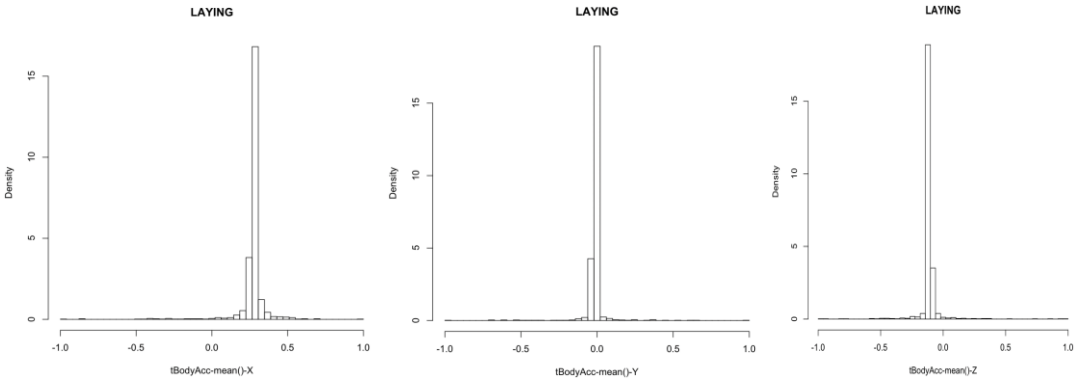
Overview:

For this project I analyzed the Human Activity Recognition dataset, which can be found on Kaggle. After analyzing the data, I trained and tested three classification models and compared their results. I used a logistic regression model, kernel support vector machine, and single-layer feedforward neural network as individual models. An ensemble method might perform better and therefore used in the future.

Analysis:

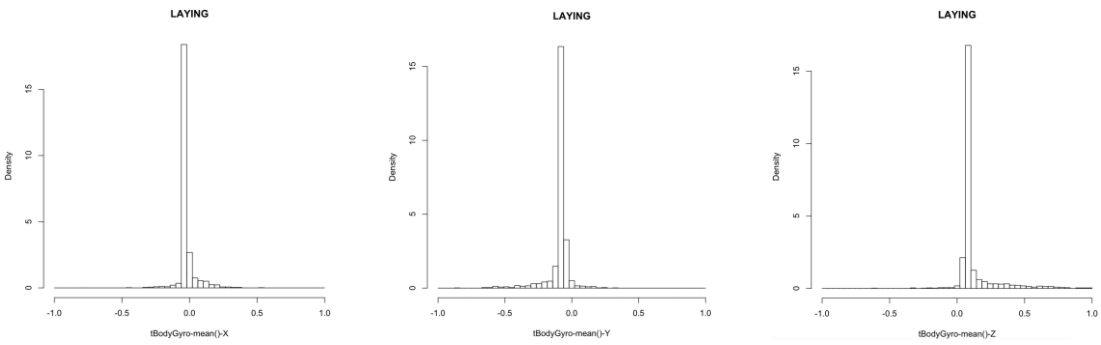
1.

Mean body accelerations in the X, Y, and Z directions for activity "LAYING"



2.

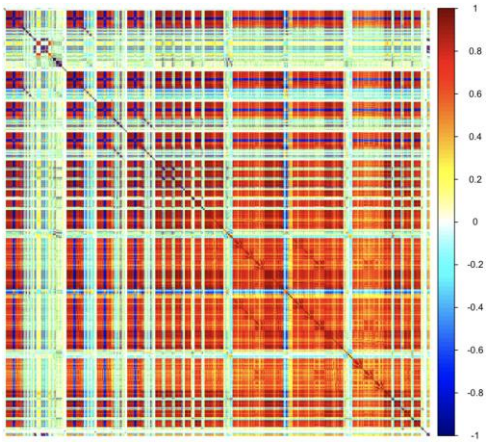
Mean angular velocities in the X, Y, and Z directions for activity "LAYING"



We can clearly see that this data is normally distributed. This is very convenient for model building.

