IST 687 - Final Project

Thomas Bahng

13 June, 2019

Table of Contents

Introduction

It is widely understood that smart phones are constantly collecting data. Users can create data directly using their smart phones by taking pictures, recording video, creating notes or events on their calendars. However, these direct acts of generating data may only be scratching the surface of how much data is being collected overall by our smart phones. This project seeks to validate how data that was indirectly collected by human interaction with their devices can be used to make predictions about human activity.

Activities such as sitting, standing, and walking describe what people do everyday with their smart phones, and even though this project focuses solely on these explicit activities, the context in which these activities are done may be even more valuable to those studying human behavior. As technology advances into augmented reality (AR) and as more and more activities are done using smart phones "outside" rather than sitting in front of a computer, these devices truly become close companions that understand people more than ever.

Business Question

The business wants to better understand dependencies between mobile product utilization and the physical activities of users. Can smart phones be used to predict active daily living activities? Which features are most important in predicting class labels overall?

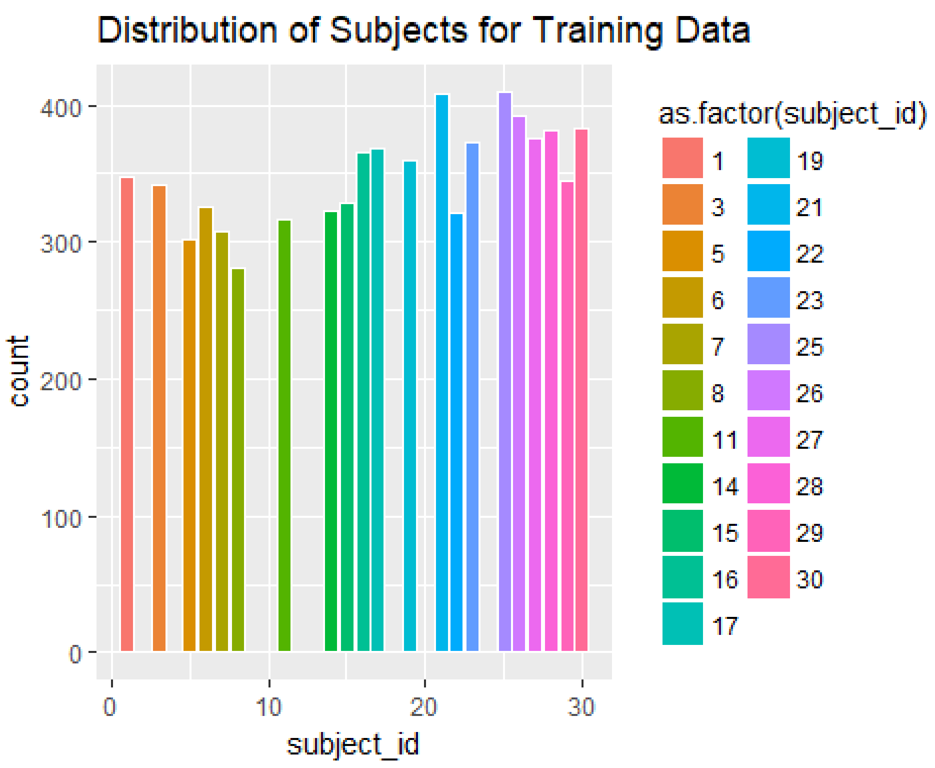
Data

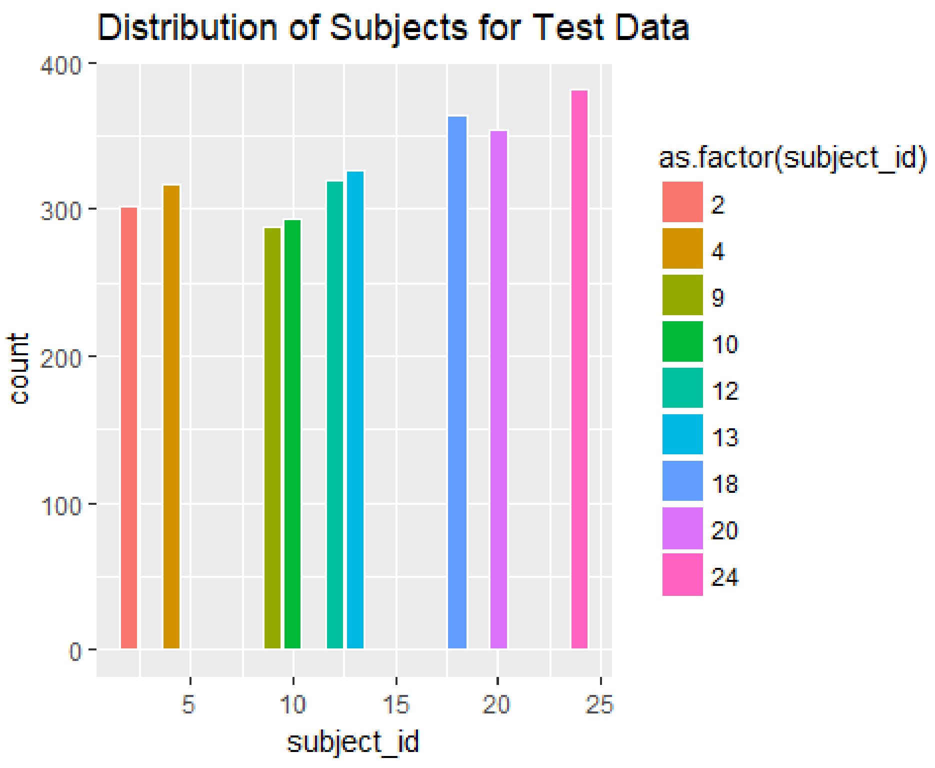
The dataset used is from the Human Activity Recognition database built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors such as the gyroscope and accelerometer. ADL consists of the following activity labels:

```
## [1] "1 WALKING | 2 WALKING_UPSTAIRS | 3 WALKING_DOWNSTAIRS | 4 SITTING | 5
STANDING | 6 LAYING"
```

The full dataset contained 29 text files with some files serving as information about the data. Of these 29 files, 6 were in an aggregated form and thus used to bind and create the training and test sets.

