MIDDLE EAST TECHNICAL UNIVERSITY

IAM 557 Statistical Learning and Simulation

Project Report

Linear Regression, Regression tress, and Classification on Diabetes Dataset

Nika Rasoolzadeh

January 24, 2020

Contents

1 Introduction

In this project using scikit-learn toy dataset diabetes a quantitative measure of disease progression one year after baseline is predicted, with the ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements given. The dataset has been standardized to have mean 0 and squared length of one[1]. Several regression methods such as least squares regression, best subset selection, and Lasso regression are used. Their performance over the test set is measured and compared. Also, regression trees, bagging and random forests are used which shown a slightly better performance. Finally, a classification problem is defined by using the `"sex"` variable as the response, and all other variables as predictors. Then multiple methods and their performance are evaluated.

2 Linear Regression

The diabetes dataset contains x a matrix with 10 columns , y a numeric vector, and x_2 a matrix with 64 columns corresponding to main effects and second-order interactions[3]. Therefore, the dataframe of x and y is used in this project which is the observation data of 442 patients.

A train and a test set is obtained by a 80%/20% random split in the dataframe. There are $n = 354$ examples in the train set and $n = 88$ in the test set. The original data provided by LARS is initially standardized so no further modification on the data is done in this project. Now the data is ready we can move on to using it.

First method of use is least squares regression with `lm()` and all predictors. The coefficients found for the model is as $Intercept = 152.394, age = -5.716, sex = -212.757, bmi = 539.809, map = 277.269, tc = -1051.532, ldl = 671.635, hdl = 266.227, tch = 265.966, ltg = 885.327, glu = 30.030$. The model plot is as below:

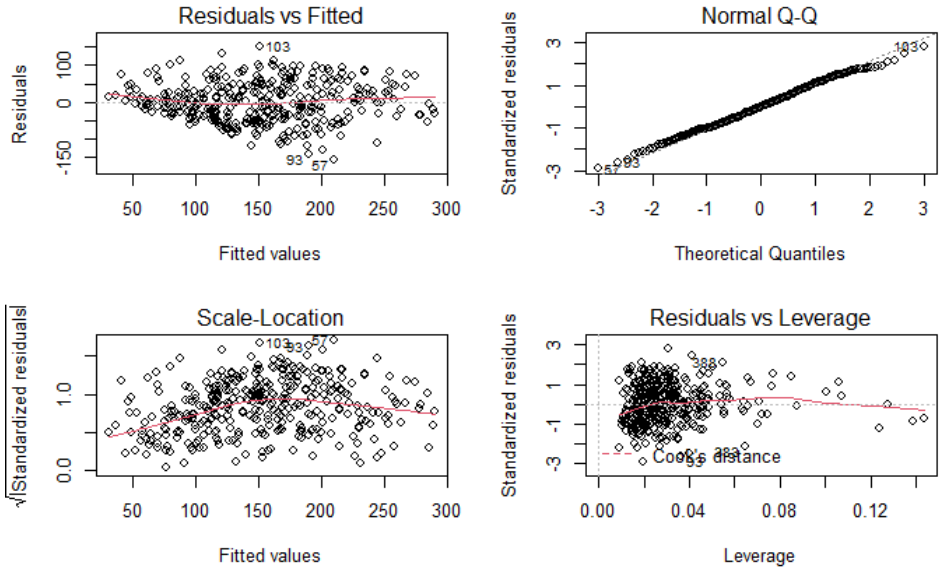

Figure 2.1: Linear regression model plots

The model above results in a 2688.096 Mean Square Error on the test data, and a Standard Error of 425.566 which are both quite large values. So to find a better performing model Best Subset Selection with both forward and backward selection will be used.

3 Best Subset Selection

Using `regsubsets()` Forward Selection method is applied. We see that C_p and adjusted R^2 values suggest 8 predictor models as the best ones, whereas BIC value suggest of a 6 predictor model.

