Project Report

Intern Name: Ahmed Islam

Department: Artificial Intelligence

Organization: IR Solutions

Designation: AI Intern

Week # 04

Submitted to: Asad Ali

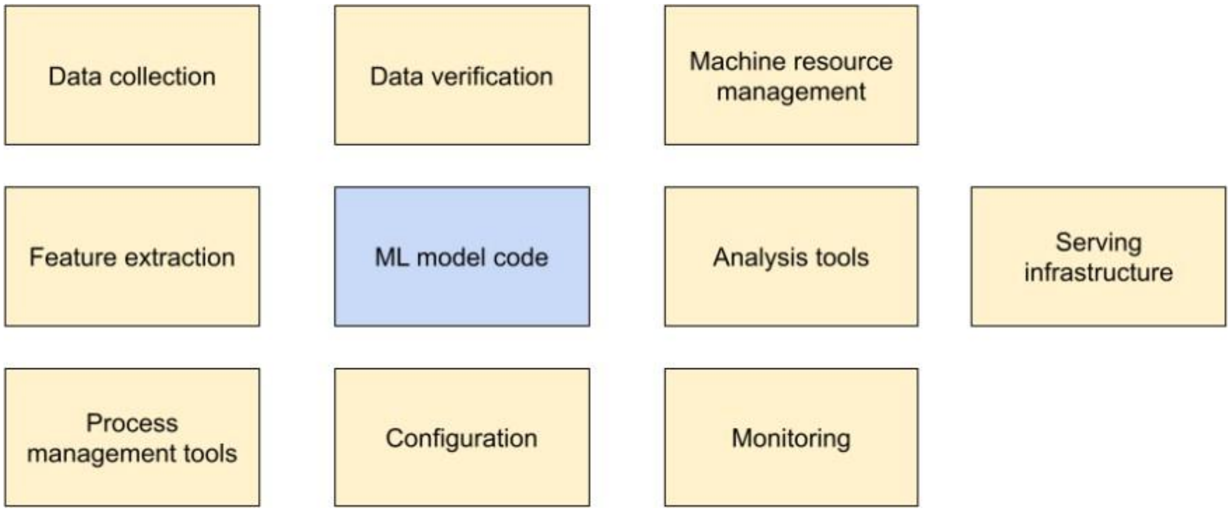
Submission Date: 01-Aug-2025

Report Type: Combined Tasks Documentation

Table of Contents

Theoretical Understanding

Production Ml system



Static vs dynamic model Training

- **Static training** (also called offline training) means that you train a model only once. You then serve that same trained model for a while.
- **Dynamic training** (also called online training) means that you train a model continuously or at least frequently. You usually serve the most recently trained model.

Inference

Inference is the process of using a trained machine learning model to make predictions on new, unseen data.

There are two types of inference

- **Static inference:** Pre-compiled and optimized before it is deployed.
- **Dynamic inference:** The model is executed at runtime, and its structure can change depending on the input.

When to transform data

Raw data must be feature engineered (transformed). Converting raw data in a meaningful format so that we can use it for model training.

- **Transforming data before training:** You manually apply preprocessing to your data before feeding it into the model.
- **Transforming data while training:** The transformation is part of the model pipeline and done automatically during training.

Unicorn Model

A single, general-purpose model that can handle many tasks across multiple domains, often with human-level performance.

Data scheme

Data schemes are basically structure and the rule of our data.

A Data Schema is like a blueprint or structure that defines:

- What kind of data you're storing
- How the data is organized
- What rules it follows

Data slicing

Data slicing helps evaluate how well your model performs on different segments of data.

A Data Slice is a subset of your dataset usually based on specific conditions or filters.

Training-serving skew

Means your input data during training differs from your input data in serving.

