**LAB # 05**

**SUPERVISED LEARNING (DECISION TREE)**

**OBJECTIVE**

Implementing supervised learning, DTS algorithm for training, testing and classification.

**THEORY**

**SUPERVISED LEARNING:**

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labeled data.

For instance, suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all different fruits one by one like this:



Fig. 1

* If shape of object is rounded and depression at top having color Red then it will be labelled as –**Apple**.
* If shape of object is long curving cylinder having color Green-Yellow then it will be labelled as –**Banana**.

Now suppose that we have trained the dataset and were given a separate fruit say banana for instance.

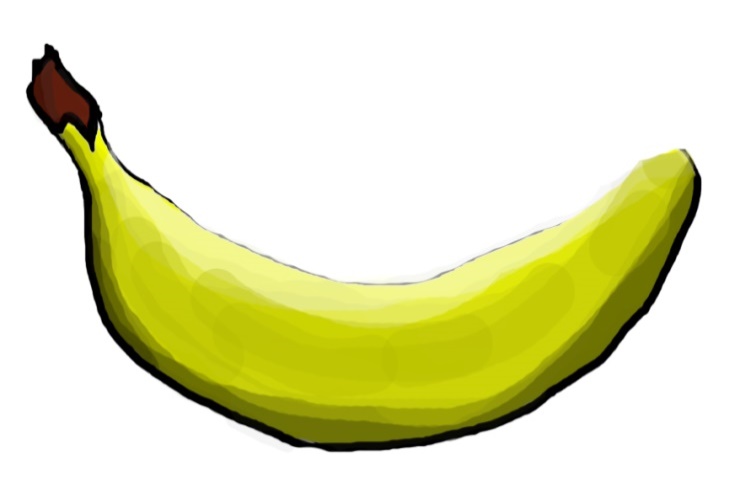


Fig. 2

Since the machine has already learned the things from previous data and this time have to use it wisely. It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in Banana category. Thus the machine learns the things from training data (basket containing fruits) and then apply the knowledge to test data (new fruit).

Supervised learning classified into two categories of algorithms:

* **Classification**: A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease”.
* **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

1. **Decision Tree Algorithm (DTS)**

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Diagram

Description automatically generated

Fig. 3

* 1. **Construction of the Decision Tree**

A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

* 1. **Decision Tree Representation**

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, and then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the sub-tree rooted at the new node.

The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf.(in this case Yes or No).

For example, the instance in fig. 4, would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance.

In other words we can say that decision tree represent a disjunction of conjunctions of constraints on the attribute values of instances as given in fig. 5.

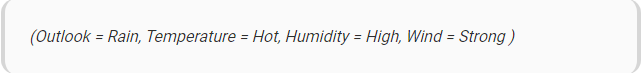


Fig. 4

Logo, company name

Description automatically generated

Fig. 5

1. **Used Python Packages**

* **Sklearn:** In python, sklearn is a machine learning package which include a lot of ML algorithms.
* **Numpy:** It is a numeric python module which provides fast maths functions for calculations. It is used to read data in numpy arrays and for manipulation purpose.
* **Pandas:** Used to read and write different files.Data manipulation can be done easily with data frames.

1. **Pseudo code**

* Import the appropriate libraries, Pandas, numpy and sklearn using,

import pandas as pd

import numpy as np

from sklearn.tree import DecisionTreeClassifier

* Construct panda data frame.
* Generate the list of features.
* Generate the list of labels. (Note: Number of label must be same as the number of feature rows)
* Call the chosen classifier container that will train the feature data frame.

Classifier = DecisionTreeClassifier()

* Train the model using features and label,

classifier = classifier.fit(features\_1, labels)

* Include a new entry having only features.

new\_entry=[[6.00 ,180,12]]

* Predict the class of the new entry using,

prediction = classifier.predict(new\_entry)

