GE 461 PROJECT 4 REPORT

Doruk Barkın Durak

21802255

Industrial Engineering

09.05.2022

Part A:

Before starting the answers, I want to share the hard points I encountered while I was doing the project. First one was to splitting the data into 2 parts, I tried train_test_split methodology but the logic was inappropriate for this part since I couldn't get the whole of 566 samples so I went with dividing the data with. values methodology. I trained the data with PCA with 2 components because there were too many dimensions in data but with pca method I reduced it down to 2. Variance calculations and data charts can be seen as following:

```
Variance of PC 1 is 0.7530724808534603
Variance of PC 2 is 0.08511590067037623
Variance captured by top 2 PCs: 0.8381883815238366
```

Figure 1: Captured Variances

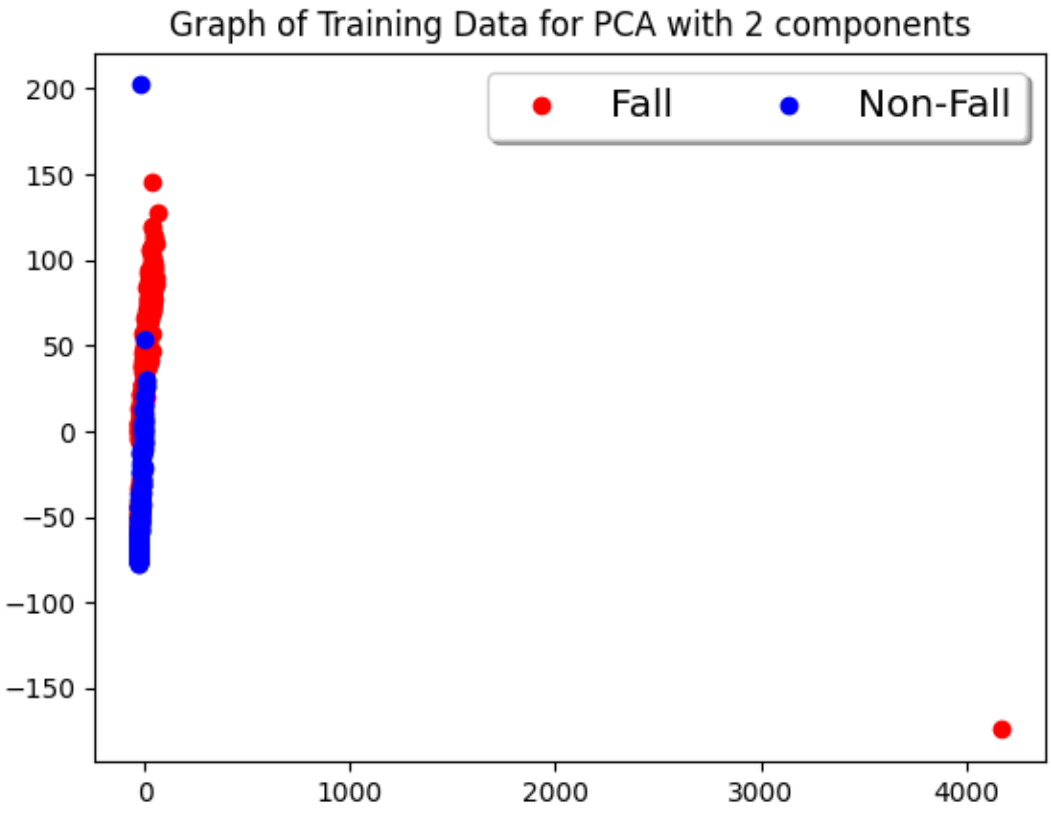

Figure 2 : Graph of the Dataset for PCA with 2 components

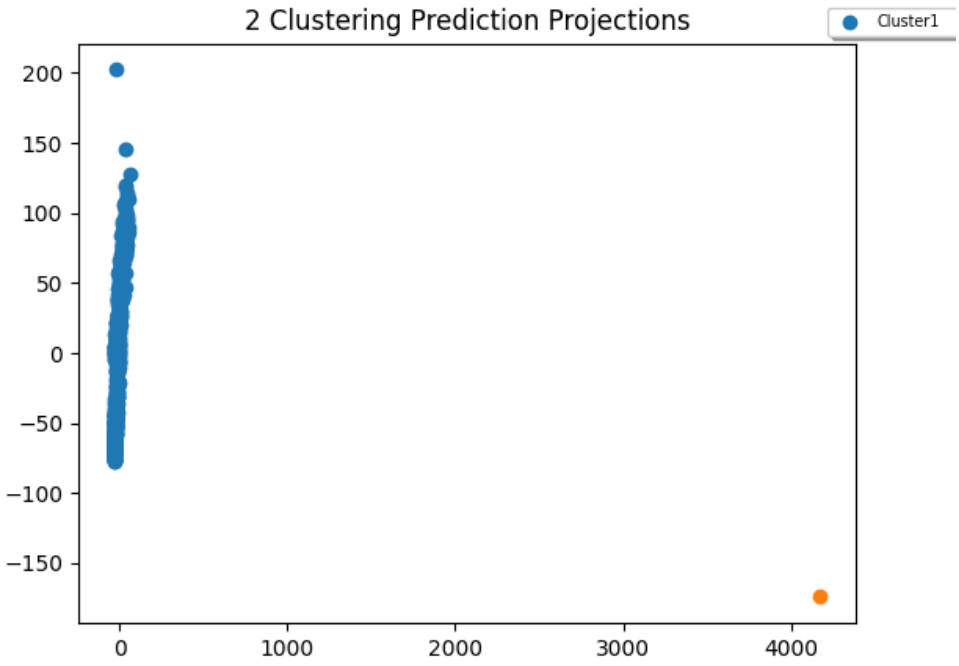

Figure 3 : 2-means Clustering

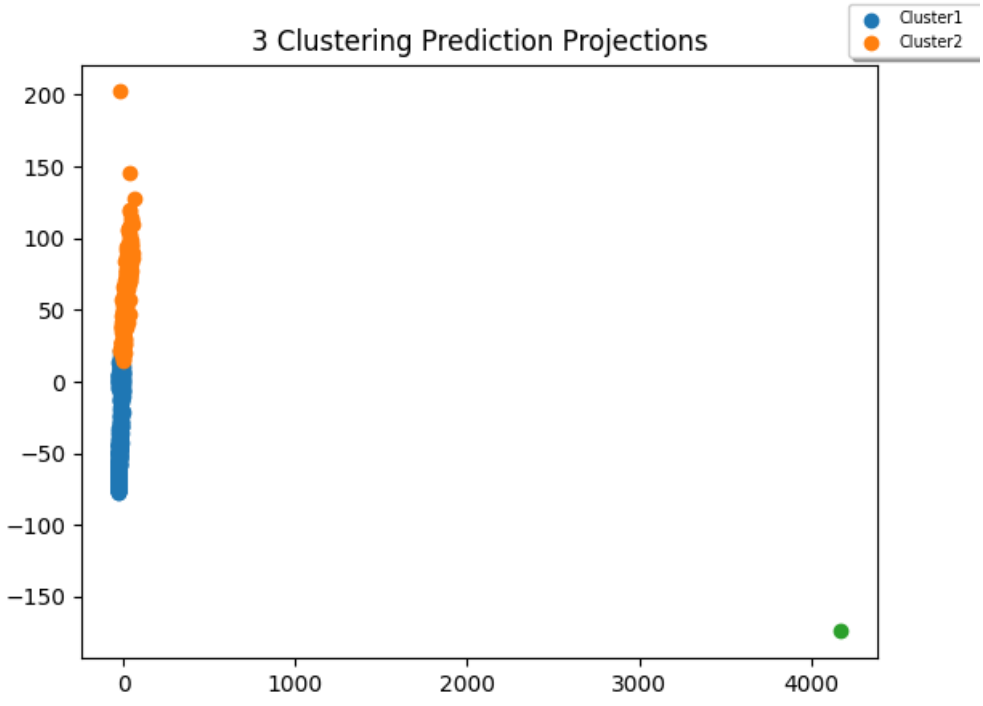

Figure 4: 3-means Clustering

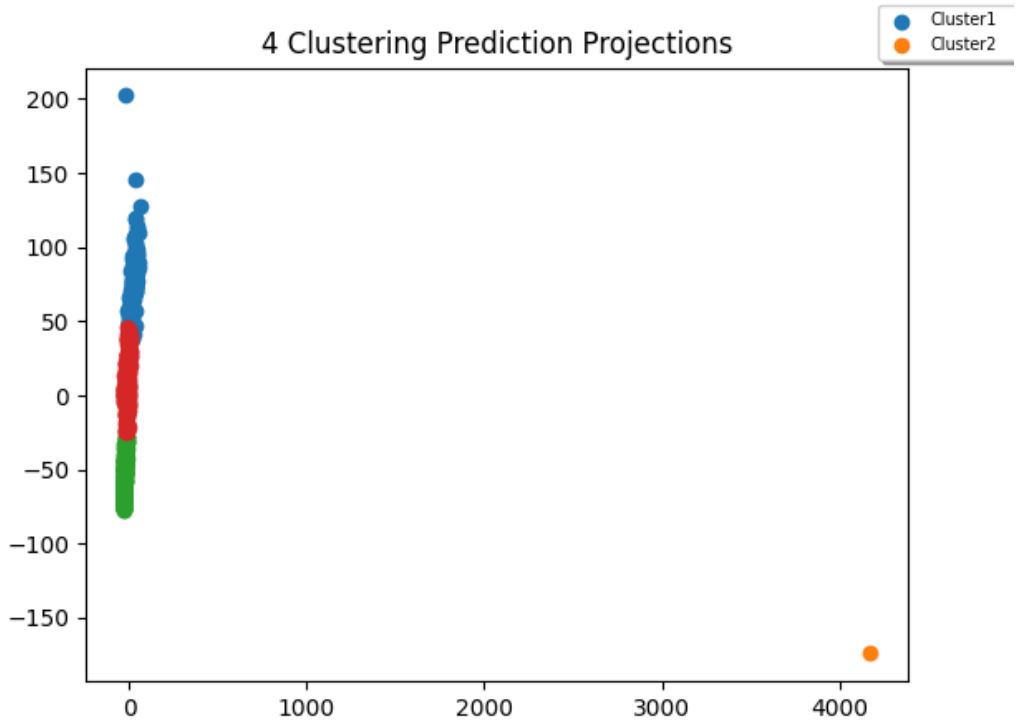

Figure 5 : 4-means Clustering

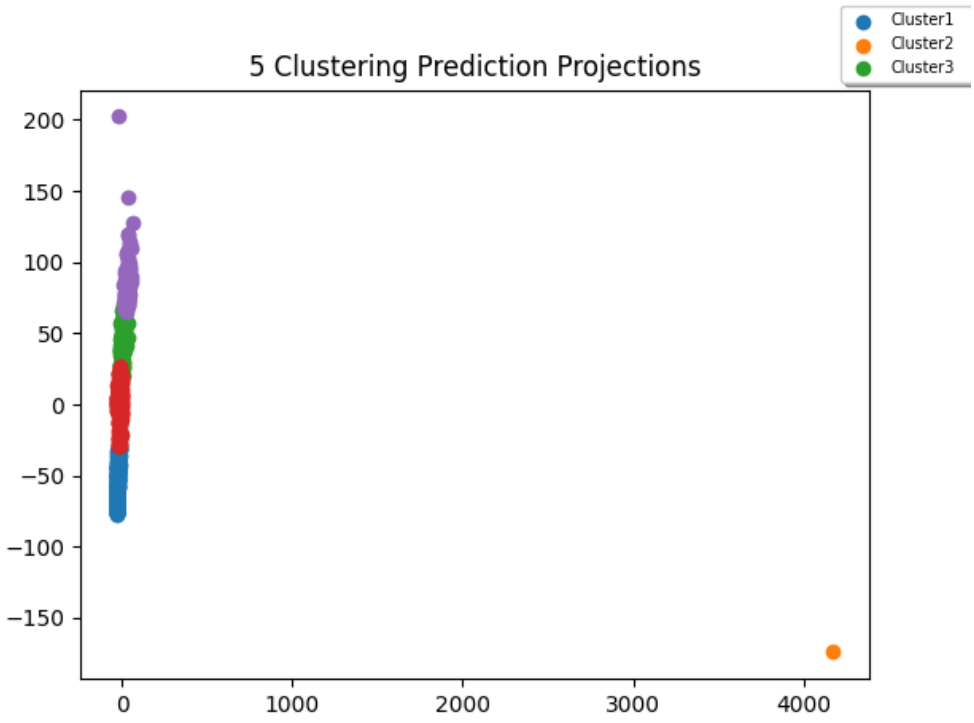

