Predictive Modeling - Concrete Compressive Strength

Rob Leonard (robleonard@tamu.edu (mailto:robleonard@tamu.edu))

Concrete Data

For this assignment, build a predictive model that estimates the compressive strength of concrete using some of the methods learned to date. Identify and discuss the important features in the model.

Load Packages

```
packs = c("dplyr","readxl","car","caret","glmnet","corrplot","earth","vip","randomForest","ranger","doParallel","xgboost","reshape2", "cowplot", "MASS" ,"ggplot2", "GGally")
lapply(packs, require, character.only = TRUE)
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
##
## [[5]]
## [1] TRUE
##
## [[6]]
## [1] TRUE
##
## [[7]]
## [1] TRUE
##
## [[8]]
## [1] TRUE
##
## [[9]]
## [1] TRUE
##
## [[10]]
## [1] TRUE
##
## [[11]]
## [1] TRUE
##
## [[12]]
## [1] TRUE
##
## [[13]]
## [1] TRUE
##
## [[14]]
## [1] TRUE
##
## [[15]]
## [1] TRUE
##
## [[16]]
## [1] TRUE
##
## [[17]]
## [1] TRUE
```

Exploratory Data Analysis

It should be standard practice to look at your dataset intensively before attempting to use it. The goals of EDA are:
- Find any issues (missing data, extreme observations)
- Gain insight into the types of transformations and methods that might be of use
- Discover interesting things about the problem you're ultimately trying to solve

Data Load, Exploration and Pre Processing Steps

General Data Overview

Review data structure

```
## Number of Observations: 1030
##  Number of Features: 8
```

```
##
##
##  Data types:
```

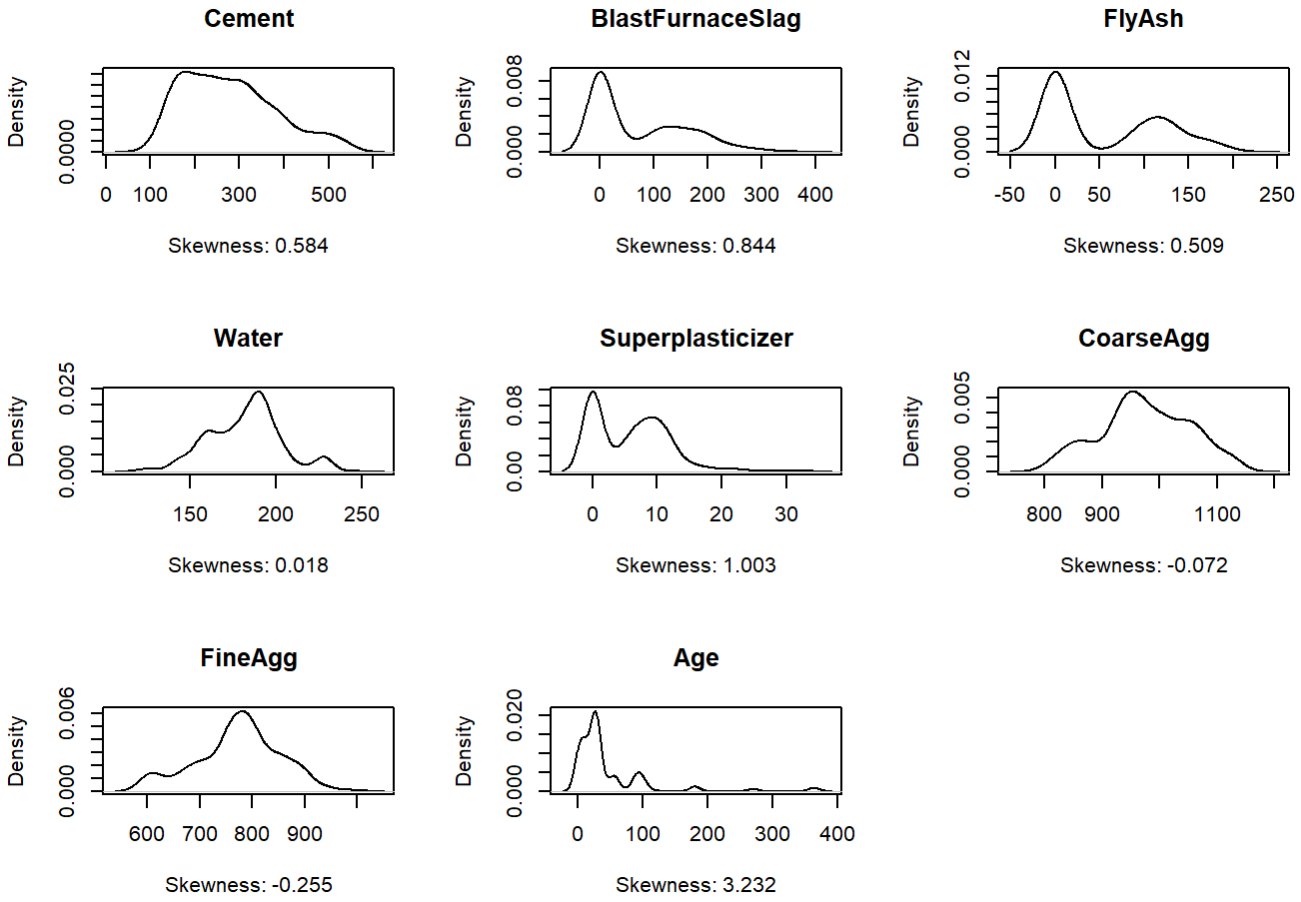
```
##
## numeric
##       9
```

```
##
##  Check for any missing values: FALSE
```

```
## Number of Duplicate Observations: 34
```

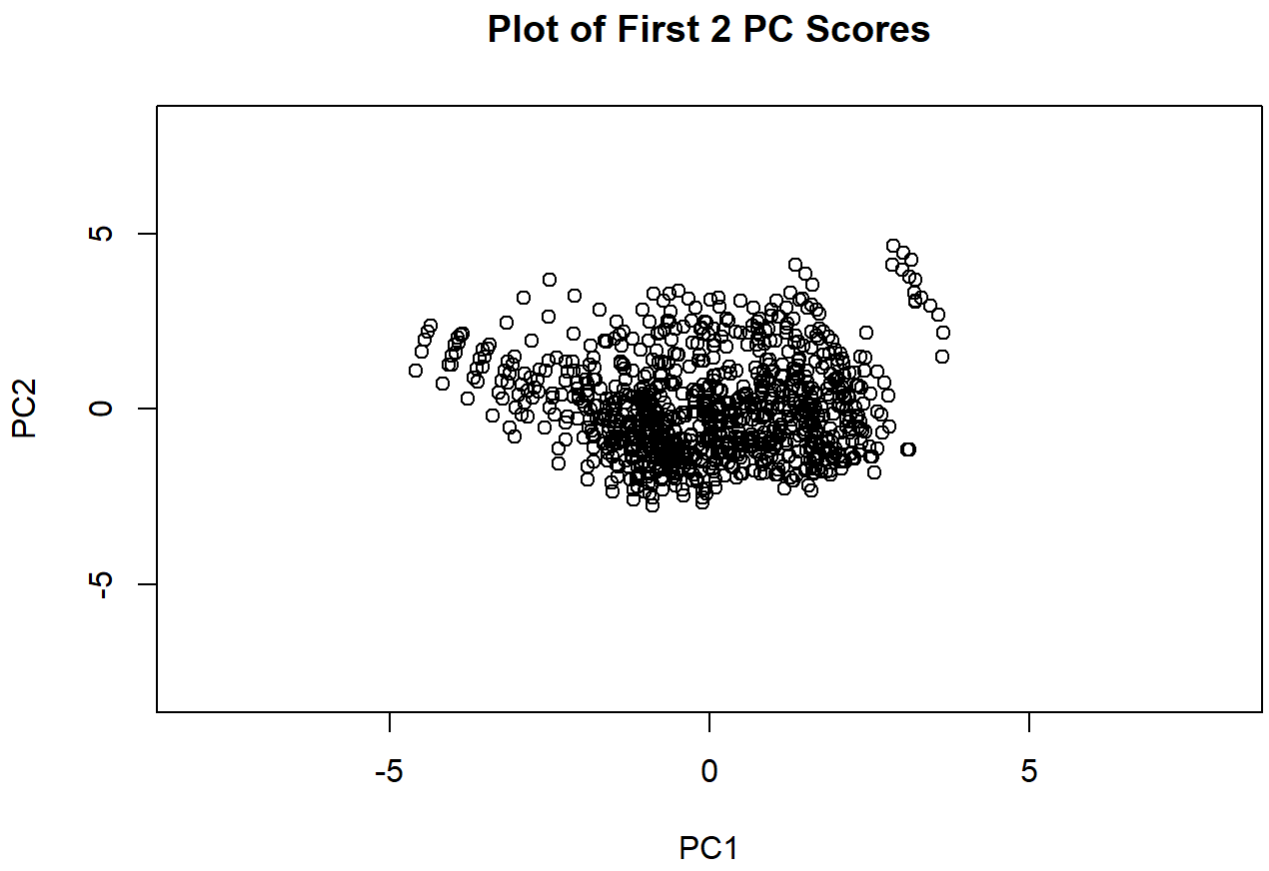
There are 34 duplicate observations out of 1030. From the description it is unclear whether or not these are true duplicates but the descriptions lead towards them being duplicates and removing them. Further investigation would be required to check whether or not these are true duplicates.

