Assignment 10/1: K – Means Clustering (In – class)

Colab file: Week10_K-means_inclass_assignment.ipynb - Colab

Dataset: Mall Customer Segmentation Data

Describe for each column

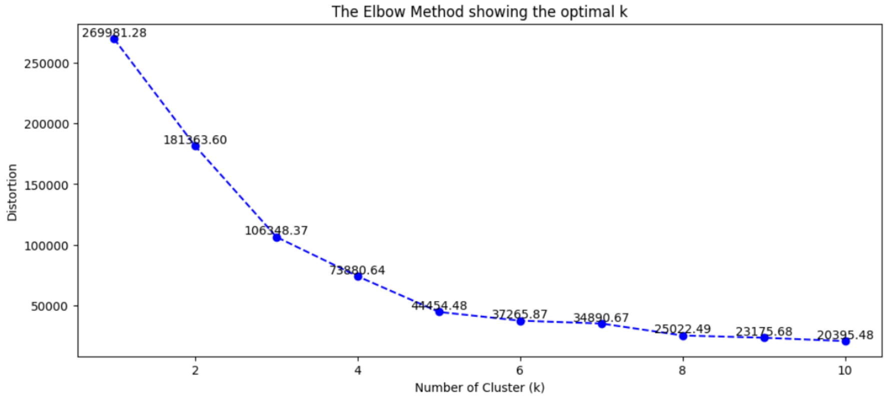
1. *CustomerID (INT)* – Unique ID assigned to the customer
2. *Gender (Category)* – Gender of customer (Female or Male)
3. *Age (INT)* – Age of customer
4. *Annual Income (k$) (INT)* – Annual Income of the customee
5. *Spending Score (INT)* – Score assigned by the mall (1 – 100) based on customer behavior and spending nature

Objectives:

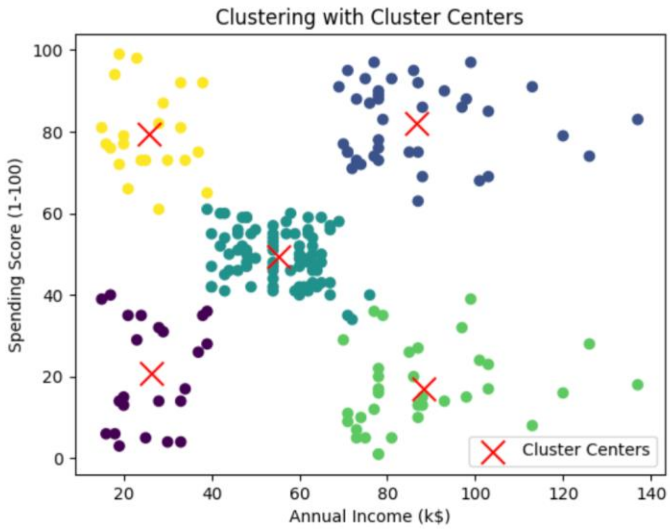
To clustering of customers segmentation between Annual Income vs. Spending Score

Summary Output:

- Use **only** X **is Annual Income, and Spending Score** while Age, and Gender isn't used because of **same spreads between them.**
- From Elbow method, Select k = 5, which has the optimal clusters.



- After clustering, my dataset has 5 clusters.



