Classification Report: MAGIC Gamma Telescope Dataset

Introduction

This report presents the results of a classification task on the MAGIC Gamma Telescope dataset (`magic_data.csv`), aimed at distinguishing gamma rays from hadron events. Four machine learning models—Decision Tree, Naive Bayes, Random Forest, and AdaBoost—were implemented, evaluated, and compared based on accuracy, precision, recall, F1-score, and confusion matrix. The dataset was balanced, split into 70% training and 30% testing sets, and parameter tuning was performed using cross-validation for Random Forest and AdaBoost.

Methodology

Data Preprocessing

- **Dataset**: The MAGIC dataset contains 19,020 samples with 10 features and a binary class label (`g` for gamma, `h` for hadron).
- **Balancing**: The dataset was imbalanced (12,332 gamma, 6,688 hadron). Random undersampling was applied to select 6,688 gamma samples, creating a balanced dataset of 13,376 samples (6,688 per class).
- **Train-Test Split**: The balanced dataset was split into 70% training (9,363 samples) and 30% testing (4,013 samples) sets using a random seed of 42 for reproducibility.

Model Implementation

Four classifiers were applied: 1. **Decision Tree**: A simple tree-based model with default parameters (`random_state=42`). 2. **Naive Bayes**: Gaussian Naive Bayes, assuming feature independence, with default parameters. 3. **Random Forest**: An ensemble of decision trees with `n_estimators` tuned over `[50, 100, 200]` using 5-fold cross-validation. 4. **AdaBoost**: A boosting algorithm with `n_estimators` tuned over `[50, 100, 200]` using 5-fold cross-validation.

Parameter Tuning

- **Cross-Validation**: 5-fold cross-validation was used via `GridSearchCV` to select the best `n_estimators` for Random Forest and AdaBoost based on accuracy.
- **Decision Tree and Naive Bayes**: No parameter tuning was performed, as per the assignment's implied scope.

Evaluation

Models were evaluated on the test set using: - **Accuracy**: Proportion of correct predictions. - **Precision**: Proportion of true gamma predictions among all gamma predictions. - **Recall**: Proportion of true gamma instances correctly identified. - **F1-Score**: Harmonic mean of precision and recall. - **Confusion Matrix**: A 2x2 matrix showing true positives, false negatives, false positives, and true negatives for classes `g` and `h`.

Results

The performance metrics for each model on the test set are summarized below:

Decision Tree

- **Accuracy**: 0.8126
- **Precision**: 0.8038
- **Recall**: 0.8255
- **F1-Score**: 0.8145
- **Confusion Matrix**: [[1659, 350], [402, 1602]]

Naive Bayes

- **Accuracy**: 0.6783
- **Precision**: 0.6327
- **Recall**: 0.8099
- **F1-Score**: 0.7101
- **Confusion Matrix**: [[1627, 382], [909, 1095]]

Random Forest

- **Accuracy**: 0.8732
- **Precision**: 0.8625
- **Recall**: 0.8880
- **F1-Score**: 0.8751
- **Confusion Matrix**: [[1782, 227], [282, 1722]]

AdaBoost

- **Accuracy**: 0.8425
- **Precision**: 0.8293
- **Recall**: 0.8621
- **F1-Score**: 0.8454
- **Confusion Matrix**: [[1731, 278], [354, 1650]]

Model Comparison

- **Decision Tree**: Simple and interpretable but prone to overfitting, yielding moderate performance (accuracy: 0.8126). It struggles with complex patterns compared to ensemble methods.

- **Naive Bayes**: Fast and assumes feature independence, resulting in the lowest accuracy (0.6783) due to violated assumptions in the dataset's feature correlations.
- **Random Forest**: Robust ensemble method with the highest accuracy (0.8732) and balanced metrics, benefiting from parameter tuning and bagging to reduce overfitting.
- **AdaBoost**: Effective boosting approach with good performance (accuracy: 0.8425), but slightly less accurate than Random Forest and sensitive to noisy data.

Conclusion

Random Forest outperformed other models, achieving the highest accuracy (0.8732) and F1-score (0.8751) due to its ensemble nature and tuned parameters. AdaBoost followed closely, while Decision Tree offered moderate performance, and Naive Bayes was the least effective due to its simplifying assumptions. Future work could explore additional hyperparameter tuning (e.g., max_depth for Decision Tree) or feature engineering to further improve performance.