ChooseYourOwn_Report

Will Powers

12/28/2020

The Dataset

The Data set used for this paper is a data set created for the following paper:

> *Özdemir, Ahmet Turan, and Billur Barshan. "Detecting Falls with Wearable Sensors Using Machine Learning Techniques." Sensors (Basel, Switzerland) 14.6 (2014): 10691–10708. PMC. Web. 23 Apr. 2017.*

The data was downloaded from kaggle.com (https://www.kaggle.com/pitasr/falldata), without reading the accompanying paper so as to not bias the predictive modeling creating for this paper. However, to understand the history and context of the data a brief synopsis was read and will be summarized here. The data was taken from the medical information and activities of elderly patients in China. There are roughly 16,000 data points of patients in different states of being. The purpose of the original paper was to create a classification algorithm for a potential wearable device to detect whether or not a patient was in the process of falling down. The reason for this is that falling can cause serious injuries in elderly patients and so medical responders can get to patients as quickly as possible to aid injured patients.

Variables

The variable *Activities* are classified as: Standing (0), Walking (1), Sitting (2), Falling (3), Cramps (4) and Running (5). Medical information that is being recorded is: monitoring time (TIME), sugar level (SL), EEG monitoring rate (EEG), blood pressure (BP), heart rate (HR) and blood circulation (Circulation). The preview of the data set is shown below:

```
## # A tibble: 6 x 7
##   ACTIVITY  TIME     SL    EEG    BP    HR CIRCLUATION
##   <fct>    <dbl>  <dbl>  <dbl> <dbl> <dbl>       <dbl>
## 1 TRUE.    4723.  4020. -1600     13    79         317
## 2 FALSE.   4059.  2191. -1146.    20    54         165
## 3 FALSE.   4774.  2788. -1263.    46    67         224
## 4 FALSE.   8271.  9546. -2849.    26   138         554
## 5 FALSE.   7102. 14149. -2381.    85   120         809
## 6 FALSE.   7015.  7337. -1700.    22    95         427
```

Project Goals

The goals of this paper will be the same as the original paper for which this data was collected. Although no bias from the original paper was used or influenced the following modeling in any way.

Key Steps

1. Prepare Data for Analysis
2. Use Binary Classification Algorithms and compare to find the optimal solution
3. Validate the optimal solution against a test set for official results

Methods & Analysis

Process

The data will be explored and visualized to see if there are any ways in which the data can further be prepared for better testing. The data will be prepared as such and further prepared for binary classification according to the primary task of detecting a falling patient. Then the data set will be partitioned for validation for final testing purposes and cross-validation sets to compare different models. We will then use the following algorithms on different cross-validation sets to train models for prediction: 1. KNN 2. QDA 3. Classification (Decision) Tree 4. Random Tree Next, we will select the optimal training algorithm and test it on our validation set for a non-biased assessment of the algorithm's performance.

Techniques

Data Cleaning

Thankfully, much of the data has already been cleaned by the aforementioned team doing the original study. To further clean the data we will look for any NAN values that may not be optimal for the simple models we are looking at in this paper. We have found no NAN values:

```
## 0
```

Data Preparation

To prepare for data modeling we will separate our data into a validation set and training set. We will be splitting the data equally into 2 parts, as the following research paper recommends:

> Korjus, Kristjan & Hebart, Martin & Vicente, Raul. (2016). An Efficient Data Partitioning to Improve Classification Performance While Keeping Parameters Interpretable. PLOS ONE. 11. e0161788. 10.1371/journal.pone.0161788.

It states "In general, for the empirical data sets used in this article and for maximizing statistical sensitivity, the optimal test set size was around 50%."

Further, in order to test out different models, we will separate our training set roughly equally into 4 cross-validation sets.

Also, we must prepare our data for bi-variate classification. A sophisticated analysis could attempt a more complex multivariate classification strategy, which would be useful for a hypothetical wearable device. However, since the main goal of this study is to detect a falling elderly person in order to avoid personal injury, we will perform a simpler classification of "falling" vs "not falling". A more nuanced model will require optimization of other categories of activities that could detract from the optimization of detecting for a fall. To do this we will simplify the categories of the ACTIVITY variable to TRUE or FALSE, where TRUE represents the condition that a person is falling and FALSE represents another condition.

Data Exploration on Original Non-Partitioned Data Set

Correlation of Rating Index with 'ACTIVITY' Value

```
## [1] -0.007598562
```

Variables with Near Zero Variance

```
## integer(0)
```

Mean of ACTIVITY

```
## Warning in mean.default(fallData$ACTIVITY): argument is not numeric or logical:
## returning NA
```

```
## [1] NA
```

