kNN_Regression

Austin Girouard

2023-03-21

kNN Regression

Load the Data

```
df <- read.csv ("job_profitability.csv", header = TRUE) # specifies where to load data from
df <- df[,c(3, 4, 5, 10)] # specifies which columns we want
str(df) # print data frame structure
```

```
## 'data.frame':    14479 obs. of  4 variables:
##  $ Jobs_Gross_Margin: num  -4.01 254.13 151.83 -32.15 222.7 ...
##  $ Labor_Pay        : num  0 91 0 0 0 ...
##  $ Labor_Burden     : num  22.2 14.9 133.2 81.2 66.3 ...
##  $ Jobs_Total       : num  79.5 360 289 49 289 ...
```

Sample Training and Testing Data

```
set.seed(1234)
i <- sample(1:nrow(df), nrow(df) * 0.8, replace = FALSE) # split data into 80/20 train/test
train <- df[i, ] # training data
test <- df[-i, ] # testing data
```

kNN Regression

Here is the correlation and mse prior to scaling

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
fit <- knnreg(train[, 2:4], train[, 1], k = 3)
# Evaluate
pred2 <- predict(fit, test[, 2:4])
cor_knn1 <- cor(pred2, test$Jobs_Gross_Margin)
mse_knn1 <- mean((pred2 - test$Jobs_Gross_Margin)^2)
print(paste("cor =", cor_knn1))
```

```
## [1] "cor = 0.949799349568583"
```

```
print(paste("mse =", mse_knn1))
```

```
## [1] "mse = 43812.1511090968"
```

Here is the correlation and mse after scaling

```
# Scale the data
train_scaled <- train[, 2:4]
means <- sapply(train_scaled, mean)
stdvs <- sapply(train_scaled, sd)
train_scaled <- scale(train_scaled, center=means, scale=stdvs)
test_scaled <- scale(test[, 2:4], center=means, scale=stdvs)

# fit the model
fit <- knnreg(train_scaled,train$Jobs_Gross_Margin,k=3)
# evaluate

pred2 <- predict(fit, test_scaled)
cor_knn1 <- cor(pred2, test$Jobs_Gross_Margin)
mse_knn1 <- mean((pred2 - test$Jobs_Gross_Margin)^2)
print(paste("cor=", cor_knn1))
```

```
## [1] "cor= 0.94888749494546"
```

```
print(paste("mse=", mse_knn1))
```

```
## [1] "mse= 44526.5698662085"
```

Finding the best k value

```
cor_k <- rep(0, 20)
mse_k <- rep(0, 20)
i <- 1
for (k in seq(1, 39, 1)){
  fit_k <- knnreg(train_scaled,train$Jobs_Gross_Margin, k=k)
  pred_k <- predict(fit_k, test_scaled)
  cor_k[i] <- cor(pred_k, test$Jobs_Gross_Margin)
  mse_k[i] <- mean((pred_k - test$Jobs_Gross_Margin)^2)
  print(paste("k=", k, cor_k[i], mse_k[i]))
  i <- i + 1
}
```