Review Skewness and Transformations



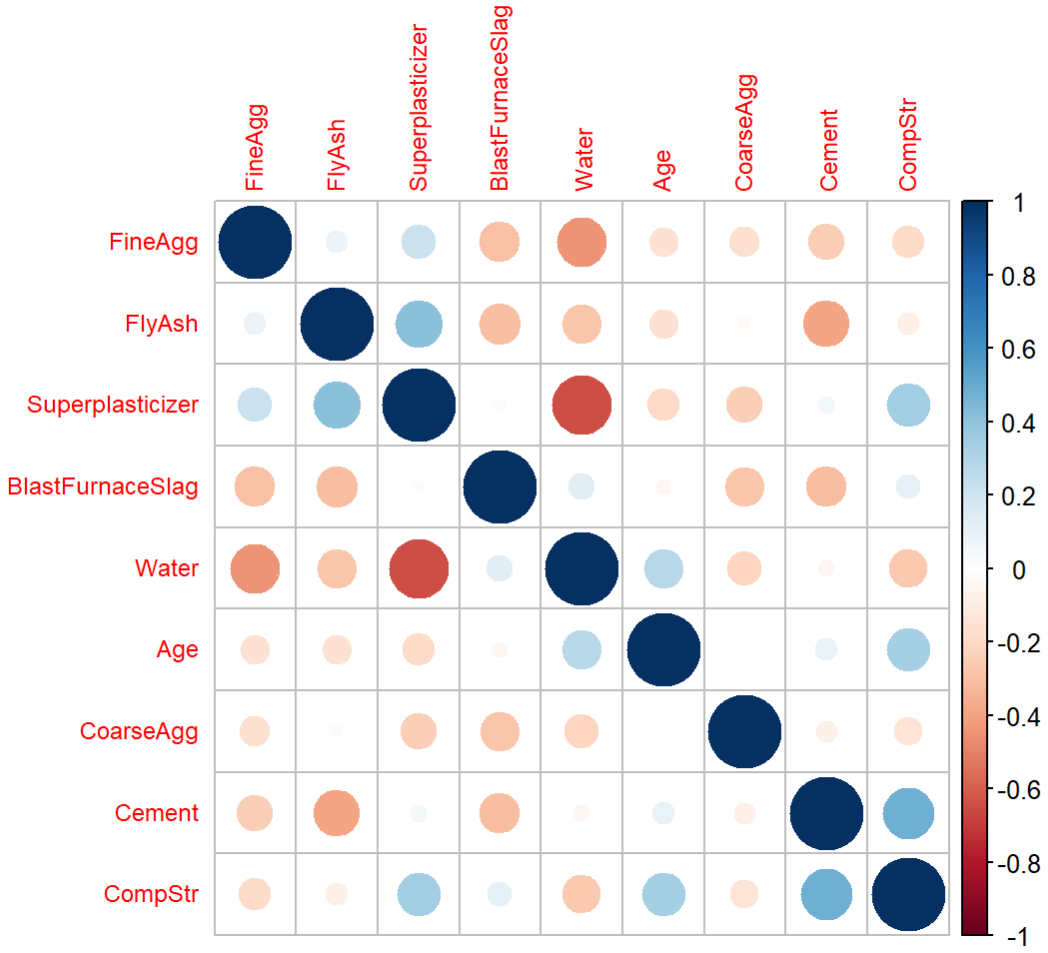
While age is highly right skewed, due to it's lumpiness, a transformation isn't likely to be useful. There are also some extreme observations in age. Tree methods look like they might be useful due to several features being bimodal like blastfurnaceslag, flyash and superplasticizer.

Check for Outliers



There don’t appear to be any extreme points.

Multicollinearity Check



High multicollinearity is not present.

