**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.

**STACKED 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.

#### 

#### 

#### **Level-1 Classifier**

#### In our case we choose as level one classifiers :

1. SVM with Gaussian Kernel:
   1. Parameters: KernelFunction: 'gaussian', KernelScale: 5
   2. Description: Constructs hyperplanes in a high-dimensional space for classification using a Gaussian kernel.
2. SVM with Polynomial Kernel:
   1. Parameters: KernelFunction: 'polynomial', KernelScale: 10
   2. Description: Finds optimal hyperplanes using a polynomial kernel function.
3. Decision Tree:
   1. Parameters: SplitCriterion: 'gdi', MaxNumSplits: 20
   2. Description: Partitions feature space into regions using the Gini diversity index as a splitting criterion, with a maximum of 20 splits.
4. Naive Bayes:
   1. Parameters: None explicitly specified.
   2. Description: A probabilistic classifier based on Bayes' theorem with the assumption of feature independence.
5. Random Forest:
   1. Parameters: None
   2. Description: Constructs multiple decision trees during training and outputs the mode of the classes for classification, improving predictive accuracy.

**Meta-Classifier**

As meta classifier instead 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.

### **Methodology**

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

**TASK1**

We trained the 1-level classifier on the first fold and then we have made predictions on the latter. And we obtain as output Predicitions and Scores

They are both composed of 5 columns, one for classiers.

Prediction represents the label predicted

Score represents the confidence level of the of the model in its predictions.

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

At the end we evaluated the performance of all classifiers, including the stacked classifiers, using accuracy metrics on the test set.And we have obtained the following results:

Table Accuracy task 1

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

**TASK 2**

We trained the 1-level classifier on all training data and then we have made predictions on the training data as well.We always obtain as outputs Predicitions and Scores.

Then we repeat the same steps above , thus we train meta classifier on Prediction and scores.

After that we reevaluate our models on test data and we obtain:

Table Accuracy task 2

| SVM Gaussian | SVM Polynomial | Decision Tree | Naive Bayes | Random Forest | Stacked  Classifier Scores | Stacked Classifier  Predictions |
| --- | --- | --- | --- | --- | --- | --- |
| 0.900 | 0.6333 | 0.9667 | 0.9917 | 0.9683 | 0.9700 | 0.9683 |

**CONCLUSION**

By observing those results we can say that:

* The meta-classifier trained on predictions outperformed the one trained on scores in both scenarios. This indicates that incorporating predicted classes directly may capture underlying patterns more effectively than utilizing classification confidence scores.
* 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 avoid overfitting