Lab 1 Report

Zuxiang Li

11/19/2019

Assignment 1. Spam classification with nearest neighbors

1. Import the data into R and divide it into training and test sets (50%/50%) by using the following code:

```
library(openxlsx)
data<-read.xlsx("material/spambase.xlsx")
n <- dim(data)[1]
set.seed(12345)
id <- sample(1:n, floor(n*0.5))
train <- data[id,]
test <- data[-id,]
```

2. Use logistic regression (functions glm(), predict()) to classify the training and test data by the classification principle and report the confusion matrices (use table()) and the misclassification rates for training and test data. Analyse the obtained results.

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## [1] "Rate= 0.5 train"
##      Target
## Model   0   1
##     0 804  93
##     1 127 346
## [1] "Misclassfication Rate= 0.160583941605839"
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## [1] "Rate= 0.5 test"
##      Target
## Model   0   1
##     0 808  92
##     1 143 327
## [1] "Misclassfication Rate= 0.171532846715328"
```

3. Use logistic regression to classify the test data by the classification principle and report the confusion matrices (use table()) and the misclassification rates for

training and test data. Compare the results. What effect did the new rule have?

```
cft(0.8,train)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## [1] "Rate= 0.8 train"
##      Target
## Model   0   1
##     0 921 333
##     1  10 106
## [1] "Misclassfication Rate= 0.25036496350365"
```

```
cft(0.8,test)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## [1] "Rate= 0.8 test"
##      Target
## Model   0   1
##     0 931 314
##     1  20 105
## [1] "Misclassfication Rate= 0.243795620437956"
```

By increasing the rate, the misclassfication rate increases too.

4. Use standard classifier kknn() with K=30 from package kknn, report the the misclassification rates for the training and test data and compare the results with step 2.

```
##      Target
## Model   0   1
##     0 702 180
##     1 249 239
```

```
## [1] 0.3131387
```

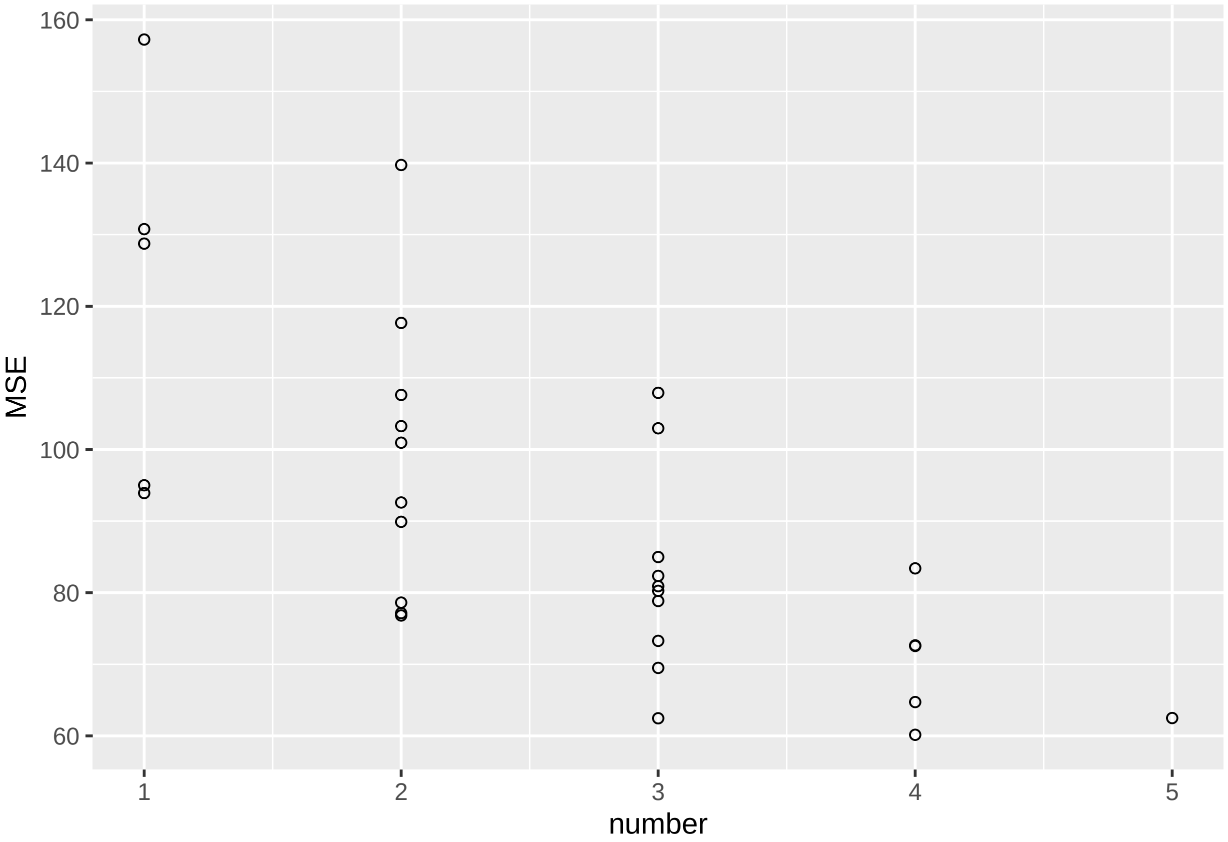
5. Repeat step 4 for K=1 and compare the results with step 4. What effect does the decrease of K lead to and why?

