Machine Learning Project

Nicholas Cemalovic

7/18/2020

Goal:

Build and cross-validate a machine learning algorithm to predict the class of a dumbell curl movement from biomotion sensor data collected from Velloso et al. 2013. The workflow, accruacy and error analysis, and test prediction is below:

Data Cleaning:

Removing unnessecary predictors like name and time stamp (the first 7 columns) along with filtering out all NA and zero variance predictors from both the training and testing data

```
set.seed(1)

training <- training[,8:length(colnames(training))]
testing <- testing[,8:length(colnames(testing))]

training <- training[,colSums(is.na(training)) == 0]
testing <- testing[,colSums(is.na(testing)) == 0]

no_var <- nearZeroVar(training, saveMetrics = TRUE)
if (sum(no_var$no_var) > 0) {
  training <- training[,no_var$no_var == FALSE]
}
```

Data Partition:

Splitting the cleaned training data into a 70:30 new split to model and validate

```
training_intermediate <- createDataPartition(training$classe,
                                     p = 0.70,
                                     list = FALSE)
training_official <- training[training_intermediate, ]
```

```
## Warning: The ‘i‘ argument of ‘‘[‘()‘ can’t be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
```

```
training_cross_valid <- training[-training_intermediate, ]
```

Model Development: Random Forest with 5-fold cross validation

Because it would be statistically taxing to run regressions and analyses of the 152 possible covariates, I've chosen a Random Forest model to automatically identify significant covariates, with reduced variance by averaging the outcome of each decision tree.

```
model <- train(classe ~ .,
               data = training_official,
               method = "rf",
               trControl = trainControl(method = "cv", 5),
               ntree = 251)
model
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10988, 10988, 10991, 10991, 10990
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9900266  0.9873831
##   27    0.9906820  0.9882126
##   52    0.9819475  0.9771615
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

The seleted optimal model (mtry 27) has an accuracy of ~99.1% from the training_official data (a major subset of the original training data) The next step is to cross validate this model with the smaller subset from the training set, now named training_cross_valid.

Model Cross-Validation

```
validation_prediction <- predict(model, training_cross_valid)
confusionMatrix(factor(training_cross_valid$classe), factor(validation_prediction))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1671    2    1    0    0
##          B    3 1135    0    1    0
##          C    0   12 1007    7    0
##          D    0    0   14  949    1
##          E    0    0    0    4 1078
##
## Overall Statistics
##
##                Accuracy : 0.9924
##                  95% CI : (0.9898, 0.9944)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9903
##
```

```
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9878   0.9853   0.9875   0.9991
## Specificity            0.9993   0.9992   0.9961   0.9970   0.9992
## Pos Pred Value         0.9982   0.9965   0.9815   0.9844   0.9963
## Neg Pred Value         0.9993   0.9971   0.9969   0.9976   0.9998
## Prevalence             0.2845   0.1952   0.1737   0.1633   0.1833
## Detection Rate         0.2839   0.1929   0.1711   0.1613   0.1832
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9987   0.9935   0.9907   0.9922   0.9991
```

Based off of the Confusion Matrix, the model had high success is prediction, quantified with an overall accuracy of ~99.2%. While this is rare that out of sample accuracy is higher than training set accuracy, it is worth nothign that precision and accuracy metrics varied slightly by class. This analysis shows little out of sample error and supports moving to true prediciton on the testing dataset.

Testing the Model

```
test_results <- predict(model, newdata = testing)
test_results
```

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

The 20 observances in the official test data were predicted to the 5 levels above, with 100% accuracy according to the coursera quiz

Conclusion

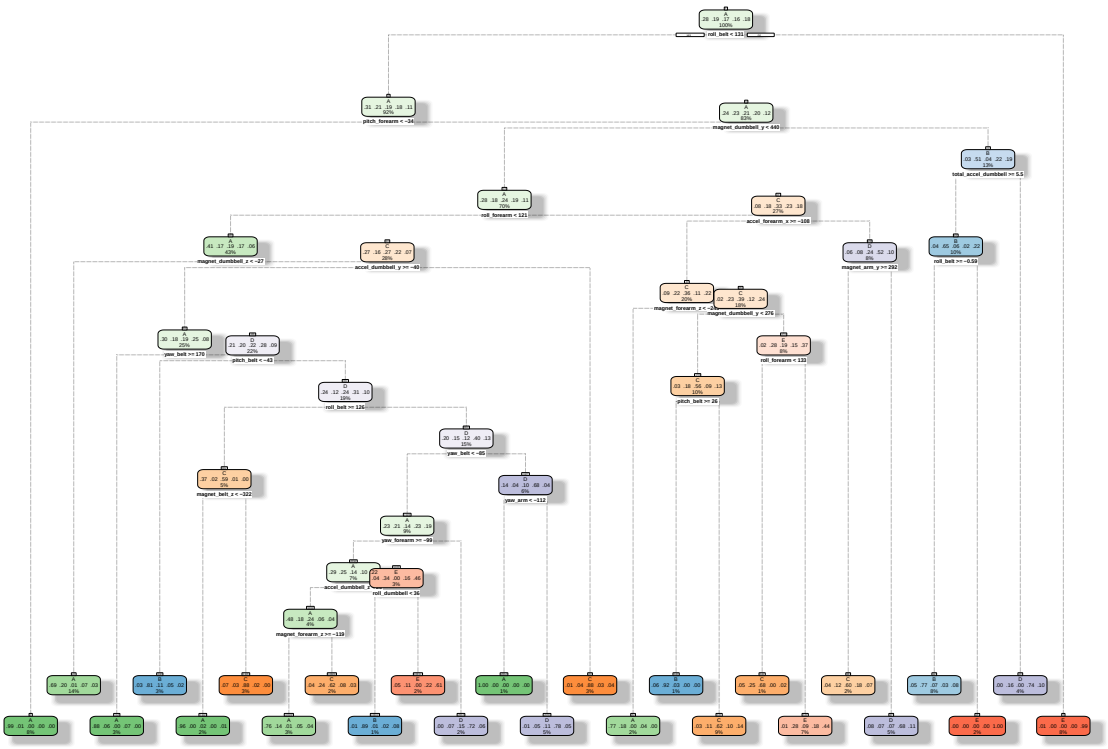
The Random Forest model with 5 folds was able to predict with an in-sample accuracy of ~99.1% and an out-of-sample validation accuracy of ~99.2%. When introduced to the test set, the model was able to predict all 20 observances with total accuracy. This model was likely most effective due to proper cleaning of insignificant covariates, the inclusion of all possible covariates (classe ~ .), and little overfitting. A decision tree averaged from all those created by randomForrests is shown in the appendix along with a reference to the data collected.

Appendix

Decision Tree of the Random Forest Model

```
tree_graph <- rpart(classe ~ ., data= training_official, method="class")
fancyRpartPlot(tree_graph)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Rattle 2020−Jul−19 13:35:49 nickcemalovic

Reference:

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.