# Dr. M. Baron, Statistical Machine Learning class, STAT-427/627

# KNN – K-nearest neighbor classification

# Data – miles per gallon and other variables from the Auto data set.

```
> library(ISLR)                          # This library contains datasets from our textbook (ISLR = name of our text)
> attach(Auto)
> names(Auto)                            # List of variables in this dataset
[1] "mpg"      "cylinders"    "displacement" "horsepower"   "weight"      "acceleration"    "year"      "origin"      "name"
> summary(mpg)                           # Economy rating will be defined based on miles per gallon
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  9.00   17.00   22.75   23.45   29.00   46.60
                                         # Initiate a fuel consumption rating variable that will be treated as categorical
> Economy = rep("Gas consumption", length(mpg))
> Economy[mpg <= 17] = "Heavy"
> Economy[mpg > 17 & mpg <= 22.75] = "OK"
> Economy[mpg > 22.75 & mpg <= 29] = "Eco"
> Economy[mpg > 29] = "Excellent"

> table(Economy)                                     # We used sample quartiles of variable mpg to define these ratings,
Economy                                              # that's why we got four approximately equal groups.
      Eco Excellent     Heavy        OK              # Now, we'll derive a classification rule, using other car characteristics
      101        95        99        97
```

# Prepare training and testing data, predictors (X) and responses (Y)

```
> n = length(mpg)
> Z = sample(n, n/2)                                 # We'll split data at random

> Auto.training = Auto[Z, ]                          # Subsample with indices from that subsample Z
> Auto.testing = Auto[-Z, ]                          # Notice the "minus" sign to denote all indices except those in Z
> dim(Auto)
[1] 392   9
> dim(Auto.training)
[1] 196   9
> dim(Auto.testing)
[1] 196   9

> names(Auto)
[1] "mpg" "cylinders" "displacement" "horsepower" "weight" "acceleration" "year" "origin" "name"
```

# KNN in R requires 4 inputs: training X, testing X, training Y, and K.

```
> X.training = Auto.training[ , 2:7 ]        # Take columns (variables) 2-7. That's from cylinders to year.
> X.testing = Auto.testing[ , 2:7 ]
> Y.training = Economy[ Z ]
> Y.testing = Economy[ -Z ]
```

```
# KNN tool is in the package “class”.

> library(class)
> knn.result = knn( X.training, X.testing, Y.training, 3 )

> table( Y.testing, knn.result )
           knn.result
Y.testing   Eco Excellent Heavy OK
  Eco        19        17     1 11
  Excellent   9        35     0  2
  Heavy       0         0    38  8
  OK          5         3     5 32

> mean( Y.testing == knn.result )
[1] 0.6702703                    # 67% correct classification rate with K=3. Is there a better K?
                                 # We’ll check all K from 1 to 20.
> class.rate = rep(0,20)         # Create a vector of length 20 and fill it with classification rates,
                                 # computed in a do-loop
> for (K in 1:20) {
+ knn.result = knn( X.training, X.testing, Y.training, K )
+ class.rate[K]=mean( Y.testing == knn.result )
+ }

> class.rate
# Apparently, K=6 and K=8 provide a slightly better prediction although still not as good as LDA

[1] 0.6378378 0.6378378 0.6702703 0.6810811 0.6810811 0.6918919 0.6702703
[8] 0.6918919 0.6648649 0.6648649 0.6594595 0.6594595 0.6756757 0.6702703
[15] 0.6702703 0.6864865 0.6702703 0.6702703 0.6702703 0.6702703
```