```
##      Target
## Model   0   1
##     0 644 185
##     1 307 234
```

```
## [1] 0.3591241
```

Assignment 3. Feature selection by cross-validation in a linear model.

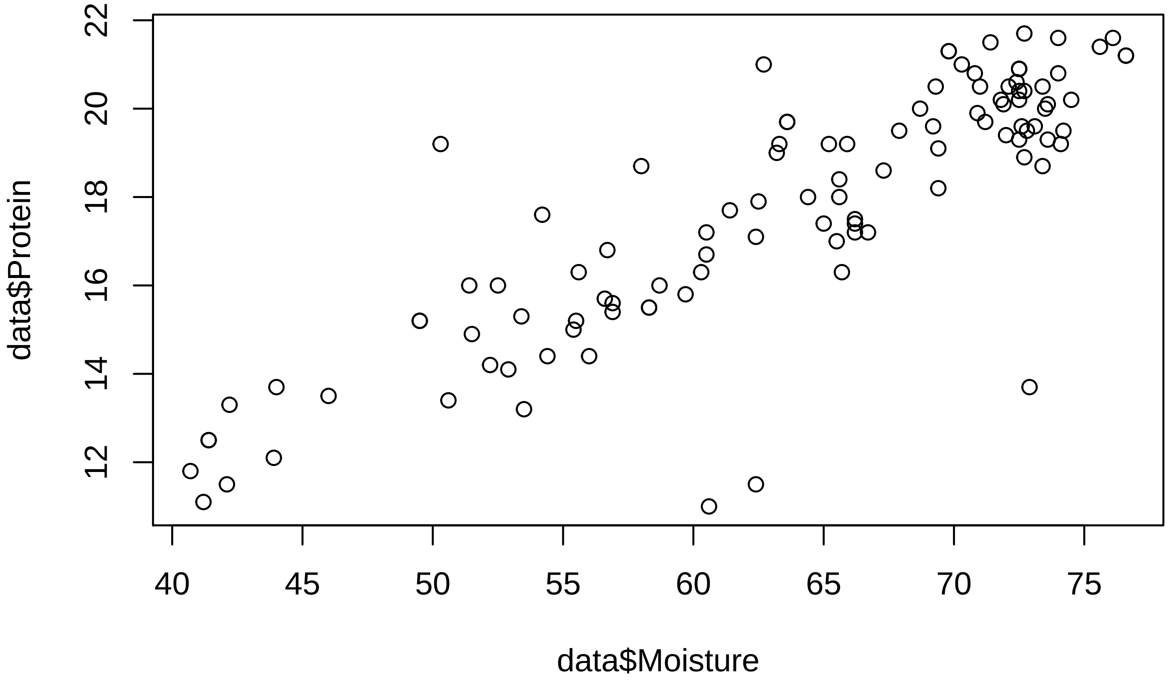
```
## $CV
## [1] 60.15763
##
## $Features
## [1] 1 0 1 1 1
##
## $plot
```



From the plot we can observe that with the number of features increases, in general, the values of MSE decreases. For the best subset, its' MSE minimized when number of features is 4. When number of features is small, model can be inaccurate, but if with too many parameters, the model will be difficult to use and interpret since it's overfitted,

