Assignment4

Jiao Qu A20386614, Yuan-An Liu A20375099, Zhenyu Zhang A20287371

10/2/2017

Question 1

Is a problem in the data in the following table? (Hint: yes, there is.) If so, what is the problem and how would you "guess" to fix it? By "guess", I mean, what data / technique would you use?

Answer: Yes,there is. The price of 3-beds in the last column is unreasonable. Because it obviously lower than the price of 2-beds. The cause of this problem may be missing data. We can delete missing data samples, if the proportion of missing data is low. Or we can use the mean or median of the variable replace the missing value. It might cause Error due to Bias. The error due to bias is taken as the difference between the expected (or average) prediction of our model and the correct value which we are trying to predict.

Question 2

You need to decide whether adding more features to your model improves fit. With the Advertising data set, compute Sales ~ TV and Sales ~ TV + Radio + Newspaper using linear regression. What are the RMSEs and the Adjusted R-squares of the two models?

Answer: The RMSEs of "Sales ~ TV" is 3.242. The adjusted R-squares is 0.609.

The RMSEs of "Sales ~ TV + Radio + Newspaper" is 1.669. The adjusted R-squares is 0.896

```r
advertising = read.csv("~/Desktop/data/Advertising.csv", header = TRUE)
RMSE <- function(error) { sqrt(mean(error^2)) }
m1=lm(sales~TV,data=advertising)
RMSE(m1$residuals)
```

```
## [1] 3.242322
```

```r
summary(m1)
```

```
##
## Call:
## lm(formula = sales ~ TV, data = advertising)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

```r
m2=lm(sales~TV+radio+newspaper,data=advertising)
RMSE(m2$residuals)
```

```
## [1] 1.66857
```

```r
summary(m2)
```

```
##
## Call:
## lm(formula = sales ~ TV + radio + newspaper, data = advertising)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

Question 3

You also want to consider a different algorithm. Repeat question 1, but use KNN. Use Euclidean distance and compute fitted values by averaging the Sales values of the K nearest neighbors. Try all values of K from 1 to n and plot the RMSE of each K. Which K and which model (Sales ~ TV or Sales ~ TV + Radio + Newspaper) yields the lowest error? Does any configuration (i.e., any value of K over either set of predictors) outperform linear regression?

Answer: For the mode "Sales ~ TV", the lowest RMSE is 3.317. For the model "Sales ~ TV + Radio + Newspaper", the lowest RMSE is 1.366. The plor is showed as blow. And the outperform of Linear regression is better.

```r
library(ISLR)
library(class)
library(FNN)
```

```
##
## Attaching package: 'FNN'

## The following objects are masked from 'package:class':
##
##     knn, knn.cv
```

```r
library(hydroGOF)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
advertising = read.csv("~/Desktop/data/Advertising.csv", header = TRUE)
train.X= cbind(advertising$TV,advertising$radio,advertising$newspaper)
xlet = c(1:199)
aylet = array(1:199)
bylet = array(1:199)
LRMSEA = Inf
LRMSEB = Inf
for(i in 1:199){
  knn.fit=knn.reg(advertising$TV,NULL,advertising$sales,k=i)
  knn.fit1=knn.reg(train.X,NULL,advertising$sales,k=i)
  a=RMSE(knn.fit$residuals)
  aylet[i]=a
  b=RMSE(knn.fit1$residuals)
  bylet[i]=b
  if(LRMSEA>=a){
    LRMSEA = a
  }
  if(LRMSEB >=b){
    LRMSEB = b
  }
}
LRMSEA
```

```
## [1] 3.316689
```

```r
LRMSEB
```

```
## [1] 1.366358
```

```r
plot(xlet,aylet)
```

```r
plot(xlet,bylet)
```

Question 4

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

Answer: There are 1089 records in the weekly data set. This data set consists of 9 features of each record. * Year and Volume are positively correlated.
* Lag1 and Lag3 are positively correlated.
* Lag3 and Lag5 are positively correlated.
* Lag4 and Lag2 are positively correlated.
* Today and Lag2 are positively correlated.
* Today and Lag5 are positively correlated.

```r
library(ISLR)
names(Weekly)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

```r
dim(Weekly)
```

```
## [1] 1089    9
```

```r
summary(Weekly)
```

```
##       Year           Lag1               Lag2               Lag3
```

```
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202
##  Median :  0.2380   Median :  0.2340   Median :1.00268
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
##      Today          Direction
##  Min.   :-18.1950   Down:484
##  1st Qu.: -1.1540   Up  :605
##  Median :  0.2410
##  Mean   :  0.1499
##  3rd Qu.:  1.4050
##  Max.   : 12.0260
```

```r
cor(Weekly[,-9])
```

```
##               Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```

```r
pairs(Weekly)
```

(b) 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?

