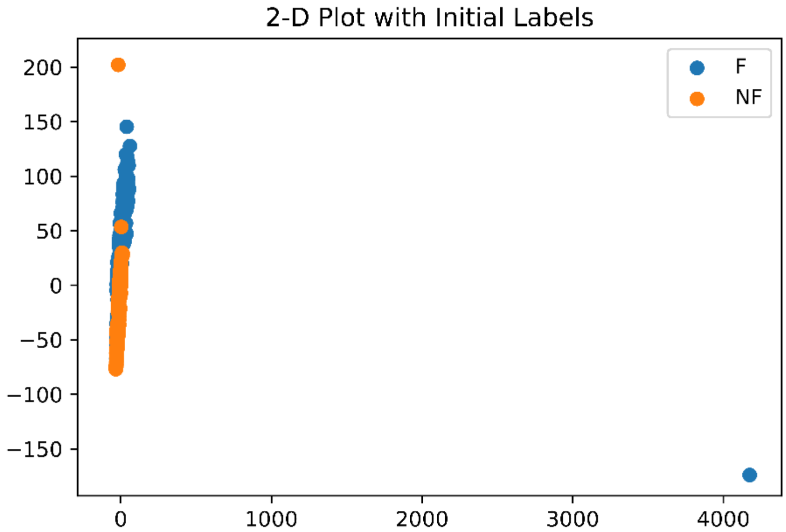
GE 461: Introduction to Data Science
Homework Telehealth - Fall Detection

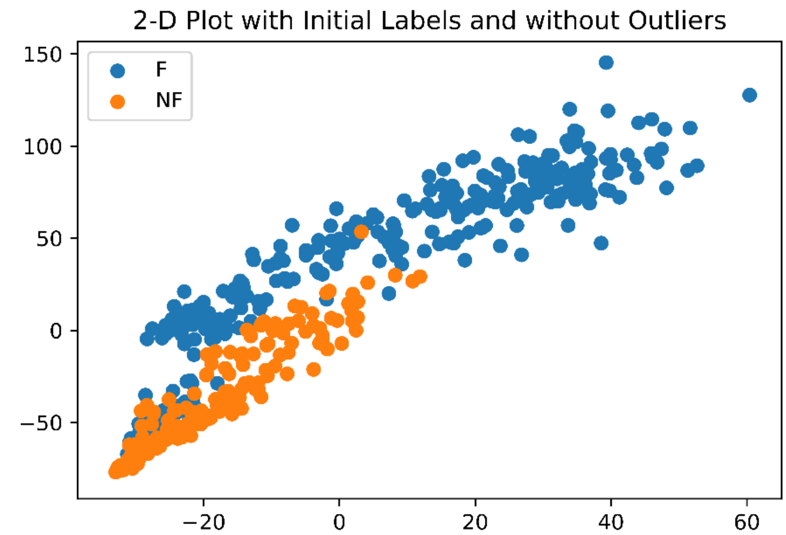
Batıhan Akça - 21502824

Part A:

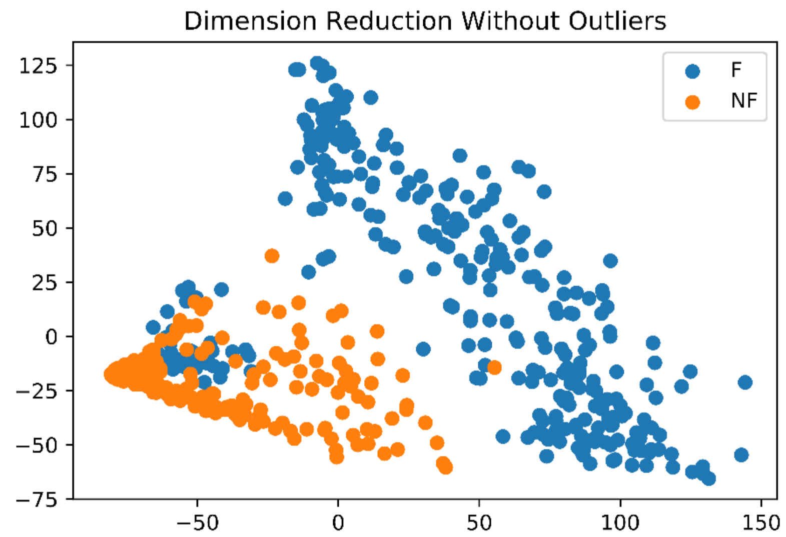


Data shows 2 outliers where the one has the maximum value along the first principal component (x-axis) and the other one has the maximum value along the second principal component (y-axis). Except those data shows a reasonable separability, but those outliers may cause a problem while performing the K-means algorithm.

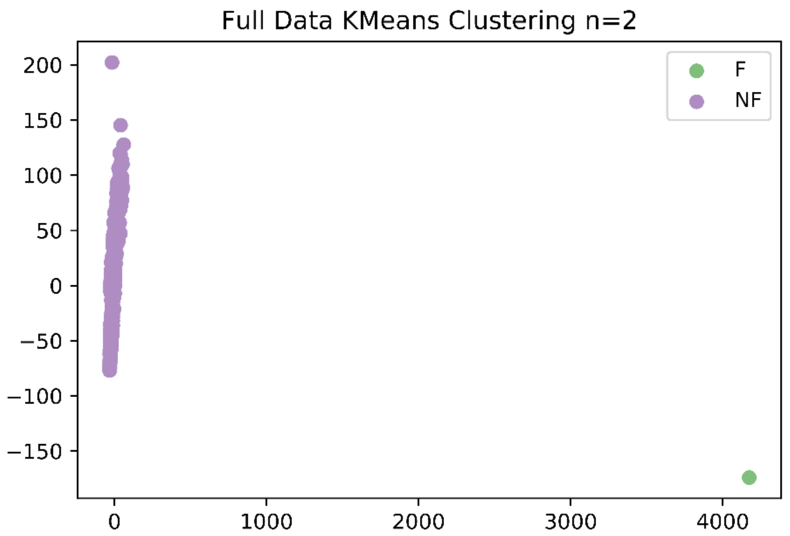
The first principal component explains **75.3072%** of the variance and the second principal component explains **8.51159%** of the variance.



This plot shows how successful the PCA is when we zoom into the left part of the first plot whose representation is deformed by the two outliers.

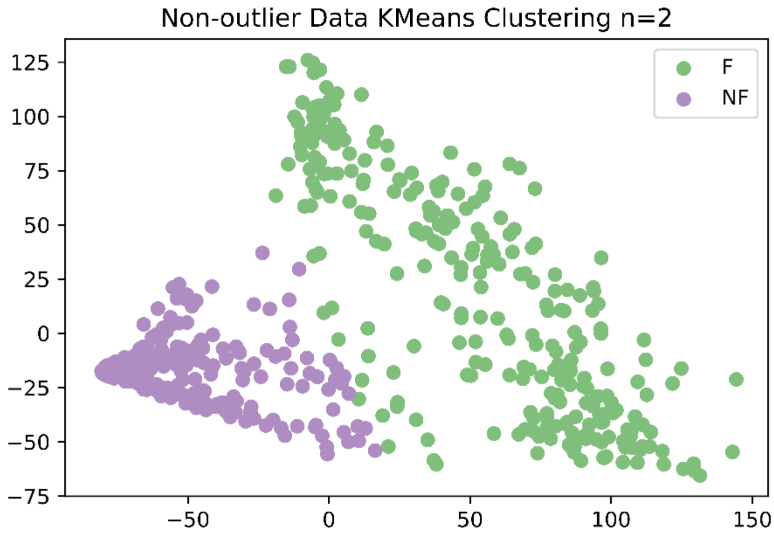


This plot shows what would happen if PCA was performed on the data without outliers (564 samples instead of 566). By just looking at it promises a better picture of unsupervised learning because it ended up with better results in the absence of outliers.

