EdX Data Science Capstone Choose Your Own Project

JWL

April 23, 2019

This report is submitted for the Choose Your Own Project for the HarvardX: PH125.9x Data Science Capstone course.

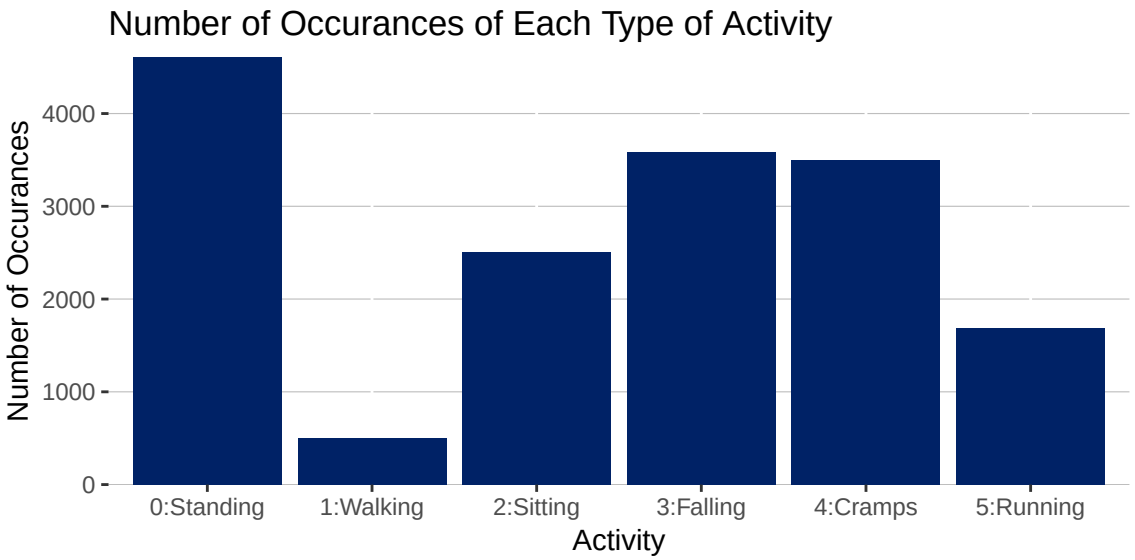
Executive Summary

The dataset for this project was downloaded from Kaggle and contains fall detection data from China. The dataset has seven variables, including the outcome variable of interest, *Activity* The goal of the excerise is to predict when the individuals fell based on the values of the other variables. The data contains information from fourteen volunteers, 2,520 trials, with a trial being a standard set of movements performed by the volunteers in addition to voluntary falls. The data is measure by sensors every 4 seconds, but only the measurements when the acceleramtor measure peak acceleration were included. Falls comprised 21.9% of all records. The sensors measured sugar level (SL), electroencephalogram (EEG) values, blood pressure, heart rate, blood circulation. Most of the predictors measured the heart. The dataset also included a variable about time, though not enough other information was provided about the variable to know if it would be useful. There was significant skewness in the values of sugar level and EEG, so much so that it's possible some measures were erroneous. To control for this the outliers for those variables were capped. Since the definition of the time variable was not clear, meaning it was not clear how it might relate to when the fall occured, it was dropped.

Since the outcome of interest was a binary classification, whether the person fell or not, only models suited to classification were tested. The models tested were logistic regression, random forest, and K nearest neighbors. The random forest and K nearest neighbors models also had their parameters tuned to produce the highest accuracy. The level of area under the curve (AUC) was also examined and the model which produced the highest accuracy and AUC was random forest. The random forest model produced a relatively high AUC of 0.78488 and was able to predict with an accuracy of 0.86512.

Methods

The full file contained 16382 measurements. Fourteen volunteers conducted 2520 trials in total, though there was no volnteer variable in the data to differentiate them, nor a variable to identify the trials as there were more records than trials. The outcome of interest was whether the volunteer had fell, this information was captured in the ACTIVITY variable.