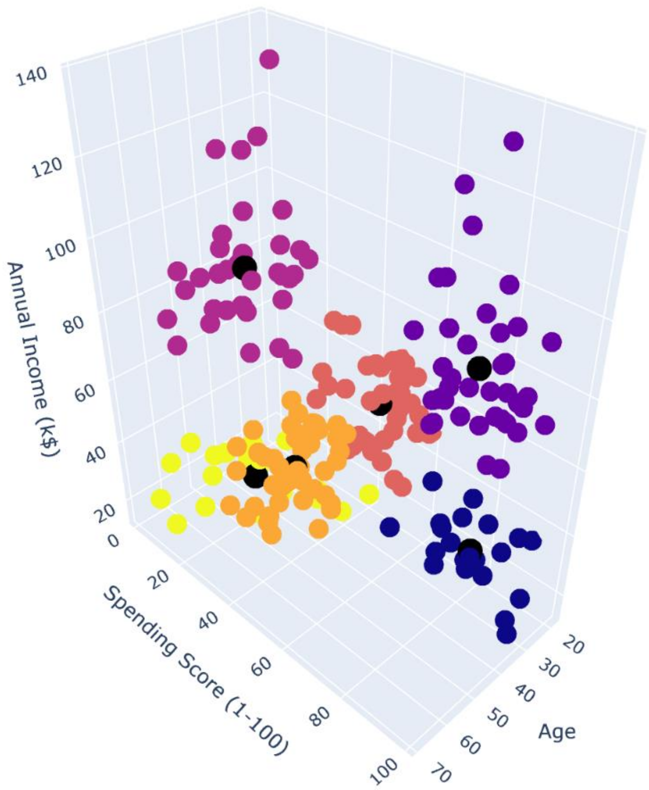
Cluster	Zone	Interpret
1	Purple (Bottom left)	**low** annual income and **low** spending scores.
2	Green (Middle)	**moderate** annual income and spending scores.
3	Yellow (Top left)	**low** annual income but **high** spending scores.
4	Blue (Top right)	**high** annual income and **high** spending scores.
5	Teal (Bottom right)	**high** annual income but **low (to moderate)** spending scores.

- Below table is describe some data.

Cluster	Zone	Center	Size
1	Purple (Bottom left)	(26.30, 20.91)	23
2	Green (Middle)	(86.54, 82.13)	81
3	Yellow (Top left)	(55.30, 49.52)	22
4	Blue (Top right)	(88.20, 17.11)	39
5	Teal (Bottom right)	(25.73, 79.36)	35

Note:

The center (or centroid) is calculated by the average of all data for each cluster.



Overall Process:

1. Introduction and K-Means Overview

- The notebook begins with a brief description of K-Means clustering, explaining its principles, how it works, and its common applications and limitations.
- It also mentions the dataset used, which contains customer information including gender, age, annual income, and spending score.
- Resource Dataset

2. Data Preprocessing

- Import Libraries and Dataset: Essential libraries like Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, and Plotly are imported.
- Exploratory Data Analysis (EDA):
    - The dataset is loaded and initial exploration is performed using df.head(), df.describe(), and df.isnull().sum().
    - The "CustomerID" column is removed as it is not relevant for clustering.
    - Box plots are generated to compare customer attributes (Age, Annual Income, Spending Score) by gender.
- Scatter plots and Pair plots:
    - Scatter plots examine the relationship between annual income and spending score. It suggests a potential structure for customer segmentation.
    - Pair plots show the relationships between all numerical variables and offer a more comprehensive view of the data.
- Correlation analysis:
    - A correlation matrix and heatmap are created to examine the linear relationships between the numerical variables in the dataset.

3. Clustering with K-Means

- Trial Clustering:
    - Initial K-Means clustering is performed with k=6. This is a trial attempt to understand the process.
    - The 'inertia' (within-cluster sum of squares) of the model is computed.
- Elbow Method:
    - The elbow method is used to determine the optimal number of clusters (k) for K-Means. This method calculates the inertia for various values of k,

helping find the point where adding more clusters doesn't significantly reduce the inertia.

- Clustering with Optimal K (k=5):
    - K-Means clustering is then performed using the optimal number of clusters found using the elbow method.
    - The model is fit to annual income and spending score.
    - Cluster labels are added to the df.
- Visualizing Clusters:
    - A scatter plot with cluster centers is created to visually show the identified customer segments.
    - The plots allows for the understanding of the customer segmentation into income and spending clusters.
    - A more beautiful plot is then created with a library *Plotly*, which allows the users to understand the customers clearly.

Assignment 10/2: Hierarchical clustering (In – class)

Colab file: [Week10_HierarchicalClustering_inclass_assignment - Colab]

Dataset: [Air Quality - UCI Machine Learning Repository]