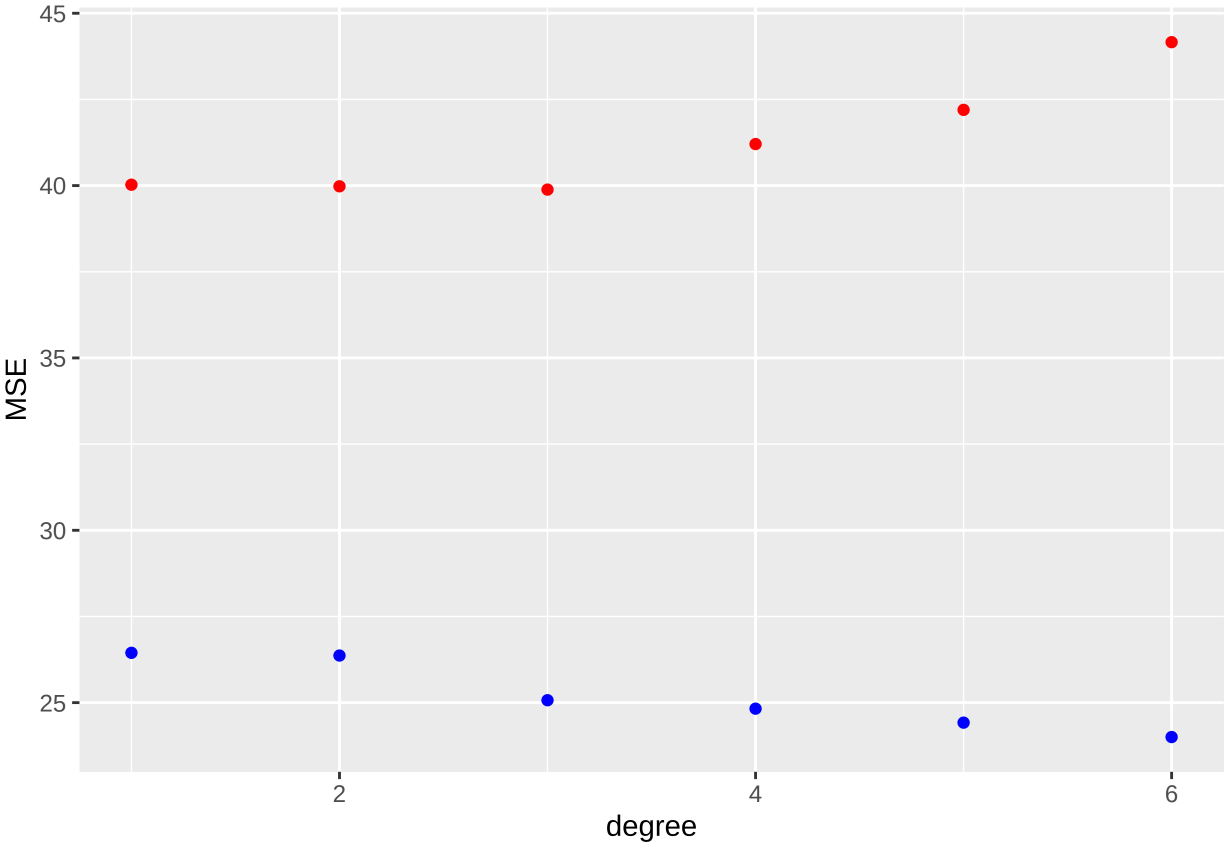
Assignment 4. Linear regression and regularization

1. Import data to R and create a plot of Moisture versus Protein. Do you think that these data are described well by a linear model?



2.Consider model in which Moisture is normally distributed, and the expected Moisture is a polynomial function of Protein including the polynomial terms up to power (i.e M1 is a linear model, M2 is a quadratic model and so on). Report a probabilistic model that describes . Why is it appropriate to use MSE criterion when fitting this model to a training data?