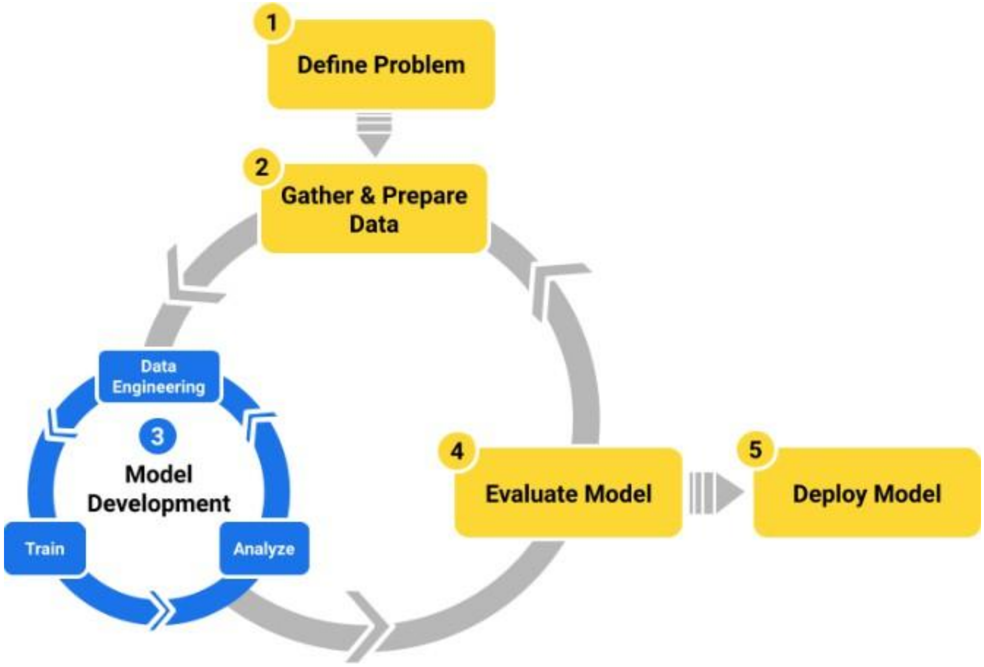
Label Leakage

Your model has access to information during training that it won't have during real use (**usually information that's related to the target/label**).

Label leakage means that your ground truth labels that you're trying to predict have inadvertently entered your training features.

Automated Machine Learning (AutoML)

Ml Flow looks like this



Automatically builds machine learning models from data preprocessing to model selection, training, and tuning with little to no human coding.

AutoML tools

AutoML tools fall into two main categories:

- **Tools that require no coding** typically take the form of web applications that let you configure and run experiments through a user interface to find the best model for your data without writing any code.
- **API and CLI tools** provide advanced automation features, but require more (sometimes significantly more) programming and ML expertise.

Bias and Its Types

Bias refers to errors or unfairness in a machine learning model caused by:

- Incorrect assumptions,
- Imbalanced data, or
- Flawed training processes.

It can lead to inaccurate, unfair, or biased predictions, especially for certain groups or data segments.

There are some types of Bias

Reporting Bias

When certain outcomes or facts are more likely to be reported than others.

Example: People only post their best moments on social media — models trained on such data may assume everyone is always happy.

Historical Bias

Bias present in data due to past social, cultural, or institutional practices.

Automation Bias

The tendency to over-rely on automated systems, even when they are wrong.

Selection Bias

When the data used is not randomly selected, it leads to unfair or incomplete results.

Coverage Bias

When some groups or regions are underrepresented or excluded in the dataset.

Non-Response Bias

It happens when certain groups choose not to respond, leading to skewed results.

Sampling Bias

When the sample does not reflect the overall population.

Group Attribution Bias

Assuming that what is true for one group member is true for the whole group.

In-Group Bias

Favoring people from your own group (race, gender, team, etc.).

Out-Group Homogeneity Bias

Believing that people in other groups are all the same, while your group is diverse.

Implicit Bias

Unconscious attitudes or stereotypes that affect decisions and actions.

Confirmation Bias

Looking for or giving more weight to information that supports your existing beliefs.

Experimenter’s Bias

When a researcher’s expectations or preferences influence how they collect or interpret data.

Mitigating bias

Once a source of bias has been identified in the training data, we can take proactive steps to mitigate its effects. There are two main strategies that machine learning (ML) engineers typically employ to remediate bias:

- Augmenting the training data.
- Adjusting the model's loss function.

Precision Recall and F1 score are not enough to check the model performance. For checking the biasness in the model we use three terms demographic parity, equality of opportunity, and counterfactual fairness.

Demographic Parity

A model satisfies demographic parity if each group (e.g., male/female) gets the same prediction rate for a positive outcome, regardless of actual outcomes or needs.

Example:

A loan approval model satisfies demographic parity if men and women are approved at the same rate, even if they have different credit scores.

Equality of Opportunity

A model satisfies equality of opportunity if qualified individuals from different groups are treated equally — i.e., equal true positive rates (TPR)

Example:

In a job screening model, if qualified female and male applicants are equally likely to be selected, then the model satisfies equality of opportunity.

Counterfactual Fairness

A model is counterfactually fair if its prediction for an individual would not change had their sensitive attribute been different all else equal.

Example:

If a female applicant is rejected for a job, the model would be unfair if she would have been accepted had she been male, keeping all other features the same.

Unsupervised Learning

Unsupervised learning is a branch of machine learning that deals with unlabeled data.

