Comprehensive Scientific Report: Ensemble Learning and Random Forest Analysis

Abstract

This report presents a comprehensive analysis of ensemble learning methods and the performance of a Random Forest classifier.Through the use of the cover type dataset and a toy dataset, we explore model training, validation, ensembling, and hyperparameter tuning. Key results, insights, and implications are documented.

1. Voting Classifier

1.1. Dataset Overview

The dataset, 'cover.csv,' contains 581,012 samples with 54 features. The target column, 'Cover_Type,' is a multi-class classification problem with seven classes.

1.2. Data Preparation

The dataset was split into training, validation, and test sets with the following proportions:
- Training: 371,847 samples (64% of the data)
- Validation: 92,962 samples (16% of the data)
- Test: 116,203 samples (20% of the data)

Feature scaling was performed using a StandardScaler to normalize the data, ensuring compatibility with algorithms like SVM, SGD, and MLP. Additionally, the target variable was adjusted to be zero-indexed for compatibility with certain libraries (e.g., XGBoost).

1.3. Model Training and Validation

Five different classifiers were trained and evaluated on the validation set. Below are the results and key details:
- **RandomForestClassifier:** Achieved a validation accuracy of **94.99%.** This ensemble method demonstrated robustness and high performance.
- **LinearSVC:** Validation accuracy: **71.23%.** Despite its simplicity, the linear SVM struggled with the complexity of the dataset.
- **SGDClassifier:** Validation accuracy: **71.50%.** Although fast and scalable, its performance was limited.
- **MLPClassifier:** Validation accuracy: **92.40%.** The neural network effectively captured non-linear patterns.
- **XGBoost:** Validation accuracy: **83.95%.** This tree-based gradient boosting method provided strong results but was outperformed by Random Forest and MLP.

1.4. Ensemble Learning

Two ensemble approaches were employed:

- **Soft Voting Classifier:** Combined the probabilities from Random Forest, XGBoost, and MLP. ***Validation accuracy: 93.78%, Test accuracy: 93.82%.***
- **Hard Voting Classifier:** Combined predictions from Random Forest, XGBoost, MLP, SVC, and SGD. ***Validation accuracy: 85.36%, Test accuracy: 85.52%.***

The results indicate that soft voting was significantly more effective than hard voting, as it leveraged probabilistic predictions to integrate model strengths.

1.5. Model Contribution Analysis

Soft Voting Contribution Analysis

In this part of the analysis, we evaluated the impact of excluding each model from the soft voting ensemble. The validation accuracies after removing individual models are as follows:

1. **Without Random Forest (rf)**:
   - Validation Accuracy: **92.35%**
   - **Impact**: Removing Random Forest caused a small decrease in accuracy. This indicates that while Random Forest contributes to the ensemble's performance, its role is not dominant.
2. **Without XGBoost (xgb)**:
   - Validation Accuracy: **92.34%**
   - **Impact**: Excluding XGBoost led to a negligible drop in accuracy. This shows that XGBoost's contribution to soft voting is limited in this setup, likely due to its lower standalone performance compared to other models.
3. **Without MLP (mlp)**:
   - Validation Accuracy: **81.94%**
   - **Impact**: Removing MLP caused a significant drop in accuracy, highlighting its critical role in the ensemble. This is expected, as MLP was one of the strongest standalone models with a validation accuracy of **92.40%**.

Key Insight: MLP is the most impactful model in the soft voting ensemble, followed by Random Forest. XGBoost plays a minor role and could potentially be excluded without major performance loss.

Hard Voting Contribution Analysis

The validation accuracies after excluding individual models from the hard voting ensemble are as follows:

1. **Without Random Forest (rf)**:
   - Validation Accuracy: **81.48%**

- **Impact**: Removing Random Forest caused a notable drop in accuracy, indicating its importance in the hard voting ensemble.

