machine learning

Julian Zamudio

12/22/2020

Summary

Colecting the data

Preliminary data analysis

cleaning Data

All the incomplete data as NA values and non- defined values were excluded from the training data set. The variables to be used are the variables with values in the testing data set, therefore the columns with NA values are removed from the testing and training data set.

```
library(caret)
```

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

```
set.seed(34567)
testing.1 <-testing[colSums(!is.na(testing)) > 0]
testing.1 <- testing.1[,-c(1:7)]
name.1 <- c(names(testing.1), "classe")
invalid <- complete.cases(training)
training <-training[invalid,]
name.2 <- names(training)
training.1 <- training[,which(name.2 %in% name.1)]
name.3 <- names(training.1)
miss.1 <- name.1[-which(name.1 %in% name.3)]
testing.1 <- testing.1[,-which(name.1 %in% miss.1)]
```

Preliminary data analysis

```
analysis <- rbind(apply(training.1[,-53], 2, mean),apply(training.1[,-53], 2, sd))
analysis <- rbind(analysis, analysis[2,]/analysis[1,])
rownames(analysis) <- c("mean", "sd", "RSD")
hvariance <- analysis[,which(analysis[3,] > 0.5)]
length(hvariance[1,])
```

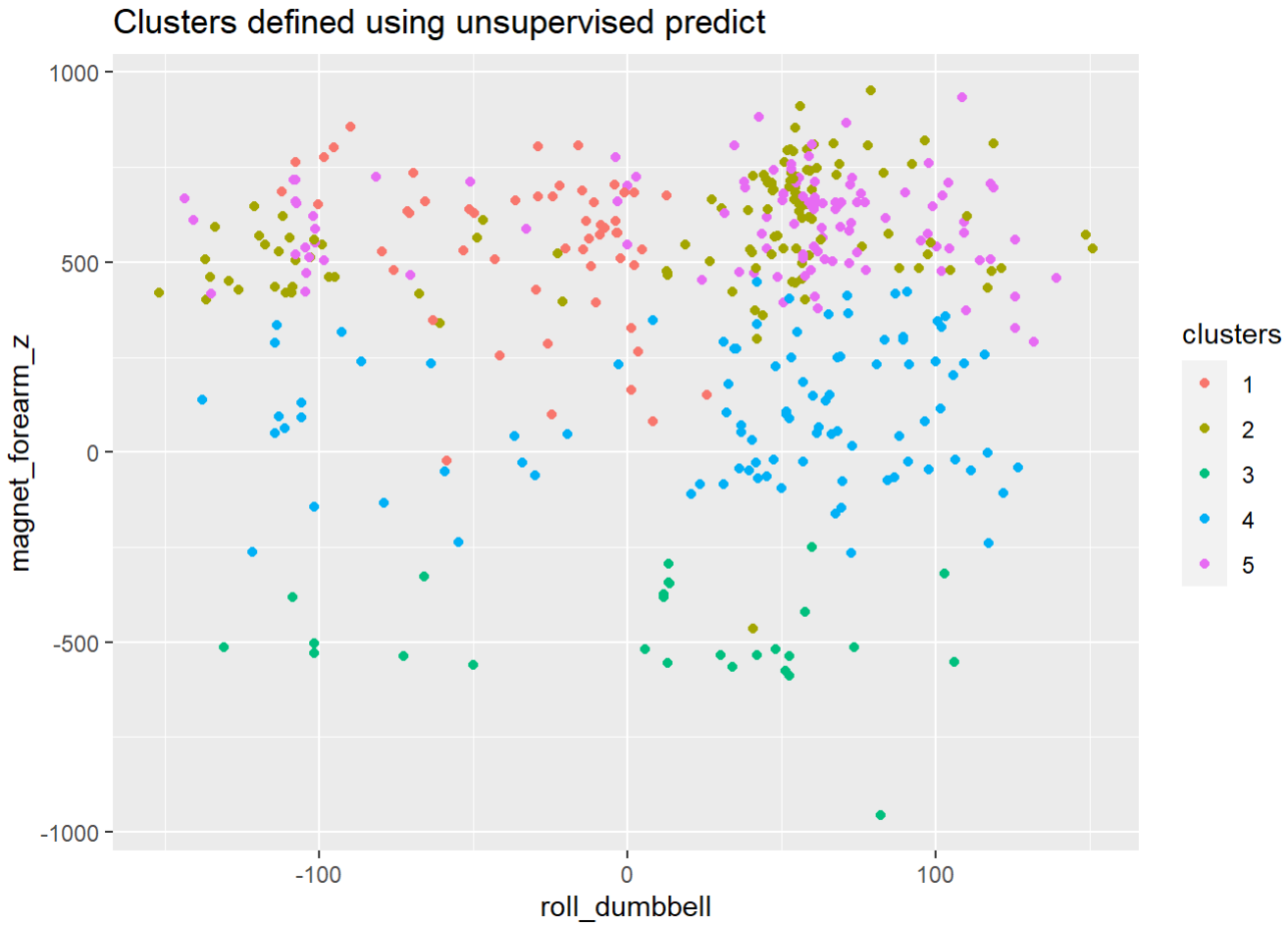
```
## [1] 29
```

the average, variance and standard deviation for the variables in the training data set are calculated. From the 52 variables 29 have strong variance. Therefore, I will use methods the methods: predicting with trees, random forest, bagging with tree bag and boosting with tree.

Predicting the class

testing unsupervised

```
Kunsuper <- kmeans(subset(training.1, select = -c(classe)), centers=5 )
training.1$clusters <- as.factor(Kunsuper$cluster)
p<- qplot(roll_dumbbell,magnet_forearm_z,col= clusters, data = training.1,main = "Clusters defin
ed using unsupervised predict")
p
```




```
modunsuper <- train(clusters ~ ., data = subset(training.1, select= -c(classe)), method = "rpar
t")
modunsuper
```

```
