Assign1_Small

R Markdown

```
#Libraries
library(caret)
library(class)
library(gmodels)
```

Reading and Analyzing the Data

```
df <- read.csv("/Users/Nick/Desktop/wisc_bc_data.csv", stringsAsFactors=FALSE
)
str(df)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ id                      : int  842302 842517 84300903 84348301 84358402
843786 844359 84458202 844981 84501001 ...
##  $ diagnosis               : chr  "M" "M" "M" "M" ...
##  $ radius_mean             : num  18 20.6 19.7 11.4 20.3 ...
##  $ texture_mean            : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ perimeter_mean          : num  122.8 132.9 130 77.6 135.1 ...
##  $ area_mean               : num  1001 1326 1203 386 1297 ...
##  $ smoothness_mean         : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ compactness_mean        : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ concavity_mean          : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ concave.points_mean     : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ symmetry_mean           : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ radius_se               : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ texture_se              : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ perimeter_se            : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ area_se                 : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_se           : num  0.0064 0.00522 0.00615 0.00911 0.01149 ..
.
##  $ compactness_se          : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ concavity_se            : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ concave.points_se       : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ symmetry_se             : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ fractal_dimension_se    : num  0.00619 0.00353 0.00457 0.00921 0.00511 .
..
##  $ radius_worst            : num  25.4 25 23.6 14.9 22.5 ...
##  $ texture_worst           : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter_worst         : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ area_worst              : num  2019 1956 1709 568 1575 ...
##  $ smoothness_worst        : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ compactness_worst       : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ concavity_worst         : num  0.712 0.242 0.45 0.687 0.4 ...
```

```
##  $ concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

```r
# Dropping ID Column
df <- df[-1]

# Number of each occurrences of Benign and Malignant
table(df$diagnosis)
```

```
##
##   B   M
## 357 212
```

```r
# Probability of Benign and Malignant
round(prop.table(table(df$diagnosis)) * 100, digits = 1)
```

```
##
##    B    M
## 62.7 37.3
```

```r
# Recoding Benign and Malignant to 0 and 1
diagnosis <- factor(df$diagnosis, levels = c("B", "M"), labels = c("0", "1"))

# Normalizing the numeric values
summary(df[c("radius_mean", "area_mean", "smoothness_mean")])
```

```
##   radius_mean       area_mean      smoothness_mean
##  Min.   : 6.981   Min.   : 143.5   Min.   :0.05263
##  1st Qu.:11.700   1st Qu.: 420.3   1st Qu.:0.08637
##  Median :13.370   Median : 551.1   Median :0.09587
##  Mean   :14.127   Mean   : 654.9   Mean   :0.09636
##  3rd Qu.:15.780   3rd Qu.: 782.7   3rd Qu.:0.10530
##  Max.   :28.110   Max.   :2501.0   Max.   :0.16340
```

```r
normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x)))}
df <- as.data.frame(lapply(df[2:31], normalize))
df <- cbind(diagnosis, df)

# Training and Testing Model 70% and 30% with Full DataSet
df_train <- df[1:398, ]
df_test <- df[399:569, ]
df_train_labels <- df[1:398, 1]
df_test_labels <- df[399:569, 1]
```

 # k-NN Algorithm

```
sqrt(569)
```

```
## [1] 23.85372
```

```
prediction <- knn(train = df_train, test = df_test, cl = df_train_labels, k=20)
CrossTable(x = df_test_labels, y = prediction, prop.chisq=F)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  171
##
##
##                | prediction
## df_test_labels |         0 |         1 | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |       132 |         0 |       132 |
##                |     1.000 |     0.000 |     0.772 |
##                |     1.000 |     0.000 |           |
##                |     0.772 |     0.000 |           |
## ---------------|-----------|-----------|-----------|
##              1 |         0 |        39 |        39 |
##                |     0.000 |     1.000 |     0.228 |
##                |     0.000 |     1.000 |           |
##                |     0.000 |     0.228 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |       132 |        39 |       171 |
##                |     0.772 |     0.228 |           |
## ---------------|-----------|-----------|-----------|
```

Analyzing the first Cross table with a K value of 20, we can observe that 132 out of 171 cases where the mass is not cancerous. This is the True Negative. The True Positive is 39 out of 171 cases. There are no false Negatives or false positives for this K value.