The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data. Here we can see that split through visualization of subject IDs recorded in the training and test sets.



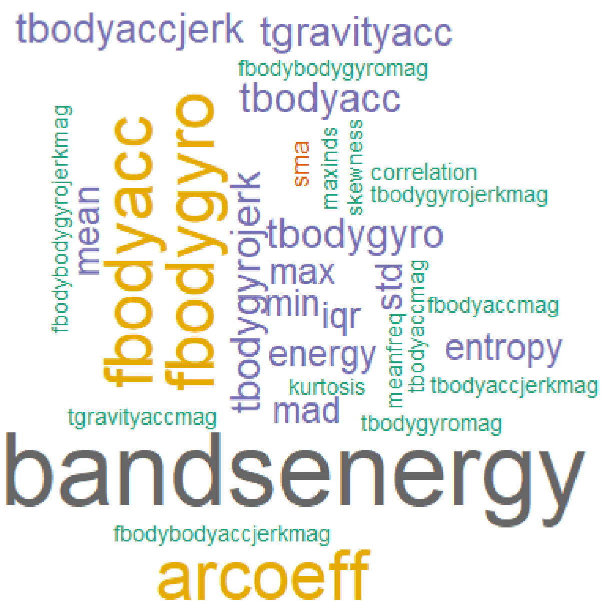


Each observation in the dataset contained:

- A 561-feature vector with time and frequency domain variables.
- Its activity label.

- An identifier of the subject who carried out the experiment.

A Word Cloud of the Features



Feature Information

These signals were used to estimate variables of the feature vector for each pattern:
'-XYZ' is used to denote 3-axial signals in the X, Y and Z directions.

```
## [1] "tBodyAcc-XYZ | tGravityAcc-XYZ | tBodyAccJerk-XYZ | tBodyGyro-XYZ |
tBodyGyroJerk-XYZ | tBodyAccMag | tGravityAccMag | tBodyAccJerkMag |
tBodyGyroMag | tBodyGyroJerkMag | fBodyAcc-XYZ | fBodyAccJerk-XYZ |
fBodyGyro-XYZ | fBodyAccMag | fBodyAccJerkMag | fBodyGyroMag |
fBodyGyroJerkMag"
```

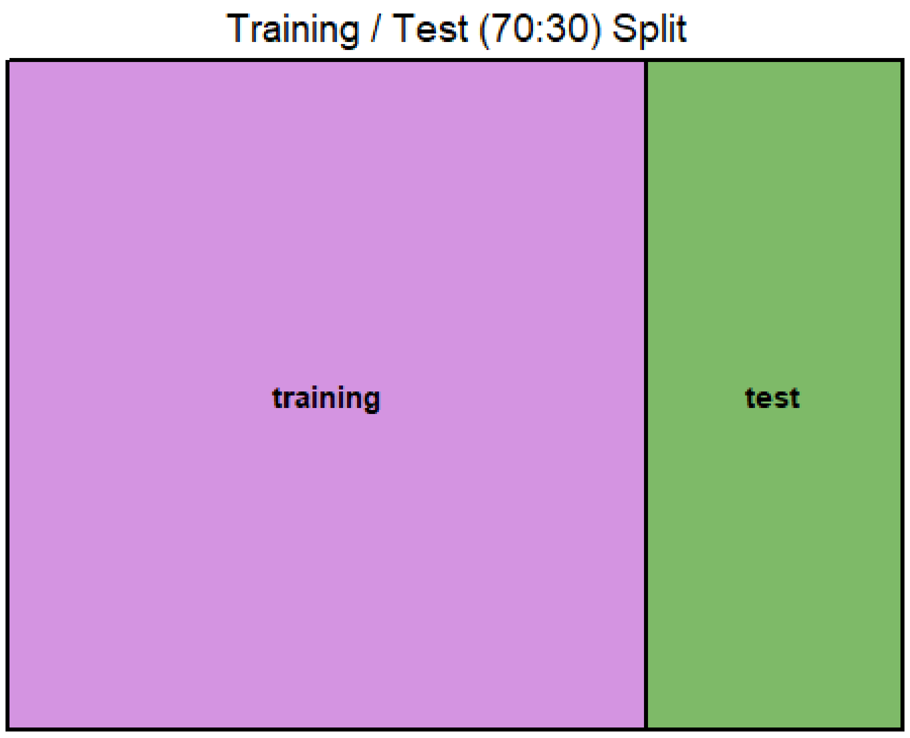
The set of variables that were estimated from these signals are:

Variable	Description
mean()	Mean value
std()	Standard deviation
mad()	Median absolute deviation
max()	Largest value in array
min()	Smallest value in array
sma()	Signal magnitude area
energy()	Energy measure. Sum of the squares divided by the number of values.
iqr()	Interquartile range
entropy()	Signal entropy

