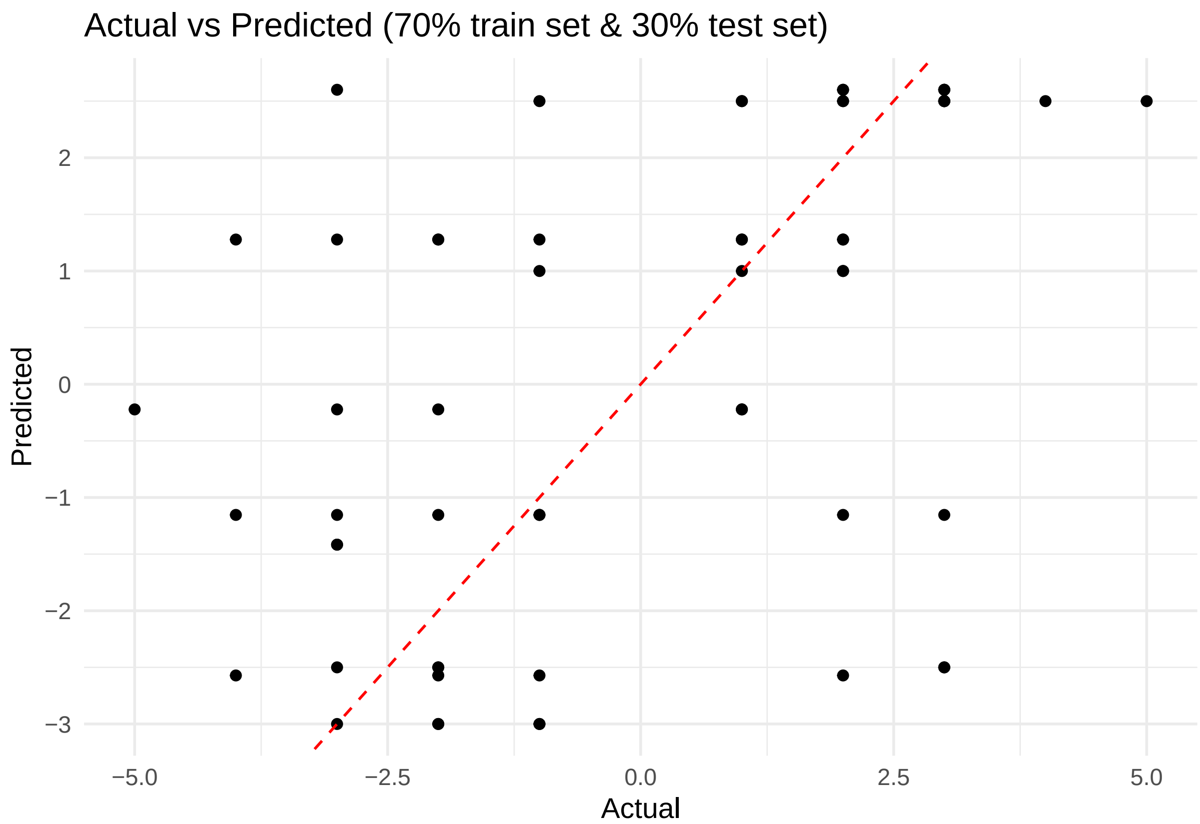
```
##          1          2          3          4          5          6          7
## -1.1538462  2.6000000 -0.2222222  1.2777778 -2.5714286 -0.2222222  2.5000000
##          8          9         10         11         12         13         14
##  2.6000000 -2.5000000 -1.1538462  2.5000000 -1.1538462  2.5000000  2.5000000
##         15         16         17         18         19         20         21
## -3.0000000 -2.5714286 -3.0000000  2.5000000  2.6000000 -1.1538462 -0.2222222
##         22         23         24         25         26         27         28
##  1.0000000  2.5000000  2.5000000  1.0000000  1.2777778  1.0000000 -2.5000000
##         29         30         31         32         33         34         35
## -3.0000000 -2.5714286 -1.4166667 -2.5000000 -0.2222222 -3.0000000  1.2777778
##         36         37         38         39         40         41         42
## -1.4166667 -0.2222222  2.5000000  1.0000000 -1.1538462  2.6000000 -2.5000000
##         43         44         45         46         47         48         49
##  2.6000000  1.2777778  1.2777778  2.5000000  2.6000000 -2.5714286  2.5000000
##         50         51         52         53         54         55         56
##  1.2777778 -1.1538462  1.2777778 -3.0000000  1.2777778  2.5000000 -1.1538462
##         57         58
##  1.2777778 -2.5000000
```

```
tree_pred_md_70 <- predict(tree_maxdepth_70, newdata = data_test_30)
tree_pred_md_70
```

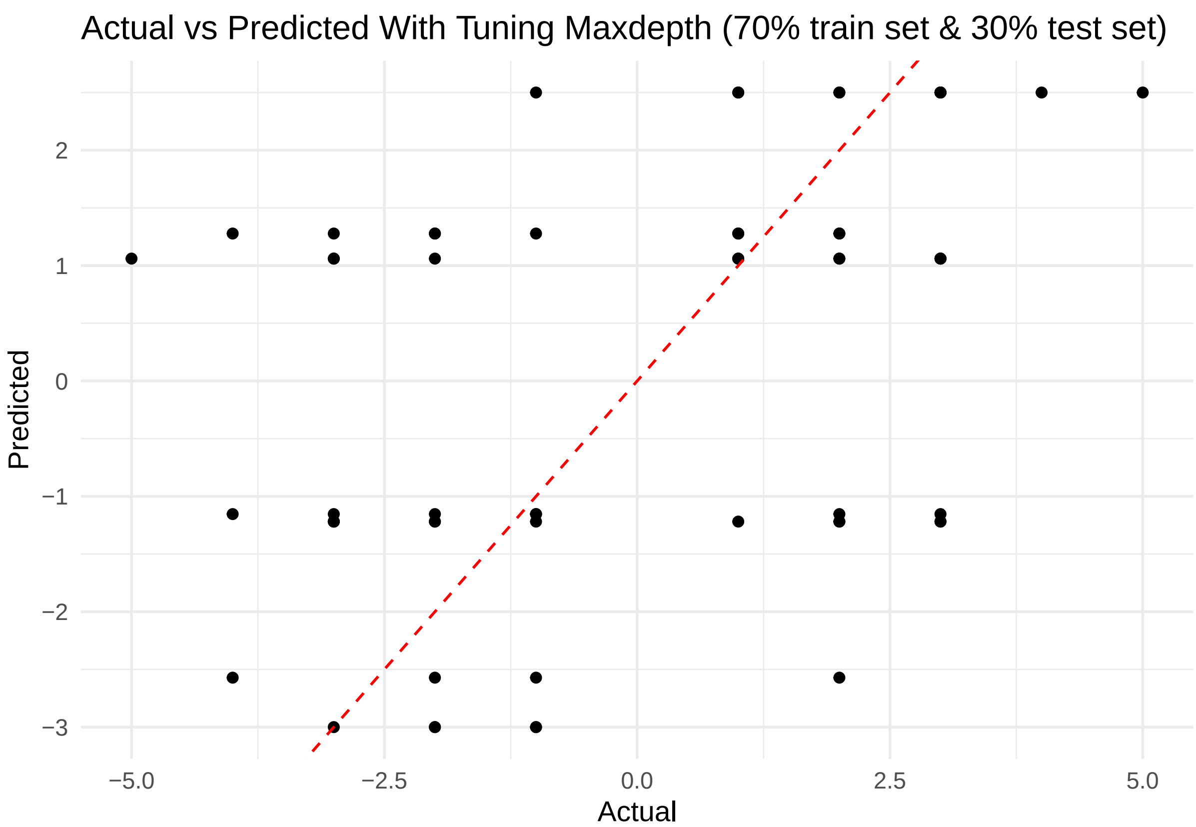
```
##         1         2         3         4         5         6         7         8
## -1.153846  1.060606  1.060606  1.277778 -2.571429  1.060606  2.500000  1.060606
```

```
##         9        10        11        12        13        14        15        16
## -1.218750 -1.153846  2.500000 -1.153846  2.500000  2.500000 -3.000000 -2.571429
##        17        18        19        20        21        22        23        24
## -3.000000  2.500000  1.060606 -1.153846  1.060606 -1.218750  2.500000  2.500000
##        25        26        27        28        29        30        31        32
## -1.218750  1.277778 -1.218750 -1.218750 -3.000000 -2.571429 -1.218750 -1.218750
##        33        34        35        36        37        38        39        40
##  1.060606 -3.000000  1.277778 -1.218750  1.060606  2.500000 -1.218750 -1.153846
##        41        42        43        44        45        46        47        48
##  1.060606 -1.218750  1.060606  1.277778  1.277778  2.500000  1.060606 -2.571429
##        49        50        51        52        53        54        55        56
##  2.500000  1.277778 -1.153846  1.277778 -3.000000  1.277778  2.500000 -1.153846
##        57        58
##  1.277778 -1.218750
```

```r
    # Plot
plot_tree_70 <- data.frame(Actual = data_test_30$rating_diff, Predicted = as.vector(tree_pred_70))
ggplot(plot_tree_70, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted (70% train set & 30% test set)",
       x = "Actual",
       y = "Predicted") +
       theme_minimal()
```



```r
plot_tree_md_70 <- data.frame(Actual = data_test_30$rating_diff, Predicted = as.vector(tree_pred_md_70))
ggplot(plot_tree_md_70, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted With Tuning Maxdepth (70% train set & 30% test set)",
       x = "Actual",
       y = "Predicted") +
       theme_minimal()
```




```r
    # Evaluation
pe_tr_70 <- plot_tree_70$Predicted - plot_tree_70$Actual
rmse_tr_70 <- sqrt(mean(pe_tr_70^2))
cat("Decision Tree (70-30) RMSE:", rmse_tr_70, "\n")
```

```
## Decision Tree (70-30) RMSE: 2.339692
```

```r
table_tr_70 <- data.frame(Model = "Decision Tree (70-30)",
                          RMSE = rmse_tr_70)
print(table_tr_70)
```

```
##                   Model     RMSE
## 1 Decision Tree (70-30) 2.339692
```

```r
pe_tr_md_70 <- plot_tree_md_70$Predicted - plot_tree_md_70$Actual
rmse_tr_md_70 <- sqrt(mean(pe_tr_md_70^2))
cat("Decision Tree With Maxdepth = 3 (70-30) RMSE:", rmse_tr_md_70, "\n")
```

```
## Decision Tree With Maxdepth = 3 (70-30) RMSE: 2.421682
```

```r
table_tr_md_70 <- data.frame(Model = "Decision Tree With Maxdepth = 3 (70-30)",
                             RMSE = rmse_tr_md_70)
print(table_tr_md_70)
```

```
##                                    Model     RMSE
## 1 Decision Tree With Maxdepth = 3 (70-30) 2.421682
```

Random Forest

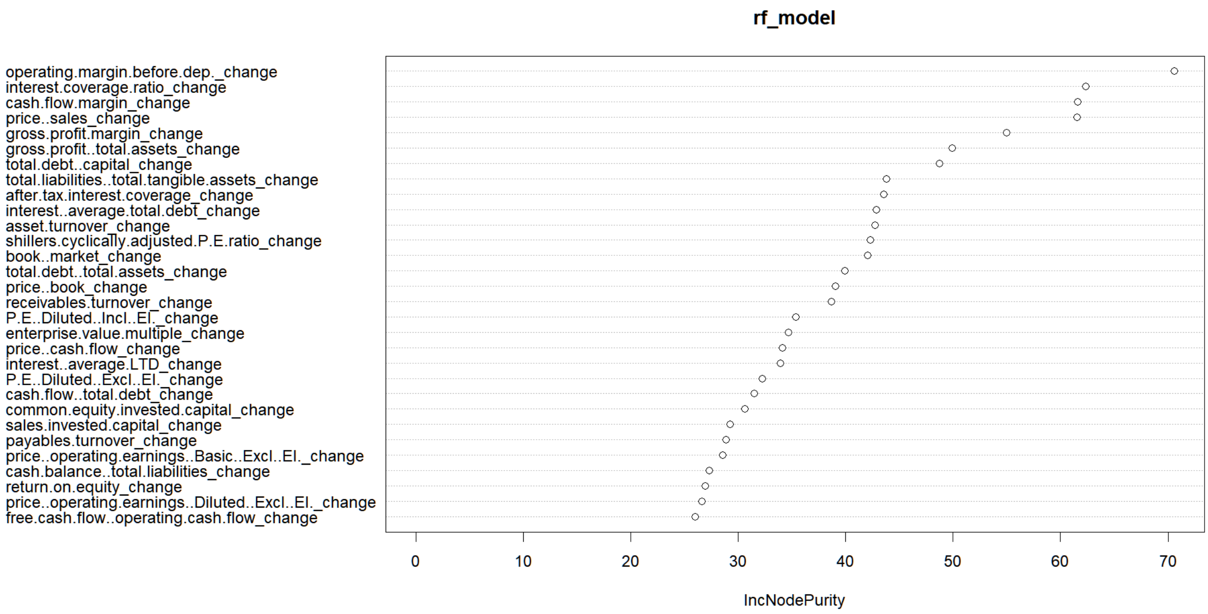
Full sample

```r
rf_model <- randomForest(data$rating_diff ~ .,
                         data,
                         ntree = 500,
                         mtry = sqrt(ncol(data)))
rf_model$importance
```

```