There are mainly 3 types of Algorithms which are used for Unsupervised dataset.

- **Clustering:** Clustering in unsupervised machine learning is the process of grouping unlabeled data into clusters based on their similarities.
- **Association Rule Learning:** This technique is a rule-based ML technique that finds out some very useful relations between parameters of a large data set.
- **Dimensionality Reduction:** Dimensionality reduction is the process of reducing the number of features in a dataset while preserving as much information as possible.

K-mean Clustering

K-Means Clustering is an Unsupervised Machine Learning algorithm which groups unlabeled dataset into different clusters. It is used to organize data into groups based on their similarity.

The algorithm will categorize the items into k groups or clusters of similarity. To calculate that similarity we will use the **Euclidean distance** as a measurement.

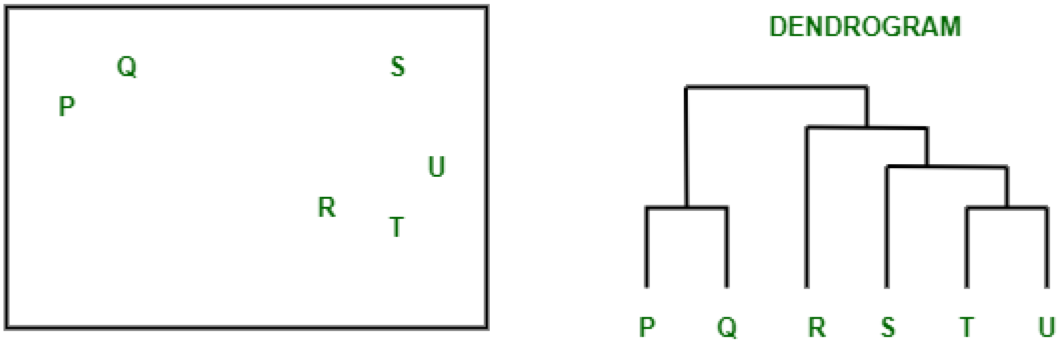
Selecting the right number of clusters is important for meaningful segmentation to do this we use **Elbow Method** for optimal value of k in KMeans which is a graphical tool used to determine the optimal number of clusters (k) in K-means.

Hierarchical Clustering

Hierarchical clustering is used to group similar data points together based on their similarity creating a hierarchy or tree-like structure. The key idea is to begin with each data point as its own separate cluster and then progressively merge or split them based on their similarity.

Dendogram

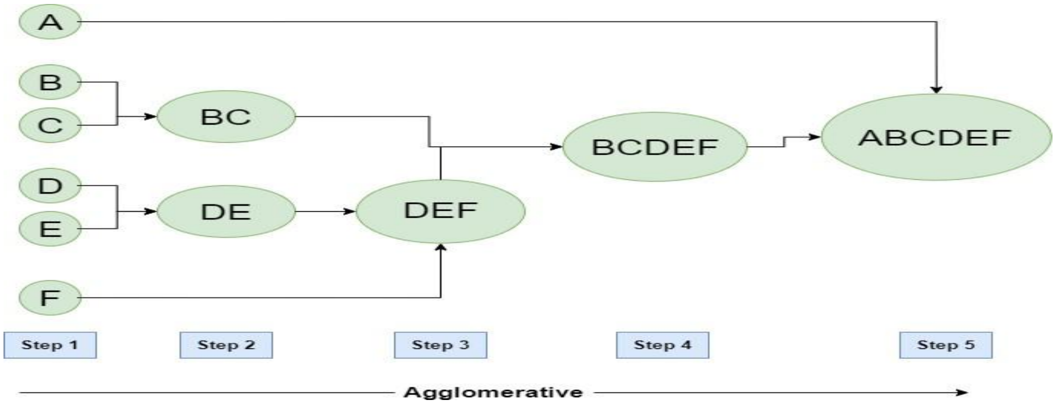
A dendrogram is like a family tree for clusters. It shows how individual data points or groups of data merge together.



