Machine Learning Assignment 1

Agustín Valencia - aguva779

11/19/2019

Assignment 1. Spam classification with nearest neighbors

2. Use logistic regression to classify the training and test data by the classification principle $\hat{Y} = 1$ if $p(Y = 1|X) > 0.5$, otherwise $\hat{Y} = 0$ and report the confusion matrices and the misclassification rates for train and test data. Analyze the obtained results.

Evaluating the model with training data :

```
## Classification Performance : train set - trigger = 0.5
## [1] "Confusion Matrix"
##        predictions
## targets   0   1
##       0 875  56
##       1 156 283
## Rates details:
##  TPR = 83.48083 % - TNR = 84.86906 % - FPR = 16.51917 % - FNR = 15.13094 %
##  Misclassification Rate =  15.47445 %
```

Now, with unseen data it can be observed that the misclassification rate increased, though numbers still consistent.

```
## Classification Performance : test set - trigger = 0.5
## [1] "Confusion Matrix"
##        predictions
## targets   0   1
##       0 865  86
##       1 162 257
## Rates details:
##  TPR = 74.92711 % - TNR = 84.2259 % - FPR = 25.07289 % - FNR = 15.7741 %
##  Misclassification Rate =  18.10219 %
```

3. Use logistic regression to classify the test data by the classification principle $\hat{Y} = 1$ if $p(Y = 1|X) > 0.8$, otherwise $\hat{Y} = 0$

Setting a higher trigger implies that the classifier will be more selective, then it is expected to decrease the amount of mails being labeled as spam.

The training stats:

```
## Classification Performance : train set - trigger = 0.8
## [1] "Confusion Matrix"
##        predictions
## targets   0   1
##       0 898  33
##       1 216 223
## Rates details:
##  TPR = 87.10938 % - TNR = 80.61041 % - FPR = 12.89062 % - FNR = 19.38959 %
##  Misclassification Rate =  18.17518 %
```

Testing stats:

```
## Classification Performance : test set - trigger = 0.8
## [1] "Confusion Matrix"
##        predictions
## targets   0   1
##       0 892  59
##       1 229 190
## Rates details:
##  TPR = 76.30522 % - TNR = 79.57181 % - FPR = 23.69478 % - FNR = 20.42819 %
##  Misclassification Rate =  21.0219 %
```

Although the misclassification rate has increased, the false positive rate, i.e., the amount of valid email being sent to the spambox, has decreased, which from a user perspective could be more valuable than a higher accuracy on true positives.

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

```
## Classification Performance : train knn - k = 30
## [1] "Confusion Matrix"
##        predictions
## targets   0   1
##       0 779 152
##       1  77 362
## Rates details:
##  TPR = 70.42802 % - TNR = 91.00467 % - FPR = 29.57198 % - FNR = 8.995327 %
##  Misclassification Rate =  16.71533 %
```

```
## Classification Performance : test knn - k = 30
## [1] "Confusion Matrix"
##        predictions
## targets   0   1
##       0 702 249
##       1 180 239
## Rates details:
##  TPR = 48.97541 % - TNR = 79.59184 % - FPR = 51.02459 % - FNR = 20.40816 %
##  Misclassification Rate =  31.31387 %
```

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

```
## Classification Performance : train knn - k = 1
## [1] "Confusion Matrix"
##        predictions
## targets   0   1
##       0 931   0
##       1   0 439
## Rates details:
##  TPR = 100 % - TNR = 100 % - FPR = 0 % - FNR = 0 %
##  Misclassification Rate =  0 %
```

```
## Classification Performance : test knn - k = 1
## [1] "Confusion Matrix"
##        predictions
## targets   0   1
##       0 644 307
##       1 185 234
```

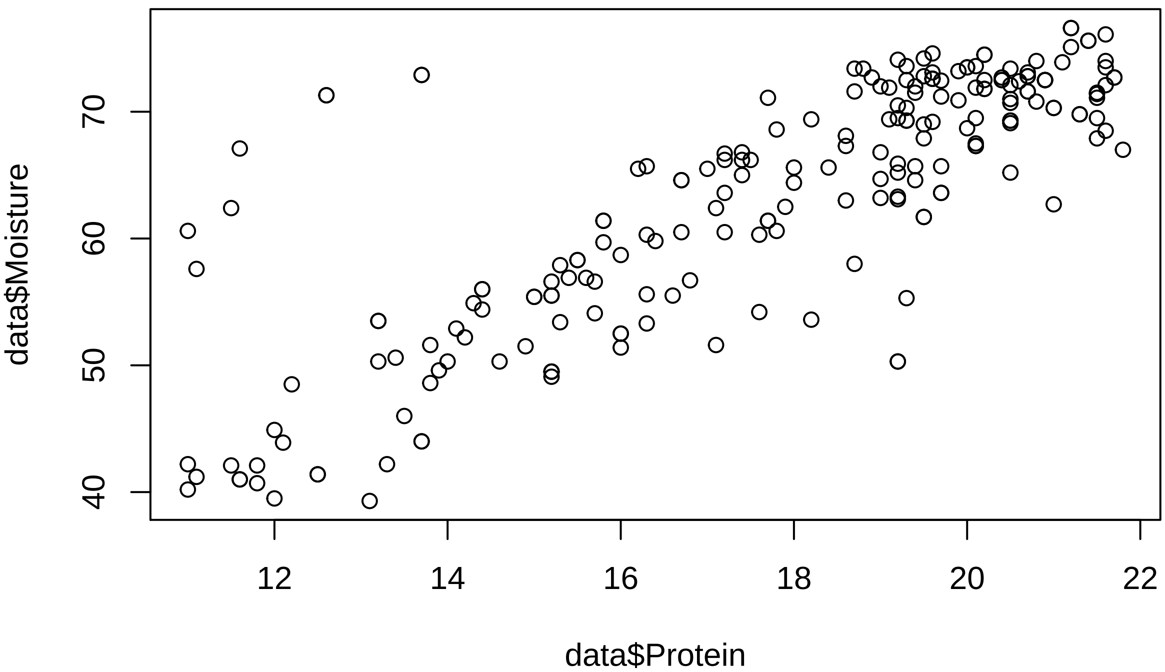
```
## Rates details:
##  TPR = 43.25323 % - TNR = 77.68396 % - FPR = 56.74677 % - FNR = 22.31604 %
##  Misclassification Rate =  35.91241 %
```