* Find Confusion matrix and accuracy score

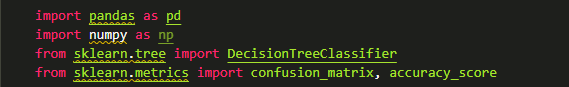
**Lab Tasks**

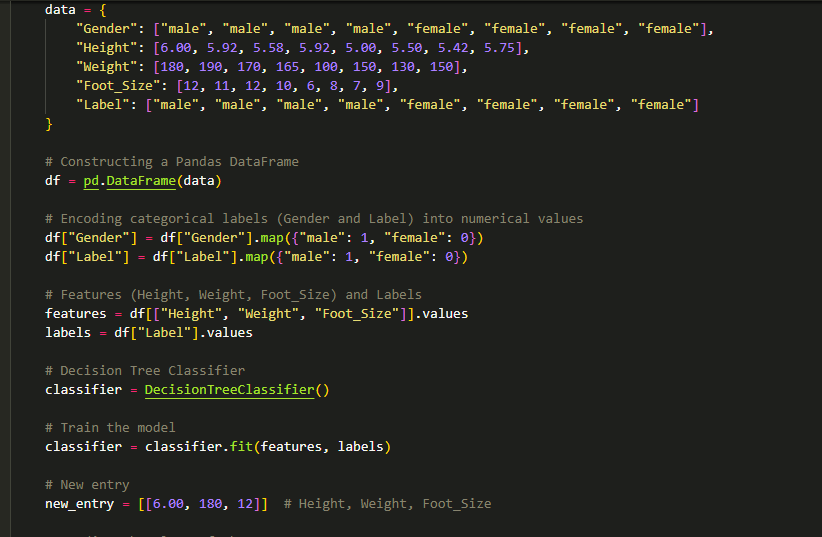
1. Implement the Decision tree algorithm on the data given in the table. 1 and predict the new entry entered by the user.

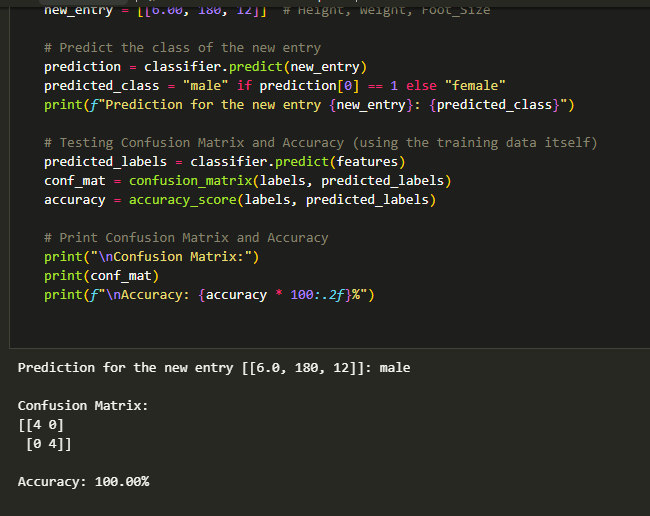
Table. 1

Text

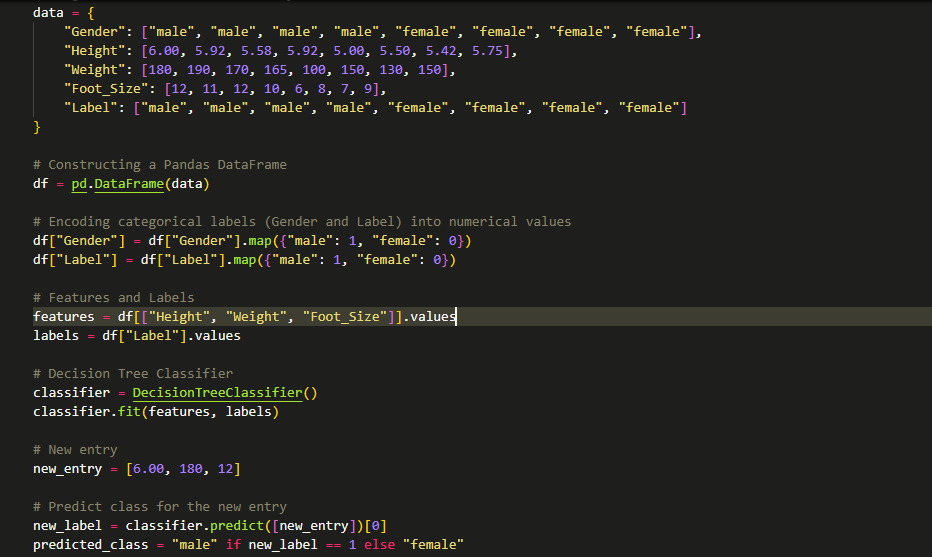
Description automatically generated with low confidence







1. Implement Decision Tree using table. 1 in such a way that the new entry becomes the part of the given dataset.





1. Implement Decision Tree using table. 1 without the use of Pandas library. You can use numpy.