##                                                 IncNodePurity
## after.tax.interest.coverage_change                   39.18373
## interest.coverage.ratio_change                       58.85127
## cash.flow..total.debt_change                         29.51348
## operating.margin.before.dep._change                  70.28877
## return.on.equity_change                              25.46974
## total.debt..total.assets_change                      38.25651
## book..market_change                                  44.85860
## interest..average.LTD_change                         34.18563
## interest..average.total.debt_change                  38.69655
## cash.balance..total.liabilities_change               28.66695
## free.cash.flow..operating.cash.flow_change           27.43812
## total.liabilities..total.tangible.assets_change      42.88594
## total.debt..capital_change                           41.03476
## total.debt..equity_change                            23.67487
## asset.turnover_change                                39.97680
## receivables.turnover_change                          40.60858
## payables.turnover_change                             28.11152
## sales.invested.capital_change                        32.95468
## sales.stockholders.equity_change                     25.90717
## price..book_change                                   39.57949
## shillers.cyclically.adjusted.P.E.ratio_change        40.09778
## enterprise.value.multiple_change                     30.60732
## price..operating.earnings..Basic..Excl..EI._change   30.30585
## price..operating.earnings..Diluted..Excl..EI._change 29.00550
## P.E..Diluted..Excl..EI._change                       35.75553
```

```
## P.E..Diluted..Incl..EI._change                           36.18619
## price..sales_change                                      64.18365
## price..cash.flow_change                                  34.85443
## gross.profit.margin_change                               59.26180
## after.tax.return.on.average.common.equity_change         23.91938
## after.tax.return.on.average.stockholders..equity_change  25.81173
## gross.profit..total.assets_change                        52.10683
## common.equity.invested.capital_change                    38.78540
## cash.flow.margin_change                                  49.01125
```

