Aim

1. Take any dataset for the classification task. If you can't find it, then use Fashion MNIST
2. Downsize with UMAP Algorithm
3. Show a graph with a dataset in a two-dimensional space.
4. Use the hdbscan algorithm to find classes.
5. Show derived classes

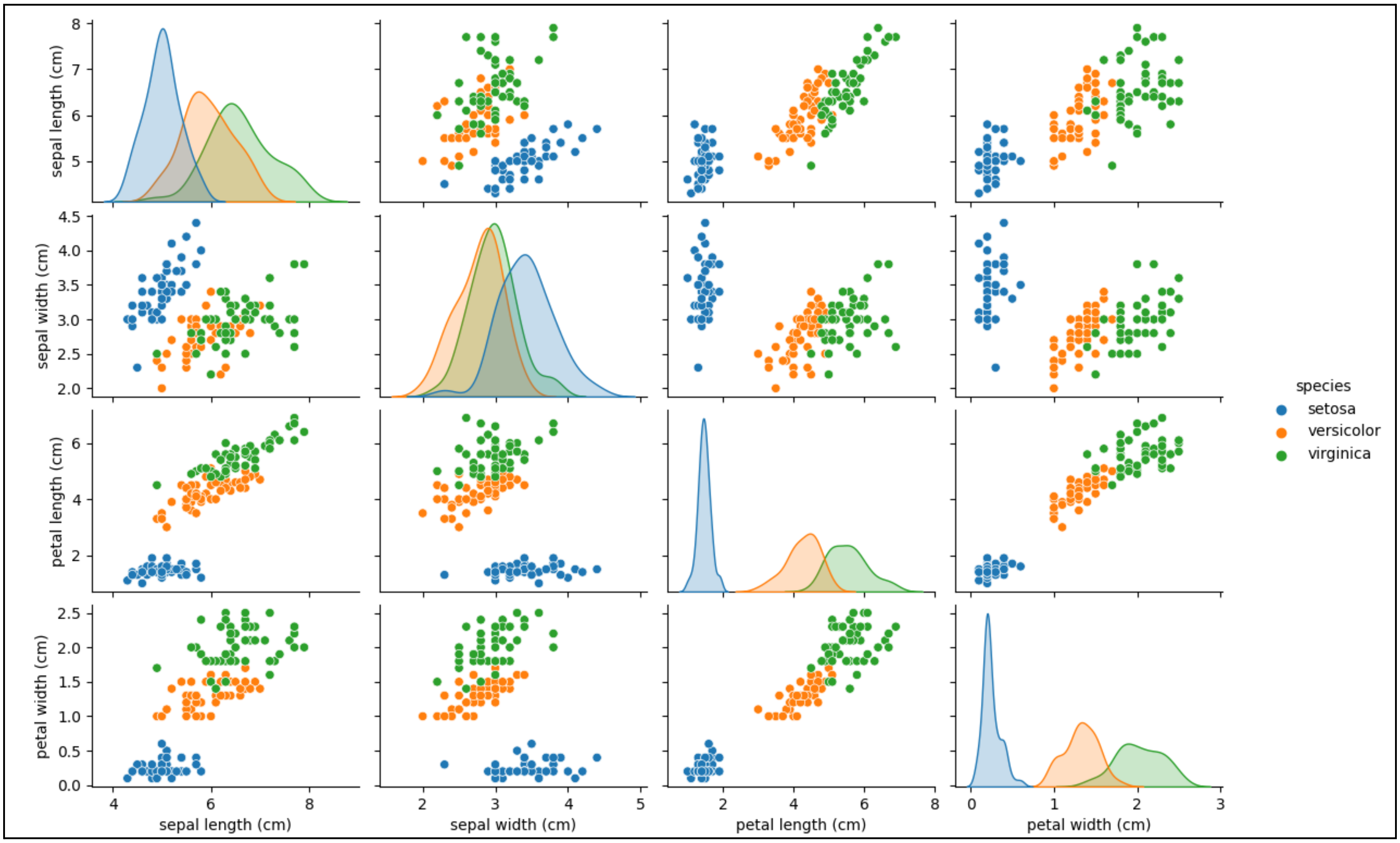
Dataset

In this research we've decided to take the IRIS dataset. It was used in R.A. Fisher's classic 1936 paper, The Use of Multiple Measurements in Taxonomic Problems, and can also be found on the UCI Machine Learning Repository. It includes three iris species with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other. The columns in this dataset are:

- Id
- SepalLengthCm
- SepalWidthCm
- PetalLengthCm
- PetalWidthCm
- Species

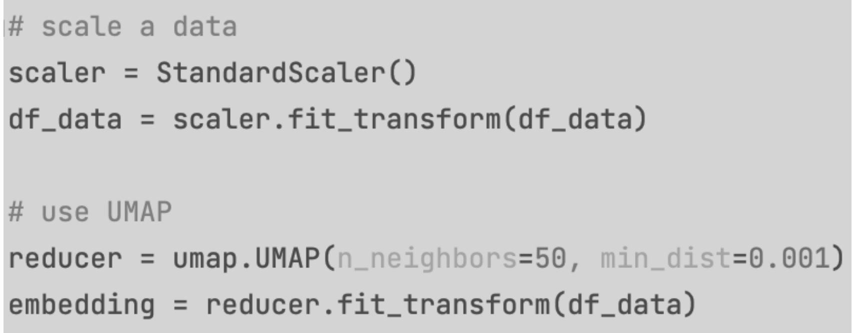
UMAP procedure

Before dimension reduction, it’s also important to show the comparison between all the iris features in 2d, so 4*4 are 16 different graphs.



It is easy to notice that mostly, the Setosa specie of iris is separated from two others. It is common for Versicolor and Virginica types to be similar, but not exactly the same.

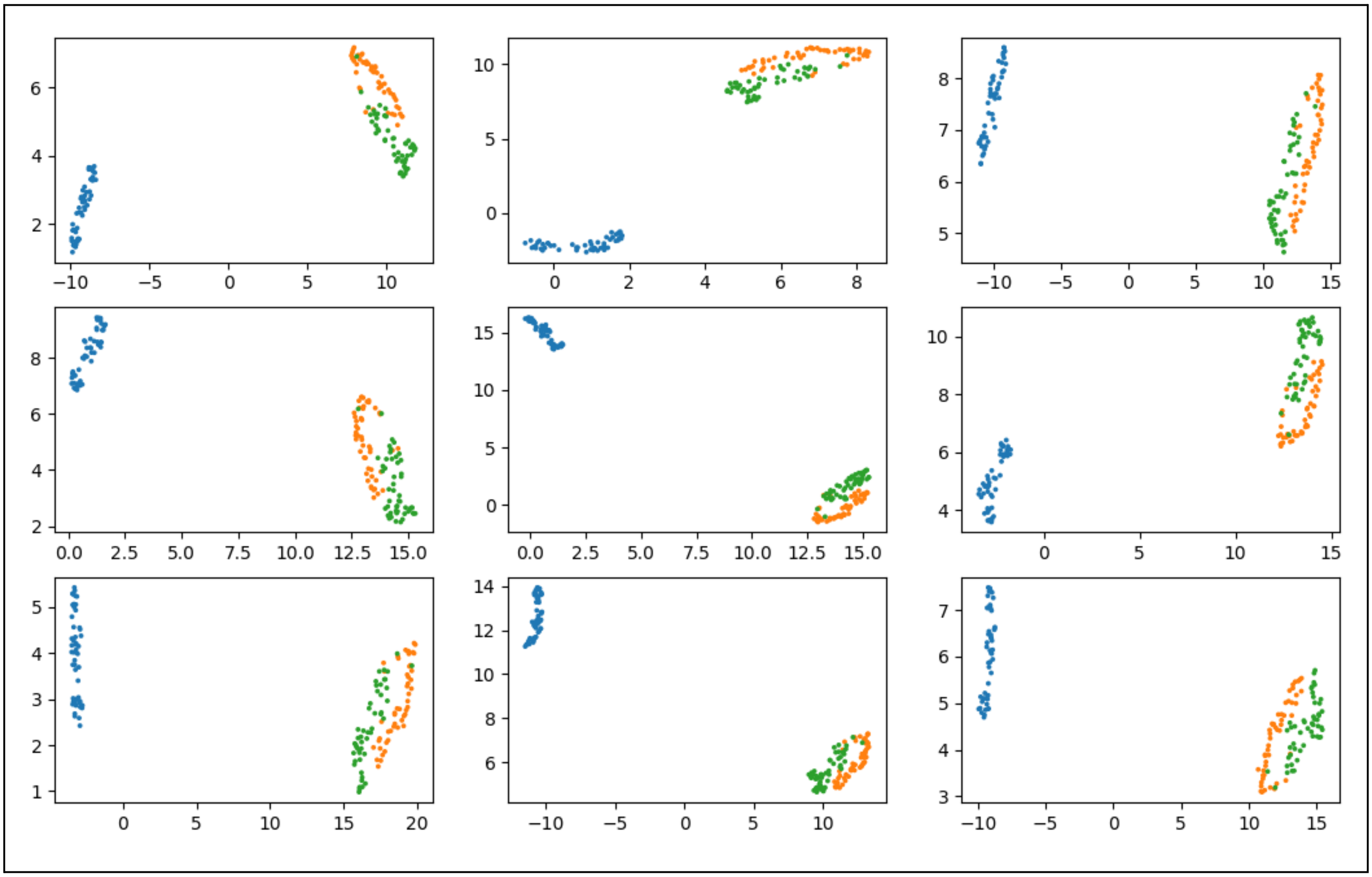
Before using the dimension reduction algorithm, our data needs to be scaled. ‘N_neighbours’ parameter shows how UMAP balances local versus global structure in the data. The min_dist parameter controls how tightly UMAP is allowed to pack points together.



