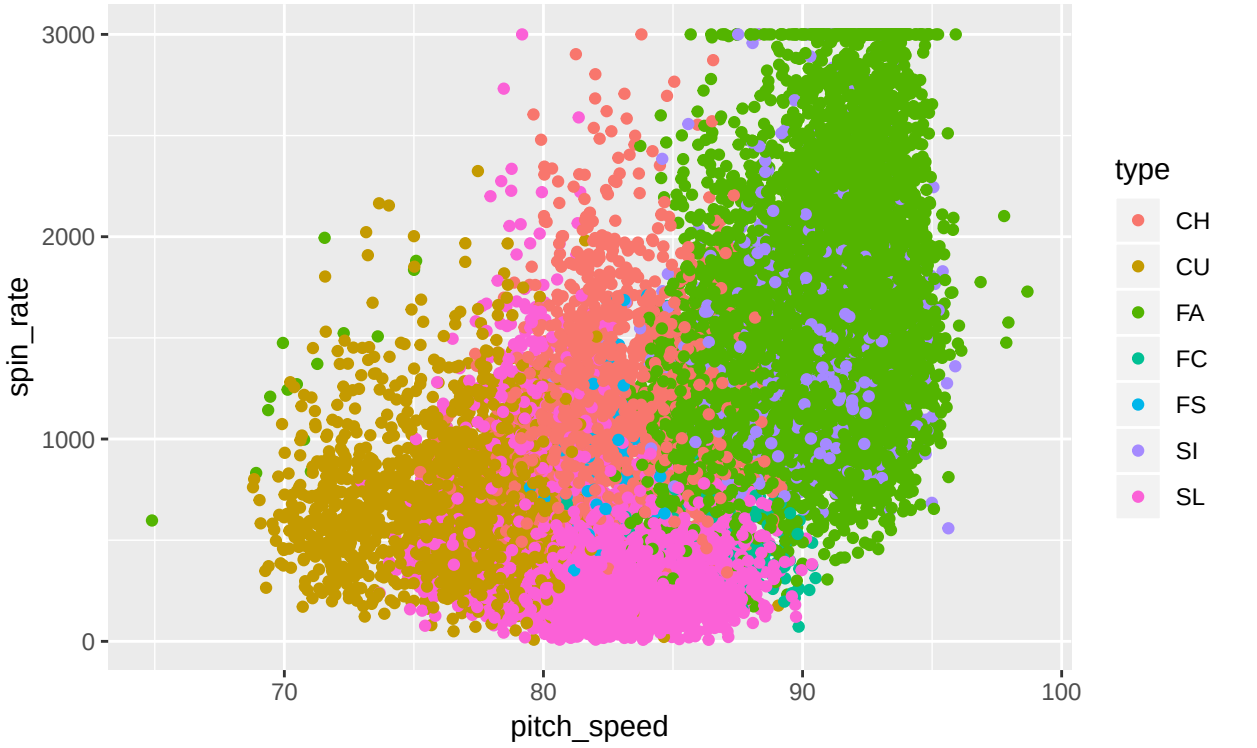
Pitch Classification – Mariners

Nolan McCafferty

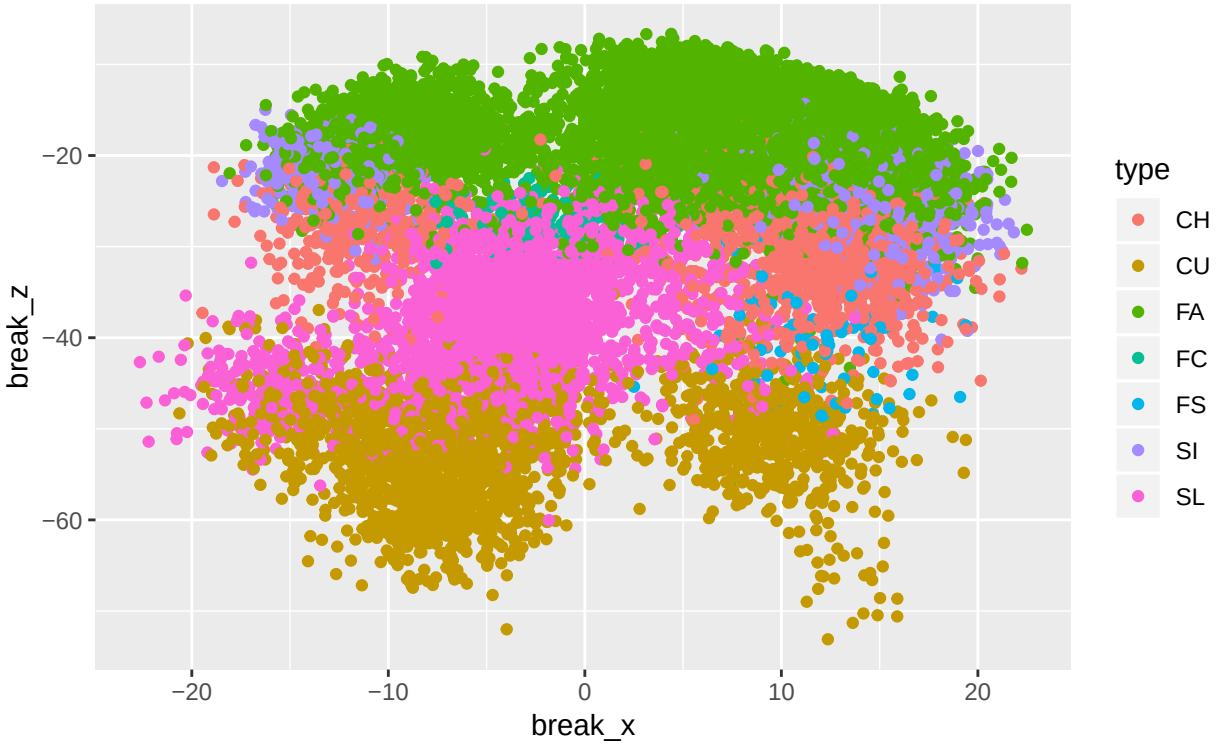
Before we create our pitch classification model, we will do some initial data exploration. Analyzing the type varibale of the pitches in the training set we see that the different pitches are fastball, cutter, sinker, slider, changeup, curveball, and split finger (FS).

