Principal Components using k-NN

kNN – Accuracy comparison using PCA components

Dataset: IRIS
150 observations and 4 features and one is classifier
Data consists of 3 species of flowers which is measured using four features of flower.
Sepal L., Sepal W., Petal L., and Petal W

```
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Specie
s"
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

k-NN – k nearest neighbors here is used as classification to classify given flower with features as one of the class by relating it to k nearest neighbors.

Diving dataset into train and test data:

```
> dim(iris)
[1] 150   5
> dim(iris.train)
[1] 112   5
> dim(iris.test)
[1] 38  5
```

Train and Test data has all species:

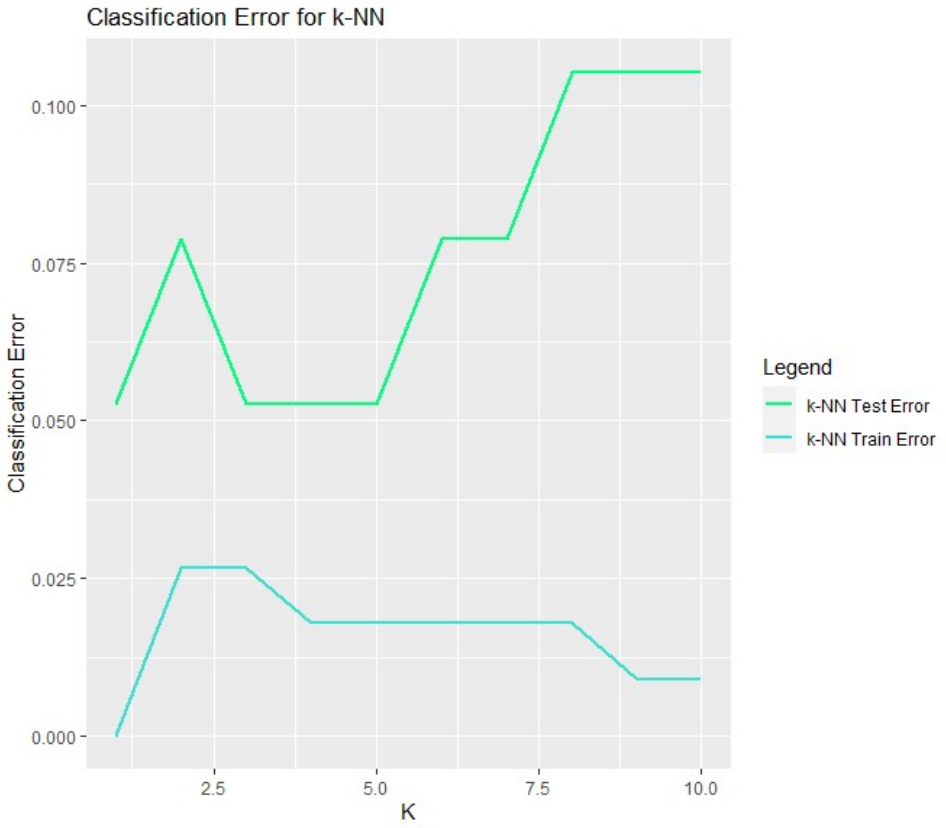
```
> unique(iris$Species)
[1] setosa     versicolor virginica
Levels: setosa versicolor virginica
> unique(iris.train$Species)
[1] virginica  versicolor setosa
Levels: setosa versicolor virginica
> unique(iris.test$Species)
[1] setosa     versicolor virginica
Levels: setosa versicolor virginica
```

For k-NN, we have to find the optimal value for k for the given dataset which minimizes the classification error. So, we have to train and test data for different values of k before choosing the appropriate k.

Let's use a function which can be called for different values of k.

