Machine Learning Model Comparison on Wine Quality Dataset

SUMMARY:

The goal of this project was to train and compare 7 different machine learning models on the Wine dataset. The main purpose was to observe how different algorithms behave on the same dataset and to understand their strengths, weaknesses, and accuracy.

SELECTED DATASET:

The dataset used is the **Wine Quality dataset from Scikit-Learn**.
It contains chemical measurements of wine samples and labels that classify them into three categories.

WHY THE DATASET IS SELECTED >>

- It is small and easy to train without requiring GPU.
- It is commonly used in ML learning.
- It allows clear comparison between algorithms.

List of Trained Machine Learning Models:

1. Logistic Regression
2. K-Nearest Neighbors (KNN)
3. Support Vector Machine (SVM)
4. Decision Tree
5. Random Forest
6. Naive Bayes
7. Neural Network (MLP)

Why Comparing ML Models Matters:

Different algorithms work differently depending on data size, noise, and patterns.
Comparing models helps identify:

- Which model performs best
- How model complexity affects accuracy
- Which model generalizes well and avoids overfitting

This helps in selecting the **best model for real-world use**.

Methodology:

- **Dataset Preparation**
  - Loaded the Wine dataset using Scikit-Learn.
  - Standardized features using **StandardScaler** so models like SVM, KNN, and neural network perform better.

- **Train/Test Split**
  - Dataset was divided into **80% training and 20% testing**.
  - Stratified split was used to maintain class balance.

- **Why CPU Only?**
  This was a project requirement to understand normal model training performance without acceleration from GPU.

- **Why These 7 Models?**
  These models represent different categories of machine learning algorithms:
  - **Linear models**
  - **Distance-based models**
  - **Tree-based models**
  - **Probabilistic models**
  - **Ensemble models**
  - **Deep learning models**

Model	Intuition	Works Best When	Weakness	Type
Logistic Regression	Fits a line to separate classes	Linear & simple datasets	Not good with complex boundaries	Linear Model
KNN	Classifies based on nearest neighbors	Small dataset, similar samples	Slow with large data	Distance-Based
SVM	Finds best separating boundary	Clear class separation	Hard to tune	Kernel-Based
Decision Tree	Learns decision rules from data	Easy interpretability	Overfitting	Tree-Based
Random Forest	Many decision trees combined	Most real-world classification	Takes more computing time	Ensemble
Naive Bayes	Uses probability and Bayes theorem	Text and simple datasets	Assumes independence	Probabilistic
Neural Network (MLP)	Learns complex patterns using layers	Large datasets	Overkill for small datasets	Deep Learning

RESULT SUMMARY >>

Model	Accuracy
Logistic Regression	0.9722
KNN	0.9722
SVM	0.9722
Decision Tree	0.9722
Naive Bayes	0.9722
Neural Network (MLP)	0.9722
Random Forest	1.0000

Findings:

- The Random Forest model performed the best with 100% accuracy.
- All other models performed similarly with ~97% accuracy, which shows the dataset is easy to classify.
- The model that performed comparatively weaker (not bad) in flexibility is Decision Tree, because it can overfit, although accuracy still stayed high.
- The results suggest that the dataset has clear patterns and tree-based ensemble models (like Random Forest) handle feature interactions better.

Takeaway from finding >>

For this dataset, Random Forest worked best because it handled feature interactions and variability effectively. But NO single machine learning model is always the best. Model selection depends on dataset structure, size, and complexity.