# Tabular Playground Series - Jul 2022

# Introduction: K-means clustering is an unsupervised machine learning algorithm. We will discuss the basics of K-means clustering and how it works, as well as its applications in real-world scenarios. We will also look at different datasets and their suitability for K-means clustering and provide code implementations using Python. PCA (Principal Component Analysis) is also a unsupervised machine learning algorithm which reduces the dimensions in case of control the overfitting. It does this by identifying the principal components (PCs) that explain most of the variance in the dataset and projecting the data onto these PCs.

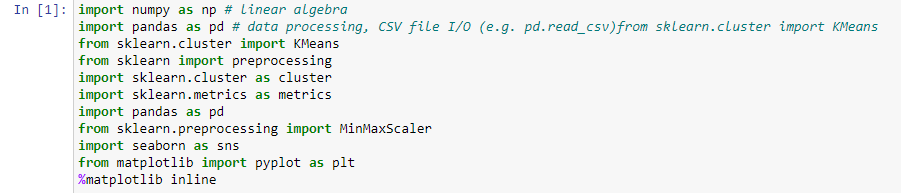
# Data Description:

# you are given (simulated) manufacturing control data that can be clustered into different control states. Your task is to cluster the data into these control states. You are not given any training data, and you are not told how many possible control states there are. This is a completely unsupervised problem one you might encounter in a real-world setting.

* **data.csv** - the file includes continuous and categorical data; your task is to predict which rows should be clustered together in a control state
* **sample\_submission.csv** - a sample submission file in the correct format, where Predicted is the predicted control state

# Importing Libraries:

First, importing the important external Python packages using the pip package manager.



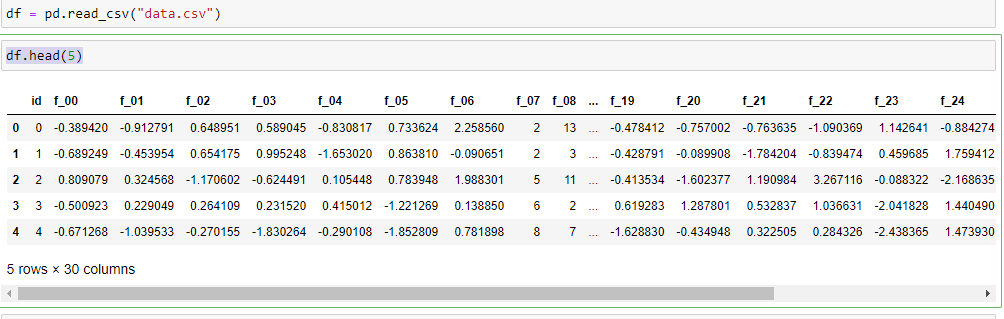
1. NumPy is used for mathematical operations like addition, subtraction, multiplication, division, etc. on arrays and matrices.
2. Pandas provides data structures for efficiently storing and manipulating large datasets, and tools for reading and writing data to and from various file formats, including CSV, Excel, and SQL databases.
3. K-Means is a clustering algorithm that partitions a set of data into K clusters, where K is a user-defined parameter.
4. Preprocessing is a module in the scikit-learn library that provides tools for data preprocessing and feature scaling.
5. The cluster module in scikit-learn contains various clustering algorithms like K-Means, Agglomerative Clustering, and DBSCAN.
6. The metrics module in scikit-learn contains functions for evaluating the performance of machine learning models. It includes metrics for classification, regression, and clustering tasks, such as accuracy, precision, recall, F1-score, and adjusted Rand index.
7. MinMaxScaler is a function in the scikit-learn preprocessing module used for feature scaling. It scales the values of a feature to a range between 0 and 1.
8. Seaborn is a data visualization library based on Matplotlib which is a plotting library used for creating static, interactive, and animated visualizations in Python.

This below code is using the OS module to walk through a directory tree and print out the path of each file in the tree such as data.csv and sample\_submission.csv

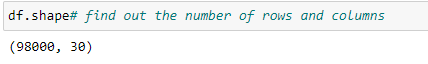
Graphical user interface, text, application

Description automatically generated

1. **Upload the data and display:**

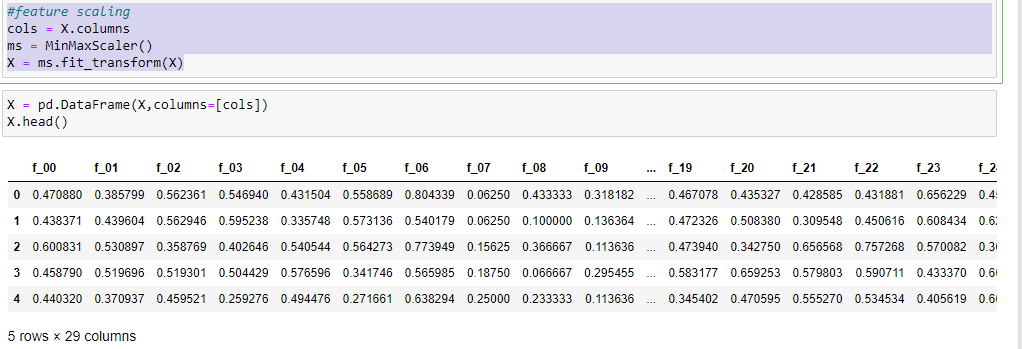


With the help of pandas library, we can read and upload the data in csv form. we can display the first five rows of the data and also by applying **df.shape** we can also find out the number of rows and columns the data has.