```
# Evaluate Model Performance with a different K value
# Using Scale() function
df_p <- as.data.frame(scale(df[-1]))
```

```
df_p <- as.data.frame(scale(df_p[-31]))


df_train <- df_p[1:398, ]
df_test <- df_p[399:569, ]
df_train_labels <- df[1:398, 1]
df_test_labels <- df[399:569, 1]
prediction3 <- knn(train = df_train, test = df_test, cl = df_train_labels, k=
9)
CrossTable(x = df_test_labels, y = prediction3, prop.chisq=F)

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  171
##
##
##                | prediction3
## df_test_labels |         0 |         1 | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |       131 |         1 |       132 |
##                |     0.992 |     0.008 |     0.772 |
##                |     0.985 |     0.026 |           |
##                |     0.766 |     0.006 |           |
## ---------------|-----------|-----------|-----------|
##              1 |         2 |        37 |        39 |
##                |     0.051 |     0.949 |     0.228 |
##                |     0.015 |     0.974 |           |
##                |     0.012 |     0.216 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |       133 |        38 |       171 |
##                |     0.778 |     0.222 |           |
## ---------------|-----------|-----------|-----------|
```

Analyzing the first Cross table with a K value of 9, we can observe that 131 out of 171 cases where the mass is not cancerous. This is the True Negative. The True Positive is 39 out of 171 cases. There are 2 False Negatives and 1 False Positive. Showing there is in increase in error. This introduced false negatives which brings the accuracy to (131+37)/171 = .98% accuracy

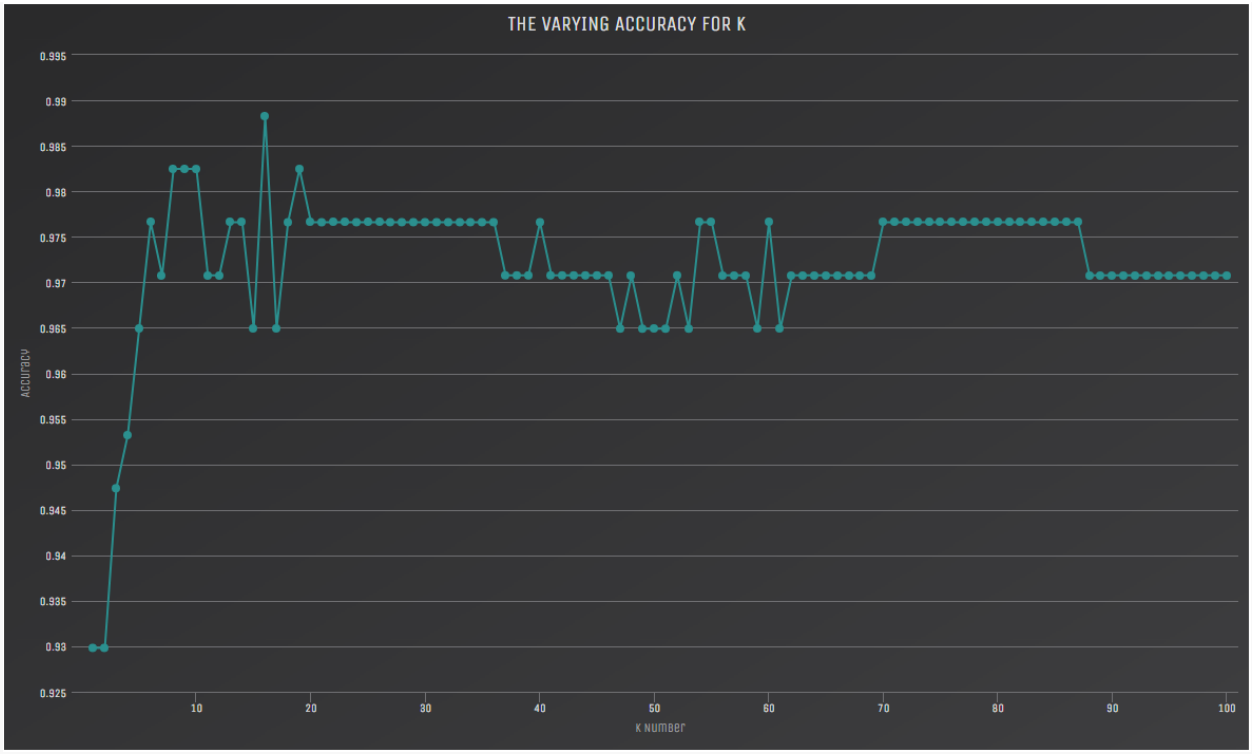
This will test varying K's from 1 to 100 and provide the best K value.

