Assignment No: 03
Report On Decision Tree

CSE-0408 Summer 2021

Khaled Saifullah Sadi
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
mdsadi4@gmail.com

***Abstract*—Decision tree classifiers are widely recognized as one of the most well-known approaches for representing data classification in classifiers.**

***Index Terms*—Python,Artificial Intelligence,Decision Tree**

I. Introduction

Technology has advanced significantly in recent years, particularly in the field of Machine Learning (ML), which is effective for minimizing human labor. A decision tree is a graphical representation of all possible solutions to a decision. These days, tree-based algorithms are the most commonly used algorithms in the case of supervised learning scenarios.They are easier to interpret and visualize with great adaptability. Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance.

Decision tree is a flowchart-like tree structure where an internal node represents feature, the branch represents a decision rule, and each leaf node represents the outcome

In this project, a thorough implementation of the most recent and most effective techniques to decision trees in several fields of machine learning that have been developed by researchers over the last three years is carried out. Making a decision tree is also a part of the process.

II. Literature Review

Lertworaprachaya et al.,2014 [1] proposed a new model for compose decision trees using interval-valued fuzzy membership values.

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.

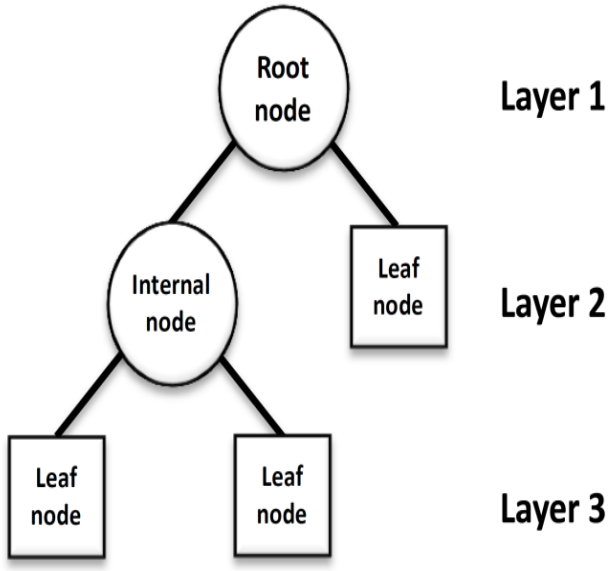


Fig. 1. Decision Tree

III. DECISION TREE ALGORITHM

Decision trees are a simple classification tool capable of separating records of data into specific categories by proposing a series of questions.Decision trees are commonly used due to many factors, including their relatively small learning curve for interpretability

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

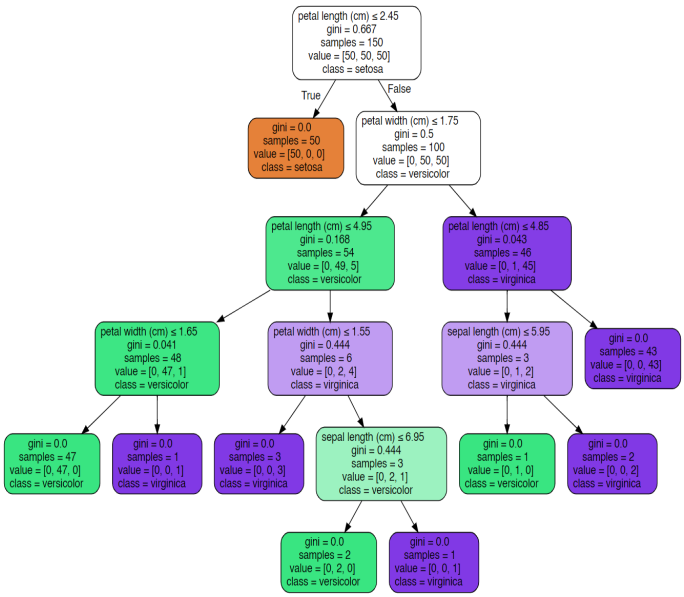


Fig. 2. Example on Decision Tree

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. TYPES OF NODES

A decision tree consists of three types of nodes:

Decision nodes – typically represented by squares

Chance nodes – typically represented by circles

End nodes – typically represented by triangles

V. CODE

Code of Decision Tree

```
In [1]: from sklearn.datasets import load_iris
        iris = load_iris()
```

```
In [2]: import pandas as pd
        iris_data = pd.read_csv('sadi.csv')
        iris_data.head()
```

Out[2]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
In [3]: from sklearn import tree
        clf = tree.DecisionTreeClassifier()
        clf = clf.fit(iris.data, iris.target)
```

