ASSIGNMENT-7

Question-1)

```
#Question 1

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Train.csv")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Test.csv')

test$Age[is.na(test$Age)] = mean(test$Age, na.rm=TRUE)

train$Age[is.na(train$Age)] = mean(train$Age, na.rm=TRUE)

levels(train$Embarked)

levels(train$Embarked)[1] = 'S'

train$Embarked <- sapply(as.character(train$Embarked), switch, 'C' = 0, 'Q' = 1, 'S' = 2)

test$Embarked <- sapply(as.character(test$Embarked), switch, 'C' = 0, 'Q' = 1, 'S' = 2)

train$Sex <- ifelse(train$Sex == 'male', 1, 0)

test$Sex <- ifelse(test$Sex == 'male', 1, 0)

#fix(train)


#logistic regression

attach(train)

LR = glm( factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked), data= train, family= 'binomial')

LR.probs = predict(LR, type= 'response', newdata= test)

LR.probs

LR.probs = ifelse(LR.probs > 0.5, 1, 0)
```

```
table(LR.probs,test$Survived)

mean(LR.probs == test$Survived)


 #LDA

library(MASS)

LDA = lda(factor(Survived)~ factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked) , data= train)

LDA.pred = predict(LDA , newdata= test)

LDA.class = LDA.pred$class

LDA.class

table(LDA.class, test$Survived )

mean(LDA.class != test$Survived)


#QDA

library(MASS)

QDA = qda(factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch + factor(Embarked), data= train, family= binomial)

QDA.pred = predict(QDA , newdata= test)

QDA.class = QDA.pred$class

QDA.class

table(QDA.class, test$Survived )

mean(QDA.class != test$Survived)
```

```
.#KNN
library(FNN)
#train$Age <- ifelse(train$Age<18, 1, 0)
#test$Age <- ifelse(test$Age<18, 1, 0)
train$Age <- ifelse(train$Age<18, 1, 0)
test$Age <- ifelse(test$Age<18, 1, 0)
train1= train[, -c(1,2,4,9,10,11)]
test1= test[, -c(1,2,4,9,10,11)]
train.Y = train$Survived
train1
set.seed(12)
knn1 <- knn(train1, test1, train.Y, k=1)
table(knn1, test$Survived)
mean(knn1 != test$Survived)
knn2 = knn(train1, test1, train.Y, k=2)
table(knn2, test$Survived)
mean(knn2 != test$Survived)
knn3 = knn(train1, test1, train.Y, k=3)
table(knn3, test$Survived)
mean(knn3 != test$Survived)
knn4 = knn(train1, test1, train.Y, k=4)
```

table(knn4, test$Survived)

mean(knn4 != test$Survived)

knn5 = knn(train1, test1, train.Y, k=5)

table(knn5, test$Survived)

mean(knn5 != test$Survived)

```
> table(LR.probs,test$Survived)

LR.probs   0   1
       0 144  29
       1  25  69
> mean(LR.probs == test$Survived)
[1] 0.7977528
> table(LDA.class, test$Survived )

LDA.class   0   1
        0 146  30
        1  23  68
> mean(LDA.class != test$Survived)
[1] 0.1985019
> table(QDA.class, test$Survived )

QDA.class   0   1
        0 130  20
        1  39  78
> mean(QDA.class != test$Survived)
[1] 0.2209738

>  table(knn1, test$Survived)

knn1   0   1
   0 137  33
   1  32  65
> mean(knn1 != test$Survived)
[1] 0.2434457
> table(knn2, test$Survived)

knn2   0   1
   0 146  39
   1  23  59
> mean(knn2 != test$Survived)
[1] 0.2322097
> table(knn3, test$Survived)

knn3   0   1
   0 149  34
   1  20  64
> mean(knn3 != test$Survived)
[1] 0.2022472
> table(knn4, test$Survived)

knn4   0   1
   0 146  37
   1  23  61
```

```
> mean(knn4 != test$Survived)
[1] 0.2247191
> table(knn5, test$Survived)

knn5   0   1
   0 148  41
   1  21  57
> mean(knn5 != test$Survived)
[1] 0.2322097
```

Ans- The suitable value of K for KNN model is 3, which as a misclassification rate of 0.18. The TP of Logistic Regression, LDA, QDA and KNN(K=3) is 144, 146, 130 and 149 respectively. The FP for the models is 25, 23, 39, 20 respectively. True positive value tells us the number of positive classifications classified as positive. True negative value tells us the number of negative classifications classified as positive.