```
    # Variable Importance Plot
# rf_var_imp_plot <- randomForest::varImpPlot(rf_model)
```



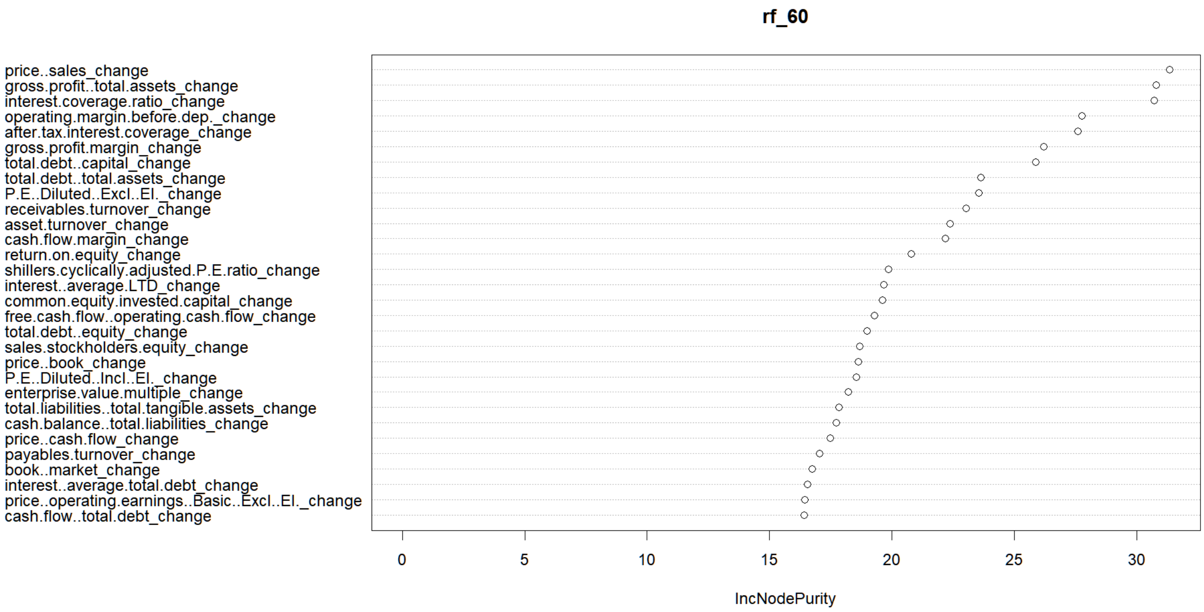
60-40

```
rf_60 <- randomForest(data_train_60$rating_diff ~ .,
                      data = data_train_60,
                      ntree = 500,
                      mtry = sqrt(ncol(data_train_60)))
rf_60$importance
```

```
##                                          IncNodePurity
## after.tax.interest.coverage_change            27.49397
## interest.coverage.ratio_change                35.43278
## cash.flow..total.debt_change                  16.99805
## operating.margin.before.dep._change           30.18004
## return.on.equity_change                       19.36530
## total.debt..total.assets_change               22.20977
## book..market_change                           19.01845
## interest..average.LTD_change                  20.79743
```

```
## interest..average.total.debt_change                        16.51152
## cash.balance..total.liabilities_change                     18.76762
## free.cash.flow..operating.cash.flow_change                 20.95775
## total.liabilities..total.tangible.assets_change            20.38889
## total.debt..capital_change                                 25.54417
## total.debt..equity_change                                  20.31248
## asset.turnover_change                                      19.45300
## receivables.turnover_change                                21.48377
## payables.turnover_change                                   16.17236
## sales.invested.capital_change                              17.59067
## sales.stockholders.equity_change                           17.37095
## price..book_change                                         17.26524
## shillers.cyclically.adjusted.P.E.ratio_change              19.21993
## enterprise.value.multiple_change                           17.58758
## price..operating.earnings..Basic..Excl..EI._change         15.72230
## price..operating.earnings..Diluted..Excl..EI._change       17.02051
## P.E..Diluted..Excl..EI._change                             24.16779
## P.E..Diluted..Incl..EI._change                             18.92618
## price..sales_change                                        30.70390
## price..cash.flow_change                                    