W5451 – **Statistical Learning with R and Python**,
winter semester 2022/2023

Applied Project II: Application of Decision Tree, Lasso and Ridge Regression on Different Dataset using R Programming.

Examiner: Prof. Dr. Yuanhua Feng

Assistant: Mr. Sebastian Letmathe

Project beginning: January 23, 2023

Project due: February 20, 2023, 11:59 p.m.

Submitted by Group No. 06:

Name	Matriculation Number	Contribution
Trusha Patel	6872844	Code and text for all problem

Contents

Introduction

As technology continues to advance, understanding the relationships between variables is becoming increasingly important. This is critical for creating accurate and reliable models that can be used for prediction and decision-making. In this paper, we aim to investigate the relationships between a set of predictors and a response variable using different statistical models. This paper focus on two specific models: Decision Tree and Ridge and Lasso regression. These models will be applied to two different datasets to demonstrate their usefulness and effectiveness.

Before diving into the examples, we provide a brief explanation of the relevant concepts and theories from chapters 6 to 8 and 10 of the book "An Introduction to Statistical Learning with Applications in R"(James et al., 2013) .These chapters cover topics such as linear model selection and regularization, moving beyond linearity, tree-based methods and deep learning neural networks.

Next, we present two examples where Decision Tree and Ridge and Lasso regression models were fitted to different datasets. The first example involves fitting Ridge and Lasso regression models to a Song dataset to predict a song's popularity based on various features. The second example involves fitting a Decision Tree model to a Pokémon dataset to predict whether a Pokémon is legendary or not based on its various attributes. Finally, empirical analysis of those examples as well as some concluding comments were provided on the overall study.

Summary of Chapters 6, 7, 8, And 10

Chapter 6: Linear Model Selection and Regularization

The standard linear model is a common way to describe the relationship between a response variable and a set of predictor variables in regression analysis. However, alternative fitting procedures can improve both the prediction accuracy and model interpretability. The prediction accuracy can be improved for future predictions by constraining or shrinking the estimated coefficients, which can reduce variance to the detriment of an insignificant increase in bias. By removing irrelevant variables through feature or variable selection, a simpler and more interpretable model can be achieved. These alternative fitting procedures can be divided into three categories: subset selection, shrinkage, and dimension reduction.

Subset selection involves identifying a subset of predictors that are believed to be related to the response and then fitting a model using least squares on this reduced set of variables. It is further decided into best subset and stepwise model selection procedures. Best subset selection is a method for selecting the best model among all possible combinations of predictors in a multiple regression analysis. The method involves fitting a separate least squares regression for each combination of predictors and comparing the resulting models to find the best one. However, this can be a challenging problem, as the number of models considered can quickly become

very large, especially as the number of predictors increases. Best subset selection is not practical for large p due to computational difficulties and statistical problems like overfitting.

Stepwise methods, which consider a limited set of models, are a more attractive alternative. Forward stepwise selection is a computationally efficient method it starts with no predictors and adds one at a time, choosing the variable that gives the greatest improvement to the fit, while in backward selection variables are removed one by one starting from the full set. Hybrid Approaches is a combination of both forward and backward selection. The optimal model can be selected either by using criterias like AIC,BIC,C_p, and adjusted R^2 or validation and cross-validation methods.

Shrinkage involves fitting a model involving all p predictors but shrinking the estimated coefficients towards zero, reducing variance and performing variable selection. The techniques used are ridge regression and the lasso. Under ridge regression coefficients are estimated by minimizing tuning parameter. The lasso uses penalty l_1 which has a effect on forcing some of the coefficients to exactly equals zero when tuning parameter is sufficiently large. Cross validation method can be used for selecting the tuning parameter.

Dimension reduction involves projecting the p predictors into an M-dimensional subspace ($M < p$), computing linear combinations of the variables, and fitting a linear regression model using these M projections as predictors. Two approaches for this method are principal components and partial least squares. The principal components analysis is an approach for deriving a low -dimensional set of features from a large set of variables.