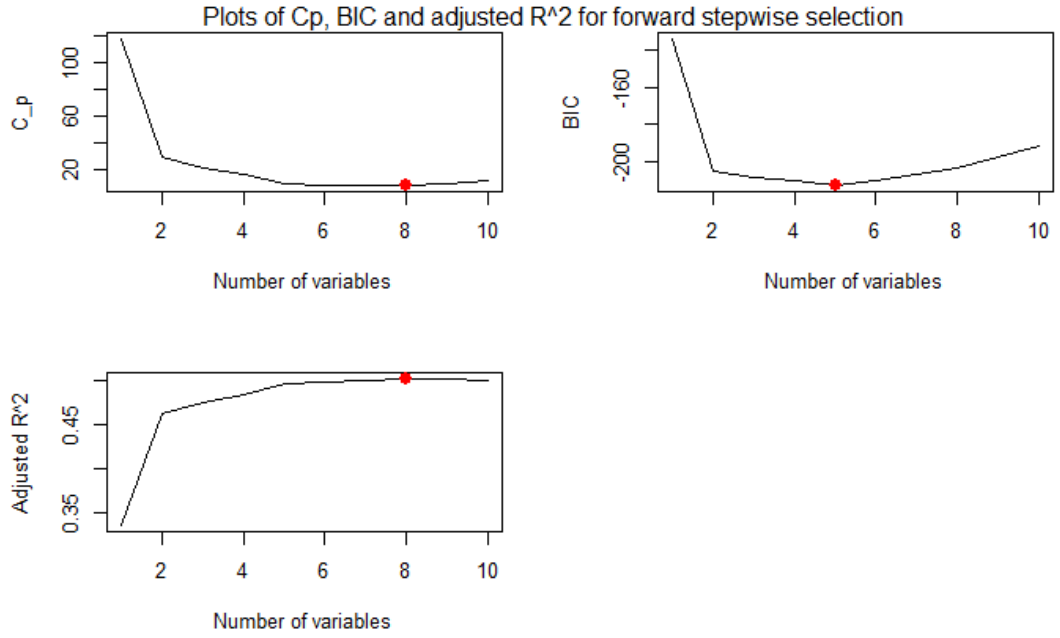

Figure 3.1: C_p, BIC, and adjusted R^2 values using Forward Selection method

To compare the results, a model with 6 predictors and a model with 8 predictors are tested on the test set. The 6 parameter model performs better than the larger one.

```
"6 predictor model test MSE: 2625.042, SE:419.803
 8 predictor model test MSE:2707.253, SE:428.927"
```

The same way Backward Selection is applied and this time C_p, BIC values suggest the choice of a 6 predictor model while adjusted R^2 value still favors a 8 predictor model.

The results of the both models on the test set shows similar behavior as the results found using forward stepwise. The 6 predictor model behavior found by backward method is slightly better over the test set.

```
"6 predictor model test MSE: 2663.205, SE:421.864
 8 predictor model test MSE:2707.253, SE:428.927"
```