17.43564
## gross.profit.margin_change                                 25.93454
## after.tax.return.on.average.common.equity_change           14.54254
## after.tax.return.on.average.stockholders..equity_change    15.51924
## gross.profit..total.assets_change                          27.46229
## common.equity.invested.capital_change                      17.32661
## cash.flow.margin_change                                    21.61605
```

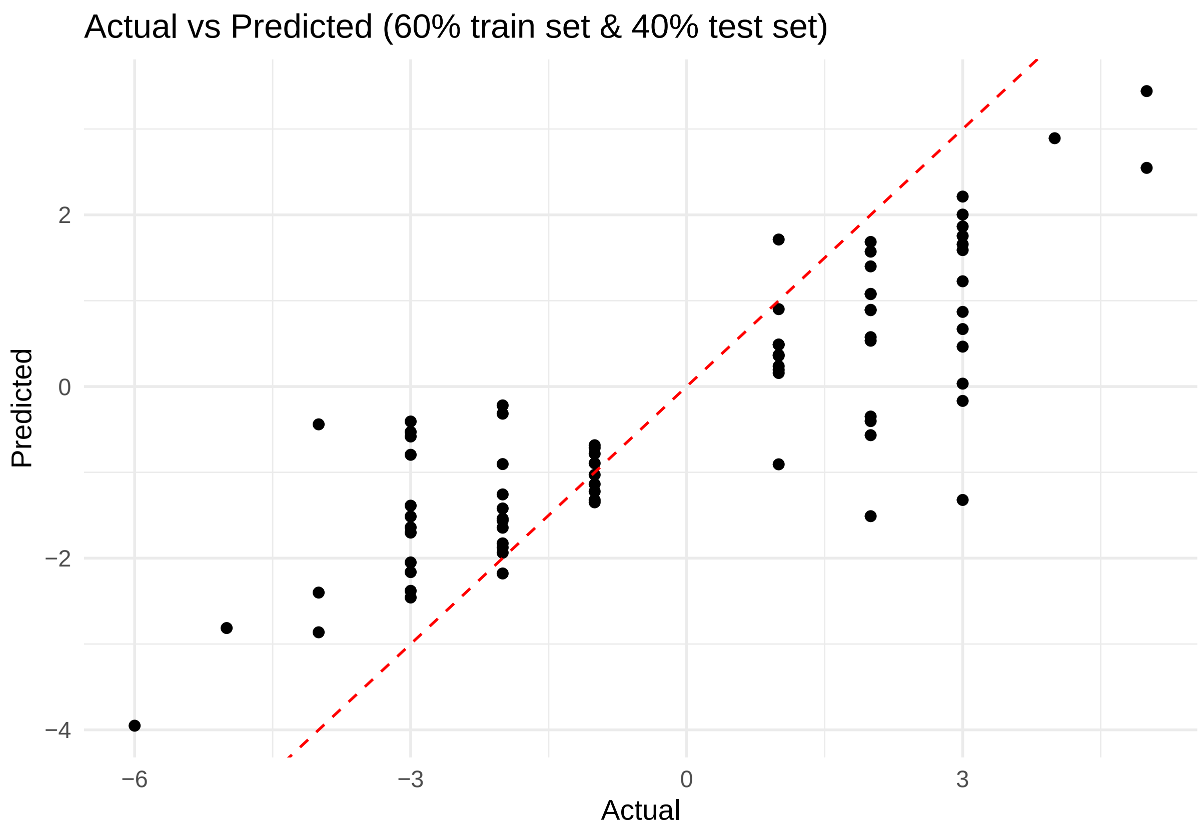
```r
    # Variable Importance Plot
# rf_60_var_imp_plot <- randomForest::varImpPlot(rf_60)
```




```r
    # Prediction
rf_pred_60 <- predict(rf_60, newdata = data_test_40)
rf_pred_60
```

```
##          1          2          3          4          5          6
## -1.82733333  1.65676667 -0.56713333 -0.40310000 -0.22000000 -2.45826667
##          7          8          9         10         11         12
## -1.38803333  3.44143333 -0.16770000 -1.93680000 -1.51460000  0.19383333
##         13         14         15         16         17         18
##  0.48973333  1.07640000  1.22606667 -3.95136667 -1.70203333  0.35773333
##         19         20         21         22         23         24
## -1.13760000 -1.32153333 -0.34960000 -1.25720000 -0.31643333  0.53336667
##         25         26         27         28         29         30
## -0.90646667  0.90130000  1.07930000  0.46446667 -0.90380000  2.54700000
##         31         32         33         34         35         36
##  1.39960000  0.66936667  2.00306667 -1.32230000 -0.71530000 -1.34936667
##         37         38         39         40         41         42
##  2.21293333  0.03333333 -0.89443333  0.15733333  0.23706667  1.71253333
##         43         44         45         46         47         48
##  1.57136667 -1.02863333  0.48630000  0.57506667 -1.54010000 -2.17816667
##         49         50         51         52         53         54
## -1.64480000 -2.04926667 -0.52893333 -0.40763333 -0.58213333 -0.44063333
##         55         56         57         58         59         60
## -2.38006667 -2.81370000  1.68420000  0.89346667 -2.86340000  1.75423333
##         61         62         63         64         65         66
##  1.86436667  0.86956667 -0.79530000  0.36740000  2.89253333 -1.64143333
##         67         68         69         70         71         72
## -2.40066667 -0.68410000 -1.56176667 -0.78340000 -1.22266667 -1.87560000
##         73         74         75         76         77
## -1.51050000  1.58920000 -2.16216667  0.88920000 -1.42070000
```