After cleaning the type variable (removing pitch types that were numbers) and plotting picth speed vs spin rate, we get the plot below:

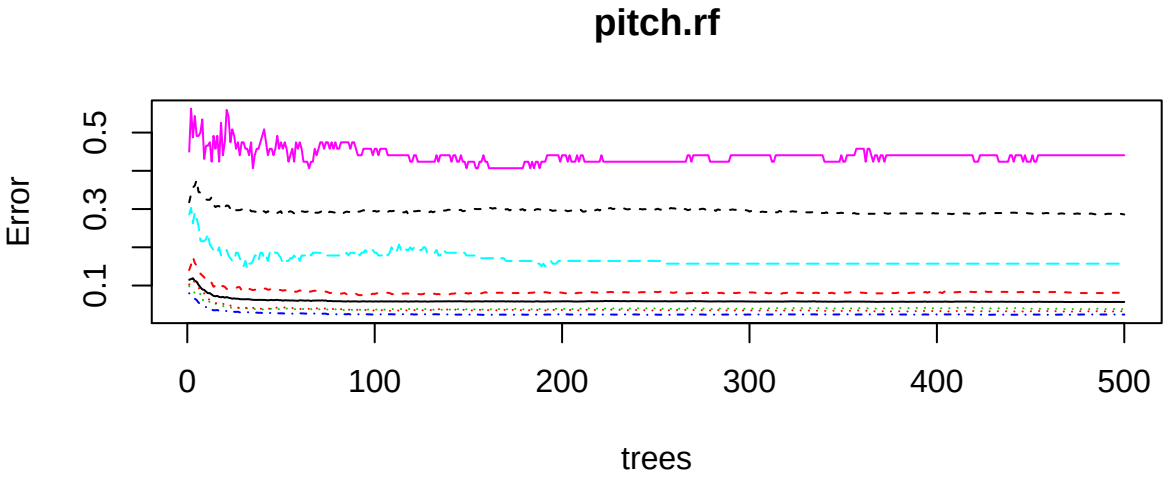


There are a couple very interesting things about this plot. First, there are quite a few pitches with spin rate exactly 3000, suggesting a "ceiling" that perhaps the machine measuring the pitches could not exceed. To improve our model, I am going to remove those pitches. Also, even more obvious is the fact that there are tons of pitches with spin rate under 500. This is almost unheard of, most pitches are thrown with spin rate between 1000-3000 rpm. This leads me to beleive that either the spin_rate is measured in different units than rpm or there has been some kind of scaling for the pitches. However, the distribution of pitch types is consistent for the velocity vs. spin rate data that I have seen, fastballs on the upper-right, sliders and changeups spread throughout the middle, and curveballs on the bottom left. Thus, I do not believe our model predictions will be affected by this scaling change.

