Project 2, STAT 452

Your Name

Due: Friday, April 28, 2023

Instructions: This assignment should be completed using R Markdown and submitted to Canvas in PDF or HTML format.

```
# load packages
library(tidyverse)
library(rpart)
library(ranger) # fast implementation of random forests
```

For this project, you will consider the Fashion MNIST data set, which consists of a training set of 60,000 images and a test set of 10,000 images. As the name suggests, this data set is similar in structure to the original MNIST data set (lecture 20), except the images are of different articles of clothing, rather than handwritten digits. The response variable consists of the following 10 categories:

- 0 T-shirt/top
- 1 Trouser
- 2 Pullover
- 3 Dress
- 4 Coat
- 5 Sandal
- 6 Shirt
- 7 Sneaker
- 8 Bag
- 9 Ankle boot

Additional information about the data set is provided on Kaggle:

https://www.kaggle.com/datasets/zalando-research/fashionmnist (https://www.kaggle.com/datasets/zalando-research/fashionmnist)

Import Data Set

Run the following commands to load the training and test set into R:

```
fmnist_train <- read_csv("fashion-mnist_train.csv")
fmnist_test <- read_csv("fashion-mnist_test.csv")
```

The data frame `fmnist_train` contains 60,000 rows by 785 columns. Each row is an image, which has $28 \times 28 = 784$ pixels. The first column is the class label (response variable). The remaining 784 columns are the pixels (with the darkness of each pixel represented as a number between 0-255).

```
dim(fmnist_train)
```

```
## [1] 60000   785
```

```
table(fmnist_train$label)
```

```
##
##    0    1    2    3    4    5    6    7    8    9
## 6000 6000 6000 6000 6000 6000 6000 6000 6000 6000
```

The other data frame with the test set images, `fmnist_test` , has similar structure.

```
dim(fmnist_test)
```

```
## [1] 10000   785
```

```
table(fmnist_test$label)
```

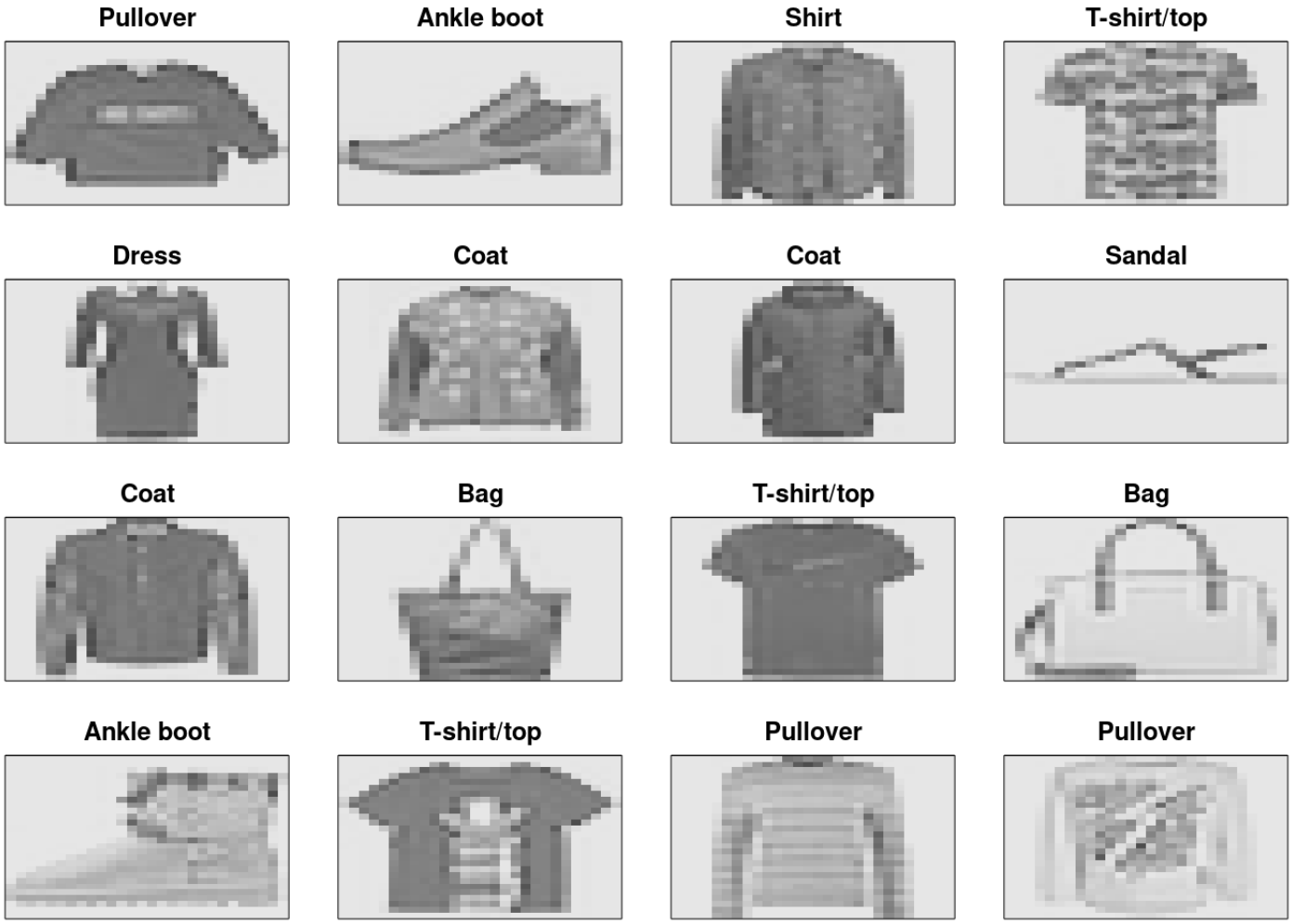
```
##
##    0    1    2    3    4    5    6    7    8    9
## 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
```

