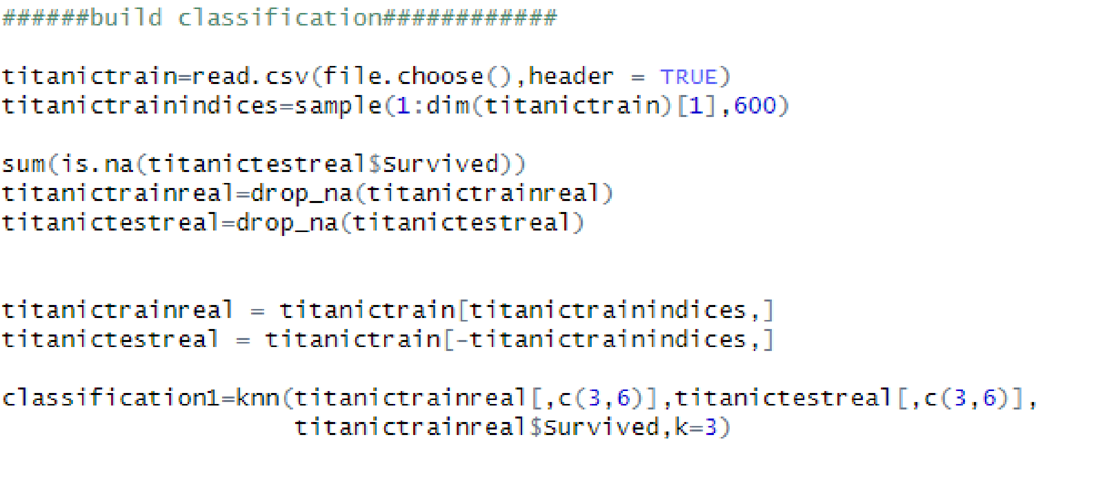
FLS 6 Haitie Liu

Titanic KNN Study



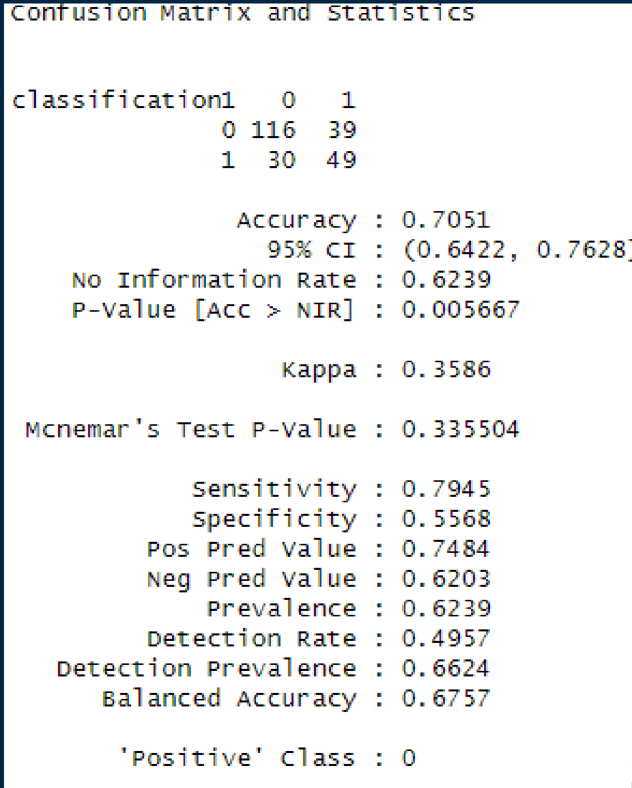
```
######build classification############

titanictrain=read.csv(file.choose(),header = TRUE)
titanictrainindices=sample(1:dim(titanictrain)[1],600)

sum(is.na(titanictestreal$Survived))
titanictrainreal=drop_na(titanictrainreal)
titanictestreal=drop_na(titanictestreal)


titanictrainreal = titanictrain[titanictrainindices,]
titanictestreal = titanictrain[-titanictrainindices,]

classification1=knn(titanictrainreal[,c(3,6)],titanictestreal[,c(3,6)],
                    titanictrainreal$Survived,k=3)
```

Loading data as csv file and "tidy" the dataset, dropping missing value using drop_na

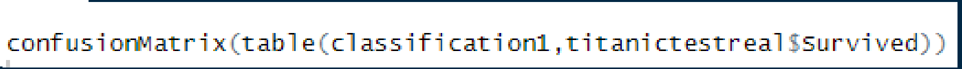
Partition the dataset into random 600 as training set and rest as test set