Mean of TIME

```
## [1] 10936.84
```

Mean of SL

```
## [1] 75271.98
```

Mean of EEG

```
## [1] -5621.125
```

Mean of BP

```
## [1] 58.25107
```

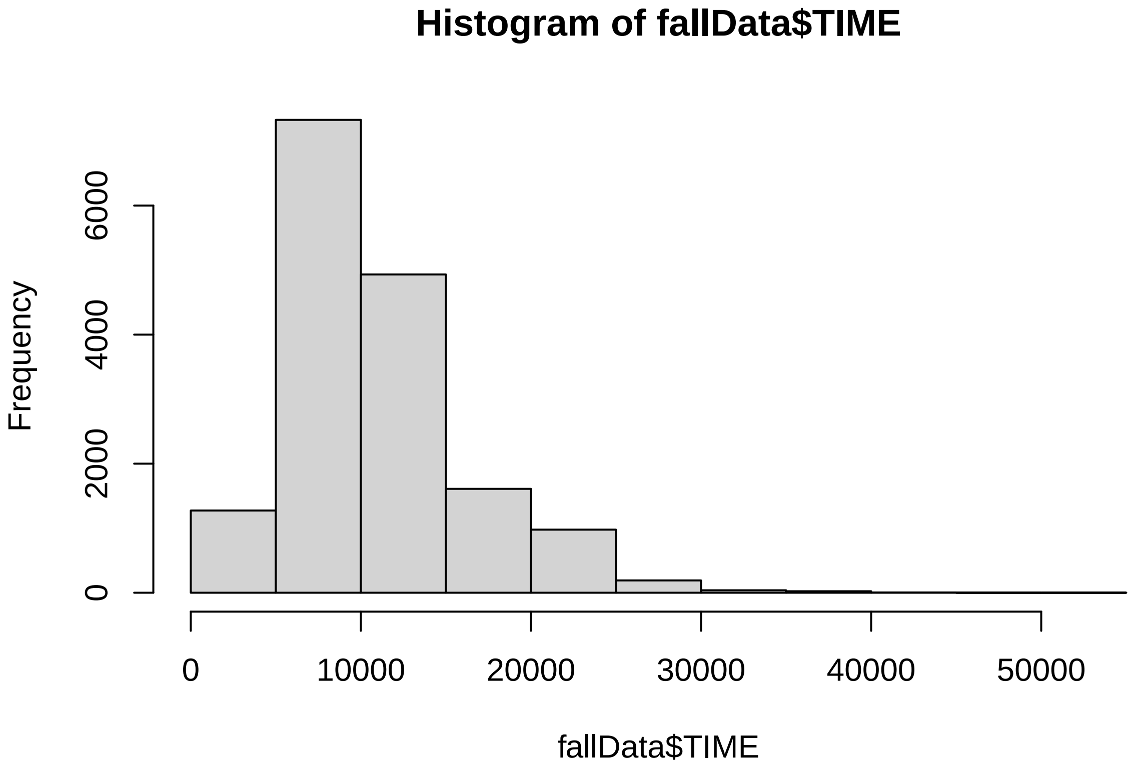