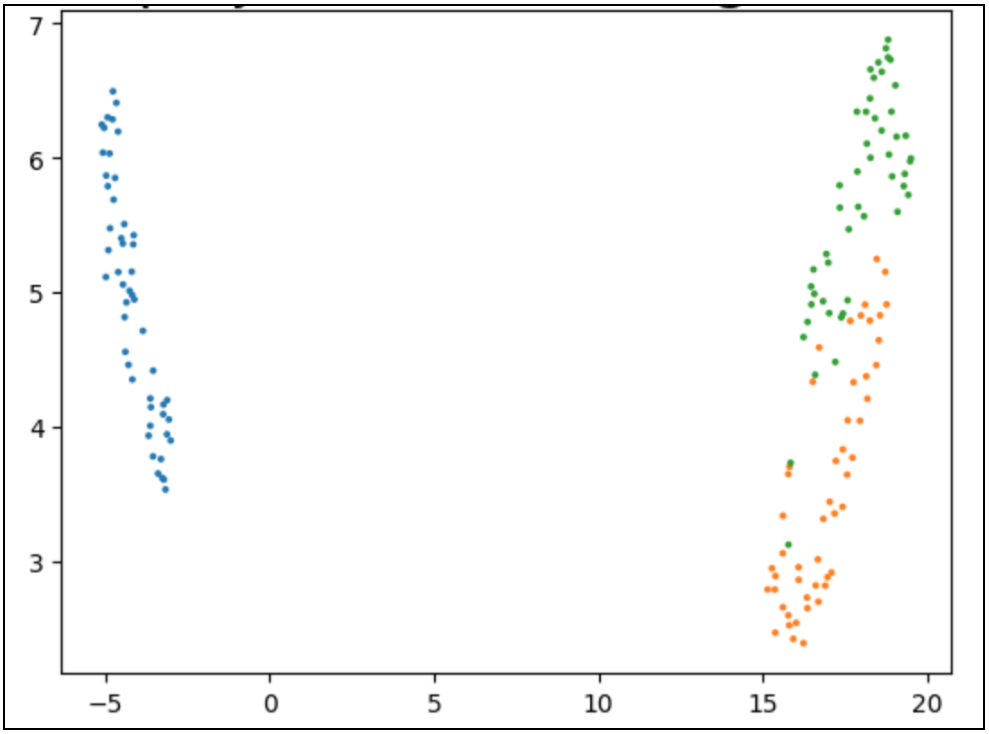
```
# scale a data
scaler = StandardScaler()
df_data = scaler.fit_transform(df_data)

# use UMAP
reducer = umap.UMAP(n_neighbors=50, min_dist=0.001)
embedding = reducer.fit_transform(df_data)
```

On the next graph we can observe 9 trials of UMAP technologie to compress our dataset from 4 to 2 dimensions. In this case our iris classes are coloured with the help of ‘iris.target’ array. The main trend looks similar to one we saw on 2d comparison of features - two resembling closely located groups of points and one is numerically different. Though the idea is the same, but the actual distance between clusters has increased distinctly, this means that UMAP has done separating classes that differ noticeably very well, what we can’t say about green and orange ones.

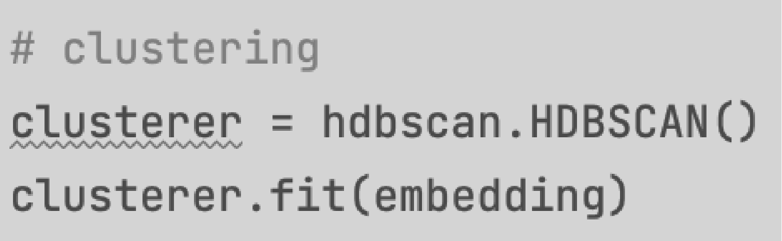


We’ve tried to use the PCA method, and the result was similar.



