final_project_problem3

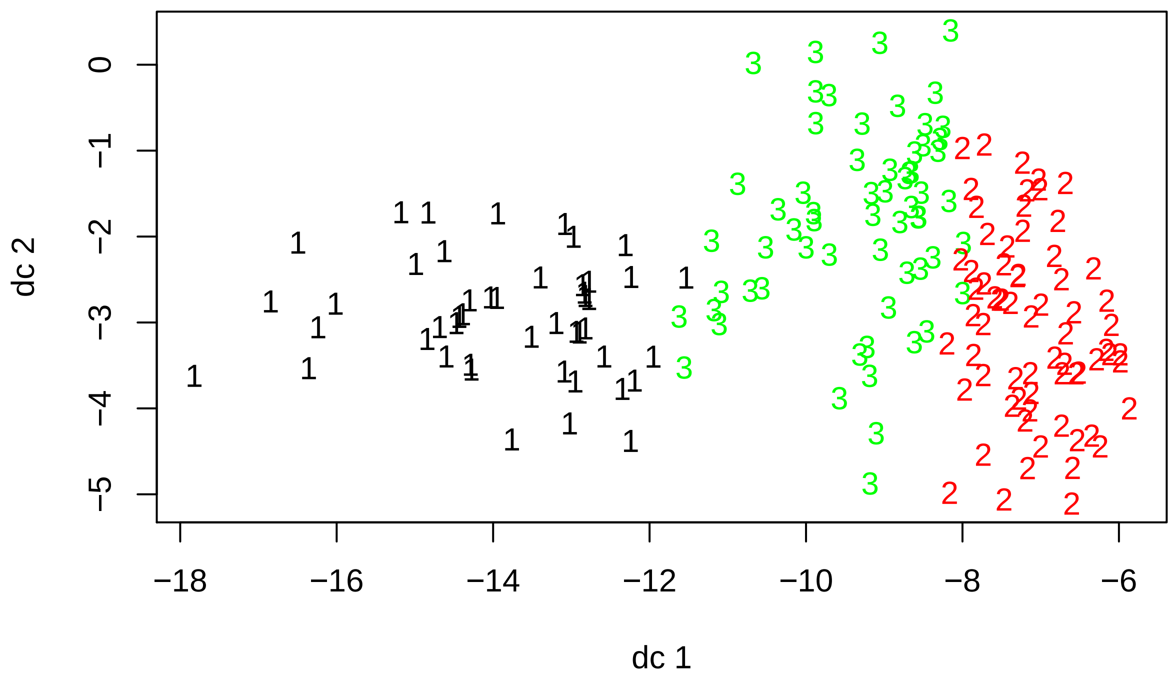
Junxiao Bu

October 25, 2015

K-Means – wine data

unscaled result

```
library('fpc')
data(wine, package="rattle")
attach(wine)
library(caret)
set.seed(1241)
Type <- as.integer(wine$Type)
## fit the model with train data
data_train <- wine[-1]
km_out <- kmeans(data_train,3)
#plot the result
plotcluster(data_train, km_out$cluster)
```



```
#calculate the misclassification rate for each each wine type
accuracy_non_scale <- confusionMatrix(table(data.frame(prediction= km_out$cluster,type = Type)))
## total accuracy
accuracy_non_scale[[3]][1]
```

```
##  Accuracy
## 0.7022472
```

		$True\,type$	
		1	2
$Predicted\,Type$	1	True Positive	False Positive
	2	False Negative	Ture Negative

```
## prediction accuracy for each class
accuracy_non_scale[[4]][,1]
```

```
##  Class: 1  Class: 2  Class: 3
## 0.7796610 0.7042254 0.6041667
```

To quantify how well of the algorithm's clusters correspond to the three wine types, we use confusion matrix to summarize each cluster's prediction accuracy. The simplest confusion matrix is:

- $True\,Positive$: True type is 1. The prediction type is 1.
- $False\,Positive$: True type is 2. The prediction type is 1.
- $False\,negative$: True type is 1. The prediction type is 2.
- $True\,Negative$: True type is 2. The prediction type is 2.
- For type 1: the prediction accuracy is: $\frac{True\,Positive}{True\,Positive+False\,Positive}$. We also called this rate as sensitivity.
- For type 2: the prediction accuracy is: $\frac{True\,Negative}{True\,Negative+False\,Negative}$
- The total prediction accuracy is: $\frac{True\,Positive+True\,Negative}{True\,Positive+True\,Negative+False\,Positive+False\,Negative}$