Mean of HR

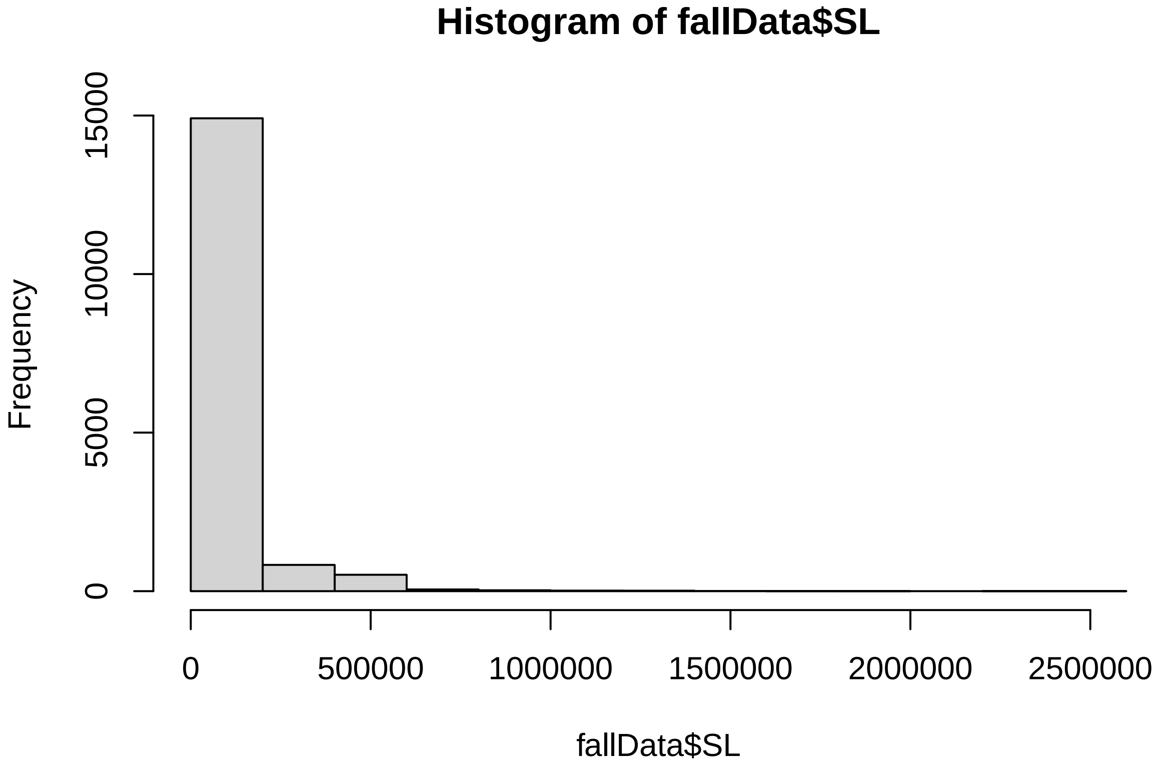
```
## [1] 211.537
```

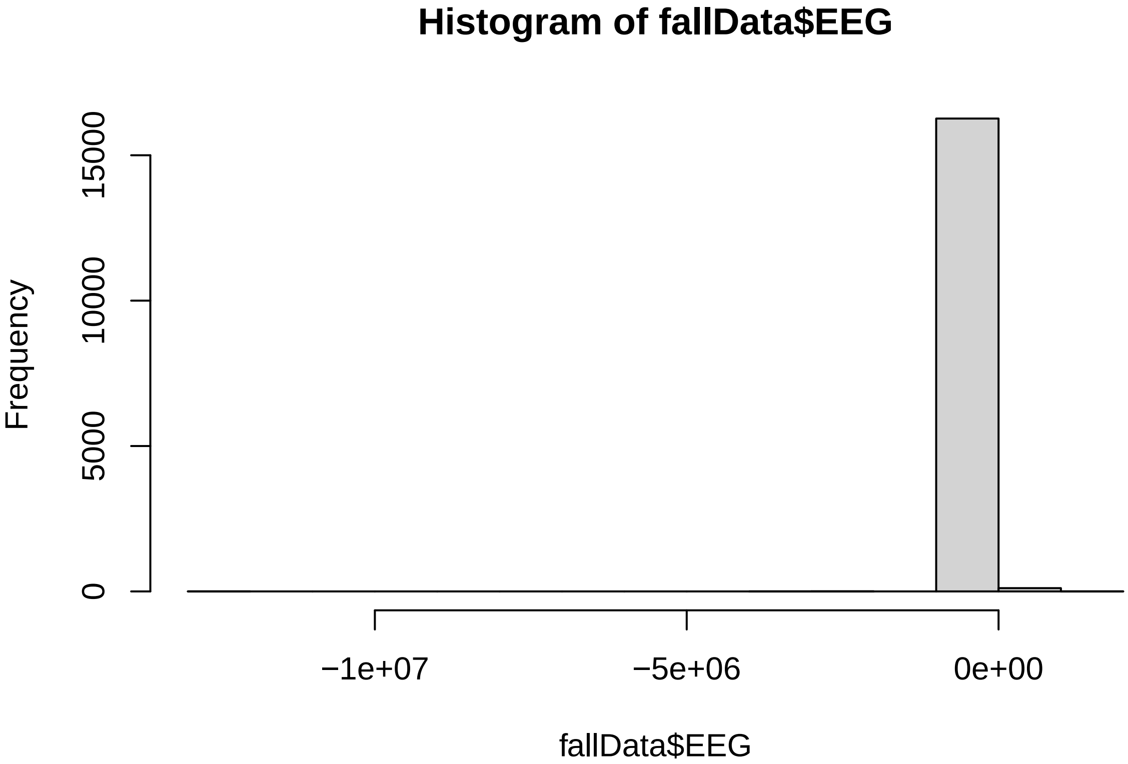
Mean of CIRCLUATION

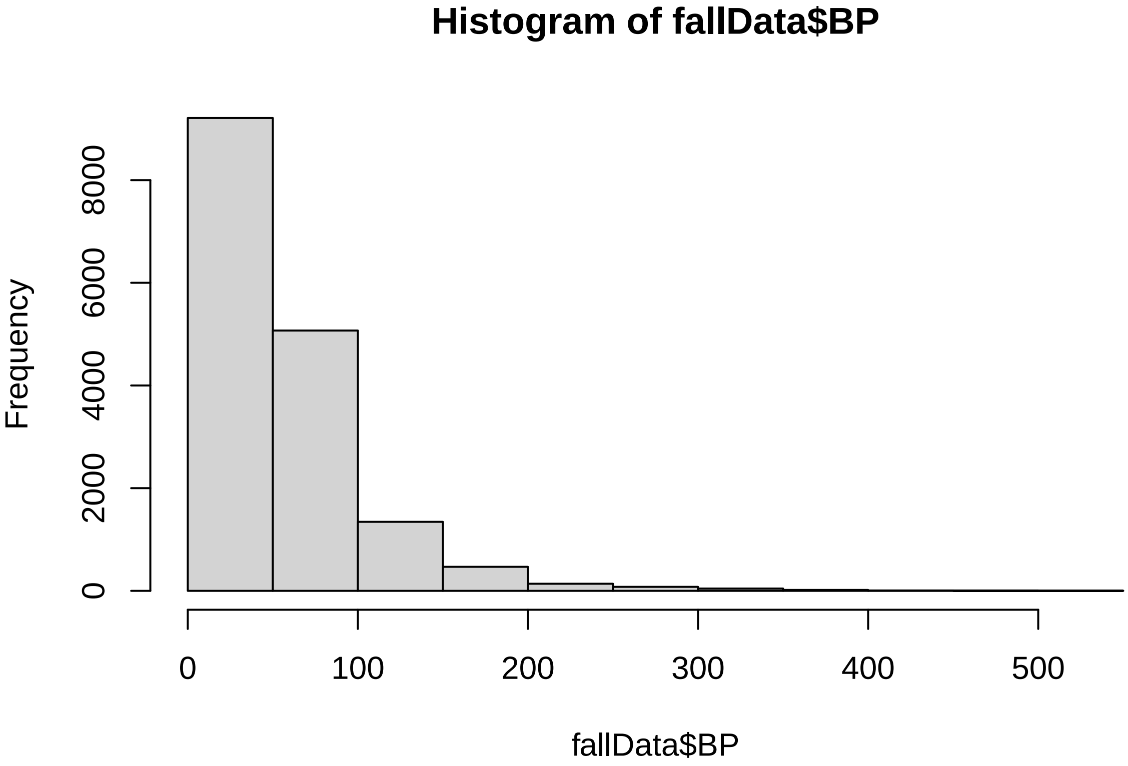
```
## [1] 2894.341
```