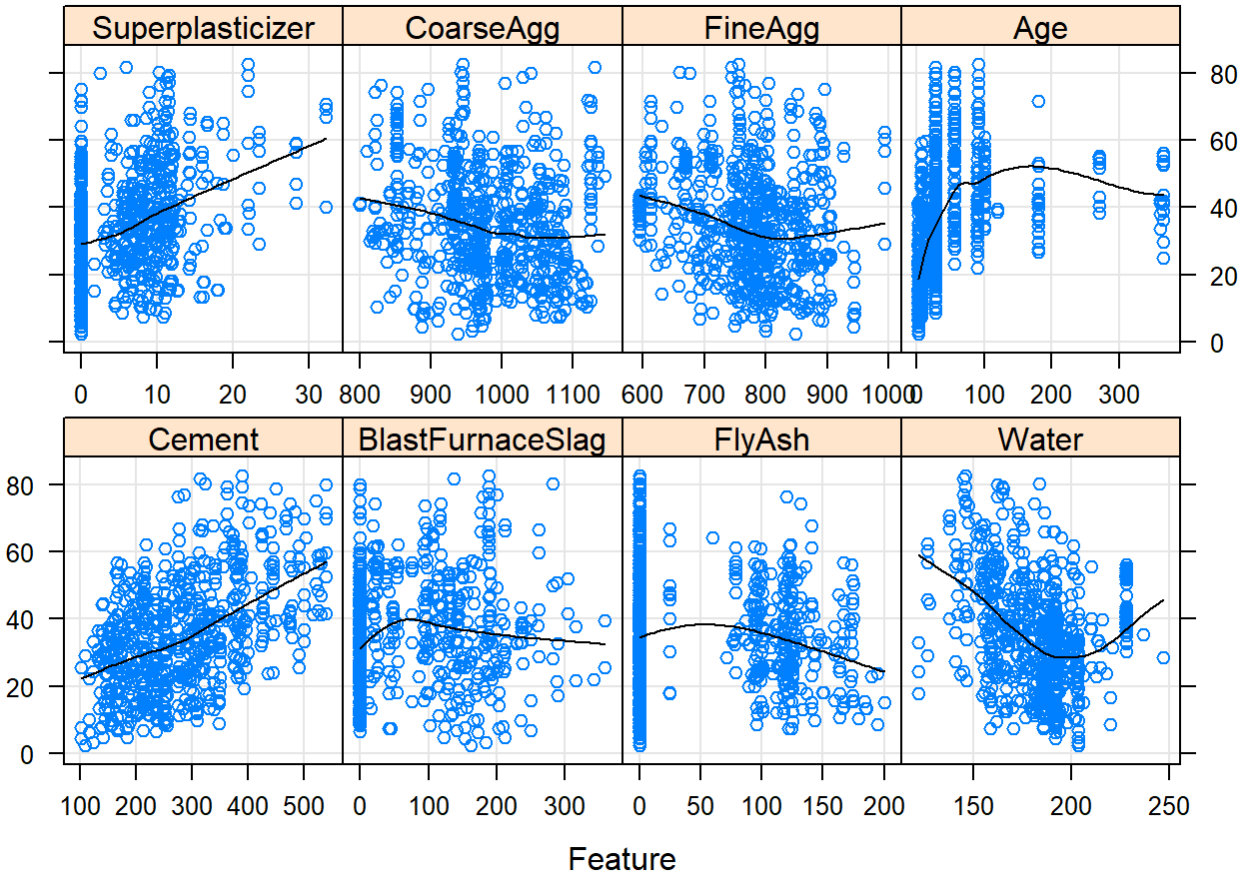
Modeling

Training and Test Data Split

```
set.seed(1999)
yAllData = concreteDataClean$CompStr
xAllData = as.data.frame(concreteDataClean[,-9])
trainIndex = createDataPartition(yAllData, p=.7, list = FALSE) %>% as.vector(.)
yTrain = yAllData[trainIndex]
xTrain = xAllData[trainIndex,]
yTest = yAllData[-trainIndex]
xTest = xAllData[-trainIndex,]
```

A 70%/30% training/testing split is applied to the dataset.

ScatterPlots



Several features look like they might need a quadratic term as the local fit isn't a straight line.

Feature Engineering

```
xTrain = xTrain %>% mutate(Cementsq = Cement^2, BlastFurnaceSlagSq = BlastFurnaceSlag^2, FlyAshS
q = FlyAsh^2, WaterSq = Water^2, SuperplasticizerSq = Superplasticizer^2, CoarseAggSq = CoarseAg
g^2, FineAggSq = FineAgg^2, AgeSq = Age^2)
xTest = xTest %>% mutate(Cementsq = Cement^2, BlastFurnaceSlagSq = BlastFurnaceSlag^2, FlyAshSq
 = FlyAsh^2, WaterSq = Water^2, SuperplasticizerSq = Superplasticizer^2, CoarseAggSq = CoarseAgg
^2, FineAggSq = FineAgg^2, AgeSq = Age^2)
# create matrix forms for some models
xTrainMat = as.matrix(xTrain)
xTestMat = as.matrix(xTest)
```

Given the limited number of features and the size of the dataset, it's feasible to add quadratic terms as identified above.

Setup train control

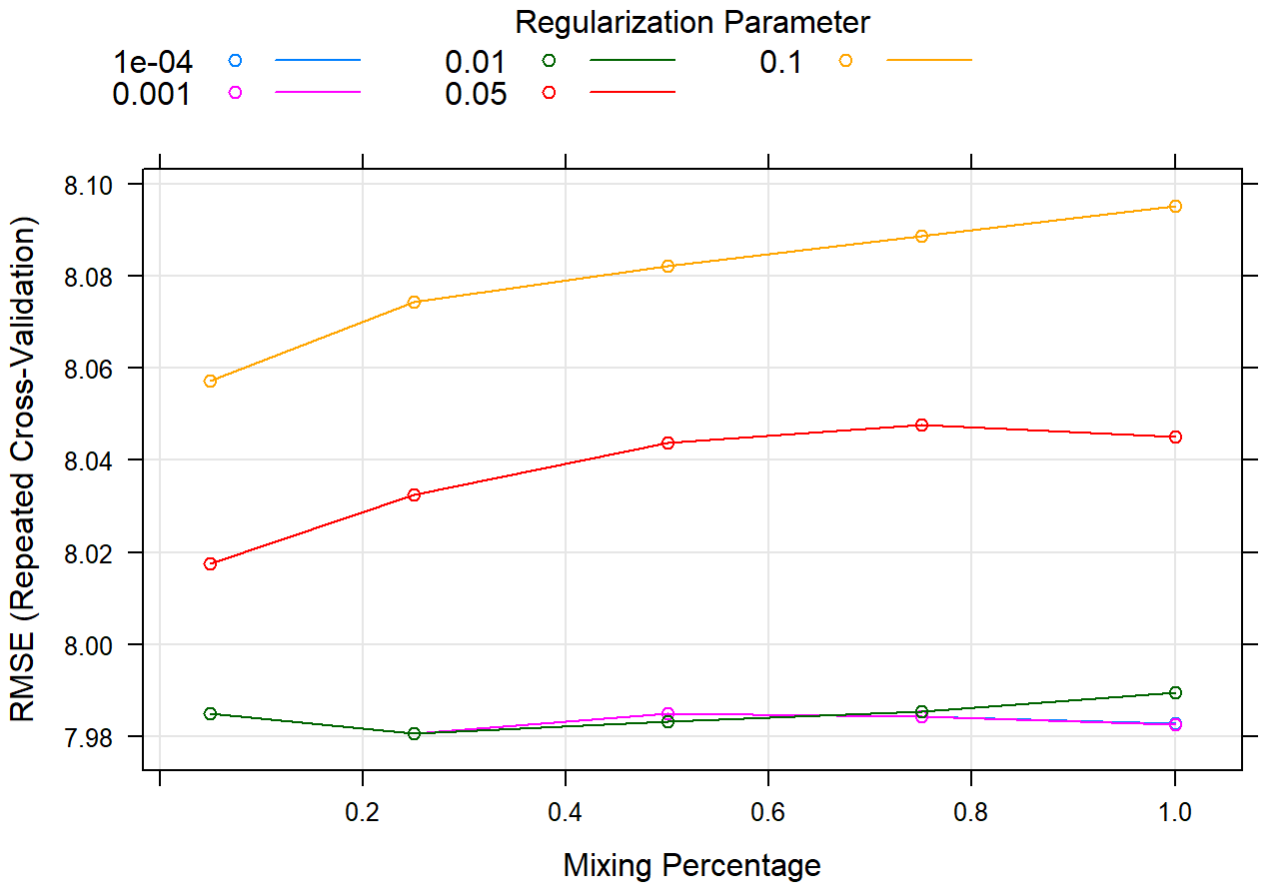
```
trControl = trainControl(method = "repeatedcv", repeats = 2, number = 10)
```