## CART
##
## 406 samples
##  52 predictor
##   5 classes: '1', '2', '3', '4', '5'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 406, 406, 406, 406, 406, 406, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.07342657  0.8807186  0.8410322
##   0.15384615  0.7669550  0.6844319
##   0.32342657  0.5831011  0.4170722
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.07342657.
```

```
table(predict(modunsuper,training.1), training.1$classe)
```

```
##
##      A  B  C  D  E
##   1 12  7  8  7 10
##   2 60 19 36 14 15
##   3  0  0  0  0  0
##   4 19 30  6 17 20
##   5 18 23 20 31 34
```

Discussion

No forecast was done in the data sets. it is defined not relevant in this study. The results are not shown, but random forest and linear regression have low accuracy. Indeed, those models were discarded. Then the focus was in models to predict factor outcome. The boosting model(also results are not shown) was tested first because it was found high variance in 0.55 of the predictors in the training data base. It was expected to have compensation for the predictors with large variance using that model.The results seems to have acceptable accuracy at 150 number of trees with 0.70.

The unsupervised prediction was use, because create cluster based on the predictors. The approach seems to be good to predict factors. Using 5 centers the accuracy is 0.73. There is a improvement regarding to the boosting.Also, become closer to the reported by Velloso et al. Results are not shown, a final option was tested, using the combined models of boosting and unsupervised models, but the accuracy was below 0,30.Therefore it was discarded. The cross validation was not possible, but the accuracy of the methods it is use to define the best fitting to the data set. In the figure, is shown the final clusters for the training data set. there are not spatially separated indicating that the selectivity it is not good enough. Also, it is shown on the comparison table. It is expected the error in the sample be high, there is no way to calculated the actual error in the prediction.

Prediction testing values

```
testCluster <- predict(modunsuper,testing.1)
qplot(x = c(1:20), y = testCluster, geom = "point", main = "Predicted classe for the training se
t", xlab = "Number", ylab = "classe")
```