Final Report-03
Decision tree

CSE-0408 Summer 2021

Durjoy Majumder (UG02-46-17-002)
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
sherajuddawlasumon@gmail.com

***Abstract*—Decision Trees are considered to be one of the most popular approaches for representing classifiers. Researchers from various disciplines such as statistics, machine learning, pattern recognition, and Data Mining have dealt with the issue of growing a decision tree from available data. This paper presents an updated survey of current methods for constructing decision tree classifiers in a top-down manner. The chapter suggests a unified algorithmic framework for presenting these algorithms and describes various splitting criteria and pruning methodologies.**

***Index Terms*—Python, Decision Tree, data, classifiers.**

I. Introduction

Definition: 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.

II. LITERATURE REVIEW

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree.

The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
Step-3: Divide the S into subsets that contains possible values for the best attributes.
Step-4: Generate the decision tree node, which contains the best attribute.
Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

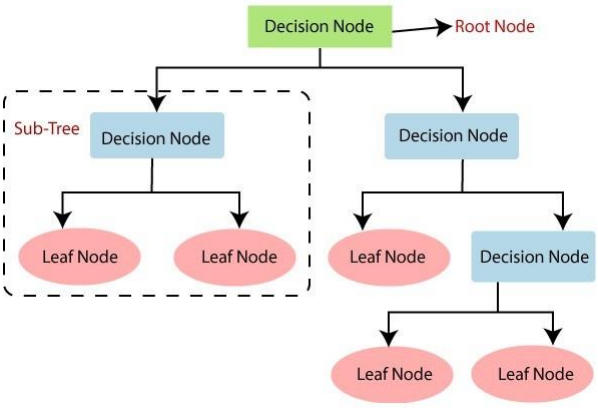

Fig. 1. Diagram explains the general structure of a decision tree:

III. Proposed Methodology

Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:

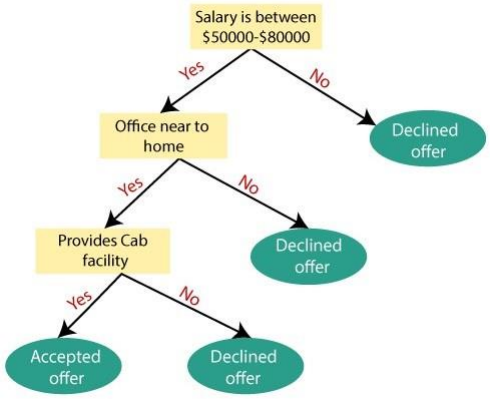

Fig. 2. Proposed Methodology

IV. Data Pre-Processing Step:

I have pre-processed the data. Where I have loaded the dataset, which is given as:

Index	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0
10	15570769	Female	26	80000	0
11	15606274	Female	26	52000	0
12	15746139	Male	20	86000	0
13	15704987	Male	32	18000	0
14	15628972	Male	18	82000	0
15	15697686	Male	29	80000	0

Fig. 3. Data Pre-Processing Step:

V. Visualizing the Test Set Result:

Visualization of test set result will be similar to the visualization of the training set except that the training set will be replaced with the test set. As we can see in the above image that there are some green data points within the purple region and vice versa. So, these are the incorrect predictions which we have discussed in the confusion matrix.

Acknowledgment

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

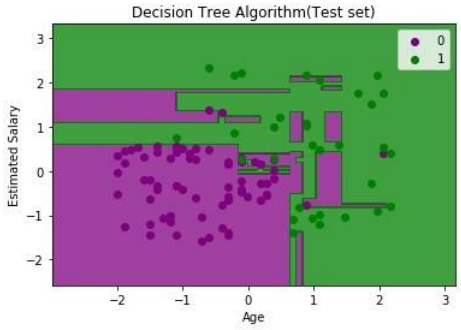

Fig. 4. Visualizing the test set result

References

[1] Shoewu, O., & Idowu, O. A. (2012). Development of attendance management system using biometrics. The Pacific Journal of Science and Technology, 13(1), 300-307.

[2] Arulogun, O. T., Olatunbosun, A., Fakolujo, O. A., & Olaniyi, O. M. (2013). RFID-based students attendance management system. International Journal of Scientific & Engineering Research, 4(2), 1-9.

[3] Nainan, S., Parekh, R., & Shah, T. (2013). RFID technology based attendance management system. arXiv preprint arXiv:1306.5381.

[4] Bhalla, V., Singla, T., Gahlot, A., & Gupta, V. (2013). Bluetooth based attendance management system. International Journal of Innovations in Engineering and Technology (IJIET), 3(1), 227-233.

[5] Chintalapati, S., & Raghunadh, M. V. (2013, December). Automated attendance management system based on face recognition algorithms. In 2013 IEEE International Conference on Computational Intelligence and Computing Research (pp. 1-5). IEEE. .