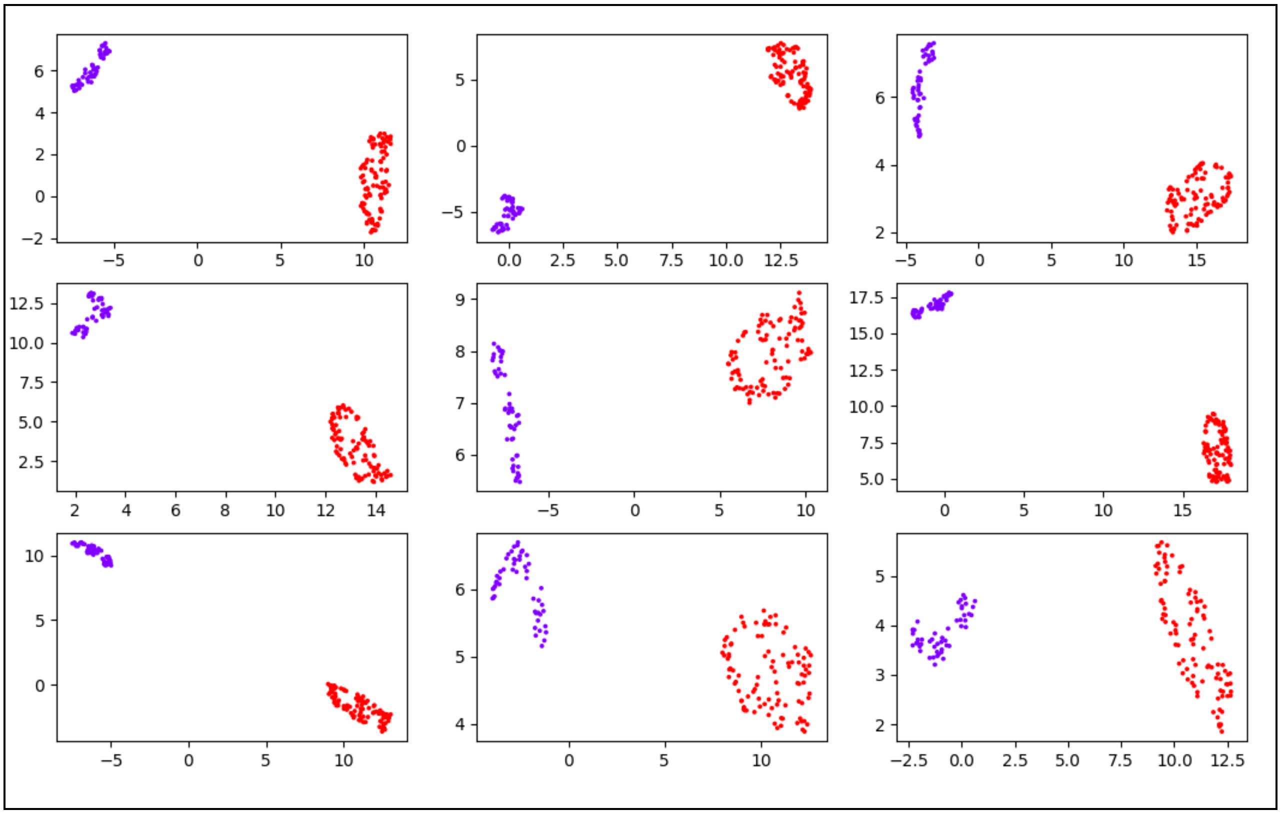
HDBSCAN

This time we need to use HDBscan which is a clustering algorithm developed by Campello, Moulavi, and Sander. It extends DBSCAN by converting it into a hierarchical clustering algorithm, and then using a technique to extract a flat clustering based on the stability of clusters. It is implemented in Python library ‘hdbscan’, basic usage is trivial.



```
# clustering
clusterer = hdbscan.HDBSCAN()
clusterer.fit(embedding)
```

Following set of graphs only differs in a coloring method from the previous one. After applying the algorithm, an array with values which represent the number of a class was created. As we can see here, two classes of three are present. While it’s clearly noticeable that different sets of points belong to different clusters, HHDBscan wasn’t able to separate Versicolor and Verginica types of IRIS. In my humble opinion, it is impossible to do even by eye.



Summary

Dimension reduction is a highly important subject in machine learning and data analysis. It helps to decrease the time spent for processing data and to upper the overall performance. Both UMAP and PCA algorithms are used nowadays, but the first one most commonly will have more efficiency because of non linear transformations. Our research dataset is created that way, so we were able to see clearly a difference between numerical characteristics of iris types on the graph, before and after using UMAP. Although UMAP has increased the distance between different clusters, HDBscan could't differ two similar types of a flower, but worked okay on clustering Setosa out of others. In case of machine training, we expect some problems because of 2 flowers' similarity