

Final Assignment Report: Decision Tree Classification in Python and K – Nearest Neighbor (KNN) Classification Using Python

CSE-0408 Summer 2021

Taskir Rahman Tasin
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
taskir.rahman72@gmail.com

Assignment Report Name : Decision Tree Classification in Python

Abstract—Among the learning algorithms, one of the most popular and easiest to understand is the decision tree induction. The popularity of this method is related to three nice characteristics: interpretability, efficiency, and flexibility. Decision tree can be used for both classification and regression kind of problem. Automatic learning of a decision tree is characterised by the fact that it uses logic and mathematics to generate rules instead of selecting them based on intuition and subjectivity. In this review, we present essential steps to understand the fundamental concepts and mathematics behind decision tree from training to building. We study criteria and pruning algorithms, which have been proposed to control complexity and optimize decision tree performance. Among the learning algorithms, one of the most popular and easiest to understand is the decision tree induction. The popularity of this method is related to three nice characteristics: interpretability, efficiency, and flexibility. Decision tree can be used for both classification and regression kind of problem. Automatic learning of a decision tree is characterised by the fact that it uses logic and mathematics to generate rules instead of selecting them based on intuition and subjectivity. In this review, we present essential steps to understand the fundamental concepts and mathematics behind decision tree from training to building. We study criteria and pruning algorithms and implement Decision Tree Classification in Python.

Index Terms—NumPy, Pandas and the standard python libraries

I. INTRODUCTION

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.

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy.

II. LITERATURE REVIEW

Machine learning is to learn machine on the basis of various training and testing data and determines the results in every condition without explicit programmed. One of the techniques of machine learning is Decision Tree. Different fields used Decision Tree algorithms and used it in their respective application. These algorithms can be used as to find data in replacement statistical procedures, to extract text, medical certified fields and also in search engines. Different Decision tree algorithms have been built according to their accuracy and cost of effectiveness. To use the best algorithm in every situations of decision making is very important for us to know

III. DEFINITION AND OBJECTIVES

Decision Tree is a tree shaped diagram used to determine a course of action.

Problems that Decision Tree can solve two different categories used on.

1. Classification
2. Regression

A. Classification

A classification tree will determine a set of logical if-then conditions to classify problems. For example, discriminating between three types of flowers based on certain features.

B. Regression

Regression tree is used when the target variable is numerical or continuous in nature. We fit a regression model to the target variable using each Of the independent variables.Each split is made based on the sum of squared error.

IV. PROPOSED METHODOLOGY

Each branch of the tree represents a possible decision,occurrence or reaction.

The basic idea behind any decision tree algorithm is as follows:

- 1.Select the best attribute using Attribute Selection Measures(ASM) to split the records.
- 2.Make that attribute a decision node and breaks the dataset into smaller subsets.
- 3.Starts tree building by repeating this process recursively for each child until one of the condition will match:
 - *All the tuples belong to the same attribute value.
 - *There are no more remaining attributes.
 - *There are no more instances.

A. Attribute Selection Measures

Attribute selection measure is a heuristic for selecting the splitting criterion that partition data into the best possible manner. It is also known as splitting rules because it helps us to determine breakpoints for tuples on a given node. ASM provides a rank to each feature(or attribute) by explaining the given dataset. Best score attribute will be selected as a splitting attribute (Source). In the case of a continuous-valued attribute, split points for branches also need to define. Most popular selection measures are Information Gain, Gain Ratio, and Gini Index.

B. Information Gain

Shannon invented the concept of entropy, which measures the impurity of the input set. In physics and mathematics, entropy referred as the randomness or the impurity in the system. In information theory, it refers to the impurity in a group of examples. Information gain is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values.

C. Gain Ratio

Information gain is biased for the attribute with many outcomes. It means it prefers the attribute with a large number of distinct values. For instance, consider an attribute with a unique identifier such as customer ID has zero info(D) because of pure partition. This maximizes the information gain and creates useless partitioning.