The principal components regression (PCR) involves constructing the first principal components and then using these components as the predictors in LRM that is fit using least squares. Its assumed that the directions in which $X_1, \dots, X_p$, shows the most variation are the directions that are associated with Y. The partial least square (PLS) is a supervised alternative and attempts to find directions that helps both the response and predictors. The high dimensional is when data set contains more features than observations. In high dimensional setting number of featuers p is larger than the number of observations n, this can overfit the data.

Chapter 7: Moving Beyond Linearity

This chapter discusses various methods of relaxing the linearity assumption in linear models to obtain improved results while maintaining interpretability. Simple extensions such as polynomial regression, step, and basis functions, as well as more flexible methods like splines, local regression, and generalized additive models can be used to achieve this goal.

Polynomial regression extends the linear model by adding extra predictors by raising the original predictors to a power, providing a non-linear fit to the data. Step functions divide the range of a variable into K distinct regions to produce a qualitative variable, while basis functions have a family of functions or transformations that can be applied to a variable X : $b_1(X), b_2(X), \dots, b_K(X)$. As basis functions can be chosen ahead of time it is fixed and known.

Regression splines, an extension of polynomials and step-functions, provide a more flexible fit by dividing the range of X into K distinct regions and fitting a polynomial function within each region. The piecewise polynomial regression involves fitting different low-degree polynomials over different regions of X .This polynomials are constrained and thus they join smoothly at the knots. More knots can lead to more flexible piecewise polynomial. Smoothing splines is natural cubic spline with knots at every unique value of x_i, it minimizes residual sum of squares with a smoothness penalty. Local regression, similar to splines, allows regions to overlap and fits flexible non-linear functions with an approach where the fit is been computed at a target point x_0 using only the nearby training observations. Generalized additive models are an extension of the standard linear model by allowing non-linear functions of each of the variable, while maintaining additivity.

Chapter 8: Tree-Based Methods

The tree based methods are used for regession as well as classification problems. This methods involves stratifying or segmenting the predictor space into a number of simple regions, in order to make a prediction for a given observations. The mean or mode response value is used for the training observations in the region it belongs. As the set of splitting rules used to segment the predictor space is summarized in a tree, this approaches are known as decision tree methods. This methods are simple and useful for interpretation. The ensemble methods such as bagging, boosting, random forests and Bayesian additive regression trees are introduced in this chapter. These methods produces multiple trees which are then combined to yield a single consensus prediction.

Growing a regression tree consist of a problem of overfitting and poor test set performance due to the complexity of resulting tree. Tree pruning technique reduces the size of the tree by removing the branches or sub-trees that do not contribute significantly to the overall prediction accuracy of the model. The goal is to select a subtree that leads to lowest test error rate. Growing a classification tree involves predicting class of each observation based on the majority class of training observations in its region. Alternatives to residual sum of squares (RSS), such as Gini index or entropy, are used to evaluate split quality. Gini index and entropy measure node purity while the classification error rate measures the fraction of incorrect class assignments. Gini index or entropy are preferred for growing the tree, while the classification error rate is preferred for pruning.

Bootstrap Aggregation (bagging) is a variance reduction technique for statistical learning methods that involves creating multiple copies of the original training data set using bootstrapping, fitting a separate decision tree to each copy, and then combining the trees to form a single predictive model, where each tree is built independently on a bootstrapped data set. Random forests provides an improvenement over bagged trees by way of a small tweak that decorrelates the trees. Boosting grows trees sequentially using information from previously grown trees and doesn't involve bootstrap sampling, instead each tree is fit on a modified version of the original data set. In Bayesian Additive Regression Trees (BART), only the original data is used and the trees are grown successively. However, each tree is perturbed to avoid local minima and allow for a more comprehensive exploration of the model space.

