[1]

Data Mining Lab on Decision Tree

Jagadish Khadka(2073/BEX/313),

Krishna Bahadur Ojha(2073/BEX/317)

Rajad Shakya(2073/BEX/332)

***Abstract*— This lab is basic overview in decision tree with selection of strong features of data and levels using the entropy. Entropy and Gini are generally used for the feature selection and different library of the python can draw tree of it.In this lab, tree generated using criteria like gini and entropy and saved them to a png file. Also, a tree is saved in form of dictionary by using modules like pprint.**

***Index Terms*—Decision Tree, CART, Entropy, Gini, ID3**

I. Introduction

Decision Tree is widely used machine learning algorithm for supervised learning.It is flowchart like structure in which each internal node represents 'test', branches represents outcomes of the test and node represents class level.It breaks down a data set into smaller and smaller subsets building along an associated decision tree at the same time. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. The leaf node represents a classification or decision.

II. Decision Tree

The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.It forces the consideration of all possible outcomes and traces each path to the conclusion.
Advantages:

- Decision trees require less effort for data preparation as compared to other algorithms during pre-processing.
- Do not require normalization of data.
- Do not require scaling of data as well.
- Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
- A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

Disadvantages:

- A small change in the data can cause a large change in the structure of the decision tree causing instability.
- For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
- It often involves a higher time to train the model.
- Its training is relatively expensive because the complexity and time taken are more.
- Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

Importance:

- Simple to easy, use, understand and explain
- They do feature selection and variable screening
- They require very little efforts to prepare and produce
- Can live with nonlinear relationships and parameters also
- Can use conditional probability-based reasoning or Bayes theorem
- They provide strategic answers to uncertain situations

A. Hunt's rule

In this algorithm, decision tree is grown in a recursive fashion by partionating the training records into successively purer subset. Let D_t be the set of training records that are associated with the node t and y={y1,y2,y3...yc} be the class labels. Then the Hunt's algorithm is
Step1:
If all the records in D_t belongs to the same class y_t, then t is a leaf node labeled as y_t.
Step2:
If Dt contains records that belongs to the different class labels an attribute test condition is selected to partitioned the records into smaller subsets. A child node is created for each outcomes of the test condition and the records in Dt are distributed into children based on the outcomes.The algorithm is then recursively applied to each child node.

B. ID3

ID3 stands for Iterative Dichotomiser, is for binary classification only . It determines the classification of objects by testing the values of the properties. It builds a decision tree for the given data in a top-down fashion, starting from a set of objects and a specification of properties.At each node of the tree, one property is tested based on maximizing information gain and minimizing entropy, and the results are used to split the object set.

[1]

Advantages:

- Understandable prediction rules are created from the training data.
- Builds the fastest tree.
- Builds a short tree.
- Only need to test enough attributes until all data is classified.
- Finding leaf nodes enables test data to be pruned, reducing the number of tests.
- Whole dataset is searched to create tree.

Disadvantages:

- Data may be over-fitted or over-classified, if a small sample is tested.
- Only one attribute at a time is tested for making a decision.
- Does not handle numeric attributes and missing values.

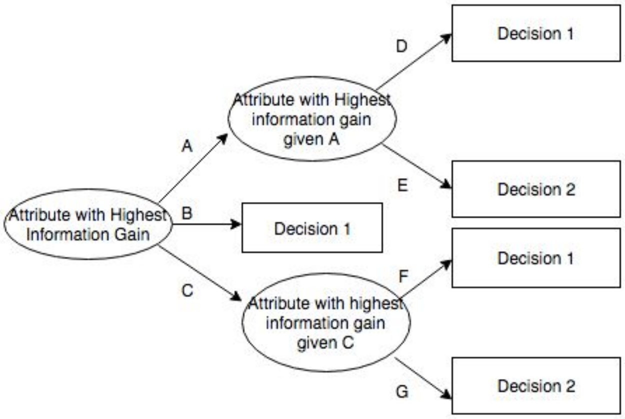

Attribute with Highest Information Gain
A
B
C
Attribute with Highest information gain given A
Decision 1
Attribute with highest information gain given C
D
E
F
G
Decision 1
Decision 2
Decision 1
Decision 2