Figure 6: 5-means Clustering

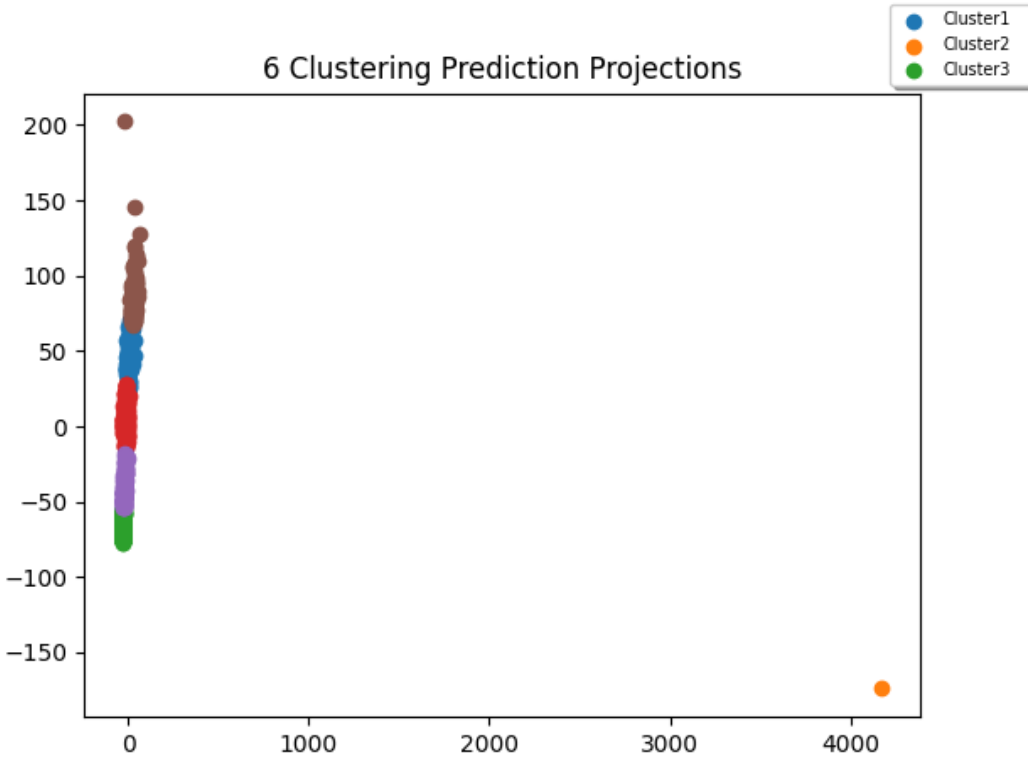

Figure 7: 6-means Clustering

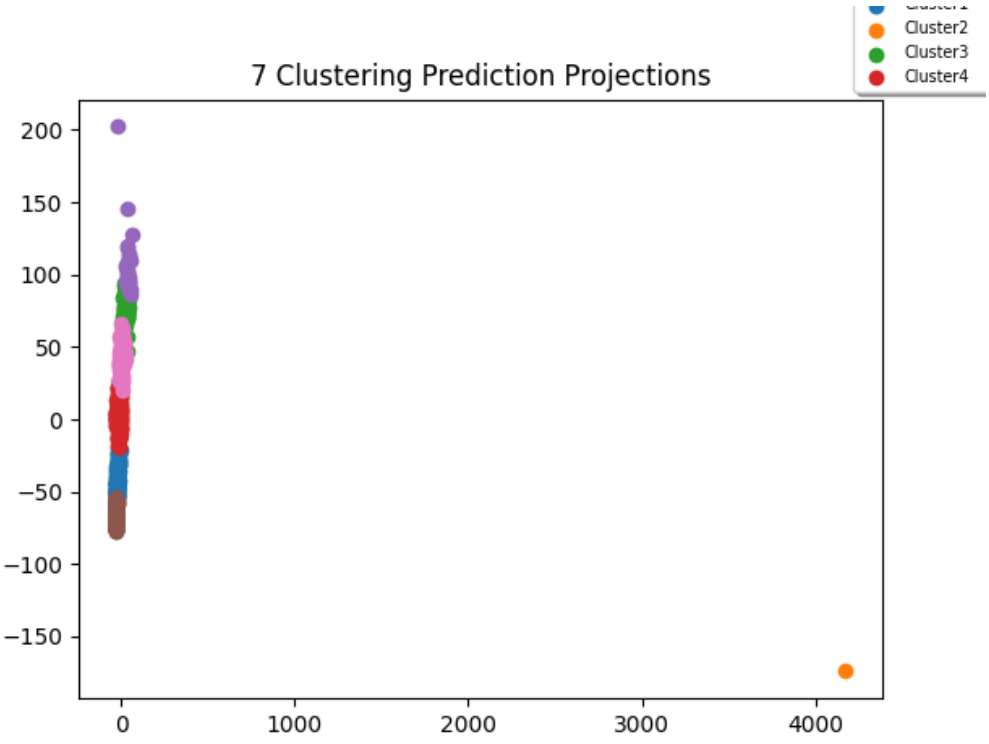

Figure 8: 7-means Clustering

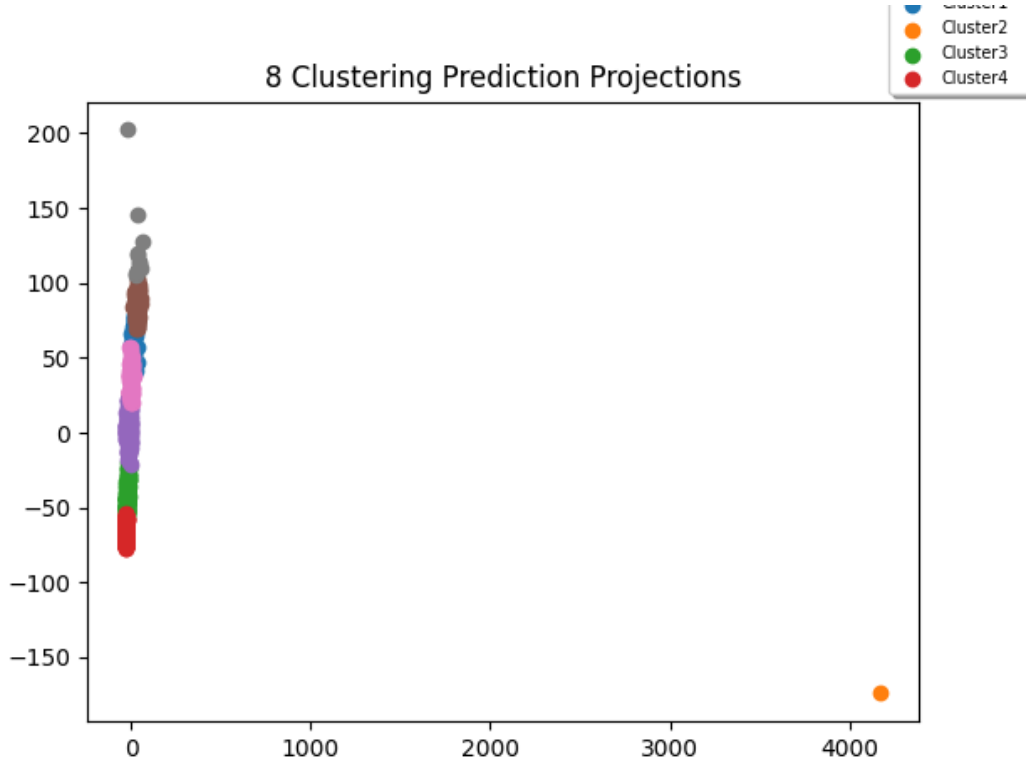

Figure 9: 8-means Clustering

As seen on the graphs the outlier points are problematic because in all distributions, they take up a cluster, resulting in inaccuracies as the clustering number increases. Silhouette scores for each cluster calculated as:

```
For number of clusters = 2 the silhouette score is 0.9806934190019767
For number of clusters = 3 the silhouette score is 0.6447760072898894
For number of clusters = 4 the silhouette score is 0.645276282219734
For number of clusters = 5 the silhouette score is 0.5963564068097033
For number of clusters = 6 the silhouette score is 0.5415928314735022
For number of clusters = 7 the silhouette score is 0.5289286597365986
For number of clusters = 8 the silhouette score is 0.507320654755809
The optimal number of clusters is 2
```

Figure 10 : Silhouette scores and optimal number of clusters of data for n clusters

I found the optimal number of clusters is 2. But looking at the graphs, a healthy analysis is not possible so I will not make an assumption about optimal number of clusters. I tried with putting 2 on the calculation as the question asked and the consistency of data was around %55.12 which was low because of 1 outlier point was taking a whole cluster, resulting in consistency decrease. Also because of the outliers, exact distribution of data is barely visible and after 3 clusters, data becomes messy and unhealthy for an analysis.

So, I removed 2 outlier points in the data for a better calculation. Graphs without outlier points and silhouette scores can be found below:

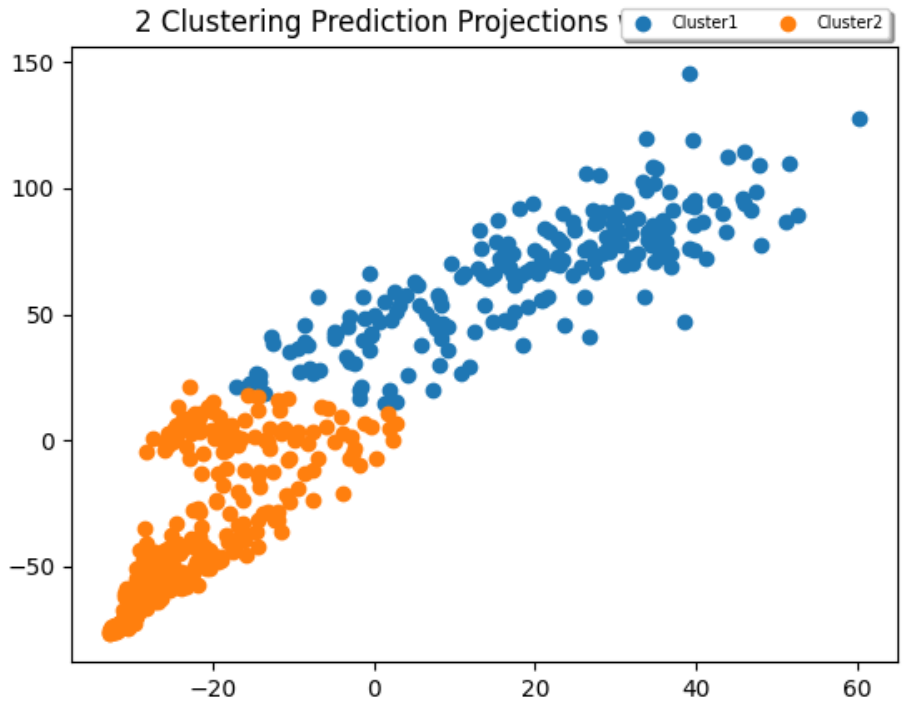

Figure 11: 2-means Clustering without outliers

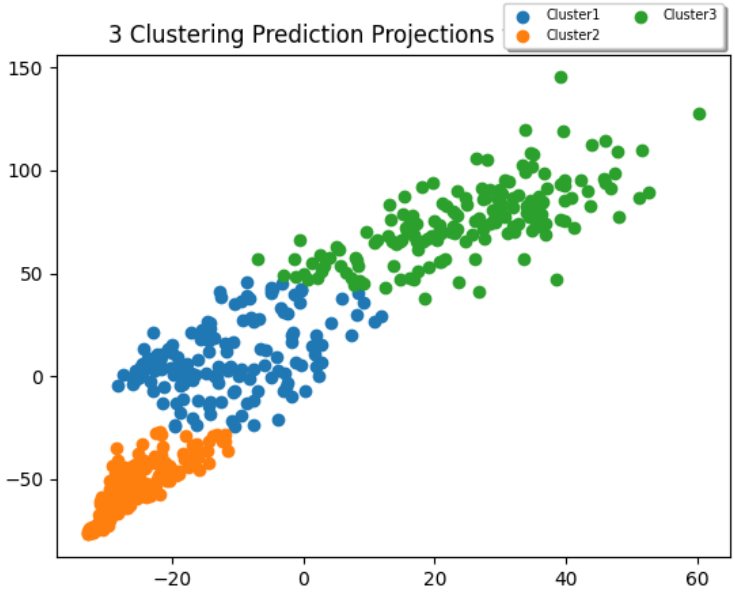

Figure 12: 3-means Clustering without outliers

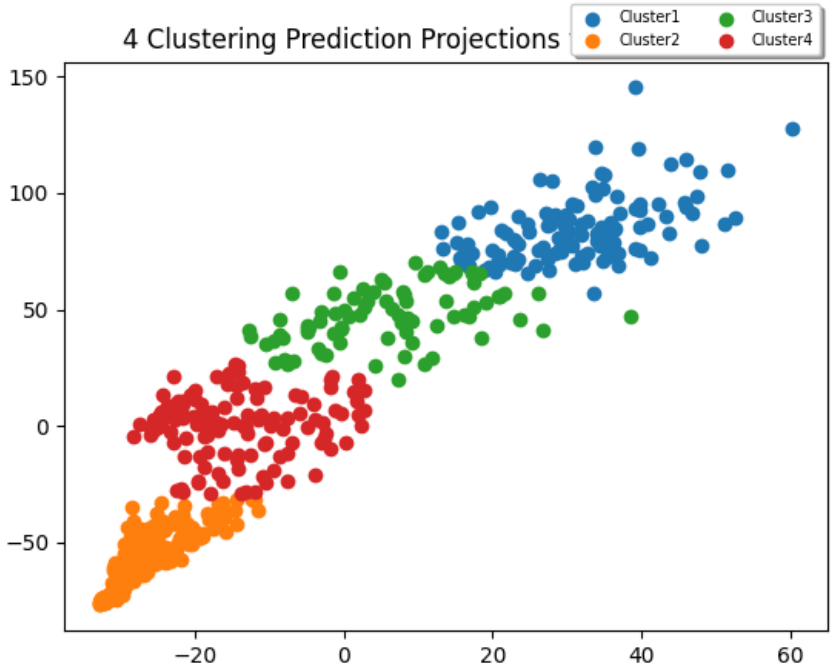

Figure 13: 4-means Clustering without outliers

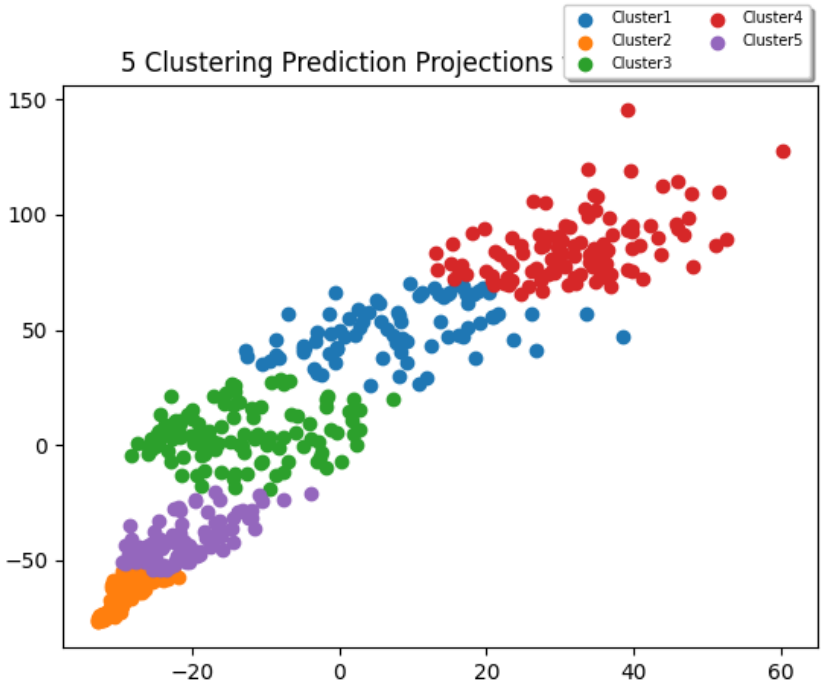

Figure 14: 5-means Clustering without outliers

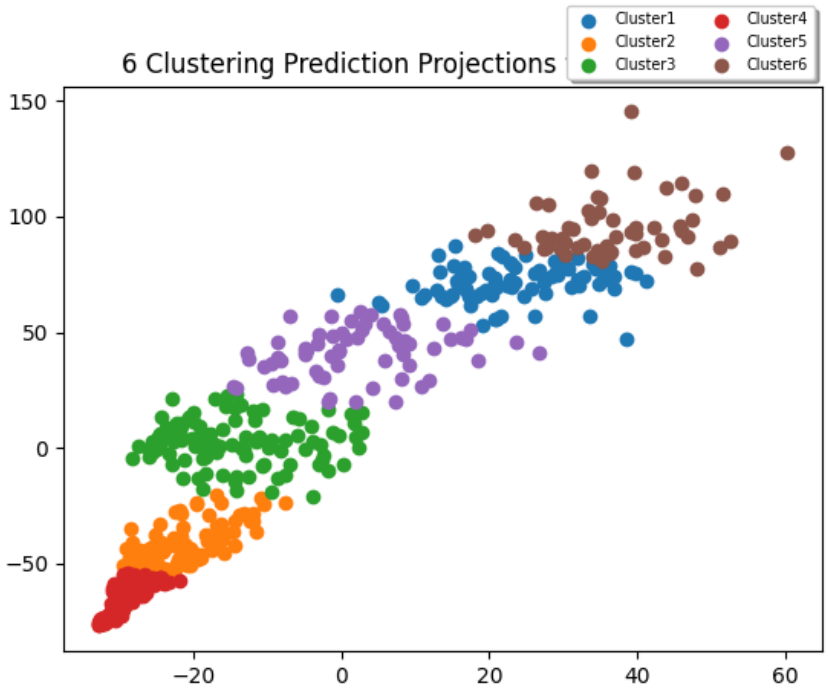

Figure 15: 6-means Clustering without outliers

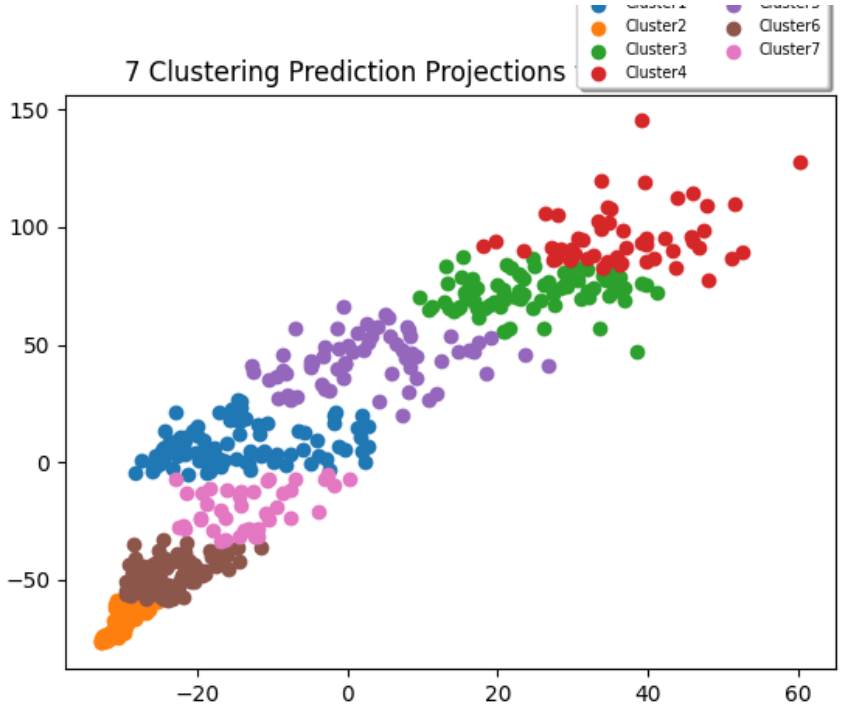

Figure 16: 7-means Clustering without outliers

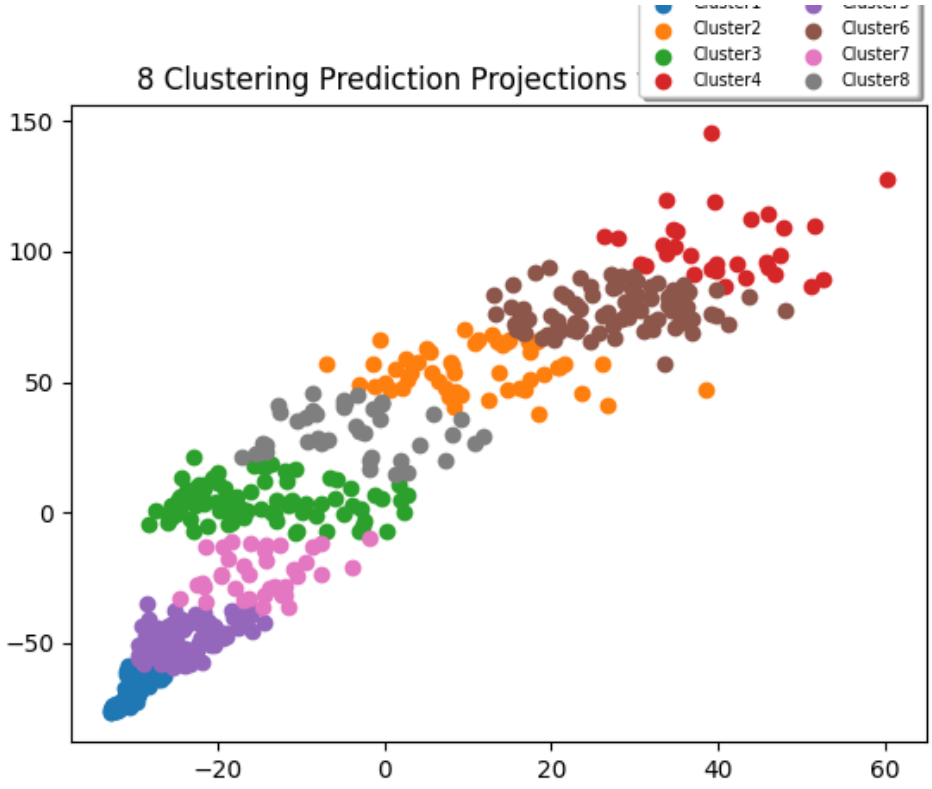

Figure 17: 8-means Clustering without outliers

```
For number of clusters = 2 the silhouette score is 0.6466810783913387
For number of clusters = 3 the silhouette score is 0.6490891649514473
For number of clusters = 4 the silhouette score is 0.6015002855078231
For number of clusters = 5 the silhouette score is 0.5465076216539775
For number of clusters = 6 the silhouette score is 0.5158800623150082
For number of clusters = 7 the silhouette score is 0.48769780327476053
For number of clusters = 8 the silhouette score is 0.4810287307420612
The optimal number of clusters is 2
```

Figure 18: Silhouette scores and optimal number of clusters of data for n clusters without outliers