V. CODE PICTURE

Input , output and main code picture

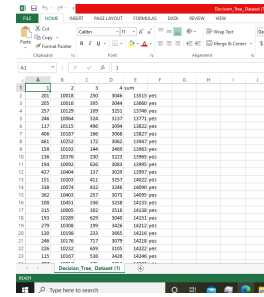


Fig. 1. Input

A. Main Code with output

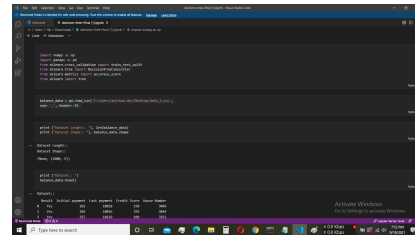


Fig. 2. Main code in python

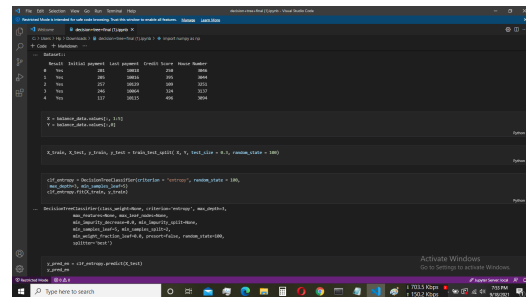


Fig. 3. main code with Output

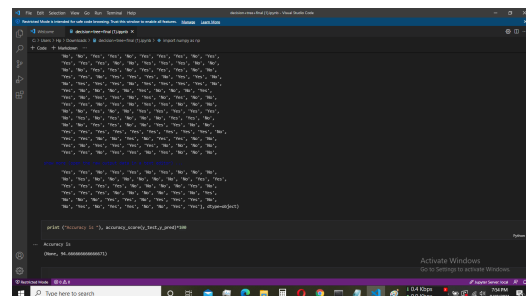


Fig. 4. main code with Output

VI. ADVANTAGES

- 1.Simple to understand, interpret and visualize.
- 2.Little effort required for data preparation.
- 3.Can handle both numerical and categorical data.
- 4.Non linear parameters don't effect its performance.

VII. DISADVANTAGES

1. Overfitting: Overfitting occurs when the algorithm captures noise in the data.
2. High variance : The model can get unstable due to small variation in data .
3. Small change in training data can result in large changes to the logic.
4. Large trees can be difficult to interpret.

VIII. CONCLUSION

Decision trees assist analysts in evaluating upcoming choices. The tree creates a visual representation of all possible outcomes, rewards and follow-up decisions in one document. Each subsequent decision resulting from the original choice is also depicted on the tree, so you can see the overall effect of any one decision. As you go through the tree and make choices, you will see a specific path from one node to another and the impact a decision made now could have down the road.

REFERENCES

- [1] Patel, H. H., Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. International Journal of Computer Sciences and Engineering, 6(10), 74-78.

End assignment 1

Assignment Report Name: K – Nearest Neighbor(KNN) Classification Using Python

Abstract—This report concerns one of the supervised ML classification algorithm-KNN(K Nearest Neighbors) algorithm. It is one of the simplest and widely used classification algorithms in which a new data point is classified based on similarity in the specific group of neighboring data points. This gives a competitive result.

Index Terms—Python’s famous packages NumPy and scikit-learn

IX. INTRODUCTION

KNN is a lazy learning, non-parametric algorithm. It uses data with several classes to predict the classification of the new sample point. KNN is non-parametric since it doesn’t make any assumptions on the data being studied, i.e., the model is distributed from the data.

What does it mean to say KNN is a lazy algorithm? It means it doesn’t use the training data points to make any generalisation. Which implies:

- 1.You expect little to no explicit training phase,
- 2.The training phase is pretty fast,
- 3.KNN keeps all the training data since they are needed during the testing phase.

Most data does not obey the typical theoretical assumptions, like when we consider a model like linear regression, which makes KNN crucial when studying data with little or no prior knowledge.

X. LITERATURE REVIEW

Multi-label learning originated from the investigation of text categorization problem, where each document may belong to several predefined topics simultaneously. In multi-label learning, the training set is composed of instances each associated with a set of labels, and the task is to predict the label sets of unseen instances through analyzing training instances with known label sets. In this paper, a multi-label lazy learning approach named ML-KNN is presented, which is derived from the traditional K-nearest neighbor (KNN) algorithm. In detail, for each unseen instance, its K nearest neighbors in the training set are firstly identified. After that, based on statistical information gained from the label sets of these neighboring instances, i.e. the number of neighboring instances belonging to each possible class, maximum a posteriori (MAP) principle is utilized to determine the label set for the unseen instance. Experiments on three different real-world multi-label learning problems, i.e. Yeast gene functional analysis, natural scene classification and automatic web page categorization, show that ML-KNN achieves superior performance to some well-established multi-label learning algorithms.

XI. KNN IS A SUPERVISED LEARNING ALGORITHM

A supervised machine learning algorithm is one that relies on labelled input data to learn a function that produces an appropriate output when given unlabeled data.

In machine learning, there are two categories

1. Supervised Learning

2. Unsupervised Learning

In supervised learning, you train your data on a labelled set of data and ask it to predict the label for an unlabeled point. For example, a tumour prediction model is trained on many clinical test results which are classified either positive or negative. The trained model can then predict whether an unlabeled test is positive or negative.

It works just like we’d do it – a teacher or a parent would teach a child new things. If a teacher wants the child to learn how an elephant looks like, he will show the child pictures of elephants, and then pictures of animals which are not elephants like zebras and monkeys.