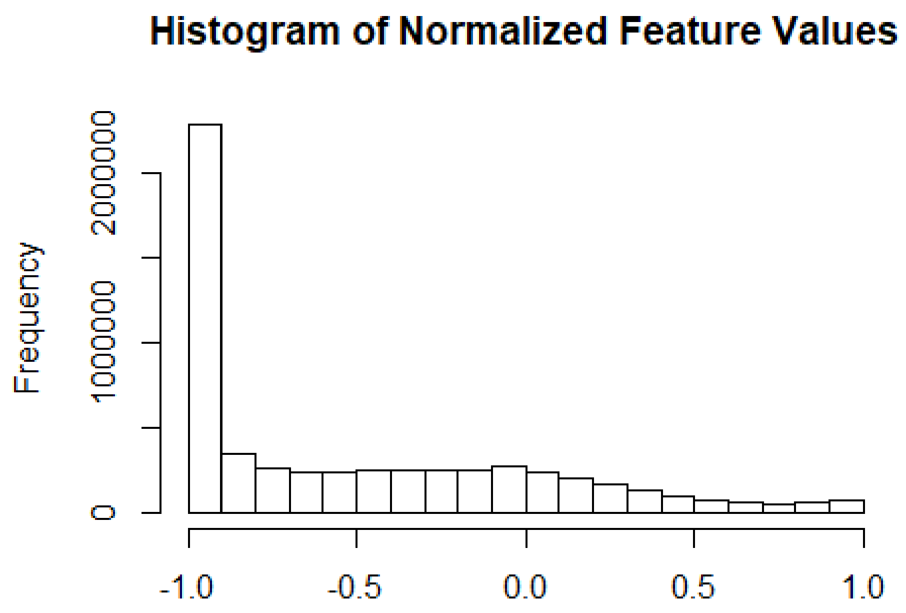
arCoeff()	Autorregresion coefficients with Burg order equal to 4
correlation()	correlation coefficient between two signals
maxInds()	index of the frequency component with largest magnitude
meanFreq()	Weighted average of the frequency components to obtain a mean frequency
skewness()	skewness of the frequency domain signal
kurtosis()	kurtosis of the frequency domain signal
bandsEnergy()	Energy of a frequency interval within the 64 bins of the FFT of each window.
angle()	Angle between to vectors.

Exploration

The training data consists of 7352 rows and 563 columns. The test data consists of 2947 rows and 563 columns. The plot below illustrates the relative scale of the two data sets.



Both training and test data have been normalized so that all values range between -1 and 1. The distribution of the full dataset is shown below:

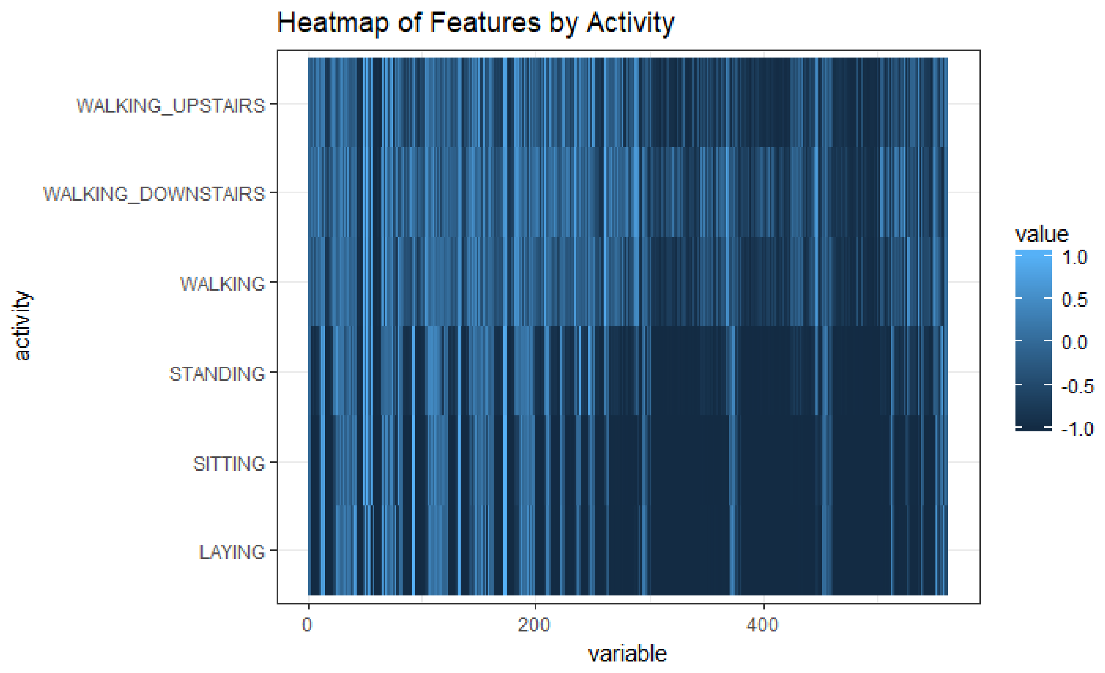


Quantiles of the feature vector

```
##           0%          25%          50%          75%          80% 
## -1.000000000 -0.987597695 -0.704334730 -0.110379050 -0.006862302 
##          85%          90%          95%         100% 
##  0.112881206  0.267911278  0.517131144  1.000000000 
```

