K Nearest Neighbor

```
library(ISLR)

## Warning: package 'ISLR' was built under R version 3.3.2

str(Caravan)

## 'data.frame':    5822 obs. of  86 variables:
##  $ MOSTYPE : num  33 37 37 9 40 23 39 33 33 11 ...
##  $ MAANTHUI: num  1 1 1 1 1 1 2 1 1 2 ...
##  $ MGEMOMV : num  3 2 2 3 4 2 3 2 2 3 ...
##  $ MGEMLEEF: num  2 2 2 3 2 1 2 3 4 3 ...
##  $ MOSHOOFD: num  8 8 8 3 10 5 9 8 8 3 ...
##  $ MGODRK  : num  0 1 0 2 1 0 2 0 0 3 ...
##  $ MGODPR  : num  5 4 4 3 4 5 2 7 1 5 ...
##  $ MGODOV  : num  1 1 2 2 1 0 0 0 3 0 ...
##  $ MGODGE  : num  3 4 4 4 4 5 5 2 6 2 ...
##  $ MRELGE  : num  7 6 3 5 7 0 7 7 6 7 ...
##  $ MRELSA  : num  0 2 2 2 1 6 2 2 0 0 ...
##  $ MRELOV  : num  2 2 4 2 2 3 0 0 3 2 ...
##  $ MFALLEEN: num  1 0 4 2 2 3 0 0 3 2 ...
##  $ MFGEKIND: num  2 4 4 3 4 5 3 5 3 2 ...
##  $ MFWEKIND: num  6 5 2 4 4 2 6 4 3 6 ...
##  $ MOPLHOOG: num  1 0 0 3 5 0 0 0 0 0 ...
##  $ MOPLMIDD: num  2 5 5 4 4 5 4 3 1 4 ...
##  $ MOPLLAAG: num  7 4 4 2 0 4 5 6 8 5 ...
##  $ MBERHOOG: num  1 0 0 4 0 2 0 2 1 2 ...
##  $ MBERZELF: num  0 0 0 0 5 0 0 0 1 0 ...
##  $ MBERBOER: num  1 0 0 0 4 0 0 0 0 0 ...
##  $ MBERMIDD: num  2 5 7 3 0 4 4 2 1 3 ...
##  $ MBERARBG: num  5 0 0 1 0 2 1 5 8 3 ...
##  $ MBERARBO: num  2 4 2 2 0 2 5 2 1 3 ...
##  $ MSKA    : num  1 0 0 3 9 2 0 2 1 1 ...
##  $ MSKB1   : num  1 2 5 2 0 2 1 1 1 2 ...
##  $ MSKB2   : num  2 3 0 1 0 2 4 2 0 1 ...
##  $ MSKC    : num  6 5 4 4 0 4 5 5 8 4 ...
##  $ MSKD    : num  1 0 0 0 0 2 0 2 1 2 ...
##  $ MHHUUR  : num  1 2 7 5 4 9 6 0 9 0 ...
##  $ MHKOOP  : num  8 7 2 4 5 0 3 9 0 9 ...
##  $ MAUT1   : num  8 7 7 9 6 5 8 4 5 6 ...
##  $ MAUT2   : num  0 1 0 0 2 3 0 4 2 1 ...
##  $ MAUT0   : num  1 2 2 0 1 3 1 2 3 2 ...
##  $ MZFONDS : num  8 6 9 7 5 9 9 6 7 6 ...
##  $ MZPART  : num  1 3 0 2 4 0 0 3 2 3 ...
##  $ MINKM30 : num  0 2 4 1 0 5 4 2 7 2 ...
##  $ MINK3045: num  4 0 5 5 0 2 3 5 2 3 ...
##  $ MINK4575: num  5 5 0 3 9 3 3 3 1 3 ...
##  $ MINK7512: num  0 2 0 0 0 0 0 0 0 1 ...
##  $ MINK123M: num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ MINKGEM : num  4 5 3 4 6 3 3 3 2 4 ...
##  $ MKOOPKLA: num  3 4 4 4 3 3 5 3 3 7 ...
##  $ PWAPART : num  0 2 2 0 0 0 0 0 0 2 ...
##  $ PWABEDR : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PWALAND : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PPERSAUT: num  6 0 6 6 0 6 6 0 5 0 ...
##  $ PBESAUT : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PMOTSCO : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PVRAAUT : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PAANHANG: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PTRACTOR: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PWERKT  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PBROM   : num  0 0 0 0 0 0 0 3 0 0 ...
##  $ PLEVEN  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PPERSONG: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PGEZONG : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PWAOREG : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PBRAND  : num  5 2 2 2 6 0 0 0 0 3 ...
##  $ PZEILPL : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PPLEZIER: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PFIETS  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PINBOED : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PBYSTAND: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AWAPART : num  0 2 1 0 0 0 0 0 0 1 ...
##  $ AWABEDR : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AWALAND : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ APERSAUT: num  1 0 1 1 0 1 1 0 1 0 ...
##  $ ABESAUT : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMOTSCO : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AVRAAUT : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AAANHANG: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ATRACTOR: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AWERKT  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ABROM   : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ ALEVEN  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ APERSONG: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AGEZONG : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AWAOREG : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ABRAND  : num  1 1 1 1 1 0 0 0 0 1 ...
##  $ AZEILPL : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ APLEZIER: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AFIETS  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AINBOED : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ABYSTAND: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Purchase: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(Caravan$Purchase)
```

```
##   No  Yes
## 5474  348
```

```
any(is.na(Caravan))
```

```
## [1] FALSE
```

```
var(Caravan[,1])
```

```
## [1] 165.0378
```

```
var(Caravan[,2])
```

```
## [1] 0.1647078
```

```
purchase <- Caravan[,86]
```

