Comparative Study of Classifiers on Three Different Datasets

Anand Kumar

February 5, 2025

Contents

1 Introduction

This assignment involves working with three different datasets:

- **Dataset 1:** A medical biopsy dataset. The target variable (*Biopsy*) indicates whether a patient is `Healthy` or has `Cancer`.
- **Dataset 2:** A fetal health dataset. The target variable (*fetal_health*) can have three classes $(1, 2, 3)$.
- **Dataset 3:** A banking (marketing) dataset. The target variable (y) is binary (`yes` or `no`), indicating whether a customer subscribed to a term deposit.

The goal is to compare four classification algorithms on each dataset:

1. Decision Tree (Q1)
2. Random Forest (Q2)
3. XGBoost (Q3)
4. AdaBoost (Q4)

We evaluate each model using **5-fold cross-validation**, reporting:

- **Accuracy**
- **Precision**
- **Recall**
- **F1 Score**
- **AUC-ROC** (Area Under the ROC Curve)

Additionally, we plot:

- **ROC curves**
- **Decision boundaries** (by selecting any two features for a 2D visualization)

2 Data Preprocessing

For each dataset, we apply the following general steps:

1. **Loading and Concatenation (if needed):** For Dataset 2, for example, two partial CSV files are concatenated into a single DataFrame.
2. **Handling Missing Values:**
    - Replace non-numeric or `"?"` values with `NaN`.

- Impute `NaN` by the mean of the respective column.

3. **Feature Selection for Visualization (2D):** We select two features (e.g., `Age` vs. `Smokes (years)` in Dataset 1) for plotting the decision boundary.

4. **Train-Test Split:** Usually, a portion (20%) is separated to evaluate or plot ROC. For 5-fold CV, the entire data is systematically split.

5. **5-Fold Cross-Validation:**

   - Use `StratifiedKFold` with `n_splits=5` to preserve class distribution in each fold.

6. **Metrics Calculation:**

   - Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
   - Precision = $\frac{TP}{TP+FP}$
   - Recall = $\frac{TP}{TP+FN}$
   - $F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
   - AUC-ROC via `roc_auc_score`, often using `predict_proba`.

3 Experiments and Results

We summarize the four classifiers below (Q1–Q4). The same methodology is applied to all three datasets.

3.1 Q1: Decision Tree Classifier

Dataset 1 (Biopsy):

- We fit a `DecisionTreeClassifier` using 5-fold cross-validation.
- For each fold, we compute Accuracy, Precision, Recall, F1 Score, and AUC-ROC.
- After cross-validation, we use a separate train/test split to plot the ROC curve and to visualize decision boundaries (using two selected features).

Dataset 2 (Fetal Health):

- Multi-class classification. We use `average='weighted'` for precision/recall/F1.
- For AUC-ROC, we use `roc_auc_score(..., multi_class='ovr')`.

Dataset 3 (Banking):

- Binary classification (`yes/no`).
- Similar approach: 5-fold CV, metrics, ROC plot, and 2D decision boundary (using two numeric features, e.g., `euribor3m` vs. `duration`).

3.2 Q2: Random Forest Classifier

The procedure is the same, except we use:

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
```

We again do 5-fold CV on each dataset, collect metrics, plot ROC, and visualize 2D boundaries.

3.3 Q3: XGBoost Classifier

Using:

```
from xgboost import XGBClassifier
clf = XGBClassifier(random_state=42)
```

We replicate the same pipeline:

- 5-fold CV (stratified).
- Evaluate all metrics.
- Plot ROC curves (multi-class for Dataset 2, binary for Datasets 1 and 3).
- Plot 2D decision boundaries with selected features.

3.4 Q4: AdaBoost Classifier

Using:

```
from sklearn.ensemble import AdaBoostClassifier
clf = AdaBoostClassifier(random_state=42)
```

Again, we use the same methodology of cross-validation, metrics, ROC, and decision boundaries.

4 Illustrative Results

Although your results will vary, an example of average (5-fold) performance metrics might look like this:
(These figures are only examples. You would replace them with the actual results from your code.)

Table 1: Illustrative Cross-Validation Results

Classifier	**Dataset**	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC-ROC**
Decision Tree	Dataset 1	0.90	0.88	0.92	0.90	0.93
Decision Tree	Dataset 2	0.91	0.90	0.89	0.89	0.90 (ovr)
Decision Tree	Dataset 3	0.88	0.85	0.84	0.84	0.89
Random Forest	Dataset 1	0.93	0.91	0.95	0.93	0.96
Random Forest	Dataset 2	0.94	0.93	0.92	0.92	0.95 (ovr)
Random Forest	Dataset 3	0.90	0.88	0.87	0.87	0.91
XGBoost	Dataset 1	0.94	0.93	0.96	0.94	0.96
XGBoost	Dataset 2	0.95	0.94	0.93	0.94	0.96 (ovr)
XGBoost	Dataset 3	0.91	0.89	0.89	0.89	0.92
AdaBoost	Dataset 1	0.91	0.89	0.92	0.90	0.94
AdaBoost	Dataset 2	0.93	0.92	0.90	0.91	0.94 (ovr)
AdaBoost	Dataset 3	0.89	0.86	0.84	0.85	0.90

5 Conclusion

In this assignment, we implemented and compared four classification algorithms—Decision Tree, Random Forest, XGBoost, and AdaBoost—on three distinct datasets. We used 5-fold cross-validation to compute:

- Accuracy
- Precision
- Recall
- F1 Score
- AUC-ROC

Additionally, we plotted ROC curves and generated 2D decision boundary visualizations (by selecting two numeric features at a time). The experiments suggest that ensemble methods (Random Forest, XGBoost, AdaBoost) typically outperform a single decision tree on average. However, the ideal choice depends on hyperparameter tuning, data characteristics, and class imbalances.

6 References

- **Scikit-learn Documentation:** `https://scikit-learn.org/stable/`
- **XGBoost Documentation:** `https://xgboost.readthedocs.io/en/stable/`
- **Pandas Documentation:** `https://pandas.pydata.org/docs/`