```
## [1] 40.02562 39.97895 39.88347 41.20548 42.19681 44.16041
```



4. Perform variable selection of a linear model in which Fat is response and Channel1-Channel100 are predictors by using stepAIC. Comment on how many variables were selected.

```
library(MASS)
data<-read.xlsx("material/tecator.xlsx")
n_data<-data.frame(Fat=data$Fat)
n_data<-cbind(n_data,data[,2:101])

fit<-lm(Fat~.,data=n_data)
step<-stepAIC(fit,direction = "both",trace = 0)
length(coef(step))-1
```

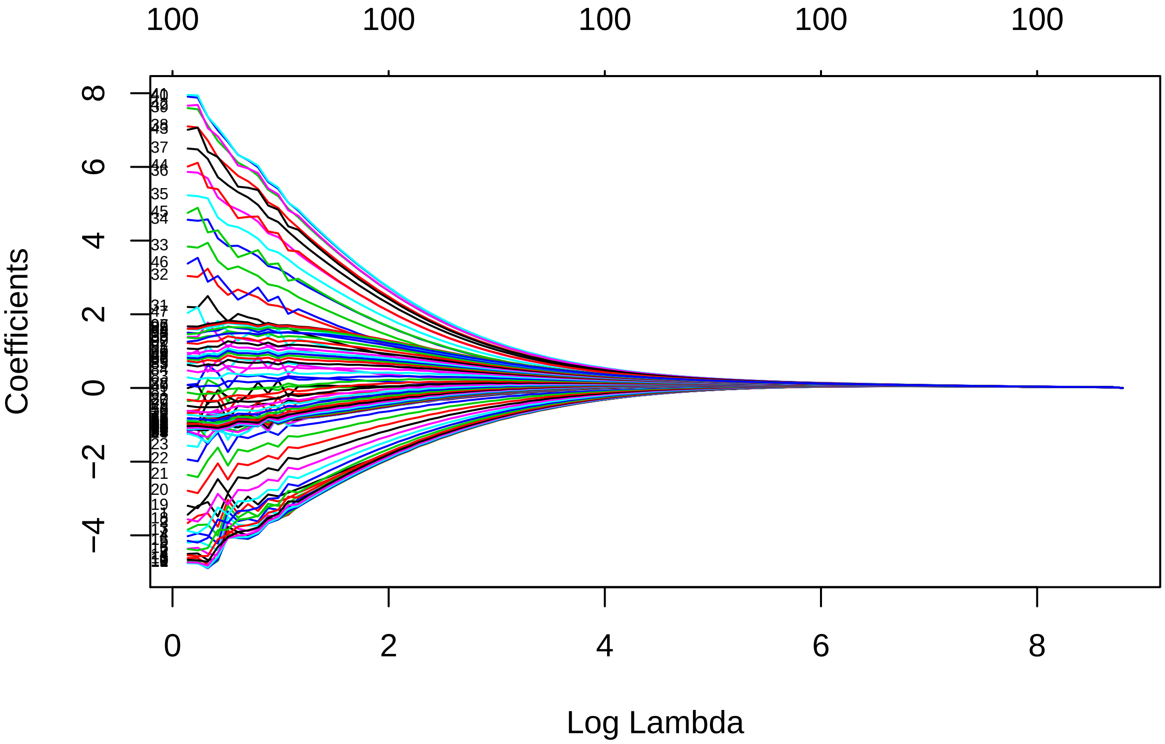
```
## [1] 63
```

5.Fit a Ridge regression model with the same predictor and response variables. Present a plot showing how model coefficients depend on the log of the penalty factor and report how the coefficients change with