Describe for each column

Column	Variable Name	Role	Type	Description	Units	Missing Values
1	Date	Feature	Date			no
2	Time	Feature	Categorical			no
3	CO(GT)	Feature	Integer	True hourly averaged concentration CO in mg/m^3 (reference analyzer)	mg/m^3	no
4	PT08.S1(CO)	Feature	Categorical	hourly averaged sensor response (nominally CO targeted)		no
5	NMHC(GT)	Feature	Integer	True hourly averaged overall Non Metanic Hydrocarbons concentration in microg/m^3 (reference analyzer)	microg/m^3	no
6	C6H6(GT)	Feature	Continuous	True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)	microg/m^3	no
7	PT08.S2(NMHC)	Feature	Categorical	hourly averaged sensor response (nominally NMHC targeted)		no
8	NOx(GT)	Feature	Integer	True hourly averaged NOx concentration in ppb (reference analyzer)	ppb	no

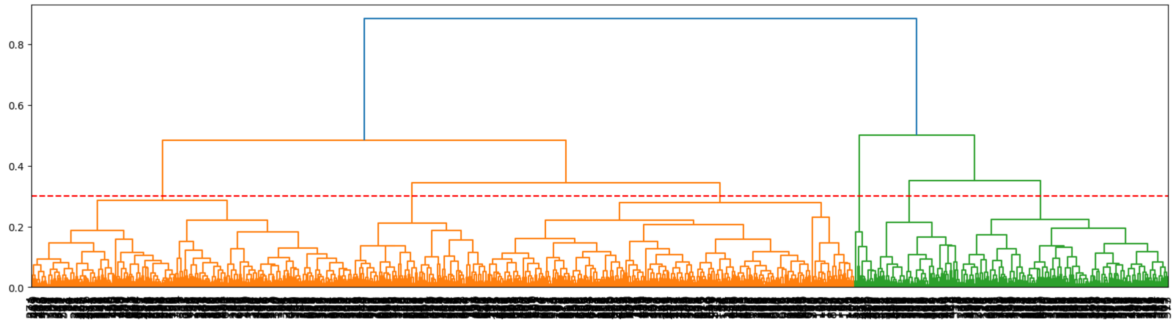
9	PT08.S3(NOx)	Feature	Categorical	hourly averaged sensor response (nominally NOx targeted)		no
10	NO2(GT)	Feature	Integer	True hourly averaged NO2 concentration in microg/m^3 (reference analyzer)	microg/m^3	no
11	PT08.S4(NO2)	Feature	Categorical	hourly averaged sensor response (nominally NO2 targeted)		no
12	PT08.S5(O3)	Feature	Categorical	hourly averaged sensor response (nominally O3 targeted)		no
13	T	Feature	Continuous	Temperature	°C	no
14	RH	Feature	Continuous	Relative Humidity	%	no
15	AH	Feature	Continuous	Absolute Humidity		no

Objectives:

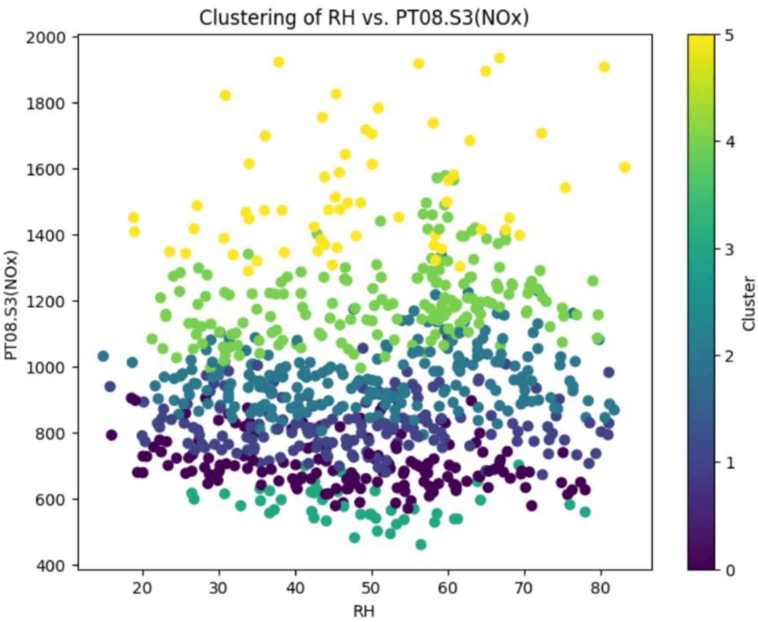
To clustering of data between Relative Humidity (RH) vs. Concentration of NO_x PT08.S3(NOx)

Summary Output:

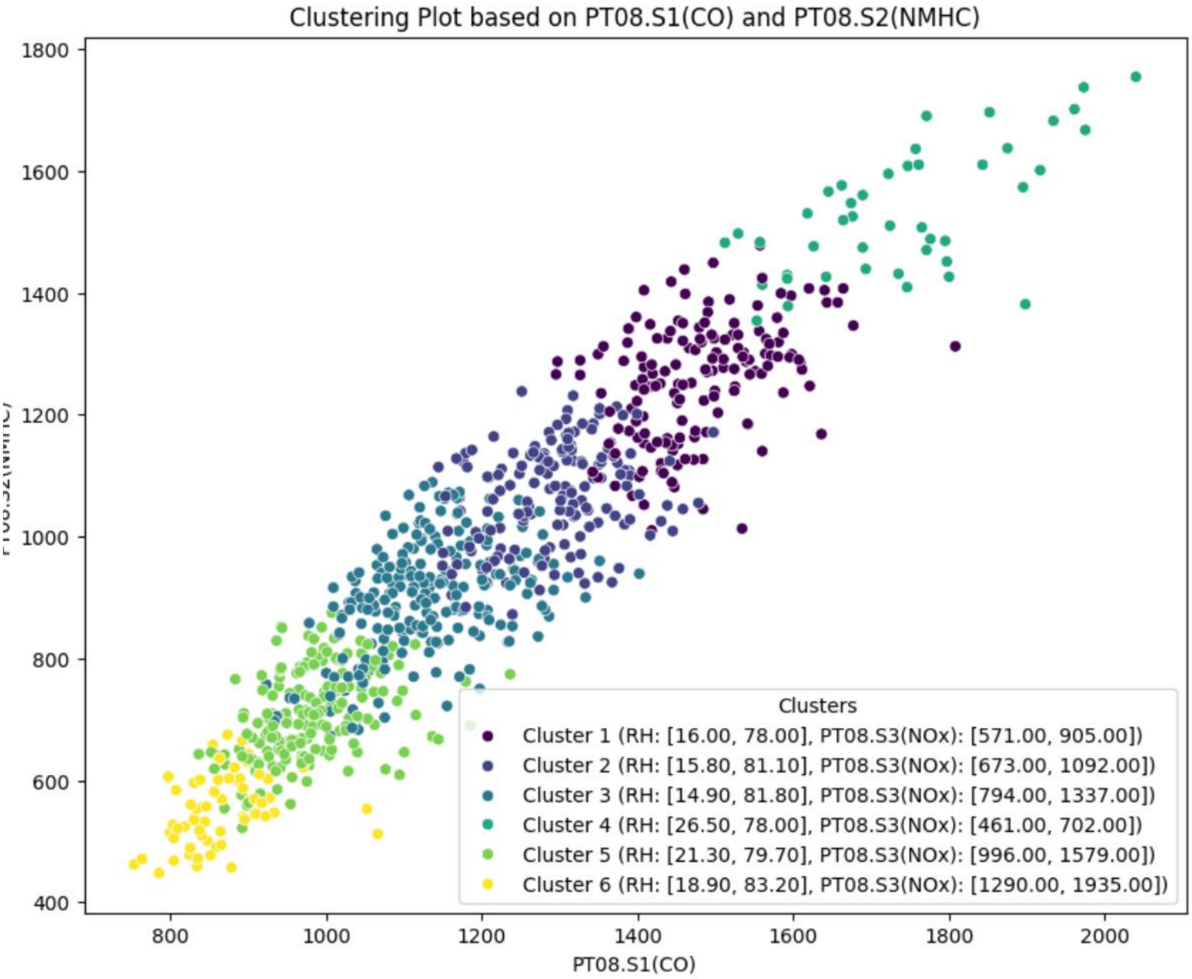
- Before data preprocessing, Use **X is dataframe, which drop "PT08.S3(NO_x)" and "RH"**. It has only 827 rows, and 12 columns.
- After normalize the movement data, from distance_threshold at 0.30, I can split the relation with ***6*** clusters.