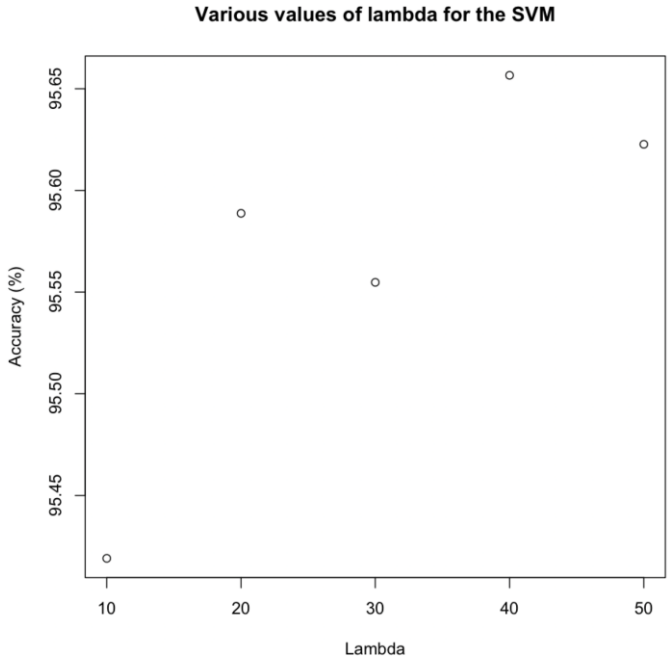
3.

Correlation matrix for the training dataset, displayed as a heatmap



From the correlation matrix we can see that there is quite a bit of correlation, both positive and negative, amongst the data. This makes sense since many of the columns contain data that are functions of the measurements in columns 1, 2, 3, 121, 122, 123.

4.



Note: Accuracy is measured as the mean of accuracy for each value of lambda using k = 10 fold cross-validation.

5.

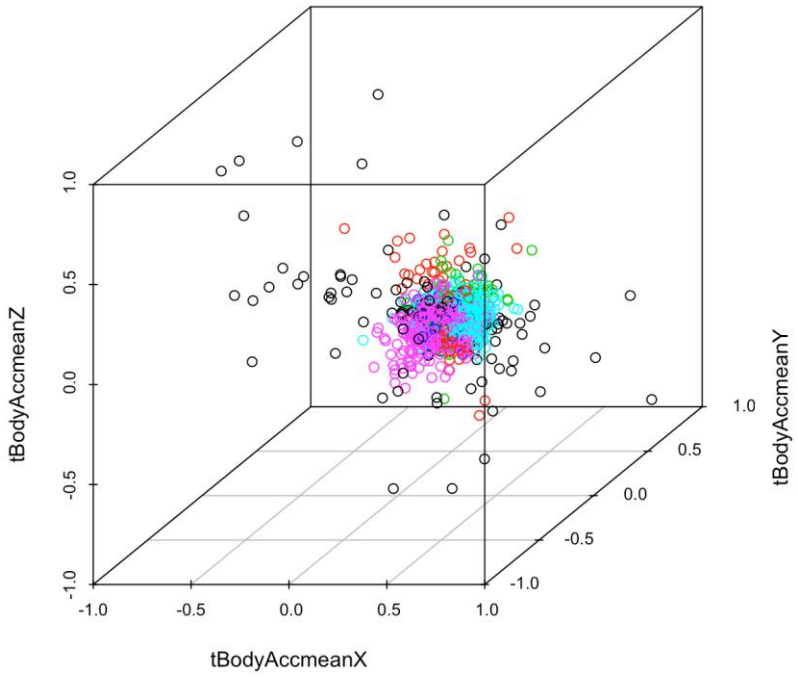
Confusion matrix, training kernel SVM with $\lambda^* = 40$

	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	537	2	0	0	0	0
SITTING	0	442	14	0	0	0
STANDING	0	46	518	0	0	0
WALKING	0	0	0	483	8	15
WALKING_DOWNSTAIRS	0	0	0	8	385	2
WALKING_UPSTAIRS	0	1	0	5	27	454

6. Like the model from lecture, only one misclassification occurs between a static and dynamic activity, this is walking upstairs and sitting. The two common misclassification activities are standing/sitting and walking downstairs/walking upstairs. This seems logical since one would expect these pairs of activities to be similar.

7.

