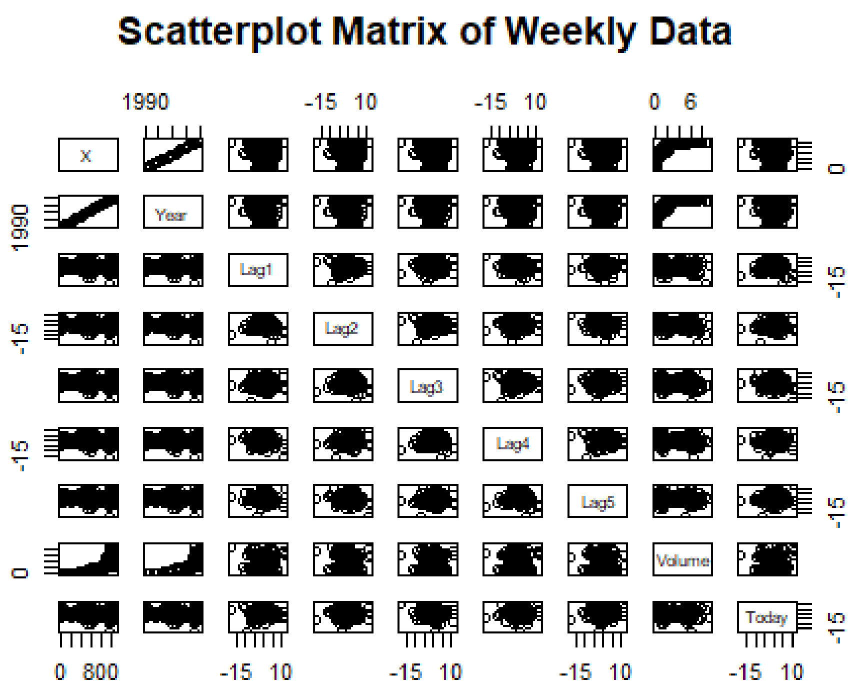
Assignment 30

John Bute

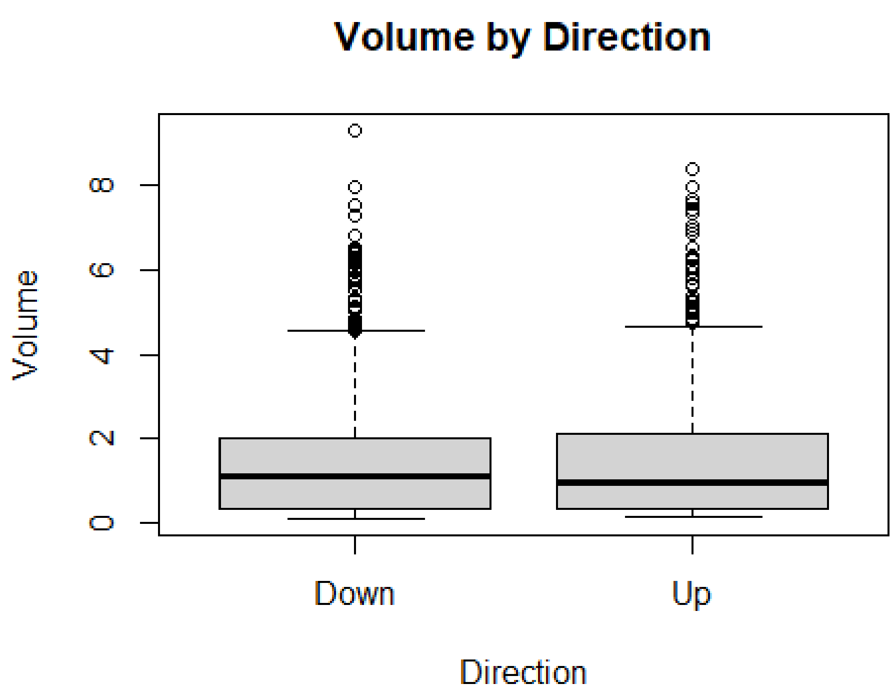
2024-11-08

Consider the Weekly data set. It contains 1,089 weekly stock market returns for 21 years, from the beginning of 1990 to the end of 2010. 1. Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