Question2)

```
#question 2

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Train.csv")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Test.csv')

test$Age[is.na(test$Age)] = mean(test$Age, na.rm=TRUE)

train$Age[is.na(train$Age)] = mean(train$Age, na.rm=TRUE)

levels(train$Embarked)

levels(train$Embarked)[1] = 'S'

install.packages('naniar')

library(naniar)

levels(test$Cabin)

levels(test$Cabin)[1] = 'NA'

attach(test)
```

formula1 = as.formula('factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked)+Cabin')

LR_C = glm(formula1,data= test, family='binomial')

LR_C.probs = predict(LR_C)

LR_C.probs = ifelse(LR_C.probs > 0.5, 1, 0)

mean(LR_C.probs!= test$Survived)

LR_C1 = glm(factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked),data= test, family= 'binomial')

LR_C1.probs = predict(LR_C1)

LR_C1.probs = ifelse(LR_C1.probs > 0.5, 1, 0)

mean(LR_C1.probs!= test$Survived)

```
> mean(LR_C.probs!= test$Survived)
[1] 0.1310861
> mean(LR_C1.probs!= test$Survived)
[1] 0.1685393
```

Ans) Training and testing are done on the test data set, since levels present in train and test data set are different. Error named – 'different levels present' is displayed when the model is trained on train data and tested on test data. This is the reason test data set is used to train and test the data. The misclassification of the model is 0.131, when cabin was included and 0.1685, when cabin is not included. This tells us that cabin feature has significance when it is included with the other predictors in the model.

Question-3)

#question3

```
install.packages('vcd')

library(vcd)

attach(train)

#fix(train)

od = glm(factor(Survived)~factor(Pclass) + Sex+ factor(Embarked), data= train, family = 'binomial')

x= od$coefficients

or = exp(x)

or
```

```
(Intercept)   factor(Pclass)2   factor(Pclass)3           Sexmale
       9.66393075        0.45464020        0.12852219        0.07174335
factor(Embarked)C factor(Embarked)Q
       1.48318932        1.51868001
```

Ans- The Adjusted odds ratio is calculated using the model, hence it's a multi-variate function. Unadjusted odd ratio is calculated for each variable hence it is a Uni-variate function. The adjusted odd ratio for P-class2, P-class3, sex, embarked C and embarked Q is 0.45, 0.12, 0.07174, 1.48 and 1.51. This tells us that when we increase P-class2 by 1 unit there will be a decrease in the response by a factor 0.45, when we increase sex by 1 unit there will be a decrease in the response by a factor 0.07174. And, when we increase Embarked C and Embarked Q by 1 unit there will be an increase in the response by a factor 1.48 and 1.51 respectively.