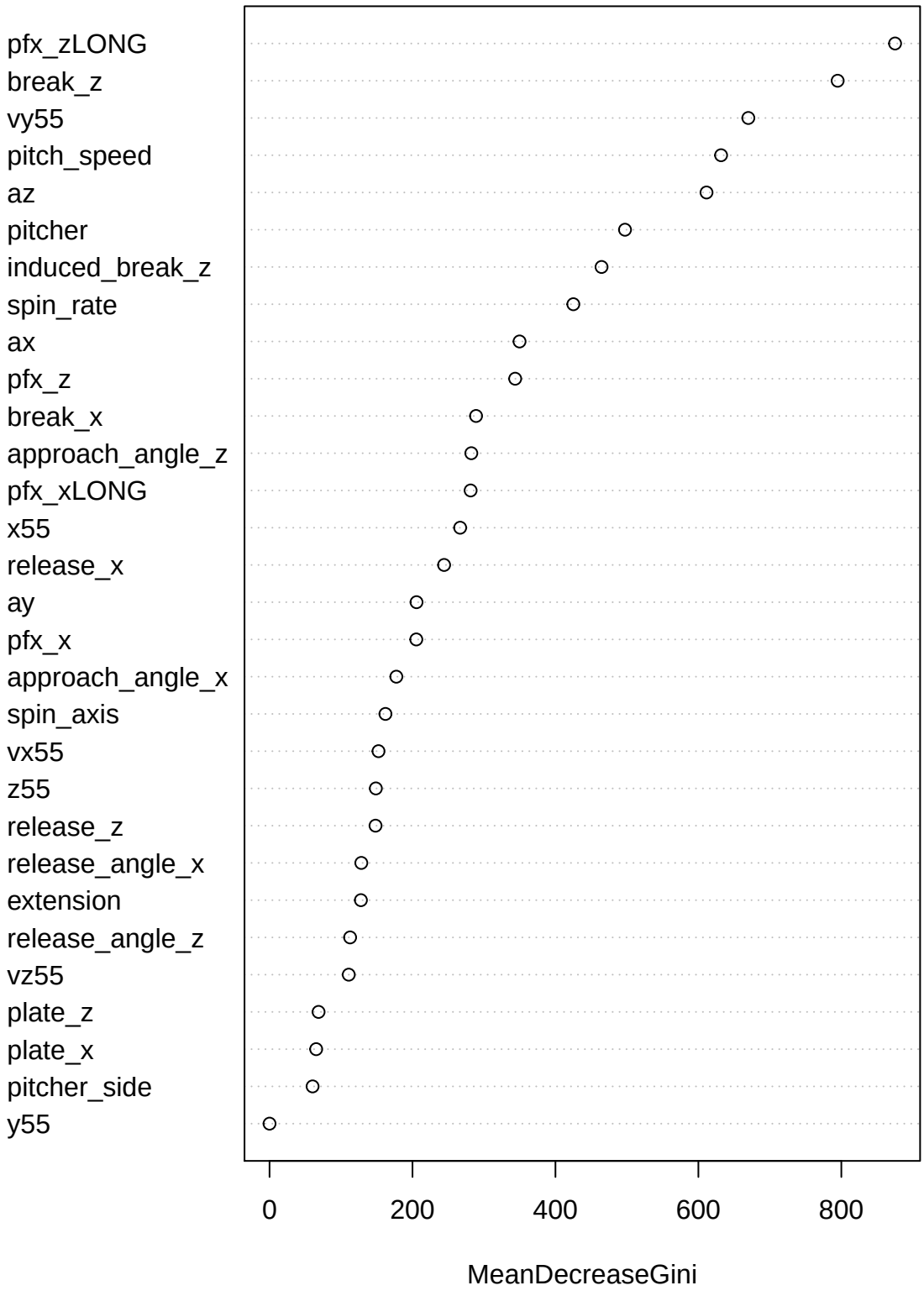
Another interesting plot is one of horizontal break vs. vertical break:



As you can see in the plot above, there are clear patterns for right and left handed pitchers. Clearly, fastballs have the least downward break, followed by changeups, sliders, and curveballs. The offspeed pitches are split into two groups in terms of horizontal break based on the handedness of the pitcher–sliders and curvebvalls have positive horizontal break for right-handed pitchers and negative horizontal break for lefties. Now to begin building our model, we will use a random forest approach. To start, we will consider all variables that seem like they could possibly be relavent in pitch classification:



The plot above clearly shows that the error rate for our primitive model plateaus at around 100 trees, so this is the value we will use to cross validate. Next, we can look at the variable importances that our model gives us:


Variable Importance
pfx_zLONG
break_z
vy55
pitch_speed
az
pitcher
induced_break_z
spin_rate
ax
pfx_z
break_x
approach_angle_z
pfx_xLONG
x55
release_x
ay
pfx_x
approach_angle_x
spin_axis
vx55
z55
release_z
release_angle_x
extension
release_angle_z
vz55
plate_z
plate_x
pitcher_side
y55
0
200
400
600
800
MeanDecreaseGini

We can see from the Variable Importance plot that vertical break is the most important variable in our model. Pitch velocity is also a very important variable. This makes intuative sense because when I think of what really seperates a fastball from a changeup from a slider, I think of how fast the pitch is moving and how the pitch breaks. Also, vertical break should be more important than horizontal break because if you think about it, pitches like cutters and sliders have very similiar horizontal break, as well as pitches like sinkers, changeups, and splitters. On the other end of the spectrum, we see that variables like plate x, plate z, and pitcher side seem to be relatively unimportant. This makes sense because any pitch can end up anywhere on the plate (or off) and pitcher handedness is already taken care of in the variable Pitcher. We will remove these unimportant variables for the next iteration of our model. To get a baseline, we will use our preliminary model to predict the validation data.