```
## [1] "k= 1 0.91443237084882 75031.6590480577"
## [1] "k= 2 0.935621725586118 55630.4417163793"
## [1] "k= 3 0.94888749494546 44526.5698662085"
## [1] "k= 4 0.952486064898813 41358.4751245161"
## [1] "k= 5 0.952248849788458 41564.6587804882"
## [1] "k= 6 0.953496947177133 40496.661097757"
## [1] "k= 7 0.95330971567066 40692.1285460223"
## [1] "k= 8 0.953156773410733 40909.7348446066"
## [1] "k= 9 0.95389419337434 40374.0799654498"
## [1] "k= 10 0.955663007864068 38919.4962736549"
## [1] "k= 11 0.957000918797561 37892.6488154394"
## [1] "k= 12 0.956836649848018 38049.3013389117"
## [1] "k= 13 0.956454882738785 38450.262654351"
## [1] "k= 14 0.956371302335428 38608.1579318218"
## [1] "k= 15 0.956853777466829 38355.6825716498"
## [1] "k= 16 0.956791040684962 38434.3227626724"
## [1] "k= 17 0.957191866567774 38144.5006673951"
## [1] "k= 18 0.956930753263041 38469.7697806985"
## [1] "k= 19 0.956908410512685 38505.4071898562"
## [1] "k= 20 0.956902179650109 38662.5505176019"
## [1] "k= 21 0.956707262019351 38924.5985663727"
## [1] "k= 22 0.956088216783822 39471.8969084156"
## [1] "k= 23 0.956014178954005 39661.614608335"
## [1] "k= 24 0.955664123663985 40065.5511307952"
## [1] "k= 25 0.955691559449313 40121.3830442157"
## [1] "k= 26 0.955626974032989 40220.8243791265"
## [1] "k= 27 0.955140260553024 40683.7983995687"
## [1] "k= 28 0.954804828032791 41039.5769561952"
## [1] "k= 29 0.954408580536707 41436.3214254921"
## [1] "k= 30 0.954385262623748 41544.8907338456"
## [1] "k= 31 0.954539078596556 41368.8995530087"
## [1] "k= 32 0.954471321791013 41421.0153402227"
## [1] "k= 33 0.954419522005575 41538.0927123375"
## [1] "k= 34 0.954394726850801 41635.3198565272"
## [1] "k= 35 0.954142175137488 41914.4564722525"
## [1] "k= 36 0.954012701873491 42132.5760480365"
## [1] "k= 37 0.954002481148275 42153.8601048852"
## [1] "k= 38 0.953608420843434 42638.5384826208"
## [1] "k= 39 0.953233992075363 43002.7780514644"
```

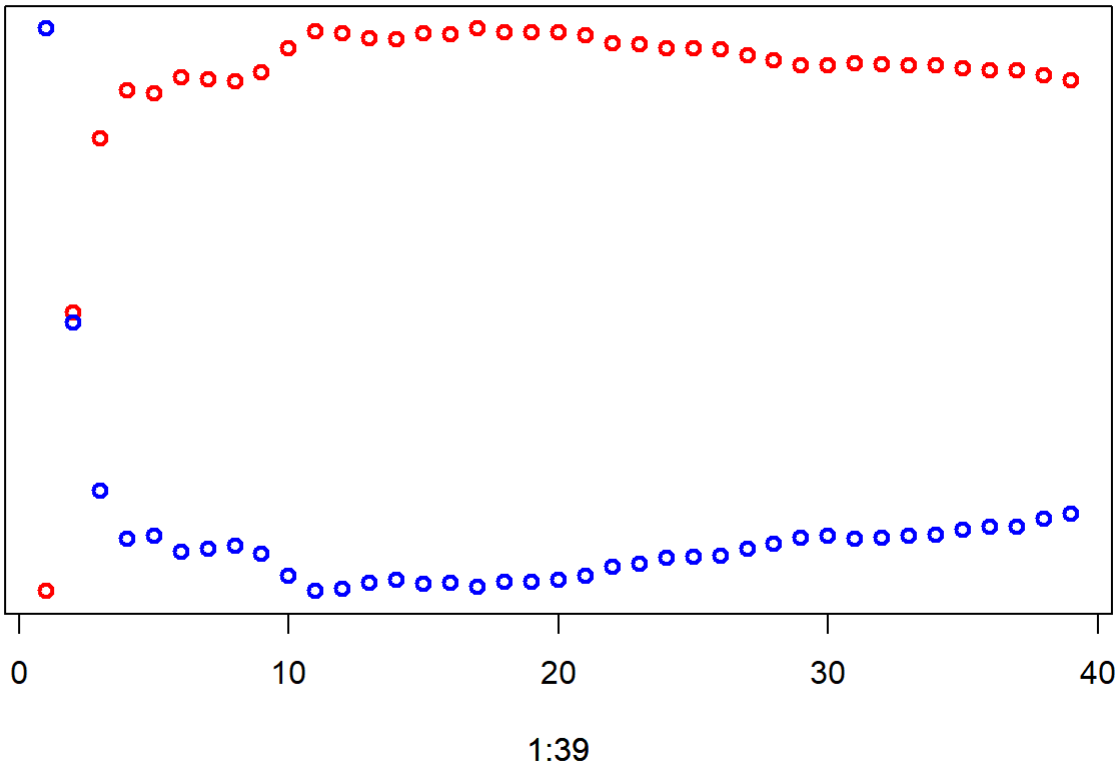
```
plot(1:39, cor_k, lwd=2, col='red', ylab="", yaxt='n')
par(new=TRUE)
plot(1:39, mse_k, lwd=2, col='blue', labels=FALSE, ylab="", yaxt='n')
```

```
## Warning in plot.window(...): "labels" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "labels" is not a graphical parameter
```

```
## Warning in box(...): "labels" is not a graphical parameter
```

```
## Warning in title(...): "labels" is not a graphical parameter
```



We can visually see that at k=17 the red correlation is highest and the blue mse is lowest.

```
which.min(mse_k)
```

```
## [1] 11
```

```
which.max(cor_k)
```

```
## [1] 17
```

```
best_k <- which.max(cor_k)
```

Here is the correlation and mse after finding the best k value

```
# fit the model using the best k value found
fit <- knnreg(train_scaled,train$Jobs_Gross_Margin,k=best_k)
# evaluate

pred2 <- predict(fit, test_scaled)
cor_knn1 <- cor(pred2, test$Jobs_Gross_Margin)
mse_knn1 <- mean((pred2 - test$Jobs_Gross_Margin)^2)
print(paste("cor=", cor_knn1))
```

```
## [1] "cor= 0.957191866567774"
```

```
print(paste("mse=", mse_knn1))
```

```
## [1] "mse= 38144.5006673951"
```

```
print(paste("rmse=",  sqrt(mse_knn1)))
```

```
## [1] "rmse= 195.306171606007"
```