Introduction to AI Coursework

Runze Yuan 2217498

May 20, 2023

1 Introduction

1.1 Question

For a Combined Cycle Power Plant, use its hourly average ambient variables to predict the net hourly electrical energy output.

To be specific, the task is to use four input value to predict one output value.

1.2 Which kind of algorithms to use and why

Use **Regression** for the task.

Reasons:

- The aim of regression algorithms is to capture the relationship between inputs and outputs, which is what the task asks for (to predict energy output with four ambient value).
- Regression algorithms are capable of predicting future or unseen data.

2 Methods

2.1 Algorithms

In this report I will show results with **K Neighbors Regressor** and **Decision Tree Regressor**.

- KNR: Find K nearest neighbors in the feature space for the input vector and use the output value of the neighbors to calculate the predicted output.
- DTR: DTR builds a decision tree by recursively splitting points into groups based on the result of separation.

2.2 Metrics

The Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) are selected as performance metrics for the algorithm. These are common metrics for regression algorithms.

- MSE and MAE are metrics that reflects the difference between the predicted result and ground-truth, the lower the better.
- R^2 reflects the goodness of fitting of the algorithm, and a value close to 1 means a good fit.

2.3 Baseline

For the baseline model, use the dummy model in the scikit learn.

Set all hyperparameters of the dummy to the default value, which means set "parameters" to mean, and both "constants" "quantile" to None.

With these parameters, the dummy model is not an actual regressor and would always output the mean value of the given training output data.

2.4 Hyperparameters

Hyperparameters are parameters that are set prior for models and not changed in the learning process. These parameters control the characteristic of the model and allow adjustments for the user by tuning the hyperparameters.

- **KNR**: In this report I will try to find the optimal **n_neighbours**, **weights**, and **p**.
  - n_neighbors: how many nearest neighbors the algorithm use to predict the output value.
  - weights: changes the weights used in the output generating.
    - uniform: all nearest neighbors have the same weight.
    - distance: use distance as weights for the neighbors, distant neighbors have lower weights.
  - p: changes the type of distance used in the algorithm
    - p=1: apply Manhattan distance for distance calculation.
    - p=2: apply Euclidean distance for distance calculation.
  - other parameters applied but not discussed in this report:
    - algorithm: "auto"
    - leaf_size: 30
    - metric: "minkowski"
    - metric_params: "None"
    - n_jobs: None
- **DTR**: In this report I will try to find the optimal **max_depth**, **min_samples_leaf**, and **splitter**
  - max_depth: the maximum depth of the tree.
  - min_samples_leaf: the minimum sample amount for a node to be a leaf node.
  - splitter: changes the strategy of splitting samples into different groups in nodes.
    - best: evaluate all possible splits and apply the best one which have the best performance.
    - random: apply the best split among a random generated strategy for randomly selected subsets of the features.
  - other parameters applied but not discussed in this report:
    - min_samples_leaf: 1
    - min_weight_fraction_leaf: 0.0
    - max_features: None
    - random_state: 42
    - max_leaf_nodes: None
    - min_impurity_decrease: 0.0
    - ccp_alpha: 0.0

2.5 Hyperparameters searching

Use GridSearchCV provided in scikit-learn to search for optimal hyperparameters.

- Use 80% of the provided data as training test in this step, and 20% for later result verification.
- Use 10-fold cross-validation to make sure the hyperparameters show their generalization ability.
- Use MSE as metric for hyperparameters performance, because MSE could better illustrate the difference between predicted output and ground-truth.

2.5.1 K Neighbors Regressor

- Preprocess data: Standardize features by removing the mean and scaling to unit variance. Standardizing could make the influence of all features equal.
- Search for optimal hyperparameters: Candidates: n_neighbors = 1∼20, weights = 'uniform', 'distance', p = 1,2

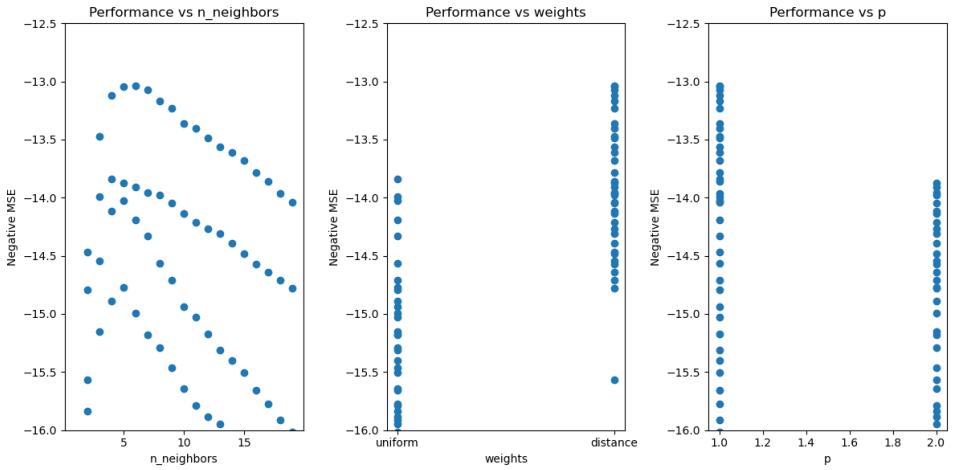

Figure 1: K Neighbors Regressor hyperparameters searching results. The vertical axis is the negative MSE, with higher values representing better performance.

