Classification Project

22 February 2015

Summary

The aim of this document is not only to find the "best" classification, but to explore few models and their constraint. We shall use three models:

1. Linear Discriminant Analysis.
2. Recursive Partitioning, in other words recursive tree with the rpart package. We shall explore bagging as well with the same tree method.
3. Random Forest, which is close to Bagging.

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

Pre Processing Before to process the data, the training and test set have to be pre-process. Based on the initial EDA it appears:

1- The first seven columns related to information about individuals are not relevant for the classification and therefore removed

2- Some exercises raws columns are populated only at special stage, when the individual will change class. We make the hypothesis that it is a special exercise or summary and therefore is one exception. We remove those records.

3- Some columns of type characters are empty, therefore we remove them

4- The class columns is our output and we transform it as factor

5- Check distribution, scale and skewness of the variables as we want to use multiple models, such as LDA, which is sensitive to outliers due to the fact that is it is based on euclidian distance (L2 norm).

After this processing the number of training observations have been reduces to `19216` and the number of predictors to `52`. As we want to use multiple models and some are more sensitives to differences in scales or skewness. It appears in the following plot that we have extreme skewness with five variables, that we should explore.

Based on the previous plot we have the following variable to assess:

1.`gyros_dumbbell_x`

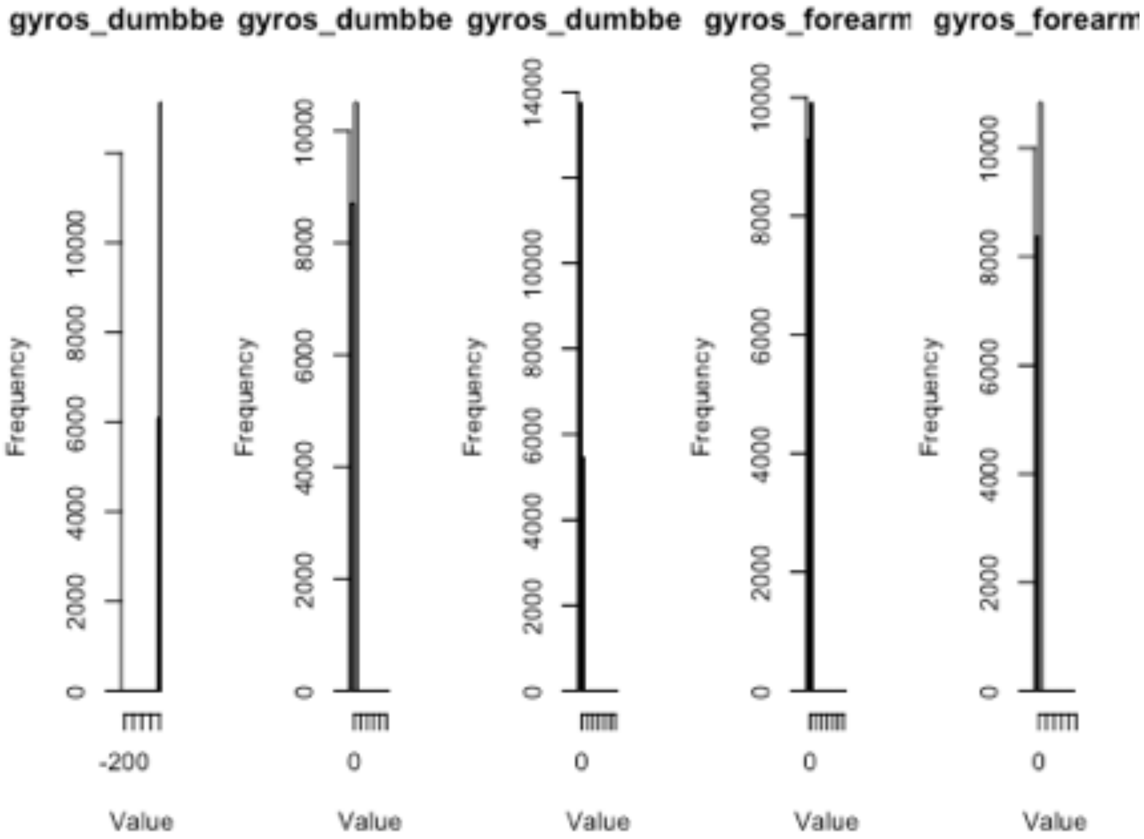
2.`gyros_dumbbell_y`

3.`gyros_dumbbell_z`

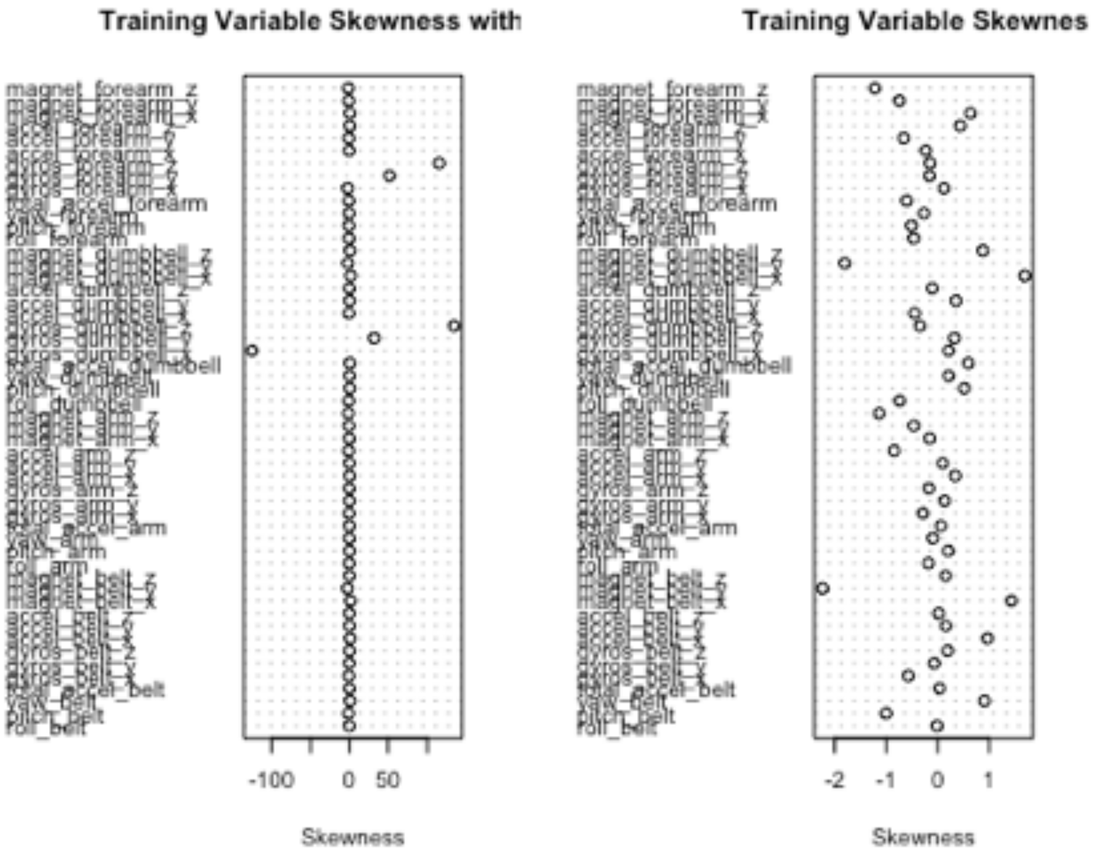
4.`gyros_forearm_y`

5.`gyros_forearm_z`

Distribution of the five variables with high skewness.



It appears that we have one outlier, creating extreme skewness with index `5270`. This observation is removed from our data set and the difference is presented in the plot below. We shall not display all the distributions here, the five presented above are now close to normal.