Set up parallel processing

```
cl = makeCluster(20)
registerDoParallel(cl)
```

Linear Modeling

Elastic Net

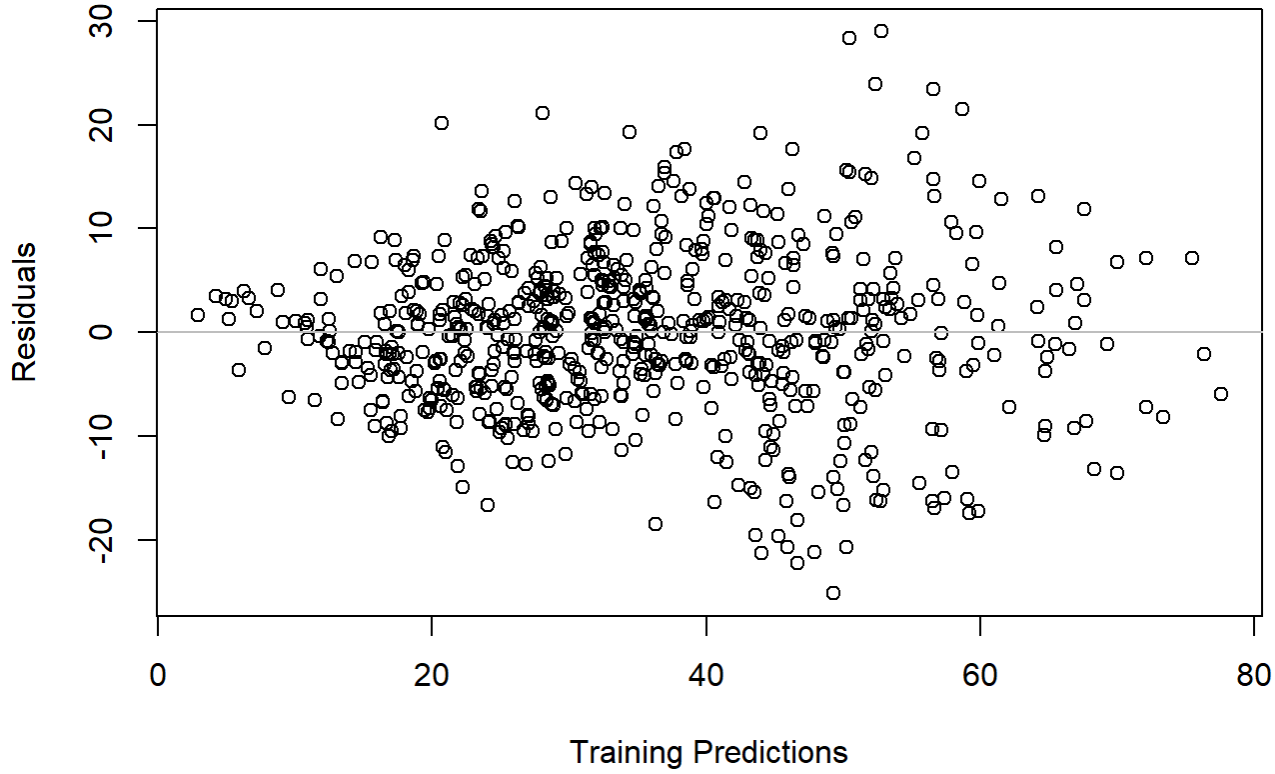


RMSE bottoms out just below 8 in terms of mean compressive strength.

Model Validation

```
## Elastic Net - Best Model Tuning Parameters:
```

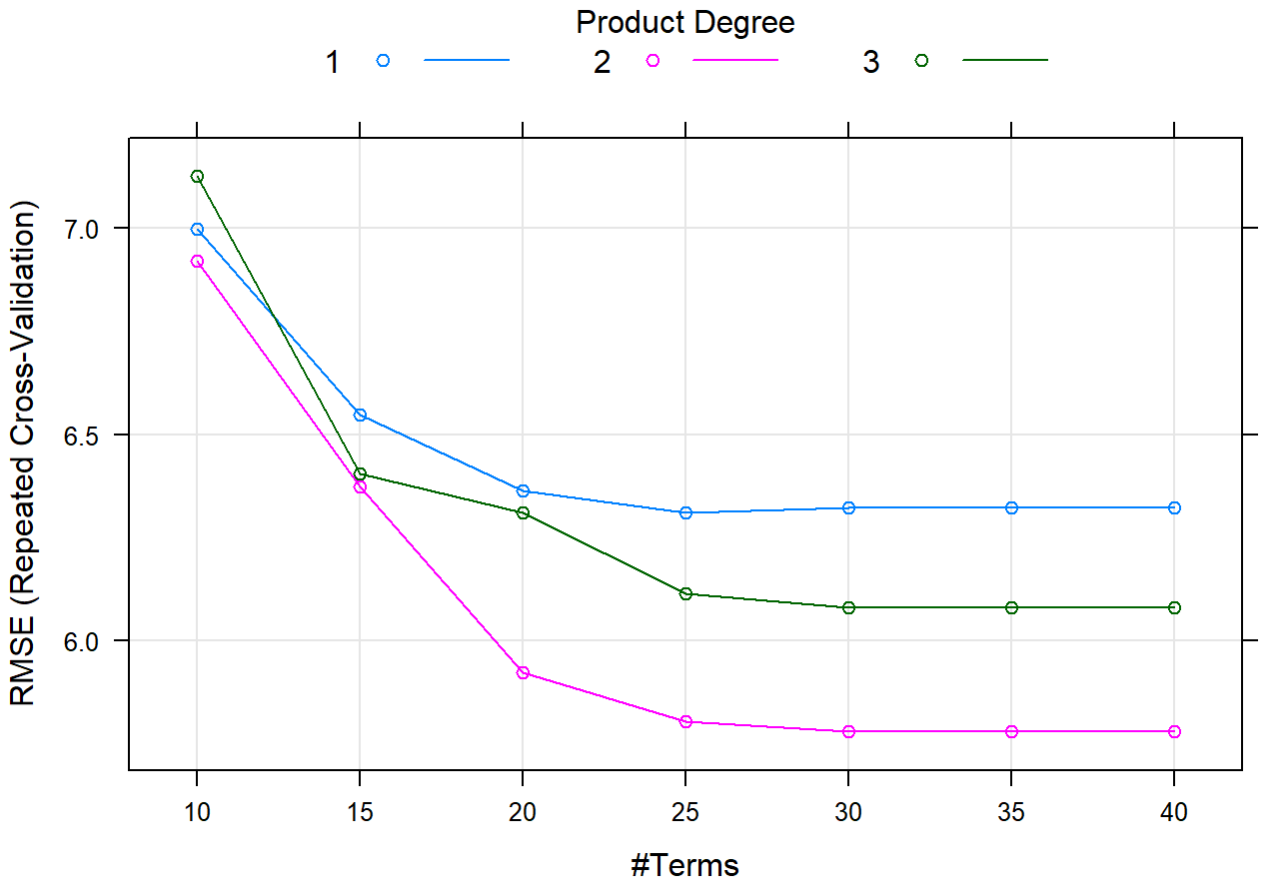
```
##   alpha lambda
## 7  0.25  0.001
```



The residuals look appropriate. They are a random scatter around the horizontal axis. The chosen EN model is skipped here as nonlinear models are likely to provide better predictions as previously discussed.