Answer: Predictor *Lag*2 is statistically significant.

```
glm.fit = glm(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data = Weekly,family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## 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
```

(c) 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.

Answer: The accuracy is 0.561. The accuracys of up or down prediction is 56.4%(up) or 52.9%(down). So the up prediction is higher than down prediction.

```r
glm.probs=predict(glm.fit,type="response")
glm.pred=rep("Down",1089)
glm.pred[glm.probs >.5]="Up"
table(glm.pred, Weekly$Direction)
```

```
##
## glm.pred Down  Up
##     Down   54  48
##     Up    430 557
```

```r
a = (54+557)/nrow(Weekly); a
```

```
## [1] 0.5610652
```

```r
up = 557/(430+557);down = 54 / (54 + 48)
up;down
```

```
## [1] 0.5643364
```

```
## [1] 0.5294118
```

(d) 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). ***I generate a training and test set, please use the first 80% of records as training and the rest as test.***

Answer: The accuracy is 0.55.

```r
train = Weekly[1:871,]
test = Weekly[872:1089,]
glm.fit1 = glm(Direction ~ Lag2,data = train,family = binomial)
summary(glm.fit1)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.379  -1.276   1.034   1.080   1.271
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.23049    0.06853   3.363  0.00077 ***
## Lag2         0.04049    0.03308   1.224  0.22087
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1195.3  on 870  degrees of freedom
## Residual deviance: 1193.7  on 869  degrees of freedom
## AIC: 1197.7
##
## Number of Fisher Scoring iterations: 4
```

```r
fit1.probs=predict(glm.fit1,test,type="response")
fit1.pred=rep("Down",218)
fit1.pred[fit1.probs >.5]="Up"
table(fit1.pred, test$Direction)
```

```
##
## fit1.pred Down  Up
##      Down    5   3
##      Up     95 115
```

```r
mean(fit1.pred == test$Direction)
```

```
## [1] 0.5504587
```

(e) Repeat (d) using LDA.

Answer: The accuracy is 0.55.

```r
library(MASS)
lda.fit=lda(Direction ~ Lag2,data=train)
lda.pred=predict(lda.fit, test)
table(lda.pred$class,test$Direction)
```

```
##
##        Down  Up
##   Down    5   3
##   Up     95 115
```

```r
mean(lda.pred$class == test$Direction)
```

```
## [1] 0.5504587
```

(f) Repeat (d) using QDA.

Answer: The accuracy is 0.54.

```r
qda.fit=qda(Direction ~ Lag2,data=train)
qda.pred=predict(qda.fit, test)
table(qda.pred$class,test$Direction)
```

```
##
##        Down  Up
```

```
##   Down    0   0
##   Up    100 118
```

```r
mean(qda.pred$class == test$Direction)
```

```
## [1] 0.5412844
```

(g) Repeat (d) using KNN with K = 1.

Answer: The accuracy is 0.49.

```r
library(class)
train.X= as.matrix(train$Lag2)
test.X=as.matrix(test$Lag2)
set.seed (1)
knn.pred=knn(train.X,test.X,train$Direction ,k=1)
table(knn.pred, test$Direction)
```

```
##
## knn.pred Down Up
##     Down   48 59
##     Up     52 59
```

```r
mean(knn.pred == test$Direction)
```

```
## [1] 0.4908257
```

(h) Which of these methods appears to provide the best results on this data?

Answer: These four methods is performed. The accuracy of methods is 0.55, 0.55, 0.54, 0.49 respectively. So the logistics regression and LDA methods provide the best results on this data.

(i) 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 for K in the KNN classifier.