```
accuracy <- numeric()

for(i in 1:100){
    predict <- knn(train=df_train, test=df_test, cl=df_train_labels, k=i, pro
b=T)
    accuracy <- c(accuracy,mean(predict==df_test_labels))
}

acc <- data.frame(k= seq(1,100), cnt = accuracy)

opt_k <- subset(acc, cnt==max(cnt))[1,]
sub <- paste("The best K = ", opt_k$k, "(Accuracy =", opt_k$cnt,")")
sub
```

"The best K = 16 (Accuracy = 0.988304093567251)

```
library(highcharter)
hchart(acc, 'line', hcaes(k, cnt)) %>%
  hc_title(text = "The varying Accuracy for K") %>%
  hc_add_theme(hc_theme_darkunica()) %>%
  hc_xAxis(title = list(text = "K Number")) %>%
  hc_yAxis(title = list(text = "Accuracy"))
```

A visual graph of K values from 1 to 100.



Logistic Regression

```
# Logistic Regression
df2 <- read.csv("/Users/Nick/Desktop/wisc_bc_data.csv", stringsAsFactors=FALSE)

# Dropping ID Column
df2 <- df2[-1]

# Recoding Benign and Malignant to 0 and 1
diagnosis <- factor(df2$diagnosis, levels = c("B", "M"), labels = c("0", "1"))

df2_s <- cbind(diagnosis, df2)
df2_s <- df2_s[-2]

# Creates a New Data set with the removal of al predictors
df2_srp <-subset(df2_s, select=-c(area_mean,radius_mean,area_worst,compactness_mean,perimeter_worst,compactness_se,concavity_worst,fractal_dimension_worst))

set.seed(1234)
# Full Data Set
```

```r
train <- sample(nrow(df2_s), 0.7*nrow(df2_s))
df2_s.train <- df2_s[train,]
df2_s.test <- df2_s[-train,]

# Data Set with removed predictors

train2 <- sample(nrow(df2_srp), 0.7*nrow(df2_srp))
df2_srp.train <- df2_srp[train,]
df2_srp.test <- df2_srp[-train,]
```

Logistic Regression of Full Data Set

```r
fit.logit <- glm(diagnosis~., data=df2_s.train, family = binomial())

summary(fit.logit)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial(), data = df2_s.train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.241e-04  -2.100e-08  -2.100e-08   2.100e-08   1.223e-04
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.646e+03  5.805e+05  -0.003    0.998
## radius_mean             -3.853e+02  2.172e+05  -0.002    0.999
## texture_mean            -2.408e+00  4.286e+03  -0.001    1.000
## perimeter_mean           3.680e+01  3.818e+04   0.001    0.999
## area_mean                1.154e+00  8.384e+02   0.001    0.999
## smoothness_mean          5.342e+02  9.319e+05   0.001    1.000
## compactness_mean        -4.233e+03  1.021e+06  -0.004    0.997
## concavity_mean          -1.384e+03  1.405e+06  -0.001    0.999
## concave.points_mean      4.587e+03  1.388e+06   0.003    0.997
## symmetry_mean           -5.396e+02  4.466e+05  -0.001    0.999
## fractal_dimension_mean   1.782e+04  4.403e+06   0.004    0.997
## radius_se                8.157e+01  6.551e+05   0.000    1.000
## texture_se              -1.314e+00  2.587e+04   0.000    1.000
## perimeter_se            -2.078e+01  3.421e+04  -0.001    1.000
## area_se                  3.361e+00  6.849e+03   0.000    1.000
## smoothness_se            7.702e+03  4.226e+06   0.002    0.999
## compactness_se          -2.658e+03  2.405e+06  -0.001    0.999
## concavity_se             2.481e+03  1.682e+06   0.001    0.999
## concave.points_se        1.564e+04  3.433e+06   0.005    0.996
## symmetry_se              4.340e+02  1.710e+06   0.000    1.000
## fractal_dimension_se    -6.080e+04  1.450e+07  -0.004    0.997
## radius_worst             1.539e+02  6.016e+04   0.003    0.998
```

