Assignment No: 03
Classification and Learning [Decision Tree]

CSE-0408 Summer 2021

Mst.Kamrunnhar Somapti
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
somaptiesuborno@gmail.com

***Abstract*—A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret. The decision trees are discussed in depth in this Assignment. Furthermore, the contents of the Assignment, such as the algorithms/approaches employed, data-sets, and outcomes achieved, are thoroughly analyzed and discussed. Furthermore, all of the methodologies examined and determine the most accurate classifiers.**

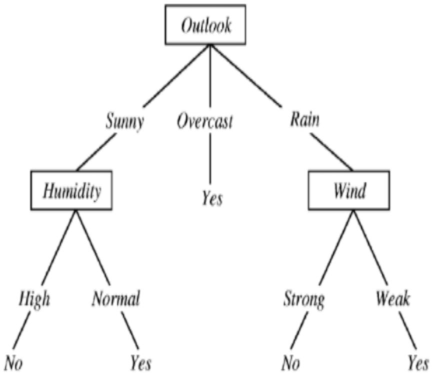
***Index Terms*—Artificial Intelligence, Machine Learning, Decision Tree (DT).**

I. Introduction

Decision trees are powerful and popular tools for classification and prediction. Decision trees represent rules, which can be understood by humans and used in knowledge system such as database. A decision tree is a hierarchical model for supervised learning whereby the local region is identified in a sequence of recursive splits in a smaller number of steps. A decision tree is composed of internal decision nodes decision node and terminal leaves. Each decision node m implements a test function fm(x) with discrete outcomes labelling the branches. Given an input, at each node, a test is applied and one of the branches is taken depending on the outcome. This process starts at the root and is repeated recursively until a leaf node is hit, at which point the value written in the leaf constitutes the output

A decision tree is also a non parametric model in the sense that we do not assume any parametric form for the class densities and the tree structure is not fixed a priori but the tree grows, branches and leaves are added, during learning depending on the complexity of the problem inherent in the data. Decision tree is a classifier in the form of a tree structure which consists of:

- Decision node: specifies a test on a single attribute.
- Leaf node: indicates the value of the target attribute.
- Edge: split of one attribute.
- Path: a disjunction of test to make the final decision.



A Decision Tree to predict the weather

Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

II. Literature Review

A decision tree was utilized as a classifier in numerous machine learning and data mining projects. Several recent works on the DT are discussed in this research.

For diabetes mellitus prediction, Zou et al. used decision trees, Random Forest (RF), and neural network techniques. Physical research data for hospitals in Luzhou, China is included in the dataset. There are 14 different characteristics to consider. The training array extracts data from 68994 healthy humans and diabetic patients at random. To reduce dimensionality, they exploited the full significance of minimum Redundancy Maximum Relevance (mRMR) and Principal Component Analysis (PCA). In certain instances, the effects of RF, as opposed to the other classifiers, appeared to be larger. Furthermore, in the Luzhou data collection, 0.8084 is the best result.

Assegie and Nair used the DT classification technique to categorize the handwritten digits in the kaggle digits standard data set and assess the model's accuracy for each digit from 0 to 9. The kaggle features comprise 42,000 rows and 720 columns for machine learning, as well as vector characteristics for digital image pixels. They applied machine learning algorithms to map the classifier's success rate graph in the reality of handwritten digits using a highly efficient language called ”python programming.” The 83.4 percent accuracy and decision tree classifier had an impact on handwritten number recognition, according to the findings.

III. DECISION TREE ALGORITHM

Systems that produce classifiers are one of the most extensively utilized strategies in data mining. Classification algorithms are capable of handling a large amount of data in data mining. It can be used to make categorical class name assumptions, categorize knowledge based on training sets and class labels, and classify newly available data. Machine learning classification techniques include a variety of algorithms, however this research focuses on the decision tree algorithm in particular. The structure of DT is depicted in Figure 1.

Decision trees are a strong tool that may be utilized in a variety of domains, including machine learning, image processing, and pattern recognition. DT is a sequential model that effectively and cohesively connects a series of fundamental tests in which a numeric feature is compared to a threshold value in each test. The numerical weights in the neural network of connections between nodes are far more difficult to construct than the conceptual rules. DT is primarily used for grouping purposes. Furthermore, in Data Mining, DT is an often used classification model. Each tree is made up of its nodes and branches. Each subset defines a value that the node can take, whereas each node represents features in a category to be categorised. Decision trees offer a wide range of applications due to their straightforward analysis and precision across many data types. An example of DT is shown in Figure 2.

Libraries Requirements

- *pandas*
- *sklearn*
- *IPython*
- *matplotlib*

Pandas is used to take input data sets, **sklearn** is used to develop and train our models, as well as **IPython** and **matplotlib** are used to visualize our decision trees graphically.

IV. Advantages of Decision trees:

- Easy to read and interpret.
- Easy to prepare.
- Less data cleaning required.

V. Disadvantages of Decision Trees:

- Unstable nature.
- Less effective in predicting the outcome of a continuous variable.