- I use linkage with ***"ward"***. Moreover, after the predicted model, the range of dataset for "PT08.S3(NO_x)" and "RH" with 6 clusters are




```
Cluster 1 Summary:
RH range: [16.0, 78.0]
PT08.S3(NOx) range: [571.0, 905.0]
--------------------------------------------------
Cluster 2 Summary:
RH range: [15.8, 81.1]
PT08.S3(NOx) range: [673.0, 1092.0]
--------------------------------------------------
Cluster 3 Summary:
RH range: [14.9, 81.8]
PT08.S3(NOx) range: [794.0, 1337.0]
--------------------------------------------------
Cluster 4 Summary:
RH range: [26.5, 78.0]
PT08.S3(NOx) range: [461.0, 702.0]
--------------------------------------------------
Cluster 5 Summary:
RH range: [21.3, 79.7]
PT08.S3(NOx) range: [996.0, 1579.0]
--------------------------------------------------
Cluster 6 Summary:
RH range: [18.9, 83.2]
PT08.S3(NOx) range: [1290.0, 1935.0]
--------------------------------------------------
```


Clustering Plot based on PT08.S1(CO) and PT08.S2(NMHC)
1800
1600
1400
1200
1000
800
600
400
800
1000
1200
1400
1600
1800
2000
PT08.S1(CO)
Clusters
Cluster 1 (RH: [16.00, 78.00], PT08.S3(NOx): [571.00, 905.00])
Cluster 2 (RH: [15.80, 81.10], PT08.S3(NOx): [673.00, 1092.00])
Cluster 3 (RH: [14.90, 81.80], PT08.S3(NOx): [794.00, 1337.00])
Cluster 4 (RH: [26.50, 78.00], PT08.S3(NOx): [461.00, 702.00])
Cluster 5 (RH: [21.30, 79.70], PT08.S3(NOx): [996.00, 1579.00])
Cluster 6 (RH: [18.90, 83.20], PT08.S3(NOx): [1290.00, 1935.00])

Overall Process:

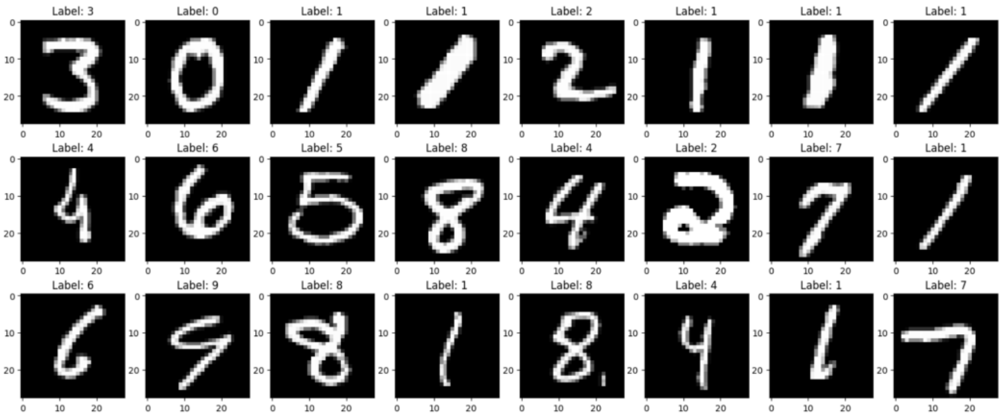
1. **Data Preparation:**
   - Imported the dataset from a provided URL.
   - Removed unnecessary columns (Date, Time).
   - Handled missing values by dropping rows with NaN.
   - Converted data types to float and replaced ',' with '.' for numerical consistency.
   - Removed rows with negative values, indicating potential errors.
2. **Exploratory Data Analysis (EDA):**
   - Visualized data distributions using histograms for each feature.
   - Created 3D and 2D scatter plots to understand relationships between variables like Relative Humidity (RH), NOx concentration, and Temperature (T).
3. **Hierarchical Clustering:**
   - Model Building: Used AgglomerativeClustering from scikit-learn to build the hierarchical model.
   - Dendrogram Visualization: Plotted a dendrogram to visualize the hierarchical structure of clusters.
   - Normalization: Normalized the data using normalize() to account for different scales among features, promoting a more equitable comparison.
   - Finding Optimal Number of Clusters (K): Used the dendrogram and a horizontal line to identify an appropriate number of clusters based on the distance at which the clusters naturally separate. The notebook shows an example using a threshold of 0.3, resulting in 6 clusters.
   - Linkage Method: 'ward' linkage was used for the final clustering model.
4. **Final Model and Results:**
   - Final Model: The final model was built with 6 clusters and 'ward' linkage.
   - Cluster Assignments: Each data point was assigned to a cluster based on the model's prediction.
   - Cluster Analysis: The notebook analyzed the characteristics of each cluster (e.g., the range of RH and NOx values).
   - Visualization: A scatter plot was used to visualize the clusters based on RH and NOx, showing the distinct groups identified by the clustering algorithm.

