Ethan Engel

COMP4431 – Assignment 4

Introduction

The goal of this application is to build a model that predicts Iris species based on sepal and petal measurements. The dataset we will be using is the Iris dataset from the scikit-learn package. A potential use case for this application is for a botanist who would like to quickly identify the iris species based on taken measurements.

Dataset Analysis

Define variables

* Iris pandas data frame
* sepal length in cm (numerical)
* sepal width in cm (numerical)
* petal length in cm (numerical)
* petal width in cm (numerical)

Define labels - Species

class:

* Iris Setosa (category)
* Iris Versicolour (category)
* Iris Virginica (category)

Inputs

* Data import – pulled from 3rd party library with scikit-learn
* From scikit-learn import datasets
* Pip install scikit-learn to virtual environment
* There are 150 observations with 5 features each (sepal length, sepal width, petal length, petal width, species).
* There are no null values, so we don't have to worry about that.
* There are 50 observations of each species (setosa, versicolor, virginica).

Proposed Libraries

Libraries

* Scikit-learn – package provides iris dataset
* Matplotlib – package provides graphs
* Pandas – dataframe support
* Numpy – numerical manipulation of matrices
* Seaborn – pretty graphs, might not use
* Tslearn – Kernel K-means (non-linear clustering)
* Scipy – Dendrograms, other visualization tools

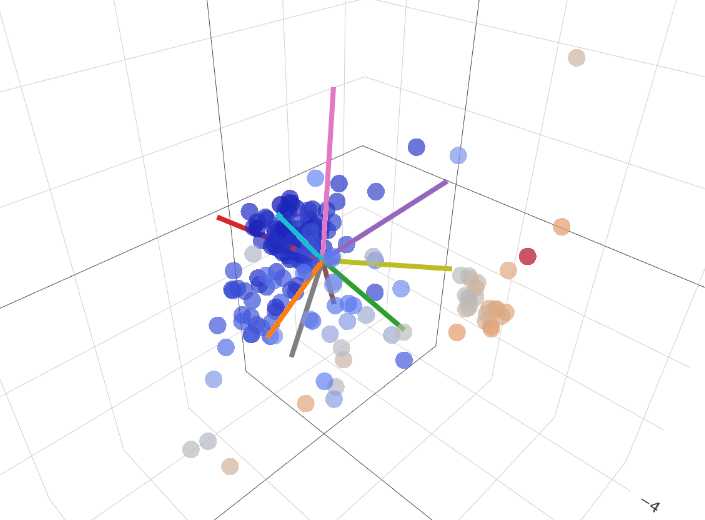
Library source

* Scikit learn - <https://pypi.org/project/scikit-learn/>
* Matplotlib - <https://pypi.org/project/matplotlib/>
* Pandas - <https://pypi.org/project/pandas/>
* Numpy - <https://pypi.org/project/numpy/>
* Python3 - \*We haven’t decided which version of python to use yet
* Seaborn - <https://pypi.org/project/seaborn/>
* Tslearn (0.5.2) - <https://tslearn.readthedocs.io/en/stable/reference.html>
* Scipy - <https://docs.scipy.org/doc/scipy/index.html>

Proposed Solution

* Create dataframe
* Generate K-means, start with K in range 1 to 10 then increase the number of clusters if required. Record the inertia value for each K-means model.
* Create Elbow Build an “Elbow” plot of the data frame. The “elbow joint” tells how many clusters to use.
* Analyze optimal K-means model (scatterplot with lines (planes?) showing where clusters are defined)
* Reduce the data frame using Principal Component Analysis.
* Use Sklearn to scale (MinMaxScaler) the data
* Generate principal components by choosing variance levels explained by each component
* 80 percent was the rule of thumb, can adjust if we have too many vectors for some reason
* Transform data using the PCA model
* Print/visualize components (scatterplot with vectors in plotly express)

Example PCA visualization



* Employ clustering techniques to the data frame and the PCA Features data frame.
* Perform Hierarchical clustering (AgglomerativeClustering object in sklearn with best K-means value)
* Generate dendrogram using scipy and find another visualization method (heatmap, radial dendrograms)
* Fit kernel K-means model and visualize the clusters and boundaries (spectral clustering in sklearn)
* Generate spectral clustering models and choose best
* Perform mean shift clustering (find blobs)
* Ensure that at least one type of clustering is a data hierarchy.
* Develop a conclusion that gives an overview of your results for each analysis, which produced the best and worst results. plot

Outputs

Visualization

* Visualize each type of analysis and describe the results obtained.
* Explore the options for improving visualizations.

Links:

https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

https://www.analytixlabs.co.in/blog/types-of-clustering-algorithms/

https://www.kaggle.com/code/khotijahs1/k-means-clustering-of-iris-dataset/notebook

https://learning.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml

https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60

https://www.mikulskibartosz.name/pca-how-to-choose-the-number-of-components/

https://kirenz.github.io/clustering/k-means.html

https://docs.scipy.org/doc/scipy/index.html

https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html

https://www.python-graph-gallery.com/404-dendrogram-with-heat-map

https://www.data-to-viz.com/graph/dendrogram.html

Conclusions

* Explore the results of each analysis and compare the results of the analysis between the two data frames, include whether PCA improved or diminished the results.

Explain the purpose of clustering models and PCA and why they are useful for classifying the iris types in overlapping areas.

Based on initial EDA, we predict that there will be 2 clusters, one dividing line between setosa and the other two where there is overlap. 3 would be ideal but we do not expect to distinguish between versicolor and virginica.

We do not expect PCA to reduce dimensionality much because there are only 4 features in the data. With PCA we expect slightly worse results because we will only set it to 80 percent of the variance.

We expect similar results between each clustering method.

Spectral might be slightly more accurate due to linearity in the different groups

Preclassified data is cheating in clustering so this is just an exercise

Memes