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))
```

2- 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)
```