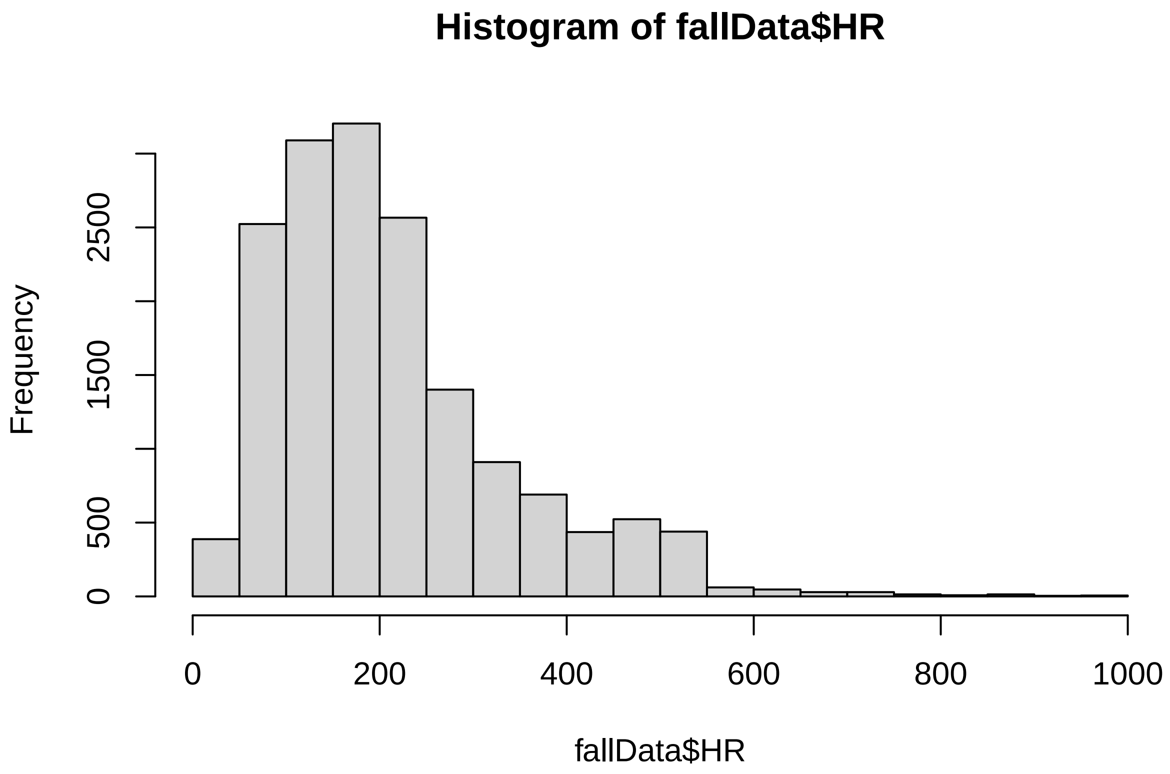
Data Visualization

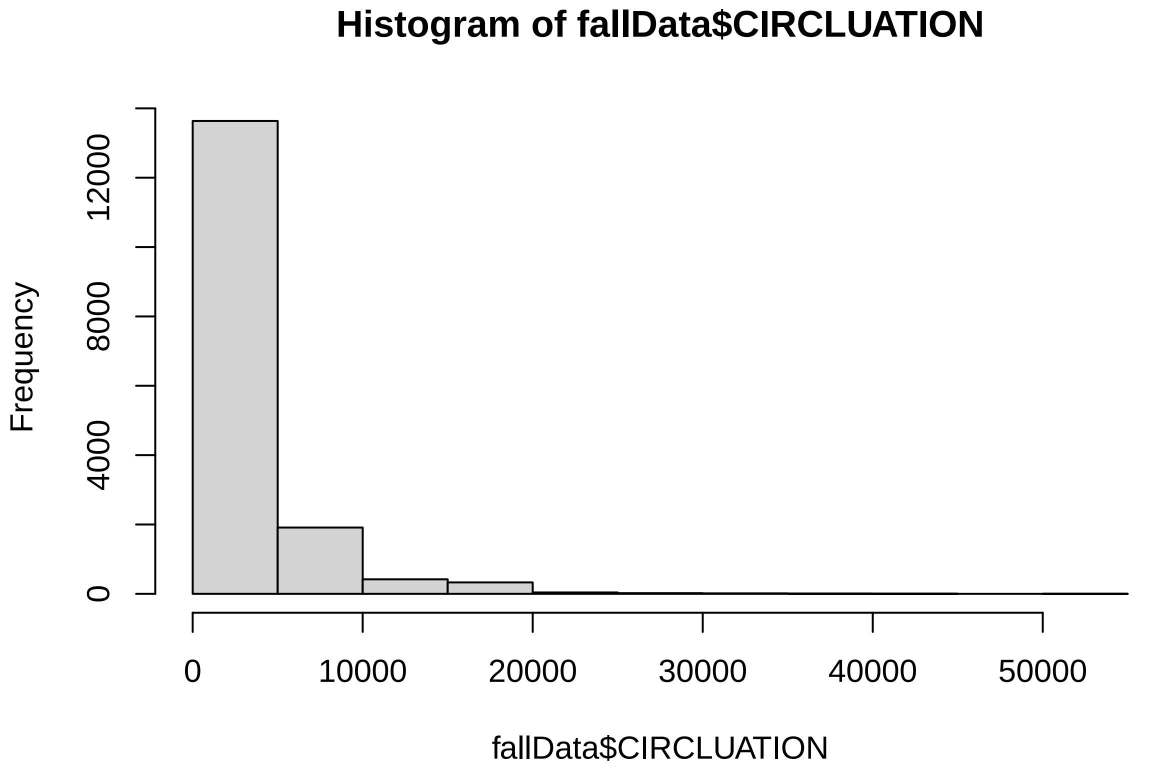





Histogram of fallData$EEG
Frequency
0
5000
10000
15000
−1e+07
−5e+06
0e+00
fallData$EEG


