Pattern Recognition and Machine Learning: Homework 3, Zhengzuo Liu

Problem 1

Answer:

(1).

$$
\begin{aligned}
E_{COM} &= E_x\left[\frac{1}{M^2}\left(\sum \varepsilon_m(x)^2 + 2\sum_{1\le m<l\le M}\varepsilon_m(x)\cdot\varepsilon_l(x)\right)\right] \\
&= \frac{1}{M^2}\cdot\sum E_x[\varepsilon_m(x)]^2 + 2\sum_{1\le m<l\le M} E_x[\varepsilon_m(x)\cdot\varepsilon_l(x)] \\
&= \frac{1}{M^2}\cdot\sum E_x[\varepsilon_m(x)]^2 \\
&= \frac{1}{M} E_{AV}.
\end{aligned}
$$

(2). Since the arithmetic mean of n non-negative real numbers is not greater than their square mean, i.e. (or use Jensen's inequality & x^2 convex)

$$\frac{\sum_{i=1}^{n} a_i}{n} \le \sqrt{\frac{\sum_{i=1}^{n} a_i^2}{n}}$$

$$\Leftrightarrow \left(\sum_{i=1}^{n} a_i\right)^2 \le n\cdot\sum_{i=1}^{n} a_i^2.$$

We get

$$
\begin{aligned}
E_{COM} &= \frac{1}{M^2} E_x\left[\sum_{m=1}^{M}\varepsilon_m(x)\right]^2 \\
&\le \frac{1}{M^2} E_x\left[\sum_{m=1}^{M}|\varepsilon_m(x)|\right]^2 \\
&\le \frac{1}{M^2} E_x\left[M\cdot\sum_{m=1}^{M}\varepsilon_m(x)^2\right] \\
&= \frac{1}{M} E_x\left[\sum_{m=1}^{M}\varepsilon_m(x)\right]^2 \\
&= \frac{1}{M}\sum_{m=1}^{M} E_x[\varepsilon_m(x)]^2 = E_{AV}.
\end{aligned}
$$

Problem 2

Answer:

(1) See in program ”decision_tree.ipynb”.

(2) The answers are listed below.

make_split

Inputs

variable: Name of the variable that the split is based on.

value: Value of the variable at which the split is made.

data: The data being split.

is_numeric: A boolean value indicating whether the variable is numeric or categorical.

Returns

data_1: A pandas DataFrame representing the left split.

data_2: A pandas DataFrame representing the right split.

get_best_split

Inputs

y: The name of the target variable.

data: The data on which to calculate the best split.

Returns

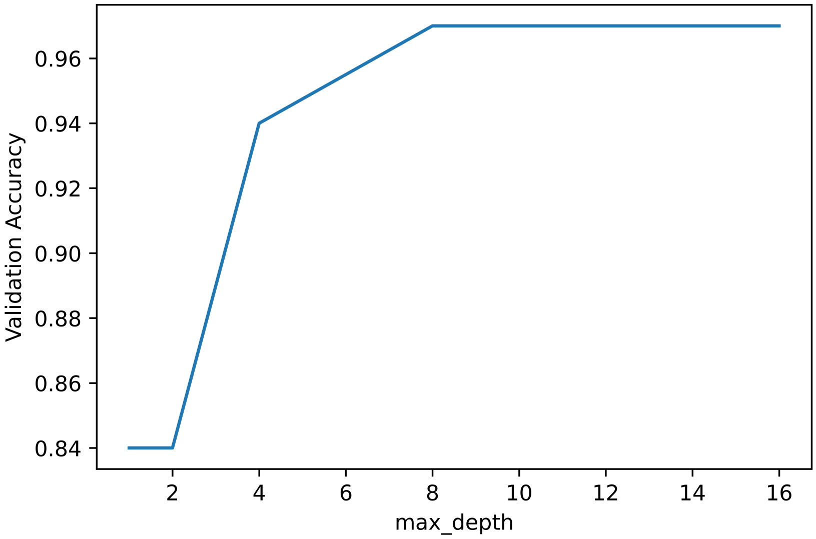
split_variable: The name of the variable that produced the best split.

split_value: The value of the variable at which the best split was made.

split_ig: The information gain obtained by splitting on
split_variable and split_value.

split_numeric: A boolean indicating whether split_variable is numeric or categorical.

(3)



Overfitting is not observed. There could be several possible reasons for this:

1. The dataset used for training the decision tree may not have a lot of noise or outliers, making it easier for the decision tree to learn a good generalization of the patterns in the data.

2. The decision tree algorithm may have implemented some form of regularization or pruning to prevent overfitting. For example, the min_samples_split parameter in the train_tree function specifies the minimum number of samples required to split an internal node.

3. The validation dataset used to evaluate the performance of the decision tree may not be diverse enough to uncover the limitations of the model, or it may not be large enough to detect overfitting.