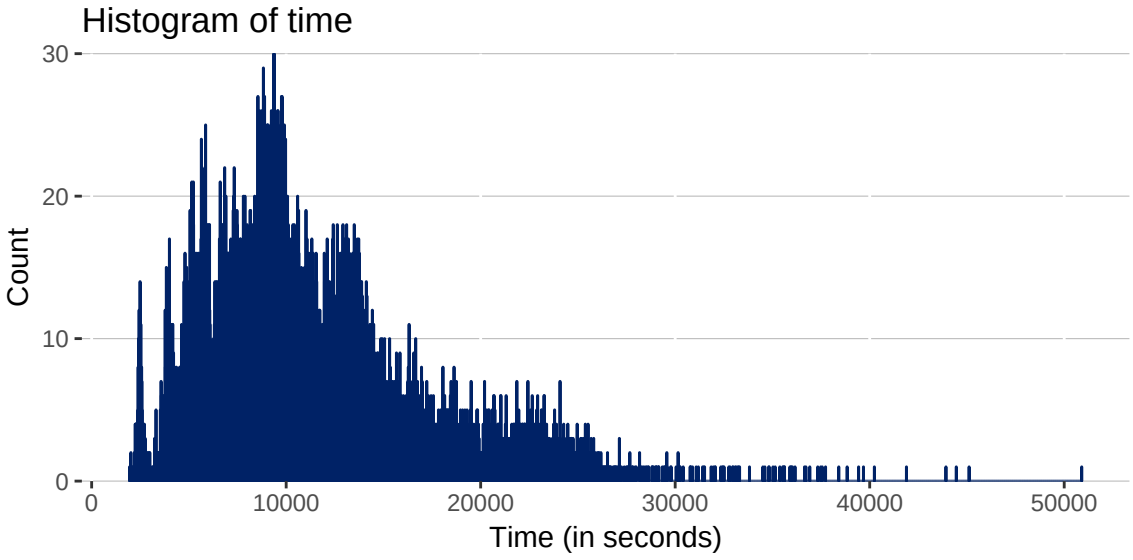


Falls or falling constitued 21.9% of all records. Falls constituted the second most likely activity behind Standing. There were a similar number of Cramps, followed by sitting and then running. There were very few instances of walking. Since all we are interested in was whether the individual was falling, the ACTIVITY variable was converted into a dummy variable called *fall*. The *fall* variable was also converted to an ordered factor. It was converted to an ordered factor so the area under the curve (AUC) could be calculated.

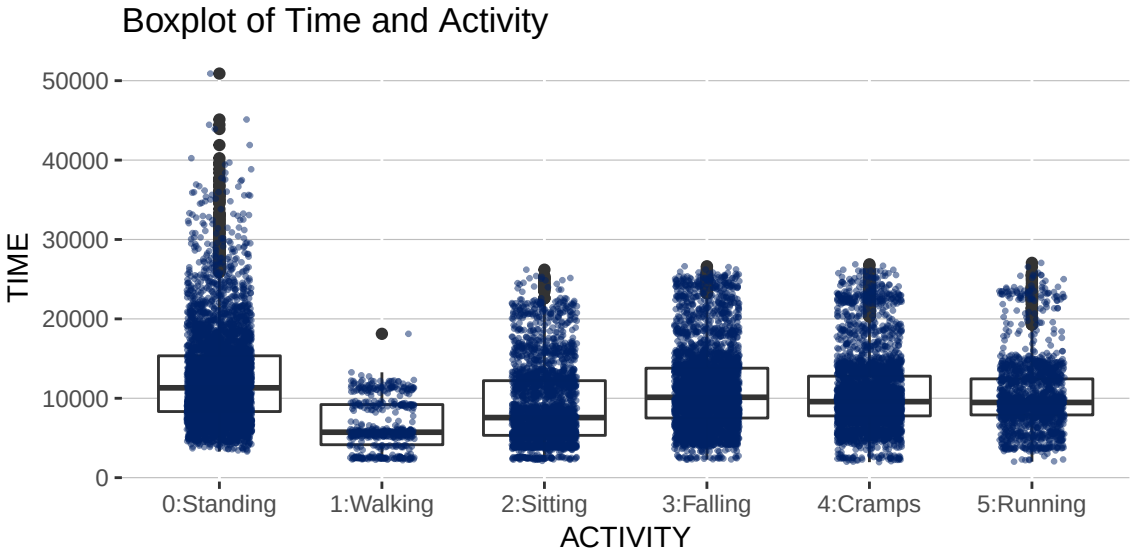
Overall there was little need for dimension reduction since there are only a few predictors. However, the predictors were checked for near zero variance using the caret packages nearZeroVar function. Removing predictors with near zero variance improves model speed and efficiency without sacrificing predictive power. No predictors were found to have near zero variance. All predictors were also checked for missing values and no predictor had missing values.

As part of the data wrangling process I created combination box plots and scatter plots to look at the relationship between each of the six possible predictors and the Activity category. Since the activity variable was categorical and the other predictors were continuous then box plots were used to show the relative means and interquartile range for each predictor and each category of activity. The individual values were also added as a scatter plot on top with the geom_jitter function added so they were not all on the same vertical line.

Relationship between Time and Activity




```
## Warning: Ignoring unknown parameters: utlier.shape
```



Looking at the boxplot of Time compared to Activity value there does not appear to be a strong relationship. It's also unclear what the TIME variable is relative to, making it difficult to know how it could be predictive. Due to this, it was deleted. Next the six remaining predictor variables were cleaned up. These predictors are values from sensor readings that measure different body functions. Because the quality of the data was unknown, some predictors were top coded or bottom coded as the data was significantly skewed. This skewness could be due to errors in the sensor readings or sensor miscalibration.