Answer: 1 logistics regression: the combinations of predictors is Lag2 and volume. the confusion matrix is showed as blow. And the accuracy is 0.559. This accuracy is a little bit higher than single predictors lag2 that is showed as (d) question.

```r
glm.fit1 = glm(Direction ~ Lag2 + Volume,data = train,family = binomial)
fit1.probs=predict(glm.fit1,test,type="response")
fit1.pred=rep("Down",218)
fit1.pred[fit1.probs >.5]="Up"
table(fit1.pred, test$Direction)
```

```
##
## fit1.pred Down Up
##      Down   29 25
##      Up     71 93
```

```r
mean(fit1.pred == test$Direction)
```

```
## [1] 0.559633
```

2 LDA: the combinations of predictors is Lag2 and volume. the confusion matrix is showed as blow. And the accuracy is 0.564. This accuracy is a little bit higher than single predictors lag2 that is showed as (e) question.

```
lda.fit=lda(Direction ~ Lag2 + Volume,data=train)
lda.pred=predict(lda.fit, test)
table(lda.pred$class,test$Direction)
```

```
##
##        Down Up
##   Down   30 25
##   Up     70 93
```

```
mean(lda.pred$class == test$Direction)
```

```
## [1] 0.5642202
```

3 QDA: the interaction of predictors is Lag2 and volume. the confusion matrix is showed as blow. And the accuracy is 0.536. This accuracy is a little bit lower than single predictors lag2 that is showed as (f) question.

```
qda.fit=qda(Direction ~ Lag2 * Volume,data=train)
qda.pred=predict(qda.fit, test)
table(qda.pred$class,test$Direction)
```

```
##
##        Down  Up
##   Down    0   1
##   Up    100 117
```

```
mean(qda.pred$class == test$Direction)
```

```
## [1] 0.5366972
```

4 KNN: We experiment with values for different Ks(5,9,19,29,39) in the KNN classifier with single predictors lag2. In the result come from the experiments, the K value equaled 19 result the best proformance among these. The accuracy is 0.583.

```
knn.pred=knn(train.X,test.X,train$Direction ,k=5)
table(knn.pred, test$Direction)
```

```
##
## knn.pred Down Up
##     Down   35 42
##     Up     65 76
```

```
mean(knn.pred == test$Direction)
```

```
## [1] 0.5091743
```

```
knn.pred=knn(train.X,test.X,train$Direction ,k=9)
table(knn.pred, test$Direction)
```

```
##
## knn.pred Down Up
##     Down   44 41
##     Up     56 77
```

```
mean(knn.pred == test$Direction)
```

```
## [1] 0.5550459
```

```r
knn.pred=knn(train.X,test.X,train$Direction ,k=19)
table(knn.pred, test$Direction)
```

```
##
## knn.pred Down Up
##     Down   41 31
##     Up     59 87
```

```r
mean(knn.pred == test$Direction)
```

```
## [1] 0.587156
```

```r
knn.pred=knn(train.X,test.X,train$Direction ,k=29)
table(knn.pred, test$Direction)
```

```
##
## knn.pred Down Up
##     Down   45 36
##     Up     55 82
```

```r
mean(knn.pred == test$Direction)
```

```
## [1] 0.5825688
```

```r
knn.pred=knn(train.X,test.X,train$Direction ,k=39)
table(knn.pred, test$Direction)
```

```
##
## knn.pred Down Up
##     Down   40 34
##     Up     60 84
```

```r
mean(knn.pred == test$Direction)
```

```
## [1] 0.5688073
```

Question 5

In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set. ###(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median() function. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.

```r
library(ISLR)
data("Auto")
mpg01 = rep(0,392)
mpg01[Auto$mpg > median(Auto$mpg)] = 1
newData = data.frame(Auto,mpg01)
summary(newData)
```

```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
```

```
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
##
##      weight      acceleration        year           origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :2804   Median :15.50   Median :76.00   Median :1.000
##  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
##
##                  name         mpg01
##  amc matador       :  5   Min.   :0.0
##  ford pinto        :  5   1st Qu.:0.0
##  toyota corolla    :  5   Median :0.5
##  amc gremlin       :  4   Mean   :0.5
##  amc hornet        :  4   3rd Qu.:1.0
##  chevrolet chevette:  4   Max.   :1.0
##  (Other)           :365
```

