Spam Classification Using Logistic Regression and K-Nearest Neighbors

Abhinay Krishna Vellala

12/19/2020

The data constains infomraiton about the frequency of words and characters of 2740 emails. OUr goal is to classify them as spam(spam = 1) or not spam(spam = 0) based on the data.

Covariates Word1 - Word48 : 48 variables which has frequency of that word.

Target Spam: 0(Not spam) or 1(spam)

1.1 Divide data

```
library(readxl)
spamdata = read_excel("spambase.xlsx")

# divide data
n = dim(spamdata)[1]
set.seed(12345)
id = sample(1:n, floor(0.5*n))
train = spamdata[id,]
test = spamdata[-id,]
```

1.2 Logistic Regression

Lets perform logistic regression with classification pronciple set to 0.5 and check the classification error.

```
# model
model = glm(Spam ~ ., family = "binomial", data = train)
```

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

```
# predict and test

logisticTest = function(p){
  ypred = predict(model, newdata = test, type = "response")
  ypred1 = ifelse(ypred > p, 1,0)
  confMat = table(test$Spam, ypred1)
  misclass = 1 - sum(diag(confMat))/sum(confMat)
  return(misclass)
}
```

```
cat("\n\nFor the classification principle 0.5, the misclassification error rate: ",logisticTest(0.5))
```

```
##
##
## For the classification principle 0.5, the misclassification error rate:  0.1715328
```

1.3 Classification principle

FOr the new classification principle

```
cat("\nFor the classification principle 0.8, the misclassification error rate: ",logisticTest(0.8))
```

```
##
## For the classification principle 0.8, the misclassification error rate:  0.2437956
```

As the classification creteria increased, the error rate also increased. This is because the function is classifying the emails as Spam only if the probability of being spam given the data is greater than 0.8 in which many data points couldn't achieve and they are misclassified.

1.4: KNN classification

Now, we perform KNN classification on the same data to check if it classifies better.

```
# knn classification
library(kknn)
knnModel = function(k){
  modelKnn = kknn(as.factor(Spam) ~ ., train = train, test = test, k = k)
  confMat = table(modelKnn$fitted.values, test$Spam)
  misclass = 1 - sum(diag(confMat))/sum(confMat)
  return(misclass)
}

cat("For k = 30, the misclassification error rate is",knnModel(30))
```

```
## For k = 30, the misclassification error rate is 0.3131387
```

Comparing with Logistic regression, the error rate is higher with KNN algorithm.

1.5 KNN Continued

```
cat("For k = 1, the misclassification error rate is",knnModel(1))
```

```
## For k = 1, the misclassification error rate is 0.3591241
```

Now, with K = 1, the error rate is higher than K = 30. This is because, the number of neighbours let the particular value decide how strong the email is spam or not spam. With 30 neighbours the algorithm was able to classify the value as spam as the result is based on 30 nearest values of spam or not spam which has maximum spam values. With K = 1, the item is classified based on one neighbour. If the nearest neighbour is spam, then even the current value is spam. This might not be true in all the cases. This is the reason the error rate is high.