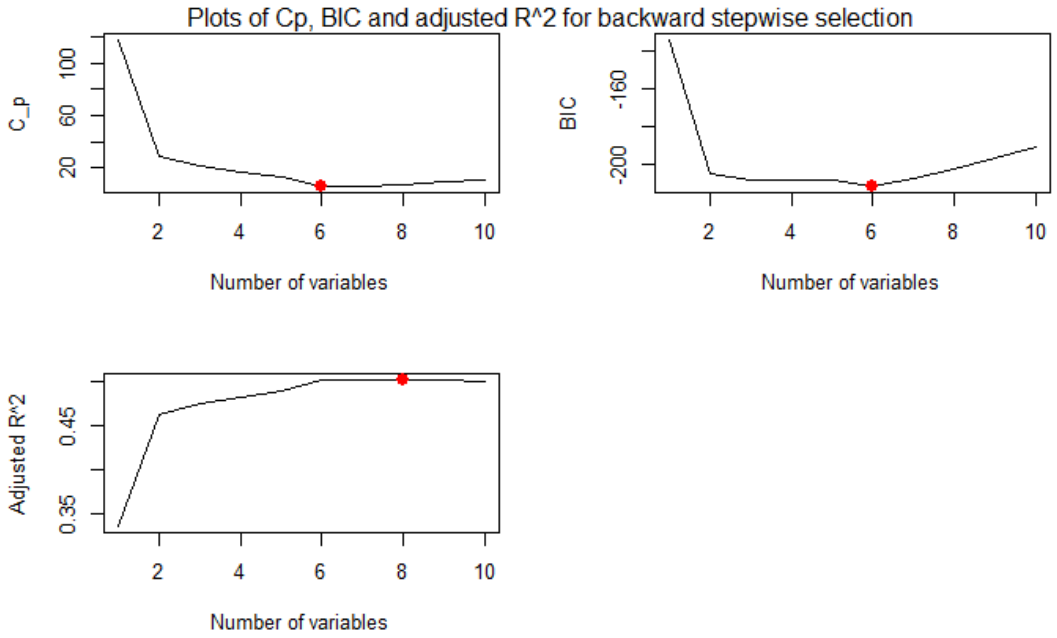

Figure 3.2: C_p, BIC, and adjusted R^2 values using Backward Selection method

4 Lasso Regression

Finally Lasso Regression is performed. Using cross validation from `cv.glmnet()` with default fold number as 10 we find the minimum lambda to fit the model accordingly. The optimal value is found to be $\lambda = 0.01819$ or $\log(\lambda) = -4.00698$.

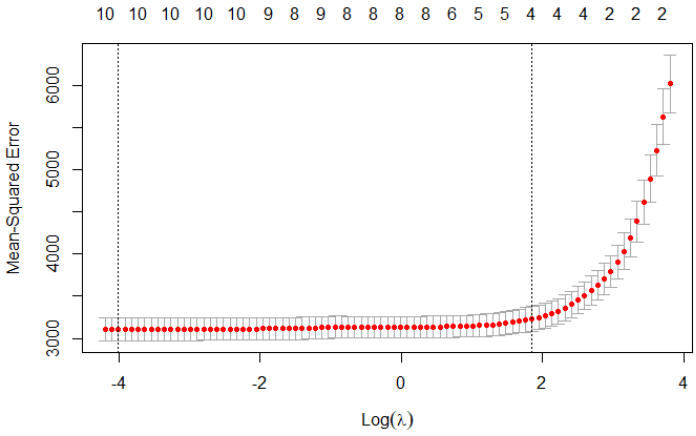

Figure 4.1: Mean Square vs Lambda plot after applying cross validation

The lasso model then is fitted with respect to this value, and used in the prediction over the test set. Lasso regression suggest to use all the predictors with the coefficients as $Intercept = 152.382, age = -4.654, sex = -211.683, bmi = 540.409, map = 276.157, tc = -966.494, ldl = 606.393, hdl = 225.918, tch = 251.420, ltg = 854.269, glu = 29.893$. The result is as `"Lasso Regression test MSE: 2679.394, SE:424.304, MAE:40.916"`

Model	Test.MSE	test.SE
<chr>	<dbl>	<dbl>
Linear Regression Model	2688.096	425.5661
Forward Selection Model with 6 predictor	2625.042	419.8030
Forward Selection Model with 8 predictor	2707.253	428.9273
Backward Selection Model with 6 predictor	2663.205	421.8643
Backward Selection Model with 8 predictor	2707.253	428.9273
Lasso Model with 10 fold CV	2679.394	424.3035

6 rows

Figure 4.2: Model Selection Comparison Results

For the given training and test data set the models and the error values found are not quite perfect. However, the forward selection model with 6 predictor performs the best among all. Overall, a larger dataset may improve these results.

5 Tree Based Algorithms

The regression problem can also be solved by Tree based method. Using library `tree` a model is trained with the train set. The variables used in the construction of the tree are ltg, bmi, tch, map, and glu. The deviance or sum of squared errors for the tree is 2693.