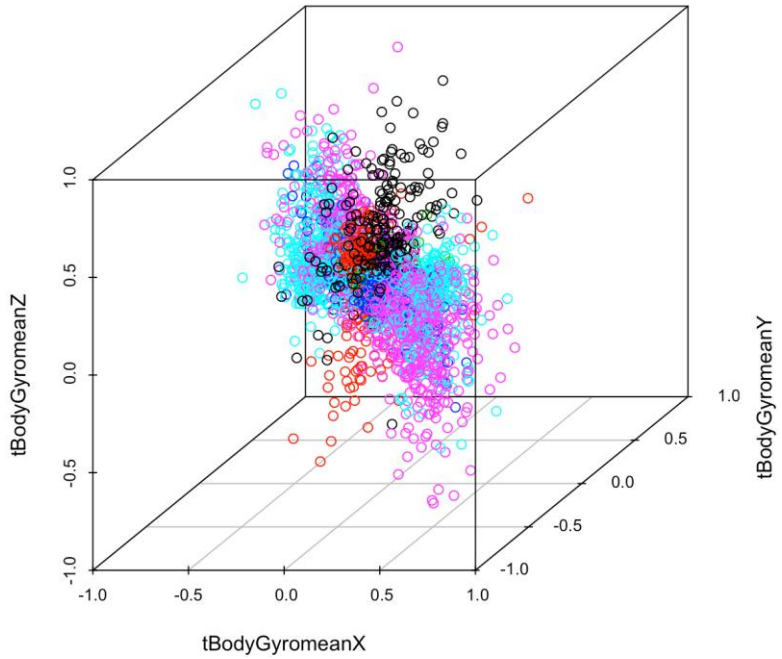
3-D plot of the six different activities using mean acceleration



8. The activity “Laying” is very distinguishable as it has the greatest variability. Our classification matrix above shows that this characteristic makes it easier to classify. The only other distinguishable activity is “Sitting”, notably also a static activity.

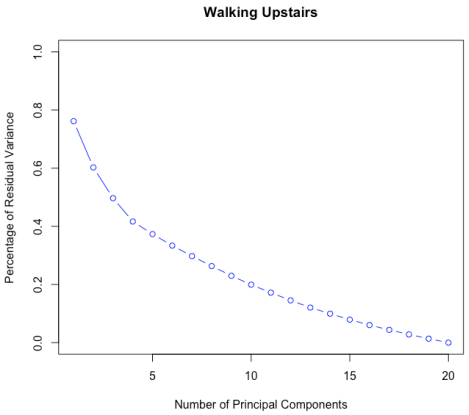
9.

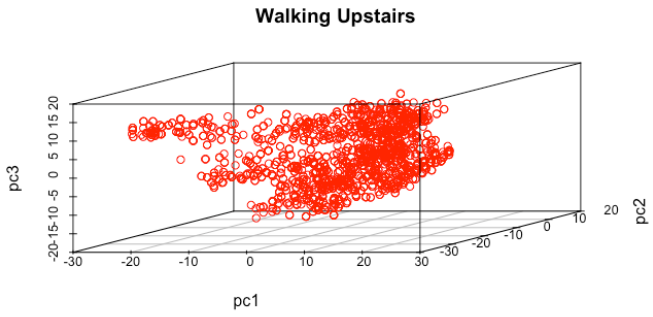
3-D plot of the six different activities using mean angular velocity



10. This plot is more difficult to analyze than the previous one. All activities appear to e centered very close to one another, but have spreads in various directions. Static activities (red and black) appear to lie on a vertical 2-D plane, while the dynamic activities are spread on a diagonal 2-D plane.