Relationship between Sugar Level and Activity

```r
summary(falls$SL)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      42.2    9941.2   31189.2   75272.0   80761.4 2426140.0
```

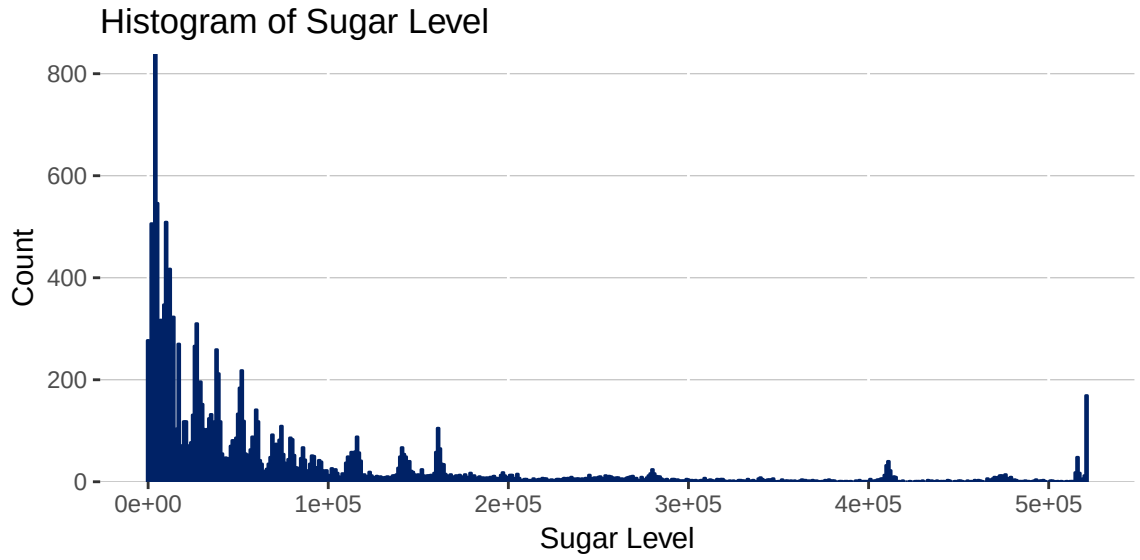
```r
quantile(falls$SL,c(0.01,0.1,0.5,0.9,0.95,0.99))
```

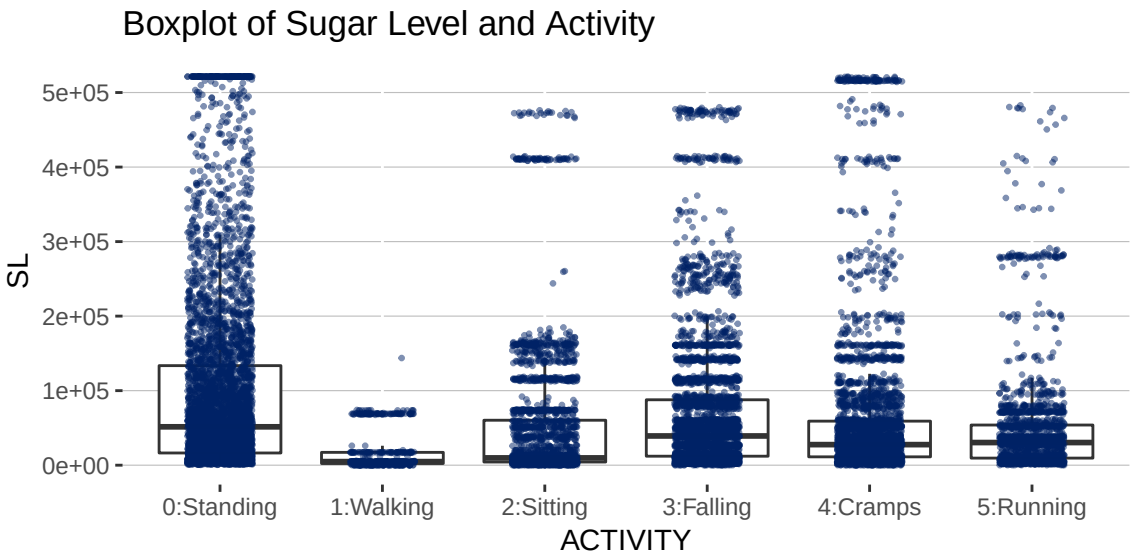
```
##         1%        10%        50%        90%        95%        99%
##     52.820   4051.841  31189.200 180354.100 309347.600 521414.680
```

```r
skewness(falls$SL)
```

```
## [1] 4.708005
```

The sugar level (SL) predictor was significantly positively skewed with a 99th percentile value of 521,414, a median of 31,189 and a mean of 75,272. To correct for possible sensor errors this value was top coded at the 99th percentile, meaning any value above that was set to the 99th percentile value. This top coded SL variable was then graphed.





