Practical Machine Learning Project

Charles Vuono

September 1, 2019

Project Summary

The goal of this project is to develop a machine learning algorithm which predicts the manner in which a weightlifting excercise was performed on a scale A, B, C, D, E. The data includes nearly 20,000 observations of physical measurements from six subjects with the A-E scale indicated. In this project, we develop a cross-validated random forest machine learning algorithm on a set of training data which we can use to predict the scaled manner and determine the out-of-sample expected error rate.

Creating the training and testing data sets

Our first step is to set the seed for our analysis (for reproducibility) and load the training and testing data. We load the unprocessessed data with the following code.

```
set.seed(2233)
UnprocessedTraining <- read.csv("pml-training.csv",na.strings=c("NA","NaN", " ",""))
UnprocessedTesting <- read.csv("pml-testing.csv",na.strings=c("NA","NaN", " ",""))
```

We shall now investigate the properties of this data. Our first observation is that although there are 160 variables, only 60 of these are complete (no NA values) and the remaining 100 are almost entirely composed of NA values.

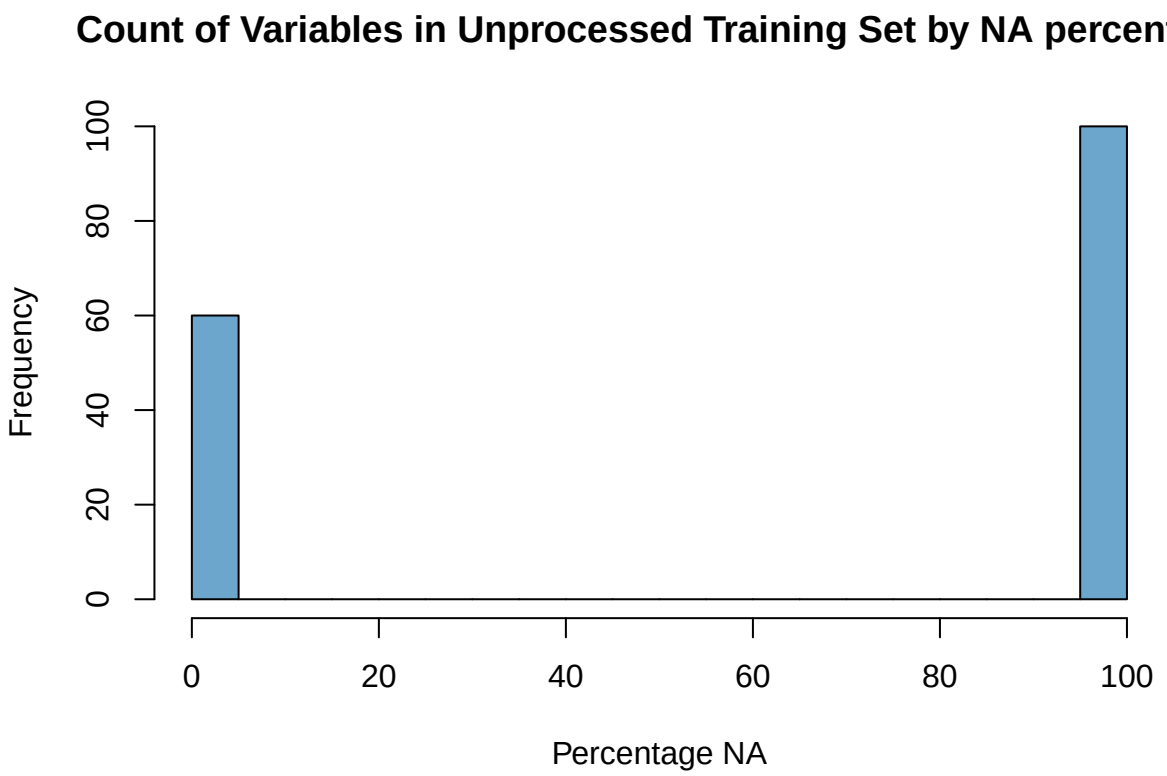
```
paste("Number of Variables: ", ncol(UnprocessedTraining))
```

```
## [1] "Number of Variables:  160"
```

```
csp<-colSums(is.na(UnprocessedTraining))/nrow(UnprocessedTraining)
paste("Number of Variables with no NA: ", sum(csp==0))
```

```
## [1] "Number of Variables with no NA:  60"
```

```
hist(csp*100, main="Count of Variables in Unprocessed Training Set by NA percentage",
     xlab="Percentage NA", col="skyblue3", breaks=20)
```



We further observe that a few variables are not relevant to our study: namely, the row indicator and the time variables. As such, we compose our training and testing data sets as follows:

```
iszero <- function (x){
  if(x==0){TRUE} else {FALSE}
}
keep <- function (x){
  cs <- colSums(is.na(x))
  sapply(cs, iszero)
}
NamesKeep <- keep(UnprocessedTraining)
NamesKeep[c(1,3,4,5)] <- FALSE  # Do not Keep row number or time elements
training <- UnprocessedTraining[,NamesKeep]
testing <- UnprocessedTesting[,NamesKeep]
```

Splitting the data and developing predictive models through Cross Validation.

