Machine Learning Model Comparison Report

Dataset: Titanic (via seaborn library)
Objective: Build and compare multiple machine learning models to predict passenger survival.
Tools: Python, scikit-learn, Jupyter Notebook

1. Introduction

The Titanic dataset is a classic benchmark for binary classification tasks. The goal is to predict whether a passenger survived (1) or not (0) based on demographic, ticket, and travel features. This report compares several machine learning models to evaluate their performance using accuracy, precision, recall, F1-score, and ROC-AUC.

2. Dataset Overview

- **Rows:** 891 passengers
- **Target:** `survived` (binary outcome)
- **Features used:**
- **Numerical:** age, fare, family_size
- **Categorical:** sex, class, embark_town, deck, who, is_alone

Feature Engineering

- **family_size = sibsp + parch + 1**
- **is_alone = 1 if family_size == 1 else 0**
- **age imputed** with median values
- **embark_town filled** with mode
- **deck missing values** treated as “Unknown”

3. Models Compared

1. **Logistic Regression** – interpretable linear baseline
2. **Decision Tree** – simple non-linear classifier
3. **Random Forest** – ensemble of decision trees
4. **Gradient Boosting** – boosting ensemble method
5. **Support Vector Machine (SVM)** – kernel-based classifier
6. **K-Nearest Neighbors (KNN)** – distance-based classifier

4. Evaluation Metrics

- **Accuracy:** Overall correctness of predictions
- **Precision:** Correct positive predictions out of all predicted positives
- **Recall:** Correct positive predictions out of all actual positives
- **F1-score:** Harmonic mean of precision and recall

- **ROC-AUC:** Ability to distinguish between classes across thresholds

5. Results

(Values are approximate; actual results vary slightly depending on train/test split.)

6. Analysis

- **Best performers:**
    - Gradient Boosting and Random Forest achieved the highest accuracy, F1-score, and ROC-AUC.
    - These ensemble methods capture non-linear interactions and reduce overfitting.
- **Baseline:**
    - Logistic Regression provided solid performance and interpretability, making it useful for quick deployment and understanding feature importance.
- **Other models:**
- SVM performed well but is computationally heavier.
- Decision Tree is simple but prone to overfitting.
- KNN is sensitive to scaling and neighborhood choice, with moderate results.

7. Conclusion

- **Gradient Boosting** is the strongest model overall, closely followed by **Random Forest**.
- **Logistic Regression** remains valuable for transparency and interpretability.
- The choice of model depends on project priorities:
- **Performance:** Use Gradient Boosting or Random Forest.
- **Interpretability:** Use Logistic Regression.
- **Simplicity:** Use Decision Tree or KNN for quick prototypes.