Types of Hierarchical Clustering

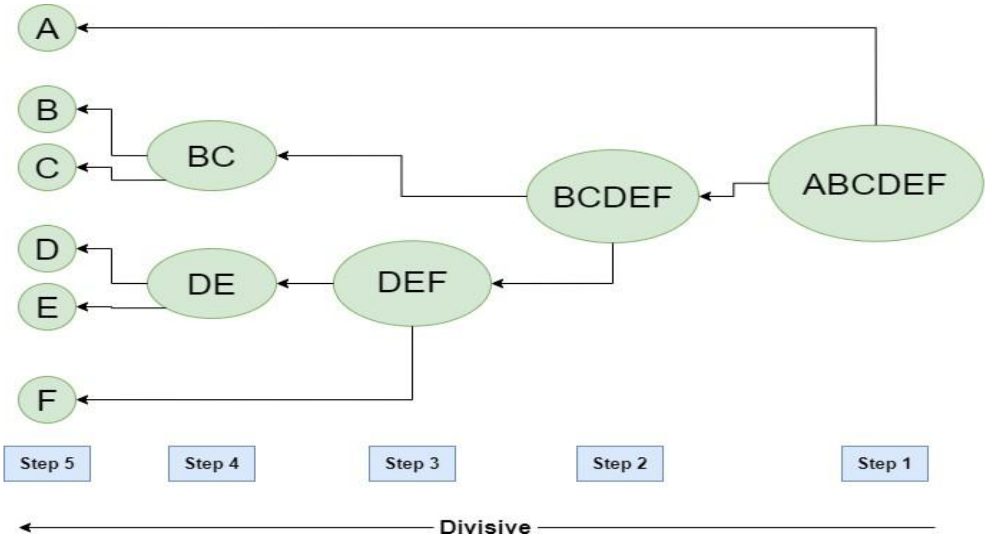
Agglomerative Clustering

It is also known as the bottom-up approach or hierarchical agglomerative clustering (HAC).

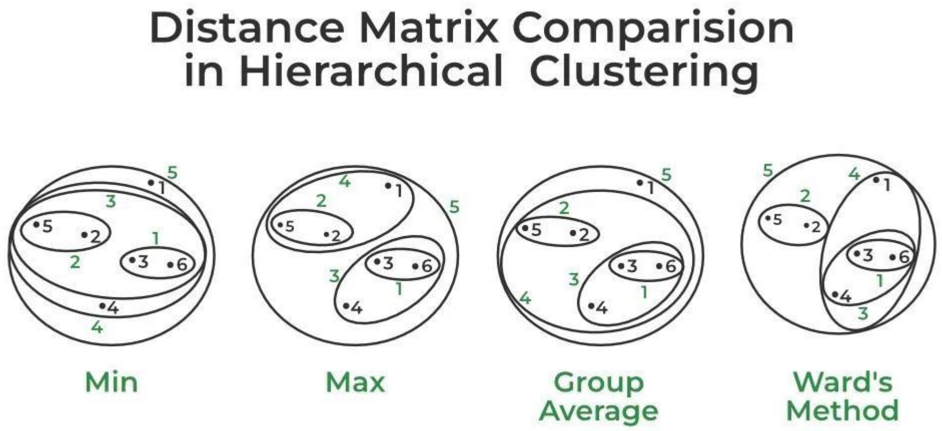


Divisive clustering

It is also known as a top-down approach.



Distance matrix that was used in hierarchical Clustering



Principal Component Analysis(PCA)

PCA (Principal Component Analysis) is a dimensionality reduction technique used in data analysis and machine learning. It helps you to reduce the number of features in a dataset while keeping the most important information.

It prioritizes the directions where the data varies the most because

more variation = more useful information.

Unsupervised Learning Evaluation Matrix

Silhouette Coefficient

Silhouette is a technique in clustering to measure the similarity of data within the cluster compared to the other cluster. The Silhouette coefficient is a numerical representation ranging from -1 to 1. Value 1 means each cluster completely differed from the others, and value -1 means all the data was assigned to the wrong cluster. 0 means there are no meaningful clusters from the data.

Formula:

Silhouette Coefficient (SC)= (b-a)/max(a,b)

Adjusted Rand Index:

The adjusted rand index measures the similarity between the true labels and the predicted labels, taking into account chance agreements. It takes values between -1 and 1, where a value close to 1 indicates that the predicted labels are identical to the true labels.

The formula for the ARI is:

ARI = (Σi,j[aij - (ai.)(aj.)/n2 - (Σi(ai.)2/n2)(Σj(aj.)2/n2)]) / (0.5[Σi(ai.)2/n2 + Σj(aj.)2/n2])

Mutual Information:

The mutual information measures the amount of information shared between the true labels and the predicted labels. It takes values between 0 and 1, where a value close to 1 indicates that the predicted labels are identical to the true labels.

DBSCAN Clustering

https://www.geeksforgeeks.org/machine-learning/dbscan-clustering-in-ml-density-based-clustering/

Why the need for DBSCAN Clustering:

In kMeans we have to first define the number of clusters and it’s also sensitive for outliers

So that’s why DBSCAN clustering need rise.

