SVM Regression RCD18001

Joshua Durana

2022-10-23

Source: https://www.kaggle.com/datasets/hmavrodiev/london-bike-sharing-dataset (https://www.kaggle.com/datasets/hmavrodiev/london-bike-sharing-dataset) This dataset measures the bikes rented in London and the weather

Read and Clean Data

```
bikeData <- read.csv("Data/BikeData.csv", header = TRUE)

#Sets columns into factors
colFactors <- c("weather_code", "is_holiday", "is_weekend", "season")
bikeData[colFactors] <- lapply(bikeData[colFactors], as.factor)

#Remove timestamp
bikeData <- subset(bikeData, select = -c(timestamp))
```

Split Data

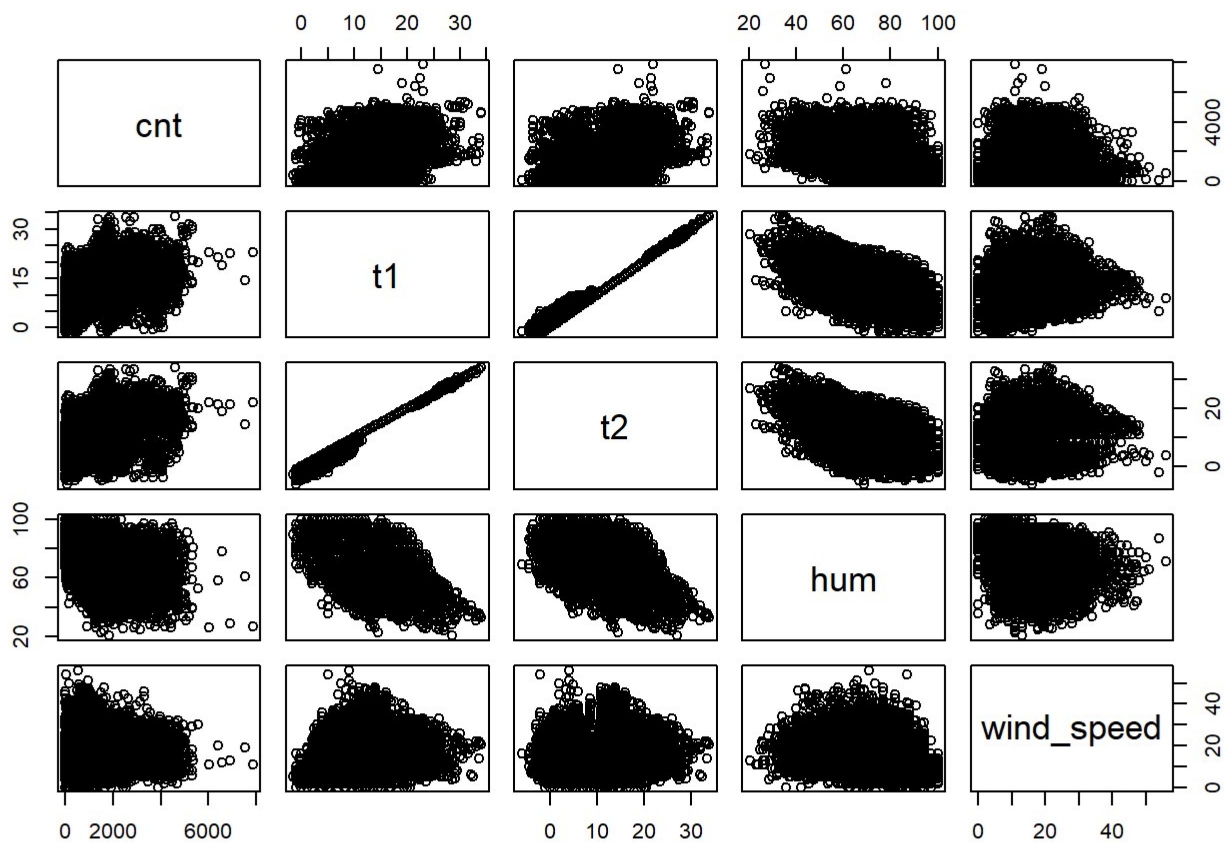
```
set.seed(9582)

spl <- c(train = .6, test = .2, validate = .2)
i <- sample(cut(1:nrow(bikeData), nrow(bikeData) * cumsum(c(0, spl)), labels = names(spl)))

bikeTrain <- bikeData[i == "train",]
bikeTest <- bikeData[i == "test",]
bikeVal <- bikeData[i == "validate",]
```

