Unsupervised Machine Learning

Final Peer Assignment

Objective of analysis

This project is focused on Principal Component Analysis (PCA) and its implementation for different algorithms. In the project, PCA was used as a pre-processing method for classification algorithms such as Gradient Booster Classifier and Logistic Regression. A comparison of algorithm accuracy scores with and without PCA's best number of components was made. Also, the clustering algorithm K-Means was implemented and visualized in three different modes (1D, 2D and 3D) with the help of the PCA method.

Dataset description

The dataset "Human Activity Recognition Using Smartphones" [1] was downloaded from *UC Irvine Machine Learning Repository* (https://archive-beta.ics.uci.edu/).

The "Human Activity Recognition" dataset was built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist [1].

The dataset contains 10299 rows and 569 columns. Each feature represents 3-axial signals in X, Y and Z directions:

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 [1]:

- 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 the 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.

Additional vectors are obtained by averaging the signals in a signal window sample. These are used on the angle() variable [1]:

- gravityMean
- tBodyAccMean
- tBodyAccJerkMean
- tBodyGyroMean
- tBodyGyroJerkMean

A target variable in the dataset is the "Activity" columns, which contains the following labels:

- Walking
- Walking upstairs
- Walking downstairs
- Sitting
- Standing
- Laying

Data exploration analysis

First, all necessary libraries were imported. The first step in EDA is to get familiar with the data set, for this purpose methods ***.head()***, ***.shape*** and .columns were used. As it was mentioned in the previous section, the dataset contains 10299 rows and 569 columns.

Method ***.info()*** showed type (float, object) of each column and it's missed values. A correlation matrix without diagonal elements was calculated, the strongest correlations between features are shown lower:

```
corr_mat.abs().idxmax()
```

```
tBodyAcc-mean()-X                              angle(tBodyAccMean,gravity)
tBodyAcc-mean()-Y                                   tBodyAcc-entropy()-Y
tBodyAcc-mean()-Z                                   tBodyAcc-entropy()-Z
tBodyAcc-std()-X                                        tBodyAcc-mad()-X
tBodyAcc-std()-Y                                        fBodyAcc-std()-Y
                                                              ...
angle(tBodyGyroMean,gravityMean)                      tBodyGyro-mean()-X
angle(tBodyGyroJerkMean,gravityMean)              tBodyGyroJerk-mean()-X
angle(X,gravityMean)                              tGravityAcc-energy()-X
angle(Y,gravityMean)                                tGravityAcc-mean()-Y
angle(Z,gravityMean)                                tGravityAcc-mean()-Z
Length: 561, dtype: object
```

Labels from the target column "Activity" were encoded with the ***LabelEncoder*** with the following labels:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
data['Activity'] = le.fit_transform(data.Activity)
le_activity_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print(le_activity_mapping)
```