Plotting Images

Run the code below to plot the first 16 images in the training set.

```
# vector with category names
class_names <- c("T-shirt/top", "Trouser", "Pullover", "Dress", "Coat", "Sandal", "Sh
irt", "Sneaker", "Bag", "Ankle boot")
```

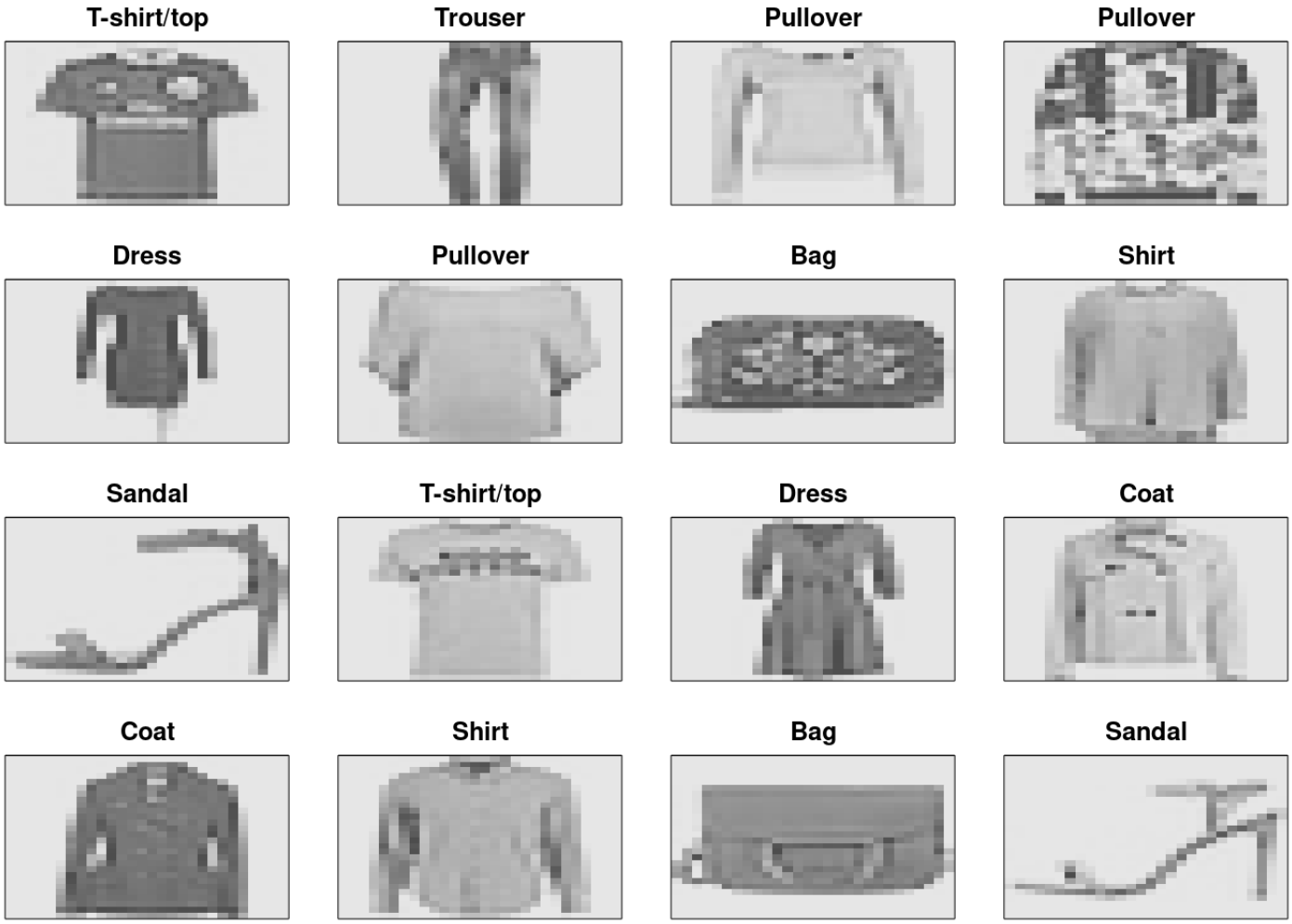
```
par(mfrow = c(4,4)) # make 4 by 4 grid
par(mar = c(1, 1, 2, 1)) # adjust margins around image
for(i in 1:16) {
  img <- as.numeric(fmnist_train[i, -1])
  image(matrix(img, 28, 28)[, 28:1],
    col = gray.colors(255, rev = TRUE), xaxt = "n", yaxt = "n",
    main = class_names[fmnist_train$label[i] + 1]) # add 1 since labels start at 0
}
```