Fig:Flowchart of ID3 algorithm

C. CART

CART stands for Classification and Regression Trees (Breiman et al., 1984).It is characterized by the fact that it constructs binary trees, namely each internal node has exactly two outgoing edges. The splits are selected using the two criteria and the obtained tree is pruned by cost–complexity Pruning. When provided, CART can consider misclassification costs in the tree induction. It also enables users to provide prior probability distribution. An important feature ofCART is its ability to generate regression trees.

Regression trees are trees where their leaves predict a real number and not a class. In case of regression, CART looks for splits that minimize the prediction squared error (the least–squared deviation). The prediction in each leaf is based on the weighted mean for node It has the following advantages and disadvantages:

Advantages:

- CART can easily handle both numerical and categorical variables.
- CART algorithm will itself identify the most significant variables and eliminate nonsignificant ones.
- CART can easily handle outliers.

Disadvantages:

- CART may have unstable decision tree. Insignificant modification of learning sample such as eliminating several observations and cause changes in decision tree: increase or decrease of tree complexity, changes in splitting variables and values.
- CART splits only by one variable.

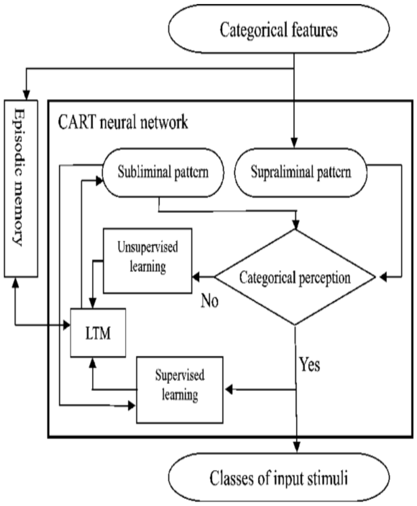


Fig:Flowchart of CART algorithm

III. MATH

ID3 and CART are the commonly used algorithm for the decision tree making. They are used to measure the information of the attribute of the data so that strong feature can be chosen and extract the knowledge of data to predict the appropriate label to the random sample. Where ID3 algorithm based on the entropy t and CART uses the Gini value to select strong feature of the sample..

Entropy H(s) = $-\sum_{i=1}^{n} P_{i} \, log_{b} P_{i}$

Gini Index: It is another measure of impurity that measures the divergences between the probability distributions of the target attribute's values.

Gini Index = 1- $\sum_{i=0}^{n}$ [p (i/t)] ^2

IV. Data

we created a dataset with taste, temperature, texture and eat as a feature of each sample. Dataset of 10 samples in the form of dictionary. In taste feature there are values like whether it is salty or spicy or sweet, in temperature feature the values might be hot or cold, in texture feature values can be soft or hard, and eat feature value can be yes or no.These datas in dictionary is converted into a Dataframe using pandas library.

V. Libraries and functions

A. Libraries

- pandas: It is an open source library for high-performance data analysis and provides an array of data into user readable format.

- numpy:It is a fundamental package for scientific computing with python. It is used as a multidimensional container of generic data.

- scikit-learn: It is an open source python library. which is used for traditional machine learning, cross-validation, visualization and preprocessing.

- matplotlib.pyplot: It is a 2D plotting library of python programming language for visualization of arrays.

- IPython.display:It provides different constructions to control over audio, video and images.

- Pydotplus:PyDotPlus is an improved version of the old pydot project that provides a Python Interface to Graphviz's Dot language.

- time:Time library provides the time related control. It contains time related functions like time(), sleep(), localtime(), gmtime() etc.