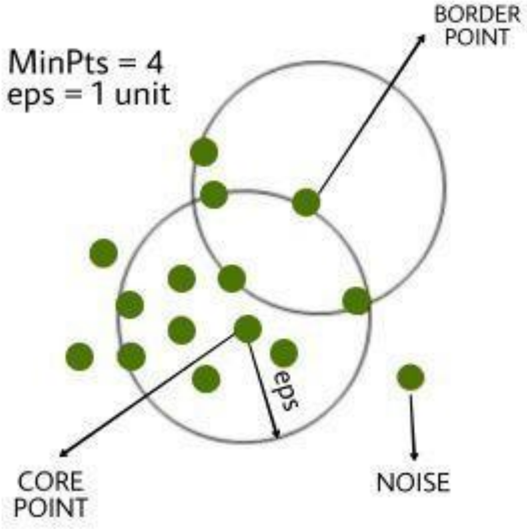
DBSCAN is a basically Density base clustering. DBSCAN stands for density based spatial clustering of applications with noise.

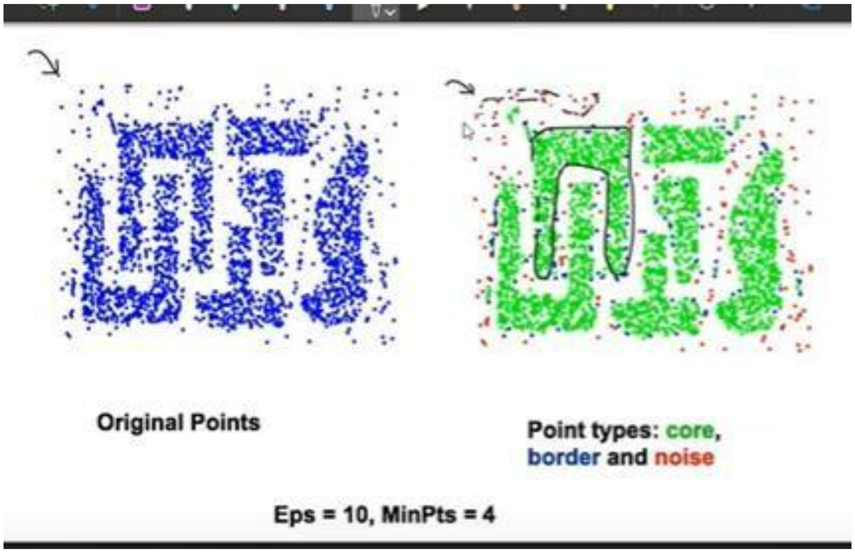
In DBSCAN we have to finetune two parameters for the best clusters, First is Min-points and Epsilon. These two parameters define the density between the datapoints.

Core points which have a sufficient number of neighbors within a specified radius (epsilon)

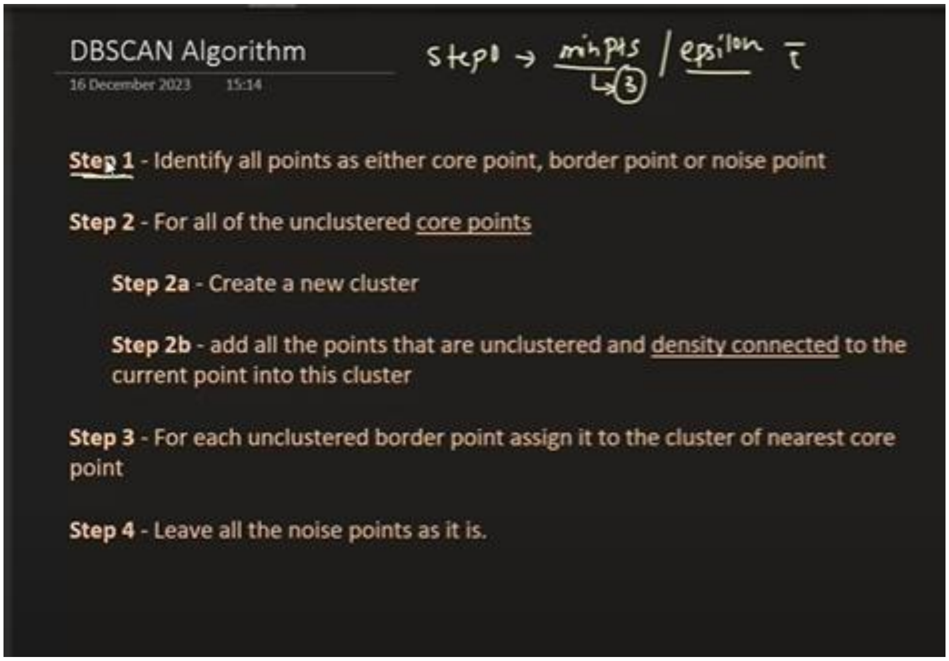
Border points which are near core points but lack enough neighbors to be core points themselves

Noise points which do not belong to any cluster.