```
library(glmnet)
```

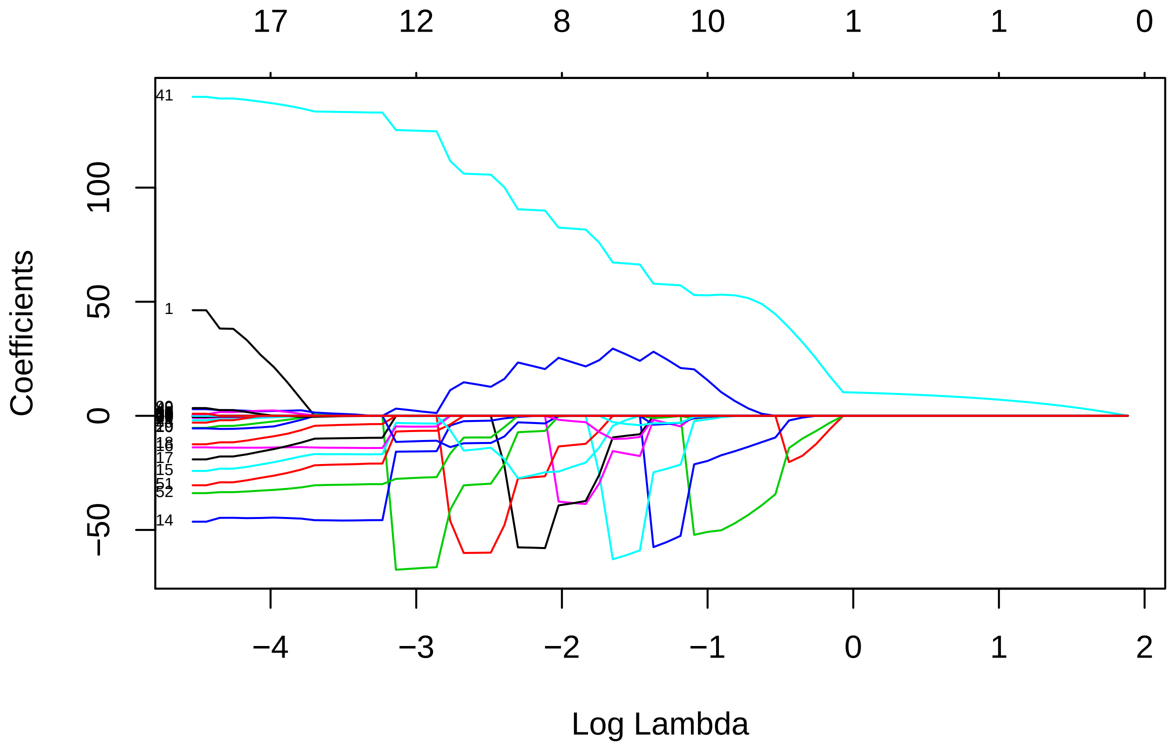
```
## Loading required package: Matrix
## Loaded glmnet 3.0
```

```
covariates=n_data[,-1]
response=n_data[,1]
y=test[,1]
model0=glmnet(as.matrix(covariates), response, alpha=0,family="gaussian")
plot(model0, xvar="lambda", label=TRUE)
```



6. dsdada

