lab2

2022-11-22

Exercise 1
This is the underlying probabilistic model:

$$Fat = \theta_1 * Channel1 + \theta_2 * Channel2 + \theta_3 * Channel3 + \ldots + \theta_{100} * Channel100 + \epsilon$$

ϵ = Noise term.

```
## [1] "Test error"
```

```
## [1] 722.4294
```

```
## [1] "Train error"
```

```
## [1] 0.005709117
```

Comment on the quality of fit and prediction and therefore on the quality of model

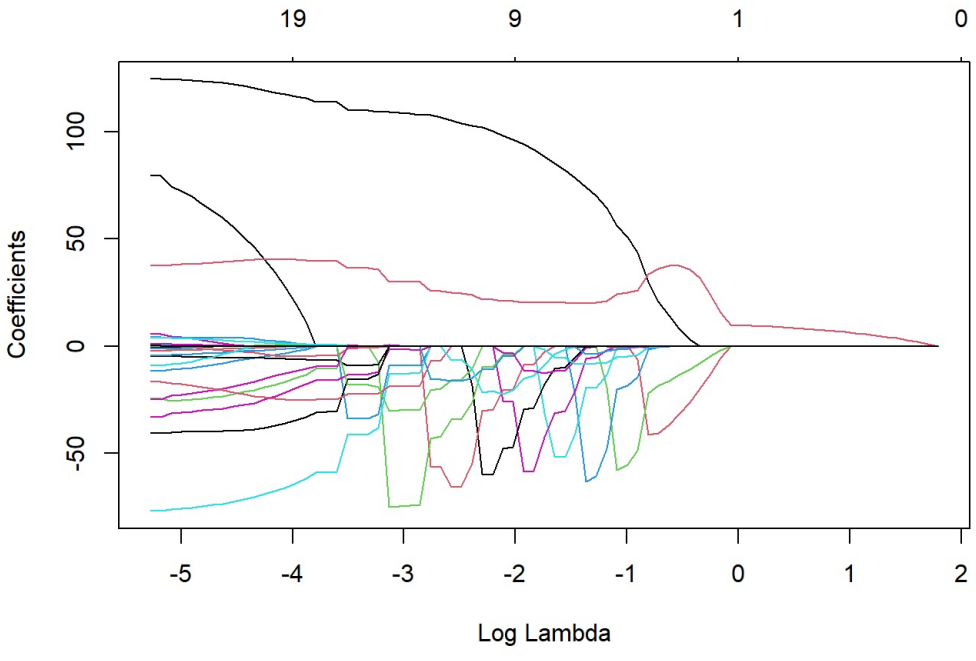
If we calculate the mean square error of the training prediction vs the test prediction we see that we get extremely high error from test compare to training. This means that the model is very specific for the data that it is trained on. Which means that the quality of the model is not great for predicting other than particularity the training data. Overfitting.

Exercise 2 L1 (also called lasso) regression model

L1 Cost function looks like this from the course elitterature:

Exercise 2
This is the cost function for L1 - regularization. The extra penalty term is $\lambda||\theta||_1$ for lasso regression.

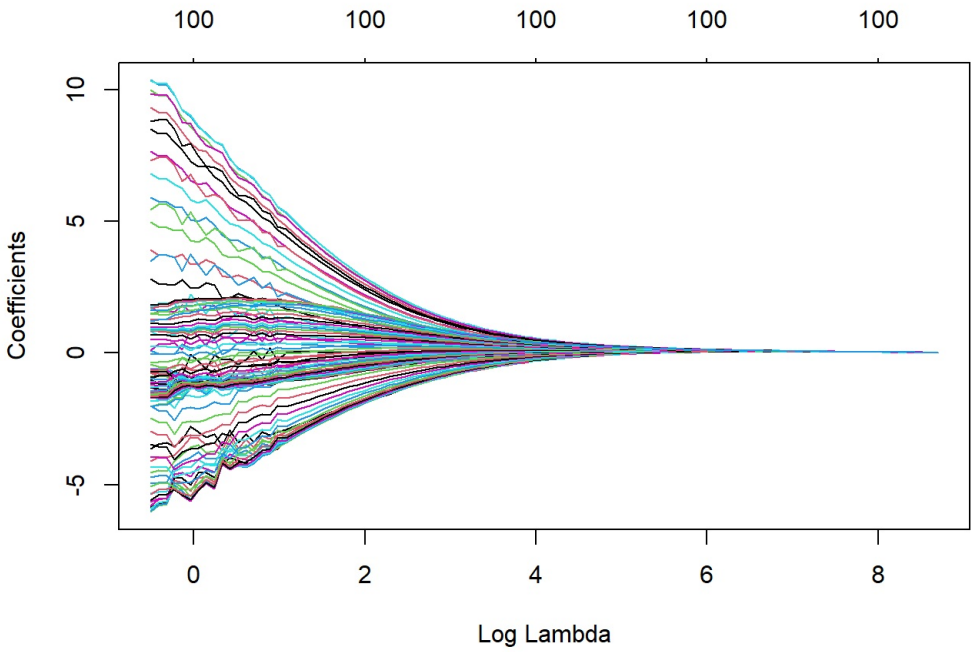
$$\theta_{hat} = arg\,min_\theta 1/n * ||X\theta - y||_2^2 + \lambda||\theta||_1$$