2. **Without XGBoost (xgb)**:
   - Validation Accuracy: **49.25%**
   - **Impact**: Excluding XGBoost significantly reduced accuracy, suggesting that while it had lower individual performance in soft voting, it plays a crucial role in hard voting. This may be due to its consistency in predictions.
3. **Without MLP (mlp)**:
   - Validation Accuracy: **49.27%**
   - **Impact**: Similar to XGBoost, removing MLP caused a dramatic drop in accuracy. MLP is vital for maintaining the performance of the hard voting ensemble.
4. **Without SVC (svc)**:
   - Validation Accuracy: **87.60%**
   - **Impact**: Excluding SVC resulted in a slight improvement in accuracy, indicating that SVC's lower standalone performance negatively impacted the ensemble.
5. **Without SGD (sgd)**:
   - Validation Accuracy: **87.56%**
   - **Impact**: Similar to SVC, removing SGD slightly improved accuracy, further supporting the observation that weaker models can hurt hard voting performance.

Key Insight: For hard voting, Random Forest, XGBoost, and MLP are essential contributors, while SVC and SGD add noise and reduce overall performance. Excluding these weaker models leads to better ensemble accuracy.

2. Random Forest

2.1. Load Dataset

The dataset used in this analysis, `data.csv`, contains:

- **15,000 samples** with two features (`x1` and `x2`).
- A binary target variable `z`.

Before proceeding, the target variable was scaled to the range [0, 1] for consistency using the formula:

```
python
Copy code
data['z'] = data['z'].astype(float) / 100
```

2.2. Prepare Dataset

The dataset was split into training and test sets using an 80-20 ratio. The split ensures that 80% of the data is available for model training, while 20% is reserved for evaluating the model's generalization performance.

2.3. Modeling

2.3.1. Hyperparameter Tuning

A `DecisionTreeClassifier` was optimized using GridSearchCV with the following parameter grid:

- `max_depth`: [None, 5, 10, 15, 20]
- `min_samples_split`: [2, 5, 10]
- `min_samples_leaf`: [1, 2, 4]

The search aimed to identify the best combination of hyperparameters to maximize accuracy. The optimal parameters were:

- **max_depth**: 5
- **min_samples_split**: 2
- **min_samples_leaf**: 2

2.3.2. Train the Best Model

Do you need to train the model on the whole training set? Yes. Training the model on the full training set is necessary to maximize its learning potential, especially for small datasets like this. The model was trained with the identified optimal hyperparameters, and its accuracy on the test set was:

- **Test Accuracy**: **85.57%**

This performance demonstrates that the model generalizes well to unseen data.

2.3.3. Training Trees on Subsets

1,200 subsets of 100 samples each were generated using `ShuffleSplit`. A decision tree was trained on each subset using the previously determined best hyperparameters. The test accuracy of each individual tree was evaluated.

- **Average Accuracy of Individual Trees**: **79.57%**

Did you get lower or higher accuracy? Why? The accuracy of individual trees was lower than the single tree trained on the full training set. This is because each subset contained only 100 samples, limiting the diversity and patterns available for learning, resulting in underfitting that observed in individual trees .

2.3.4. Ensemble of Trees

The predictions from all 1,200 trees were aggregated using majority voting (mode). For each instance in the test set, the most frequent prediction across the 1,200 trees was chosen as the final prediction.

- **Ensemble Test Accuracy**: **85.70%**

Did you get lower or higher accuracy? Why? The ensemble accuracy was slightly higher than the single optimized tree's accuracy (85.57%). The improvement in accuracy observed when using the ensemble of 1,200 trees can be explained through the **Law of Large Numbers** and **Wisdom of Crowds**:

1. **Law of Large Numbers**:
    - This theorem states that as the number of independent observations (or models, in this case) increases, the average of these observations approaches the expected value.
    - In this context:
        - Individual trees are weak learners with high variance.
        - By aggregating predictions from 1,200 trees through majority voting (mode), the ensemble reduces the impact of any single tree's errors, converging on a more accurate prediction.
    - The ensemble essentially "averages out" the bias introduced by smaller subsets, leading to improved overall accuracy.