```
applyknn<-function(k,tdata,testdata){
    knn(train=tdata[,-5],
    test=testdata[,-5],
    cl=as.factor(tdata$Species),
    k=k)
}
```

Plot classification error for different k values:



From the plot we can observe the test error is less for k values 3,4,5 its better to have lower k to avoid mis classification.

Now let's find the confusion matrix for the **optimal k which is 3.**

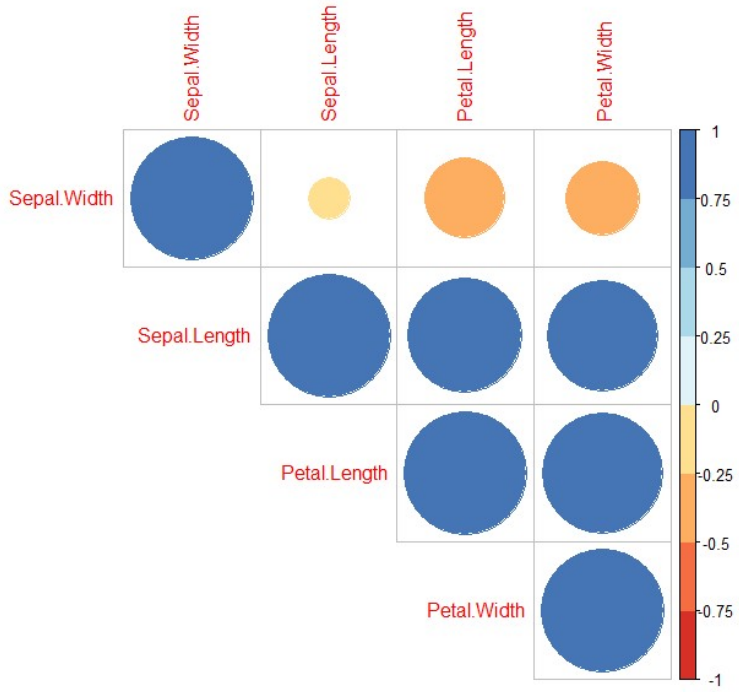
```
> testconfmat
Confusion Matrix and Statistics

          Reference
Prediction   setosa versicolor virginica
  setosa         12          0         0
  versicolor      0         17         0
  virginica       0          3         6

Overall Statistics

               Accuracy : 0.9211
                 95% CI : (0.7862, 0.9834)
    No Information Rate : 0.5263
    P-Value [Acc > NIR] : 1.726e-07

                  Kappa : 0.8742

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.8500           1.0000
Specificity                 1.0000            1.0000           0.9062
Pos Pred Value              1.0000            1.0000           0.6667
Neg Pred Value              1.0000            0.8571           1.0000
Prevalence                  0.3158            0.5263           0.1579
Detection Rate              0.3158            0.4474           0.1579
Detection Prevalence        0.3158            0.4474           0.2368
Balanced Accuracy           1.0000            0.9250           0.9531
```

k-NN was able to classify given test data with 92% accuracy but misclassified 3 flowers of species virginica as versicolor. If the data points of different classes are close by and separated by true decision boundary then there might be more chances of mis classification that's the reason choosing optimal k matters.

k-NN with PCA:

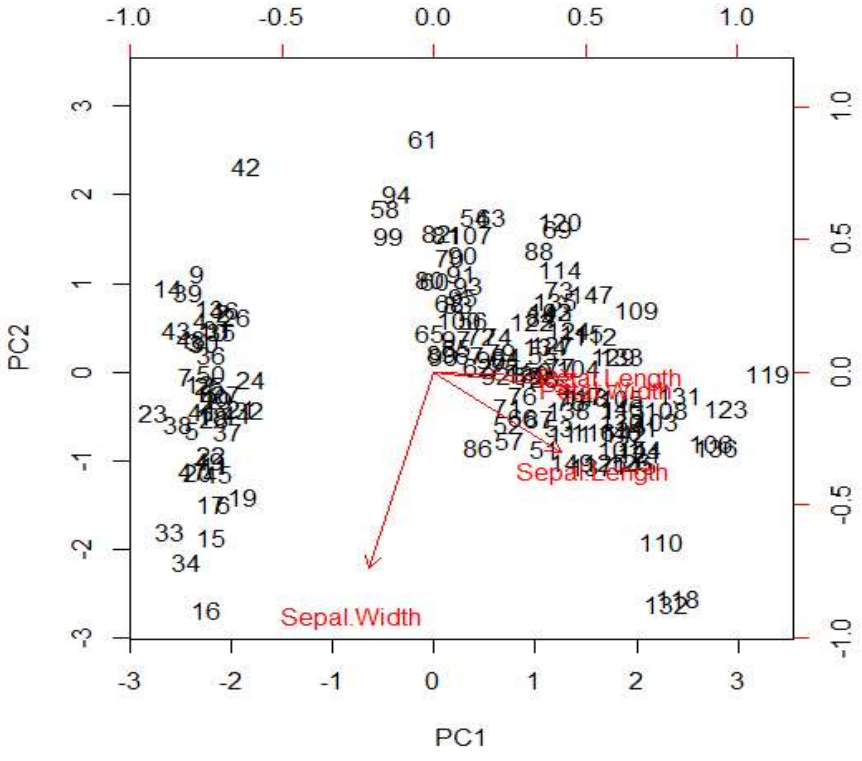
Principal component analysis shows the components which are responsible for the variance in the dataset by using only those components we can apply kNN and see how it behaves.

Let's understand the correlation between features in the dataset.



Sepal width is loosely correlated with other features but other three features are highly correlated.

Principal components:



Actually, the much divergence in the dataset is created by petal length and petal width. So, using two principal components is enough which covers variance of all of the dataset.

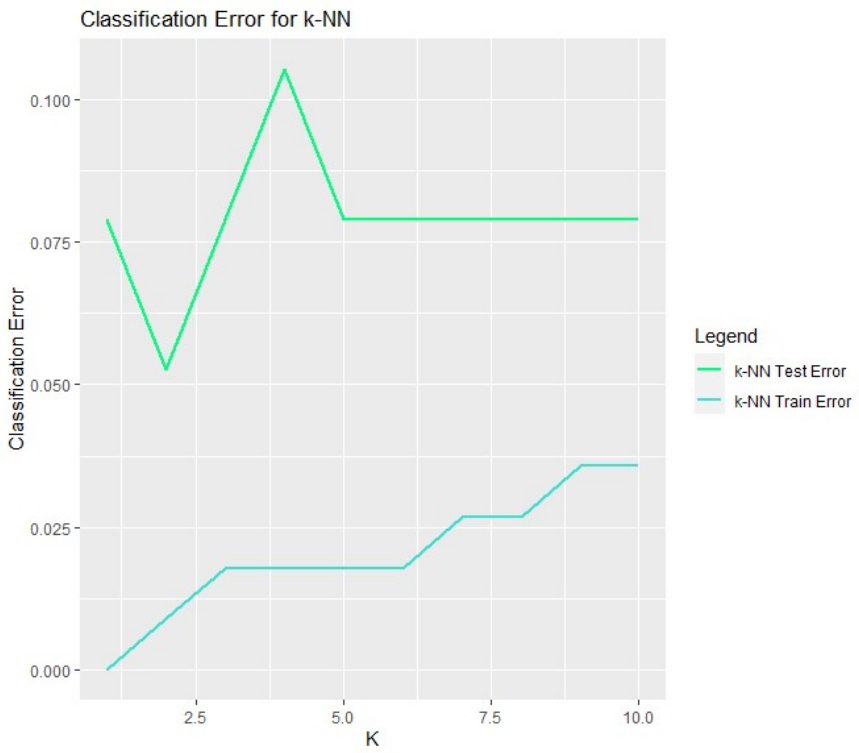
```
> pcomp$rotation
                PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
> var<-pcomp$sdev^2
>
> pervar<-var/sum(var)
> pervar
[1] 0.729624454 0.228507618 0.036689219 0.005178709
```

Considering first two principal components it covers 95% of dataset. Let's use first two principal components for k-NN and determine the accuracy of classification.