Question-4)

```
#question 4

formula = as.formula('factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked)')

LR_0.5 = glm(formula, data= train, family= 'binomial')
```

```
LR.probs_0.5 = predict(LR_0.5, type= 'response', newdata= test)

LR.probs_0.5 = ifelse(LR.probs_0.5 > 0.5, 1, 0)

table(LR.probs_0.5,test$Survived)

mean(LR.probs_0.5!= test$Survived)


LR_0.2 = glm(formula, data= train, family= 'binomial')

LR.probs_0.2 = predict(LR_0.2, type= 'response', newdata= test)

LR.probs_0.2 = ifelse(LR.probs_0.2> 0.2, 1, 0)

table(LR.probs_0.2,test$Survived)

mean(LR.probs_0.2!= test$Survived)




LR_0.8= glm(formula, data= train, family= 'binomial')

LR.probs_0.8 = predict(LR_0.8, type= 'response', newdata= test)

LR.probs_0.8 = ifelse(LR.probs_0.8 > 0.8, 1, 0)

table(LR.probs_0.8,test$Survived)

mean(LR.probs_0.8!= test$Survived)
```

```
> table(LR.probs_0.5,test$Survived)

LR.probs_0.5   0   1
           0 144  29
           1  25  69
> mean(LR.probs_0.5!= test$Survived)
[1] 0.2022472
> table(LR.probs_0.2,test$Survived)

LR.probs_0.2  0  1
           0 97 16
           1 72 82
> mean(LR.probs_0.2!= test$Survived)
[1] 0.329588
```

```
> table(LR.probs_0.8,test$Survived)

LR.probs_0.8   0   1
           0 167  59
           1   2  39
> mean(LR.probs_0.8!= test$Survived)
[1] 0.2284644
```

Ans- Threshold value of 0.5 is appropriate for survival prediction. The misclassification rate for threshold 0.5 is 0.20.

Question-5)

#question 5

install.packages('ROCR')

library(ROCR)

pred= predict(LR, type= 'response', newdata= test)

pr = prediction(pred,test$Survived)

prf<- performance(pr, measure='tpr', x.measure='fpr')

plot(prf)

abline(0,1)