Nonlinear Models

MARS

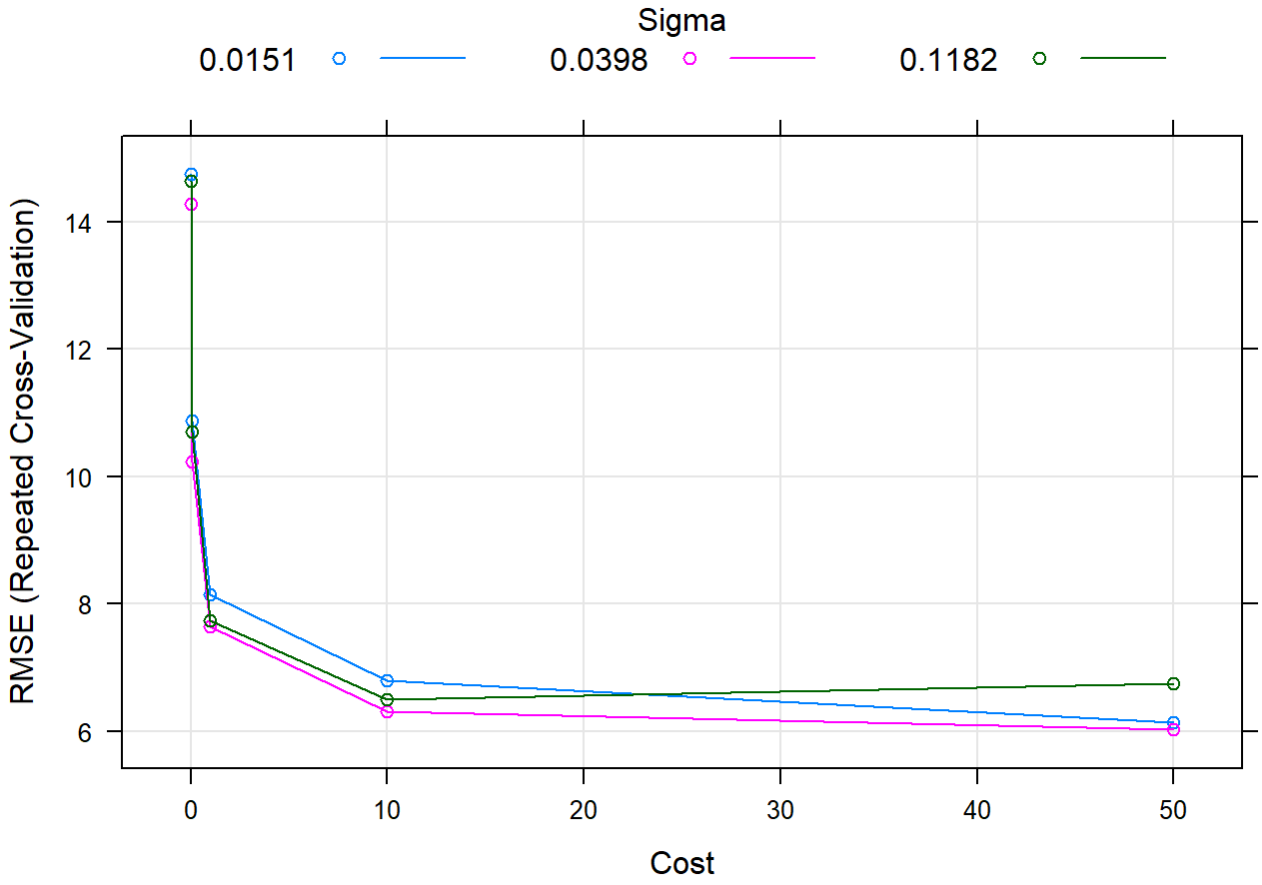



```
## MARS - Best Model Tuning Parameters:
```

```
##    nprune degree
## 12     30      2
```

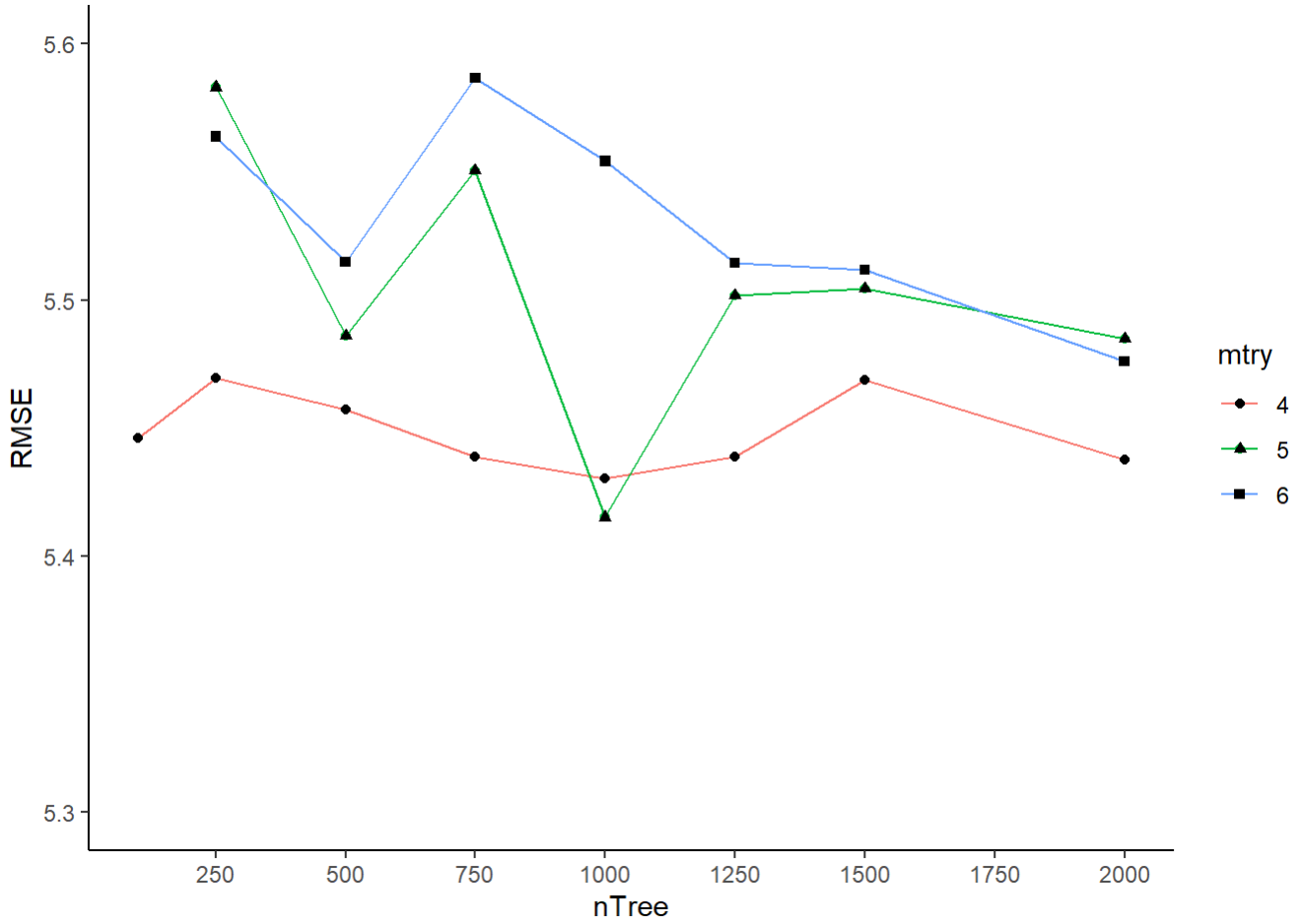
The best MARS model includes quadratic terms. RMSE has been reduced to under 6.