What value of the penalty factor can be chosen if we want to select model with only three features?

From my interpretation, to select a model with 3 feature you need to pick a log-lambda value of -0.5.

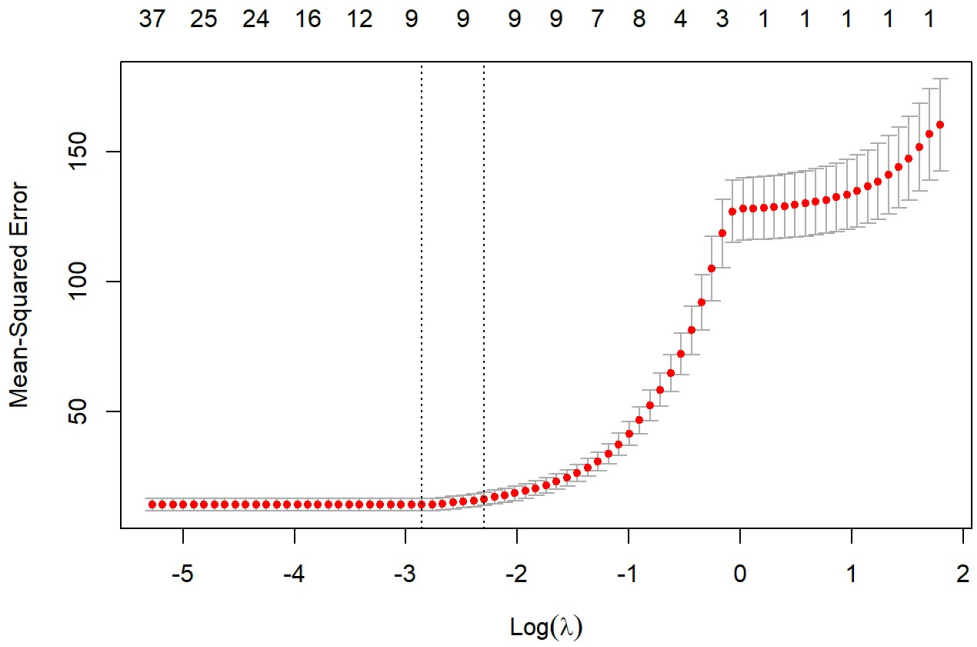
Exercise 4

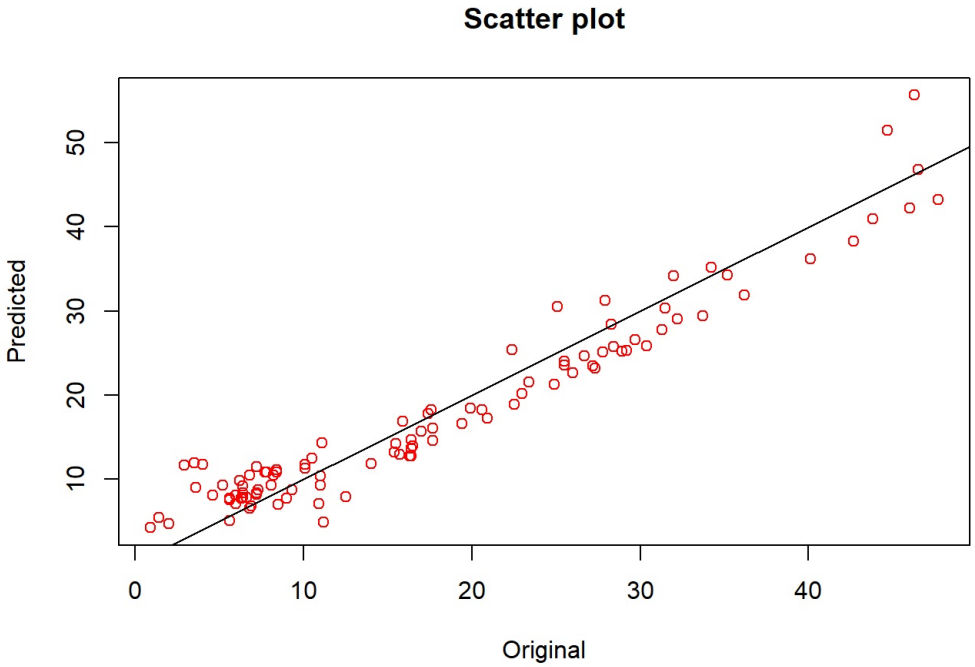


In ridge regression plot we can clearly see that the coefficients are penalized in a way that makes the least effective in the estimation shrink faster. In the lasso regression plot we cannot make any conclusions about the coefficients in the same way. However, we can draw conclusions about the features being used depending on the log-lambda values.

In ridge regression as lambda increases all coefficients go down to zero but never become zero. In lasso you can see that the coefficients are set to zero as lambda increases. This is the main difference between ridge and lasso regression.