Data Exploration

```
summary(bikeTrain)
```

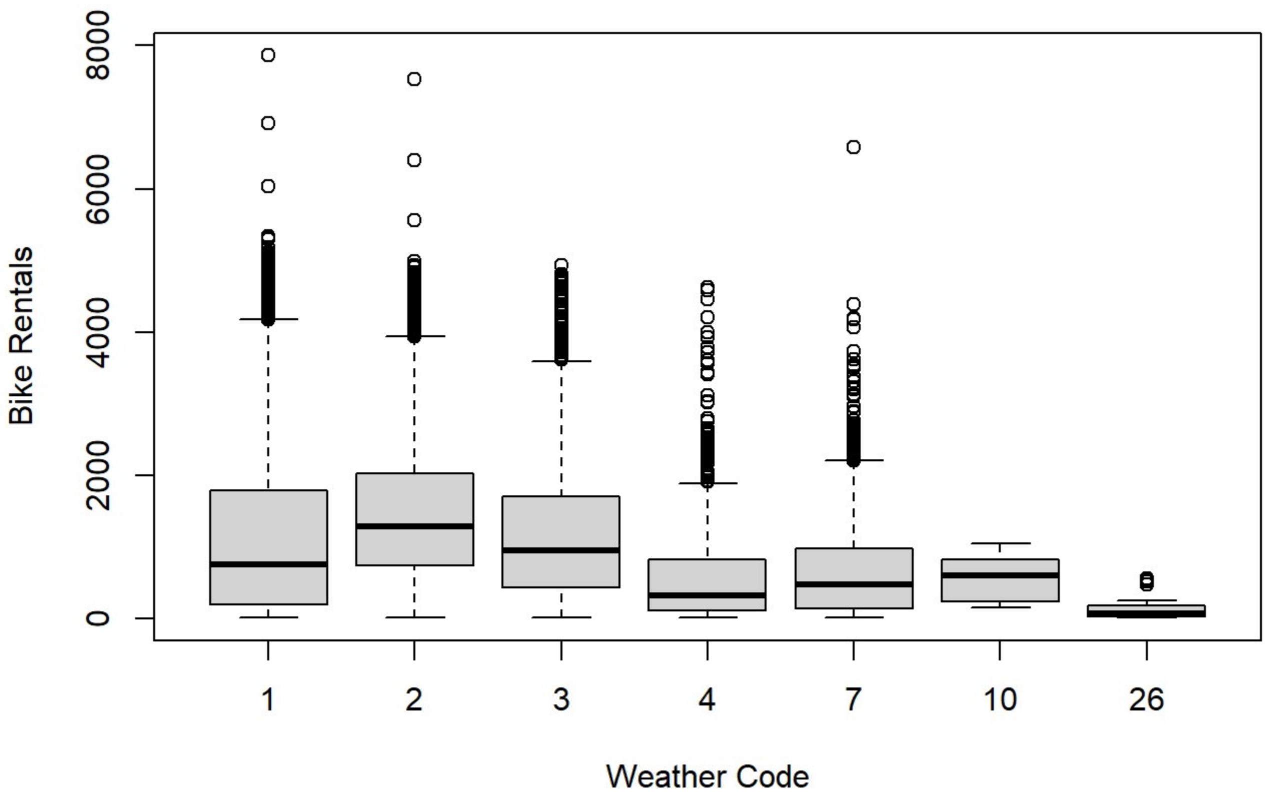
```
##       cnt               t1              t2             hum
##  Min.   :   9.0   Min.   :-1.50   Min.   :-6.00   Min.   : 20.50
##  1st Qu.: 255.8   1st Qu.: 8.00   1st Qu.: 6.00   1st Qu.: 63.00
##  Median : 833.5   Median :12.50   Median :12.50   Median : 75.00
##  Mean   :1138.5   Mean   :12.49   Mean   :11.55   Mean   : 72.46
##  3rd Qu.:1658.2   3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.: 83.00
##  Max.   :7860.0   Max.   :34.00   Max.   :34.00   Max.   :100.00
##
##    wind_speed    weather_code is_holiday is_weekend season
##  Min.   : 0.00   1 :3652      0:10224    0:7464     0:2643
##  1st Qu.:10.00   2 :2401      1:  224    1:2984     1:2648
##  Median :15.00   3 :2159                            2:2563
##  Mean   :15.93   4 : 886                            3:2594
##  3rd Qu.:20.50   7 :1310
##  Max.   :56.00   10:  10
##                  26:  30
```

```
numCol <- unlist(lapply(bikeTrain, is.numeric))
pairs(bikeTrain[,numCol])
```