```
## texture_worst             8.152e+00  3.616e+03   0.002    0.998
## perimeter_worst          -2.131e+00  7.067e+03   0.000    1.000
## area_worst               -7.715e-01  5.097e+02  -0.002    0.999
## smoothness_worst         -1.986e+03  1.069e+06  -0.002    0.999
## compactness_worst         6.324e+02  2.884e+05   0.002    0.998
## concavity_worst           2.093e+02  2.366e+05   0.001    0.999
## concave.points_worst     -6.824e+02  6.664e+05  -0.001    0.999
## symmetry_worst            1.020e+03  3.459e+05   0.003    0.998
## fractal_dimension_worst   2.205e+03  1.764e+06   0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5.3122e+02  on 397  degrees of freedom
## Residual deviance: 1.7384e-07  on 367  degrees of freedom
## AIC: 62
##
## Number of Fisher Scoring iterations: 25
```

```
prob <- predict(fit.logit, df2_s.test, type="response")
logit.pred <- factor(prob > .5, levels=c(FALSE, TRUE), labels = c("benign","m
alignant"))

logit.perf <- table(df2_s.test$diagnosis, logit.pred, dnn = c("Actual", "Pred
icted") )

accuracy <- table(logit.pred, df2_s.test$diagnosis)
sum(diag(accuracy))/sum(accuracy)
```

The accuracy of the full logistic regression is .91%

```
## [1] 0.9181287
```

 # Logistic Regression with removal of predictors

```
fit.logit2 <- glm(diagnosis~., data=df2_srp.train, family = binomial())

summary(fit.logit2)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial(), data = df2_srp.train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -9.537e-04  -2.000e-08  -2.000e-08   2.000e-08   1.114e-03
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -7.436e+03  9.636e+05  -0.008    0.994
```

```
## texture_mean             9.592e+01  3.481e+04   0.003    0.998
## perimeter_mean          -1.206e+01  2.019e+04  -0.001    1.000
## smoothness_mean         -1.911e+04  8.015e+06  -0.002    0.998
## concavity_mean          -2.992e+03  3.287e+06  -0.001    0.999
## concave.points_mean      5.996e+03  1.603e+07   0.000    1.000
## symmetry_mean           -1.598e+04  2.977e+06  -0.005    0.996
## fractal_dimension_mean   6.711e+04  2.132e+06   0.031    0.975
## radius_se                7.221e+03  2.597e+05   0.028    0.978
## texture_se               1.487e+02  5.652e+04   0.003    0.998
## perimeter_se            -4.204e+02  2.936e+04  -0.014    0.989
## area_se                 -7.462e+00  2.325e+03  -0.003    0.997
## smoothness_se            7.561e+03  1.234e+06   0.006    0.995
## concavity_se             1.897e+04  2.531e+06   0.007    0.994
## concave.points_se       -3.189e+04  2.049e+07  -0.002    0.999
## symmetry_se             -2.290e+04  2.076e+07  -0.001    0.999
## fractal_dimension_se    -1.919e+05  2.354e+07  -0.008    0.993
## radius_worst             1.590e+02  5.915e+04   0.003    0.998
## texture_worst           -3.493e+01  1.932e+04  -0.002    0.999
## smoothness_worst         4.853e+03  2.119e+05   0.023    0.982
## compactness_worst       -2.828e+03  5.455e+05  -0.005    0.996
## concave.points_worst     1.991e+04  3.679e+06   0.005    0.996
## symmetry_worst           9.590e+03  3.818e+06   0.003    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5.3122e+02  on 397  degrees of freedom
## Residual deviance: 7.4007e-06  on 375  degrees of freedom
## AIC: 46
##
## Number of Fisher Scoring iterations: 25
```

```
prob2 <- predict(fit.logit2, df2_srp.test, type="response")
logit.pred2 <- factor(prob2 > .5, levels=c(FALSE, TRUE), labels = c("benign",
"malignant"))

logit.perf2 <- table(df2_srp.test$diagnosis, logit.pred2, dnn = c("Actual", "
Predicted") )

accuracy2 <- table(logit.pred2, df2_srp.test$diagnosis)
sum(diag(accuracy2))/sum(accuracy2)
```

The accuracy of the removal of predictors logistic regression is .89%

```
## [1] 0.8947368
```

Analysis of Logistic Regression: For both of the logistic regression the models did not converge and many of the variables were not of significance. This can give us a quick clue that this model will have less accuracy when compared to the KNN algorithm.

CONCLUSION: The K-NN Algorithm is more accurate in predicting if a mass is cancerous or benign. The KNN Algorithm has an accuracy of .988% compared to the .91% accuracy of logistic regression.