Histogram of fallData$BP
Frequency
0
2000
4000
6000
8000
0
100
200
300
400
500
fallData$BP





Insights

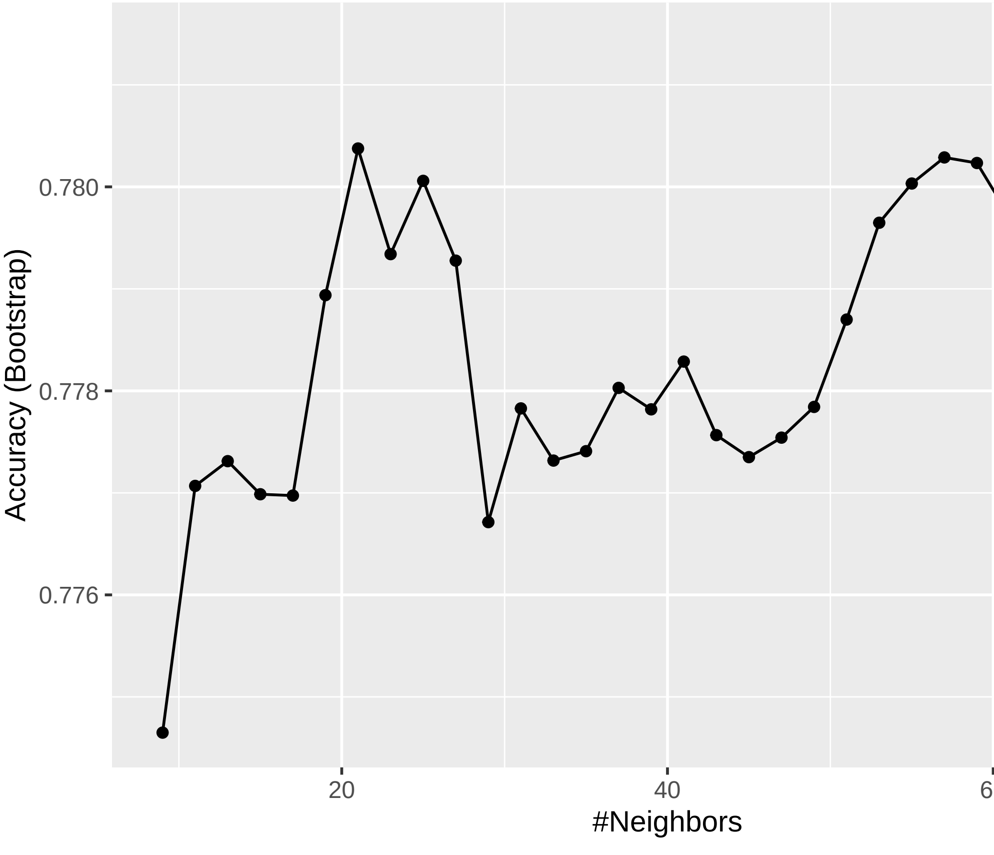
First we checked the correlation of the row indexes with the values of the ACTIVITY variable to make sure that rows are sufficiently randomized. Since we get a pearson's correlation coefficient of between -.01 and

.01, this meets the more rigorous scientific definitions of guaranteed randomness. We also looked to see if there were any variables with near zero variance that could be removed to save memory and processing time, but none were found. We also notice that variables are not normally distributed. This study will not use mean normalization and feature scaling to correct these distributions, however in further analyses it should be used.

Modeling Approach

For our model, we will attempt to predict our new bi-variate variable *ACTIVITY* using all of our other variables. We will use the modeling techniques listed above. We will use a separate cross-validation set for each type of modeling. First however we will do an initial assessment of our modeling techniques by looking at the performance numbers of KNN optimizing for the "caret" package's default performance metric of Accuracy using bootstrapping. Also with our KNN model we will repeat the modeling using values of K from 9 to 71, increment by 2. We will then use the optimal value of K in the final model that we assess.

KNN optimized for Accuracy (Bootstrap)