SVM



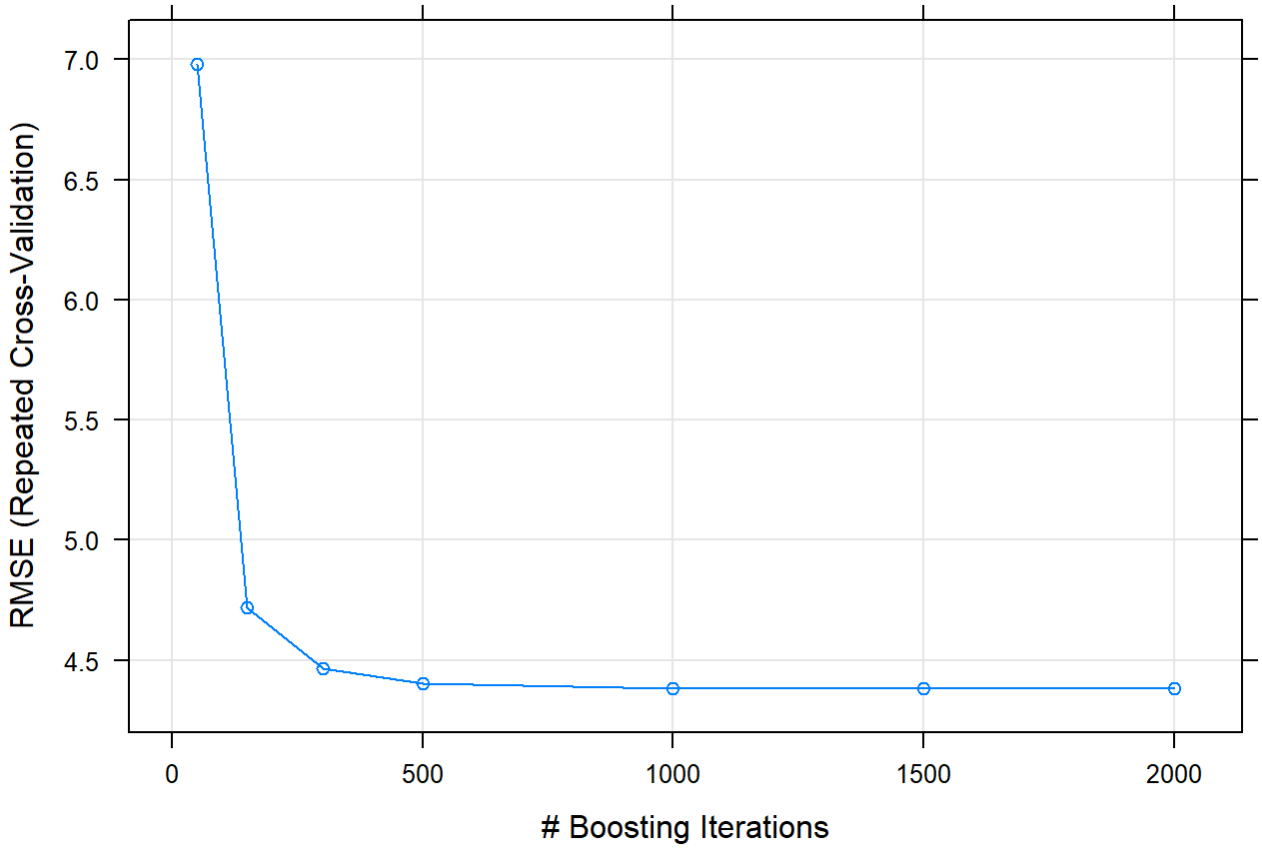
A support vector machine does not perform better than MARS, although there are numerous tuning paramters that might be adjusted to increase performance.

Random Forest



Random Forest improves the estimated predictive performance. RMSE is well below 6.

Boosting

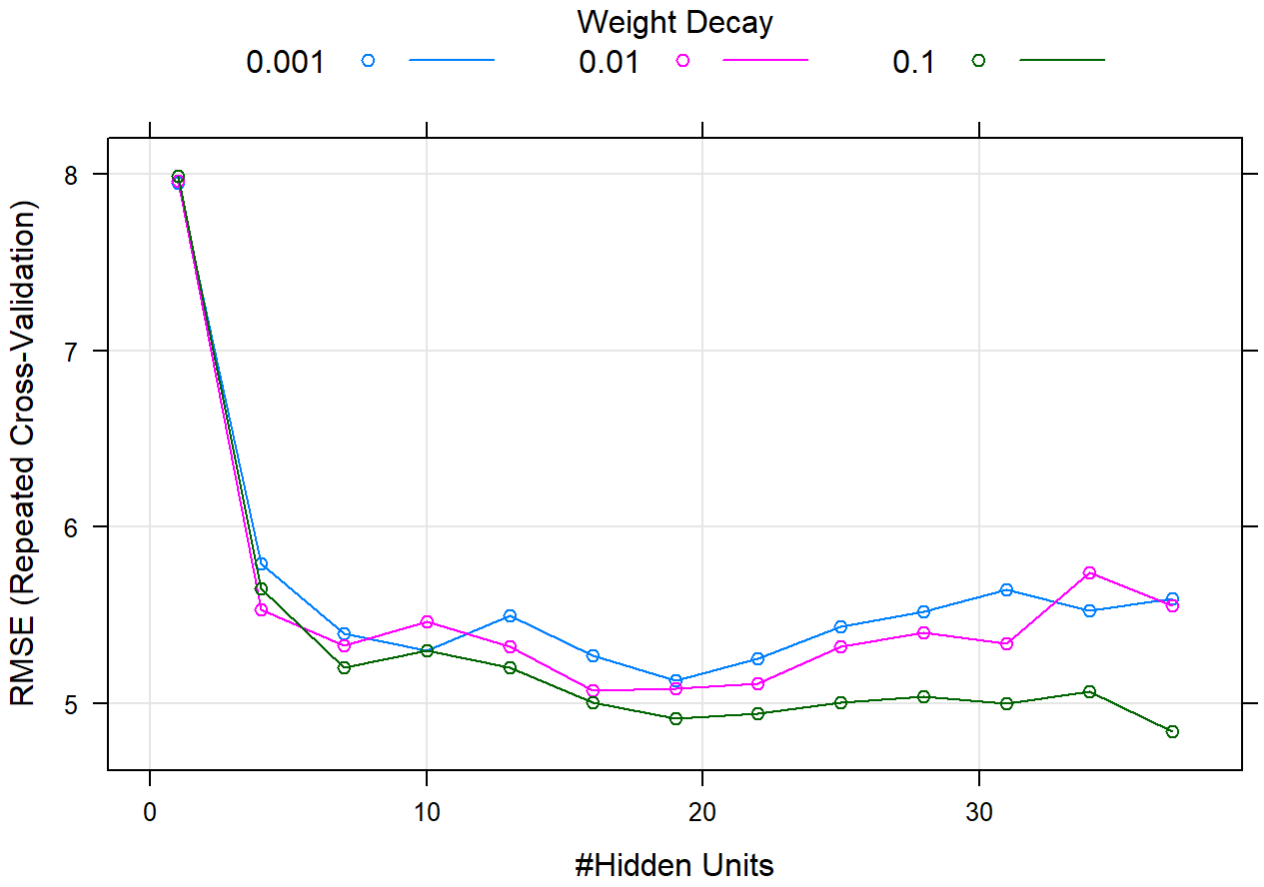



```
## Boosting - Best Model Tuning Paramters:
```

```
##   nrounds max_depth  eta gamma colsample_bytree min_child_weight subsample
## 6    1500         6 0.05     0                1                0       0.5
```

Boosted trees significantly improves performance. This is the best model so far.

