PML_Project

Libardo Lopez

Friday, October 24, 2014

This project models and predicts the user activity from wearable sensors.

Preprocessing

We load the training and test data sets. Initially there are 160 variables.

All preprocess methods would be apply to both datasets.
In order to reduce the variables for the model, we follow this strategy
Missing values: Verify columns with nas < 10%; if so, make Imputation. If not, elliminate them.
Near Zero Variance: If so, elliminate them.
Correlated Predictors: If abs(correlation) > 0.86, elliminate them.

We elliminate 13 high correlated variables.
I verify linear combination of variables but the result was none.
At the end, we have 44 predictors and split it into 70% for training and 30% for testing.
(You can see all the **R code** in the .Rmd File)

Modeling

I try with random forest because it has very good performance with classification tasks.
I run **rf** alone and with **cv**; but **cv** is very time consuming. The results are very similar.
Finally, i run an option with 7 variables randomly sampled as candidates at each split (mtry=7).

```
random.forest <- train(training[,-45], training$classe, tuneGrid=data.frame(mtry=7), trControl=trainControl(method="none"))
```

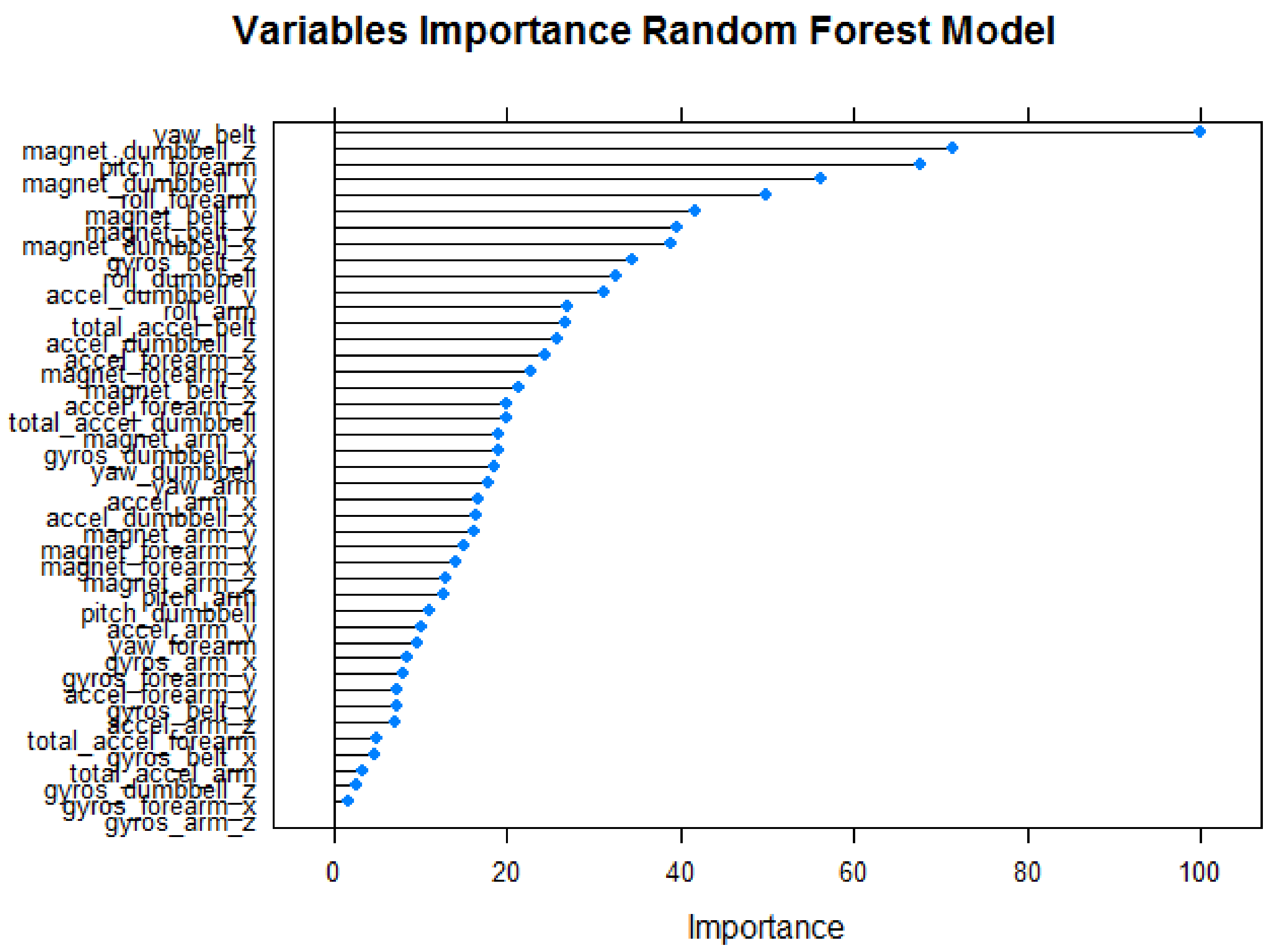
Now, we made prediction with the model on testing file and verify results

```
confusionMatrix(predict(random.forest, newdata=testing[,-45]), testing$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1672    9    0    0    0
##          B    2 1125   13    0    0
##          C    0    5 1013   12    1
##          D    0    0    0  951    1
##          E    0    0    0    1 1080
## 
## Overall Statistics
## 
##                Accuracy : 0.993
##                  95% CI : (0.99, 0.995)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
## 
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.999    0.988    0.987    0.987    0.998
## Specificity             0.998    0.997    0.996    1.000    1.000
## Pos Pred Value          0.995    0.987    0.983    0.999    0.999
## Neg Pred Value          1.000    0.997    0.997    0.997    1.000
## Prevalence              0.284    0.194    0.174    0.164    0.184
## Detection Rate          0.284    0.191    0.172    0.162    0.184
## Detection Prevalence    0.286    0.194    0.175    0.162    0.184
## Balanced Accuracy       0.998    0.992    0.992    0.993    0.999
```

The results are excellent with a very hig accuracy, we do not have overfitting.
Now, we can see the importance of each variable for the model.

```
plot(varImp(random.forest), main ="Variables Importance Random Forest Model")
```



Prediction

```
Model_evaluation <- predict(random.forest, test.file)
summary(Model_evaluation)
```

```
## A B C D E
## 7 8 1 1 3
```

Conclusions

- Preprocess is a very important task in order to obtain good models
- Random Forest has a very good performance on classification tasks
- If we need to improve our model, we can think about uses PCA and compare results (not included)