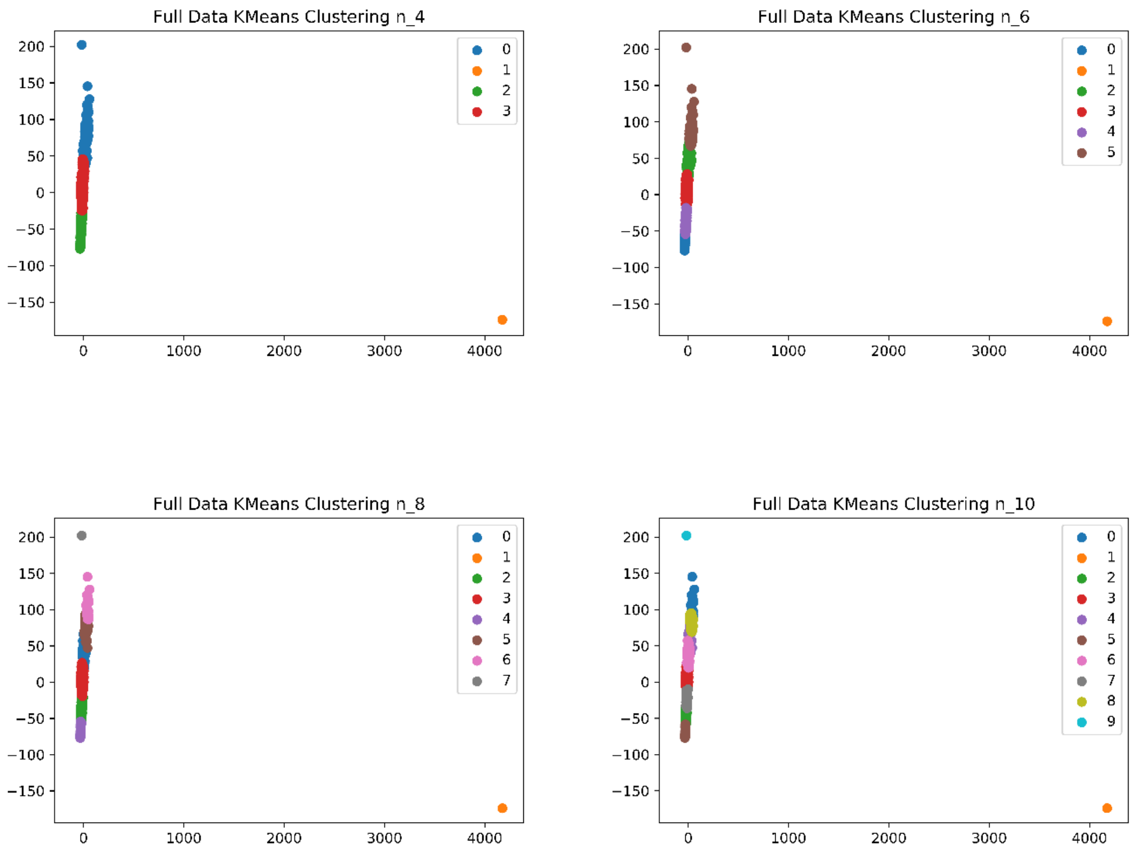


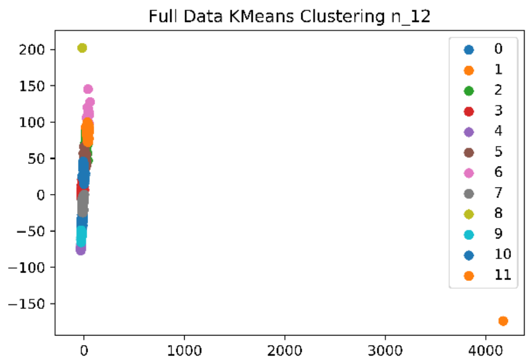
As expected, k means is unsuccessful when there are some outliers with enormously large values compared to the overall. Green cluster is expected to be the "F" so it labeled by hand, but the results are not good in the opposite way also.