Exercise 5





The optimal lambda is 0.0574. Derrived from the model.

The CV score increases as lambda increases, where around 0 the increase slows down.

We interpret that the lambda_min is the optimal lambda and uses a model of 9 features.

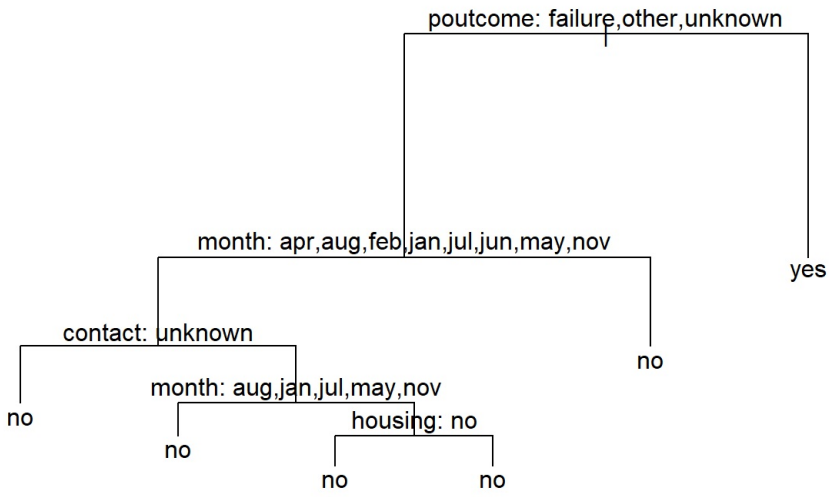
It would seem that they have the same amount of CV-score but the optimal lambda uses less model features.Therefor it would not generate significantly better predictions then lambda = -4.

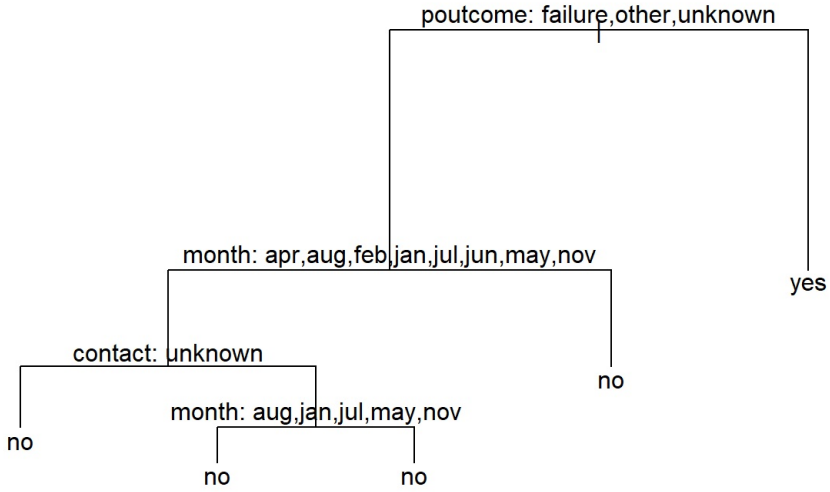
Using the scatter plot we conclude that the predictions are good but not great. There seems to be a few data points where the deviation from the line is larger compared to others such as in the beginning and at the end. However, overall it does a fairly good job at predicting close to the line.