Relationship between EEG and Activity

```
summary(falls$EEG)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -12626000     -5630     -3361     -5621     -2150   1410000
```

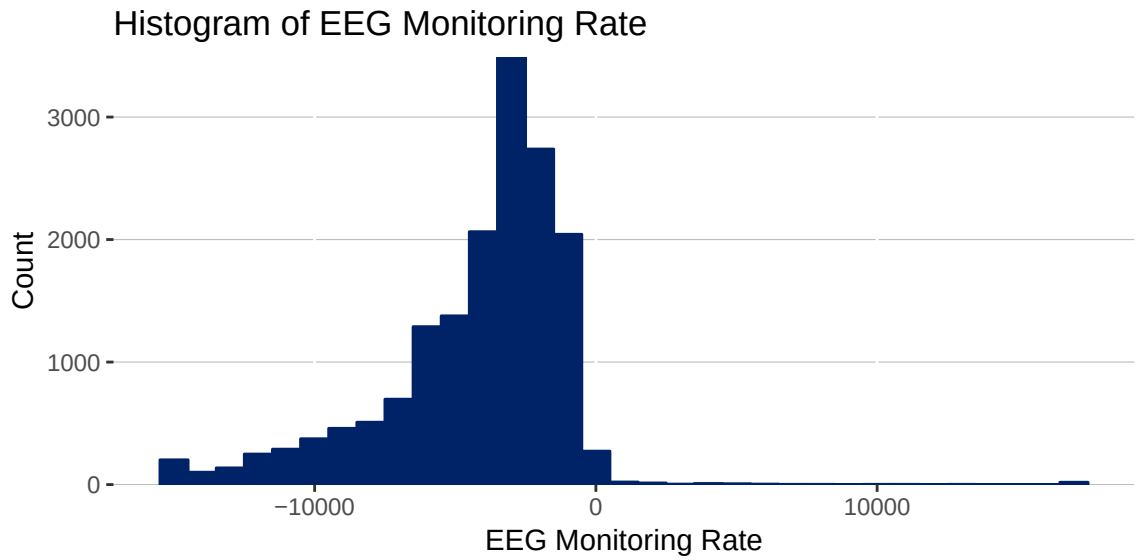
```
quantile(falls$EEG,c(0.001,0.01,0.1,0.5,0.9,0.95,0.99,0.999))
```

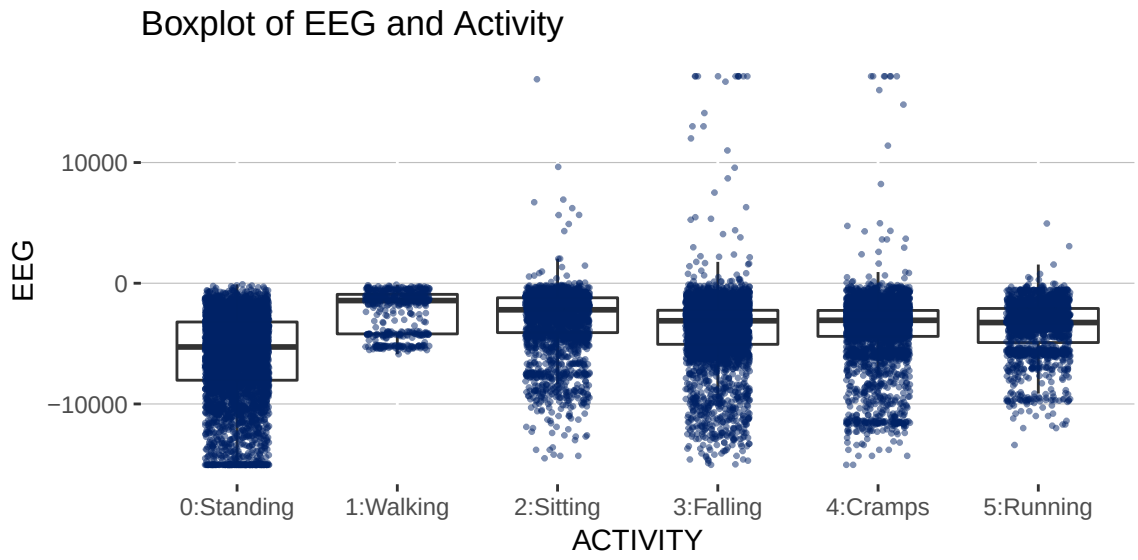
```
##        0.1%          1%         10%         50%         90%         95%
## -25305.2370 -15043.3800  -8870.0000  -3361.2750  -1169.9460   -839.1564
##         99%       99.9%
##   -236.8100  17147.6000
```

```
skewness(falls$EEG)
```

```
## [1] -100.8115
```

The EEG values were significantly negatively skewed. However, it also had a very high maximum value, so the 99.9th percentile was checked as well. The EEG variable was then bottom coded at the 0.1st percentile, meaning any value below that was set to -15,043 and top coded at the 99.9th percentile, meaning value value above that was top coded at 17,148.