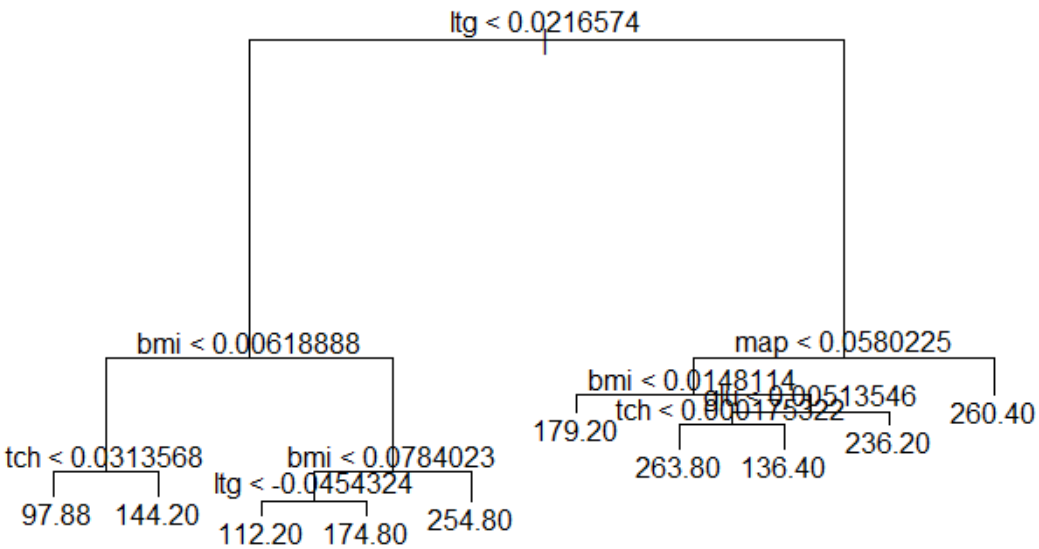

Figure 5.1: Tree plot of the model

Making predictions for the test set with this tree delivers a mean squared error of 3808.152 which is quite large. It is even larger than the linear regression method's error.

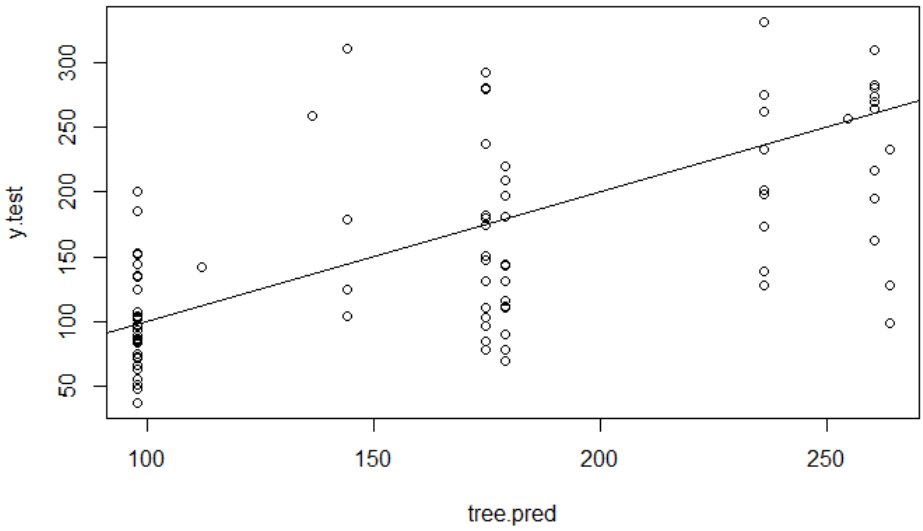

Figure 5.2: prediction vs real values over test set

In order to see whether pruning the tree improves the performance of the model, `cv.tree()` is used to select the most complex tree by cross validation. The results shows a subtree of size 6 gives the smallest deviance.