Chapter 10: Deep Learning

Deep learning has become a significant area of interest in the fields of machine learning and artificial intelligence. The key component of deep learning is the neural network. In this chapter, the fundamental concepts of neural networks and deep learning, along with their various applications are discussed. These applications include specialized neural networks like convolutional neural networks (CNNs) for image classification and recurrent neural networks (RNNs) for sequences and time series. A neural network takes an input vector of p variables X = (X1, X2, . . . , Xp) and builds a nonlinear function f(X) to predict the response Y. A simple feed-forward neural network is used to model a quantitative response.

Modern neural networks often have multiple hidden layers, each containing many units. A single hidden layer with a large number of units can approximate most functions. However, it is easier to discover a suitable solution with multiple layers, each having moderate size. If the response is qualitative, we seek coefficient estimates that minimize the negative multinomial log-likelihood or cross-entropy. On the other hand, if the response is quantitative, we minimize the squared-error loss. Convolutional neural networks utilize convolution and pooling layers to create a hierarchy for modeling image inputs.

Many data sources, such as financial time series, recorded speech, and handwritten digits, have a sequential nature and require special consideration when building predictive models. RNNs are designed to take advantage of the sequential structure of these inputs, much like CNNs use the spatial structure of image inputs. The output of an RNN can also be a sequence or a scalar. The more elaborate versions of RNNs are long-term and short-term memory (LSTM), which require a long time to train, making architecture exploration and parameter optimization a tedious process. Although neural networks with huge number of parameters can provide good results with zero training error, it is important to remember the bias-variance trade-off, which always holds.

Empirical Analysis

Ridge Regression and Lasso

Ridge regression and Lasso are statistical techniques that shrink the regression coefficients towards zero, which can significantly reduce their variance. Ridge regression is similar to ordinary least squares (OLS) but estimates the coefficients by minimizing a slightly different quantity. The Ridge regression coefficients estimates $\hat{\beta}^R$ are the values that minimize:

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

The ridge regression seeks coefficient estimates that fit data by making RSS small. The shrinkage penalty : $\lambda\sum_j\beta_j^2$ is small when $\beta_j, \ldots, \beta_p$ are close to zero and so the shrinking effects leads β_j towards zero. This shrinkage penalty is not applied on the intercept β_0, as it is simply

a measure of the mean value of the response. Here $\lambda \geq 0$ is a tuning parameter. It serves to control the relative impact of RSS and Shrinkage penalty on the regression coefficient estimates. Selecting a good value of λ is essential, as ridge regression produces different set of coefficient estimates, $\hat{\beta}_{\lambda}^{R}$ for each value of λ. As λ inceares, the flexibility of Ridge regression fit decreases, leading to decrease variance but increased bias. Ridge regression has bias-variance trade-off advantage over OLS.

In situations where the relationship between the response and predictors is almost linear, the least squares estimates may have high variance and low bias. When the number of variables is almost as large as the number of observations or when the number of variables is greater than the number of observations, the least squares estimates will be extremely variable or even have no unique solution. Ridge regression can perform well in these situations by trading off a small increase in bias for a large decrease in variance. Ridge regression is particularly effective when the least squares estimate have high variance. Furthermore, ridge regression has computational advantages over best subset selection as it requires searching through fewer models

Lasso overcomes the model interpretation problem of ridge regression when the number of predictors, denoted as p, is large. Ridge regression shrinks the coefficients towards zero but never sets them exactly equal to zero, resulting in a model that involves all predictors. Increasing the value of λ will tend to reduce the magnitudes of the coefficients but will not result in exclusion of any of the variables. The lasso coefficients, $\hat{\beta}_{\lambda}^{L}$, minimize the quantity :

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

The lasso uses an ℓ_1(pronounced “ell 1”) penalty instead of an ℓ_2 penalty. The ℓ_1 norm of a coefficient vector β is given by $\|\beta\|_1 = \Sigma|\beta_j|$. In contrast to ridge regression, lasso shrinks the coefficients towards zero and can force some of them to be exactly equal to zero when the tuning parameter λ is sufficiently large. Lasso performs variable selection, due to which model generated from lasso are much easier to interpret than produced by ridge regression.