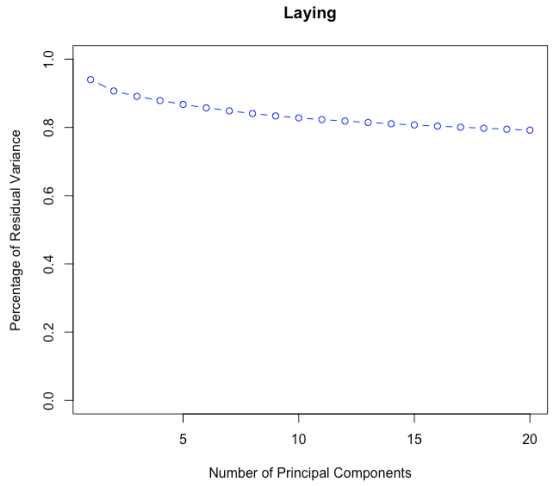
11/12. Kernel PCA with the matrix containing all WALKING UPSTAIRS activities

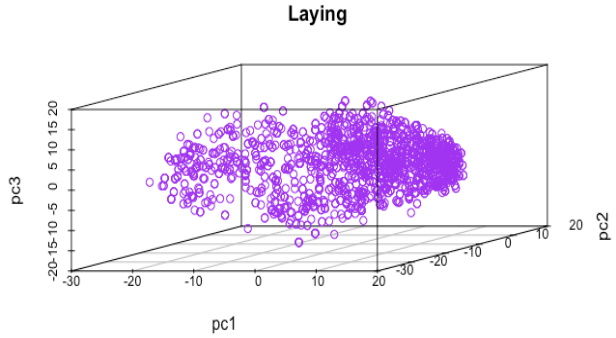




We can see that the percentage of residual variance drops below 10% at around the fifteenth principal component. This seems to be a reasonable result.

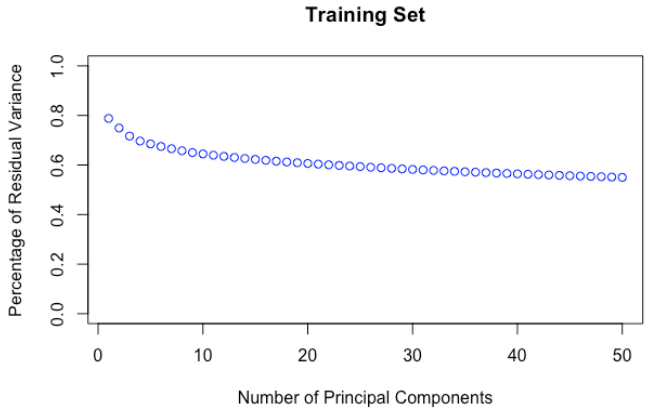
13. Kernel PCA with the matrix containing all LAYING activities





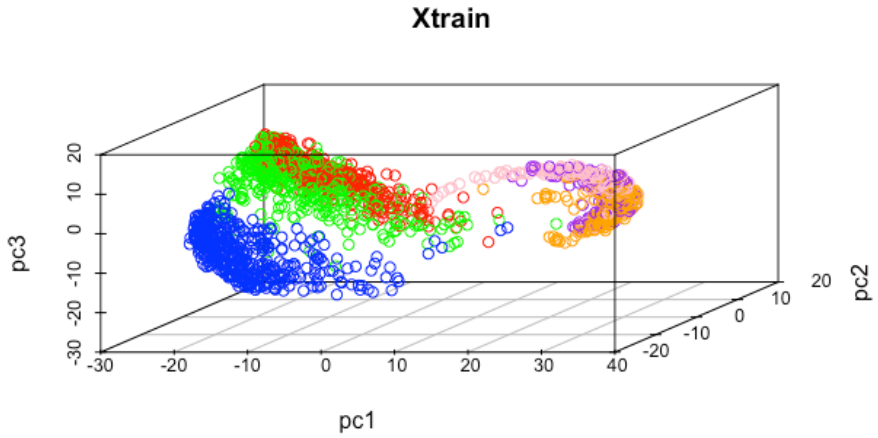
Here the percentage of residual variance does not drop very much even though the same algorithm was run on this matrix as was in the previous section. I am not sure why this happened.

14. Plotting the residual variance as a function of the number of principal components for X.train:



This is the same problem as the LAYING activity matrix, again the same algorithm was run for all three matrices.