As seen the silhouette scores are more closely distributed now and optimal number of clusters remained same. Calculating with 2 clusters, consistency of data is now around %78 which is a significant increase and after removing 2 outlier points it is safe to say that based on the silhouette scores, optimal number of clusters is equal to 2. From this data and charts, I can say that for given data set the fall detection is possible.

Part B:

This is the supervised learning stage, at first, I assumed that these 2 methods will perform better than KMean clustering which was an unsupervised learning algorithm. I was correct about my assumptions because these 2 methods gave me %100 validation accuracies on multiple occations.

For SVM PART:

All the data from SVM calculations can be found on SVM_results.csv. I used 4 types of parameters for this calculation, they were c_val for regularization parameter, kernel for kernel functions, degree for polynomial type function and g_val for coefficients for Radial Basis Function, Polynomial Function, Sigmoid Function. From the input the given optimal paramaters were as follows: c_val=0.1, kernel=polynomial, gamma = auto and degree = 3 gave me %100 validation accuracy but because of being extra safe from squared l2 penalty I decided that the optimal parameters will be equal to: c_val=0.001, kernel = polynomial, gamma =auto and degree=4.

For MLP PART:

All the data obtained from MLP calculations can be found on MLP_results.csv .For MLP part I had 34 different parameter combinations all resulted in %100 accuracy my top 1 solution parameters were as follows:

Layer size= (32, 32)

Activation=tanh

Solver=adam

learning rate=0.0001

alpha=0.0001

Which gave me %100 accuracy for this solution. Since my alpha value is quite low, I am not concerned about l2 penalty.

Comparison of SVM and MLP:

Both methods are applicable because both of them managed to get a %100 accuracy on given data. In general accuracy wise SVM method found 13 %100 from 393 tries while MLP method found 34 %100 out of 601 tries. MLP's rate is higher compared to the SVM so I would say that MLP better in this aspect but because of the parameters of MLP, it takes more time to train the

model than SVM method. In this case if we have a time constraint, I would say SVM is the better solution because of its performance.

References:

1) https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

2) https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

3) https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

4) https://en.wikipedia.org/wiki/Multilayer_perceptron

5) Python sci-kit learn library

6) Python numpy library

7) Python pandas library

8) Python matplot library

9) https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a

10) https://hackernoon.com/introduction-to-machine-learning-algorithms-logistic-regression-cbdd82d81a36