Run KNN to build classification

Titanic KNN Study



```
Confusion Matrix and Statistics


classification1   0   1
              0 116  39
              1  30  49

               Accuracy : 0.7051
                 95% CI : (0.6422, 0.7628)
    No Information Rate : 0.6239
    P-Value [Acc > NIR] : 0.005667

                  Kappa : 0.3586

 Mcnemar's Test P-Value : 0.335504

            Sensitivity : 0.7945
            Specificity : 0.5568
         Pos Pred Value : 0.7484
         Neg Pred Value : 0.6203
             Prevalence : 0.6239
         Detection Rate : 0.4957
   Detection Prevalence : 0.6624
      Balanced Accuracy : 0.6757

       'Positive' Class : 0
```

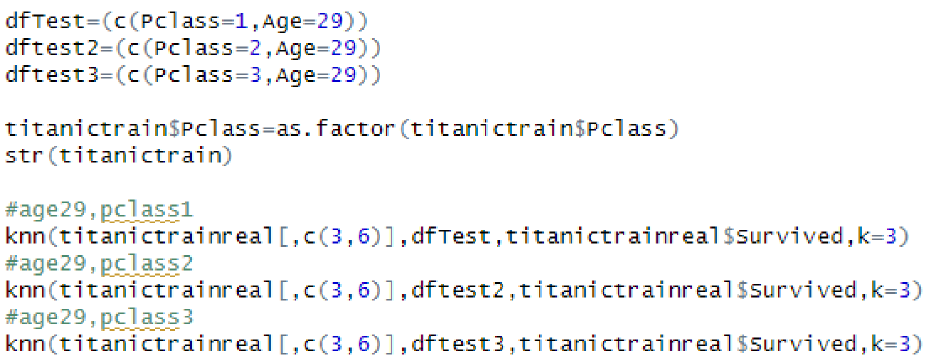


```
confusionMatrix(table(classification1,titanictestreal$Survived))
```

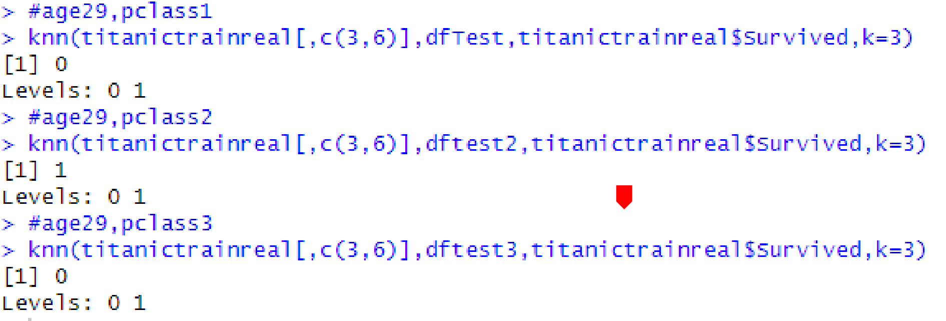
Looking at the image to the left

Running confusion table on the classification, we get the accuracy at 0.7051, Sensitivity at 0.7945, Specificity at 0.5568

Titanic KNN Study



```
dfTest=(c(Pclass=1,Age=29))
dftest2=(c(Pclass=2,Age=29))
dftest3=(c(Pclass=3,Age=29))

titanictrain$Pclass=as.factor(titanictrain$Pclass)
str(titanictrain)

#age29,pclass1
knn(titanictrainreal[,c(3,6)],dfTest,titanictrainreal$Survived,k=3)
#age29,pclass2
knn(titanictrainreal[,c(3,6)],dftest2,titanictrainreal$Survived,k=3)
#age29,pclass3
knn(titanictrainreal[,c(3,6)],dftest3,titanictrainreal$Survived,k=3)
```