Assignment 10/3: t-SNE (Homework)

Colab file: [Week10_t-SNE_Homework - Colab]()

Dataset: MNIST_784

Sample Data:



t-SNE Visualization of MNIST with KMeans Evaluation

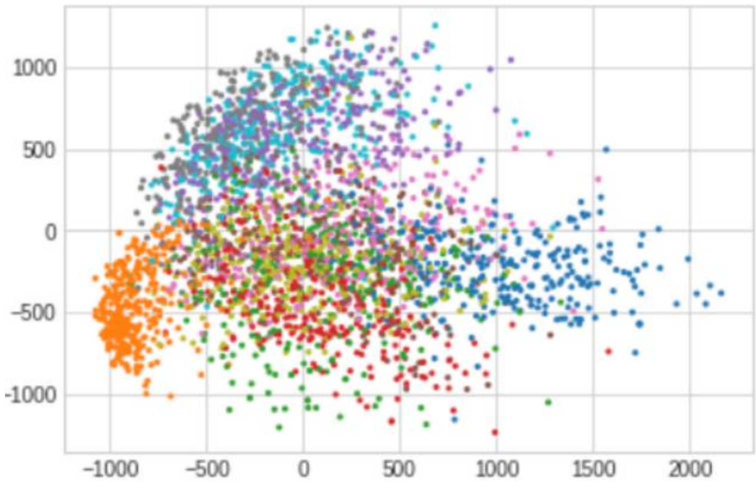
explores the use of t-SNE for visualizing the MNIST dataset and evaluates different perplexity values using KMeans clustering.

Process Overview

1. **Data Loading and Preprocessing:**
    - The MNIST dataset is loaded using fetch_openml.
    - The data is standardized using StandardScaler.
    - PCA Dimensionality Reduction: PCA is applied to reduce the dimensionality of the **data to 30 components from 784 components** to improve the speed of t-SNE computation and reduce the computational burden.
2. **t-SNE Parameter Tuning (Perplexity):**
    - A loop iterates through different perplexity values (defined in perplexity_values).
    - For each perplexity:
        - t-SNE Transformation: t-SNE is applied to the PCA-reduced data with the current perplexity, *learning_rate = 1000, max_iter = 3000, and random_state = 42.*

        - KMeans Clustering: KMeans clustering (with *n_clusters=10* and *random_state=42*) is performed on the t-SNE reduced data.
        - Inertia Evaluation: The KMeans inertia is calculated as a metric to evaluate the quality of the clustering for the given perplexity. Lower inertia generally indicates better clustering.
        - Visualization: A scatter plot is generated to visualize the t-SNE transformed data, colored by the true labels (digits 0 - 9).
    - The perplexity that results in the lowest KMeans inertia is chosen as the best perplexity.

3. **Final Visualization with Best Perplexity:**
    - Using the best perplexity found, t-SNE is applied again to the PCA reduced data.
    - The final visualization of the MNIST dataset in 2D space is created with the best perplexity.
4. **Analysis of Perplexity Impact:**
    - A plot shows the relationship between perplexity and the KMeans inertia.