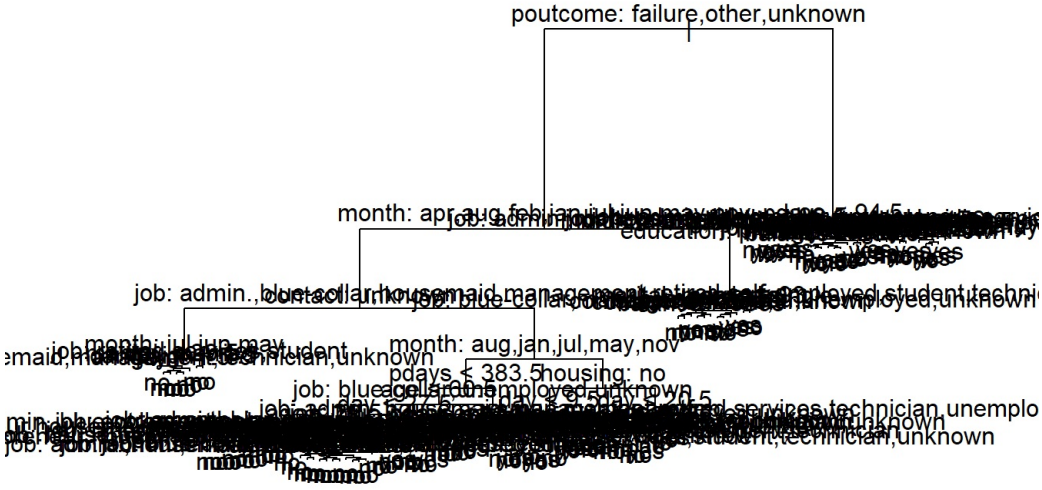
Task 2.

Exercise 1

Exercise 2


poutcome: failure,other,unknown
month: apr,aug,feb,jan,jul,jun,may,nov
yes
contact: unknown
no
month: aug,jan,jul,may,nov
no
housing: no
no
no
no


poutcome: failure,other,unknown
month: apr,aug,feb,jan,jul,jun,may,nov
yes
contact: unknown
no
month: aug,jan,jul,may,nov
no
no
no


poutcome: failure,other,unknown
month: aug,jan,jul,may,nov
pdays < 383.5
housing: no

Misclassification rate

```
## [1] "1) Training and validation"
```

```
## [1] 0.1048441
```

```
## [1] 0.1092679
```

```
## [1] "2) Training and validation"
```

```
## [1] 0.1048441
```

```
## [1] 0.1092679
```

```
## [1] "3) Training and validation"
```

```
## [1] 0.09400575
```