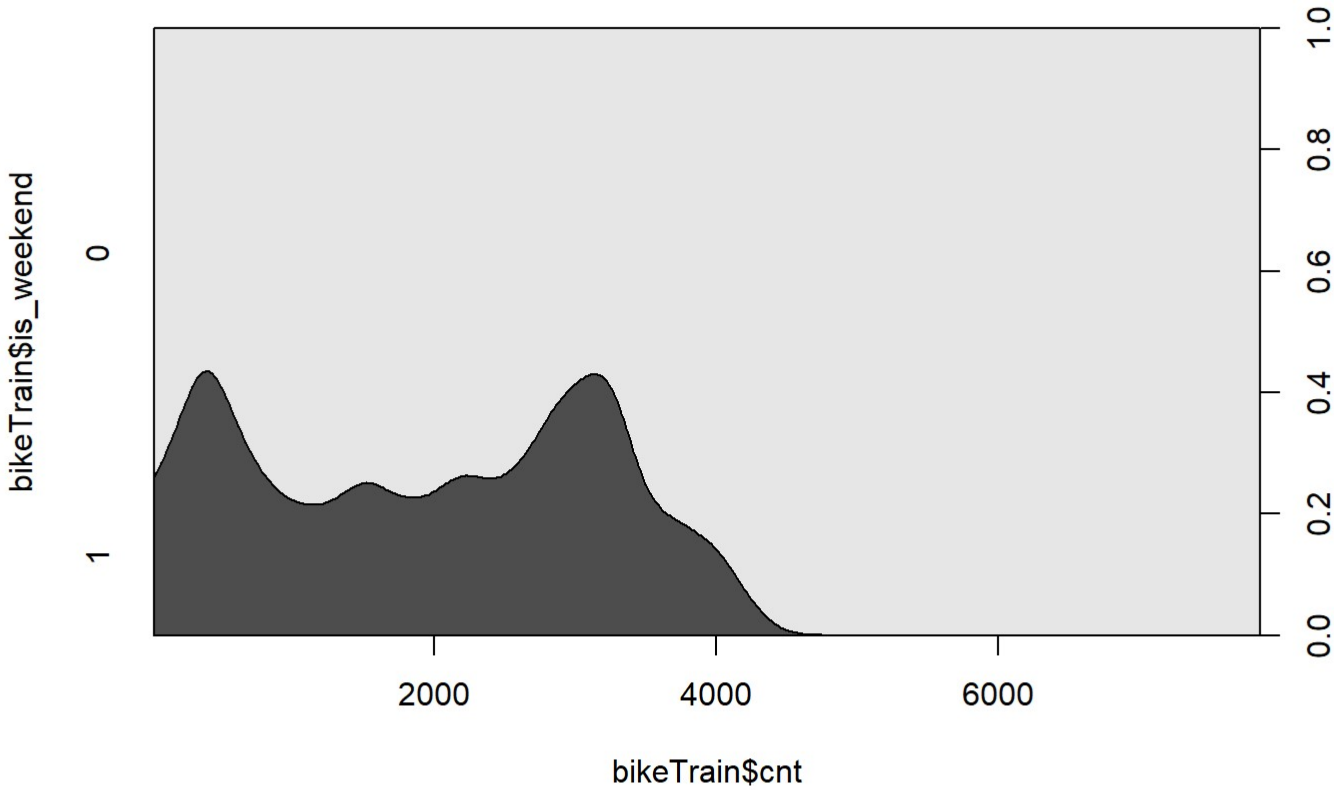
T2 and T1 have very similar plots with cnt and they both seem to have somewhat a linear shape to their plots.

```
plot(bikeTrain$weather_code, bikeTrain$cnt, xlab = "Weather Code", ylab = "Bike Rentals")
```



There seems to be more rentals when the weather is clear or cloudy. There seems to be a lot of outliers for all weather types except for thunderstorms.

```
cdplot(bikeTrain$cnt, bikeTrain$is_weekend)
```



It seems that there are majority of bike rentals are during the weekdays.