DBSCAN Working



DBSCAN Algorithm

16 December 2023 15:14

step0 → minPts / epsilon $\bar{\varepsilon}$

↳ $\boxed{3}$

Step 1 - Identify all points as either core point, border point or noise point

Step 2 - For all of the unclustered core points

Step 2a - Create a new cluster

Step 2b - add all the points that are unclustered and density connected to the current point into this cluster

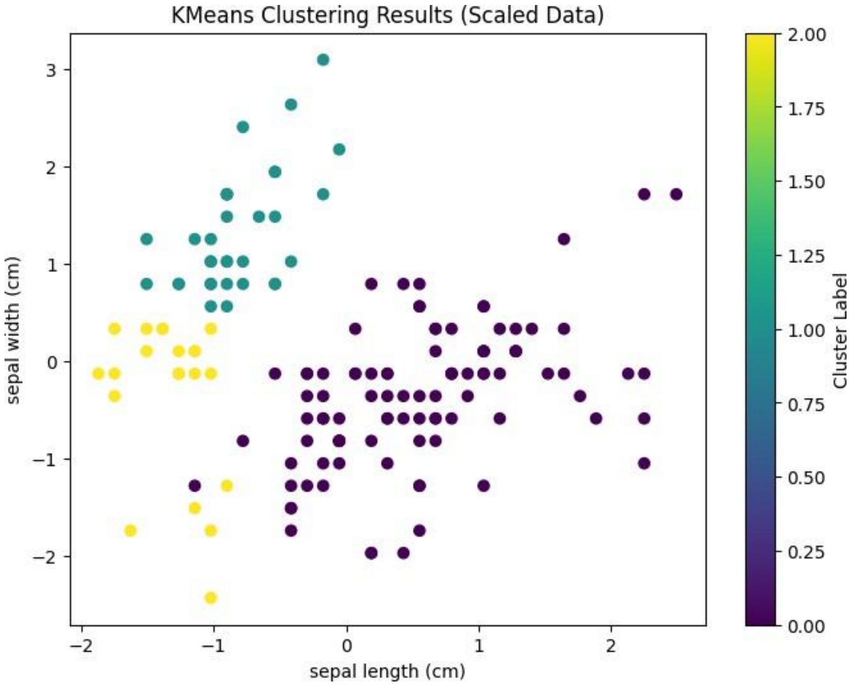
Step 3 - For each unclustered border point assign it to the cluster of nearest core point

Step 4 - Leave all the noise points as it is.

Best website for the DBSCAN visualization like how this algorithm work:
https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