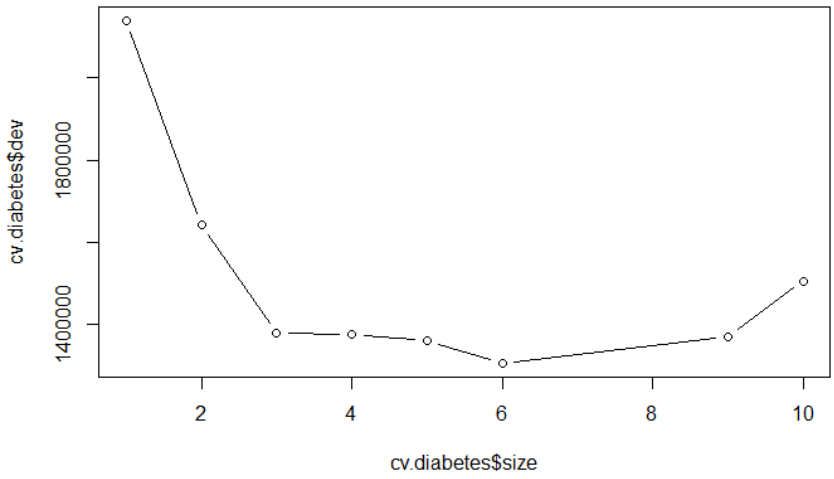

Figure 5.3: Cross validation plot of the tree

After pruning the tree will be as follows and the mean squared error of the test set will decrease to 3633.415.

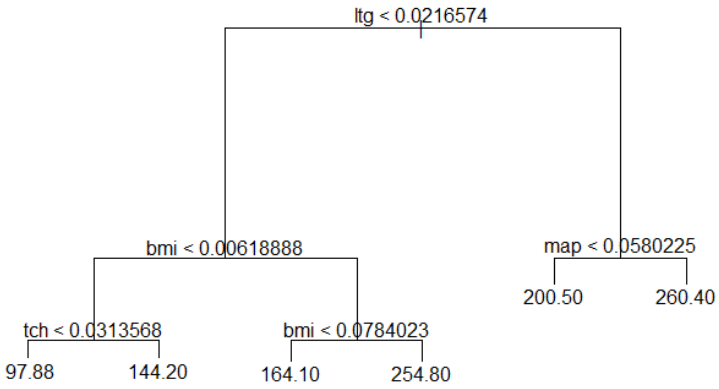

Figure 5.4: Pruned tree plot

6 Bagging and Random Forests

Using `randomForest` package in R we can perform bagging and random forests on our data. With `mtry = 10` the bagging will be done with all predictors considered in each tree split. A mean squared error of 3339.051 is obtained. The model would result in a test mean squared of 3165.734 which is by far the smallest MSE we got.

In the same way we can grow a random forest with a smaller `mtry` value. Here, for regression we use `mtry = 3` which results in a mean squared error of 3023.855 for the test set. Since it is smaller than the previous MSE we can say that random forest performed better than bagging. We can examine the importance of each variable by using importance function and plotting it. We can see that across all trees in the random forest `ltg` and `bmi` predictors are the two most important.

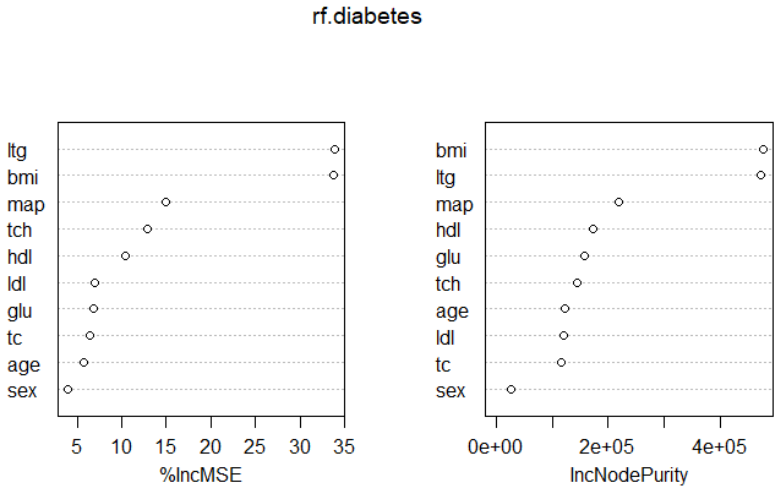

Figure 6.1: Random Forest Importance Plot

7 Classification

In this part `sex` is used as the response and the variable that is to predicted while other 10 variables are predictors. Since the data is numeric and standardized first we should categorize them meaning modifying the `sex` column values to dummy values $0, 1$. In the original paper $1, 2$ values are used and the corresponding gender is not mentioned. So, here we assume that 0 corresponds to male and 1 corresponds to female patients.