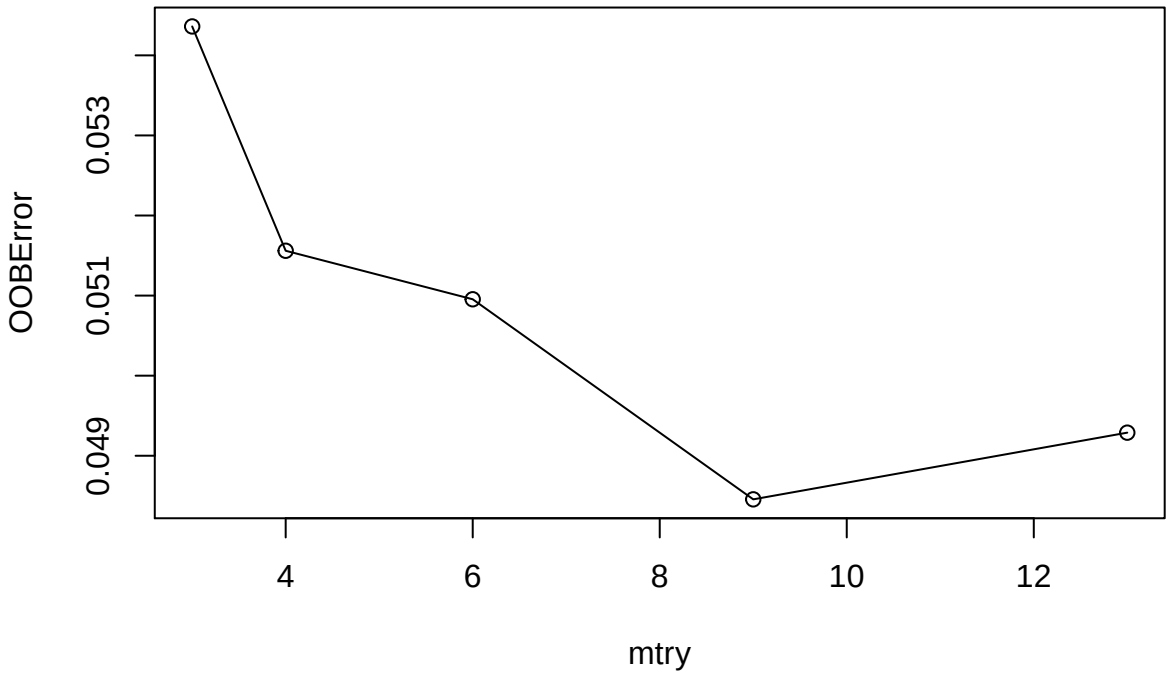
```
pitch.matrix$overall[1]
```

```
##  Accuracy
## 0.9382239
```

The accuracy of the predicted data is 93.8%. Now we will make the improvements on our model.

```
##  Accuracy
## 0.9443561
```

By removing the bottom 11 variables in terms of importance we were able to increase the accuracy of our model by 0.6%. Now we will cross-validate to find the optimal value of mtry (the number of variables sampled at each split).



As we can see from the plot above, the optimal value of the mtry parameter is 9. Thus, our final model will use mtry=9 and the most important variables:

```
pitch.rf.final <- randomForest(factor(type) ~ ., data=training, mtry=9, ntree=100)

predictions <- predict(pitch.rf.final, validate)

pitch.matrix.final <- confusionMatrix(data=predictions,
                reference=validate$type, positive='yes')
pitch.matrix.final$table
```

```
##           Reference
## Prediction   CH   CU   FA   FC   FS   SI   SL
##         CH  400    0   10    0    2    6    3
##         CU    0  477    1    0    0    0    9
##         FA   15    0 2240    1    0   98    6
##         FC    0    0    1   38    0    0    0
##         FS    1    0    0    0   21    0    0
##         SI    2    0   44    0    0  254    0
##         SL    7   16    1    1    0    0  749
```

```
pitch.matrix.final$overall[1]
```

```
##  Accuracy
## 0.9491256
```

The accuracy of our final model is about 95%. From the confusion matrix above, we can see that the most commonly misclassified pitches by the model are the fastball and the sinker. This makes sense because fastballs and sinkers can be very similar pitches depending on who is throwing them.

Now we predict the test data. The predictions for the test data can be found in the csv file pitch_predictions.csv