If we assign k=1 training misclassification is 0%, this means we are overfitting our model, thus the misclassification for the testing set may be bigger than other scenarios.

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

Assignment 4. Linear regression and regularization

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

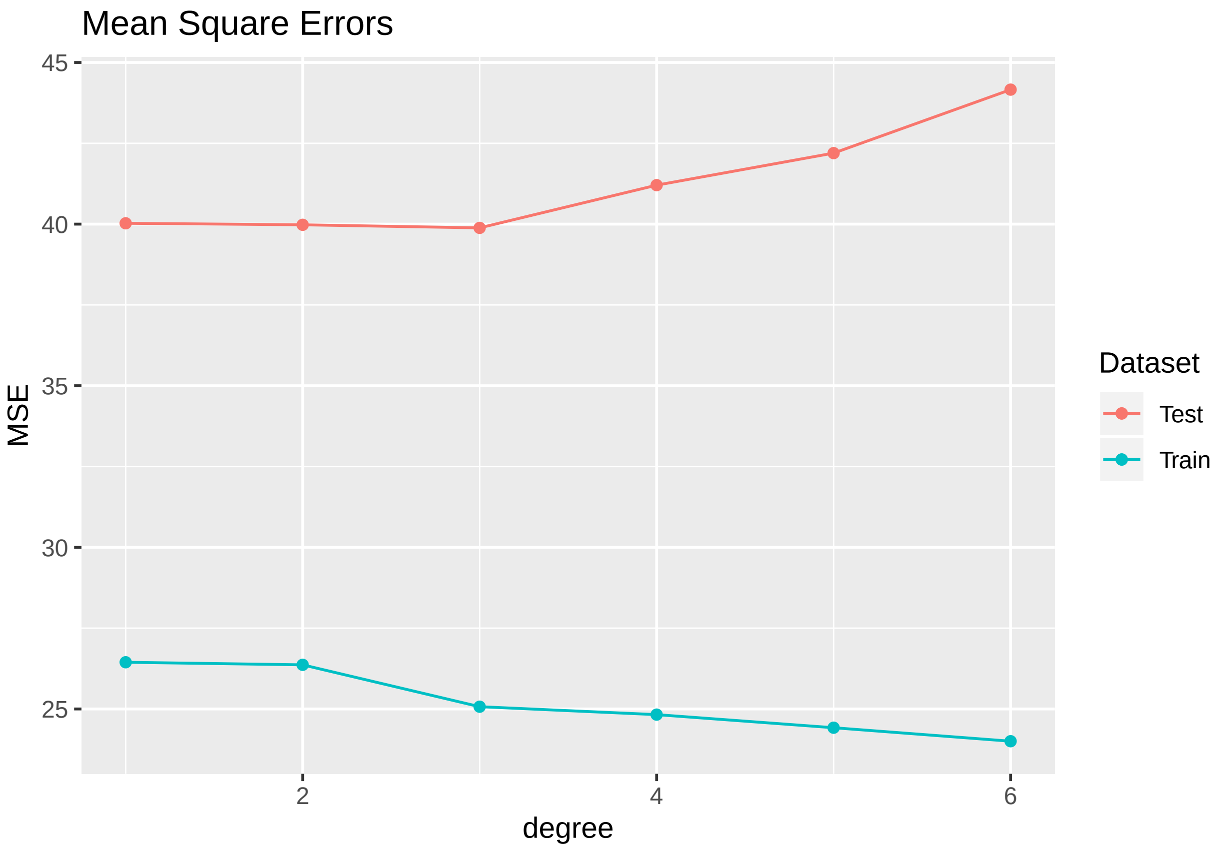


By the plot, although there are some outliers, it seems that the data could be approximated by a linear model.

2. Consider model M_i in which Moisture is normally distributed and the expected Moisture is polynomial

PUT DOWN SOME MATH BAAAAAM !!

3. Divide the data (50/50) and fit models $M_i, i = 1, \cdots, 6$. For each model, record the training and validation MSE and present a plot showing how training and validation MSE depend on i. Which model is best according to this plot? How do MSE values change and why? Interpret this picture in bias-variance tradeoff.



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.

```
cat("There were selected", length(selected_vars), "variables\n")
```

```
## There were selected 64 variables
```