1. **Feature Scaling:**

The significance of feature scaling is that it can improve the performance and accuracy of some machine learning algorithms. the code applies MinMaxScaler to scale the features in DataFrame X to have the same scale,which can be important for some machine learning algorithms that are sensitive to the scale of the input features, in order to improve their performance and accuracy.



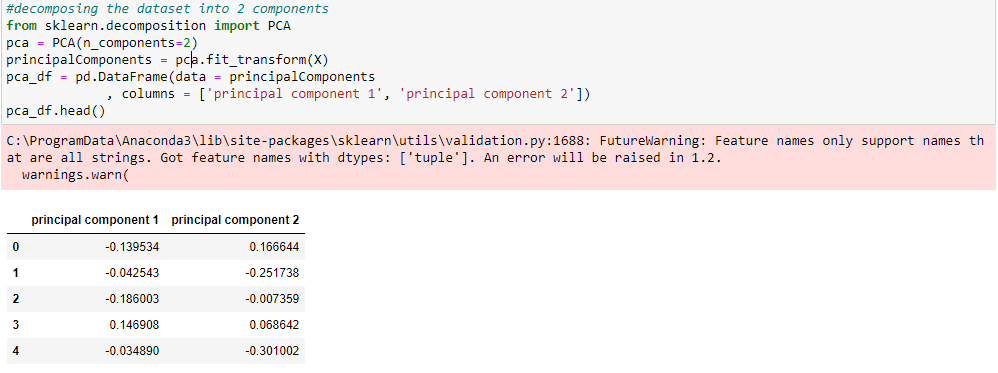
In the above code we did the feature scaling and make a DataFrame using pandas and display the data.

The code **ms = MinMaxScaler()** creates an instance of the MinMaxScaler class, which is a feature scaling technique used in machine learning.

The code **X = ms.fit\_transform(X)** applies the MinMaxScaler to the DataFrame X and assigns the scaled values to a new variable also called X. The .**fit\_transform()** method first fits the scaler to the data (determining the min and max values for each feature) and then applies the transformation to scale the values.

1. **Decomposing the dataset into 2 components:**

For the decomposition part we need to install Principal Component Analysis (PCA) from sklearn on the feature-scaled data in the DataFrame X. PCA is a technique used for dimensionality reduction, where many input features are transformed into a smaller number of principal components (linear combinations of the original features), while retaining most of the variation in the data.



The code **pca = PCA(n\_components=2)** creates an instance of the PCA class and specifies that we want to transform the data into 2 principal components.

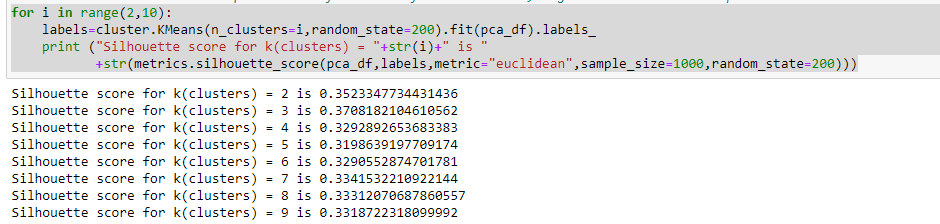
The code **principalComponents = pca.fit\_transform(X)** fits the PCA model to the feature-scaled data in X and transforms it into the 2 principal components. The resulting principalComponents variable contains the transformed data.

After that we made a DataFrame with the help of pandas and gave the two components column name as **principal component 1, principal component 2** as you have shown in the above.

1. **Silhouette score:**

The silhouette value is a measure of how similar an object is to its own cluster compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

Silhouette method to obtain optimal no. of clusters for our dataset, higher value indicates optimal cluster.