Using logistic regression a model is fit to the train data. Prediction over the test set with this model is 0.75% accurate.

```
Confusion Matrix and Statistics


test.classes  0  1
           0 34  7
           1 15 32

               Accuracy : 0.75
                 95% CI : (0.6463, 0.8362)
    No Information Rate : 0.5568
    P-Value [Acc > NIR] : 0.0001392

                  Kappa : 0.5038

 Mcnemar's Test P-Value : 0.1355930

            Sensitivity : 0.6939
            Specificity : 0.8205
         Pos Pred Value : 0.8293
         Neg Pred Value : 0.6809
             Prevalence : 0.5568
         Detection Rate : 0.3864
   Detection Prevalence : 0.4659
      Balanced Accuracy : 0.7572

       'Positive' Class : 0
```

Figure 7.1: Confusion Matrix of the prediction on test set

And the estimated coefficients are as $Intercept = 1.240, age = 1.239, bmi = -2.546, map = 14.604, tc = -26.036, ldl = 20.086, hdl = -8.749, tch = 8.592, ltg = 6.517, glu = 4.798, y = -0.009$.

Using K Nearest Neighborhood we train a model with $k = 4$ and make predictions. However, the performance is not improved and the accuracy over test set is dropped to 53%. And as k increases or decreases over 4 the accuracy seems to drop for both training set and test set.

Using validation set approach, with a bigger validation train set and a smaller test set of size 13, 84% of accuracy is achieved.

```
Confusion Matrix and Statistics


classes 0 1
      0 9 2
      1 0 2

               Accuracy : 0.8462
                 95% CI : (0.5455, 0.9808)
    No Information Rate : 0.6923
    P-Value [Acc > NIR] : 0.1862

                  Kappa : 0.5806

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 1.0000
            Specificity : 0.5000
         Pos Pred Value : 0.8182
         Neg Pred Value : 1.0000
             Prevalence : 0.6923
         Detection Rate : 0.6923
   Detection Prevalence : 0.8462
      Balanced Accuracy : 0.7500

       'Positive' Class : 0
```

Figure 7.2: Confusion Matrix of the prediction on validation test set

Finally, Quadratic Discriminant Analysis is implemented which proved to perform better than the above methods on the given test set.

```
Confusion Matrix and Statistics

          Survived
Predicted  0  1
        0 31  7
        1 18 32

               Accuracy : 0.7159
                 95% CI : (0.6098, 0.807)
    No Information Rate : 0.5568
    P-Value [Acc > NIR] : 0.001585

                  Kappa : 0.4405

 Mcnemar's Test P-Value : 0.045500

            Sensitivity : 0.6327
            Specificity : 0.8205
         Pos Pred Value : 0.8158
         Neg Pred Value : 0.6400
             Prevalence : 0.5568
         Detection Rate : 0.3523
   Detection Prevalence : 0.4318
      Balanced Accuracy : 0.7266

       'Positive' Class : 0
```

Figure 7.3: Confusion Matrix of the prediction on test set using QDA

8 Conclusion

There are various methods to solve both a regression and a classification problem. In this projects some of these approaches were implemented and evaluated for the given diabetes dataset. Though the performance of some models were not as perfect as one might desire, the general concept of each method and their implementation in R is comprehensive through this project. First, some regression methods were examined from which the linear regression method performed best, then tree based algorithms were implemented. The random forest method gave the smallest test MSE for the regression problem of this project. For the classification problem, logistic regression, KNN, validation set approach, and Quadratic Discriminant Analysis were utilized. From which validation set approach and QDA performed best with 84%, and 72% accuracies respectively. To sum up, the model selection and its training depend on dataset size, model size, significant variables, the predictor correlations, etc. And one should implement and use the model according to these factors inline with the desired output, and cost.

9 References

References

[1] Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani(2004) "Least Angle Regression," Annals of Statistics (with discussion), 407-499.

[2] https://www4.stat.ncsu.edu/ boos/var.select/diabetes.html

[3] https://rdrr.io/cran/biglars/man/diabetes.html