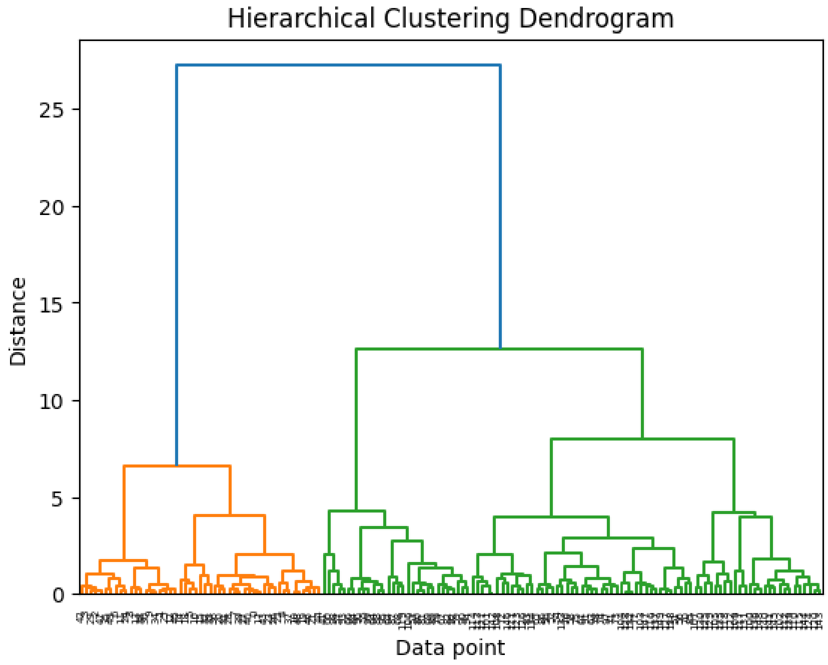
Results and Analysis

K-Mean Clustering Results



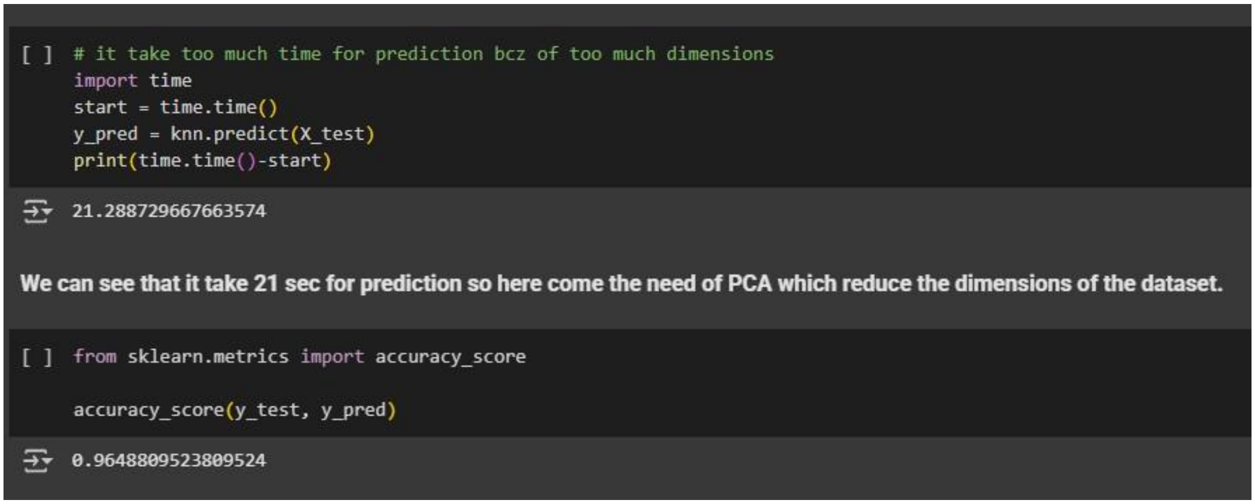
K-mean cluster for the Iris Dataset

Hierarchical Clustering



PCA on Mnist dataset

Without PCA



```
[ ] # it take too much time for prediction bcz of too much dimensions
    import time
    start = time.time()
    y_pred = knn.predict(X_test)
    print(time.time()-start)
```

```
21.288729667663574
```

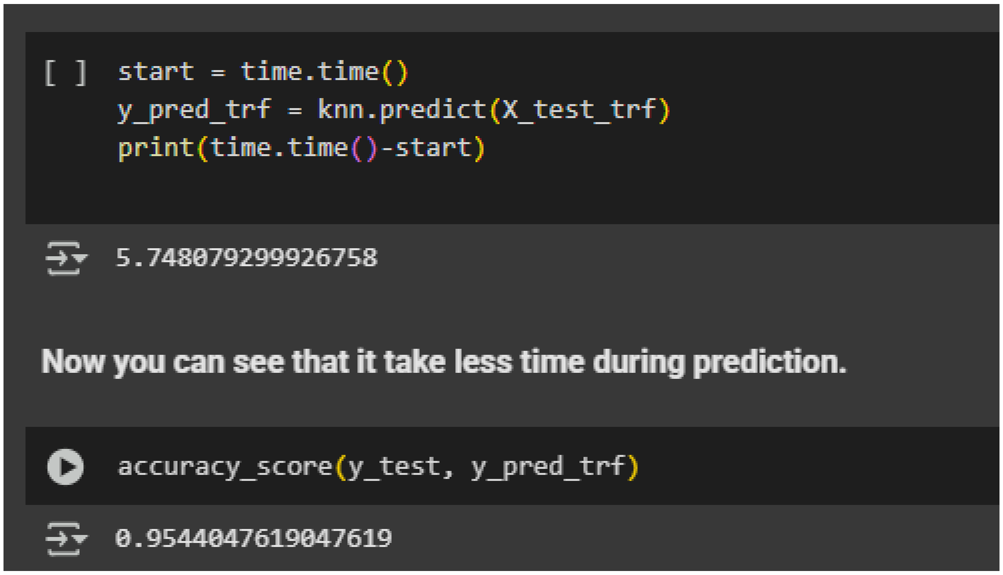
We can see that it take 21 sec for prediction so here come the need of PCA which reduce the dimensions of the dataset.

```
[ ] from sklearn.metrics import accuracy_score

    accuracy_score(y_test, y_pred)
```

```
0.9648809523809524
```

With PCA



```
[ ] start = time.time()
    y_pred_trf = knn.predict(X_test_trf)
    print(time.time()-start)
```

```
5.748079299926758
```

Now you can see that it take less time during prediction.