plot of Accuracy's by value of K

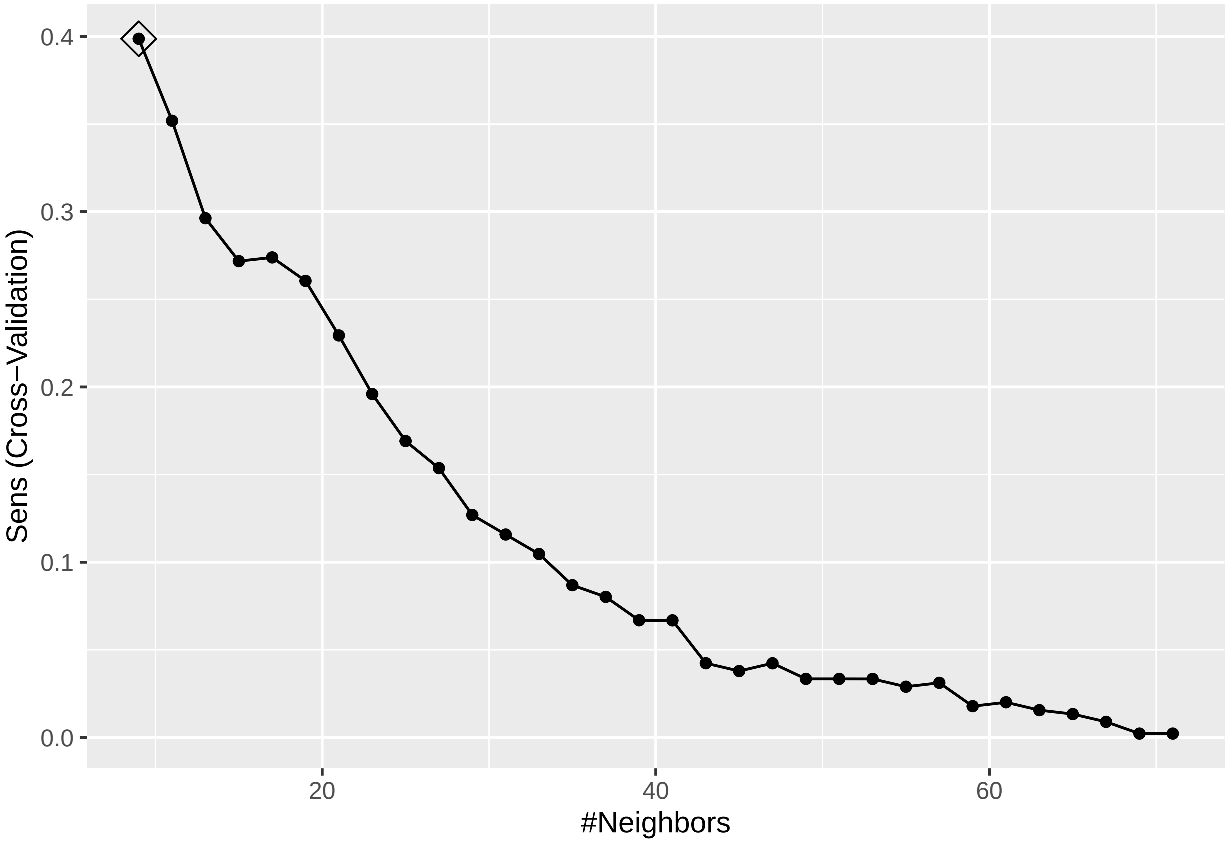
Confusion Matrix and Performance Data

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction TRUE. FALSE.
##     TRUE.     14     12
##     FALSE.   435   1588
##
##                Accuracy : 0.7818
##                  95% CI : (0.7633, 0.7996)
##     No Information Rate : 0.7809
##     P-Value [Acc > NIR] : 0.4701
##
##                   Kappa : 0.0358
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.031180
##             Specificity : 0.992500
##          Pos Pred Value : 0.538462
##          Neg Pred Value : 0.784973
##              Prevalence : 0.219131
##          Detection Rate : 0.006833
##    Detection Prevalence : 0.012689
##       Balanced Accuracy : 0.511840
##
##        'Positive' Class : TRUE.
##
```

Initial Analysis We found that the optimal k value for this model was k=69

Looking at the different metrics of performance we can see that we are sacrificing perhaps too much sensitivity for accuracy. If we inspect the goals of our study, which are to respond to injuries of elderly patients as quickly as possible, then medical professionals may tolerate a high amount of false positives if it maximizes the amount patients that are adequately detected as having fallen and treated in a fast manner. Therefore we will perform all of our models optimizing for sensitivity.

KNN - Optimized for Sensitivity



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction TRUE. FALSE.
##     TRUE.    226    119
##     FALSE.   223   1481
##
##                Accuracy : 0.8331
##                  95% CI : (0.8162, 0.849)
##     No Information Rate : 0.7809
##     P-Value [Acc > NIR] : 2.234e-09
##
##                   Kappa : 0.468
##
##  Mcnemar's Test P-Value : 2.553e-08
##
##             Sensitivity : 0.5033
##             Specificity : 0.9256
##          Pos Pred Value : 0.6551
##          Neg Pred Value : 0.8691
##              Prevalence : 0.2191
##          Detection Rate : 0.1103
##    Detection Prevalence : 0.1684
##       Balanced Accuracy : 0.7145
```

```
##
##        'Positive' Class : TRUE.
##
```