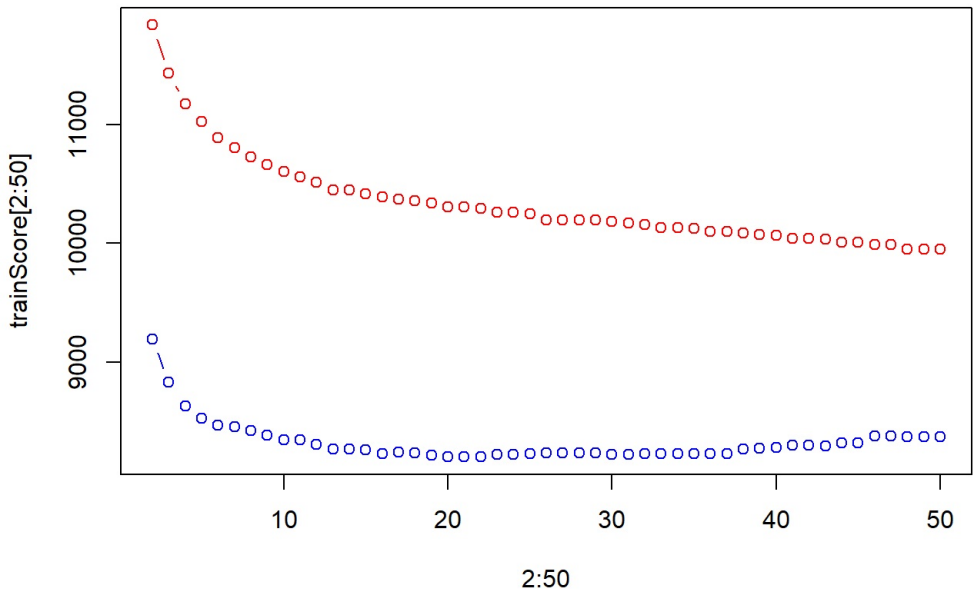
```
## [1] 0.1119221
```

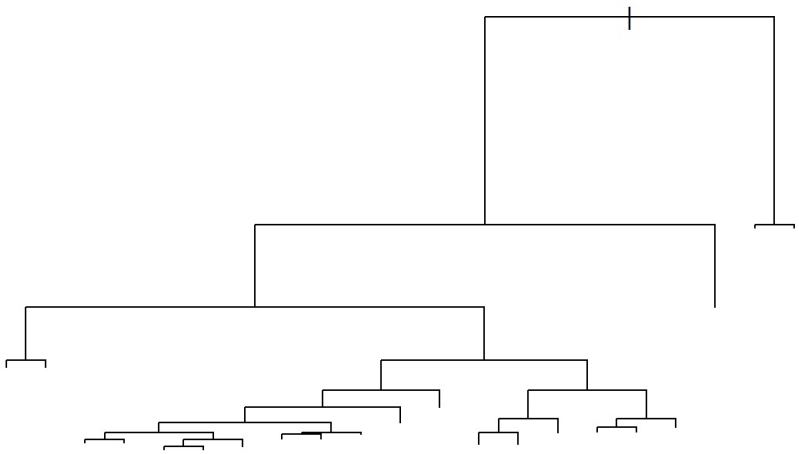
Complementary answer

The best model is not C. It is A and B because of the validation error they generate. Both of their validation error is lower than C. C gets higher error due to the fact of it being overfitted.

Changing deviance resulted in a much larger tree, probably because more values were allowed to be included even though the deviance was low (this is my guess). Setting minsize to 7000 made the tree smaller, as it didn't expand the last node, "housing", compared to the original tree.

Exercise 3





```
##
## Classification tree:
## tree(formula = as.factor(y) ~ ., data = train, mindev = 5e-04)
## Variables actually used in tree construction:
##  [1] "poutcome"  "month"     "contact"   "marital"   "day"       "campaign"
##  [7] "job"       "pdays"     "age"       "balance"   "housing"   "education"
## [13] "previous"
## Number of terminal nodes:  122
## Residual mean deviance:  0.5213 = 9363 / 17960
## Misclassification error rate: 0.09362 = 1693 / 18084
```

```
##
## Classification tree:
## snip.tree(tree = fit3, nodes = c(581L, 17L, 577L, 79L, 37L, 77L,
## 576L, 153L, 580L, 6L, 1157L, 16L, 5L, 1156L, 156L, 152L, 579L,
## 7L))
## Variables actually used in tree construction:
## [1] "poutcome" "month"    "contact"  "pdays"    "age"      "day"      "balance"
## [8] "housing"
## Number of terminal nodes:  21
## Residual mean deviance:  0.5706 = 10310 / 18060
## Misclassification error rate: 0.1041 = 1882 / 18084
```

Complementary answer

The optimal amount of leaves is 21.Looking at the summary we can see that fit3 uses 13 variables and our optimal tree using 8. These are the variables are the ones who is most important for the tree construction. The variables are: “poutcome”,“month”,“contact”,“pdays”,“age”, “day”, “balance” , “housing”.

Exercise 4

```
##      ffitTest
##         no   yes
##   no  11812   167
##   yes  1294   291
```

Complementary answer

Accuracy = 0.8922884

f1_score = 0.2848752

The accuracy is close to 90%, so the predictive power of the model seems to be quite good. The F1 score is usually preferred when data is unbalanced (for instance, when the quantity of examples in one class outnumbers the ones from the other class).