This picture is shown as Naively using PCA

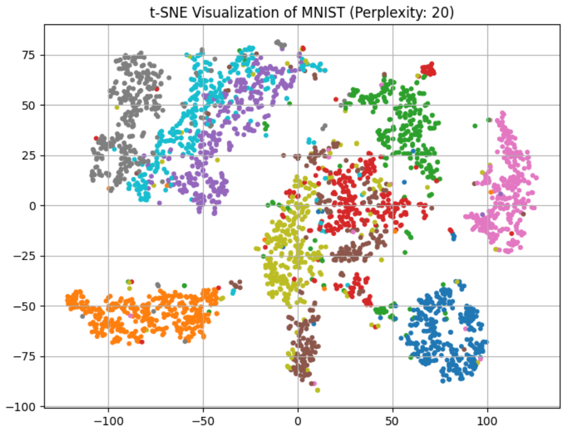
Hyperparameters

- PCA: n_components = 30, random_state = 42
- t-SNE: n_components = 2, learning_rate = 1000, max_iter = 3000, random_state = 42
- Perplexity: Iterated through perplexity_values = [20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 40, 42, 43, 45, 48, 50]
- KMeans: n_clusters = 10, random_state = 42

Results

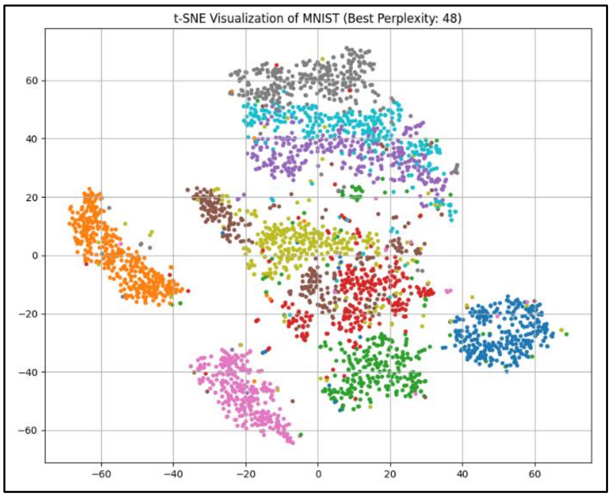
The code produces the following outputs:

- **Scatter Plots:** A series of scatter plots for each perplexity value, visualizing the 2D representation of the MNIST data after t-SNE transformation.

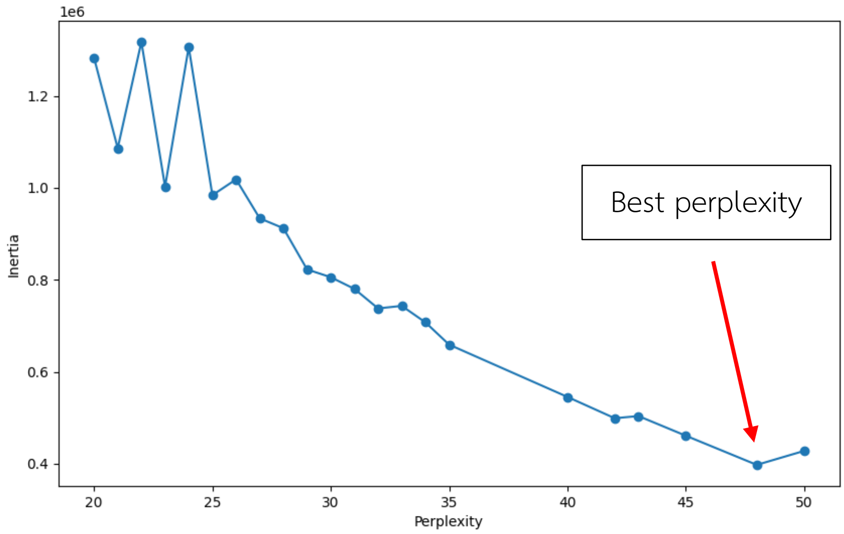


Example; This is MNIST where perplexity = 20

- **Best Perplexity:** The perplexity value that <u>minimizes the KMeans inertia</u> is reported as the best perplexity.
- **Final Visualization:** A scatter plot using the best perplexity showing the 2D representation of the MNIST data.



- **Perplexity vs. Inertia Plot:** A plot demonstrating how the KMeans inertia changes with different perplexity values.