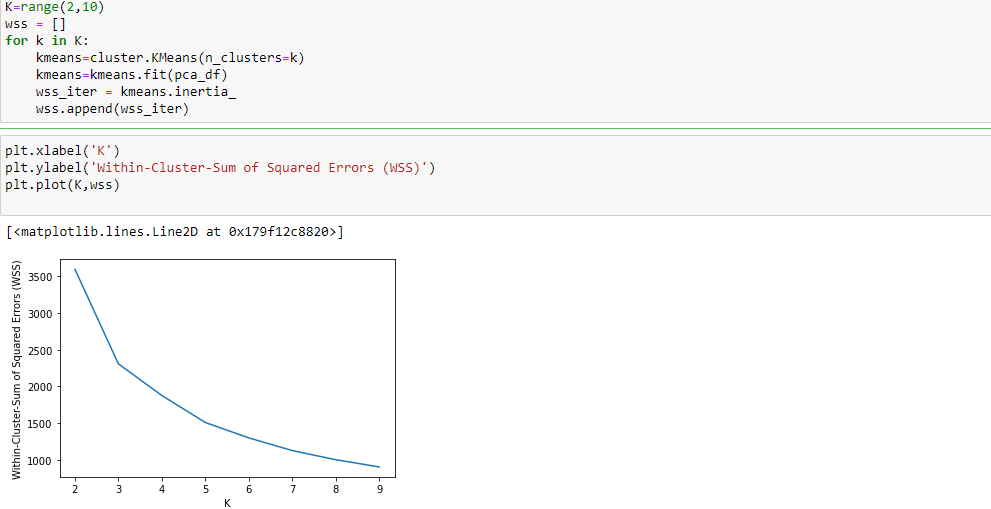


The code performs K-Means clustering on the 2 principal components of the feature-scaled data in X for a range of values of k (number of clusters) from 2 to 9, and computes the silhouette score for each value of k.

This can be useful for determining the optimal number of clusters for the data, as a higher silhouette score indicates a better clustering result.

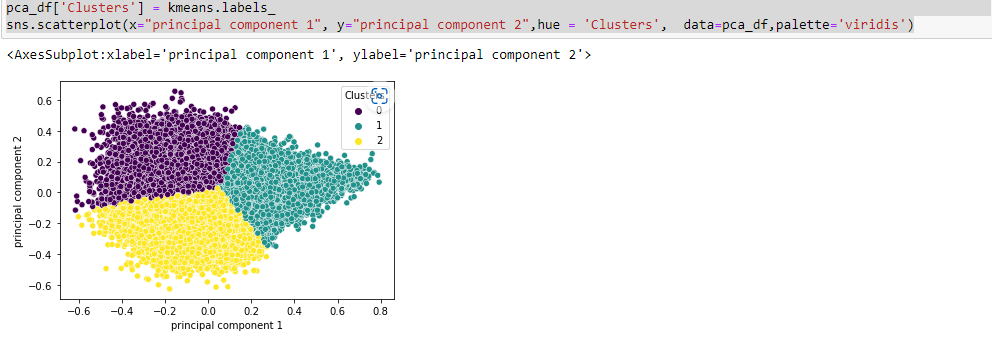
1. **Elbow method:**

The Elbow Method is a graphical approach used to determine the optimal number of clusters in a clustering algorithm.

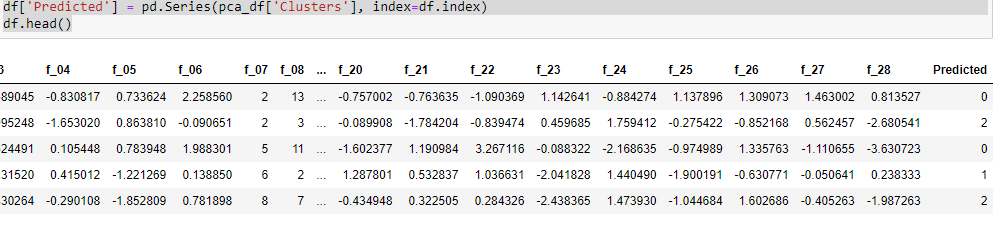


As we can see, there is a sharp elbow in the plotted graph at K=3,and the highest silhouette score is for K=3, therefore we can assume optimal number of clusters is 3.

1. **Visualization of clusters using Scatterplot:**

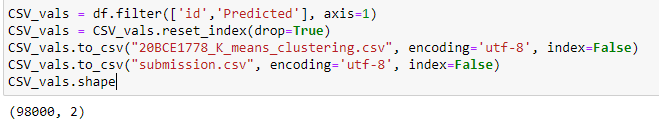


1. **Fitting Cluster labels:**



These lines of code are creating a new column called 'Predicted' in the 'df' DataFrame and filling it with the cluster labels assigned to each data point by the K-means algorithm, which are stored in the 'Clusters' column of the 'pca\_df' DataFrame.

1. **Save files in CSV format:**



These lines of code are creating a new DataFrame called 'CSV\_vals' by filtering the 'id' and 'Predicted' columns from the original 'df' DataFrame using the 'filter()' method.

Finally, the 'shape' attribute is used to get the dimensions of the 'CSV\_vals' DataFrame, which returns a tuple containing the number of rows and columns.