Data Mining Assignment

- Classification using Decision tree –

202030069 park hyun woo

**Data Set**

First, the dataset used mushroom data obtained from Kaggle. This data has characteristic data about mushrooms, and there is information on whether it can be eaten or poisoned as a target attribute.

The overall description of the data is as follows.

*This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom.*

Data is briefly shown in the following figure, all attributes are categorical data, and an attribute named ‘class’ is a target attribute. ‘p’ means poisonous and ‘e’ means eligible.

옅은, 대형, 거리, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

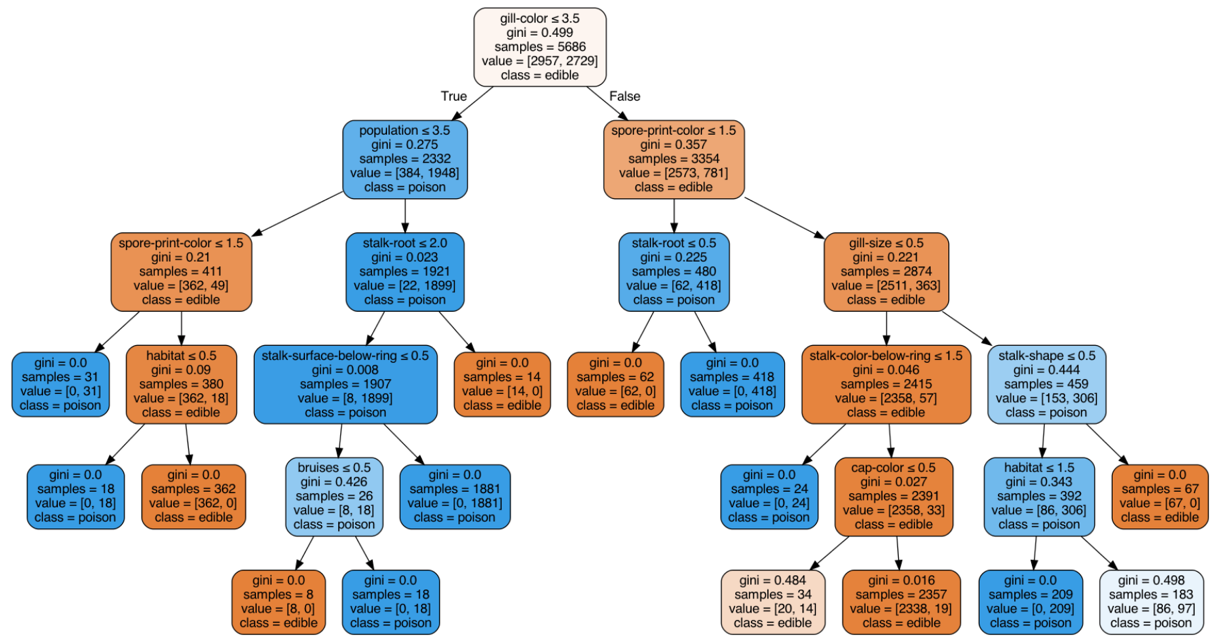
There are 22 attributes excluding the Target attribute and 8124 records. Fortunately, there were no missing or outliers in the data.

**Decision Tree with sklearn**

First, a decision tree was created using python library sklearn. Sklearn's decision tree supports gini for gini impurity or entropy for information gain. By default, gini is used.

The data preprocessing process was not so complex because there were no missing values and there were no attributes to exclude. The only data preprocessing was data transformation. Since Sklearn's decision tree model needs to receive data types as numbers for categorical data, all of the data has been transformed into numbers. At this time, each categorical data was easily converted into numeric data by utilizing sklearn's preprocessing function.

Next, the training data and test data were divided into 70% and 30%, respectively, and then a decision tree model with a maximum depth of 5 was created using the training data. And this model was created using the default gini. As a result, the following tree was created.



When the maximum depth was 5, grill-color was the most important attribute, followed by population and spore print color.

In this case, the confusion matrix was as follows.

|  |  |  |
| --- | --- | --- |
|  | Real Eligible | Real Poisonous |
| Predict Eligible | 1214 | 12 |
| predict Poisonous | 38 | 1174 |

Therefore, the accuracy is (1214 + 1174) / (1214 + 12 + 38 + 1174), which is about 0.98.

Next, when the maximum depth was set to 7, also using gini, the following model appeared.

냉장고이(가) 표시된 사진

자동 생성된 설명

For this model, the confusion matrix was as follows.

|  |  |  |
| --- | --- | --- |
|  | Real Eligible | Real Poisonous |
| Predict Eligible | 1284 | 0 |
| predict Poisonous | 0 | 1154 |

In other words, it was a model with 100% accuracy.

After that, i created a decision tree with a maximum depth of 5 using entropy. Then, a decision tree was created that was smaller than when using gini.

표지판이(가) 표시된 사진

자동 생성된 설명

Also in this case, grill-color was the most important attribute. As a result of the verification procedure for the test set, the confusion matrix is as follows.

|  |  |  |
| --- | --- | --- |
|  | Real Eligible | Real Poisonous |
| Predict Eligible | 1180 | 0 |
| predict Poisonous | 63 | 1195 |

As a result, the accuracy was 97%.

Finally, we created a model using entropy with the maximum depth set to 7. In this case, the following model

was created.

표지판이(가) 표시된 사진

자동 생성된 설명

Again, the grill-color attribute had the best discrimination, and the confusion matrix was as follows.

|  |  |  |
| --- | --- | --- |
|  | Real Eligible | Real Poisonous |
| Predict Eligible | 1268 | 0 |
| predict Poisonous | 0 | 1170 |

Even in the model made with entropy, the accuracy was 100% when the maximum depth was 7.

**Decision tree algorithm**

This time, I made a classification model using a decision tree implemented by code, not a python library.

From the code posted on Github, the model was able to created by correcting the version-related error, transforming the data appropriately. This code creates a Decision tree using the ID3 method, which also uses entropy.

Unfortunately, I couldn't find any visualization-related code.

So, I couldn't see the shape of the tree directly, but through the additional coding, the shape inside the tree was found. In the case of this code, unlike sklearn, it is not a binary tree, and child nodes are created according to each value in the attributes.

Again, the training set and test set were set to 70% and 30%, respectively, and the experiment was conducted. After experimenting with setting the maximum depth to 5, the most important attribute was odor unlike sklearn, and the confusion matrix was as follows.

|  |  |  |
| --- | --- | --- |
|  | Real Eligible | Real Poisonous |
| Predict Eligible | 1256 | 0 |
| predict Poisonous | 0 | 1182 |

As a result, it has 100% accuracy. The maximum depth was shorter than that of sklearn, but the accuracy was very high. In my opinion, this algorithm created child nodes for each value in selected attribute, but since sklearn uses a binary tree, it has relatively few leaves, so sklearn seems to have low accuracy if the depth of the tree is not deep enough.

Additionally, a comparison of learning time was conducted.

For sklearn decision tree, learning time was 5.4 ms ± 614 µs and for python decision tree algorithm, it tooks 1.69 s ± 28.2 ms for learning.

Reference

- <https://www.kaggle.com/mig555/mushroom-classification>

- <https://scikit-learn.org/stable/modules/tree.html>

- <https://github.com/tiepvupsu/DecisionTreeID3>