Neural Network

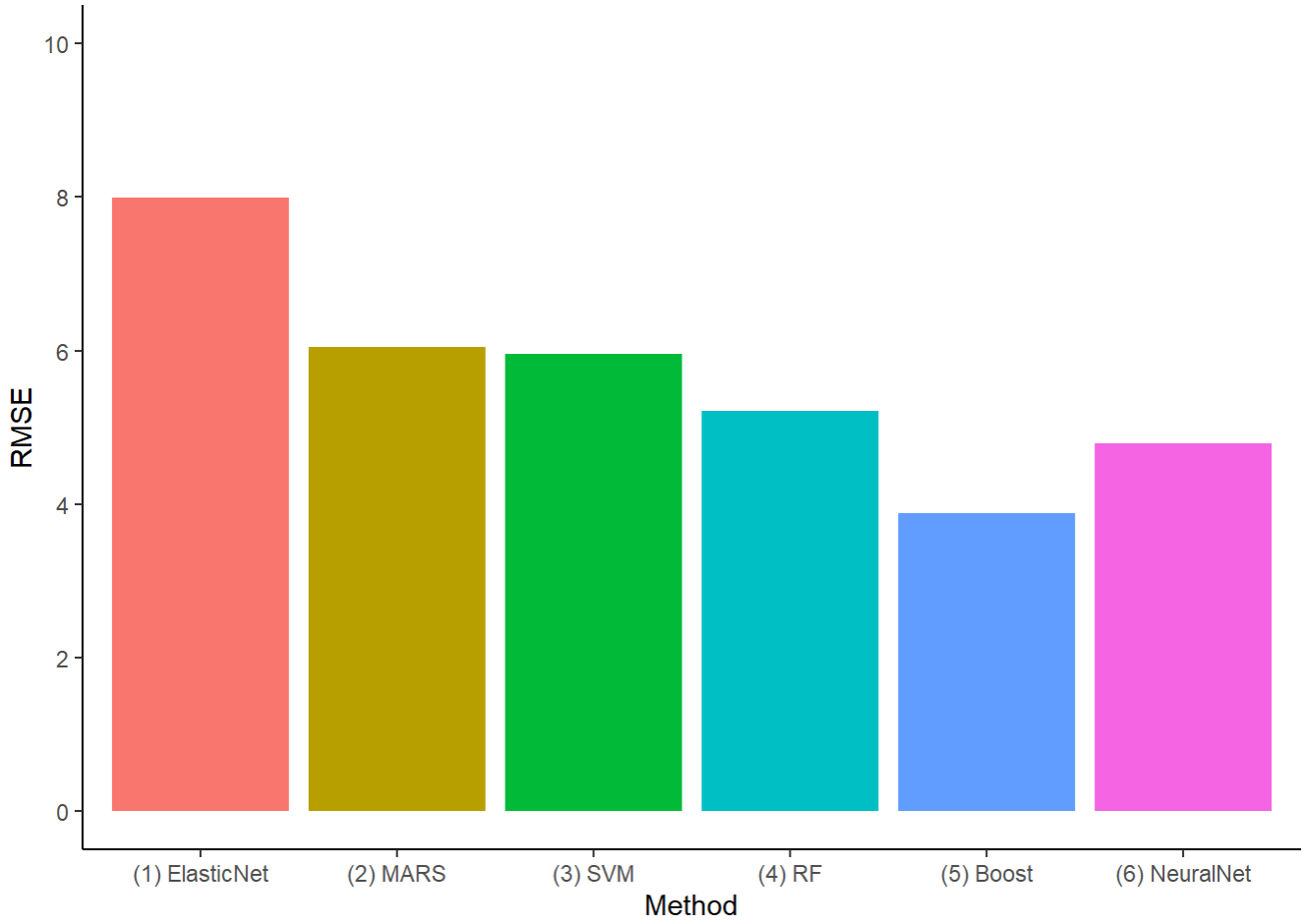



```
## Neural Network - Best Model Tuning Paramters:
```

```
##    size decay   bag
## 39   37   0.1 FALSE
```

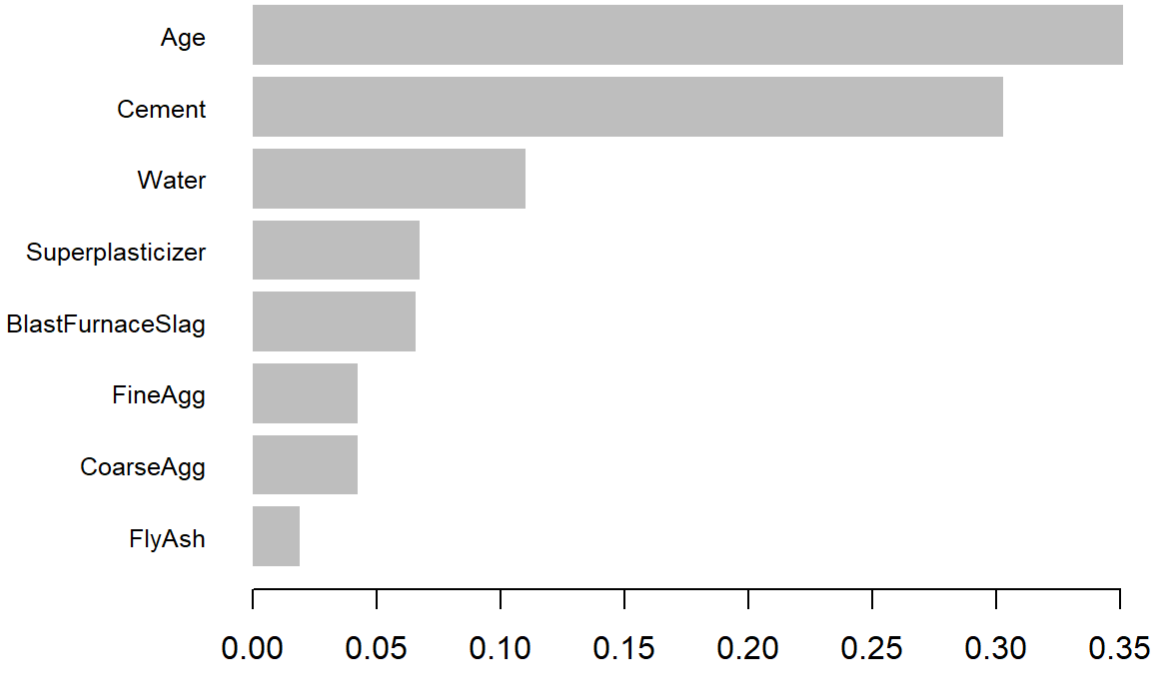
A neural network also show promising results. However, computational time was significantly longer than boosted trees.

Judging Performance



The nonlinear methods significantly outperformed the linear model. Boosted trees provided the best performance.