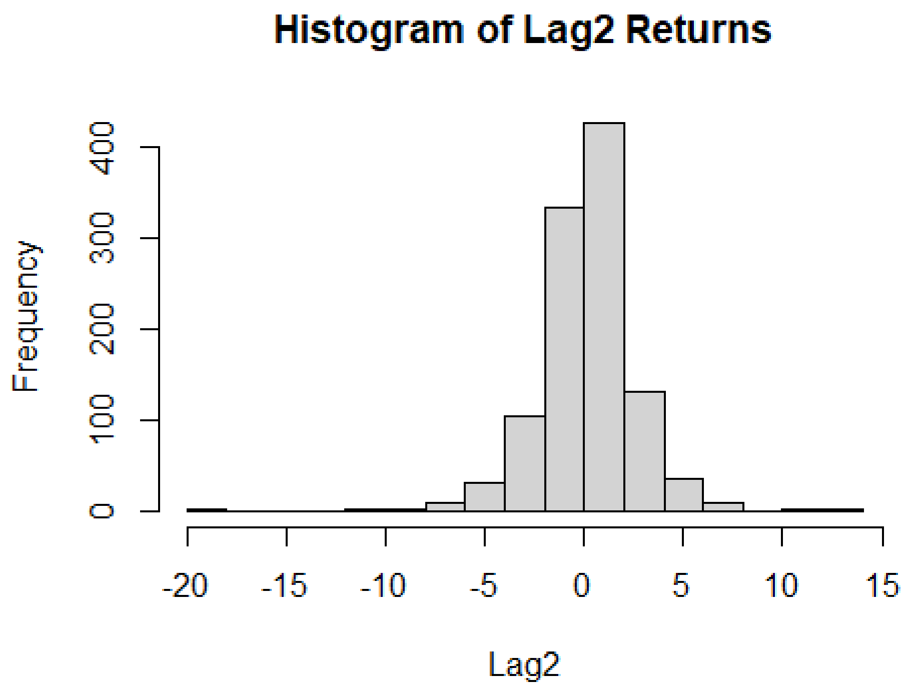
```
Weekly <- read.csv("C:/Users/johnb/Desktop/Machine Learning/data/Weekly.csv")

summary(Weekly)

##        X              Year           Lag1               Lag2
##  Min.   :   1   Min.   :1990   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.: 273   1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540
##  Median : 545   Median :2000   Median :  0.2410   Median :  0.2410
##  Mean   : 545   Mean   :2000   Mean   :  0.1506   Mean   :  0.1511
##  3rd Qu.: 817   3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090
##  Max.   :1089   Max.   :2010   Max.   : 12.0260   Max.   : 12.0260
##       Lag3               Lag4               Lag5              Volume
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747
##  1st Qu.: -1.1580   1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202
##  Median :  0.2410   Median :  0.2380   Median :  0.2340   Median :1.00268
##  Mean   :  0.1472   Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
##  3rd Qu.:  1.4090   3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
##  Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
##      Today             Direction
##  Min.   :-18.1950   Length:1089
##  1st Qu.: -1.1540   Class :character
##  Median :  0.2410   Mode  :character
##  Mean   :  0.1499
##  3rd Qu.:  1.4050
##  Max.   : 12.0260
pairs(Weekly[ , sapply(Weekly, is.numeric)], main = "Scatterplot Matrix of
Weekly Data")
```




```r
boxplot(Volume ~ Direction, data = Weekly, main = "Volume by Direction")
```




```r
hist(Weekly$Lag2, main = "Histogram of Lag2 Returns", xlab = "Lag2")
```



The lag variables all have similar ranges, while Direction is a categorical variable that corresponds to if the market moved up or down. Volume shows an upwards trend over time with Year, suggesting that the trading volume has increased throughout the years. Furthermore, we see that volume may not be the best predictor for market direction, as the boxplots are quite similar. Finally, lag2 has a bell-shaped distribution, which shows that returns are generally symmetric with occasional extreme values

2. Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
Weekly$Direction <- factor(Weekly$Direction, levels = c("Down", "Up"))
log_model_full <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                      data = Weekly,
                      family = binomial)

summary(log_model_full)

##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
```

```
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

We see that lag1, lag3, lag4, lag5, and volume are all statistically insignificant. The only one that is sinificant is lag2.

3. Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
pred_probs <- predict(log_model_full, type = "response")
pred_classes <- ifelse(pred_probs > 0.5, "Up", "Down")
conf_matrix <- table(Predicted = pred_classes, Actual = Weekly$Direction)
conf_matrix

##          Actual
## Predicted Down  Up
##      Down   54  48
##      Up    430 557

accuracy_full <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy_full

## [1] 0.5610652
```

We see that our confusion matrix shows that our model will often times predict up more likely than down, and so we see that although it did a good job at correctly predicting times when the stock market went up, it generated alot of false positives too, thus misclassifying instances of down with up.

4. Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
train <- Weekly$Year < 2009
test <- !train
log_model_train <- glm(Direction ~ Lag2, data = Weekly, family = binomial,
```

```r
subset = train)

pred_probs_test <- predict(log_model_train, Weekly[test, ], type =
"response")
pred_classes_test <- ifelse(pred_probs_test > 0.5, "Up", "Down")
conf_matrix_test <- table(Predicted = pred_classes_test, Actual =
Weekly$Direction[test])
conf_matrix_test

##          Actual
## Predicted Down Up
##      Down    9  5
##      Up     34 56

accuracy_test <- sum(diag(conf_matrix_test)) / sum(conf_matrix_test)
accuracy_test

## [1] 0.625
```