```
In [4]: import graphviz
        dot_data = tree.export_graphviz(clf, out_file=None,feature_names=iris.feature_names,class_names=iris.target_names,filled=True, rounded=True,special_characters=True)
        graph = graphviz.Source(dot_data)
```

VI. ADVANTAGES

A. Are simple to understand and interpret. People are able to understand decision tree models after a brief explanation.

B. Help determine worst, best and expected values for different scenarios.

C. Use a white box model. If a given result is provided by a model.

D. Can be combined with other decision techniques.

VII. DISADVANTAGES

A.They are unstable, meaning that a small change in the data can lead to a large change in the structure of the optimal decision tree.

B.They are often relatively inaccurate.

C.Calculations can get very complex, particularly if many values are uncertain and/or if many outcomes are linked.

VIII. 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.

Assignment No: 04
Report On [K- Nearest Neighbors]

CSE-0408 Summer 2021

Khaled Saifullah Sadi
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
mdsadi4@gmail.com

***Abstract*—K-Nearest Neighbors method is one of methods used for classification which calculate a value to find out the closest in distance. In this Assignment we are going to implement K-Nearest Neighbors using Jupyter Notebook.**

***Index Terms*—Machine Learning, Python, K nearest neighbors.**

I. Introduction

The K-Nearest-Neighbors (KNN) is a nonparametric classification algorithm, i.e. it does not make any presumptions on the elementary dataset.It is known for it's simplicity and effectiveness. It is a supervised learning algorithm. A labeled training dataset is provided where the data points are categorized into various classes, so that the class of the unlabeled data can be predicted.In Classification, different characteristics determine the class to which the unlabeled data belongs. KNN is mostly used as a classifier.It is used to classify data based on closest or neighbouring training examples in a given region.

In this assignment, we will implement another widely used machine learning classification technique called K-nearest neighbors (KNN). Our focus will be primarily on how does the algorithm work and how does the input parameter affects the output/prediction.

II. LITERATURE REVIEW

Along the years, a great effort was done in the scientific community in order to solve or mitigate the imbalanced dataset problem. Specifically for KNN, there are several balancing methods based on this algorithm. This section will provide a bibliographic review about the KNN and its derivate algorithms for dataset balancing. A

III. KNN ALGORITHM

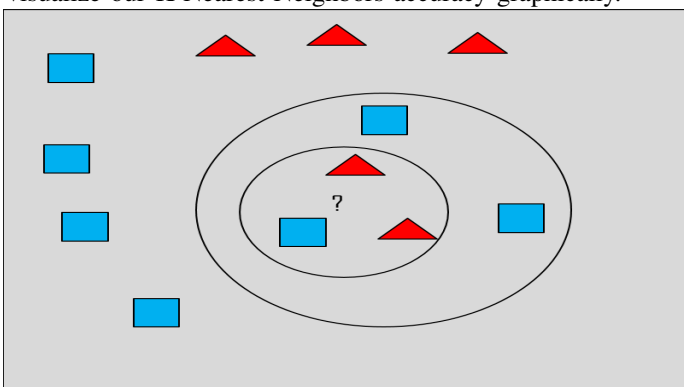
We can implement a KNN model by following the below steps:

1) Load the data
2) Initialise the value of k
3) For getting the predicted class, iterate from 1 to total number of training data points
   a) Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
   b) Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
   c) Sort the calculated distances in ascending order based on distance values
   d) Get top k rows from the sorted array
   e) Get the most frequent class of these rows
   f) Return the predicted class

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.



KNN

IV. ADVANTAGES

A. No Training Period: KNN is called Lazy Learner.It does not learn anything in the training period. It does not derive any discriminative function from the training data.

B. Since the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm.

C. KNN is very easy to implement. There are only two parameters required to implement KNN i.e. the value of K and the distance function.

V. DISADVANTAGES

A. Does not work well with large dataset.

B. Does not work well with high dimensions.

C. Need feature scaling: We need to do feature scaling before applying KNN algorithm to any dataset.

D. Sensitive to noisy data, missing values and outliers: KNN is sensitive to noise in the dataset.

VI. CODE

Here code of KNN

```
In [2]: dataset = pd.read_csv("data.csv")
        dataset.head()
```

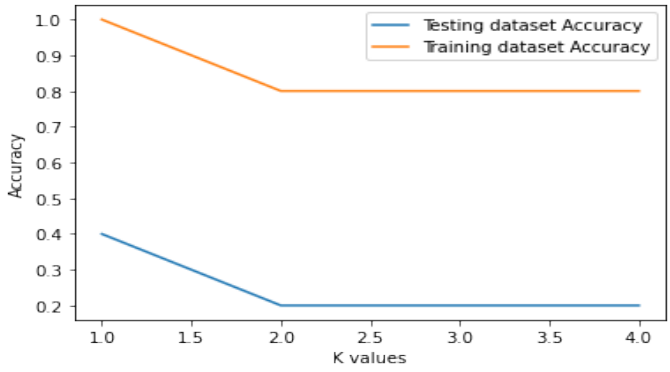
Out[2]:

	Gender	Age	Salary	Purchased
0	Male	19	19000	No
1	Male	35	20000	No
2	Female	26	29000	Yes
3	Female	27	43000	No
4	Male	19	50000	Yes

```
In [3]: dataset.info()
Out[3]: <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 10 entries, 0 to 9
        Data columns (total 4 columns):
         #   Column     Non-Null Count  Dtype
        ---  ------     --------------  -----
         0   Gender     10 non-null     object
         1   Age        10 non-null     int64
         2   Salary     10 non-null     int64
         3   Purchased  10 non-null     object
        dtypes: int64(2), object(2)
        memory usage: 448.0+ bytes

In [4]: X = dataset.drop(['Purchased'], axis = 'columns')
        Y = dataset['Purchased']

In [5]: from sklearn.preprocessing import LabelEncoder #Labeling string with number

In [6]: new_Gender = LabelEncoder()
```

Code

Output of this code



Output

VII. CONCLUSION

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

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.