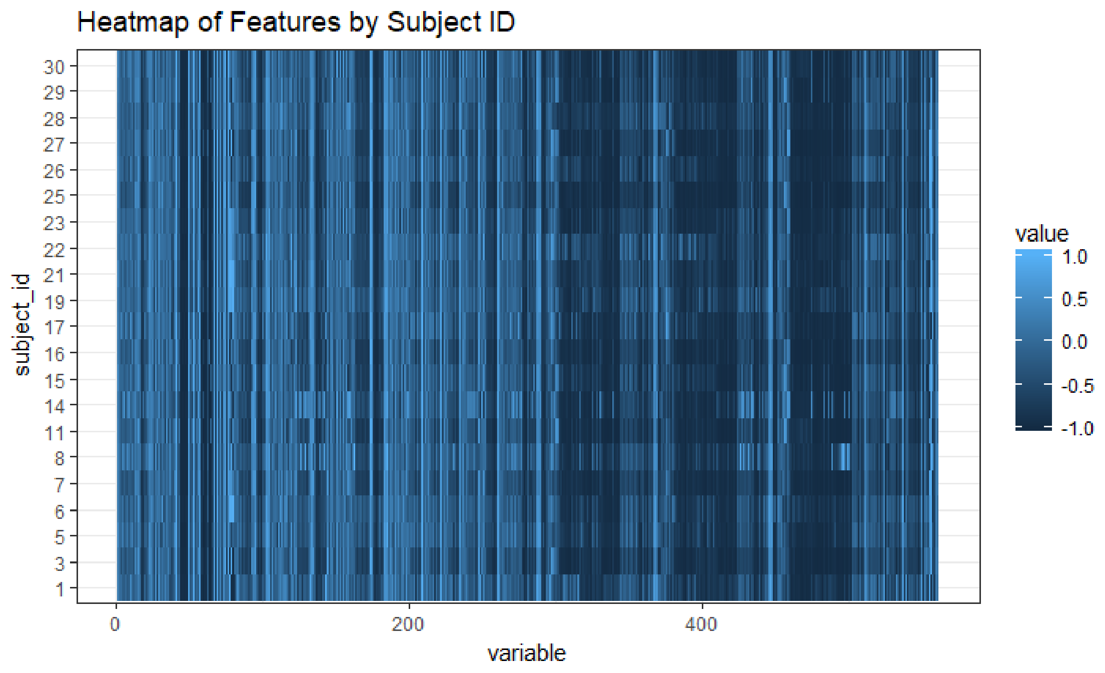
Over 80% of the values in the dataset are in the negative range. The mean value of the feature vector is -0.51 and the standard deviation is 0.53.

Next we visualize how the values differ by class label with a heat map. (561 variables shown along x-axis; class labels along y-axis)



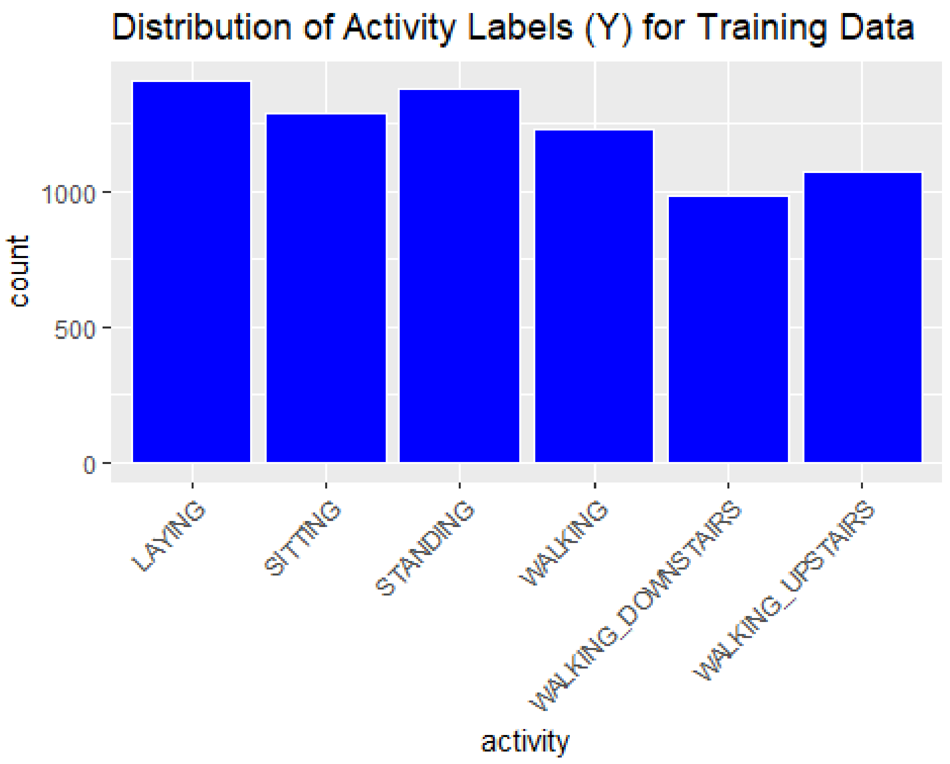
We can clearly see that sedentary activities (i.e. STANDING, SITTING, LAYING) contain more values in the negative range than do motion activities.

Below is the same plot except by subject ID. Here we are comparing subjects based on their numeric features in the data gathered.



Although minor differences clearly exist in the data respective to subject ID, overall it's safe to say that all subjects displayed similar patterns across all the features.

Lastly, investigate to see that there is representative sampling across all classes in the training set.



There are at least about 1000 observations per class label. This distribution is roughly equal across classes, although the classes associated with motion have relatively fewer observations than do the sedentary activities.