Exercise 5

```
##      pred5
##         no   yes
##   no  11030   949
##   yes   771   814
```

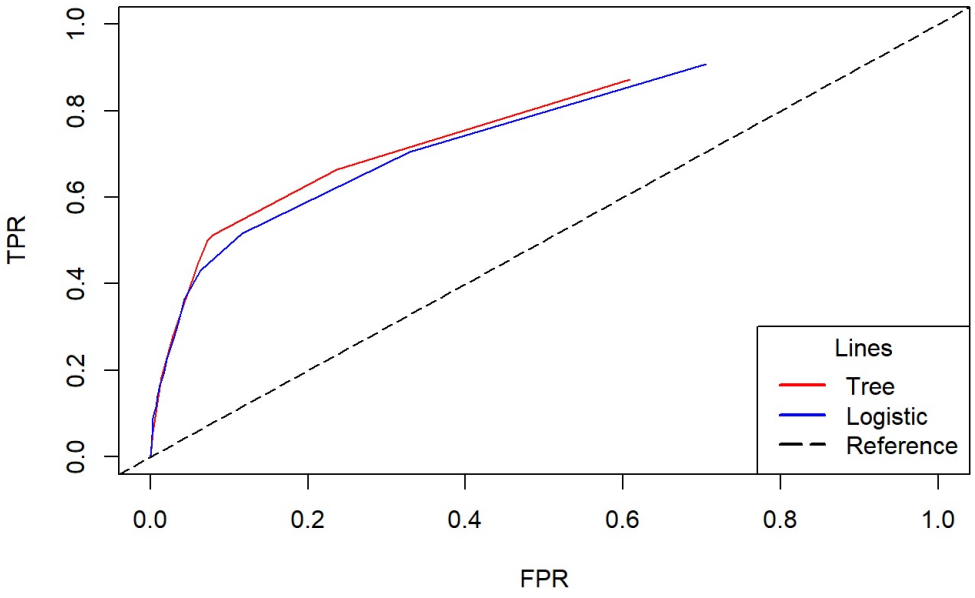
Complementary answer

Accuracy = 0.8731937

F1 = 0.4862605

We can see that in task 4 the accuracy was better than the accuracy obtained from the loss matrix. This means that the other model has a better predictive power than this one. However, we can see that the f1-score has increased almost by a factor of 2. Which means the that this model’s ability to both capture positive cases and be accurate with the cases it does capture is better.

Exercise 6



Complementary answer

We can see that the the area under the red line is greater than the blue line. Meaning that the decision tree provides a better model. The logistic model is still a great model. But in comparison to the decision tree, it is not as good.

The precision-recall curve might be a better option if there are more data points available, which would make the plot more accurate.

TASK 3

Exercise 1

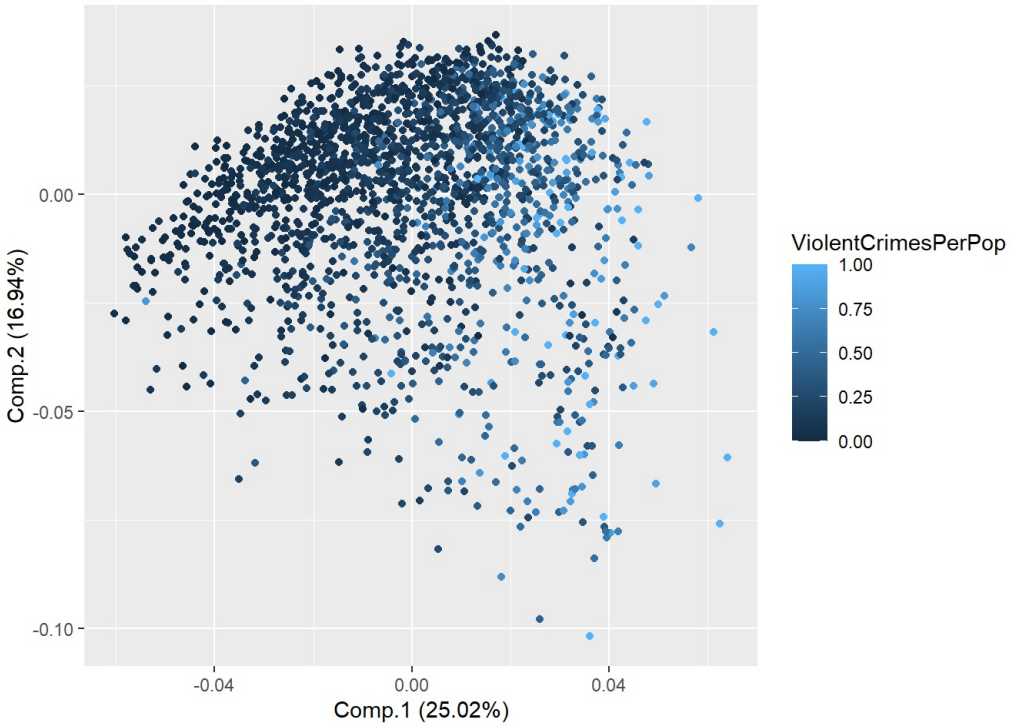
```
## [1] 34
```

```
## [1] 0.2501699 0.1693597
```

