GE-461 Fall Detection Project Report

Muhammet Musa Gezer

21703973

Principal Component Analysis (PCA):

In my investigation of fall detection using sensor data, I first performed a Principal Component Analysis (PCA) to reduce the dimensionality of the dataset. The objective of this step was to simplify the multi-dimensional data into a more manageable form by transforming it into a new set of variables (principal components) while retaining as much information as possible.

The PCA transformation with two components resulted in an explained variance ratio of [0.23094147 0.17637566]. This implies that the first component accounts for approximately 23.1% of the total variance in the data, and the second component covers about 17.6%. In total, these two components explain roughly 40.7% of the variance in the dataset.

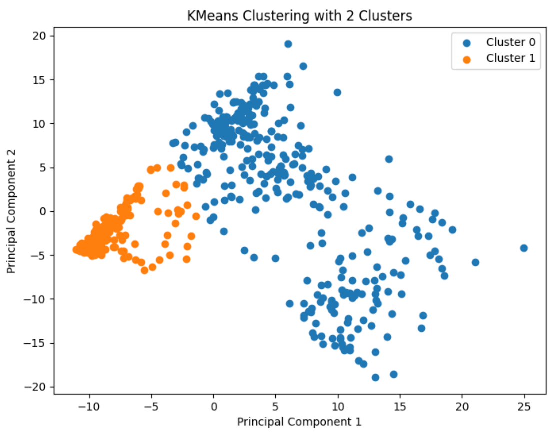
Although a combined explained variance ratio of 40.7% might initially appear low, it's important to understand that a two-component PCA represents a significant reduction from the original high-dimensional feature space. In simpler terms, these two components can be considered as new axes in the feature space that best capture the variation in the data.

```
Explained variance ratio: [0.23094147 0.17637566]
Total explained variance ratio: 0.40731713235759026
```

Clustering Analysis:

After reducing the data using PCA, I proceeded with clustering analysis. The goal of clustering was to identify underlying patterns and groupings within the dataset based on the reduced features. I implemented K-means clustering for this purpose, starting with two clusters and gradually increasing the number of clusters up to eight.

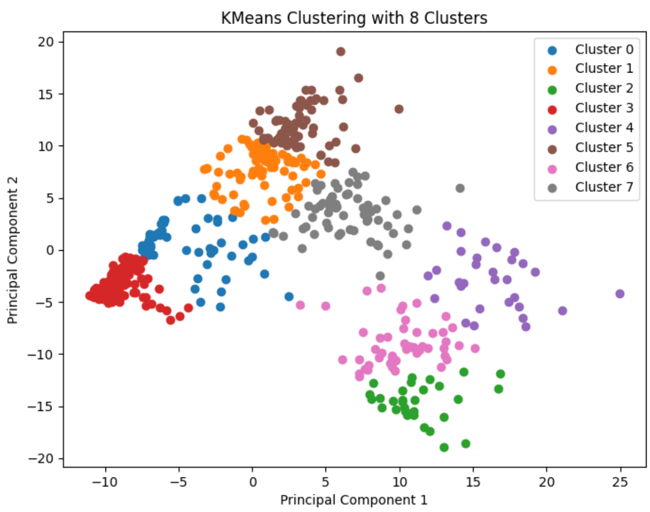
The results from K-means with two clusters (K=2) showed a certain degree of overlap between clusters representing fall and no-fall events, indicating some ambiguity in clear separation between these two classes. The confusion matrix and the plot of distriubtution for this case was as follows:



F	46	266
NF	194	59

This matrix gives the accuracy of 82%. That means we can use the first 2 principal component for clustering.

When I increased the number of clusters to eight (K=8), the confusion matrix showed a different pattern of data distribution:



These results suggest that while the PCA-reduced features maintain a fair amount of the data's structure and enable some degree of separation between fall and no-fall events, the patterns underlying these events might exist in a more complex or higher-dimensional space.

In conclusion, both PCA and K-means clustering provided valuable insights into the structure of the sensor data and its relation to fall and no-fall events. However, due to overlapping regions and misclassifications, these techniques might not be sufficient for highly accurate fall detection. My further steps involved experimenting with supervised learning models, such as Support Vector Machines (SVM) and Multi-Layer Perceptron (MLP) classifiers, to improve fall detection accuracy.

Report on Fall Detection Using SVM and MLP Models

In the process of building a machine learning model for fall detection based on wearable sensor data, I experimented with two classifiers as stated in the project guideline: a Support Vector Machine (SVM) and a Multi-Layer Perceptron (MLP).

Both models were implemented using scikit-learn and optimized using 5-fold cross-validation via GridSearchCV. The data was divided into training (70%), validation (15%), and testing (15%) subsets to ensure a correct evaluation of our model's performance.

SVM Model:

The optimal SVM model was obtained with an 'rbf' kernel and a C parameter of 10. The best score on the validation set achieved by this model was approximately 0.858, indicating that the model correctly identified the action label (fall or no fall) about 85.8% of the time during cross-validation.

On the test set, the SVM model exhibited an accuracy of 87%. It demonstrated a precision, recall, and F1-score of 90%, 88%, and 89% respectively for 'Fall' class, and 82%, 85%, and 84% respectively for 'No Fall' class.

MLP Model:

The optimal MLP model utilized the 'adam' solver with a constant learning rate. It used 'relu' activation and a hidden layer size of (50, 100, 50). The regularization parameter alpha was set to 0.0001. This model achieved a best score of 0.878 on the validation set during cross-validation, indicating that the model correctly identified the action label about 87.8% of the time.

On the test set, the MLP model demonstrated an accuracy of 91%. The model achieved a precision, recall, and F1-score of 91%, 94%, and 92% respectively for 'Fall' class, and 90%, 85%, and 88% respectively for 'No Fall' class.

Conclusions:

In comparison, the MLP model outperformed the SVM model in both validation and test evaluations, indicating it might be a better fit for this fall detection problem. However, both models showed relatively high accuracy, demonstrating that wearable sensor data can indeed be utilized effectively for fall detection. Further improvements might be possible through the investigation of different feature engineering techniques, alternative model architectures, or more comprehensive hyperparameter tuning.

```
SVM Test Performance:
              precision    recall  f1-score   support

           F       0.90      0.88      0.89        52
          NF       0.82      0.85      0.84        33

    accuracy                           0.87        85
   macro avg       0.86      0.87      0.86        85
weighted avg       0.87      0.87      0.87        85

MLP Test Performance:
              precision    recall  f1-score   support

           F       0.91      0.94      0.92        52
          NF       0.90      0.85      0.88        33

    accuracy                           0.91        85
   macro avg       0.91      0.90      0.90        85
weighted avg       0.91      0.91      0.91        85
```