```
{'LAYING': 0, 'SITTING': 1, 'STANDING': 2, 'WALKING': 3, 'WALKING_DOWNSTAIRS': 4, 'WALKING_UPSTAIRS': 5}
```

For further analysis, ***X*** and ***y*** variables were associated with all features and target labels ("Activity"), respectively. X variable was scaled with ***StandardScaler***. X and y subsets were divided into train and test sets with ***train_test_split*** in a 70/30 ratio and ***stratify*** parameter by ***y*** variable.

Classification

As it was mentioned above two classification algorithms, namely Gradient Booster Classifier and Logistic Regression, were implemented. To compare accuracy scores with/without PCA, following parameters were used for each algorithm:

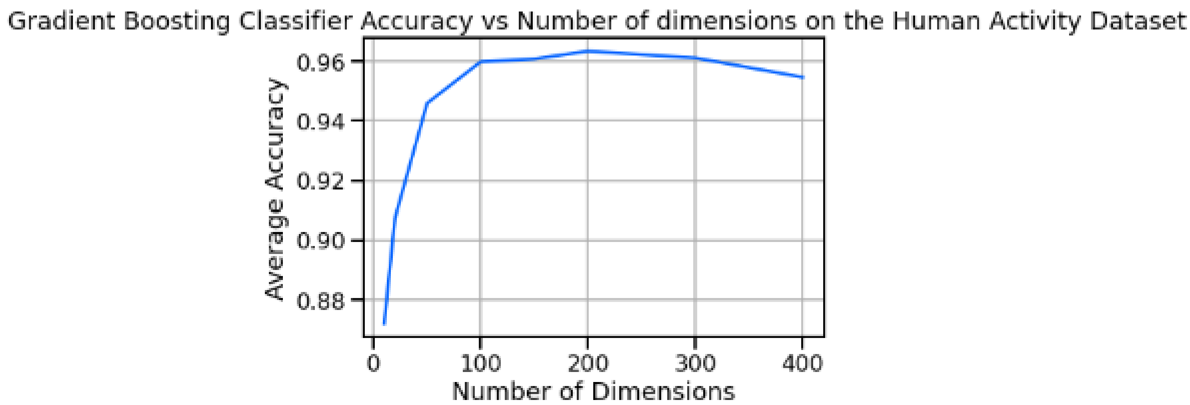
- For Gradient Booster Classifier: ***max_features=4, n_estimators=400, random_state=42, subsample=0.5***

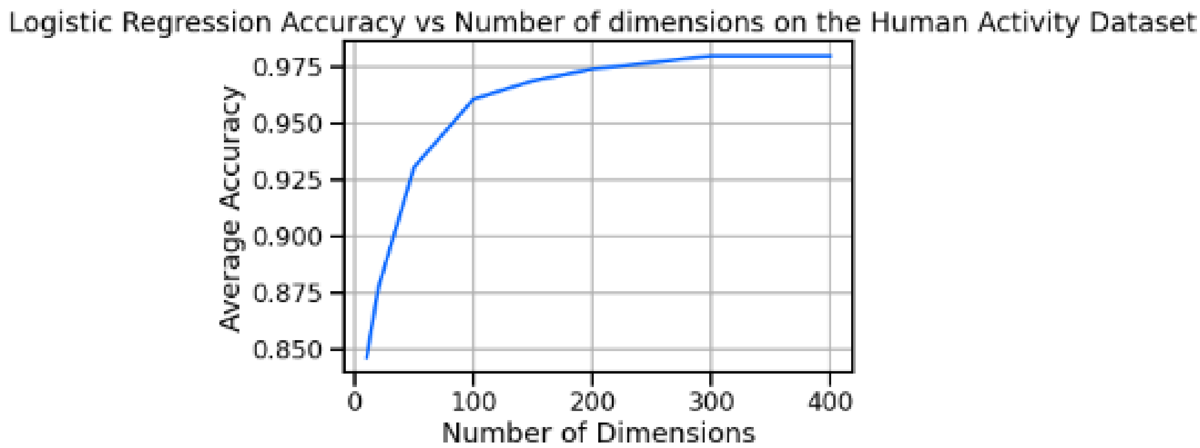
- For Logistic Regression: ***C=0.1, random_state=42, max_iter=10000, fit_intercept=True, solver='liblinear', penalty = "l2"***