Question 1

Modify the code above to plot the first 16 images in the **test** set. Make sure to include a title with the category name at the top of each image.

```
par(mfrow = c(4,4)) # make 4 by 4 grid
par(mar = c(1, 1, 2, 1)) # adjust margins around image
for(i in 1:16) {
  img <- as.numeric(fmnist_test[i, -1])
  image(matrix(img, 28, 28)[, 28:1],
    col = gray.colors(255, rev = TRUE), xaxt = "n", yaxt = "n",
    main = class_names[fmnist_test$label[i] + 1]) # add 1 since labels start at 0
}
```



Decision Tree Model

Run the code below to fit a classification tree model using the training set data (this may take several minutes to run). Make sure to fill in any “blanks” prior to running the code.

```
tree1 <- rpart(label ~ ., data = fmnist_train, method = "class")
```

Next, make predictions for the image labels on the test set, and compute the confusion matrix.

```
pred_tree1 <- predict(tree1, newdata = fmnist_test, type = "class")
# confusion matrix
cm <- table(predicted = pred_tree1, actual =fmnist_test$label )
addmargins(cm)
```

```
##          actual
## predicted    0    1    2    3    4    5    6    7    8    9  Sum
##       0    714    0   18   16    2    0  200    0    4    0  954
##       1      6  840   12   11    1   29   13    0   11    0  923
##       2     71    2  598   33   59    0  177    0   52    5  997
##       3    158  150   10  848  112   18  131    1   25    6 1459
##       4     16    1  335   67  796    1  449    0   88    1 1754
##       5     11    2    7    4    3  712   10   74   26   19  868
##       6      0    0    0    0    0    0    0    0    0    0    0
##       7      2    0    0    0    0  119    0  794   28   93 1036
##       8     17    3   19    0   27   52   20   18  758   42  956
##       9      5    2    1   21    0   69    0  113    8  834 1053
##       Sum 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 10000
```

Question 2

What is the accuracy (percent of images correctly classified) of the decision tree model on the test set?

```
# Accuracy
accuracy<-(714+840+598+848+796+712+0+794+758+834)/10000
accuracy
```

```
## [1] 0.6894
```

There is 68.94% that we correctly identify the label of clothes using decision tree model # Random Forest Model

Run the code below to fit a random forest model using the training set data (this may take several minutes to run). Make sure to fill in any “blanks” prior to running the code. Note that we need to set the argument `classification = TRUE` so that the `ranger()` function fits classification trees; otherwise, regression trees would be built, since the response variable is coded using numeric values (0-9).

```
set.seed(452) # for reproducibility
rf1 <- ranger(label ~ ., data = fmnist_train,
              num.trees = 200, mtry = 28,
              classification = TRUE)
```

```
## Growing trees.. Progress: 40%. Estimated remaining time: 48 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 9 seconds.
```

Next, make predictions for the image labels on the test set, and compute the confusion matrix.

```
p1 <- predict(rf1, data = fmnist_test)
pred_rf1 <- p1$predictions
```

```
# confusion matrix
cm1 <- table(predicted = pred_rf1, actual = fmnist_test$label)
addmargins(cm1)
```

```
##          actual
## predicted    0    1    2    3    4    5    6    7    8    9  Sum
##       0    866    2    9   15    1    0  168    0    1    0 1062
##       1      0  972    1    8    0    0    1    0    1    0  983
##       2     11    7  800    6   64    0   99    0    7    0  994
##       3     30   14   10  932   29    0   26    0    0    0 1041
##       4      0    1  114   21  858    0   76    0    2    0 1072
##       5      1    1    0    0    0  948    0   17    2    7  976
##       6     80    3   56   18   45    0  615    0    8    1  826
##       7      0    0    0    0    0   36    0  931    2   40 1009
##       8     12    0   10    0    3    3   15    0  976    3 1022
##       9      0    0    0    0    0   13    0   52    1  949 1015
##       Sum 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 10000
```

