The table below shows comparison between the different algorithms in accuracy, and training (fit method) and prediction (score method) time.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classifier | Digits Dataset | | | Time-series Dataset | | |
| Accuracy | Running Time(ms) | | Accuracy | Running Time(ms) | |
| Training | Prediction | Training | Prediction |
| Perceptron | 95.10% | 11.47 | 1.49 | 23.08% | 35.93 | 2.99 |
| SVM-SVC | 99.89% | 166.18 | 138.24 | 98.71% | 749.12 | 989.16 |
| SVM-LinearSVC | 98.94% | 157.22 | 0.99 | 26.01% | 6318.27 | 1.49 |
| 1SVM-RBF | 97.88% | 494.58 | 227.57 | 96.4% | 2247.36 | 137.58 |
| Decision Tree | 100% | 21.46 | 0.99 | 100% | 17.96 | 1.51 |
| KNN | 100% | 5.48 | 35.43 | 100% | 23.47 | 40.92 |
| Logistic Regression | 100% | 127.76 | 1.99 | 55.12% | 620.83 | 2.98 |

Pruning the decision tree:

1. Optimizing the tree structure using weight-based pre-pruning criterion such as “min\_weight\_fraction\_leaf “, which ensure that leaf nodes contain at least a fraction of the overall sum of the sample weights.

if isinstance(self.min\_samples\_leaf, (numbers.Integral, np.integer)):

if not 1 <= self.min\_samples\_leaf:

raise ValueError("min\_samples\_leaf must be at least 1 "

"or in (0, 0.5], got %s"

% self.min\_samples\_leaf)

min\_samples\_leaf = self.min\_samples\_leaf

else: # float

if not 0. < self.min\_samples\_leaf <= 0.5:

raise ValueError("min\_samples\_leaf must be at least 1 "

"or in (0, 0.5], got %s"

% self.min\_samples\_leaf)

min\_samples\_leaf = int(ceil(self.min\_samples\_leaf \* n\_samples))

1. Post-pruning is done by removing a rule’s precondition if the accuracy of the rule improves without it.

|  |
| --- |
|  |
|  | if max\_leaf\_nodes < 0: |
|  | builder = DepthFirstTreeBuilder(splitter, min\_samples\_split, |
|  | min\_samples\_leaf, |
|  | min\_weight\_leaf, |
|  | max\_depth, |
|  | self.min\_impurity\_decrease, |
|  | min\_impurity\_split) |
|  | else: |
|  | builder = BestFirstTreeBuilder(splitter, min\_samples\_split, |
|  | min\_samples\_leaf, |
|  | min\_weight\_leaf, |
|  | max\_depth, |
|  | max\_leaf\_nodes, |
|  | self.min\_impurity\_decrease, |
|  | min\_impurity\_split) |