Modeling

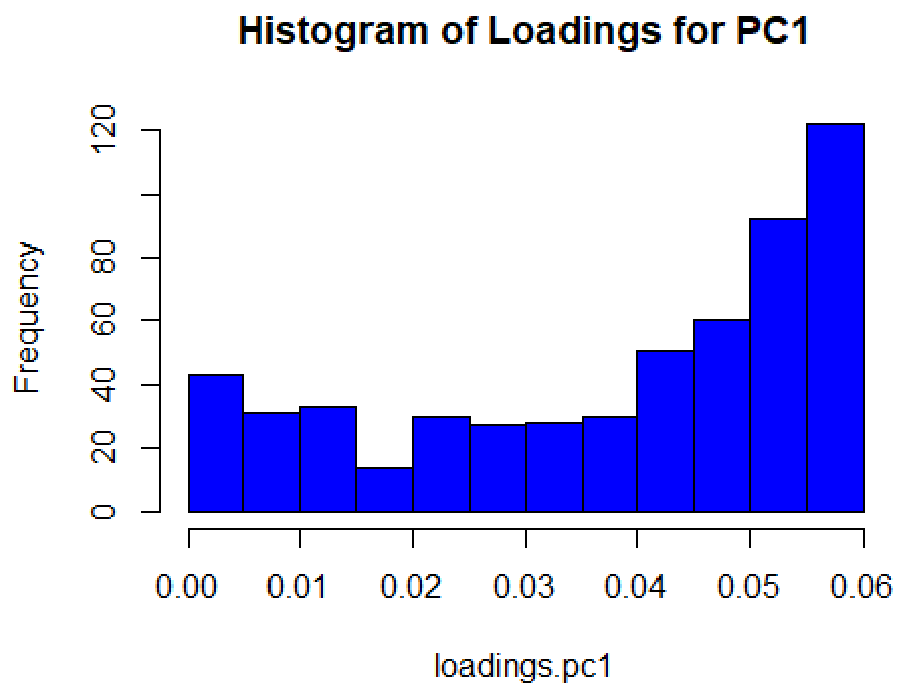
Different modeling techniques were used to answer our business questions. Principal component analysis (PCA) was used as an unsupervised learning approach to understand feature importance as well as to evaluate the effectiveness of a classifier built on the principal components. We compared the results to modeling just on all of the features in the original training set to determine the best model and the final recommendation.

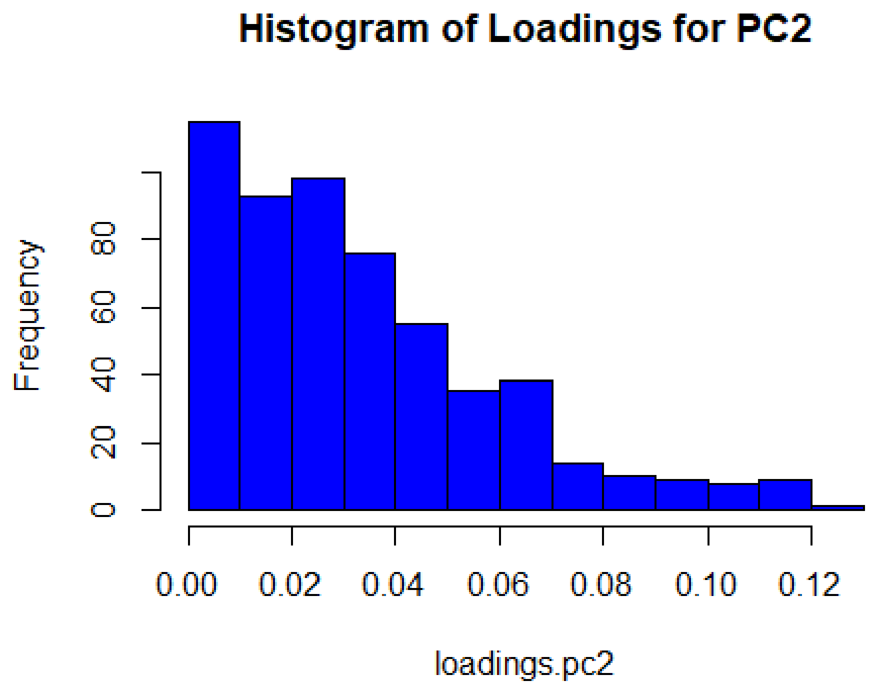
Unsupervised Learning

PCA and Feature Importance

Pre-processing for PCA requires identifying near-zero-variance features in the dataset; these are variables that are either constants or have very few unique values. In our case there were none using a percentage-unique threshold of 10% and thus no variables were excluded. Variables were then centered and scaled for the principal component analysis.

Once the principal components were created, a quick look into the distribution of absolute loading values tell us that none of the features loaded (i.e. correlated) very high on the first two principal components. However, we can see that some features stand out more than others in their importance.