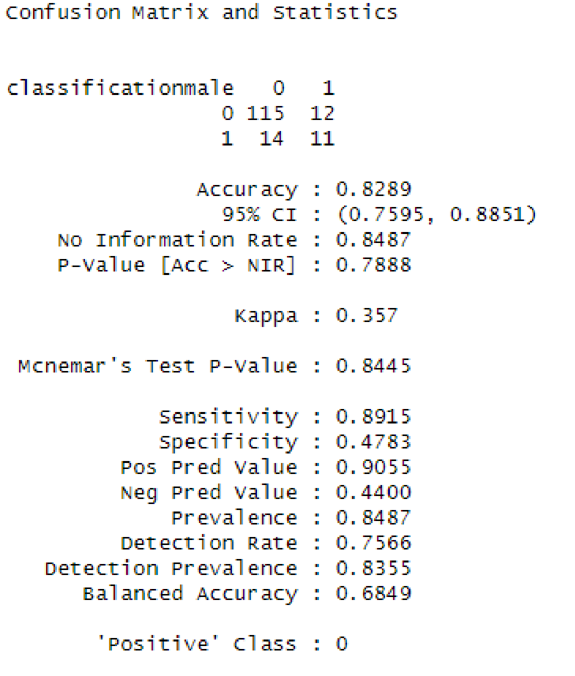


```
> #age29,pclass1
> knn(titanictrainreal[,c(3,6)],dfTest,titanictrainreal$Survived,k=3)
[1] 0
Levels: 0 1
> #age29,pclass2
> knn(titanictrainreal[,c(3,6)],dftest2,titanictrainreal$Survived,k=3)
[1] 1
Levels: 0 1
> #age29,pclass3
> knn(titanictrainreal[,c(3,6)],dftest3,titanictrainreal$Survived,k=3)
[1] 0
Levels: 0 1
```

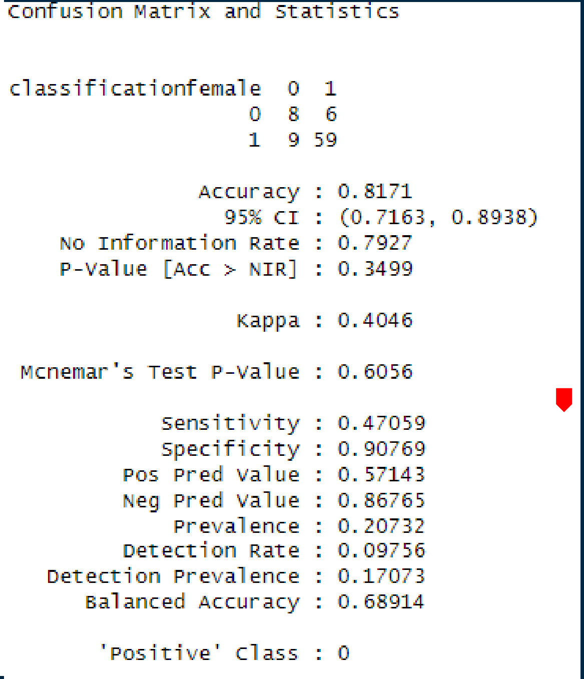
Running the test using age = 29, and three different classes.

We get the result that if I were to book second passenger class, I would have survived

Titanic KNN Study



```
Confusion Matrix and Statistics


classificationmale   0   1
                 0 115  12
                 1  14  11

               Accuracy : 0.8289
                 95% CI : (0.7595, 0.8851)
    No Information Rate : 0.8487
    P-Value [Acc > NIR] : 0.7888

                  Kappa : 0.357

 Mcnemar's Test P-Value : 0.8445

            Sensitivity : 0.8915
            Specificity : 0.4783
         Pos Pred Value : 0.9055
         Neg Pred Value : 0.4400
             Prevalence : 0.8487
         Detection Rate : 0.7566
   Detection Prevalence : 0.8355
      Balanced Accuracy : 0.6849

       'Positive' Class : 0
```