Final Report-04
K-Nearest Neighbors (KNN)

CSE-0408 Summer 2021

Durjoy Majumder (UG02-46-17-002)
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
sherajuddawlasumon@gmail.com

***Abstract*—Decision Trees are considered to be one of the most popular approaches for representing classifiers. Researchers from various disciplines such as statistics, machine learning, pattern recognition, and Data Mining have dealt with the issue of growing a decision tree from available data. This paper presents an updated survey of current methods for constructing decision tree classifiers in a top-down manner. The chapter suggests a unified algorithmic framework for presenting these algorithms and describes various splitting criteria and pruning methodologies.**

***Index Terms*—Python, KNN, Neighbor, Nearest.**

I. Introduction

KNN is a non-parametric and lazy learning algorithm. Non-parametric means there is no assumption for underlying data distribution. In other words, the model structure determined from the dataset. This will be very helpful in practice where most of the real world datasets do not follow mathematical theoretical assumptions. Lazy algorithm means it does not need any training data points for model generation. All training data used in the testing phase. This makes training faster and testing phase slower and costlier. Costly testing phase means time and memory. In the worst case, KNN needs more time to scan all data points and scanning all data points will require more memory for storing training data.

II. LITERATURE REVIEW

In this project, we are going to cover the following topics:

1. K-Nearest Neighbor Algorithm.
2. How does the KNN algorithm work?
3. Eager Vs Lazy learners.
4. How do you decide the number of neighbors in KNN?
5. Curse of Dimensionality.
6. Classifier Building in Scikit-learn
7. Pros and Cons.
8. How to improve KNN performance?
9. Conclusion.

KNN is a non-parametric and lazy learning algorithm. Non-parametric means there is no assumption for underlying data distribution. In other words, the model structure determined from the dataset. This will be very helpful in practice where most of the real world datasets do not follow mathematical theoretical assumptions. Lazy algorithm means it does not need any training data points for model generation. All training data used in the testing phase. This makes training faster and testing phase slower and costlier. Costly testing phase means time and memory. In the worst case, KNN needs more time to scan all data points and scanning all data points will require more memory for storing training data.

III. Proposed Methodology

In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbor algorithm. This is the simplest case. Suppose P1 is the point, for which label needs to predict. First, you find the one closest point to P1 and then the label of the nearest point assigned to P1.

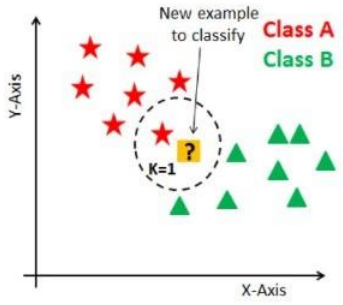

Fig. 1. How does the KNN algorithm work?

Suppose P1 is the point, for which label needs to predict. First, you find the k closest point to P1 and then classify points by majority vote of its k neighbors. Each object votes for their class and the class with the most votes is taken as the prediction. For finding closest similar points, you find the distance between points using distance measures such as

Euclidean distance, Hamming distance, Manhattan distance and Minkowski distance.

KNN has the following basic steps:

1. Calculate distance
2. Find closest neighbors
3. Vote for labels

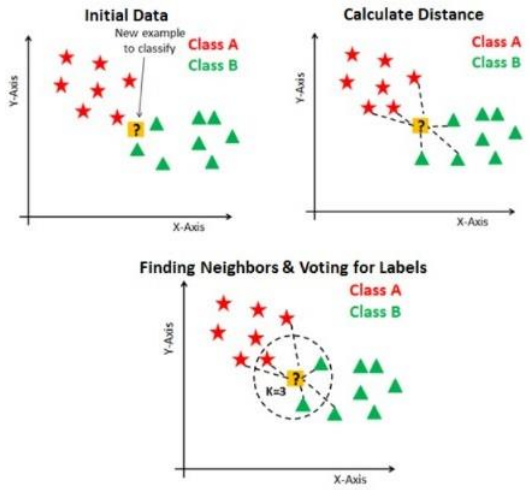

Fig. 2. How does the KNN algorithm work?

IV. Pros & Cons

Pros:

1. The training phase of K-nearest neighbor classification is much

2. faster compared to other classification algorithms.

3. There is no need to train a model for generalization, That is why KNN is known as the simple and instance-based learning algorithm.

4. KNN can be useful in case of nonlinear data. It can be used with the regression problem.

5. Output value for the object is computed by the average of k closest neighbors value.

Cons:

1. The testing phase of K-nearest neighbor classification is slower and costlier in terms of time and memory.

2. It requires large memory for storing the entire training dataset for prediction.

3. KNN requires scaling of data because KNN uses the Euclidean distance between two data points to find nearest neighbors.

4. Euclidean distance is sensitive to magnitudes.

5. The features with high magnitudes will weight more than features with low magnitudes.

6. KNN also not suitable for large dimensional data.

V. Conclusion:

In this project, you have learned the K-Nearest Neighbor algorithm; it's working, eager and lazy learner, the curse of dimensionality, model building and evaluation on wine dataset using Python Scikit-learn package. Also, discussed its advantages, disadvantages, and performance improvement suggestions.

Acknowledgment

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

References

[1] Shoewu, O., & Idowu, O. A. (2012). Development of attendance management system using biometrics. The Pacific Journal of Science and Technology, 13(1), 300-307.

[2] Arulogun, O. T., Olatunbosun, A., Fakolujo, O. A., & Olaniyi, O. M. (2013). RFID-based students attendance management system. International Journal of Scientific & Engineering Research, 4(2), 1-9.

[3] Nainan, S., Parekh, R., & Shah, T. (2013). RFID technology based attendance management system. arXiv preprint arXiv:1306.5381.