Question 3

What is the accuracy (percent of images correctly classified) of the random forest model on the test set? How does this compare with the single decision tree model?

```
# Accuracy
acc_rf<-(866+972+800+932+858+948+615+931+976+949)/10000
acc_rf
```

```
## [1] 0.8847
```

```
sum(diag(cm1))/10000
```

```
## [1] 0.8847
```

There is 88.47% that we correctly identify the label of clothes using random forest model. The random forest performs the better model than the decision tree since it has a higher accuracy.

Question 4

- What percent of the “Shirt” images are correctly classified by the random forest model?

```
# Shirt correctly identify (label 6)
acc_shirt<-625/1000
acc_shirt
```

```
## [1] 0.625
```

There is 62.5% that we correctly identify the label of shirt using random forest model

- What percent of the “Bag” images are correctly classified by the random forest model?

```
# Bag correctly identify (label 8)
acc_bag<-976/1000
acc_bag
```

```
## [1] 0.976
```

There is 97.6% that we correctly identify the label of bag using random forest model

Question 5

Fit another random forest model on the training set, but this time choose different values for `mtry` and `ntree`. How does changing the values of these tuning parameters affect the performance of the random forest classifier on the test set images?

```
set.seed(452) # for reproducibility
rf2 <- ranger(label ~ ., data = fmnist_train,
              num.trees = 500, mtry = 70,
              classification = TRUE)
```

```
## Growing trees.. Progress: 7%. Estimated remaining time: 9 minutes, 11 seconds.
## Growing trees.. Progress: 15%. Estimated remaining time: 6 minutes, 30 seconds.
## Growing trees.. Progress: 22%. Estimated remaining time: 5 minutes, 57 seconds.
## Growing trees.. Progress: 29%. Estimated remaining time: 5 minutes, 24 seconds.
## Growing trees.. Progress: 36%. Estimated remaining time: 4 minutes, 50 seconds.
## Growing trees.. Progress: 44%. Estimated remaining time: 4 minutes, 8 seconds.
## Growing trees.. Progress: 52%. Estimated remaining time: 3 minutes, 35 seconds.
## Growing trees.. Progress: 59%. Estimated remaining time: 3 minutes, 3 seconds.
## Growing trees.. Progress: 66%. Estimated remaining time: 2 minutes, 28 seconds.
## Growing trees.. Progress: 74%. Estimated remaining time: 1 minute, 56 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 1 minute, 22 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 53 seconds.
## Growing trees.. Progress: 96%. Estimated remaining time: 19 seconds.
```

Next, make predictions for the image labels on the test set, and compute the confusion matrix.

```
p2 <- predict(rf2, data = fmnist_test)
pred_rf2 <- p2$predictions
```

```
# confusion matrix
cm2 <- table(predicted = pred_rf2, actual = fmnist_test$label)
addmargins(cm2)
```

```
##          actual
## predicted    0    1    2    3    4    5    6    7    8    9  Sum
##       0    861    2    7   17    1    0  161    0    1    0 1050
##       1      0  976    1    5    0    0    1    0    1    0  984
##       2     12    3  811    9   61    0   97    0    8    0 1001
##       3     31   13   12  929   28    0   27    0    0    0 1040
##       4      1    0  109   23  867    0   79    0    3    0 1082
##       5      1    1    0    0    0  948    0   13    2    8  973
##       6     83    5   50   17   40    0  622    0    8    1  826
##       7      0    0    0    0    0   35    0  936    2   38 1011
##       8     11    0   10    0    3    5   13    0  975    2 1019
##       9      0    0    0    0    0   12    0   51    0  951 1014
##       Sum 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 10000
```

```
# Accuracy
acc_rf2<-(861+976+811+929+867+948+622+936+975+951)/10000
acc_rf2
```

```
## [1] 0.8876
```

```
# another way to calculate accuracy
sum(diag(cm2))/10000
```

```
## [1] 0.8876
```

There is 88.76% that we correctly identify the label of clothes using random forest model with num.trees = 500, mtry = 70. This is a little higher than the first random forest model (88.47%) which mean the second random forest performs better than the first one.