HCMUS – University of Science in Ho Chi Minh City

REPORT
Subjects: Introduction to Artificial Intelligence

LAB 02:
DECISION TREE WITH SCIKIT-LEARN

Class: 21CLC05

Information:

21127608 Trần Trung Hiếu

Teacher:

D.Sc. Nguyễn Hải Minh
D.Sc. Nguyễn Ngọc Thảo
MS. Hồ Thị Thanh Tuyến

TABLE OF CONTENTS

I. TOPIC 3
I. REPORT 3
1. Preparing the data sets 3
2. Building the decision tree classifiers 3
3. Evaluating the decision tree classifiers 4
4. The depth and accuracy of a decision tree 5
II. REFERENCE 5

I. TOPIC

- In this assignment, you are going to build a decision tree on the UCI Poker Hand Data Set, with the support from scikit-learn library.
  About the Poker Hand Data Set from UCI Machine Learning Repository
  There are **1,025,010 records** in the data set. Each of which is an example of a hand consisting of five playing cards drawn from a standard deck of 52. Each card is described using two attributes (suit and rank), for a total of **10 predictive attributes**. There is **one Class attribute** that describes the "Poker Hand". The order of cards is important, which is why there are 480 possible Royal Flush hands as compared to 4
    + Preparing the data sets
    + Building the decision tree classifiers
    + Evaluating the decision tree classifiers
    + The depth accuracy of a decision tree

I. REPORT

1. Preparing the data sets

- First, I want to merge 2 files **'poker-hand-training-true.data'** and **'poker-hand-testing.data'**, I use function**'concat()'** in **pandas** library, then to shuffle the data do i use **'sample()'** function in **pandas** library and save in file poker-hand-data.csv (Because the data is too big, I have attached the data file here:link data)
- I use function **'train_test_split()'** of the **sklearn.model_selection** module of the **Scikit-learn** library to split the data into train sets and test sets based on the division ratio in the required problem
- .To visualize the distributions of classes in all the data sets, I count the number of repetitions of the labels for each term. To visualize the data I define 3 types: the original sets, the train sets and the test sets. I use the library **Mathplotlib** and **Seaborn** to draw a column chart showing the correlation between each class of the original set, each set of train - set with the corresponding ratio that the problem gives. Bar chart showing where the x-axis is the Poker hand and the y-axis is the number of each term

2. Building the decision tree classifiers

- **DecisionTreeClassifier()** is a class in **scikit-learn** library that represents a decision tree classifier. Decision trees are hierarchical structures that recursively split the dataset based on different features to make decisions. I use chaos criteria (also known as

entropy) to choose the best possibility to split the data at each inner node, in order to reduce entropy.

- I use the **Graphviz** library which is a separate library used to visualize decision trees. I use the **export_graphviz()** function to export the decision tree from the **sklearn.tree** module. It converts the decision tree model into a Graphviz compatible representation, which is a textual representation of the tree structure in the DOT language. The DOT language is a plain text graph description language used by Graphviz.
- The data is then saved as images in the folder 'Decision tree' with each given scale, I get a decision tree.
- My computer can't render unlimited depth graphs so I extend the limit for decision trees

3. Evaluating the decision tree classifiers

- **Confusion_matrix** is a table that summarizes the results of the classification model's predictions during the evaluation process. The **Confusion_matrix** provides an overview of the data's categorization capabilities. It is used to measure the accuracy of the model by comparing the actual class labels and the predicted class labels:

  + True Positive (TP): Number of samples correctly predicted to be in the positive class.

  + False Positive (FP): The number of samples that are falsely predicted to be in the positive class.

  + False Negative (FN): The number of samples that are falsely predicted to be not in the negative class.

  + True Negative (TN): Number of samples correctly predicted that are not in the positive class (negative).

