Lab 2 Report

A Preprint

Daniele cecca
Artificial Intelligence for Science and Technology
Milano Bicocca University
Supervised Learning

March 22, 2024

1 Introduction

The aim of this study is to implement a stacked classifier using five level-1 classifiers and compare its performance under different conditions. Specifically, we examined how the meta-classifier's performance varies when trained on Predictions instead of Scores, and when the training split is not performed, using the same data to train both level-1 classifiers and the meta-classifier.

2 Stacking Classifier

A stacking classifier is an ensemble method where the output from multiple classifiers is passed as an input to a meta-classifier for the task of the final classification. The individual classification models are trained based on the training data, then the meta-classifier is fitted based on the outputs (meta-features) of the individual classification models.

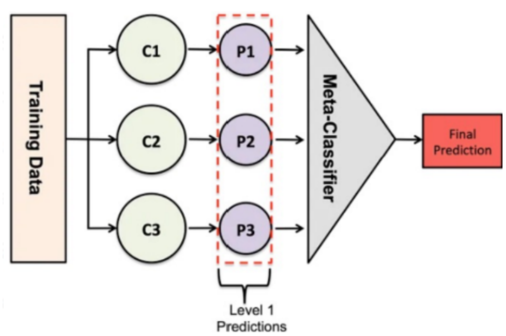

Figure 1: Stacking Classifier

2.1 Level-1 Classifier

- **SVM with Gaussian Kernel:**
  - **Parameters:** KernelFunction: 'gaussian', KernelScale: 5
  - **Description:** Constructs hyperplanes in a high-dimensional space for classification using a Gaussian kernel.
- **SVM with Polynomial Kernel:**
  - **Parameters:** KernelFunction: 'polynomial', KernelScale: 10
  - **Description:** Finds optimal hyperplanes using a polynomial kernel function.

- **Decision Tree:**
  - **Parameters:** SplitCriterion: 'gdi', MaxNumSplits: 20
  - **Description:** Partitions feature space into regions using the Gini diversity index as a splitting criterion, with a maximum of 20 splits.
- **Naive Bayes:**
  - **Parameters:** Default
  - **Description:** A probabilistic classifier based on Bayes' theorem with the assumption of feature independence.
- **Random Forest:**
  - **Parameters:** Default
  - **Description:** Constructs multiple decision trees during training and outputs the mode of the classes for classification, improving predictive accuracy.

2.2 Meta-Classifier

As the meta-classifier, we used Bagging (bootstrap aggregating), which is an ensemble method that involves training multiple models independently on random subsets of the data and aggregating their predictions through voting or averaging.

3 Methodology

3.1 Data

We began by loading the dataset and visualizing it. Stratified sampling was performed to create two folds from the training set.

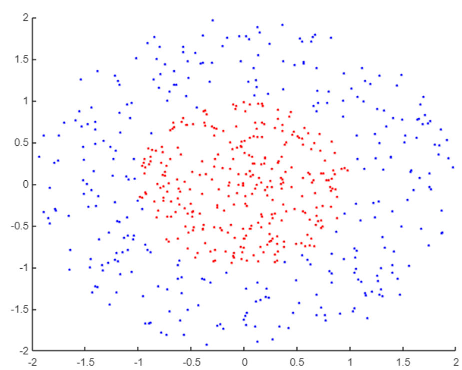

Figure 2: Data

3.2 Task 1: Predictions vs Scores

We trained the level-1 classifier on the first fold and then made predictions on the latter. We obtained outputs for both Predictions and Scores, each composed of 5 columns, one for each classifier. Prediction represents the predicted label, while Score represents the confidence level of the model in its predictions.

Then two stacked classifiers were trained. The first one used training data Scores, while the second one used Predictions. Thus, we used as predictors the output of the classifiers on the previous layer.

In the end, we evaluated the performance of all classifiers, including the stacked classifiers, using accuracy metrics on the test set, and obtained the following results:

Table 1: Accuracy Task 1

Classifier	Accuracy
SVM Gaussian	0.8683
SVM Polynomial	0.6250
Decision Tree	0.9483
Naive Bayes	0.9783
Random Forest	0.9533
Stacked Classifier (Scores)	0.9900
Stacked Classifier (Predictions)	0.9700

3.3 Task 2: Training on the same data

We trained the level-1 classifier on all training data and then made predictions on the training data as well, obtaining both Predictions and Scores as outputs.

We repeated the same steps above, training the meta-classifier on Predictions and Scores. After that, we reevaluated our models on the test data and obtained:

Table 2: Accuracy Task 2

Classifier	Accuracy
SVM Gaussian	0.9000
SVM Polynomial	0.6333
Decision Tree	0.9667
Naive Bayes	0.9917
Random Forest	0.9683
Stacked Classifier (Scores)	0.9700
Stacked Classifier (Predictions)	0.9683

4 Conclusion

By observing these results, we can conclude that:

- The meta-classifier trained on Scores outperformed the one trained on Predictions in both scenarios. This indicates that incorporating confidence scores directly may capture underlying patterns more effectively than utilizing classification predicted classes .
- Utilizing the same data for training both the level-1 classifiers and the meta-classifier yielded slightly lower accuracies compared to using separate training folds. This suggests that a distinct training split helps in generalization and avoids overfitting.