The first distribution has a left skew meaning that there were more features in the original data that were in the higher range of values and most important on the first principal component (PC1). The number of features with absolute loading values greater than 0.05 were 214. These were the top 10 most important features for PC1:

```
##             fBodyAcc-sma()          fBodyAccJerk-sma()
##                 0.05857110                  0.05856808
##         tBodyAccJerk-sma()             fBodyGyro-sma()
##                 0.05853759                  0.05853133
##     tBodyAccJerkMag-mean()      tBodyAccJerkMag-sma()
##                 0.05849275                  0.05849275
## fBodyBodyAccJerkMag-mean()  fBodyBodyAccJerkMag-sma()
##                 0.05812810                  0.05812810
##      tBodyAccJerkMag-mad()      tBodyAccJerkMag-std()
##                 0.05801689                  0.05798475
```

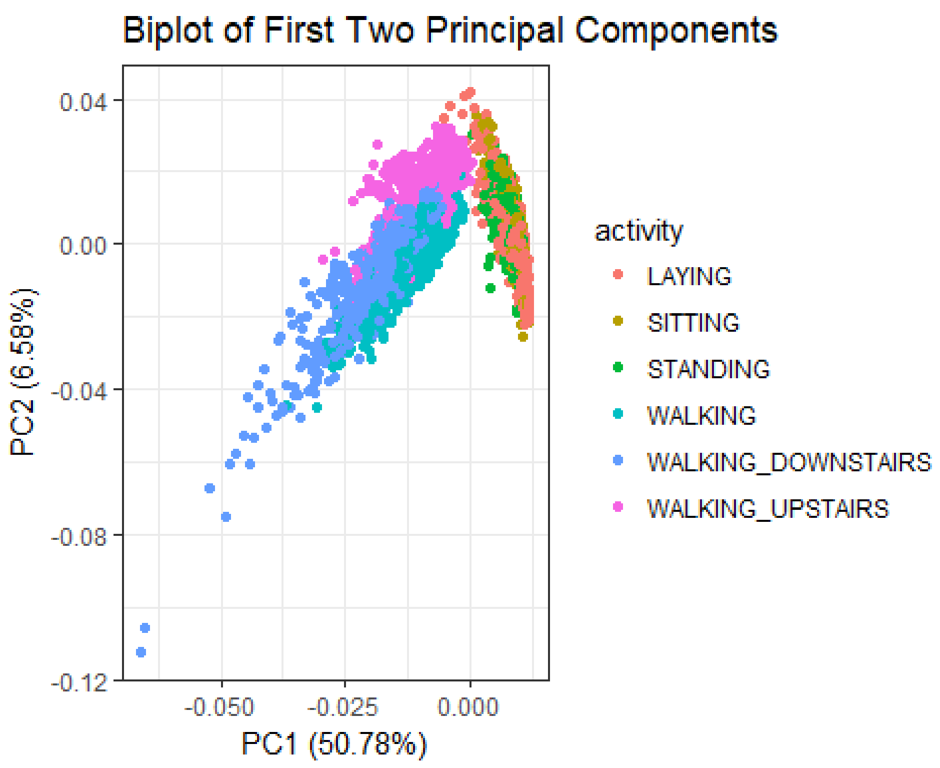
The second distribution has a right skew meaning that there were more features in the original data that were in the lower range of values and least important on the second principal component (PC2). The number of features with absolute loading values greater than 0.05 were 124. These were the top 10 most important features for PC2:

```
##     fBodyAcc-meanFreq()-Z   tBodyGyroMag-arCoeff()1
##                 0.1223853                 0.1197829
##    tBodyAccMag-arCoeff()1 tGravityAccMag-arCoeff()1
##                 0.1181618                 0.1181618
##    fBodyAccMag-meanFreq() tGravityAcc-arCoeff()-Z,1
##                 0.1178545                 0.1175610
## tGravityAcc-arCoeff()-Z,2 tGravityAcc-arCoeff()-Z,3
##                 0.1170931                 0.1152887
## tGravityAcc-arCoeff()-Z,4   tBodyGyroMag-arCoeff()2
##                 0.1120128                 0.1118668
```

It can be said that there are hundreds of important features based on this principal components analysis as there isn't much overlap between the top features for PC1 and PC2. In PC1, the top 75% of the features can vary in importance by a factor of 2.5. For PC2, the top 75% of the features can vary in importance by a factor of 10.6.

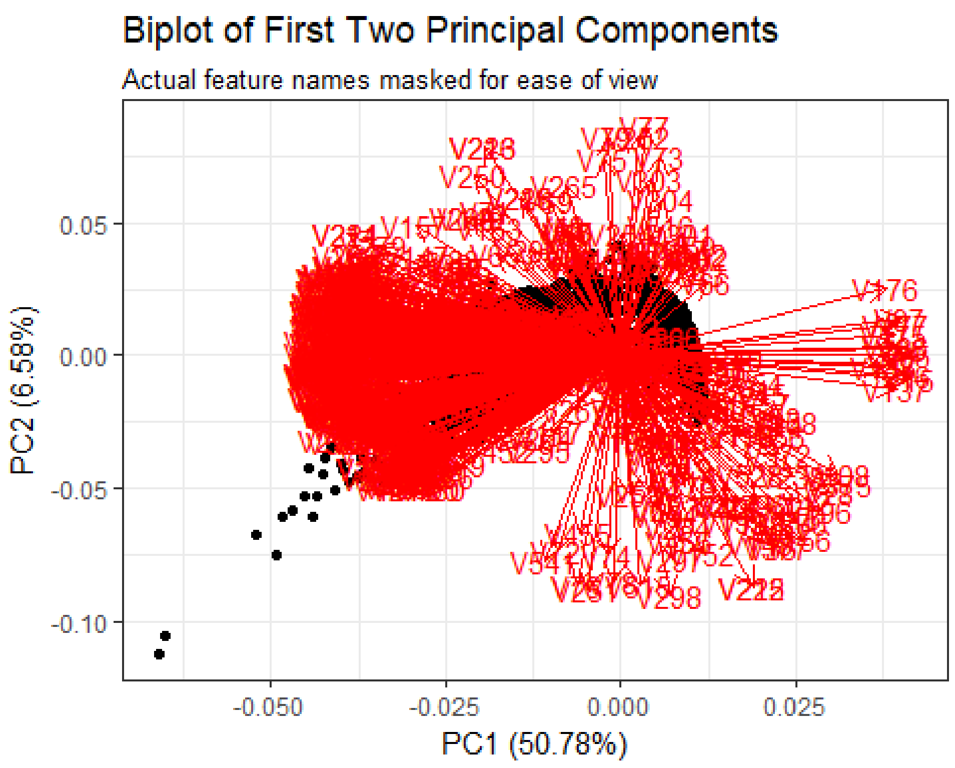
Effectiveness of the Principal Components

