**🧠 Pulsar Detection using Support Vector Machines (SVM)**

**1. Dataset Overview**

* **Training Set Shape: (12528, 9)**
* **Test Set Shape: (9999, 9)**
* **Target Distribution:**
  + **Class 0 (Non-Pulsar): 11,375 samples (~90.8%)**
  + **Class 1 (Pulsar): 1,153 samples (~9.2%)**
* **Class Imbalance: High imbalance observed, favoring class 0.**

**2. Missing Data Analysis**

| **Feature** | **Train Missing** | **Test Missing** |
| --- | --- | --- |
| **Excess kurtosis of integrated profile** | **1,735** | **767** |
| **Standard deviation of DM-SNR curve** | **1,178** | **524** |
| **Skewness of DM-SNR curve** | **625** | **244** |
| **target\_class (Test Set)** | **0** | **5,370** |

* **Imputation strategy: Mean Imputation**

**3. SVM Kernel Evaluation**

| **Kernel** | **Accuracy** | **Recall (Pulsar)** | **AUC** | **CV Mean Accuracy** |
| --- | --- | --- | --- | --- |
| **Linear** | **98.49%** | **81%** | **0.968** | **0.9756** |
| **Poly** | **98.12%** | **79%** | **0.961** | **0.9723** |
| **RBF** | **98.32%** | **83%** | **0.961** | **0.9767** |
| **Sigmoid** | **86.23%** | **32%** | **0.859** | **0.8710** |

**✅ Best Kernel Based on AUC: Linear**

**4. Hyperparameter Tuning (Linear Kernel)**

* **Best Parameter: C = 0.1**
* **Best CV AUC Score: 0.9702**

**5. Final Model Performance (Validation Set)**

* **Accuracy: 98.49%**
* **Precision (Pulsar): 96%**
* **Recall (Pulsar): 81%**
* **F1-Score (Pulsar): 0.88**
* **ROC-AUC Score: 0.969**

**6. Feature Importance (Linear Kernel Coefficients)**

| **Feature** | **Importance** |
| --- | --- |
| **Skewness of integrated profile** | **1.35** |
| **Excess kurtosis of integrated profile** | **0.90** |
| **Std dev of integrated profile** | **0.24** |
| **Mean of DM-SNR curve** | **0.21** |
| **Std dev of DM-SNR curve** | **0.19** |
| **Excess kurtosis of DM-SNR curve** | **0.18** |
| **Skewness of DM-SNR curve** | **0.03** |
| **Mean of integrated profile** | **0.02** |

**📌 Top Feature: Skewness of the integrated profile**

**7. Predictions on Test Set**

* **Test samples predicted with pulsar probability.**
* **Sample Output:**

| **Index** | **Predicted\_Class** | **Pulsar\_Probability** |
| --- | --- | --- |
| **0** | **0.0** | **0.0085** |
| **1** | **1.0** | **0.9752** |
| **2** | **0.0** | **0.0224** |
| **...** | **...** | **...** |

**✔️ Saved to pulsar\_predictions.csv**

**✅ Conclusion**

* **The linear kernel SVM with C=0.1 showed the best balance of accuracy, AUC, and interpretability.**
* **Imputation and standard scaling helped manage missing values and feature scaling.**
* **Despite class imbalance, the model achieved strong performance, especially in identifying the minority Pulsar class.**

**📊 Ensemble Model Evaluation Report**

**🧪 Experiment Overview**

This experiment aimed to compare the performance of two ensemble methods — **Bagging** and **Gradient Boosting** — implemented from scratch and evaluated on two popular image classification datasets: **MNIST** (handwritten digits) and **Fashion-MNIST** (clothing items). Simple statistical and distributional features were added to enhance model performance while maintaining reasonable training efficiency.

**📁 Datasets**

* **MNIST**: 10,000 training and 2,000 testing samples (28×28 grayscale images of digits 0–9)
* **Fashion-MNIST**: 10,000 training and 2,000 testing samples (28×28 grayscale images of fashion items)

Each image was flattened and enhanced with additional handcrafted features such as mean, standard deviation, dynamic range, and row/column distributional statistics (total 807 features).

**⚙️ Model Configurations**

| **Model** | **Base Estimator** | **Estimators** | **Other Parameters** |
| --- | --- | --- | --- |
| **Bagging** | DecisionTreeClassifier(max\_depth=8) | 15 | Bootstrap: True |
| **Gradient Boosting** | DecisionTreeRegressor(max\_depth=2) | 25 | Learning rate: 0.1, Subsample: 0.8 |

**📈 Results Summary**

**🟢 MNIST Dataset**

| **Metric** | **Bagging** | **Gradient Boosting** |
| --- | --- | --- |
| **Accuracy** | 0.8090 | 0.7210 |
| **MAE** | 0.6315 | 0.9840 |
| **R² Score** | 0.6684 | 0.4534 |
| **Training Time (s)** | 23.95 | 107.76 |

**Conclusion**: Bagging significantly outperformed Gradient Boosting in terms of accuracy and training time, achieving over **8%** higher accuracy with **~4.5×** faster training.

**🟠 Fashion-MNIST Dataset**

| **Metric** | **Bagging** | **Gradient Boosting** |
| --- | --- | --- |
| **Accuracy** | 0.7460 | 0.7125 |
| **MAE** | 0.7460 | 0.8655 |
| **R² Score** | 0.6518 | 0.5950 |
| **Training Time (s)** | 50.84 | 174.99 |

**Conclusion**: Bagging again outperformed Gradient Boosting across all metrics with **~3% higher accuracy** and **~3.5× faster training**.

**🔍 Analysis**

* **Bagging Pros**:
  + Fast training time.
  + Strong generalization on both datasets.
  + Robust to overfitting due to diversity of estimators.
* **Gradient Boosting Cons**:
  + Higher computational cost.
  + Slightly worse generalization.
  + More sensitive to hyperparameters and learning rate tuning.

**📌 Final Observations**

* **Best Performing Model**: **Bagging**
* **Best Dataset Performance**: MNIST with Bagging (80.9% accuracy)
* **Efficiency Recommendation**: For applications requiring fast and reasonably accurate models on image-derived tabular features, **Bagging** is the preferred method.