Machine Learning assignment

Gr

1/9/2021

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

The goal of your project is to predict the manner in which they did the exercise.

Reading Data and initial exploration

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
set.seed(12463)
raw_input_data <- read.csv("pml-training.csv")
```

```
head(raw_input_data)
unique(raw_input_data$classe)
str(raw_input_data)
```

```
sum(is.na(raw_input_data))
```

```
## [1] 1287472
```

```
sum(!is.na(raw_input_data))
```

```
## [1] 1852048
```

Data contains 5 levels of classe variable which is expected based on data description. There is a lot of missing data, at first I will take strategy to remove columns with missing data. If final result will not be satisfactory this approach might be modified. Near zero variable also will be removed and columns which doesn`t seem to be relevant for model building.

Basic transformation

```
input_data <- raw_input_data
input_data$classe <- as.factor(input_data$classe)
input_data <- input_data[,-nearZeroVar(input_data)]
input_data <- input_data[,-c(1,2,3,4,5,6,7)]
input_data <- input_data[, colSums(is.na(input_data)) == 0]
```

Spliting data into Training and Testing data set - cross validation

```
inTrain <- createDataPartition(y=input_data$classe, p=0.75, list=FALSE)
training <- input_data[inTrain,]
testing <- input_data[-inTrain,]
```

We use knn method to train model and pre process data by centering, scaling and applying PCA

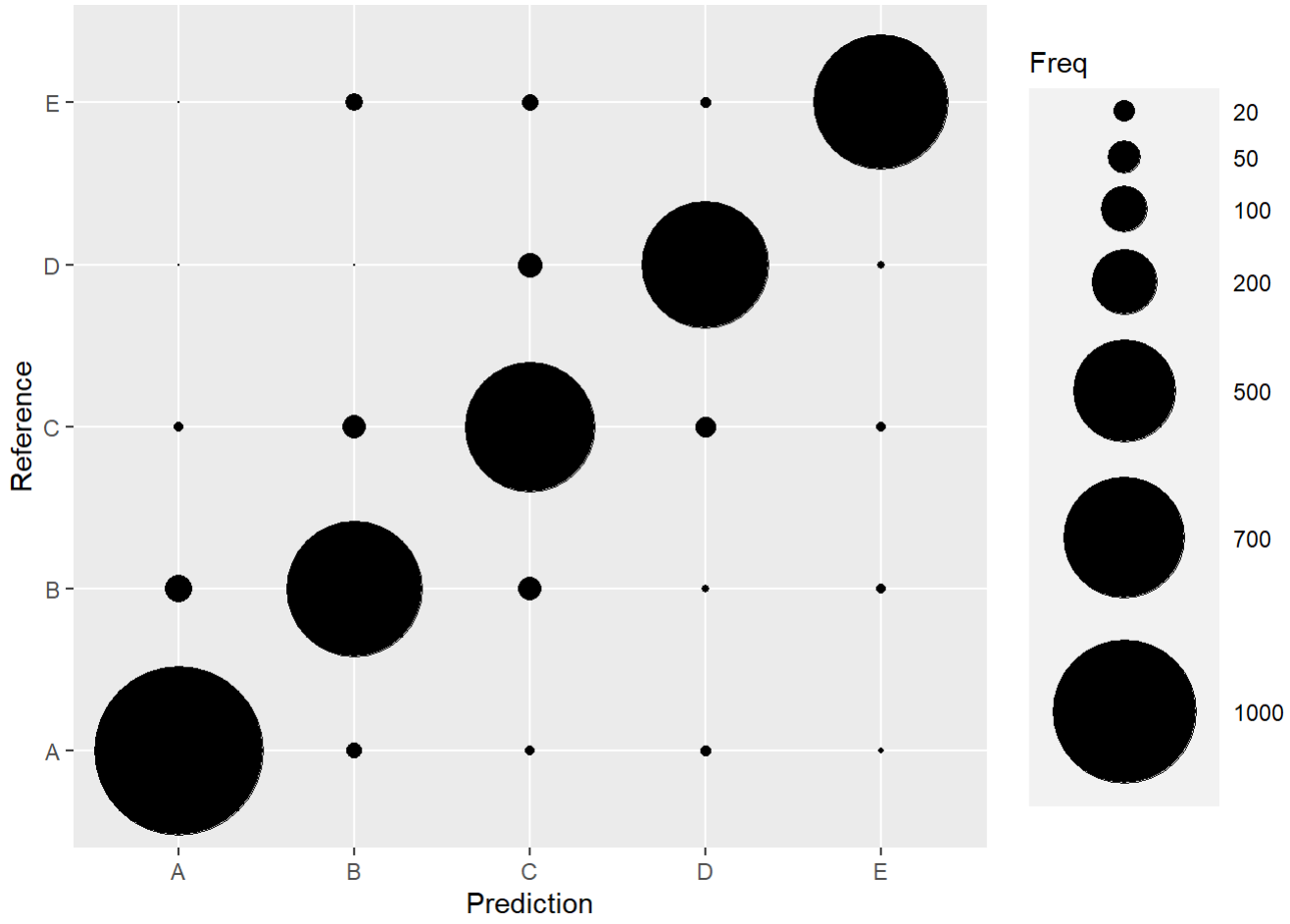
```
trained_model_KNN <- train(classe ~.,data=training, method="knn", preProcess = c("center", "scal
e", "pca"))
confMatKNN <- confusionMatrix(predict(trained_model_KNN, testing),testing$classe)
confMatKNN$overall[1:4]
```

```
##      Accuracy         Kappa AccuracyLower AccuracyUpper
##     0.9637031     0.9540739     0.9580831     0.9687619
```

```
confMatKNN$byClass[ ,1:2]
```

```
##          Sensitivity Specificity
## Class: A   0.9870968   0.9903106
## Class: B   0.9378293   0.9886220
## Class: C   0.9450292   0.9846876
## Class: D   0.9664179   0.9931707
## Class: E   0.9700333   0.9977517
```

```
results <- as.data.frame(confMatKNN$table)
ggplot() + geom_point(data = results, aes(x = Prediction,y = Reference, size = Freq)) + scale_si
ze(breaks = c(20,50,100,200,500,700,1000), range = c(0, 30))
```



Overall model statistics are satisfactory, with accuracy above 95%, we will use the model on a final test data.

Prediction on test data

```
final_test_raw <- read.csv("pml-testing.csv")
final_test <- final_test_raw
final_test <- final_test[,names(final_test) %in% names(training)]
predicted_results <- predict(trained_model_KNN, final_test)
```

Predicted results for each person:

```
cbind.data.frame(final_test_raw$user_name, predicted_results )
```

```
##    final_test_raw$user_name predicted_results
## 1                     pedro                 B
## 2                    jeremy                 A
## 3                    jeremy                 B
## 4                    adelmo                 A
## 5                    eurico                 A
## 6                    jeremy                 E
## 7                    jeremy                 D
## 8                    jeremy                 B
## 9                  carlitos                 A
## 10                  charles                 A
## 11                 carlitos                 D
## 12                   jeremy                 C
## 13                   eurico                 B
## 14                   jeremy                 A
## 15                   jeremy                 E
## 16                   eurico                 E
## 17                    pedro                 A
## 18                 carlitos                 B
## 19                    pedro                 B
## 20                   eurico                 B
```