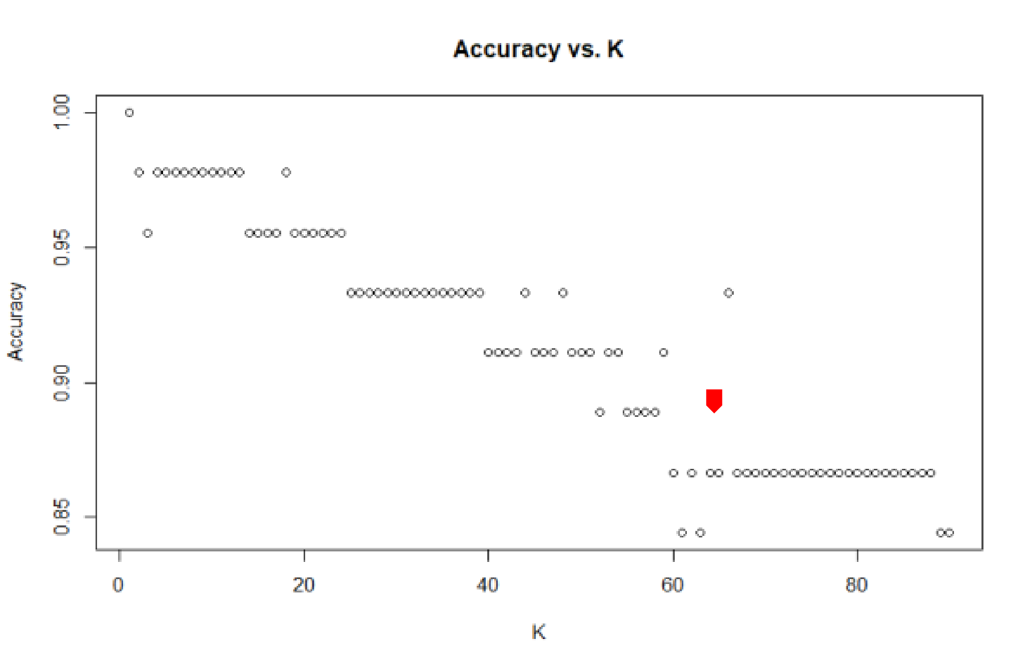
```
Confusion Matrix and Statistics


classificationfemale  0  1
                   0  8  6
                   1  9 59

               Accuracy : 0.8171
                 95% CI : (0.7163, 0.8938)
    No Information Rate : 0.7927
    P-Value [Acc > NIR] : 0.3499

                  Kappa : 0.4046

 Mcnemar's Test P-Value : 0.6056

            Sensitivity : 0.47059
            Specificity : 0.90769
         Pos Pred Value : 0.57143
         Neg Pred Value : 0.86765
             Prevalence : 0.20732
         Detection Rate : 0.09756
   Detection Prevalence : 0.17073
      Balanced Accuracy : 0.68914

       'Positive' Class : 0
```

We can compare the differences if we partition the data sets into male and female

On the left is male, with accuracy at 0.8289.

On the right is female, with accuracy at 0.8171

Iris Study

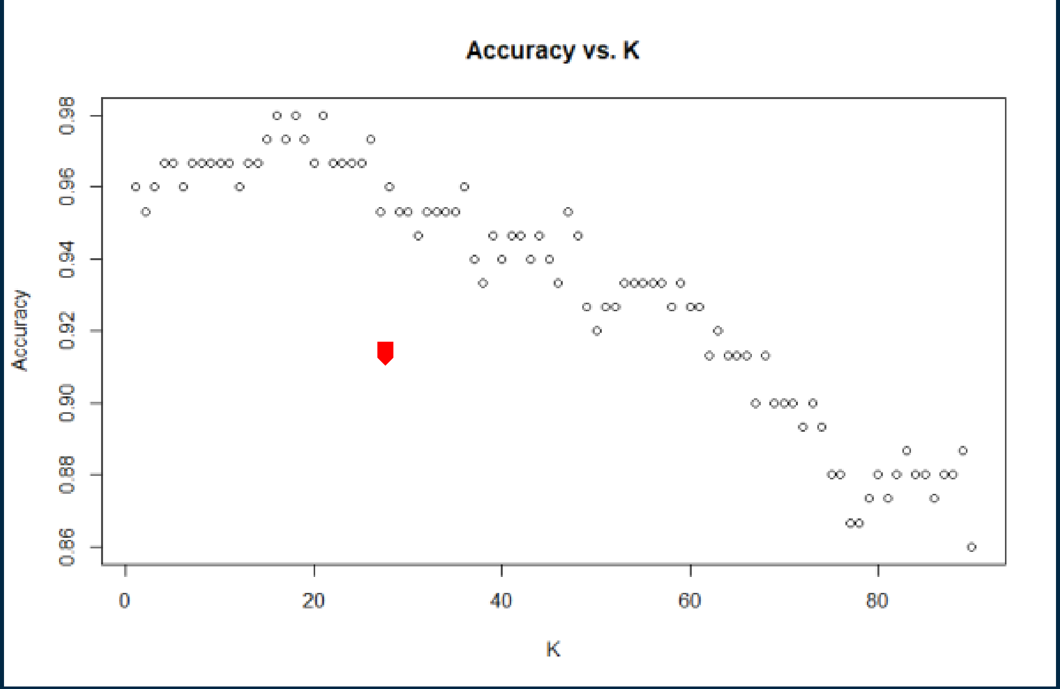


We can see that from the chart to the left.

When k=1, we seem to have the highest accuracy with 100%.

As K goes down, the accuracy stalls and decreases

Iris Study (Leave One Out)



Using leave one out method, we can see that when K equals 19 – 22, we have the highest accuracy close to 98%

When K is above 80, we have the lowest accuracy around 86%

TAKE AWAY

Would love to try using a Json file to practice on these questions, it was fun using my age to run against to dataset to test my survivability.