Relationship between Blood Pressure and Activity

```
summary(falls$BP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   25.00   44.00   58.25   78.00  533.00
```

```
quantile(falls$BP,c(0.001,0.01,0.1,0.5,0.9,0.95,0.99,0.999))
```

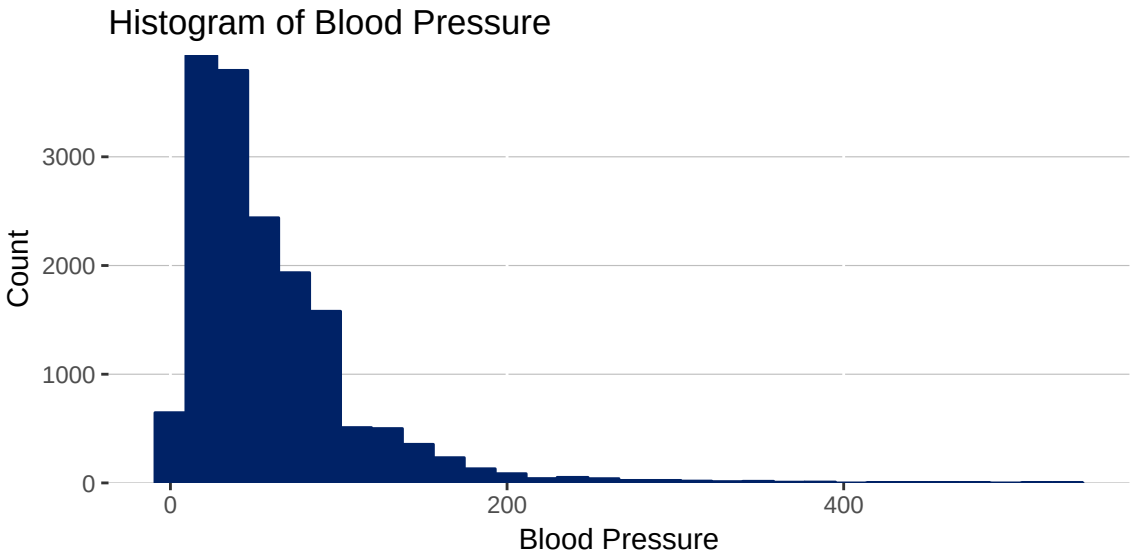
```
##    0.1%      1%     10%     50%     90%     95%     99%   99.9%
##   0.000   5.000  15.000  44.000 116.000 148.000 245.000 378.619
```

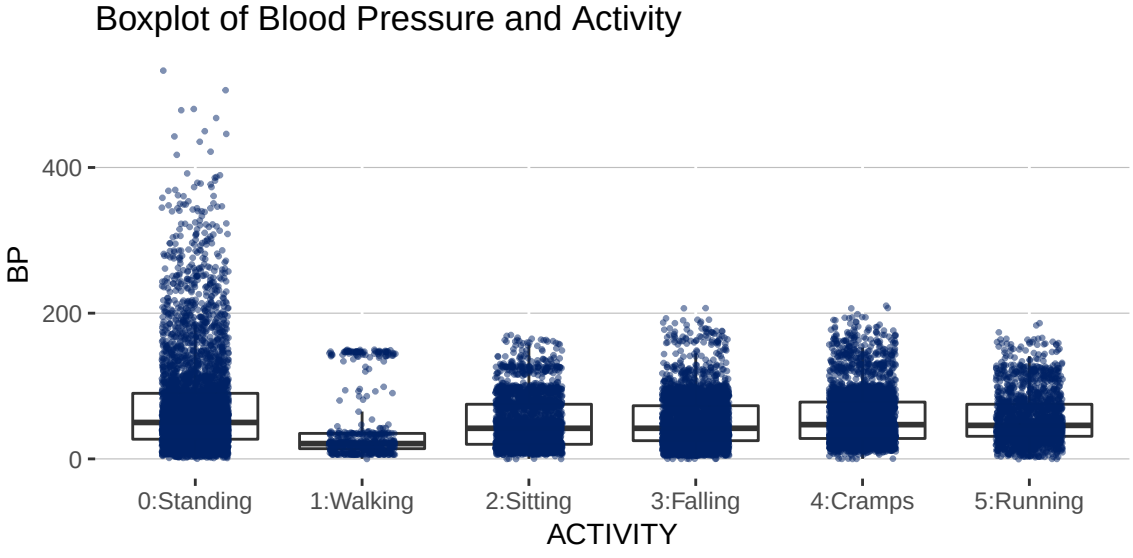
```
skewness(falls$BP)
```

```
## [1] 2.361218
```