SVM Linear

```
library(e1071)

#SVM
linearBikeSVM <- svm(cnt ~ ., data = bikeTrain, kernel = "linear", scale = TRUE)
summary(linearBikeSVM)
```

```
##
## Call:
## svm(formula = cnt ~ ., data = bikeTrain, kernel = "linear", scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  1
##       gamma:  0.0625
##     epsilon:  0.1
##
##
## Number of Support Vectors:  8815
```

```
#Predict and RMSE
linearSVMPred <- predict(linearBikeSVM, newdata = bikeTest)

rmse <- mean((linearSVMPred - bikeTest$cnt)^2)
print(paste("RMSE = ", rmse))
```

```
## [1] "RMSE =  884254.417275349"
```

```
#Get best cost
linearTune <- tune(svm, cnt ~ ., data = bikeVal, kernel = "linear", ranges = list(cost = c(.0
01, .01, .1, 1, 5, 10)))

#Predict and RMSE
linearSVMPredTuned <- predict(linearTune$best.model, newdata = bikeTest)

rmse <- mean((linearSVMPredTuned - bikeTest$cnt)^2)
print(paste("RMSE = ", rmse))
```

```
## [1] "RMSE =  877378.574588575"
```

SVM Polynomial

```
#SVM
polyBikeSVM <- svm(cnt ~ ., data = bikeTrain, kernel = "polynomial", scale = TRUE)
summary(polyBikeSVM)
```

```
##
## Call:
## svm(formula = cnt ~ ., data = bikeTrain, kernel = "polynomial", scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##       gamma:  0.0625
##      coef.0:  0
##     epsilon:  0.1
##
##
## Number of Support Vectors:  8546
```

```
#Predict and RMSE
polySVMPred <- predict(polyBikeSVM, newdata = bikeTest)

rmse <- mean((polySVMPred - bikeTest$cnt)^2)
print(paste("RMSE = ", rmse))
```

```
## [1] "RMSE =  847140.39982906"
```

```
polyTune <- tune(svm, cnt ~ ., data = bikeVal, kernel = "polynomial", ranges = list(cost = c
(.001, .01, .1, 1, 5, 10)))

#Predict and RMSE
polySVMPredTuned <- predict(polyTune$best.model, newdata = bikeTest)

rmse <- mean((polySVMPredTuned - bikeTest$cnt)^2)
print(paste("RMSE = ", rmse))
```

```
## [1] "RMSE =  818610.782797605"
```

SVM Radial

```
#SVM
radialBikeSVM <- svm(cnt ~ ., data = bikeTrain, kernel = "radial", scale = TRUE)
summary(radialBikeSVM)
```

```
##
## Call:
## svm(formula = cnt ~ ., data = bikeTrain, kernel = "radial", scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.0625
##     epsilon:  0.1
##
##
## Number of Support Vectors:  8444
```

```
#Predict and RMSE
radialSVMPred <- predict(radialBikeSVM, newdata = bikeTest)

rmse <- mean((radialSVMPred - bikeTest$cnt)^2)
print(paste("RMSE = ", rmse))
```

```
## [1] "RMSE =  822889.597421451"
```

```
radialTune <- tune(svm, cnt ~ ., data = bikeVal, kernel = "radial", ranges = list(cost = c(.0
01, .01, .1, 1, 5, 10)))

#Predict and RMSE
radialSVMPredTuned <- predict(radialTune$best.model, newdata = bikeTest)

rmse <- mean((radialSVMPredTuned - bikeTest$cnt)^2)
print(paste("RMSE = ", rmse))
```

```
## [1] "RMSE =  805716.640765122"
```

Conclusion

Radial seems to be the better kernel due to the lower rmse than all the other kernels. I think this is due the data not really being linear or polynomial, so radial seems to be the best fit for the hyperplane. But, I really don’t think each kernel made much of a difference. The tuning for each kernel only improved the rmse by a bit, while taking a long time to compute.