(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

Answer: From graphica shown as blow, the mpg directly associates the mpg01. and the displacement, horsepower, weight, and acceleration seem most likely to be useful in predicting mpg01.

```r
pairs(newData)
```

```r
stNewData = scale(newData[,-9])
boxplot(stNewData)
```

(c) Split the data into a training set and a test set.*I generate a training and test set, please use the first 80% of records as training and the rest as test.*

Answer: The total number of records is 392. 80 % records is 313

```r
train = newData[1:313,]
test = newData[314:392,]
```

(d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

Answer: The test error of the model obtained is 0.1139

```r
lda.fit=lda(mpg01 ~ displacement+horsepower+weight+acceleration,data=train)
lda.pred=predict(lda.fit, test)
table(lda.pred$class,test$mpg01)
```

```
##
##      0  1
##   0  5  9
##   1  0 65
```

```r
mean(lda.pred$class != test$mpg01)
```

```
## [1] 0.1139241
```

(e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

Answer: The test error of the model obtained is 0.1139

```r
qda.fit=qda(mpg01 ~ displacement+horsepower+weight+acceleration,data=train)
qda.pred=predict(qda.fit, test)
table(qda.pred$class,test$mpg01)
```

```
##
##      0  1
##   0  5  9
##   1  0 65
```

```r
mean(qda.pred$class != test$mpg01)
```

```
## [1] 0.1139241
```

(f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

Answer: The test error of the model obtained is 0.1772

```r
glm.fit1 = glm(mpg01 ~ displacement+horsepower+weight+acceleration,data = train,family = binomial)
fit1.probs=predict(glm.fit1,test,type="response")
fit1.pred=rep(0,79)
fit1.pred[fit1.probs >.5]= 1
table(fit1.pred, test$mpg01)
```

```
##
## fit1.pred  0  1
##         0  5 14
##         1  0 60
```

```r
mean(fit1.pred != test$mpg01)
```

```
## [1] 0.1772152
```

(g) Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

Answer: The test error of the model obtained is 0.2151 under the value of k is 1

```r
library(class)
train.X= cbind(train$displacement,train$horsepower,train$weight,train$acceleration)
test.X=cbind(test$displacement,test$horsepower,test$weight,test$acceleration)
set.seed (1)
knn.pred=knn(train.X,test.X,train$mpg01 ,k=1)
table(knn.pred, test$mpg01)
```

```
##
## knn.pred  0  1
##        0  5 17
##        1  0 57
```

```r
mean(knn.pred != test$mpg01)
```

```
## [1] 0.2151899
```

For the different k values, The test error of the model obtained is 0.25, 0.24, 0.20, 0.16, 0.18, 0.18 respectively. the K value equaled 29 result the best proformance among these. The test error is 0.164

```r
knn.pred=knn(train.X,test.X,train$mpg01 ,k=3)
table(knn.pred, test$mpg01)
```

```
##
## knn.pred  0  1
##        0  5 20
##        1  0 54
```

```r
mean(knn.pred != test$mpg01)
```

```
## [1] 0.2531646
```

```r
knn.pred=knn(train.X,test.X,train$mpg01 ,k=9)
table(knn.pred, test$mpg01)
```

```
##
## knn.pred  0  1
##        0  5 19
##        1  0 55
```

```r
mean(knn.pred != test$mpg01)
```

```
## [1] 0.2405063
```

```r
knn.pred=knn(train.X,test.X,train$mpg01 ,k=19)
table(knn.pred, test$mpg01)
```

```
##
## knn.pred  0  1
##        0  5 16
##        1  0 58
```

```r
mean(knn.pred != test$mpg01)
```

```
## [1] 0.2025316
```

```r
knn.pred=knn(train.X,test.X,train$mpg01 ,k=29)
table(knn.pred, test$mpg01)
```

```
##
## knn.pred  0  1
##        0  5 13
##        1  0 61
```

```r
mean(knn.pred != test$mpg01)
```

```
## [1] 0.164557
```

```r
knn.pred=knn(train.X,test.X,train$mpg01 ,k=39)
table(knn.pred, test$mpg01)
```

```
##
## knn.pred  0  1
##        0  5 15
##        1  0 59
```

```r
mean(knn.pred != test$mpg01)
```

```
## [1] 0.1898734
```

```r
knn.pred=knn(train.X,test.X,train$mpg01 ,k=49)
table(knn.pred, test$mpg01)
```

```
##
## knn.pred  0  1
##        0  5 15
##        1  0 59
```

```r
mean(knn.pred != test$mpg01)
```

```
## [1] 0.1898734
```

Question 6 : Read the paper and answer the questions. This is a seminal paper with over 1,000 citations (I think)! So, it's worth reading. Note that these questions are NOT specific to the paper, rather, they are general to the structure of all analyses. The questions below should be answerable in 2-3 sentences.