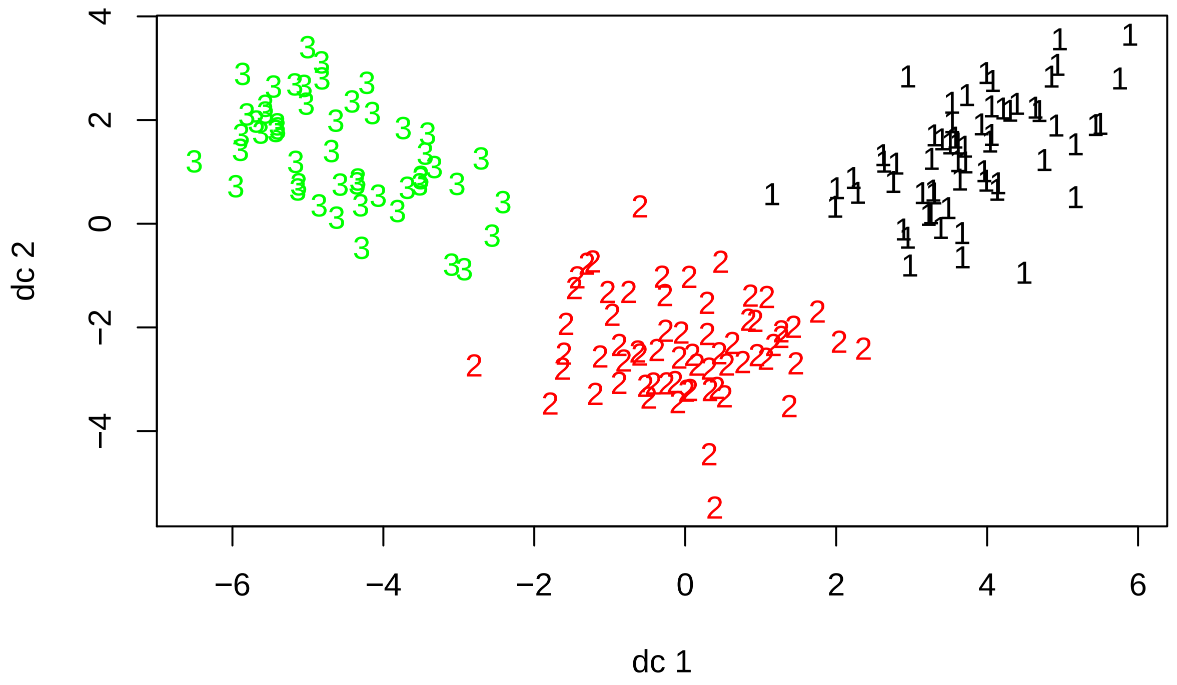
In this case, the prediction accuracy for type 1 is 0.779. The prediction accuracy for type 2 is 0.7042254. The prediction accuracy for type 3 is 0.6042. The total prediction accuracy is 0.702.

Now we compare the corresponding results with the scaled data.

scaled result

```
# fit the model with scale data

data_train_scale <- scale(wine[-1])
km_out_scale <- kmeans(data_train_scale,3)
plotcluster(data_train_scale, km_out_scale$cluster)
```



```
accuracy_scale <- confusionMatrix(table(data.frame(prediction= km_out_scale$cluster,type =Type)))
accuracy_scale[[3]][1]
```

```
##  Accuracy
## 0.9662921
```

```
accuracy_scale[[4]][,1]
```

```
## Class: 1 Class: 2 Class: 3
## 1.000000 0.915493 1.000000
```

In this case, the prediction accuracy for type 1 is 1. The prediction accuracy for type 2 is 0.9154. The prediction accuracy for type 3 is 1. The total prediction accuracy is 0.966.

After scaling the training data, the prediction accuracy for each class increased. The reason why we should scale the data before applying the K-means algorithms is simple. The K-means minimizes the error function using the Newton algorithm, i.e. a gradient-based optimization algorithm. Normalizing the data improves convergence of such algorithms. The idea is that if different components of feasures have different scales, then derivatives tend to align along directions with higher variance, which leads to poorer/slower convergence.