Standardize Dataset in R

```
standardized.Caravan <- scale(Caravan[,-86])

print(var(standardized.Caravan[,1]))
```

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

```
print(var(standardized.Caravan[,2]))
```

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

Test

```
test.index <- 1:1000
test.data <- standardized.Caravan[test.index,]
test.purchase <- purchase[test.index]
```

Train

```
train.data <- standardized.Caravan[-test.index,]
train.purchase <- purchase[-test.index]
```

KNN Model

```
library(class)

set.seed(101)

predicted.purchase <- knn(train.data,test.data,train.purchase,k=1)

print(head(predicted.purchase))
```

```
## [1] No No No No No No
## Levels: No Yes
```

Using Different K value Where k=3

```
predicted.purchase <- knn(train.data,test.data,train.purchase,k=3)
mean(test.purchase != predicted.purchase)

## [1] 0.073
```

k=5

```
predicted.purchase <- knn(train.data,test.data,train.purchase,k=5)
mean(test.purchase != predicted.purchase)

## [1] 0.066
```

Null vs. NA

```
predicted.purchase = NULL
error.rate = NULL

for(i in 1:20){
  set.seed(101)
  predicted.purchase = knn(train.data,test.data,train.purchase,k=i)
  error.rate[i] = mean(test.purchase != predicted.purchase)
}

print(error.rate)

##  [1] 0.116 0.107 0.074 0.070 0.066 0.064 0.062 0.061 0.058 0.058
0.059
## [12] 0.058 0.059 0.059 0.059 0.059 0.059 0.059 0.059 0.059
```

Elbow Method

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.2

k.values <- 1:20

error.df <- data.frame(error.rate,k.values)

error.df

##    error.rate k.values
## 1       0.116        1
## 2       0.107        2
## 3       0.074        3
## 4       0.070        4
## 5       0.066        5
## 6       0.064        6
## 7       0.062        7
```

```
## 8       0.061         8
## 9       0.058         9
## 10      0.058        10
## 11      0.059        11
## 12      0.058        12
## 13      0.059        13
## 14      0.059        14
## 15      0.059        15
## 16      0.059        16
## 17      0.059        17
## 18      0.059        18
## 19      0.059        19
## 20      0.059        20
```

Determining Misclassification

```
ggplot(error.df,aes(x=k.values,y=error.rate)) + geom_point()+
geom_line(lty="dotted",color='red')
```