Accuracy results are when Green Cluster = F, Purple Cluster = NF is **44.87%** and **55.12%** otherwise.



If K-means is performed on non-outlier dimension reduced data, it gives much better results than the previous one. Accuracy is **88.47%** which almost doubles the first accuracy results.





For several K-means iterations with an increasing k, except right bottom point, results are reasonable if there is a search for some other sub-clusters, however, for our case since there are only F and NFs increasing k does not give any insights about the data.

In overall, by exploration, it can be said that fall detection is possible on the given data because the unsupervised learning gives enough insights about the data.

Part B:

SVM RESULTS

HyperParameters : C = 1.0, degree = 5, max_iter = 10, shrinking = False

SVM Validation Accuracy: 96.47058823529412%

SVM Test Accuracy: 92.94117647058823%

HyperParameters : C = 1.0, degree = 5, max_iter = 1000, shrinking = False

SVM Validation Accuracy: 98.82352941176471%

SVM Test Accuracy: 97.6470588235294%

HyperParameters : C = 1.0, degree = 3, max_iter = 1000, shrinking = False

SVM Validation Accuracy: 98.82352941176471%

SVM Test Accuracy: 97.6470588235294%

HyperParameters : C = 1.0, degree = 3, max_iter = 1000, shrinking = True

SVM Validation Accuracy: 98.82352941176471%

SVM Test Accuracy: 97.6470588235294%

HyperParameters : C = 1.0, degree = 3, max_iter = 10000, shrinking = True

SVM Validation Accuracy: 98.82352941176471%

SVM Test Accuracy: 97.6470588235294%

HyperParameters : C = 1.0, degree = 6, max_iter = 10000, shrinking = True

SVM Validation Accuracy: 98.82352941176471%

SVM Test Accuracy: 97.6470588235294%

HyperParameters : C = 1.0, degree = 12, max_iter = 10000, shrinking = True

SVM Validation Accuracy: 98.82352941176471%

SVM Test Accuracy: 97.6470588235294%

HyperParameters : C = 1.0, degree = 12, max_iter = 10000, shrinking = True

SVM Validation Accuracy: 96.47058823529412%

SVM Test Accuracy: 92.94117647058823%

HyperParameters : C = 1.0, degree = 12, max_iter = 10000, shrinking = True, kernel = 'linear'

SVM Validation Accuracy: 92.94117647058823%

SVM Test Accuracy: 94.11764705882352%

HyperParameters : C = 1.0, degree = 3, max_iter = 10000, shrinking = True, kernel = 'linear'

SVM Validation Accuracy: 98.82352941176471%

SVM Test Accuracy: 98.82352941176471%

MLP RESULTS

Hyperparameters: solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(10, 2), random_state=1, learning_rate_init = 0.0001, max_iter = 200

MLP Validation Accuracy: 68.23529411764706%

MLP Test Accuracy: 68.23529411764706%

Hyperparameters: solver='adam', alpha=1e-5, hidden_layer_sizes=(2, 2), random_state=1, learning_rate_init = 0.0001, max_iter = 200

MLP Validation Accuracy: 87.05882352941177%

MLP Test Accuracy: 87.05882352941177%

Hyperparameters: solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1, learning_rate_init = 0.0001, max_iter = 200

MLP Validation Accuracy: 87.05882352941177%

MLP Test Accuracy: 100.0%

Hyperparameters: solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1, learning_rate_init = 0.001, max_iter = 200

MLP Validation Accuracy: 98.82352941176471%

MLP Test Accuracy: 100.0%

Hyperparameters: solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1, learning_rate_init = 0.1, max_iter = 200

MLP Validation Accuracy: 100.0%

MLP Test Accuracy: 100.0%

In the end, both of the algorithms give very good results, and while MLP could give 100 % accuracy. MLP results changed a lot with very sharp changes in its hyperparameters whereas SVM results did not change that much with parameter changes. It shows that one should be more careful while using MLP to find the optimal configuration for the model.

For both of the algorithms, optimal results are reasonable so it can be said that fall detection based on wearable sensors is successful.