(1) What "business" problem are the authors trying to solve?

Answer: The question that authors metioned in the paper occors in on-line discussion groups and review sites. The characteristic of business problem is sentiment of overall towards the subject matter. The aurthor try to solve that whether a product review is positive or negative, and then labeling these product withe sentiment would provide succinct summaries to readers.

(2) What work has been done before in the area and how is this work different? (The related works section of research papers indicate how the work is novel, and also demonstrates that this is an actually "interesting" problem worth solving.)

Answer: In literature review, the pervious work focuse on categorizaion according to source and source style in statistivally approaches. And another pervious work determine genre of text, subjective genres and so fouth depending on finding the feature indicating. For sentiment classification, some previous work focuse on the partially knowledge-based or mutual information . In author's work, he utilize prior-knowledge-free methods to understand the task.

(3) What metrics do the authors use to measure performance? Include also whether there is a test and control set, what numeric metrics there are, etc. (The authors have to use a metric where, if you improve on it, you should ultimately make a better product and make more money for the company. If you don't think the metric / measurement methodology is good, please indicate.)

Answer: The metrics is accuracy that is percentage of documents classified correctly and tie rate that is percentage of documents where the two sentiments were rated equally likely.

(4) What data do the authors use for their experiments? (Note that the authors should use data that is representative of real world data. Hence, if they get results using the experimental data, the implication is the same solution would work in practice. If you have any comments on whether the data is valid, please mention it here.)

Answer:The author uses the movie-review dataset from the Internet Movie Database in practive.

(5) How do the authors pre-process their data?

Answer: The author creates a date set with uniform class distribution at first, then randomly selected 700 positive-sentiment and 700 negative-sentiment documents. He divides the 1400 documents into three part which each part maintain balanced class distributions. Author removes the rating indicator and extracted the textual information form data. And then clear opposite sentiment orientations words, finally add the tag NOT_ in specific position.

(6) What approach to solving the problem did the authors use?

Answer: Under the pro-process dateset, author use the naive Bayes, maximum entropy, support vector machines approaches to solve the problems

(7) What are the key parameters they vary in their experiments? (You don't just throw data at an algorithm. You usually have to apply an algorithm and somehow tune the performance of the algorithm as appropriate. Think of how we would tune alpha or the number of iterations in the gradient descent algorithm.)

Answer: In MaxEnt, the author use ten iterations of the improved iterative scaling algorithm for parameter training. In experiment, author focues on features based on unigram and bigrams.

(8) What is the baseline the authors use? (Performance of the proposed system should be better than some baseline that an average good engineer would develop or whatever is the current state of the art, obviously. If the authors can't beat the current systems, then the algorithm is generally not accepted for publishing. If the baseline is unreasonable, please describe why.)

Answer:The baselines the authors used are the experience for human-based classification based on the fuman word lists as Figure 1 showed. These two is created by independent CS grad students Positive vs. negative word count The third baseline that author mentioned in paper is 69% resulted by examination with entire corpus plus introspection. The third one might be difficult to beat.

(9) What are the authors' results?

Answer: The results produced via ML are quite good compared with human-generated baselines. For experiment, the Naive Bayes tends to do the worst and SVMs tend to do the best. Finally, author mentions a human easily detect the true sentiment of the review, but bag-of-features classifiers would presumably find these instances difficult.

(10) What do the authors indicate they can do to improve their results?

Answer: The authors indicate he want to investigate ways to improve sentiment categorization. And author find the identification of features indicating whether sentences are on-topic.

(11) Can you think of ways to improve this paper, either in its presentation, or its solution? Or, why did you like or dislike the paper?

Answer: This paper seeks to indentify the viewpoints underling a text span. I think extracting thest portions can be using the optimazation algorithm to finding the minimum cuts in the data. It will improve the performance of the algorithm mentioned by author.