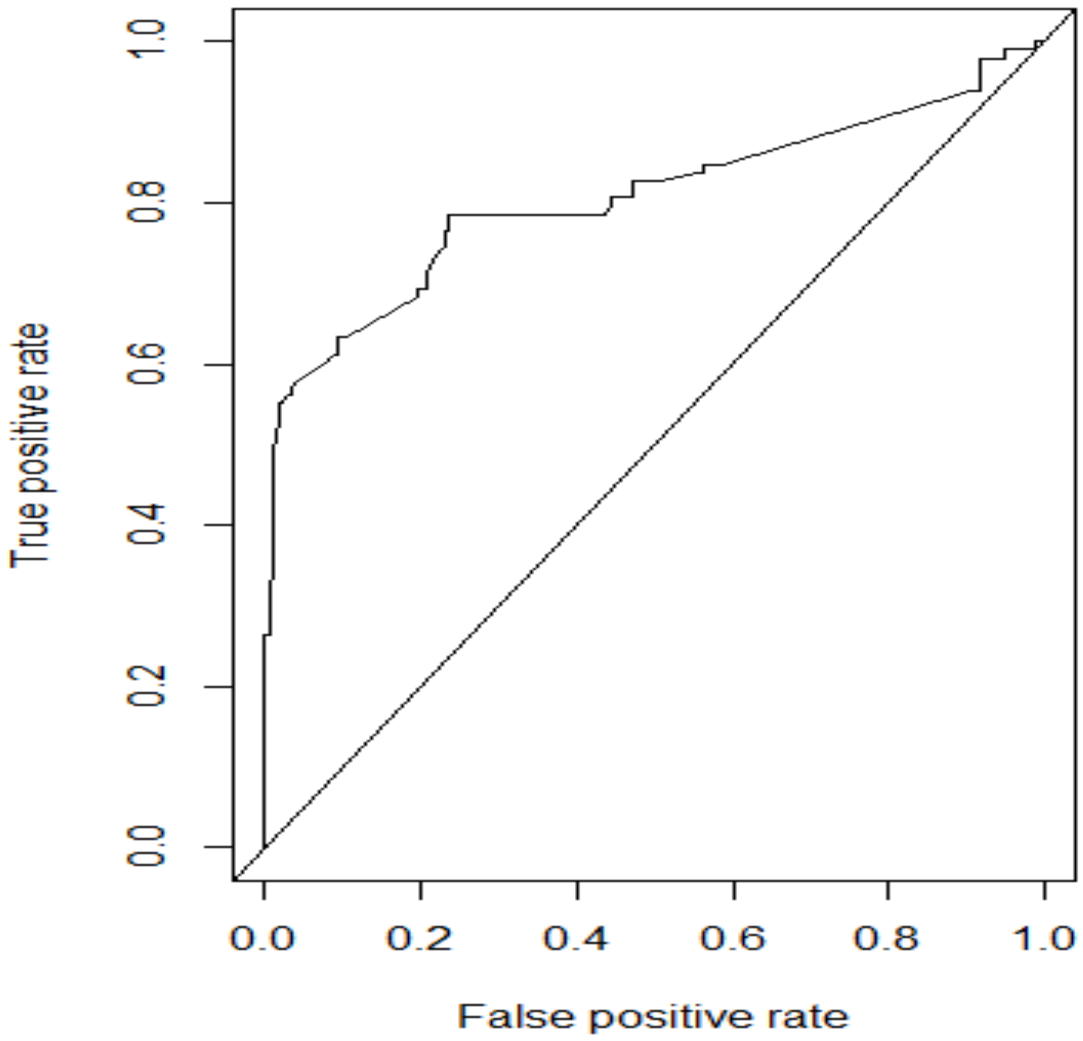
prf

auc_ROCR <- performance(pr, measure = "auc")

auc<- auc_ROCR@y.values[[1]]

auc

```
> auc
[1] 0.8060621
```


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

Question-6)

```
#question 6

#question 6

library(FNN)

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Train.csv")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Test.csv')

test$Age[is.na(test$Age)] = mean(test$Age, na.rm=TRUE)

train$Age[is.na(train$Age)] = mean(train$Age, na.rm=TRUE)

levels(train$Embarked)

levels(train$Embarked)[1] = 'S'

train$Embarked <- sapply(as.character(train$Embarked), switch, 'C' = 0, 'Q' = 1, 'S' = 2)

test$Embarked <- sapply(as.character(test$Embarked), switch, 'C' = 0, 'Q' = 1, 'S' = 2)

train$Sex <- ifelse(train$Sex == 'male', 1, 0)

test$Sex <- ifelse(test$Sex == 'male', 1, 0)

LR_7 = glm(factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked)+Parch+Fare,data = train, family= 'binomial')

summary(LR_7)

train2= train[, -c(1,2,4,8,9,10,11,12)]

test2= test[, -c(1,2,4,8,9,10,11,12)]

train.Y1= train$Survived

train1

set.seed(12)
```

```
knn11 <- knn(train2, test2, train.Y1, k=1)

table(knn11, test$Survived)

mean(knn11 != test$Survived)

knn21 = knn(train2, test2, train.Y1, k=2)

table(knn21, test$Survived)

mean(knn21 != test$Survived)

knn31 = knn(train2, test2, train.Y1, k=3)

table(knn31, test$Survived)

mean(knn31 != test$Survived)

knn41 = knn(train2, test2, train.Y1, k=4)

table(knn41, test$Survived)

mean(knn41 != test$Survived)

knn51 = knn(train2, test2, train.Y1, k=5)

table(knn51, test$Survived)

mean(knn51 != test$Survived)
```

Output-

```
> table(knn11, test$Survived)

knn11   0   1
    0 135  35
    1  34  63
> mean(knn11 != test$Survived)
[1] 0.258427
> table(knn21, test$Survived)

knn21   0   1
    0 143  27
    1  26  71
> mean(knn21 != test$Survived)
[1] 0.1985019
> table(knn31, test$Survived)
```

```
knn31   0   1
    0 139  29
    1  30  69
> mean(knn31 != test$Survived)
[1] 0.2209738
> table(knn41, test$Survived)

knn41   0   1
    0 148  33
    1  21  65
> mean(knn41 != test$Survived)
[1] 0.2022472
> table(knn51, test$Survived)

knn51   0   1
    0 145  36
    1  24  62
> mean(knn51 != test$Survived)
[1] 0.2247191

>
```

Ans) The significant features can be found out using p-values of the predictors from Logistic regression model and they are P-class, sex, age, SibSp. Using these features KNN model is built. The model performs better than the previous model since the misclassification error for K=2 is 0.198.