Another reason for using principal components is to reduce the dimensionality of our data and visualize observations to their class labels. Here is a biplot of the first two principal components with respect to the activity class labels in our data:

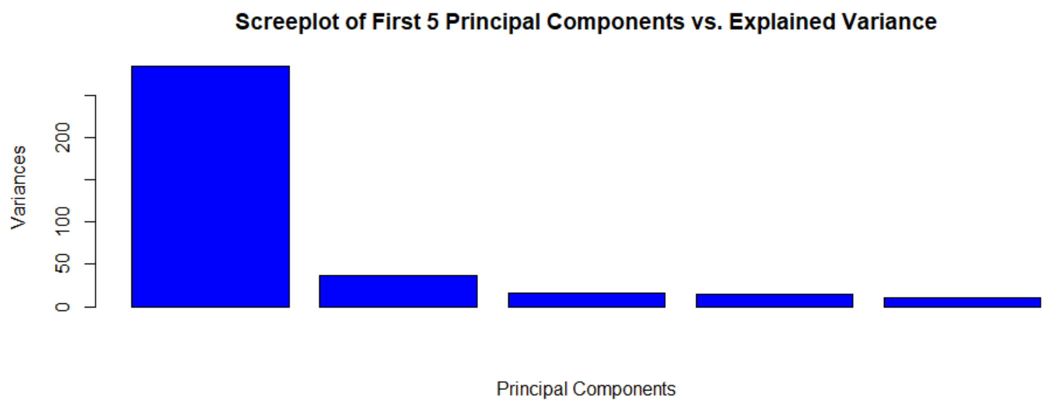


Now we can see a clear distinction between observations associated with certain activities. For instance, points associated with the activity WALKING DOWNSTAIRS are furthest away from points associated with the activity LAYING. We can see two very distinct clusters of observations, one associated with motion-related activities and the other associated with sedentary activities.

Next we can examine a biplot of the first two principal components with loadings displayed. The features most correlated with the principals components are located on the outer perimeter. 50.78% of the variability in the data is explained by the first principal component.

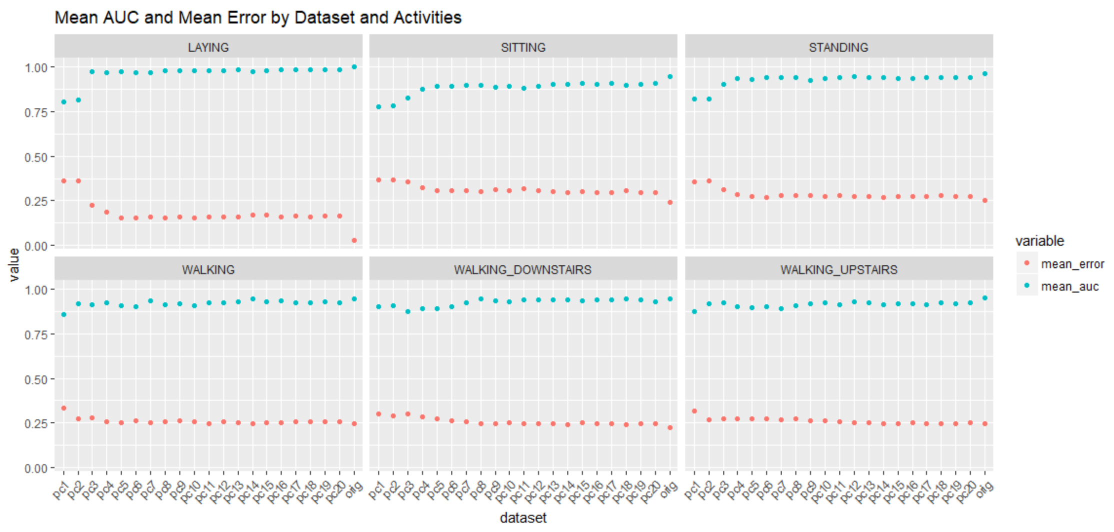


A screeplot of variance explained by the principal components



As expected, the first principal component explains the most amount of variance in the dataset relative to the others. But how effectively would these principal components perform as predictors in a classifier?

An experiment was conducted to measure the effectiveness of the principal components using a binary classifier (i.e. logistic regression model) and evaluating AUC (area under curve). The AUC is the area under the ROC (receiver-operator characteristic curve). This is the probability that a randomly chosen positive observation has a higher predicted value than a randomly chosen negative value. For this experiment, I had to convert my multiclass response variable into binary labels, test for each class in my data, and iterate AUC evaluation over the first 20 principal components.



Here we can see that for every class, the binary classifier performed the best when trained on the original data and not the principal components. This makes sense because data is lost during PCA and it appears to reinforce the integrity of the original data up to this point. The blue dots represent mean AUC. The red dots represent mean error (i.e. root mean squared error). The principal components are effective, but not as effective as using the original training set features. However, the number of principal components that yielded the best model was the first 20 (i.e. the closest performance-wise to the original data).