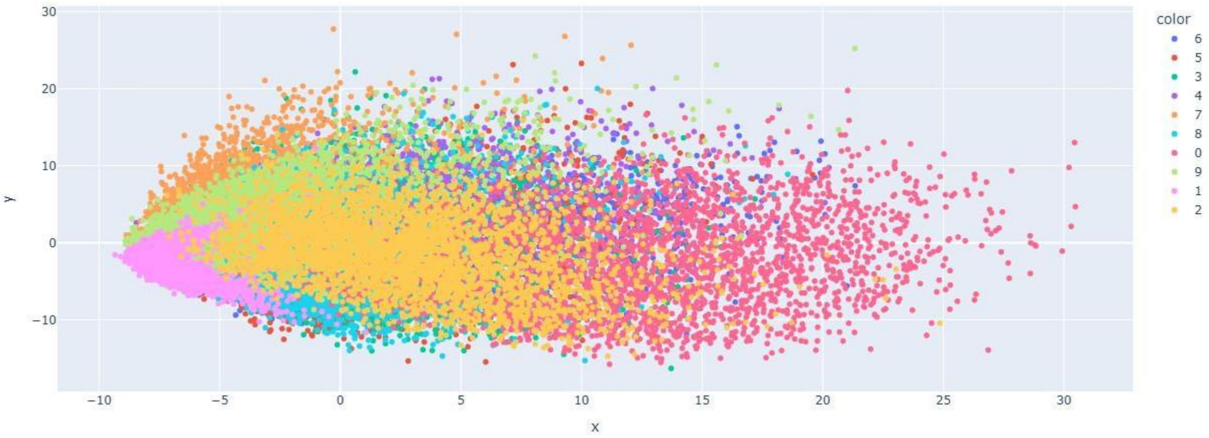
```
accuracy_score(y_test, y_pred_trf)
```

```
0.9544047619047619
```

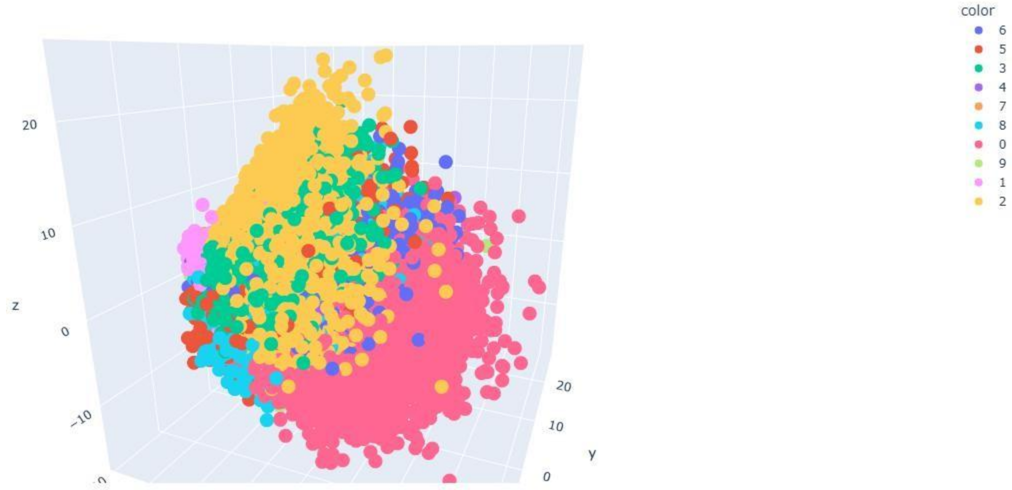
Conclusion

This week enhanced my understanding of deploying ML models in real-world scenarios and the growing relevance of fairness and automation. My self-directed exploration into unsupervised learning deepened my conceptual and practical skills in clustering and dimensionality reduction, which are critical in real-life data analysis pipelines.

Mnist dataset 2D visualization:



Mnist dataset 3D Visualization



Additional Learning

Building Multimodal AI Agents From Scratch — Apoorva Joshi, MongoDB

Learn and build understanding about transformers.

Read Attention is all you need research paper

Also explore about the auto Data analysis tool sweetviz

https://www.geeksforgeeks.org/data-analysis/sweetviz-automated-exploratory-data-analysis-eda/

Pandas Profiling: https://youtu.be/E69Lg2ZgOxg?si=DCDWr-YUMQinhOjC