R-CODE-

```
#Question 1

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Train.csv")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Test.csv')

test$Age[is.na(test$Age)] = mean(test$Age, na.rm=TRUE)

train$Age[is.na(train$Age)] = mean(train$Age, na.rm=TRUE)

levels(train$Embarked)

levels(train$Embarked)[1] = 'S'

train$Embarked <- sapply(as.character(train$Embarked), switch, 'C' = 0, 'Q' = 1, 'S' = 2)
```

```
test$Embarked <- sapply(as.character(test$Embarked), switch, 'C' = 0, 'Q' = 1, 'S' = 2)

train$Sex <- ifelse(train$Sex == 'male', 1, 0)

test$Sex <- ifelse(test$Sex == 'male', 1, 0)

#fix(train)


#logistic regression

attach(train)

LR = glm( factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked), data= train, family= 'binomial')

LR.probs = predict(LR, type= 'response', newdata= test)

LR.probs

LR.probs = ifelse(LR.probs > 0.5, 1, 0)

table(LR.probs,test$Survived)

mean(LR.probs == test$Survived)


 #LDA

library(MASS)

LDA = lda(factor(Survived)~ factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked) , data= train)

LDA.pred = predict(LDA , newdata= test)

LDA.class = LDA.pred$class

LDA.class
```

```
table(LDA.class, test$Survived )

mean(LDA.class != test$Survived)


#QDA

library(MASS)

QDA = qda(factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch + factor(Embarked), data= train, family= binomial)

QDA.pred = predict(QDA , newdata= test)

QDA.class = QDA.pred$class

QDA.class

table(QDA.class, test$Survived )

mean(QDA.class != test$Survived)


.#KNN

library(FNN)

#train$Age <- ifelse(train$Age<18, 1, 0)

#test$Age <- ifelse(test$Age<18, 1, 0)

train$Age <- ifelse(train$Age<18, 1, 0)

test$Age <- ifelse(test$Age<18, 1, 0)

train1= train[, -c(1,2,4,9,10,11)]

test1= test[, -c(1,2,4,9,10,11)]

train.Y = train$Survived
```

```
train1

set.seed(12)

knn1 <- knn(train1, test1, train.Y, k=1)

table(knn1, test$Survived)

mean(knn1 != test$Survived)

knn2 = knn(train1, test1, train.Y, k=2)

table(knn2, test$Survived)

mean(knn2 != test$Survived)

knn3 = knn(train1, test1, train.Y, k=3)

table(knn3, test$Survived)

mean(knn3 != test$Survived)

knn4 = knn(train1, test1, train.Y, k=4)

table(knn4, test$Survived)

mean(knn4 != test$Survived)

knn5 = knn(train1, test1, train.Y, k=5)

table(knn5, test$Survived)

mean(knn5 != test$Survived)


#question 2

train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Train.csv")

test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Test.csv')

test$Age[is.na(test$Age)] = mean(test$Age, na.rm=TRUE)
```

```
train$Age[is.na(train$Age)] = mean(train$Age, na.rm=TRUE)

levels(train$Embarked)

levels(train$Embarked)[1] = 'S'

install.packages('naniar')

library(naniar)

levels(test$Cabin)

levels(test$Cabin)[1] = 'NA'

attach(test)

formula1 = as.formula('factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked)+Cabin')

LR_C = glm(formula1,data= test, family='binomial')

LR_C.probs = predict(LR_C)

LR_C.probs = ifelse(LR_C.probs > 0.5, 1, 0)

mean(LR_C.probs!= test$Survived)


LR_C1 = glm(factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked),data= test, family= 'binomial')

LR_C1.probs = predict(LR_C1)

LR_C1.probs = ifelse(LR_C1.probs > 0.5, 1, 0)

mean(LR_C1.probs!= test$Survived)
```

```
#question3

install.packages('vcd')

library(vcd)

attach(train)

#fix(train)

od = glm(factor(Survived)~factor(Pclass) + Sex+ factor(Embarked), data= train, family = 'binomial')

x= od$coefficients

or = exp(x)

or


#question 4

formula = as.formula('factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked)')

LR_0.5 = glm(formula, data= train, family= 'binomial')

LR.probs_0.5 = predict(LR_0.5, type= 'response', newdata= test)

LR.probs_0.5 = ifelse(LR.probs_0.5 > 0.5, 1, 0)

table(LR.probs_0.5,test$Survived)

mean(LR.probs_0.5!= test$Survived)


LR_0.2 = glm(formula, data= train, family= 'binomial')

LR.probs_0.2 = predict(LR_0.2, type= 'response', newdata= test)
```

```
LR.probs_0.2 = ifelse(LR.probs_0.2> 0.2, 1, 0)
table(LR.probs_0.2,test$Survived)
mean(LR.probs_0.2!= test$Survived)



LR_0.8= glm(formula, data= train, family= 'binomial')
LR.probs_0.8 = predict(LR_0.8, type= 'response', newdata= test)
LR.probs_0.8 = ifelse(LR.probs_0.8 > 0.8, 1, 0)
table(LR.probs_0.8,test$Survived)
mean(LR.probs_0.8!= test$Survived)

#question 5
install.packages('ROCR')
library(ROCR)
pred= predict(LR, type= 'response', newdata= test)
pr = prediction(pred,test$Survived)
prf<- performance(pr, measure='tpr', x.measure='fpr')
plot(prf)
abline(0,1)
prf
auc_ROCR <- performance(pr, measure = "auc")
auc<- auc_ROCR@y.values[[1]]
```