Application on Song Dataset

In this example, OLS, Ridge, and Lasso regression is applied to song popularity dataset (Song Popularity Dataset, n.d.). Using this dataset, response variable song popularity is predicted dependent on predictors such as duration (ms), acousticness, danceability, energy, instrumentalness, key, liveness, loudness, audio mode, speechiness, tempo, time signature and valence. The response variable popularity has values between 0 and 100. For building regression models, R software will be used. The glmnet package is used to fit ridge and lasso regression models.

To construct, validate and compare all the models, the data is divided into training and test set. The multiple linear regression model is fitted on the training set. Using the predict() function in R programming, the model fit is predicted with respect to test set. MSE (Mean Square Error) is obtained to check the accuracy of the model fit. It is calculated by taking the average of the squared differences between the predicted and actual values of the response variable for each

observation in the dataset. The test MSE of multiple linear regression model is 468.4044. This means that on average, the squared difference between the predicted values and the actual values of the response variable is 468.4044.

Now, ridge regression and lasso are fitted using glment() function in R software. The glmnet() function has alpha argument that determines what type of model is fit, if alpha =0 then a ridge regression model is fit and if alpha =1 then a lasso model is fit. This function also by default standardizes the variables so that they are on the same scale. Before fitting the model, grid value is set using the seq() function in R, such that it starts at 0.01, ends at 20, and has a step size of 0.01. This sequence is used to generate the value of λ. Instead of arbitrarily choosing λ , it would be better to use cross-validation to choose the tuning parameter. This is done by built-in cross-validation function, cv.glment().

Ridge regression is fitted on the dataset. The lambda value that results from cross-validation and has the smallest cross- validation error is 0.07. The ridge regression model is refitted on the full dataset using the value of λ, choosen by the cross-validation. The resulted test MSE is 468.379. According to the MSE, we can state that ridge regression with a wise choice of λ had outperform multiple linear regression model.

Now, we are interested in examining whether lasso can yield either a more accurate or interpretable model than ridge regression. To fit Lasso, we use glment() function, with the argument alpha=1. The rest of the process is similar as ridge regression. The lambda value after performing cross-validation is 0.05. The test MSE of Lasso is 468.3696.

Overall MSE Results		
MSE. lm	MSE. ridge	MSE. lasso
468.4044	468.379	468.3696

Table 1: MSE results of lm, ridge, and lasso regression.

According to the MSE of each model, lasso performs slightly better than multiple linear regression and ridge regression. However, the difference in MSE between these models is relatively small and may not be statistically significant. The choice between these models may depend on other factors such as the interpretability of the coefficients and the computational complexity of the algorithms used to fit the models.

Decision Tree

Tree-based methods can be applied to both regression and classification problems. For this paper, we will only focus on classification tree. The classification trees predict a qualitative response by assigning each observation to the most commonly occurring class of training observations in its corresponding region. In addition to predicting class labels, we are often interested in the class proportions among the training observations that fall into a terminal node region.

The process of building a classification tree is similar to regression tree. The first step is to use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations. This large tree is then pruned using cost complexity pruning to obtain a sequence of best subtrees, as a function of α. The next step is to use K-fold cross-validation to choose α, where the training observations are divided into K folds, and the mean squared prediction error is evaluated on the left-out fold for each value of α. The average of the results for each value of α is then calculated, and the α value that minimizes the average error is chosen. Finally, the subtree from Step 2 that corresponds to the chosen value of α is returned. This algorithm ensures that the resulting regression tree is not overfit to the training data and has the best predictive accuracy on new data.

In classification setting, the classification error rate is considered, which is the fraction of training observations in a region that do not belong to the most common class, is a natural alternative to the residual sum of squares (RSS) for evaluating binary splits. However, the classification error rate is not sensitive enough for tree-growing, and alternative measures such as the Gini index and entropy are typically preferred. Both measures evaluate the quality of a split by measuring node impurity, which refers to the presence of multiple classes in a node. A low Gini index or entropy value indicates that a node contains mostly observations from a single class. These measures are quite similar numerically, so either one can be used when building a classification tree.