In order to perform cross-validation on predictive models of our training set, we perform a k-fold cross validation with k = 5. We split the training data and prepare for the model generation with the following code:

```
train_control <- trainControl(method="cv", number=5)
```

We would like to compare the efficacy of several predictive models using the training data with the 5-fold cross validation. We consider several machine learning algorithms which are well-suited to an unsupervised learning classification problem such as ours. Namely, we consider Random Forest ("rf"), Naive Bayes ("nb") and Linear Support Vector Machine ("svm"). Each of these models will ge generated 5 times (one for each of

the partitions of the cross-validation being held out as a testing set). The code to run these models is as follows:

```
rfModel <-
  train(classe~., data=training, trControl=train_control, method="rf")
nbModel <-
  train(classe~., data=training, trControl=train_control, method="nb")
svmModel <-
  train(classe~., data=training, trControl=train_control, method="svmLinear")
```

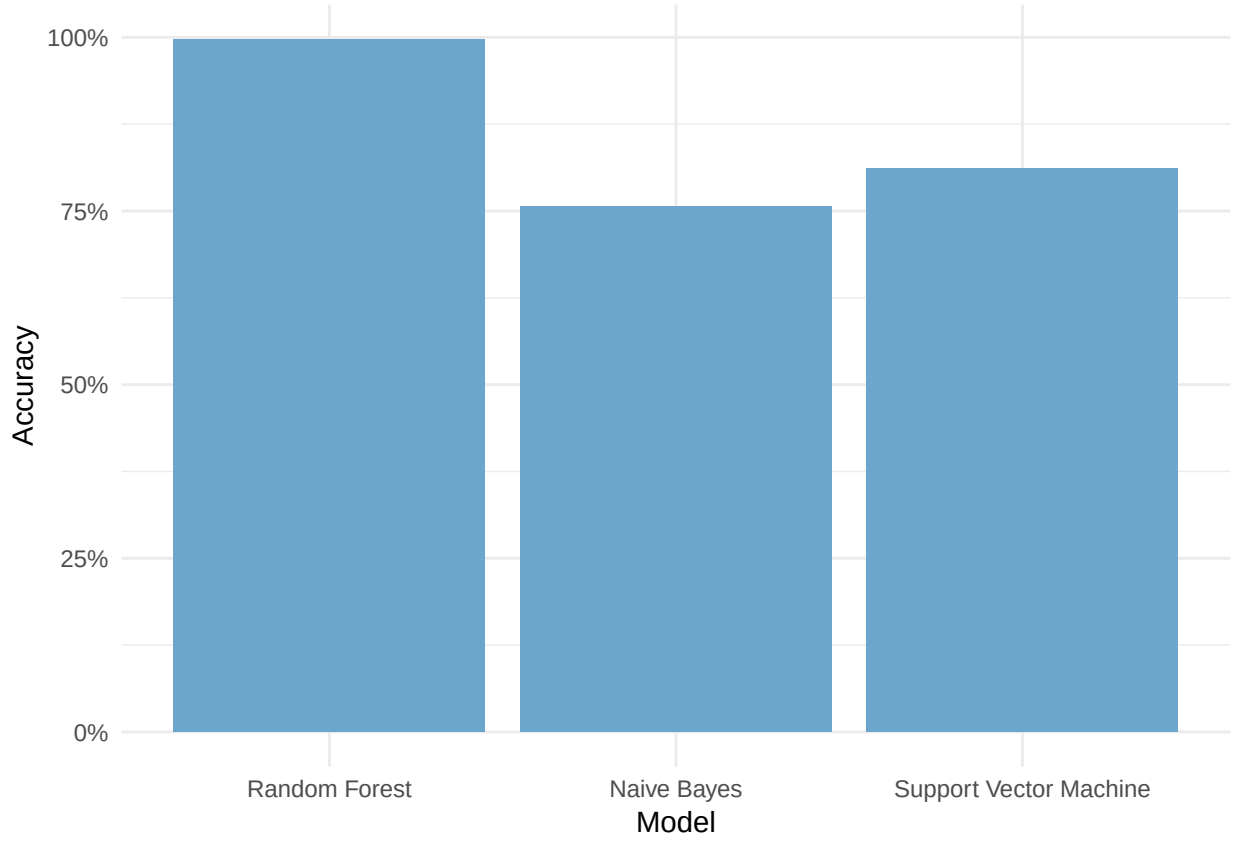
Final model selection

The predictive accuracy of each of our three models is given as follows:

```
rfAccuracy <- max(rfModel$results$Accuracy)
nbAccuracy <- max(nbModel$results$Accuracy)
svmAccuracy <- max(svmModel$results$Accuracy)
paste ("Random Forest: ", round(rfAccuracy,3),
       ", Naive Bayes: ", round(nbAccuracy, 3),
       " SVM: ", round(svmAccuracy,3))
```

```
## [1] "Random Forest:  0.997 , Naive Bayes:  0.756  SVM:  0.811"
```

```
accuracyDF<-data.frame(Model = c("Random Forest", "Naive Bayes", "Support Vector Machine"),
                       Accuracy=c(rfAccuracy, nbAccuracy, svmAccuracy))
accuracyDF$Model <- factor(accuracyDF$Model, levels = accuracyDF$Model)
p <- ggplot(data=accuracyDF, aes(x=Model, y=Accuracy)) +
    geom_bar(stat="identity", fill="skyblue3") +
    scale_y_continuous(labels = scales::percent) + theme_minimal()
p
```



The Random Forest model clearly has the highest accuracy and will be our choice for the final predictive model. We choose not to stack this model with the others given the very high accuracy (99.7%) and the rather low accuracies of the other models.

Using this model, we can expect an out-of-sample accuracy of 99.7%.

Using the model to predict the Testing dataset

Having chosen the 5-fold cross-validated Random Forest model, we can now make our predictions from the Testing dataset. We accomplish this thusly:

```
prediction <- predict(rfModel, testing)
prediction
```

```
##  [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
```

These predictions were confirmed to be 100% accurate.