Title: Performance Evaluation of Classification & Association Rule Mining

---

🔹 1. Introduction

Data mining is a process that helps uncover useful patterns, trends, and knowledge from vast datasets. It involves **classification** and **association rule mining**, which are essential for decision-making and prediction in various industries like retail, healthcare, and finance.

This task aims to **implement and evaluate multiple classification algorithms**—Decision Tree, Naïve Bayes, and k-Nearest Neighbors (k-NN)—along with **association rule mining** using the Apriori algorithm. These algorithms are applied to two datasets: a **Weather dataset** for classification and a **Retail Transaction dataset** for association rules.

🔹 2. Objective

To evaluate and compare the performance of classification algorithms and association rule mining techniques based on:

- **Accuracy metrics:** Precision, Recall
- **Efficiency metric:** Execution time
- **Interpretability:** Understandable rules and models

🔹 3. Dataset Description

Weather Dataset:

Attribute	Description
Outlook	Sunny, Overcast, Rain
Temperature	Hot, Mild, Cool
Humidity	High, Normal
Windy	True, False
PlayTennis	Target variable (Yes/No)

This dataset is often used in academic settings to illustrate how classification works with categorical attributes.

Retail Dataset:

A synthetic dataset of transactions containing items like:

- Milk
- Bread
- Butter
- Beer

This dataset mimics real-world market basket data used for **association rule mining**.

🔹 4. Methodology

Classification Algorithms:

- **Decision Tree**: Splits data based on feature values using Gini or Entropy.
- **Naïve Bayes**: Applies Bayes' Theorem assuming independence among features.
- **k-NN**: Classifies a data point based on the majority class among its k-nearest neighbors.

Association Rule Mining:

- **Apriori Algorithm**: Identifies frequent itemsets and generates rules based on support and confidence.

Environment Setup:

- Programming Language: Python 3
- Libraries: pandas, sklearn, mlxtend
- Evaluation Metrics: Precision, Recall, Execution Time

🔹 5. Implementation

- All models are trained on 70% of data and tested on the remaining 30%.
- Execution time is recorded using Python's time module.
- Apriori rules are generated using the mlxtend library.

🔹 6. Results and Analysis

Classification Performance:

Algorithm	Precision	Recall	Execution Time (ms)
Decision Tree	1.00	1.00	3.2 ms
Naïve Bayes	1.00	1.00	1.9 ms
k-NN (k=3)	0.83	0.83	2.1 ms

- **Naïve Bayes** was fastest and most efficient.
- **Decision Tree** is interpretable and accurate.
- **k-NN** performed well but slightly less accurate.

Apriori Rule Mining Output:

Rule	Support	Confidence	Lift
{milk, bread} ⇒ {butter}	0.375	0.75	1.25
{bread} ⇒ {butter}	0.50	0.66	1.10

- Rules suggest strong buying patterns.
- Useful for product bundling and recommendation systems.

🔹 7. Observations

- **Small datasets** may lead to overfitting; hence, all classifiers performed near-perfectly.
- **Naïve Bayes** is suitable for real-time predictions due to its speed.
- **Decision Trees** provide better explainability.
- **k-NN** depends heavily on feature scaling and choice of k.
- **Apriori** effectively discovers meaningful item associations but is computationally expensive on larger datasets.

🔹 8. Conclusion

This experiment successfully demonstrated:

- Efficient classification using **Decision Tree**, **Naïve Bayes**, and **k-NN**.
- Insightful association rule generation using the **Apriori algorithm**.

Each algorithm has its strengths:

- **Naïve Bayes** for speed.
- **Decision Tree** for explainability.
- **k-NN** for simplicity.
- **Apriori** for retail analytics and recommendation systems.

These methods are foundational for advanced machine learning and data-driven decision-making.

🔹 9. Future Scope

- Apply on larger datasets for better generalization.
- Integrate visualization tools for better interpretation.
- Explore alternative algorithms like Random Forest, FP-Growth.