When pruning the tree, any of the three measures (classification error rate, Gini index, entropy) can be used, but the classification error rate is preferable if the goal is to optimize prediction accuracy of the final pruned tree. Overall, classification trees are simple and interpretable models that can be used for predicting qualitative responses.

There are ensemble methods to obtain a single and potentially powerful model. The method relevant for this paper is Bagging. It is a general technique used to decrease the variance of statistical learning methods. It involves creating B different bootstrapped training datasets by repeatedly sampling from a single training dataset. In this approach, model is trained on the bth bootstrapped training set in order to obtain $\hat{f}^{*b}(x)$ and average all predictions to obtain bagging:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

In the case of classification trees, bagging involves constructing multiple trees on bootstrap samples of the data, and then aggregating the predicted classes of the trees by majority vote. This can help to reduce the variance of the models and improve their predictive accuracy, especially if the trees are prone to overfitting.

Application on Pokémon Dataset

In this example, Pokémon dataset is utilized, which is created by Alberto Barradas (Pokémon with Stats, n.d.). This dataset includes information's of 800 Pokémon's, their number, name, first and second type, as well as basic statistics includes : HP (Hit point), Attack, Defense, SPAtk (Special Attack), Defense, SPDef (Special Attack Défense), Speed, and Generation. The variable of interest in this analysis is whether or not a given Pokémon is Legendary, with a

value of True indicating it is Legendary, False indicating otherwise. The goal is to classify that Pokémon is Legendary or not based on 6 other variables.

Firstly, the required R programming libraries for conducting this analysis are loaded which are tree, randomForest and stats libraries. Then the dataset was loaded using read.csv() function. The Legendary variable was Boolean variable, so it was converted to a factor variable using a factor() function in R, and the values set as NL (Non-Legendary) and L (Legendary). This is an important step before starting the data analysis task because it makes output more interpretable, facilitate statistical analysis, and handle missing values.

The dataset was divided into training and test set using the sample() function. The test set includes 200 observations and training set includes 600 observations. Using the tree() function, we fitted a tree where Legendary as response and variables HP, Attach, Defense, SpAtk , SpDef, and Speed as predictors on training dataset. According to the summary of tree, the training error rate is 3.33% and the number of terminal nodes is 15. The residual mean deviance is 0.1386, which means that, on average, there is a difference of 0.1386 between the predicted values and the actual values of the response variable in the model. A smaller residual mean deviance indicates a better fit of the model to the data.

The prediction is done using predict () function on the test data, and confusion matrix comparing test labels to predicted test labels are achieved. This matrix is represented in below table 2. The MSE value of this tree fit is 0.09 and the model accuracy is 91%.

	Poki.Tree		**Pruned. Tree**		**Bagging**	
	Test		Test		Test	
Predict	L	NL	L	NL	L	NL
L	9	12	9	6	11	7
NL	6	173	6	179	4	178

Table 2: Confusion matrix of all tree models fitted.

Now, the focus is to check whether or not pruning the tree will result better accuracy then the current approach. For this, cv.tree() function is used on the training set to determine optimal tree size. According to the results visible in the plot, the tree with 15, 8, and 5 terminal nodes results into the lowest cross-validation error rate. Thus, a pruned tree with 5 terminal nodes is produced.

The pruned tree accuracy is 94% which is higher than the original tree. Even the MSE is smaller than original tree. The pruned tree is easier to read as well as the terminal nodes are far lesser than original tree. Bagging is applied to the dataset to further evaluate the possibility of increasing the fit accuracy of tree. The bagging tree is produce using randomForest() function. Its accuracy is 94.5 %, which is similar to pruned tree. The MSE of the bagging is 0.05 which is smallest of all fitted models. To further compare our results, logistic regression is applied to the dataset.