Models Building As mentioned before we shall use multiple models, check the results and then select the right model. Based on the model, the testing and training set will change. As we deal with classification of multiple categories, tree type of models could be good candidates. To be able to compare, we use linear discriment as well.

Linear Discriminant Analysis With linear discriminant analysis, we make the assumption, that we could have a linear relation between the observation and the probability assign to the four categories of class. The result of the prediction is with a probability > .5. The training set used is made of .7 of the original data set

```
set.seed(533)
ldamodel <-lda(classe~.,data=trainingset)
predicclass <-predict(ldamodel, testingset)
ldaresults <-confusionMatrix(predicclass$class, testingset$classe)
ldaaccuracy <-ldaresults$overall[1]
```

This first simple model delivers an accuracy of `0.703106`.

Simple recursive tree Before to use "black box" methods, which we will give better prediction by an increase we do not know, we use simple recursive tree. We shall define the complexity level to arrive to the best accuracy with this type of tree or more exactly to have the most adequate pruning. We shall use cross validation method with ten folds.

See caret for details about the trainControl caret function.

```
numFolds = trainControl( method = "cv", number = 10 )
rpartcomplexity <-expand.grid( .cp = seq(0.01,0.5,0.01))  # Set
various values for the cp paramenters of rpart
result<-train(classe~., data = trainingset, method = "rpart",
trControl = numFolds, tuneGrid = rpartcomplexity )  ## this function
will take time !!
```

```
## Loading required package: rpart
```

To build the model, we should use a cp value of `0.01`, which in this special case is the default value of rpart.control().

```
rpartmodel <-rpart(classe~., data = trainingset, method="class")
trainingpredicttree <-predict(rpartmodel, data= trainingset,
type="class")
trainingtreeaccuracy <-confusionMatrix(trainingpredicttree,
trainingset$classe)$overall[1]
predicttree <-predict(rpartmodel, newdata= testingset, type="class")
rpartresults <-confusionMatrix(predicttree, testingset$classe)
rpartaccuracy <-rpartresults$overall[1]
```

Using the rpart with K fold of ten the accuracy is `0.7213257`, which is a significant increase.

Bagging with rpart

As tree have tendancy to overfitting as previously, our accuracy with the training set was of `0.7285905` and of`0.7213257`` with the testing set. Bagging also known as Bootstrap aggregation is used with the same recursive tree method. We shall use the default parameters of the function bagging() from ipred package, 25 bootstrap with out of bag estimate of error.

```
bagmodel <- bagging(classe ~ ., data = trainingset, coob=TRUE)  ## As
cross validation takes time
predictbagging <-predict(bagmodel, newdata=testingset)
```

```
baggingaccuracy<-confusionMatrix(predictbagging, testingset$classe)
$overal[1]
```

With this method we now reach one accuracy of `0.9857713`, which is a big gain comare to the two previous methods.

Random forest

One other method, which is not part of ensemble is the random forest. This method includes bagging as well as random feature selections.

```
rfmodel10trees <-train(classe~., data=trainingset, method="rf",
ntree=10)
predictrf10 <-predict(rfmodel10trees, newdata= testingset)
rf10treesaccuracy <-confusionMatrix(predictrf10,testingset$classe)
$overal[1]
```

The new accuracy with random forest and only ten tress is `0.9897623` compare to `0.9857713` for bagging only.

Conclusion

As expected it appears that the accuracy of the prediction increases with the complexity of the method used. The table below summarises the levels of accuracy achieved by various methods. Based on the results, we take the random forest method to do prediction of type competition. If the aim was to have on explicit model a good enough prediction rather than nothing the recursive tree could be a good candidate due to its simplicity.

Method	Accuracy
Linear Discriminant	0.703106
Recursive Tree	0.7213257
Bagging Tree	0.9857713
Random Forest 10	0.9897623