```
    # Plot
plot_rf_60 <- data.frame(Actual = data_test_40$rating_diff,
                         Predicted = as.vector(rf_pred_60))
ggplot(plot_rf_60, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted (60% train set & 40% test set)",
       x = "Actual",
       y = "Predicted") +
       theme_minimal()
```




```
# Evaluation
pe_rf_60 <- plot_rf_60$Predicted - plot_rf_60$Actual
rmse_rf_60 <- sqrt(mean(pe_rf_60^2))
cat("Random Forest (60-40) RMSE:", rmse_rf_60, "\n")
```

```
## Random Forest (60-40) RMSE: 1.572859
```

```
table_rf_60 <- data.frame(Model = "Random Forest (60-40)",
                          RMSE = rmse_rf_60)
print(table_rf_60)
```

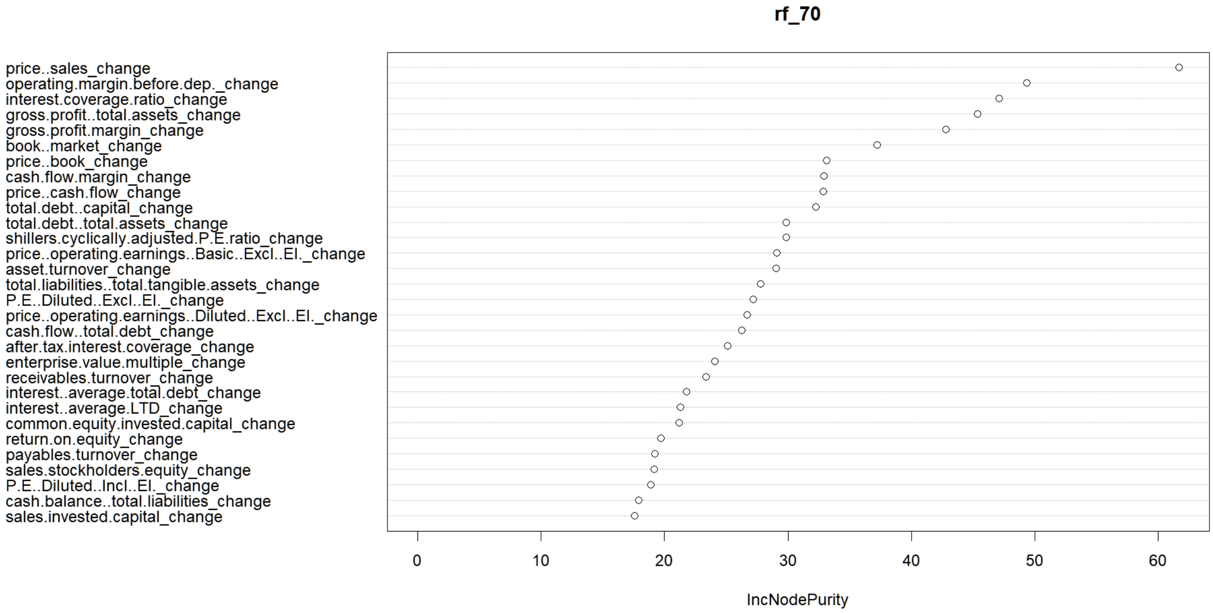
```
##                   Model     RMSE
## 1 Random Forest (60-40) 1.572859
```

70-30

```
rf_70 <- randomForest(data_train_70$rating_diff ~ .,
                      data = data_train_70,
                      ntree = 500,
                      mtry = sqrt(ncol(data_train_70)))
rf_70$importance
```

```
##                                                          IncNodePurity
## after.tax.interest.coverage_change                           21.21509
## interest.coverage.ratio_change                               43.50948
## cash.flow..total.debt_change                                 24.21416
## operating.margin.before.dep._change                          51.32184
## return.on.equity_change                                      22.21652
## total.debt..total.assets_change                              29.84490
## book..market_change                                          36.27567
## interest..average.LTD_change                                 20.87641
## interest..average.total.debt_change                          25.10464
## cash.balance..total.liabilities_change                       17.99276
## free.cash.flow..operating.cash.flow_change                   14.95837
## total.liabilities..total.tangible.assets_change              26.92639
## total.debt..capital_change                                   33.76045
## total.debt..equity_change                                    