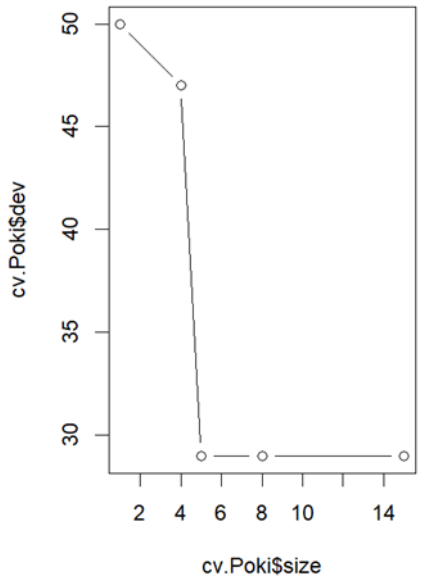

Figure 1: Error rate function of size

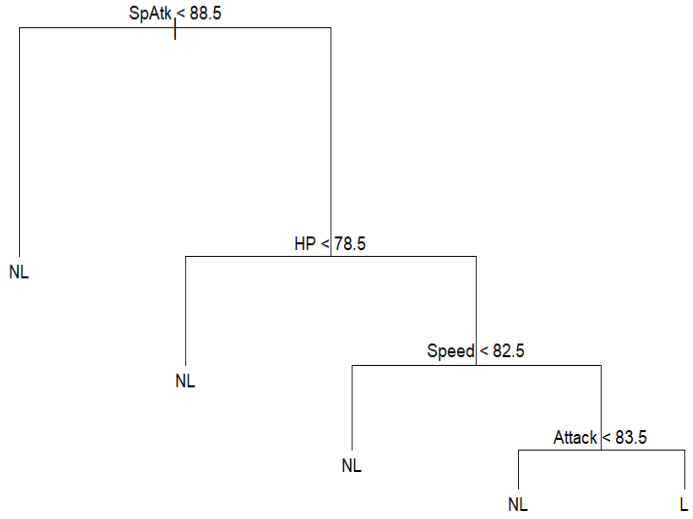

Figure 2: Prune tree of 5 node

Logistic Regression model is fitted using glm() function. As the outcome variable is binary provides value 0 or 1, logistic regression is fitted rather than multiple linear regression. The AIC is 156.03. The accuracy of this model 96.5%, which is higher than all the tree approaches. It can be stated that logistic regression performs better compared to decision trees in this example.

Conclusion

This paper focused on application of different regression and decision tree techniques on two datasets. In the first example of song popularity dataset, we applied three regression models, namely multiple linear regression, ridge regression, and lasso regression. The aim was to predict song popularity based on various predictors. The dataset was divided into training and test sets, and the models were fitted and evaluated based on their respective test MSEs. The optimal value of the tuning parameter λ was determined through cross-validation for both ridge and lasso models. We found that the ridge regression model performed slightly better than multiple linear regression model, and lasso regression performed even better than ridge regression based on the test MSEs. However, the difference in MSE between these models is relatively small and may not be statistically significant to state which model is better and may depend on other factors.

The other example utilized the Pokémon dataset to classify whether a given Pokémon is Legendary or not based on six other variables. The tree algorithm was initially used, resulting in a training error rate of 3.33% and accuracy of 91% on the test set. The tree was then pruned, resulting with 5 terminal nodes that will lead to achieve model accuracy of 94%, higher than the original tree. Bagging was applied to the dataset, which produces a tree with accuracy of 94.5%. Logistic regression was also fitted to the dataset, attaining an accuracy of 96.5%, indicating that logistic regression performs better compared to decision trees in this example.

References

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning (Vol. 103). Springer. https://doi.org/10.1007/978-1-4614-7138-7
- Pokemon with stats. (n.d.). Retrieved 14 February 2023, from https://www.kaggle.com/datasets/abcsds/pokemon
- Song Popularity Dataset. (n.d.). Retrieved 19 February 2023, from https://www.kaggle.com/datasets/yasserh/song-popularity-dataset