VI. Conclusion

This assignment is based on a graphic representation of a decision tree. A data-set is given for the training and visualization of this decision tree.

Acknowledgment

I would like to thank my honourable **Khan Md. Hasib Sir** for his time, generosity and critical insights into this project.

References

[1] D. Abdulqader, A. Mohsin Abdulazeez, and D. Zeebaree, “Machine Learning Supervised Algorithms of Gene Selection: A Review,” Apr. 2020.

[2] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, “Machine learning: a review of classification and combining techniques,” Artif Intell Rev, vol. 26, no. 3, pp. 159–190, Nov. 2006, doi: 10.1007/s10462-007-9052-3.

[3] Anuradha and G. Gupta, “A self explanatory review of decision tree classifiers,” in International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), Jaipur, India, May 2014, pp. 1–7, doi: 10.1109/ICRAIE.2014.6909245.

[4] S. Taneja, C. Gupta, K. Goyal, and D. Gureja, “An enhanced k-nearest neighbor algorithm using information gain and clustering,” in 2014 Fourth International Conference on Advanced Computing & Communication Technologies, 2014, pp. 325–329.

Assignment No: 04
Classification and Learning [K- Nearest Neighbors]

CSE-0408 Summer 2021

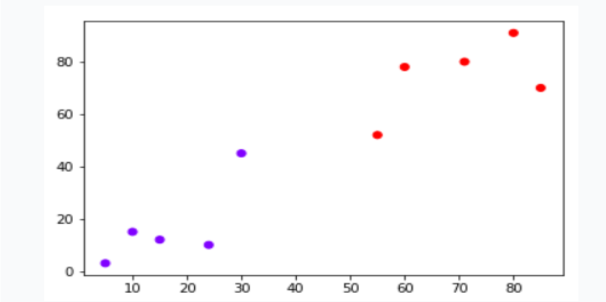
Kamrunnahar Somapti
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
somaptiesuborno@gmail.com

***Abstract*—The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows. The K-nearest neighbors (KNN) algorithm is a type of supervised machine learning algorithms. KNN is extremely easy to implement in its most basic form, and yet performs quite complex classification tasks. Rather, it uses all of the data for training while classifying a new data point or instance. KNN is a non-parametric learning algorithm, which means that it doesn't assume anything about the underlying data. This is an extremely useful feature since most of the real world data doesn't really follow any theoretical assumption e.g. linear-separability, uniform distribution, etc.**

***Index Terms*—Machine Learning, Supervised, Classification, K nearest neighbors.**

I. Introduction

The intuition behind the KNN algorithm is one of the simplest of all the supervised machine learning algorithms. It simply calculates the distance of a new data point to all other training data points. The distance can be of any type e.g Euclidean or Manhattan etc. It then selects the K-nearest data points, where K can be any integer. Finally it assigns the data point to the class to which the majority of the K data points belong.let's see this algorithm in action with the help of a simple example.



The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems.Nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well-

Lazy learning algorithm-KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.

Parametric learning algorithm-KNN is also a non-parametric learning algorithm because it dosen't assume anything about the underlying data.

II. KNN ALGORITHM

We can implement a KNN model by following the below steps:

1) For implementing any algorithm, we need data set. So during the first step of KNN, we must load the training as well as test data.
2) Next, we need to choose the value of K. The nearest data points. K can be any integer.
3) For each point in the test data do the following
    a) Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
    b) Now, based on the distance value, sort them in ascending order.
    c) Now, it will assign a class to the test point based on most frequent class of these rows.
4) The k-Nearest Neighbors algorithm for short is a very simple technique. The entire training dataset is stored

Libraries Requirements

- *pandas*
- *sklearn*
- *matplotlib*

Pandas is used to take input data sets, **sklearn** is used to develop and train our models, as well as **matplotlib** are used to visualize our K-Nearest Neighbors accuracy graphically.

III. Some KNN Advantages and Disadvantages

Some Advantages of KNN

- Quick calculation time.
- Simple algorithm – to interpret.
- High accuracy – you do not need to compare with better-supervised learning models.
- No assumptions about data – no need to make additional assumptions, tune several parameters, or build a model. This makes it crucial in nonlinear data case.

Some Disadvantages of KNN

- Accuracy depends on the quality of the data.
- With large data, the prediction stage might be slow.
- Sensitive to the scale of the data and irrelevant features.
- Given that it stores all of the training, it can be computationally expensive.

IV. Conclusion

K-Nearest neighbor classification is a general technique to learn classification based on instance and do not have to develop an abstract model from the training data set. However the classification process could be very expensive because it needs to compute the similiary values individually between the test and training examples. K-nearest neighbor classifer also suffers from the scaling issue. It computes the proximity among the test example and training examples to perform classification. If the attributes have different scales, the proximity distance might be donimated by one of the attributes, which is not good.

Acknowledgment

I would like to thank my honourable **Khan Md. Hasib Sir** for his time, generosity and critical insights into this project.

References

[1] Solichin, A. (2019, September). Comparison of Decision Tree, Naïve Bayes and K-Nearest Neighbors for Predicting Thesis Graduation. In 2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 217-222). IEEE.