PML Course Project

Yashan Wang

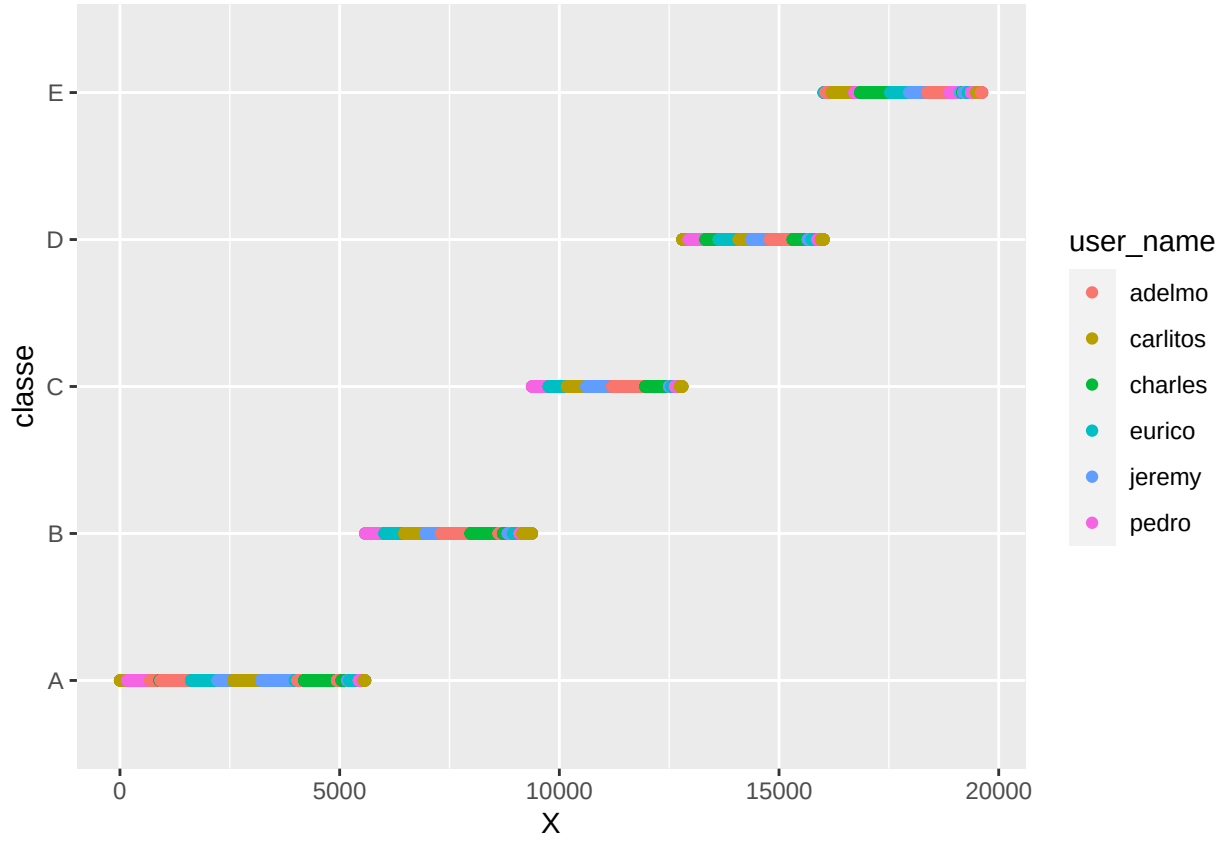
2021/1/24

Loading data and exploratory analysis

```
pml.training<-read.csv("pml-training.csv")
pml.testing<-read.csv("pml-testing.csv")
summary(pml.training$classe)
```

```
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

```
qplot(X, classe, data = pml.training, colour= user_name)
```



The outcome we were interested was the classe which was a categorical varible that had 5 levels and each of the levels contained the 6 users. A **random forest** could be used to predict which category each obs belongs to.

However, there were 159 covariates. By observering the variables, we could see that some of them had less observations and were generated by other variables, such as avg_roll_belt or var_yaw_arm (1). So I decided to remove those variables and let machine learning algorithm to create variables using a **principal components analysis** as preprocess.

Building model for the project

1. Data cleaning and creating partition

First, redundant variables were removed from the training set. 3/4 of the trainingset was used to train the model and 1/4 was used as validation for calculating out of sample error.

```
pml.training1<-pml.training[,c(2:11,37:49,60:68,84:86,102,113:124,140,151:160)]
set.seed(666)
inTrain<-createDataPartition(y=pml.training1$classe,p=0.75,list = F)
training<-pml.training1[inTrain,]
validation<-pml.training1[-inTrain,]
dim(training);dim(validation)
```

```
## [1] 14718    59
```

```
## [1] 4904   59
```

2. Configure parallel processing in trControl

To improve the processing of random forest models in caret package, Dr. Leonard Greski (2) offered a parallel processing.

```
library(foreach);library(iterators)
library(parallel);library(doParallel)
cluster <- makeCluster(detectCores() - 1)
registerDoParallel(cluster)
fitControl <- trainControl(method = "cv",number = 5, allowParallel = TRUE)
```

3. Developing random forest model

Here I used **train** function in **caret** package , **trainControl** object we built, and **pca preProcess** to build a random forest model.

```
fit2<-train(classe ~.,data = training,
            method="rf", preProcess="pca",
            trControl= fitControl)
```

4. De-register parallel processing cluster

```
stopCluster(cluster)
registerDoSEQ()
```

5. Using the model to predict validation set

Checking model accuracy on training set.

```
confusionMatrix.train(fit2)
```

```
## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    A    B    C    D    E
##          A 28.1  0.3  0.0  0.0  0.0
##          B  0.2 18.8  0.2  0.0  0.0
##          C  0.1  0.3 17.1  0.6  0.0
##          D  0.0  0.0  0.1 15.7  0.2
##          E  0.0  0.0  0.0  0.1 18.2
##
##  Accuracy (average) : 0.9781
```

Checking out of sample accuracy on validation set.

```
predValid<-predict(fit2,validation)
confusionMatrix(predValid, validation$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1389    5    0    0    0
##          B    5  941   21    0    0
##          C    1    3  831   23    0
##          D    0    0    3  778    9
##          E    0    0    0    3  892
##
## Overall Statistics
##
##                Accuracy : 0.9851
##                  95% CI : (0.9813, 0.9883)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9812
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9957   0.9916   0.9719   0.9677   0.9900
## Specificity            0.9986   0.9934   0.9933   0.9971   0.9993
## Pos Pred Value         0.9964   0.9731   0.9685   0.9848   0.9966
## Neg Pred Value         0.9983   0.9980   0.9941   0.9937   0.9978
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2832   0.1919   0.1695   0.1586   0.1819
## Detection Prevalence   0.2843   0.1972   0.1750   0.1611   0.1825
## Balanced Accuracy      0.9971   0.9925   0.9826   0.9824   0.9946
```

Predict test dataset

At last, I implemented my model to predict the classe of the testing set given.

```
pml.testing1<-pml.testing[,c(2:11,37:49,60:68,84:86,102,113:124,140,151:159)]
pml.testing1[,"classe"]<-0
pred<-predict(fit2,pml.testing1)
pred
```

```
##  [1] B A A A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

Reference:

1. Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.
2. [https://github.com/lgreski/datasciencectacontent/blob/master/markdown/pml-randomForestPerformance.md]