```
> iris.pcomp<-cbind(pca.comp1,pca.comp2)
> head(iris.pcomp)
  pca.comp1  pca.comp2
1  2.684126 -0.3193972
2  2.714142  0.1770012
3  2.888991  0.1449494
4  2.745343  0.3182990
5  2.728717 -0.3267545
6  2.280860 -0.7413304
> dim(iris.pcomp)
[1] 150   2
>
> train_index <- sample(1:nrow(iris.pcomp), (3/4) * nrow(iris.pcomp))
> test_index <- setdiff(1:nrow(iris.pcomp), train_index)
>
> iris.train<-iris.pcomp[train_index,]
> iris.test<-iris.pcomp[test_index,]
> train.label<-iris[train_index,5]
> test.label<-iris[test_index,5]
> dim(iris.train)
[1] 112   2
> dim(iris.test)
[1] 38  2
> length(train.label)
[1] 112
> length(test.label)
[1] 38
```

Train and test data created with first two principal components.

Find optimal k for the principal components' dataset.



For the k=2 the test error was low. But to compare accuracy with non PCA knn let's use the k=3 and find the confusion matrix.

```
> conftest
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         12          0         0
  versicolor      0         17         0
  virginica       0          3         6

Overall Statistics

               Accuracy : 0.9211
                 95% CI : (0.7862, 0.9834)
    No Information Rate : 0.5263
    P-Value [Acc > NIR] : 1.726e-07

                  Kappa : 0.8742

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.8500           1.0000
Specificity                 1.0000            1.0000           0.9062
Pos Pred Value              1.0000            1.0000           0.6667
Neg Pred Value              1.0000            0.8571           1.0000
Prevalence                  0.3158            0.5263           0.1579
Detection Rate              0.3158            0.4474           0.1579
Detection Prevalence        0.3158            0.4474           0.2368
Balanced Accuracy           1.0000            0.9250           0.9531
```

Conclusion: Accuracy of k-NN using PCA is 92% same as non PCA k-NN- Part A as the miss classification which is due to no true decision boundary between classes versicolor and virginica even after using principal components as a result the flowers of species virginica is misclassified as versicolor which is clearly evident from below plot.