16. Displaying the points in the matrix X.train, in the basis formed by the eigenvectors:



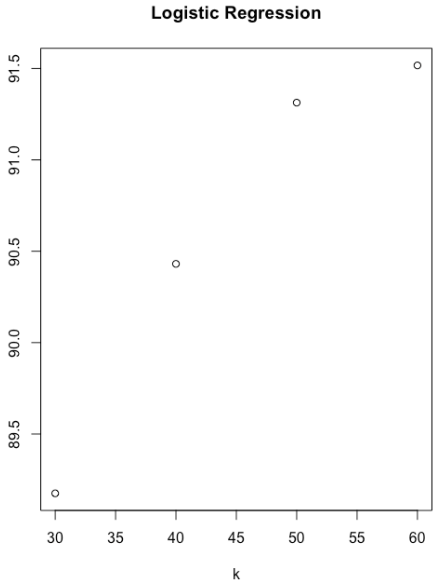
Visually we can still see a separation between static and dynamic activities. One set of these activities is not clearly separable in this space (pink, orange, purple), but the other is when projecting onto only three principal components. Now the blue activity has separated itself from green and red in the third dimension. This is very promising as we have many more principal components to project on, while still reducing the dimensionality significantly.

Classification using the reduced coordinates

17. Logistic regression classifier, the confusion matrix was constructed with k =40

	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	528	0	0	0	0	0
SITTING	9	413	59	0	0	0
STANDING	0	77	473	1	1	0
WALKING	0	0	0	467	16	47
WALKING_DOWNSTAIRS	0	0	0	16	360	4
WALKING_UPSTAIRS	0	1	0	12	43	420

18. Classification accuracy as a function of k

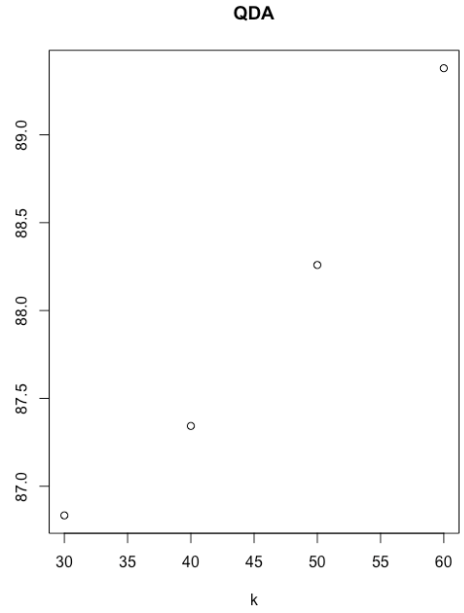


19. QDA classifier

Accuracy = 89.4%, k = 60

	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	520	5	3	0	0	0
SITTING	17	338	24	0	0	0
STANDING	0	145	499	0	0	0
WALKING	0	0	3	462	8	20
WALKING_DOWNSTAIRS	0	0	0	30	376	12
WALKING_UPSTAIRS	0	3	3	4	36	439

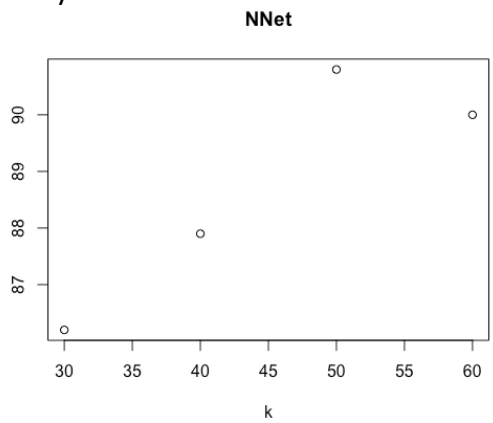
20. Plotting the classification accuracy as a function of k:



21. Feedforward neural net, one hidden layer, as the classifier

	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	530	1	0	0	0	0
SITTING	6	409	69	0	0	0
STANDING	1	80	463	0	1	0
WALKING	0	0	0	473	11	31
WALKING_DOWNSTAIRS	0	0	0	6	370	9
WALKING_UPSTAIRS	0	1	0	17	38	431

22. Plot the classification accuracy as a function of k



Note: When running the nnet for 60 principal components at 10 nodes, lambda = 0.01, I received an error message about having too many weights. I reduced the number of nodes for this pc and therefore might have caused the accuracy to decrease.

Summary:

Logistic regression provided the highest accuracy at the lowest number of principal components. However, if analyzing larger datasets, I would seriously consider the computing time for each algorithm, as the difference could be substantial.