For each classification PCA best ***n_components*** parameter and accuracy score, respectively, were calculated.

Gradient Booster Classification			Logistic Regression		
Accuracy score without PCA	Accuracy score with PCA		Accuracy score without PCA	Accuracy score with PCA	
	n_compo-nents	Accuracy score		n_compo-nents	Accuracy score
0.98899	10	0.872168284789644	0.979935	10	0.8462783171521036,
	20	0.9074433656957929		20	0.8776699029126214
	50	0.9459546925566343		50	0.9307443365695793
	100	0.9598705501618123		100	0.96084142394822
	150	0.96084142394822		150	0.9689320388349515
	200	0.9634304207119742		200	0.9741100323624595
	300	0.9611650485436893		300	0.9799352750809062
	400	0.9546925566343042		400	0.9799352750809062

Table 1. Accuracy scores for Gradient Booster Classification and Logistic Regression.





As can be seen in Table 1 and the two graphics below, accuracy scores for the selected classifications are slightly higher without Principal Component Analysis, but not dramatically. The reason for such behavior is that with dimension reduction some amount of information is lost.

K Means clustering

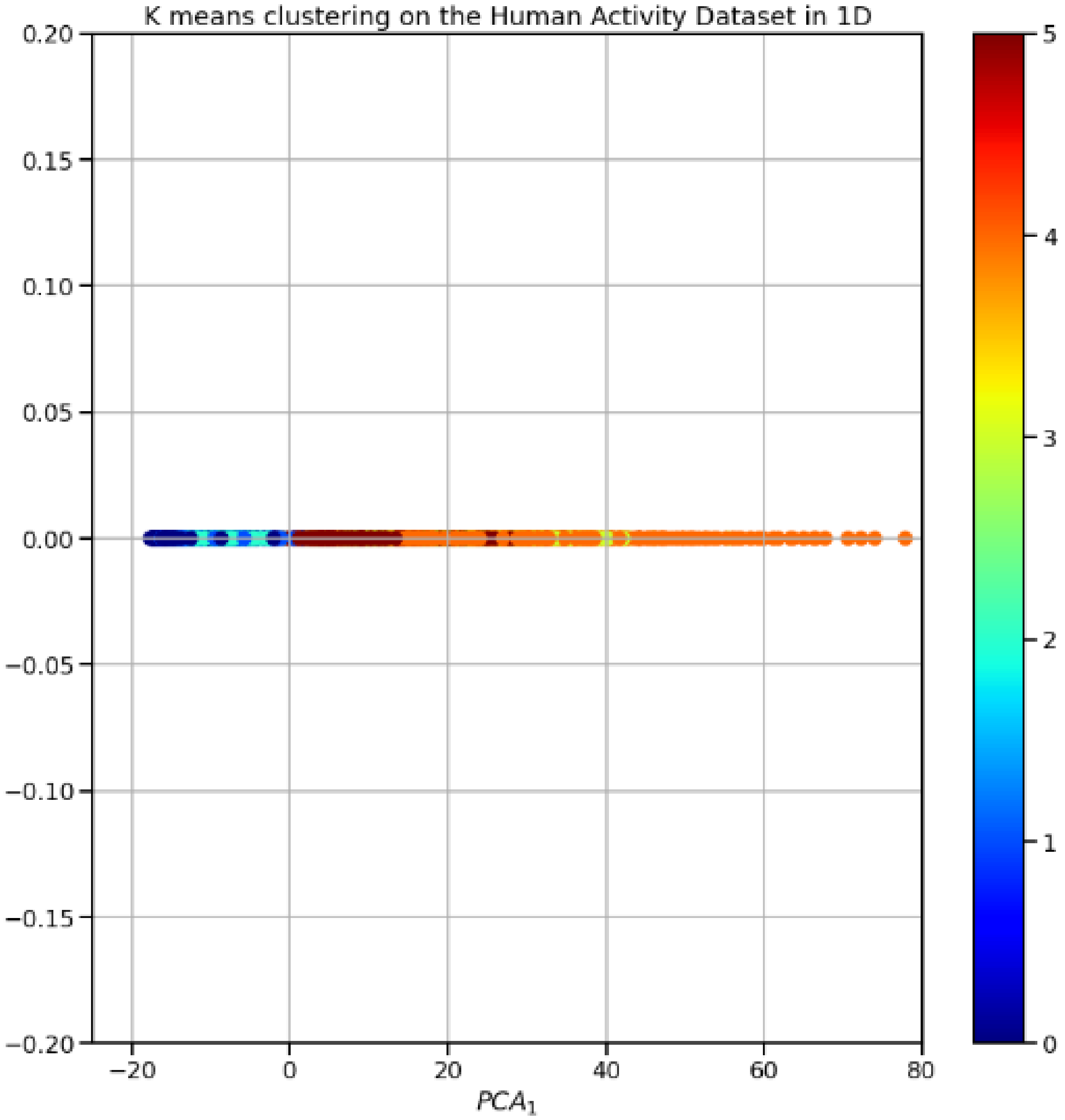
In this section, K means clustering algorithm was implemented. The algorithm had the following parameters:

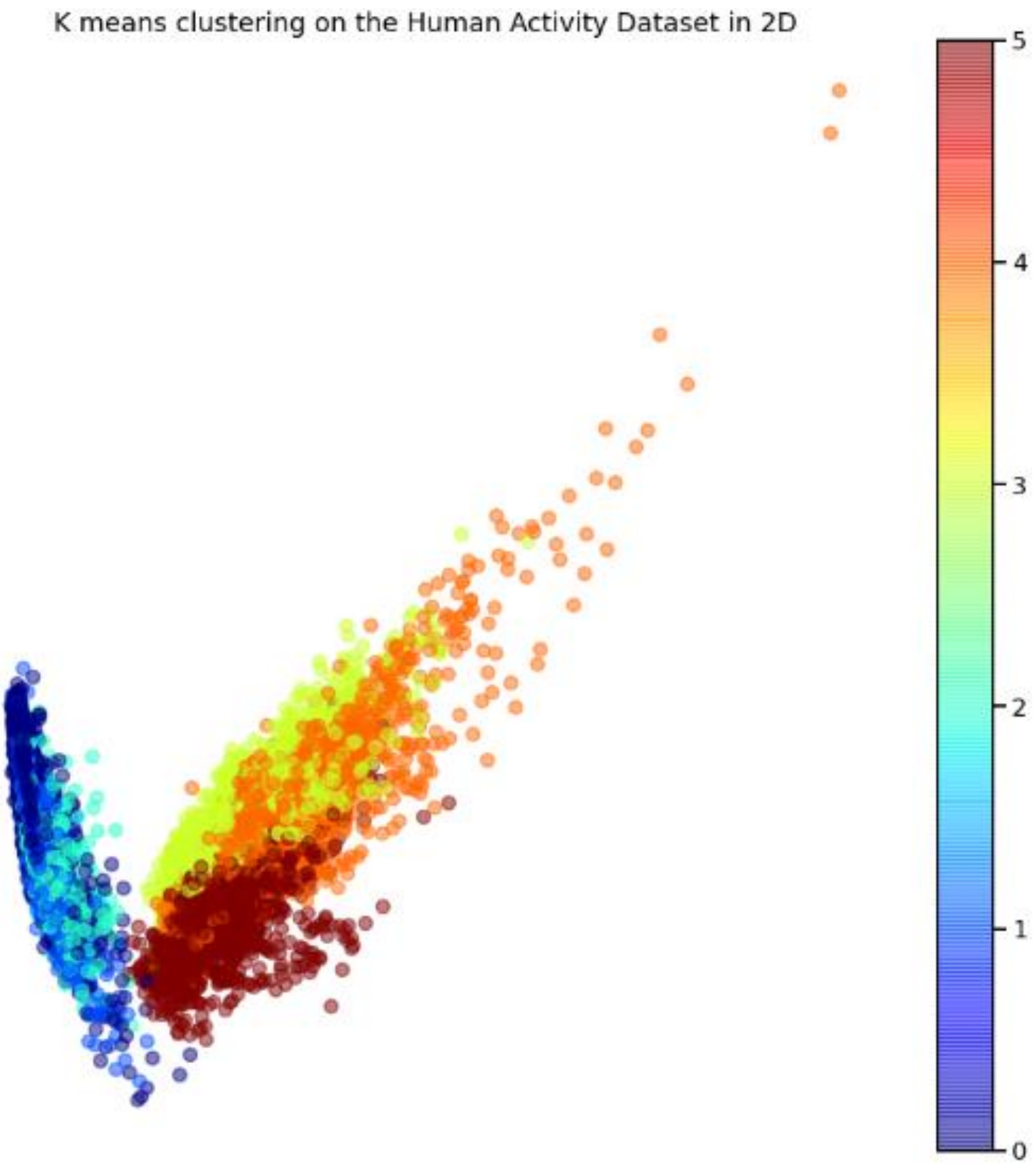
- n_clusters = 6 (because there are 6 defined types of activities)
- init = "k-means++"
- n_init = 12

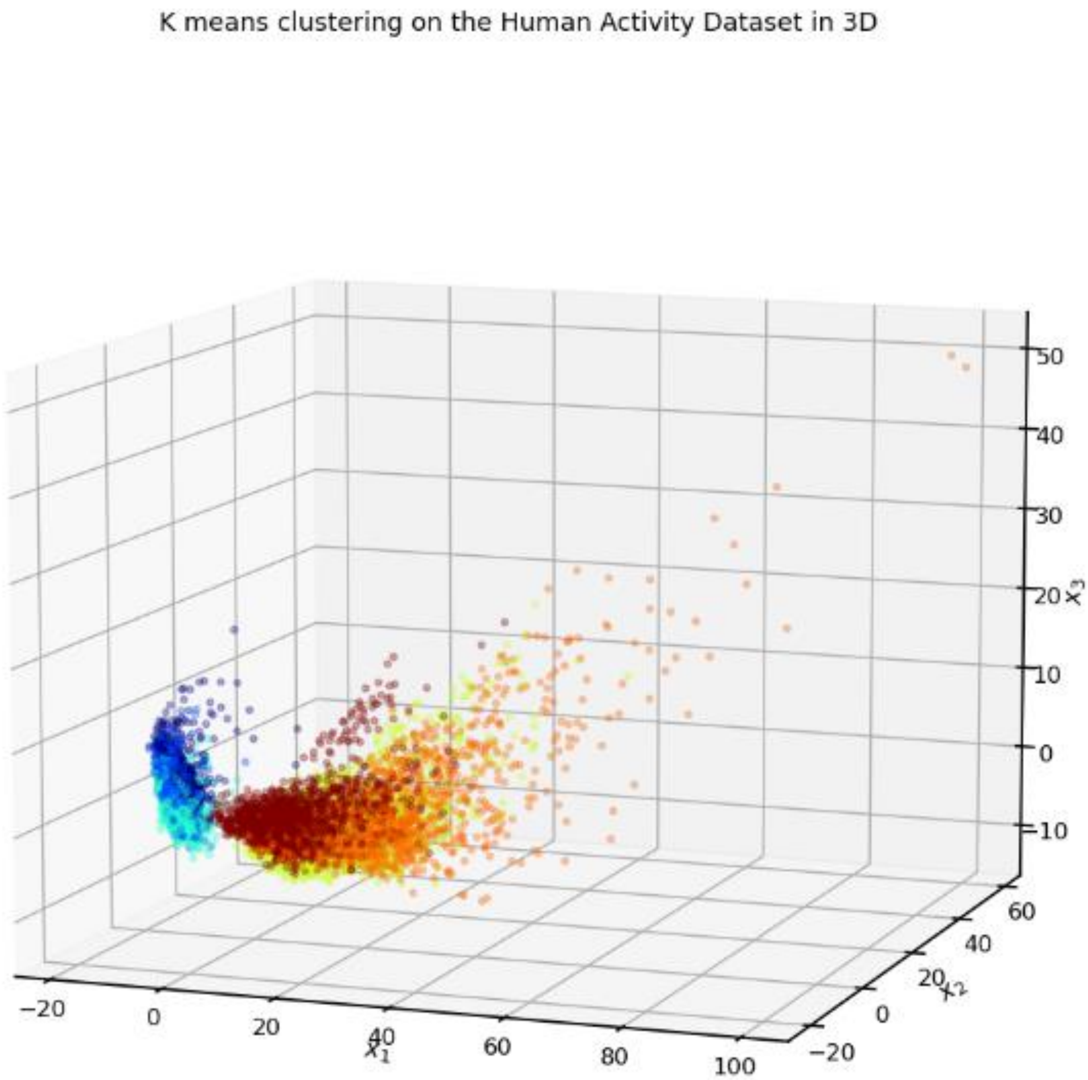
And the following cluster encoding:

- Laying: 0
- Sitting: 1
- Standing: 2
- Walking:3
- Walking downstairs: 4
- Walking upstairs: 5

Silhouette score is 0.11 (closer to 0), which means, that the distance between clusters is not significant or not indifferent. Results of clustering were visualized with the help of PCA method in 1D, 2D and 3D.


K means clustering on the Human Activity Dataset in 1D
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−20
0
20
40
60
80
PCA_1
5
4
3
2
1
0


K means clustering on the Human Activity Dataset in 2D
5
4
3
2
1
0



As shown in the figures above, clusters indeed are not well distinguished and distanced from each other. Reduction or increase of the cluster number will not bring better results.

Conclusion

In this project, different classification models with and without the use of the Principal Component Analysis were created, as well as accuracy scores for each algorithm were calculated. As shown above, the accuracy scores are slightly lower after dimension reduction, but this is explained by a loss of some information during PCA.

K Means clustering shows not really good results, clusters are not good differentiated, this is proved by visualization in 1D, 2D and 3D projections, where clusters are not distinguished good and are overlapping each other.

A general conclusion is: applied classification models suit well for this particular dataset, but clustering, in particular K Means, has not a really good outcome. But for classification models, it is recommended to apply GridSearchCV method to find optimal parameters and also tune hyper ones.

Citations

1. Reyes-Ortiz, Jorge, Anguita, Davide, Ghio, Alessandro, Oneto, Luca & Parra, Xavier. (2012). Human Activity Recognition Using Smartphones. UCI Machine Learning Repository.