B. Functions

- numpy.finfo:Machine limits for floating point types.
- pd.DataFrame:converts the given data into a tabular format.
- unique(): selects the unique values of list and returns it.
- value_counts():counts the number of data in the list.
- len():length of data in the list or a string
- np.log2():Base-2 logarithm of input number
- round():rounds off the input number to less number of decimal points.
- abs():returns the absolute value of the input number
- list():convert the input data structure into a list
- get_dummies(): used to convert categorical variable into dummy/indicator variables.
- range():generates the integer numbers between the given start integer to the stop integer, which is generally used to iterate over with for loop.
- Image():used to display image in the jupyter notebook and takes parameter as filename.

The index is defined as a function which takes attribute as a parameter and displays the entropy and information gain of that attribute.This function works first by storing the unique values of target variable and attribute variable in a list and then, entropy for each variable is calculated and information gain for each attribute is calculated and printed.

- DecisionTreeClassifier():
  it is a decision tree classifier that generates a tree based on various parameters like gini or entropy.It is a class capable of performing multi-class classification on a dataset.This class takes various parameters :
  criterion:
  The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.
  splitter :
  The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.
  max_depth:
  The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
  random_state:
  If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

- fit():The train and test data of dataset separated by using the train_test_split is fed to this function to generate decision tree.
- export_graphviz():This function generates a GraphViz representation of the decision tree, which is then written into out_file, where out_file is a return value generally an output file.
- graph_from_dot_data():Load graph as defined by the data in DOT format.The data is assumed to be in DOT format. It will be parsed and a Dot class will be returned, representing the graph.
- create_png(): converts the graph that is generated from graph_from_dot_data function into a png format.
- write_png():saves the graph that is generated from graph_from_dot_data function into a png with name given in the parameter.

VI. Result

The decision tree is constructed using two criteria entropy and gini. Various other parameters are set max_features to 3, random_state to rand, max_depth= 5,and splitter to best. Then, they are exported first into a graph and then saved into a png file.And, the tree obtained is also stored in dictionary by defining functions like find_entropy which takes a dataframe as parameter and returns the entropy value, find_entropy_attribute function which takes dataframe and attribute as a parameter and returns entropy using given attribute,find_winner which returns the values in the dataframe that has maximum information gain taking dataframe as parameter, get_subtable that returns that returns a part of a table and takes dataframe,node and value as as parameters and buildTree function which is recursively called until we get into the last node of the tree and saving each node to a dictionary. The graph obtained are:

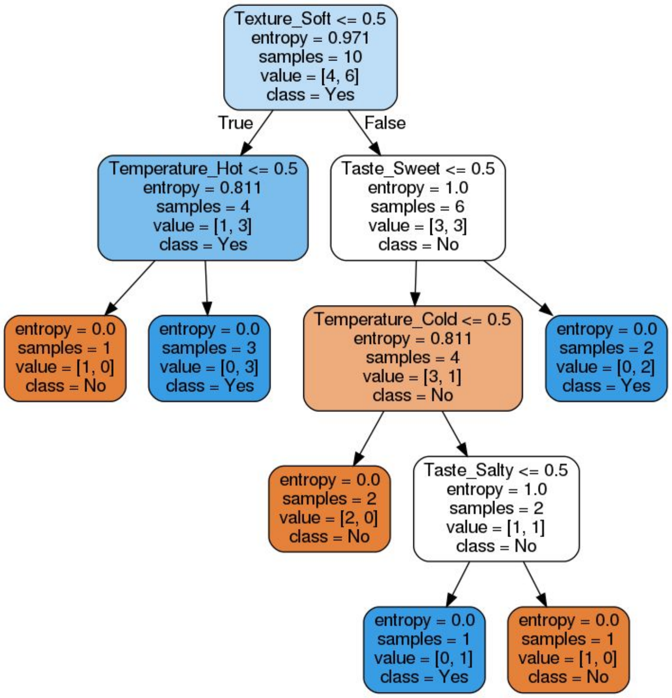


Fig: Decision tree obtained by taking entropy as criterion

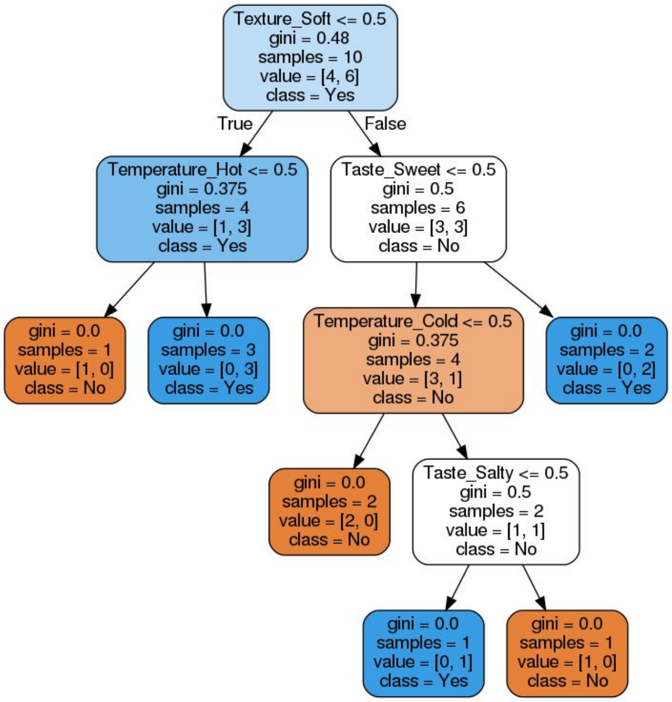
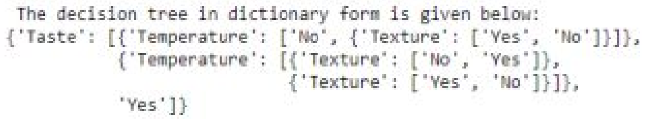


Fig: Decision tree obtained by taking gini as criterion

This dictionary obtained from tree is:



```
 The decision tree in dictionary form is given below:
{'Taste': [{'Temperature': ['No', {'Texture': ['Yes', 'No']}]},
          {'Temperature': [{'Texture': ['No', 'Yes']},
                           {'Texture': ['Yes', 'No']}]},
          'Yes']}
```

Fig : Decision tree represented in dictionary format

VII. Discussion and conclusion

Decision tree is technically simple and have no effect on missing value of data. It does not require normalization and scaling, but calculation is far more complex and small change of data leads to a large change of decision tree.It is concluded that gini and entropy play a vital role in splitting of decision tree and obtained tree can be easily stored in the graph and png files.Also, the obtained tree can also be stored in a dictionary format.

Appendix

HTTPS://GITHUB.COM/RJ7SHAKYA/DATAMINING/BLOB/MASTER/LAB/LAB2.IPYNB

References

[1]L. Breiman, J. H. Friedman, R. Olshen, and C. J.Stone. *Classification and Regression trees. Chapman* & Hall, New York,1984.

[2]P. Domingos. The Role of Occam;s Razor in Knowledge Discovery. *Data Mining and Knowledge Discovery*, 3(4):409-425,1999.

[3]R. O. Duda, P. E. Hart , and D. G. Stock. *Pattern Classification*. John Wiley & Sons,Inc., New York,2nd edition, 2001.

[4]Pang-Ning Tan, Michael Steinbach, Vipin Kumar Introduction to Data Mining, 2017

[5]Anju Rathee, Robin prakash mathur, "Survey onDecision Tree classification algorithm for the evaluation of student performance" International Journal ofComputers & Technology, Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

[6] S.Anupama Kumar and Dr. Vijayalakshmi M.N. (2011) "Efficiency of decision trees in predicting a student's academic performance", D.C. Wyld, et al. (Eds): CCSEA 2011, CS & IT 02, pp. 335-343, 2011

[7] Jiawei Han and Micheline Kamber Data Mining: Concepts and Techniques, 2ndedition. [8] Baik, S. Bala, J. (2004), A Decision Tree Algorithm For Distributed Data Mining.

[9] Prof. Nilima Patil and Prof. Rekha Lathi(2012), Comparison of C5.0 & CART Classification algorithms using pruning technique.

[10] Baik, S. Bala, J. (2004), A Decision Tree Algorithm For Distributed Data Mining.