We seem to have the same issue, where our model is still stuck on predicting up, even when it is actually down. 5. Repeat 4. using LDA. (optional) 6. Repeat 4. using QDA. (optional) 7. Repeat 4. using kNN with k = 1.

```r
library(class)
train_X <- as.matrix(Weekly[train, "Lag2", drop = FALSE])
test_X <- as.matrix(Weekly[test, "Lag2", drop = FALSE])
train_Y <- Weekly$Direction[train]

knn_pred <- knn(train_X, test_X, train_Y, k = 1)
conf_matrix_knn <- table(Predicted = knn_pred, Actual =
Weekly$Direction[test])
conf_matrix_knn

##          Actual
## Predicted Down Up
##      Down   21 30
##      Up     22 31

accuracy_knn <- sum(diag(conf_matrix_knn)) / sum(conf_matrix_knn)
accuracy_knn

## [1] 0.5
```

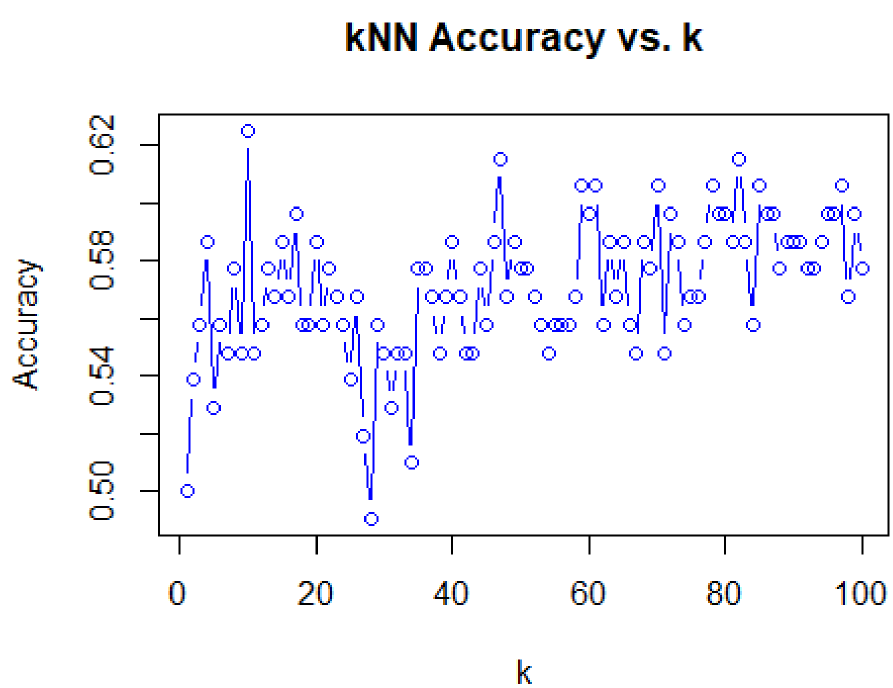
Overall we seem to perform worse, as our model is having a baseline accuracy of 53%

8. Which of these methods appears to provide the best results on this data? Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values of k in the kNN classifier.

Logistic regression performs better than1NN on the accuracy and true positive rates, but not on the true negative rate. The 1NN does better at this, but by sacrificing the true positive rate.

Let us see the accuracy of various KNN classifiers, for k = 1 to 100:

```
library(class)

train_X <- as.matrix(Weekly[train, "Lag2", drop = FALSE])
test_X <- as.matrix(Weekly[test, "Lag2", drop = FALSE])
train_Y <- Weekly$Direction[train]
test_Y <- Weekly$Direction[test]

b <- numeric(100)
for (j in 1:100) {
  knn_pred <- knn(train_X, test_X, train_Y, k = j)
    b[j] <- mean(knn_pred == test_Y)
}
summary(b)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4904  0.5577  0.5673  0.5708  0.5865  0.6250

plot(1:100, b, type = "b", xlab = "k", ylab = "Accuracy",
     main = "kNN Accuracy vs. k", col = "blue")
```



It seems that KNN is just not the right model, as all k = 1 to 100 perform worse than our logistic regression model.