Supervised Learning

Earlier we saw that a logistics regression model trained on all the features in the original data yielded a mean AUC of 0.96 and a mean error of 0.21. Here we'll use the support vector machine algorithm to predict multi-class labels as well as binary labels.

The preliminary fitted model generated fairly good results. All 561 features were utilized with a 10-fold cross-validation on the training dataset. The cost of constraints was specified at a fairly low value of 5, which meant I could expect a larger margin of separation between class labels at the expense of more errors.

Model Summary

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.0020884600450684
##
## Number of Support Vectors : 2086
##
```

```
## Objective Function Value : -92.2094 -32.6935 -21.5106 -28.6539 -20.0042 -
1331.433 -15.9471 -16.4768 -16.9269 -13.2725 -11.899 -13.9377 -111.0658 -
122.2173 -132.8648
## Training error : 0.002992
## Cross validation error : 0.012378
## Probability model included.
```

Model results show a training error of 0.3% and cross-validation error of 1%. It's A fairly complex model with 15 sets of 2086 support vectors.

Estimate of out-of-sample prediction errors on test data

Flat-error (0-1) rate: 4.48%

Class-specific Flat-error rate:

class	error
LAYING	0%
SITTING	10.59%
STANDING	3.2%
WALKING	2.62%
WALKING_DOWNSTAIRS	8.1%
WALKING_UPSTAIRS	3.4%

Accuracy of prediction on test data

The accuracy of the model was estimated at 95.52%

Confusion Matrix

	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	537	3	0	0	0	0
SITTING	0	439	17	0	0	0
STANDING	0	48	515	0	0	0
WALKING	0	0	0	483	8	14
WALKING_DOWNSTAIRS	0	0	0	9	386	2
WALKING_UPSTAIRS	0	1	0	4	26	455

The model had most difficulty predicting the class label of SITTING. As we saw earlier in the Biplot of First Two Principal Components, the observations of the class SITTING was clustered very closely with the observations of STANDING. This reinforces the large error that we are seeing.

Based on the biplot from earlier, it also appears that the original data was effective enough that a preliminary SVM could easily distinguish between sedentary (e.g. LAYING) activities and motion activities (e.g. WALKING). There were two very distinguishable clusters of observations for these super classes.

Let us now look at how well the SVM algorithm performs when instead we choose to have a binary classifier rather than a multiclass classifier. As we saw in the confusion matrix and class errors, there are quite a few observations that were "hard to classify" given the class-specific label. SVMs are generally described to be binary classifiers. If the business decision is to deploy a model with the least amount of error yet still be effective in classifying human activity to some extent, then perhaps a binary classification model might be more appropriate.

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.00203862928745564
##
## Number of Support Vectors : 265
##
## Objective Function Value : -40.2771
## Training error : 0
## Cross validation error : 0.001088
## Probability model included.
```

This new binary classifier performs exceedingly better than the multiclass classifier from the training error and CV error standpoint at the bottom of the summary. It is also much less complex, requiring fewer support vectors.

Estimate of out-of-sample prediction errors on test data

Flat-error (0-1) rate: 0%

Class-specific Flat-error rate:

class	error
INMOTION	0%
SEDENTARY	0%

Accuracy of prediction on test data

The accuracy of the model was estimated at 100%

Confusion Matrix

	INMOTION	SEDENTARY

INMOTION	1387	0
SEDENTARY	0	1560

Conclusion

To recap, the Human Activity Recognition dataset was a large dataset with a dimensionality of 561 features by 10,299 observations. This data was generated by internal cell-phone sensors and was heavily pre-processed (i.e. feature engineering and normalization) from the data source. Upon inspection by a non-specialist in the subject area such as myself, these were not "human-readable" features in the sense that one could manage understanding what each feature represents, feasibly determine how they relate together, and how they impact the response variable "active daily living activity" in a simple table view. The business question posed at the forefront was whether we can use this data to predict a humans physical activity in order to test dependencies between mobile product utilization and active daily living.

We conducted principle component analysis to identify the top features that accounted for the most variance in the data and assessed the effectiveness of the principal components themselves. The first principal component accounted for over 50% of the variance in the data. By plotting the observations in the context of the first two principal components we saw a clear distinction between motion and sedentary activities, as well as some clear separation of ADL classes. We measured the performance of several logistics regression models trained on principal components and the original data by evaluating AUC. The mean AUC of the logistics regression models trained on the original 561 features came out to 96% and the mean error came out to 21%.

Supervised learning with support vector machines was then applied to the original 561 feature vector to predict the class labels of (LAYING | SITTING | STANDING | WALKING | WALKING_DOWNSTAIRS | WALKING_UPSTAIRS) as well as the binary class labels of (SEDENTARY | INMOTION). The accuracies of the two SVM models were 96% and 100% respectively. We can conclude that we have two models that performed very well in human activity recognition.

One model can predict to a good degree of accuracy, six unique types of human activity and the other is a highly accurate classifier of motion activity. Based on the business need, accuracy and versatility are two trade-off factors that can be taken into account when making a decision to deploy a model to test categorical dependencies between mobile products and human activity. Model deployment for a pilot study on a larger sample of new subjects is the final recommendation of this project.

Final Considerations

As was mentioned earlier, the majority of the pre-processing of sensor data was handled by the data source. For considerations on reproducibility and more information about this dataset please contact: activityrecognition '@' smartlab.ws.

Appendix

References

Watch Experiment on Youtube

PCA Analysis in R

Predict Principal Components on Test Set

PCA Evaluate AUC

Code