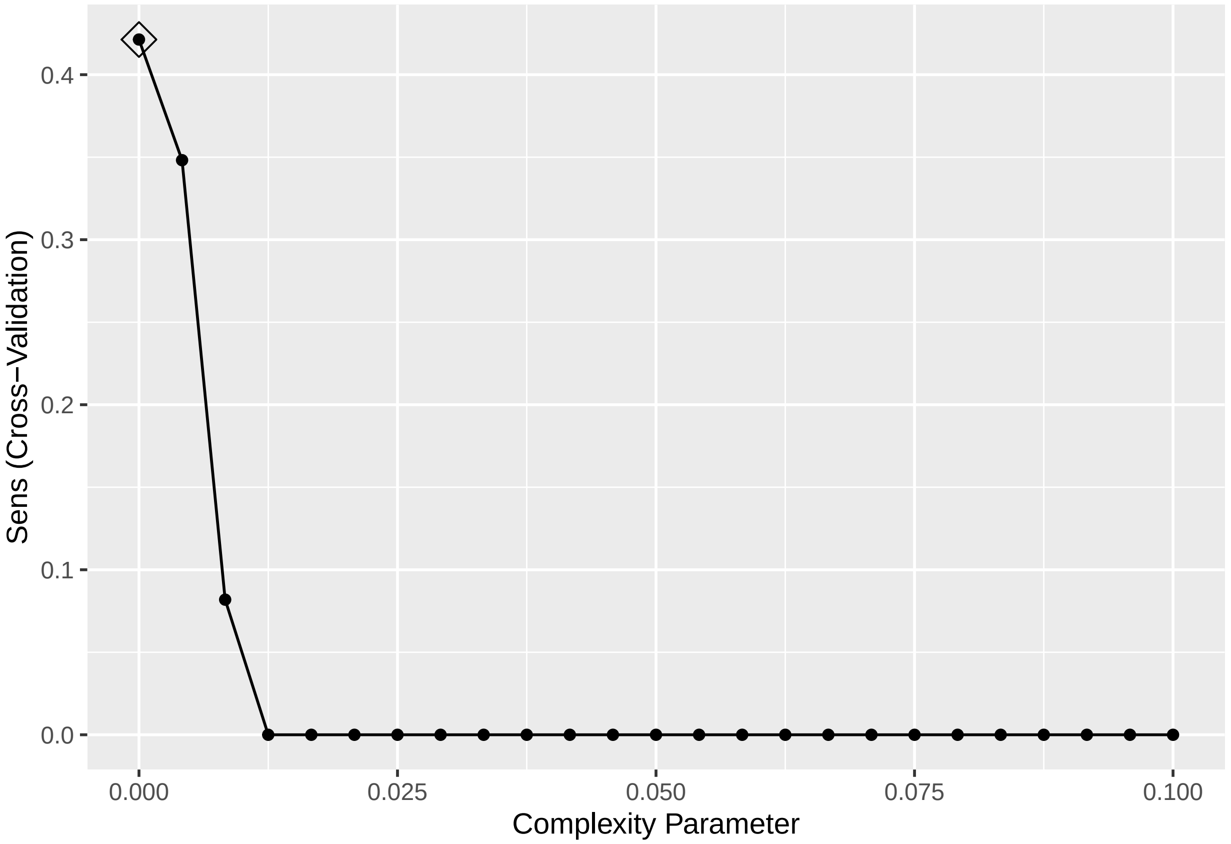
We found that the optimal k value for this model was k=9

QDA - Optimized for Sensitivity

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction TRUE. FALSE.
##     TRUE.    160    423
##     FALSE.   284   1161
##
##                Accuracy : 0.6514
##                  95% CI : (0.6302, 0.6721)
##     No Information Rate : 0.7811
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0839
##
##  Mcnemar's Test P-Value : 2.103e-07
##
##             Sensitivity : 0.3604
##             Specificity : 0.7330
##          Pos Pred Value : 0.2744
##          Neg Pred Value : 0.8035
##              Prevalence : 0.2189
##          Detection Rate : 0.0789
##    Detection Prevalence : 0.2875
##       Balanced Accuracy : 0.5467
##
##        'Positive' Class : TRUE.
##
```

Classification (Decision) Tree - Optimized for Sensitivity

With our Classification Tree model we will repeat the modeling using 25 different values of cp from 0 to 71, at equal intervals.



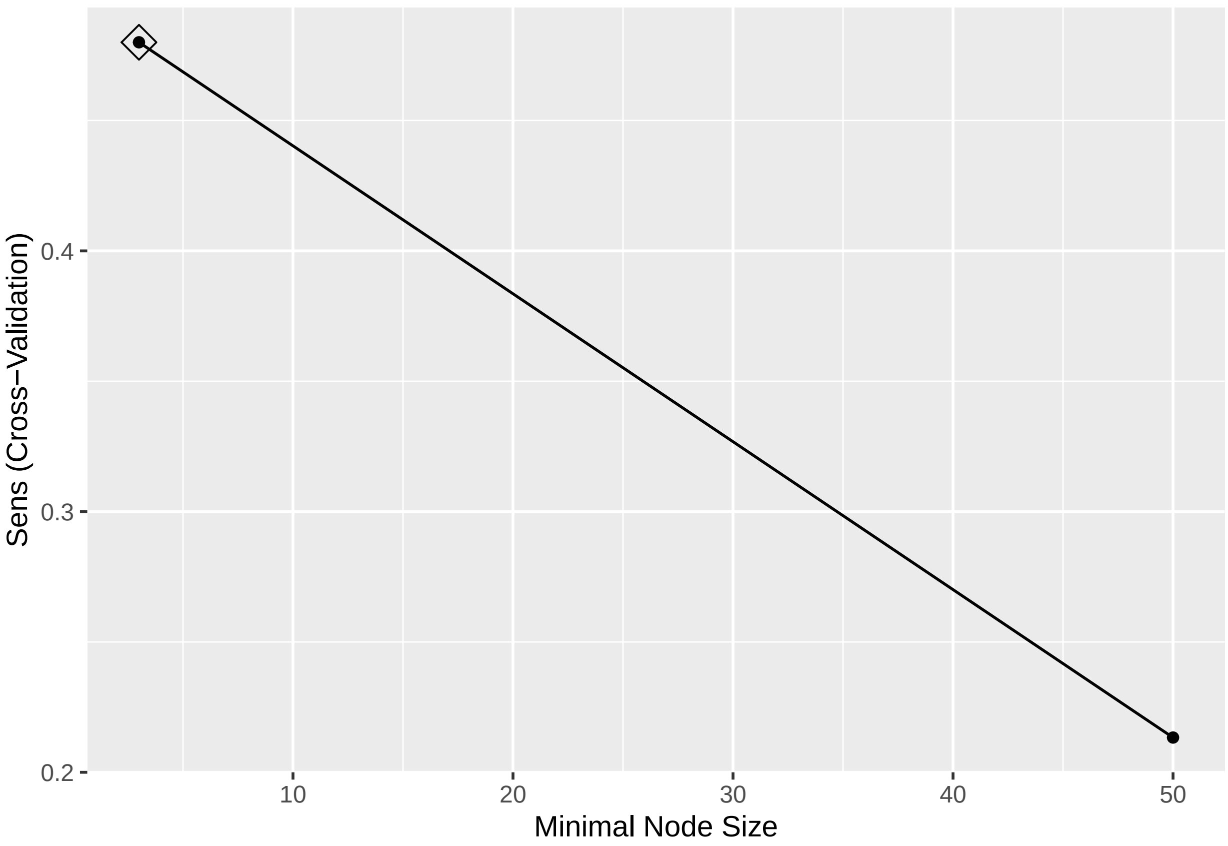
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction TRUE. FALSE.
##     TRUE.    287     96
##     FALSE.   164   1511
##
##                Accuracy : 0.8737
##                  95% CI : (0.8585, 0.8877)
##     No Information Rate : 0.7809
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6097
##
##  Mcnemar's Test P-Value : 3.251e-05
##
##             Sensitivity : 0.6364
##             Specificity : 0.9403
##          Pos Pred Value : 0.7493
##          Neg Pred Value : 0.9021
##              Prevalence : 0.2191
##          Detection Rate : 0.1395
##    Detection Prevalence : 0.1861
##       Balanced Accuracy : 0.7883
##
##        'Positive' Class : TRUE.
##
```