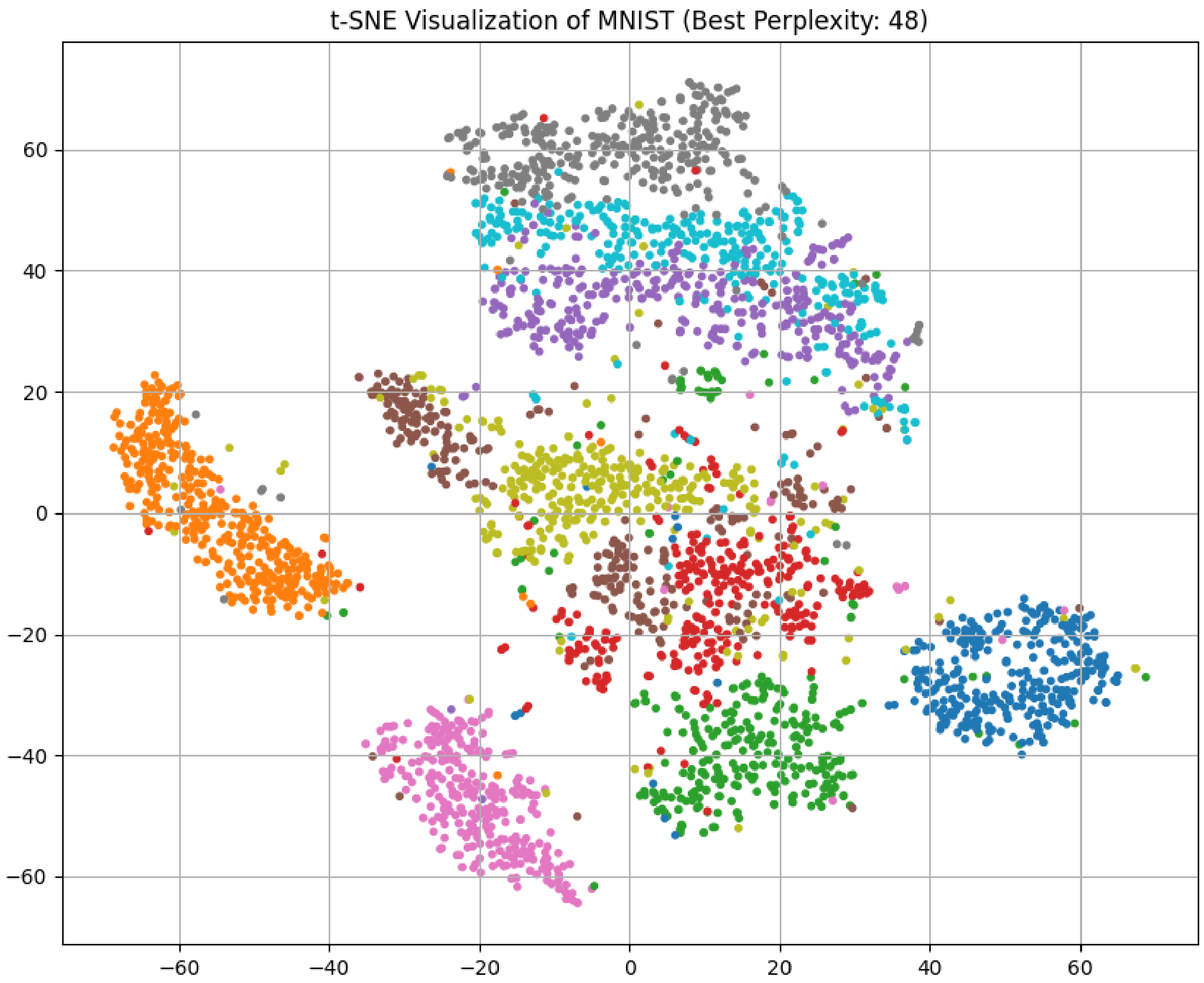


How to achieve beautiful results?

The good results are achieved by carefully tuning the t-SNE hyperparameters, specifically the perplexity. The perplexity parameter controls the local neighborhood size used by t-SNE. By <u>iterating through different perplexity values</u> and <u>evaluating the resulting clustering quality with KMeans inertia</u>, we are able to find a perplexity that optimally balances local and global structure in the data.

Furthermore, applying PCA for dimensionality reduction helps to significantly speed up the t-SNE process, especially for high-dimensional datasets like MNIST.

By utilizing the combination of PCA, t-SNE, and KMeans, we are able to obtain an effective and insightful visualization of the complex structure within the MNIST dataset.


t-SNE Visualization of MNIST (Best Perplexity: 48)
60
40
20
0
−20
−40
−60
−60
−40
−20
0
20
40
60