95% at 35, 25% and 17% (16.9)

Exercise 2

```
##      medFamInc      medIncome    PctKids2Par     pctWInvInc PctPopUnderPov
##     -0.1833080     -0.1819830     -0.1755423     -0.1748683      0.1737978
```



Yes many features seem to have a relatively big contribution.

The 5 values sound reasonable and should have a logical relationship to the crime level

the area up to left seems both most dense and the darkest, a low pc1 seems to contibute alot toward lower VCPP

Pov1 = 0.2502 Pov2 = 0.1693

Exercise 3

```
## [1] "Train error"
```

```
## [1] 0.2591772
```

```
## [1] "Test error"
```

```
## [1] 0.4000579
```

Compute training and test errors for these data and comment on the quality of model.

The the error for both test and train seems to be high. It is a complex case with a lot of affecting factors so some errors are to be expected. The difference between train and test does not seem to be that big which indicates a relatively well fitted model.

Exercise 4

```
## [1] "calculated optimal train"
```

```
## [1] 0.2592247
```

```
## [1] "Lm train error"
```

```
## [1] 0.2591772
```

```
## [1] "calculated optimal test"
```

```
## [1] 0.3997238
```

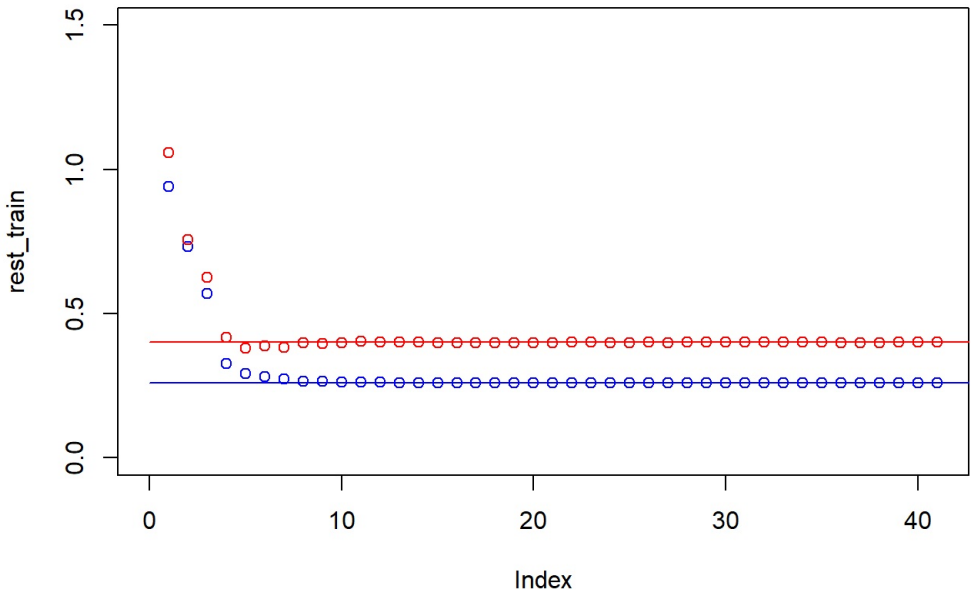
```
## [1] "Lm test error"
```

```
## [1] 0.4000579
```

```
## [1] "Early stopping index and MSE"
```

```
## [1] 2183
```

```
## [1] 0.3769468
```



Min test_error and the early stopping point appears at index 2183 with MSE of 0.377. The results from the 3rd task and the computed optimal values in this exercise are basically the same both in the plot and the printed results. We can also see that the test error as indicated by the early stopping point did reach a lower error rate at an earlier theta but this is with a higher train error.