Searching results:

The performance reaches its peak when the value of n_neighbors is around 5. Weighting based on distance generally yields better overall performance, and p=1 (Manhattan distance) exhibiting better overall performance.

The optimal hyperparameters found in the searching are:

- n_neighbors = 7
- p = 1
- weights = 'distance'

and this set of hyperparameters has an MSE of 13.16 on the training data.

2.5.2 Decision Tree Regressor

- Search for optimal hyperparameters: Candidates: max_depth = 5∼20, min_samples_leaf = 1∼30, splitter = 'best', 'random'.

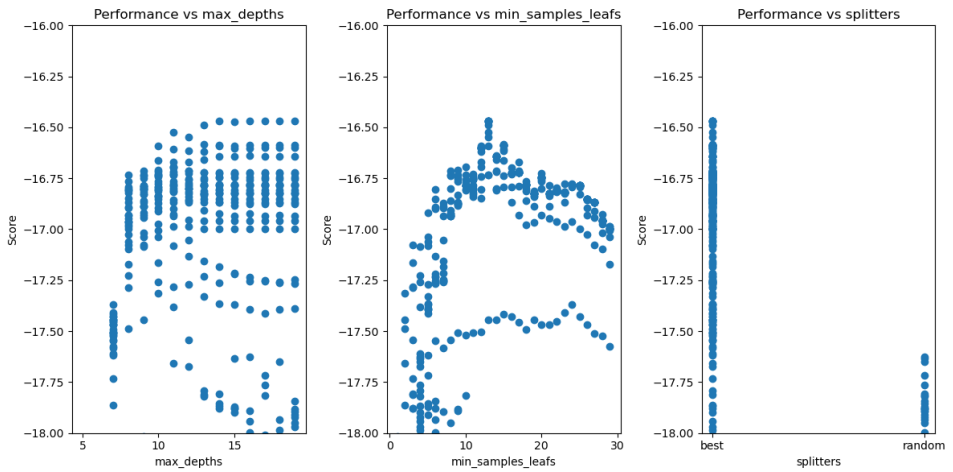

Figure 2: Decision Tree Regressor hyperparameters searching results. The vertical axis is the negative MSE, with higher values representing better performance.

Searching results

The performance reaches its peak when the value of max_depth is around 15, min_samples_leaf is between 10 and 20. Splitter based on the best splitting yields better overall performance.

The optimal hyperparameters found in the searching are:

- max_depth = 14
- min_samples_leaf = 13
- splitter = "best"

and this set of hyperparameters has an MSE of 16.40 on the training data.

3 Result and Analysis

3.1 Results

Like previously mentioned in Section 2.5, the results are performances on the test data which is not used in hyperparameter searching.

Hyperparameters applied are searching results in the previous section.

Algorithms	**MSE**	**MAE**	R^2
Dummy	287.79	14.81	≈ 0
K Neighbors	12.41	2.52	0.96
Decision Tree	16.22	2.98	0.94

3.2 Analysis

3.2.1 K Neighbors Regressor

- **n_neighbors**: When n_neighbors is too large, it can lead to a difficulty to capture local pattern due to an excessive number of neighbors. On the other hand, when the value of n_neighbors is too small, it becomes challenging to capture any patterns effectively (no enough neighbors). In this context, selecting a value of 7 is an optimal trade-off point that strikes a balance.
- **p**: The performance of Manhattan distance and Euclidean distance primarily depends on the dataset. In this case, Manhattan distance (p = 1) exhibits better performance. This may be caused by the patterns in the provided features, like the features have significant difference between each other, or the features are placed on the border of a high dimensional space.
- **weights**: It is normal for weights based on distance have better performance. Distance weighting allows for higher weights to be assigned to more similar points, and suppress the influence of noises in the training data.

3.2.2 Decision Tree Regressor

- **max_depth**: An excessively deep decision tree can result in the model being more affected by outliers and noise (overfitting). Additionally, there is a limit to the depth of a decision tree that the dataset can support. Therefore, beyond a certain point, increasing the depth does not improve accuracy. For this dataset, the optimal value for the depth is determined to be 14.
- **min_samples_leaf**: A higher value of min_samples_leaf can limit the growth of the decision tree and serve as a pruning mechanism, improving generalization and accuracy. However, setting it too high can overly simplify the model and prevent the model from performing detailed classification. For this dataset, the balance point is determined to be 13.
- **splitter**: It is natural for the 'best' splitting strategy to exhibit better performance because it chooses the optimal splitting decisions on every node. The advantage of using the random splitting strategy is that it could generate more efficient decision strategies for complex datasets, but that is at the cost of sacrificing accuracy.