# Fashion Dataset

Our data was imported using TensorFlow library using command:

 Command already splits data into train and test, so I concatenated them into arrays X,Y.

A screenshot of a computer program

Description automatically generated

Our data now is ready! We can check the shape of data -> (70000,28,28) -> we have 70k 28x28 arrays, that contain numbers from range [0-255], that represents pixels. For example :

A black and white image of a shoe

Description automatically generated

So our data contains pictures, that we need to flatten to analyse them. (I also flattened data split to train and test set by 1st function to compare it in future analysis.

A computer screen with text

Description automatically generated

Then we reduced dimensions to 2D and 3D using PCA. After that I plotted data in 2D, 3D with labels to see if we can see some dependency between labels and shape of data.

A computer screen with text

Description automatically generated A graph with many dots

Description automatically generated with medium confidence

A black background with white text

Description automatically generated A colorful dot diagram with numbers

Description automatically generated with medium confidence

As we can see its realy hard to see any classification here.

Then we performed clustering. Number of clusters is equal to number of labels of data. Using accuracy score we evaluated the precision of clustering when it comes to precision of labeling. Then we visualize the data in 2D now using labels of clusters.

A computer screen with text

Description automatically generated

Accuracy score = 0.226

Clusters visualised:

A colorful blob of dots

Description automatically generated

Then we performed classification using KNeighborsClassifier with n of neighbors = 5. We used presplitted data from 1st fuction and compared it to our split using model\_selection.train\_test\_split, that was performed 10 times to negate the random factor.

A computer screen shot of a code

Description automatically generated

Score of prefitted data = 0.8554

Score of our fit -> Average = 0.8577

So our fit is better