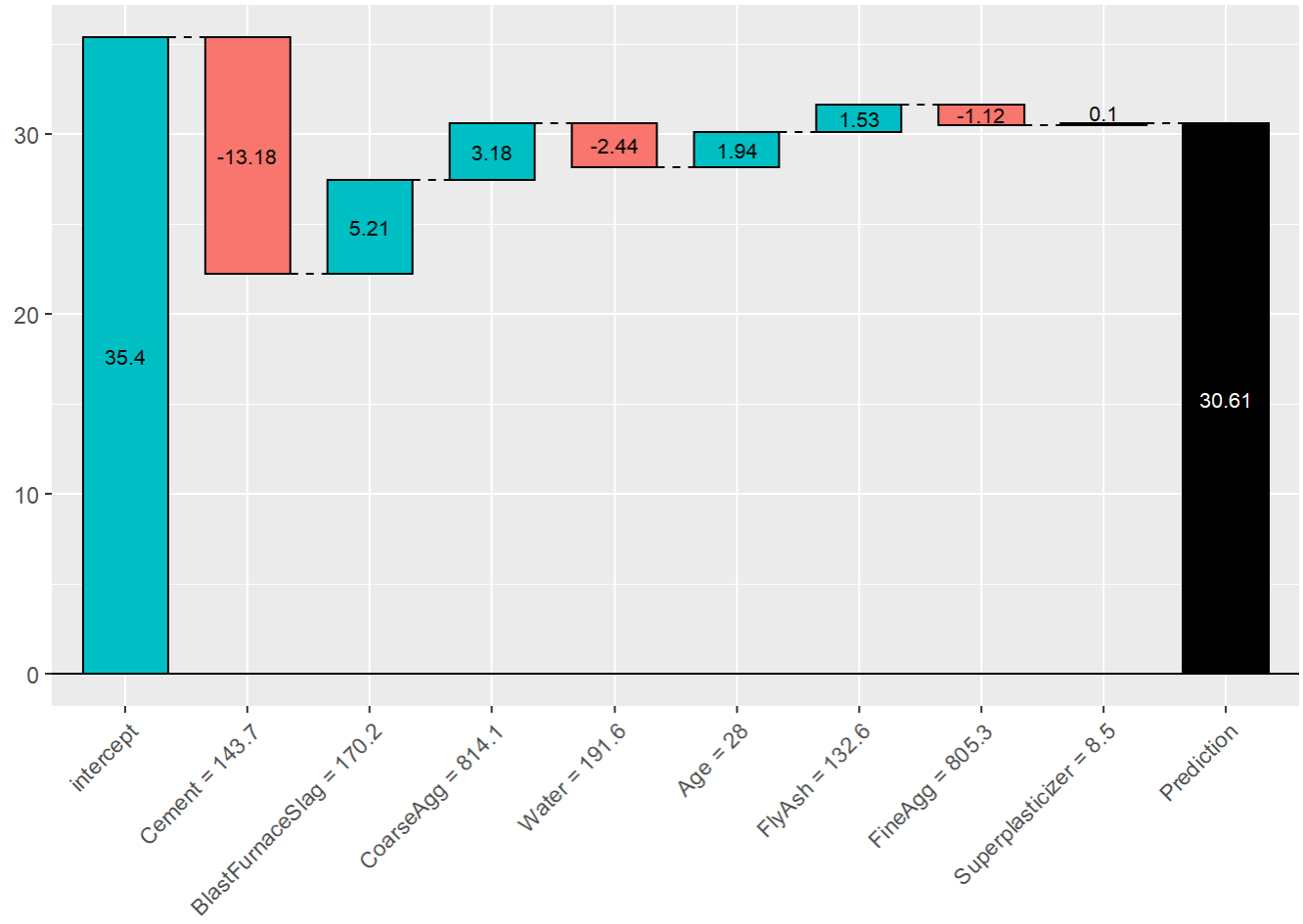
Variable Importance



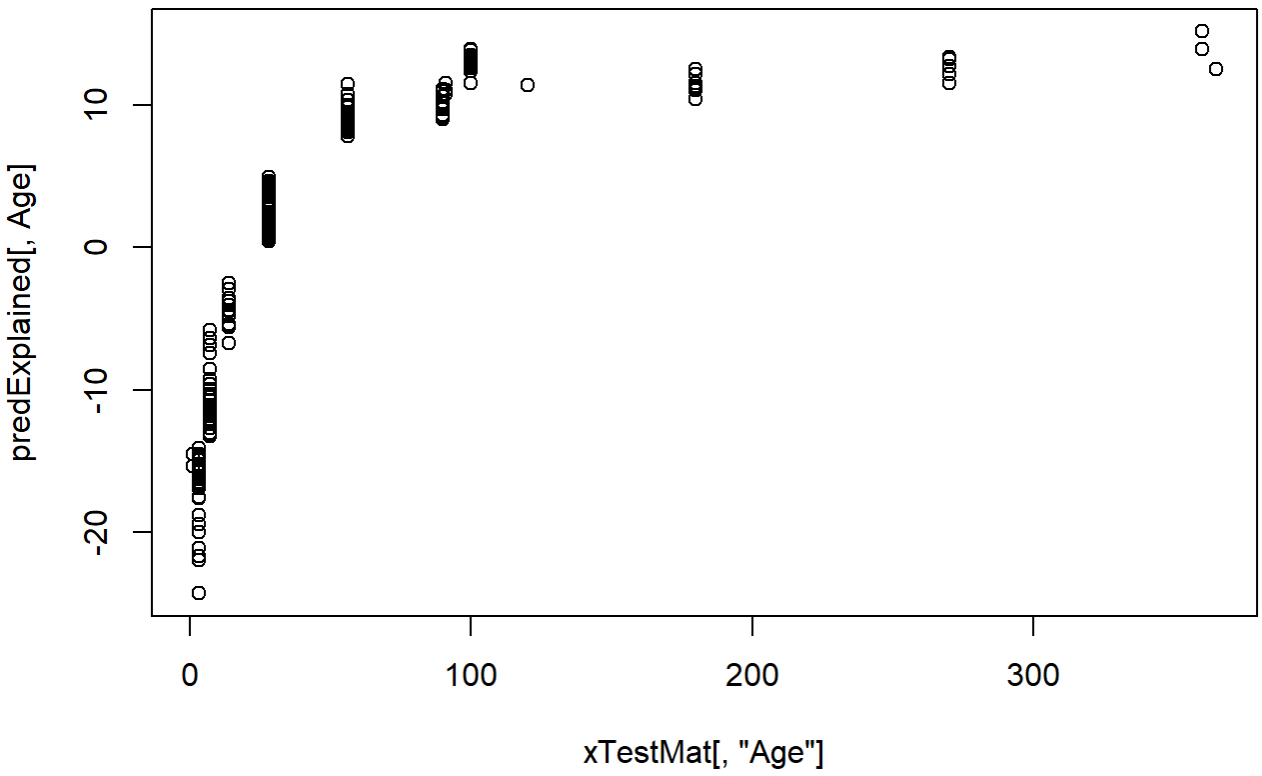
Age and cement are the two most important features in predicting compressive strength. Interpreting each feature's contribution to a prediction is a little more difficult with boosted trees than with linear models. However, the feature waterfall chart for observation 20 is shown, and provides an idea for how much each feature contributes to the estimated compressive strength. This can be applied to any observation of interest. Individual feature plots can also be extracted, and the final plot shows the effect of age on compressive strength, with the effects leveling off after 100 days. Concrete mixture samples younger than 100 days have a decrease in estimated compressive strength. One further item to note is that these observations really are mixtures. Further investigation should consider mixture modeling but is beyond the scope of this course.

```
library(xgboostExplainer)
xTrainDM = xgb.DMatrix(data = xTrainMat, label = yTrain)
xTestDM = xgb.DMatrix(data = xTestMat)
boostExplainer = buildExplainer(boostOut$finalModel, xTrainDM, type = "regression")
```

```
  |
  |=================================================================       |  91%
  |
  |=================================================================       |  91%
  |
  |=================================================================       |  92%
  |
  |=================================================================      |  92%
  |
  |=================================================================      |  93%
  |
  |=================================================================      |  94%
  |
  |=================================================================     |  94%
  |
  |=================================================================     |  95%
  |
  |=================================================================    |  95%
  |
  |=================================================================    |  96%
  |
  |=================================================================   |  96%
  |
  |=================================================================   |  97%
  |
  |=================================================================   |  98%
  |
  |=================================================================  |  98%
  |
  |=================================================================  |  99%
  |
  |=================================================================| 99%
  |
  |=================================================================| 100%
##
## DONE!
##
## Prediction:  30.60514
## Weight:  30.60514
## Breakdown
##        intercept           Cement BlastFurnaceSlag        CoarseAgg
##       35.3970563      -13.1818601        5.2120465        3.1822757
##            Water              Age           FlyAsh          FineAgg
##       -2.4448818        1.9371738        1.5298462       -1.1242284
## Superplasticizer
##        0.0977123
```




```
plot(xTestMat[,"Age"], predExplained[,Age])
```


predExplained[, Age]
10
0
-10
-20
0
100
200
300
xTestMat[, "Age"]