CSCI 4587 – Machine Learning

Project #1: Classifiers and Ensemble Learning

Brandon Vo

**Introduction**

­­The following report is an analysis of various classifiers and an ensemble learning method of stacking. The procedure consists of a comprehensive evaluation of machine learning classification algorithms, namely Extra Tree Classifier (ETC), Bagging, Decision Tree Classifier (DTC), Logistic Regression (LR), Support Vector Classifier (SVC), and k-Nearest Neighbor (kNN), utilizing the widely recognized IRIS dataset.

This dataset, comprising 150 samples, four distinct input features, and three unique output classes, serves as a benchmark for assessing the performance of these classifiers. To ensure a robust and unbiased evaluation, a 10-fold cross-validation (10 FCV) method is employed, which systematically partitions the dataset into training and testing subsets.

In addition to comparing the performance of the individual classifiers, this report also ventures into the realm of ensemble learning, specifically the stacking method. Stacking stands out as an advanced technique that strategically combines multiple classifiers to enhance prediction accuracy and stability. In this context, two different ensemble classifiers are constructed, each consisting of a base layer with three base classifiers and a meta-layer with a single classifier. These ensemble models are then meticulously evaluated using the same performance metrics applied to the individual classifiers.

To provide a comprehensive understanding of the classifiers' performance, a variety of metrics including accuracy, balanced accuracy, Matthews Correlation Coefficient, Sensitivity, Specificity, F1-score, and confusion matrix are computed and presented. This multifaceted approach to performance evaluation ensures a holistic understanding of each classifier’s strengths and weaknesses, facilitating an informed selection of the most suitable model for specific applications.

**Results**

Table 1) 10 FCV Classifier Average Results (Normalized %)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Classifier | Average Accuracy | Standard Deviation | Balanced Acc. | MCC | Sensitivity | Specificity | F1 |
| Extra Tree | 0.9333 | 0.0789 | 0.9467 | 0.9202 | 0.9.400 | 0.97 | 0.9315 |
| Bagging | 0.9600 | 0.0533 | 0.9533 | 0.9301 | 0.9600 | 0.9800 | 0.9636 |
| Decision Tree | 0.9467 | 0.0499 | 0.9467 | 0.9200 | 0.9400 | 0.9700 | 0.9450 |
| Logistic Regression | 0.9733 | 0.0442 | 0.9600 | 0.9400 | 0.9600 | 0.9850 | 0.9650 |
| SVC | 0.9667 | 0.0447 | 0.9667 | 0.9667 | 0.9501 | 0.9550 | 0.9850 |
| kNN | 0.9733 | 0.0442 | 0.9467 | 0.9202 | 0.9459 | 0.9750 | 0.9420 |

Figure 1) Confusion Matrices for ETC (top left) through kNN (bottom right)

In general, we noticed that the overall best was split quite closely between logistic regression, SVC, and kNN due to their slightly higher average accuracy and slightly lower standard deviation than the other 3 classifiers. I proceeded to run this training multiple times and found that there are some instances where all classifiers have around 95% accuracy, so they can be roughly equal, but there are also some instances where latter 3 perform slightly better on average.

In general, Extra Tree is least consistent and can range from 90% to upwards of 96% accuracy. SVC and Logistic Regression were the most consistently accurate.

Table 2) Average Metrics for Stacked Classifier 1

|  |  |
| --- | --- |
| Metric | Value (Normalized %) |
| Accuracy | 0.9333 |
| Balanced Accuracy | 0.9259 |
| Matthews Correlation Coefficient | 0.9051 |
| Sensitivity | 0.9259 |
| Specificity | 0.9683 |
| F-1 Score | 0.9250 |

Figure 2) Confusion Matrix for Stacked Classifier 1

This stacked classifier was developed from the three base classifiers: Extra Tree, Bagging, and Decision Tree, with a meta classifier of Logistic Regression. The values for the results are all quite good, ranging above 90%, but not as good as the original classifiers or as good as the 2nd stacking classifiers. This could potentially be just a bad mix of classifiers or just a bad mix of data due to the small sample size. I ran this multiple times and was able to get ranges from ~90% metrics anywhere to 100% metrics, depending on the test/train split I was using.

Table 3) Average Metrics for Stacked Classifier 2

|  |  |
| --- | --- |
| Metric | Value (Normalized %) |
| Accuracy | 0.9667 |
| Balanced Accuracy | 0.9630 |
| Matthews Correlation Coefficient | 0.9511 |
| Sensitivity | 0.9630 |
| Specificity | 0.9841 |
| F-1 Score | 0.9628 |

Fig 3) Confusion Matrix for Stacked Classifier 2

This stacked classifier was developed from the three base classifiers: Logistic Regression, SVC, and kNN with a meta classifier of Decision Trees. The values for the results are all quite good, ranging above 95%. Again, this model does make sense. The iris dataset is a very simple dataset that has relatively simple patterns. Stacking to a more complex model with various classifiers that already have a very high accuracy would be reasonable to develop and exact fit with such a simple dataset. Stacking is a relatively advanced ensemble technique, and it's very possible for it to achieve this higher accuracy on the dataset. These classifiers also already have good metrics as well, so stacking them would probably give similarly good results. I ran this multiple times and was able to get ranges from ~93% metrics anywhere to 100% metrics, depending on the test/train split I was using.

**Conclusion and Remarks**

Based off the results of this project, we can see that machine learning methods of classification are powerful tools. The results of this short study revealed a surprising uniformity in performance across all the tested models. Each classifier demonstrated a commendable ability to accurately predict the output classes, with metrics such as accuracy, balanced accuracy, Matthews Correlation Coefficient, Sensitivity, Specificity, F1-score, and confusion matrix showcasing their efficacy. This uniformity in performance underscores the versatility and reliability of these classifiers, particularly when applied to well-behaved datasets like IRIS. Some of the classifiers performed slightly better at a relatively marginal value, but statistical analysis would have to be performed to deem the performance significant. The stacking ensemble classifiers, constructed with a base layer of three classifiers and a meta-layer of a single classifier, also mirrored this trend of consistent performance.

The ensemble approach, known for its potential to enhance predictive performance and stability, did not exhibit a significant advantage in this particular context. This could be attributed to the simplicity and relatively low dimensionality of the IRIS dataset, where the individual classifiers are already capable of achieving high accuracy.

In general, it could be wise to think that there would be some difference in performance if a larger data set is used, since 50 samples with 3 classes and only a small split portion for testing has potential errors and flaws. A data set of 1,000 samples in each class might be suitable for performances that can better show the advantages and disadvantages of each classification algorithm used.