22.42057
## asset.turnover_change                                        32.18519
## receivables.turnover_change                                  22.16384
## payables.turnover_change                                     21.27851
## sales.invested.capital_change                                21.00825
## sales.stockholders.equity_change                             21.86910
## price..book_change                                           31.17700
## shillers.cyclically.adjusted.P.E.ratio_change                26.74118
## enterprise.value.multiple_change                             22.64152
## price..operating.earnings..Basic..Excl..EI._change           23.35586
## price..operating.earnings..Diluted..Excl..EI._change         25.61400
## P.E..Diluted..Excl..EI._change                               26.97796
## P.E..Diluted..Incl..EI._change                               26.03189
## price..sales_change                                          65.31396
## price..cash.flow_change                                      30.92882
## gross.profit.margin_change                                   44.73555
## after.tax.return.on.average.common.equity_change             14.29065
## after.tax.return.on.average.stockholders..equity_change      13.23672
## gross.profit..total.assets_change                            42.93626
## common.equity.invested.capital_change                        20.60591
## cash.flow.margin_change                                      34.48007
```

```
    # Variable Importance Plot
# rf_70_var_imp_plot <- randomForest::varImpPlot(rf_70)
```

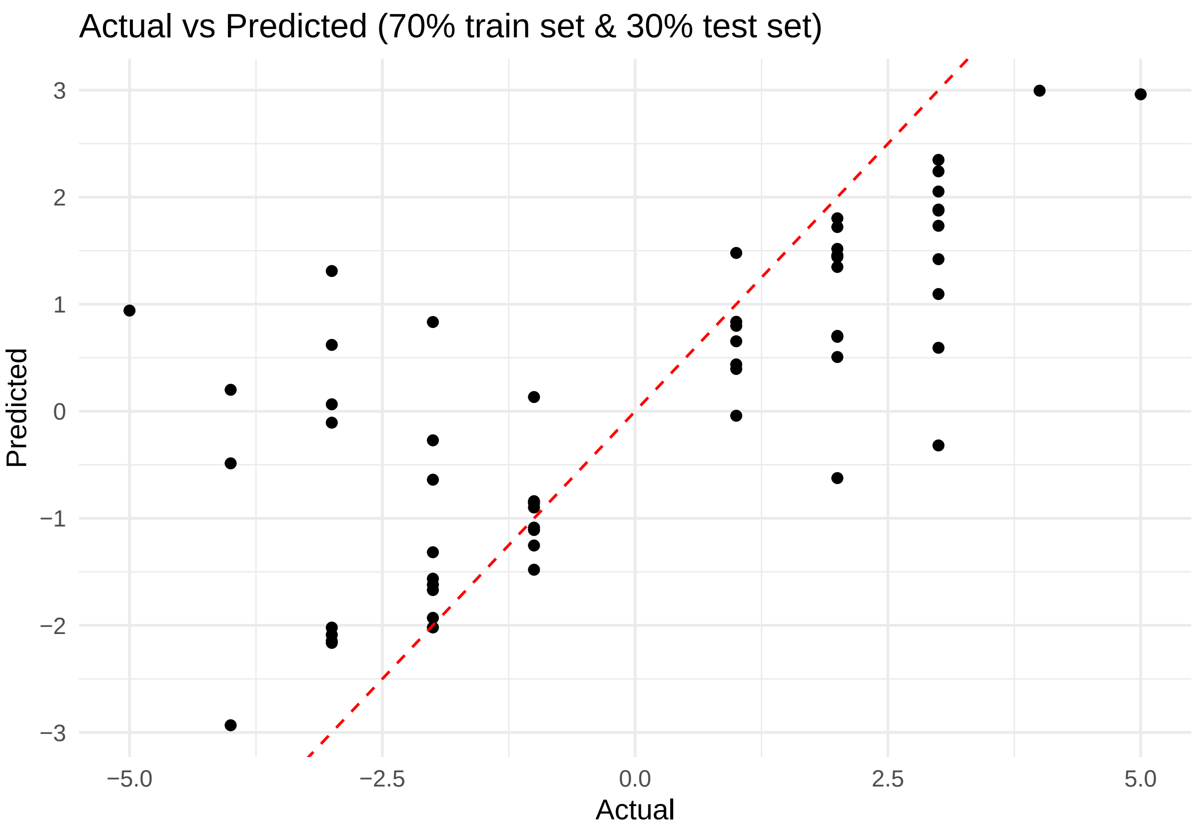



```
    # Prediction
rf_pred_70 <- predict(rf_70, newdata = data_test_30)
rf_pred_70
```

```
##          1          2          3          4          5          6
##  0.59306667  1.45776667 -1.31686667 -0.63910000 -0.62410000  0.83573333
##          7          8          9         10         11         12
##  0.39520000  1.80270000 -0.31898095  0.83376667  2.96093333  1.44133333
##         13         14         15         16         17         18
##  2.05290000  2.34843333 -1.10900000  0.13313333 -1.48030000  1.09530000
##         19         20         21         22         23         24
##  2.24123333 -1.25373333  0.79813333 -0.04164762  1.47890000  