```
model1=glmnet(as.matrix(covariates), response, alpha=1,family="gaussian")
plot(model1, xvar="lambda", label=TRUE)
```

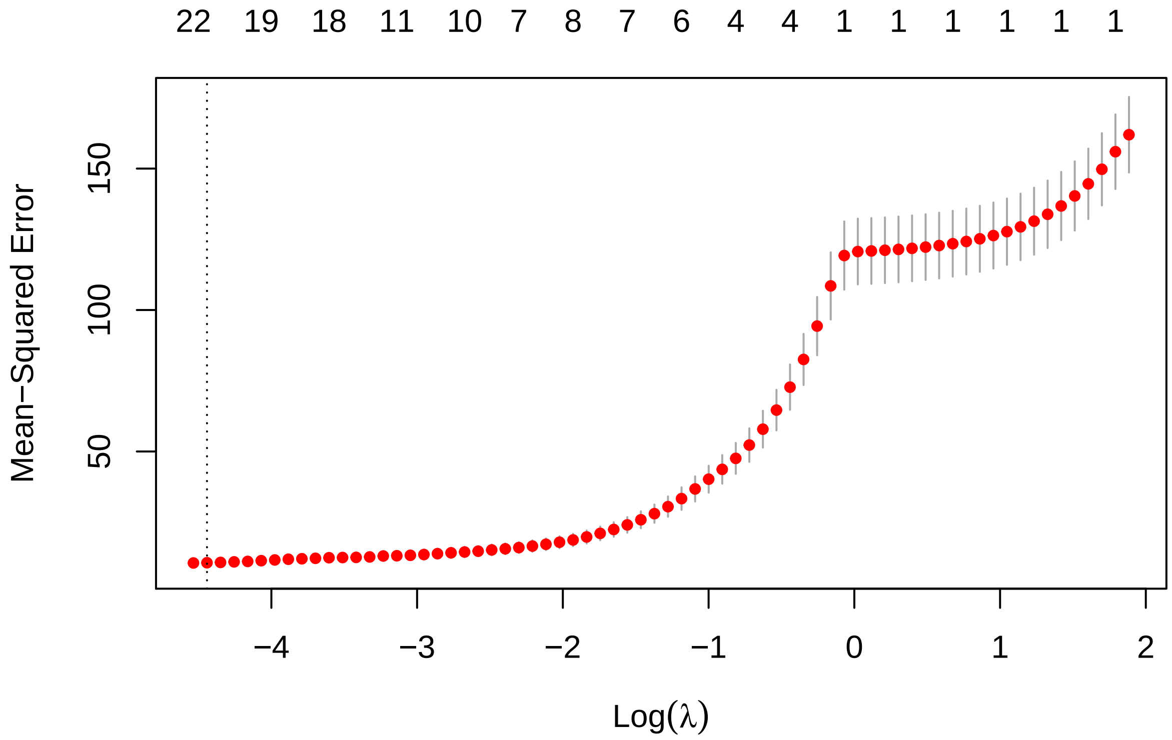


7.

```
new_lambda<-c(model1$lambda,0)
new_lambda
```

```
##  [1] 6.58364950 5.99877622 5.46586148 4.98028941 4.53785422 4.13472376
##  [7] 3.76740630 3.43272031 3.12776690 2.84990471 2.59672703 2.36604095
## [13] 2.15584840 1.96432877 1.78982321 1.63082025 1.48594268 1.35393562
## [19] 1.23365572 1.12406114 1.02420265 0.93321531 0.85031103 0.77477174
## [25] 0.70594316 0.64322911 0.58608641 0.53402011 0.48657924 0.44335288
## [31] 0.40396663 0.36807935 0.33538019 0.30558594 0.27843853 0.25370282
## [37] 0.23116456 0.21062854 0.19191688 0.17486751 0.15933276 0.14517808
## [43] 0.13228086 0.12052939 0.10982189 0.10006562 0.09117606 0.08307623
## [49] 0.07569597 0.06897135 0.06284412 0.05726123 0.05217430 0.04753928
## [55] 0.04331602 0.03946794 0.03596172 0.03276698 0.02985605 0.02720372
## [61] 0.02478702 0.02258501 0.02057862 0.01875047 0.01708473 0.01556697
## [67] 0.01418404 0.01292397 0.01177584 0.01072971 0.00000000
```

```
mod_lasso=cv.glmnet(as.matrix(covariates), response, alpha=1,family="gaussian",lambda = new_lambda,stanc
plot(mod_lasso)
```



```
c<-coef(mod_lasso, s = "lambda.min")
length(which(c!=0))-1
```

```
## [1] 100
```