auc

#question 6

```
library(FNN)
train = read.csv("C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Train.csv")
test = read.csv('C:¥¥Users¥¥KIRAN KONDISETTI¥¥Desktop¥¥Test.csv')
test$Age[is.na(test$Age)] = mean(test$Age, na.rm=TRUE)
train$Age[is.na(train$Age)] = mean(train$Age, na.rm=TRUE)
levels(train$Embarked)
levels(train$Embarked)[1] = 'S'
train$Embarked <- sapply(as.character(train$Embarked), switch, 'C' = 0, 'Q' = 1, 'S' = 2)
test$Embarked <- sapply(as.character(test$Embarked), switch, 'C' = 0, 'Q' = 1, 'S' = 2)
train$Sex <- ifelse(train$Sex == 'male', 1, 0)
test$Sex <- ifelse(test$Sex == 'male', 1, 0)
LR_7 = glm(factor(Survived)~factor(Pclass) + Sex+ Age+SibSp+ Parch+factor(Embarked)+Parch+Fare,data = train, family= 'binomial')
summary(LR_7)
train2= train[, -c(1,2,4,7,9,10,11,12)]
test2= test[, -c(1,2,4,7,9,10,11,12)]
train.Y1= train$Survived
train1
```

```
set.seed(143)

knn11 <- knn(train2, test2, train.Y1, k=1)

table(knn11, test$Survived)

mean(knn11 != test$Survived)

knn21 = knn(train2, test2, train.Y1, k=2)

table(knn21, test$Survived)

mean(knn21 != test$Survived)

knn31 = knn(train2, test2, train.Y1, k=3)

table(knn31, test$Survived)

mean(knn31 != test$Survived)

knn41 = knn(train2, test2, train.Y1, k=4)

table(knn41, test$Survived)

mean(knn41 != test$Survived)

knn51 = knn(train2, test2, train.Y1, k=5)

table(knn51, test$Survived)

mean(knn51 != test$Survived)
```