The blood pressure values are not nearly as significantly skewed as the other predictors so no changes were made.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Relationship between Heart Rate and Activity

```
summary(falls$HR)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    33.0   119.0   180.0   211.5   271.0   986.0
```

```
quantile(falls$HR,c(0.001,0.01,0.1,0.9,0.95,0.99,0.999))
```

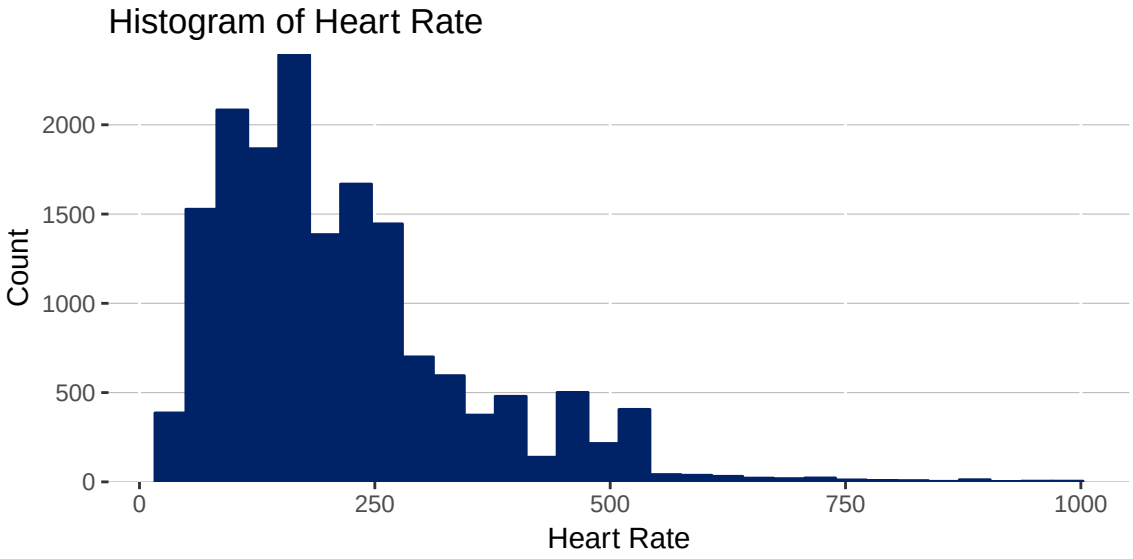
```
##    0.1%      1%     10%     90%     95%     99%   99.9%
##  33.000  33.000  79.000 393.900 482.950 585.190 877.476
```

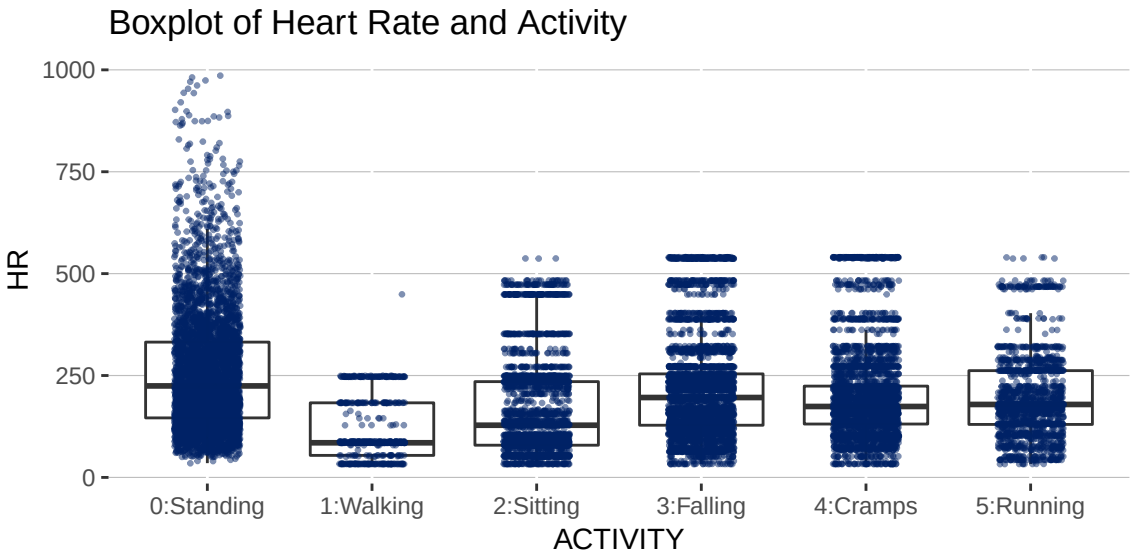
```
skewness(falls$HR)
```

```
## [1] 1.329404
```