- **Classification_report** is a report that summarizes the performance evaluation metrics of the classification model. The **classification_report** calculates and displays the model performance metrics based on the comparison between predicted labels and true labels. It provides detailed information about:

  + Precision: The ratio between the number of correct predictions belonging to the positive class and the total number of positive predictions

$$\frac{TP}{TP + FP}$$

  + Recall(coverage): Ratio between the number of correct predictions in the positive class and the total number of positives

$$\frac{TP}{TP + FN}$$

  + F1-score: The harmonic mean of precision and recall, which is a combined measure of both (2 * (precision * recall) / (precision + recall)).

$$\frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}}$$

+ Support (number of samples)

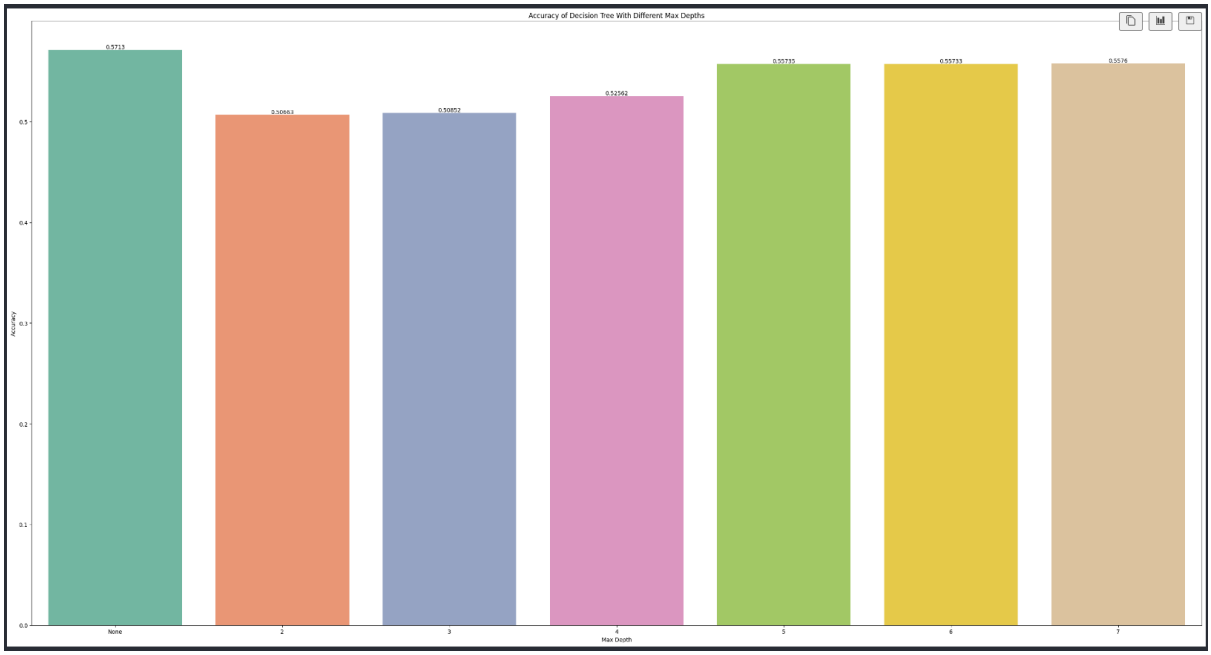
- For each training rate, I have a set of predictions and a set of tests to be able to create **Classification report** and **Confusion matrix**

4. The depth and accuracy of a decision tree

- **Accuracy_score** is a metric used to evaluate the performance of a **classification model**. It measures the proportion of correctly classified instances out of the total number of instances.
- Accuracy Score formula:

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{total nmber of predictions}}$$

Max_depth	None	2	3	4	5	6	7
Accuracy	0.5713163774011961	0.5066292036175257	0.508524307079930 9	0.5256192622511	0.5573470014926684	0.5573311479888001	0.5576018770548 58



→ As the depth of the decision tree increases, the classification accuracy also increases.

II. REFERENCE