1.51713333
##         25         26         27         28         29         30
## -0.85913333  0.43763333  0.50686667 -1.67016667 -2.01940000 -1.56390000
##         31         32         33         34         35         36
## -2.14651429 -0.10633333  0.61996667  0.06520000  0.20066667 -2.16340000
##         37         38         39         40         41         42
##  0.94073333  1.72113333  0.69530000 -0.48640000  1.73266667  1.42075238
##         43         44         45         46         47         48
##  1.87493333 -2.08773333  0.65383333  2.99480000  1.31020000 -2.93303333
##         49         50         51         52         53         54
## -0.89870000 -0.27140000 -0.83963333 -1.08660000 -1.92980000  0.70440000
##         55         56         57         58
##  1.88426667 -2.02073333  1.34830000 -1.62000000
```

```
    # Plot
plot_rf_70 <- data.frame(Actual = data_test_30$rating_diff,
                         Predicted = as.vector(rf_pred_70))
ggplot(plot_rf_70, aes(x = Actual, y = Predicted)) +
       geom_point() +
       geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
       labs(title = "Actual vs Predicted (70% train set & 30% test set)",
       x = "Actual",
```

```r
       y = "Predicted") +
       theme_minimal()
```




```r
    # Evaluation
pe_rf_70 <- plot_rf_70$Predicted - plot_rf_70$Actual
rmse_rf_70 <- sqrt(mean(pe_rf_70^2))
cat("Random Forest (70-30) RMSE:", rmse_rf_70, "\n")
```

```
## Random Forest (70-30) RMSE: 1.78454
```

```r
table_rf_70 <- data.frame(Model = "Random Forest (70-30)",
                          RMSE = rmse_rf_70)
print(table_rf_70)
```

```
##                   Model    RMSE
## 1 Random Forest (70-30) 1.78454
```

Summary

```r
result_table <- rbind.data.frame(table_tr_60, table_tr_md_60,
                                 table_tr_70, table_tr_md_70,
```

```
                                   table_rf_60, table_rf_70)
result_table
```

```
##                                      Model     RMSE
## 1                   Decision Tree (60-40) 2.363091
## 2 Decision Tree With Maxdepth = 3 (60-40) 2.374456
## 3                   Decision Tree (70-30) 2.339692
## 4 Decision Tree With Maxdepth = 3 (70-30) 2.421682
## 5                   Random Forest (60-40) 1.572859
## 6                   Random Forest (70-30) 1.784540
```