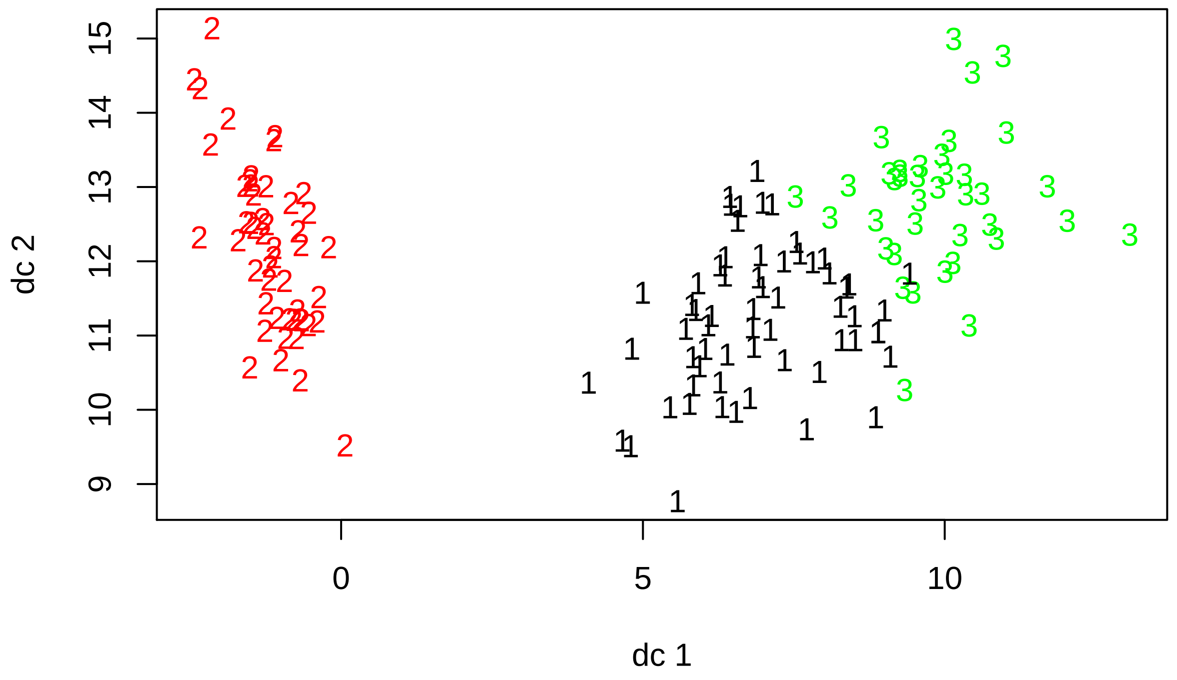
K-means – iris dataset

unscaled result

```
data(iris)
attach(iris)
set.seed(1241)
Type_iris <- as.integer(iris$Species)
```

```
iris_train <- iris[,-5]

kmean_iris_unscaled <- kmeans(iris_train,3)
plotcluster(iris_train, kmean_iris_unscaled$cluster)
```



```
accuracy_non_scale_iris <- confusionMatrix(table(data.frame(prediction= kmean_iris_unscaled$cluster,type

accuracy_non_scale_iris[[3]][1]
```

```
## Accuracy
##     0.24
```

```
accuracy_non_scale_iris[[4]][,1]
```

```
## Class: 1 Class: 2 Class: 3
##     0.00     0.00     0.72
```

scaled result

```
iris_train_scale <- scale(iris[,-5])
kmean_iris_scaled <- kmeans(iris_train_scale,3)
#plotcluster(iris_train_scale, kmean_iris_scaled$cluster)
accuracy_scale_iris <- confusionMatrix(table(data.frame(prediction= kmean_iris_scaled$cluster,type = Ty

accuracy_scale_iris[[3]][1]
```

```
## Accuracy
##     0.44
```

```
accuracy_scale_iris[[4]][,1]
```

```
## Class: 1 Class: 2 Class: 3
##     0.32     0.00     1.00
```

The total prediction accuracy of scaled data is also better than the unscaled data.