The heart rate values are not skewed either so no changes were made to the heart rate variable.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Relationship between Circulation and Activity

```
summary(falls$CIRCLUATION)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       5     587    1581    2894    3539   52210
```

```
quantile(falls$CIRCLUATION,c(0.001,0.01,0.1,0.9,0.95,0.99,0.999))
```

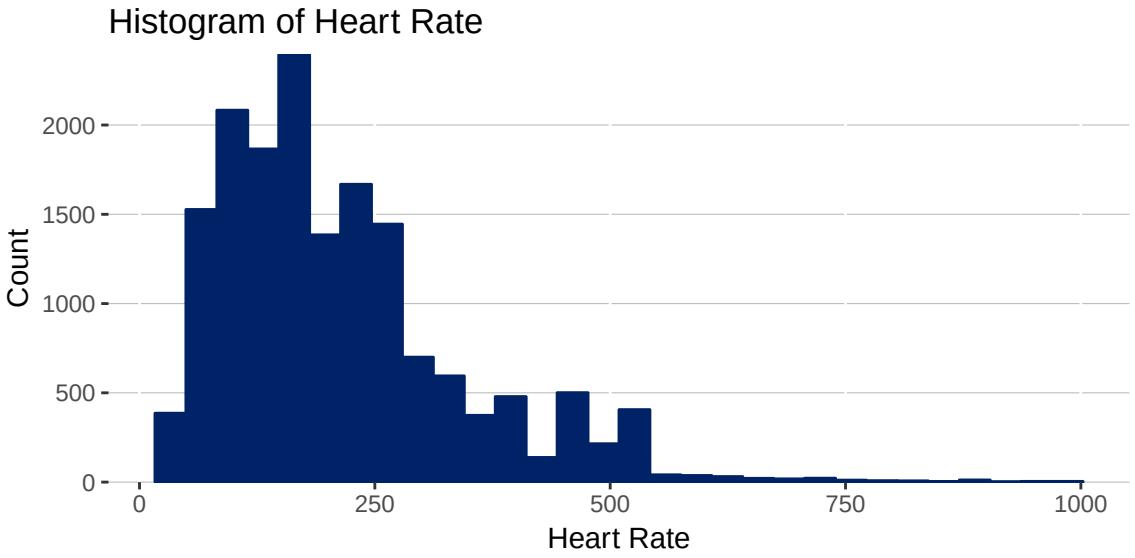
```
##     0.1%       1%      10%      90%      95%      99%    99.9%
##     5.00     5.00   249.00  6746.00 10237.90 18067.00 32299.86
```

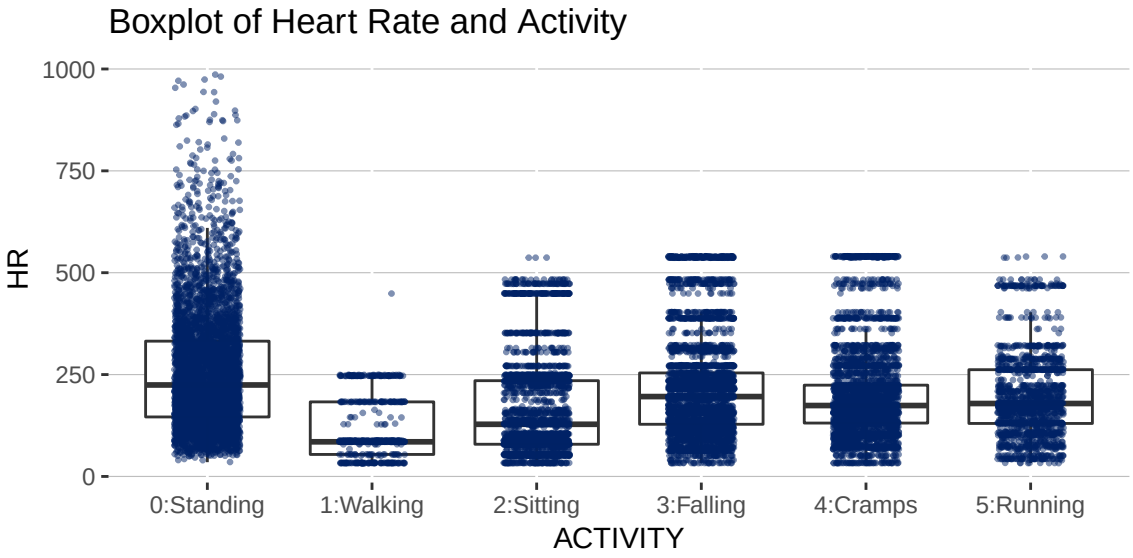
```
skewness(falls$CIRCLUATION)
```

```
## [1] 3.111124
```

The circulation values are not significantly skewed so no changes were made.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





While the relationship was not always clear looking at the boxplots, there did appear to be variation in each predictor by activity, therefore all predictors (except Time) were kept. After examining the relationship between the Activity variable and the other predictors the Activity variable was deleted since only interested in falls. It was not necessary to take a sample to run analysis on since the full dataset was only approximately 16,000 observations and the number of variables as low. Now that the final training dataset was created, the model was developed.