We found that the optimal k value for this model was cp = 0.

Random Forest - Optimized for Sensitivity

TODO: use smaller tuning values

With our Random Forest model we will repeat the modeling using 25 different values of minNode of all integers from 3 to 50.



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction TRUE. FALSE.
##     TRUE.    448      1
##     FALSE.     2   1605
##
##                Accuracy : 0.9985
##                  95% CI : (0.9957, 0.9997)
##     No Information Rate : 0.7811
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9957
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9956
##             Specificity : 0.9994
##          Pos Pred Value : 0.9978
##          Neg Pred Value : 0.9988
##              Prevalence : 0.2189
##          Detection Rate : 0.2179
##    Detection Prevalence : 0.2184
##       Balanced Accuracy : 0.9975
##
##        'Positive' Class : TRUE.
##
```

We found that the optimal k value for this model was minNode=9

Final Model

Process

Looking at our models, we have seen that the optimal model seems to be our Random Forest algorithm with minNode value of 9.

Therefore we will rerun that optimal mode with that min value on our validation set and assess the performance of our model.

Results

Our final model has a sensitivity metric of 0.8790, which means that 0.8790% of all patients who were falling were accurately reported as such. This is a much larger percentage of the

Further Perfomance Metrics:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction TRUE. FALSE.
##     TRUE.   1577    126
##     FALSE.   217   6271
##
##                Accuracy : 0.9581
##                  95% CI : (0.9536, 0.9624)
##     No Information Rate : 0.781
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8753
##
##  Mcnemar's Test P-Value : 1.177e-06
##
##             Sensitivity : 0.8790
##             Specificity : 0.9803
##          Pos Pred Value : 0.9260
##          Neg Pred Value : 0.9666
##              Prevalence : 0.2190
##          Detection Rate : 0.1925
##    Detection Prevalence : 0.2079
##       Balanced Accuracy : 0.9297
##
##        'Positive' Class : TRUE.
##
```

Performance

To assess the speed at which the final model took, we will at the number of seconds that it took to run on a specific computer.

Computer Specifications

MacBook Pro (Retina, 13-inch, Early 2015) Processor: 3.1 GHz Dual-Core Intel Core i7 Memory: 16 GB 1867 MHz DDR3

Time in Seconds

```
## 28.004
```

conclusion

Summary

We see now that the modeling approach outlined multiple times above was somewhat successful and regularization was effective in reducing the RMSE. During the final modeling, the RMSE was recorded of the simplified model using only the mean of all ratings, which we will call the "Baseline" RMSE. We will now see how that compares to the final RMSE and use that as a baseline to see how effective our optimization techniques were.

Limitations

The models used in this paper where quite simplistic analysis. For a more nuanced and better performing model we could potentially also reduce the amount of false positives to more avoid wasting the time of medical professionals and patients responding to non-existent falls.

Future work

In the future, I hope to continue this work, using a higher-performing machine with more rigorous models for predicting movie ratings. Further, since the distribution of variables were not normally distributed and did not have the same mean values, a future study should use mean normalization and feature scaling in hopes of getting a more accurate model. Also, doing a Principle Component Analysis, we can see that the cumulative proportion of variance is greater tham .95 at PC4. This indicates that there is the ability to do singular value decomposition to reduce the amount of data being used while maintaining a variance of > .95, which is a recommended level of variance to keep when doing SVD

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.9937 1.0165 0.9949 0.8524 0.48270 0.17112 0.11555
## Proportion of Variance 0.5678 0.1476 0.1414 0.1038 0.03329 0.00418 0.00191
## Cumulative Proportion  0.5678 0.7154 0.8568 0.9606 0.99391 0.99809 1.00000
```