Model development

The first step was to divide the dataset into a training and test set, with 80% of observations allocated to the training set.

The outcome in question here, whether the person fell, is binary. Since the outcome is binary then models such as logistic regression and classification machine learning models such as K nearest neighbor (KNN) and random forest models are appropriate. Since there are several predictors then models such as Naive Bayes or Linear Discriminate Analysis (LDA) are not a good fit (Irizarry 2019). As the outcome is binary the metric used to evaluate will be the area under the Receiver Operating Characteristic (ROC) Curves or AUC and the accuracy, which is the percentage of values predicted correctly. The AUC metric helps to balance desires for specificity and sensitivity.

A logistic regression model was fitted on the train dataset and then estimates of the rating (y hat) were generated on the test data set. All of the predictors were included in the model and no transformations were made. This produced an AUC of .

Other machine learning algorithims are rule based or decision tree models. These decision tree models create rules whereby certain values within cutoffs of the predictors result in different predictions. These rules are multilevel, so that different predictors align in a tree with different rules as to when to predict certain values. A row of data's predictor values will then fall down one branch resulting in a prediction, which for regression trees is the mean value of the outcome variable for all the observations that fit that branch's criteria. These rules are created to minimize the residual sum of squares (RSS). The caret package by default will use cross validation to decide the best tuning parameter value when the train function is used. The default is 25 bootstrap samples of 25 percent of the observations (Irizarry 2019).

A potentially improved version of regression tree models are random forest models. Random forest models average the predictions of a number of trees created from different bootstrap random samples. When they make a split for a new tree they don't look at all predictors but just a random sample of them (Irizarry 2019). There are multiple models for random forests in the caret package and they have different tuning

parameters. First the Rborist method was used and the tuning parameter was the minNode, which is the number of nodes or values in a tree that are needed to be split. Predfixed was set to 2 and values for minNode between 1 and 10 were tried. The value with the highest accuracy RMSE was when minNode was equal to 1. This produced an accuracy of 0.8528, higher than logistic regression. To attempt to improve on this the rf method was used and the mtry paramter was tuned. mtry is the number of variables randomly sampled as candidates at each split and is equiavalent to predFixed in the Rborist method. This intially took too long to run, so the trainControl caret function was used to limit the number of cross validations to 5 of 90 percent of the observations. A good rule of thumb on the best value of mtry to test is the number of predictors divided by 3, which in this case is 2. mtry values of 1, 2, and 3 were tried. The best value of mtry was found to be 1 and this produced a high AUC of 0.78488.

The last model tested was the KNN model, or K nearest neighbors model. This model looks for similar observations in the dataset, observations that are close in distance. For this model the parameter to tune is K, with larger K values leading to smoother estimates and smaller Ks resulting in more flexible estimates (Irizarry 2019). Again the caret package train function was used and the number of cross validation samples used was 2, each comprised of 90 percent of the observations to limit computation time, which was significant for the KNN model. Values of K ranging from 3 to 35 in increments of 10 were tested. The optimal K value was found to be 15 and this produced an AUC of 0.68854.

Results

The model which produced the highest accuracy and the highest AUC was the Random Forest model. It was able to predict 87% of fall instances in the test dataset.

Table of RMSEs and models

method	Accuracy	AUC
Logistic Regression	0.7818126	0.5065980
Random Forest	0.8651205	0.7848837
KNN	0.8132438	0.6885429

Conclusion

Overall given the small number of predictors and overall small sample size an AUC of 0.78488 is satisfactory. An AUC of 0.5 would be a completely random guess and an AUC of 1 would be perfect. It's possible that given additional time more preprocessing could have been attempted or more models could have been used. However, most of the other possible models were out of the scope of this EdX course. The significant outliers of sugar level and EEG were perplexing, but without more information it's unclear if those values could be trusted. Bottom or top coding their values preserves the records while reducing the amount of damage they could do to the models if they were measurement errors. The variable importance of the different predictors in the Random Forest model was examined using the *varImp* function in the *caret* package. Measures of importance are scaled with a maximum value of 100. For the model found in this report, the sugar variable had a variable importance value of 100, EEG had a value of 52, BP had a value of 21, circulation had a value of 7 and HR had a value of 0. This does not seem intuitive, since the falls were voluntary it's unclear why blood sugar levels would be predictive. However, looking at the box plot falls did have a relatively high mean and more values in the higher range for sugar level. While the Random Forest model was fairly predictive, the quality of the data would seem to be suspect and worth further inquiry.