When we see an elephant, we shout, “elephant!” when it’s not an elephant; we shout, “no, not an elephant!” After the teacher does this for a while with the kid, and he shows a child a picture and asks “elephant?” and the child will (mostly) correctly say “elephant!” or “no, not elephant!” depending on the picture. That is supervised learning. When we substitute the child with a computer, it becomes supervised machine learning.

We train it using the labelled data already available to us. In a dataset consisting of observation (x, y), we want to learn a function $g: X \rightarrow Y$ so that with X, we can use $g(x)$ to

predict corresponding output Y.

XII. WHERE TO USE KNN

KNN can be used in both regression and classification predictive problems. However, when it comes to industrial problems, it's mostly used in classification since it fairs across all parameters evaluated when determining the usability of a technique

1.Prediction Power

2.Calculation Time

3.Ease to Interpret the Output

KNN algorithm fairs across all parameters of considerations. But mostly, it is used due to its ease of interpretation and low calculation time.

XIII. PROPOSED METHODOLOGY

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

1.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.

2.Sort the calculated distances in ascending order based on distance values

3.Get top k rows from the sorted array

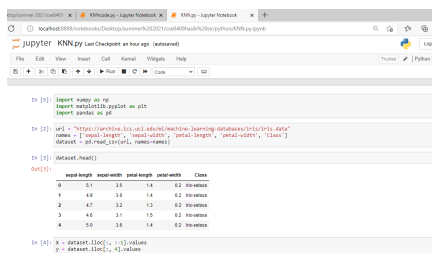
4.Get the most frequent class of these rows

5.Return the predicted class

XIV. CODE PICTURE

Input , output and main code picture

A. Input Table



```
In [10]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

In [11]: url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']
dataset = pd.read_csv(url, names=names)

In [12]: dataset.head()
```

| | sepal-length | sepal-width | petal-length | petal-width | Class |
|---|--------------|-------------|--------------|-------------|--------|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | Setosa |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | Setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | Setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | Setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | Setosa |

```
In [13]: X = dataset.iloc[:, :4].values
y = dataset.iloc[:, 4].values
```

Fig. 5. Example of given input table

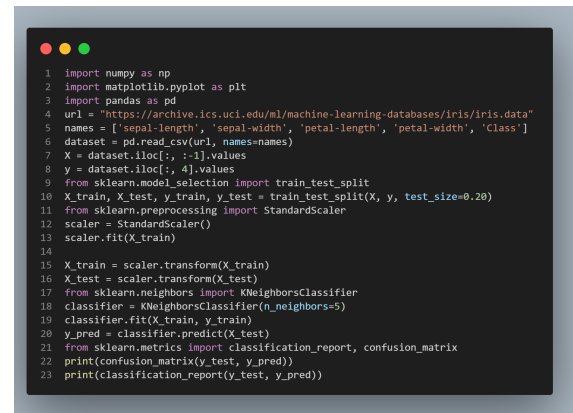
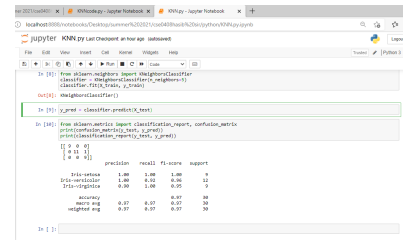


Fig. 6. Main code in python



```
In [14]: from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

[[ 0  0]
 [ 0 15]]

precision    recall    f1-score   support

Setosa:       1.00      1.00      1.00         14
Versicolour:  0.86      0.86      0.86         15
Virginica:     0.80      0.80      0.80         15

accuracy:    0.97      0.97      0.97         44
avg prec:    0.87      0.87      0.87         44
weighted avg: 0.87      0.97      0.97         44
```

Fig. 7. Output

B. Main Code

C. Output

XV. ADVANTAGE

- 1.Quick calculation time
- 2.Simple algorithm – to interpret
- 3.Versatile – useful for regression and classification
- 4.High accuracy – you do not need to compare with better-supervised learning models
- 5.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.

XVI. DISADVANTAGE

- 1.Accuracy depends on the quality of the data
- 2.With large data, the prediction stage might be slow
- 3.Sensitive to the scale of the data and irrelevant features
- 4.Require high memory – need to store all of the training data
- 5.Given that it stores all of the training, it can be computationally expensive

XVII. CONCLUSION

KNN algorithm is one of the simplest classification algorithm. Even with such simplicity, it can give highly competitive results. KNN algorithm can also be used for regression problems. The only difference from the discussed methodology will be using averages of nearest neighbors rather than voting

from nearest neighbors. We covered the main concepts behind the K Nearest Neighbor algorithm. Take different datasets and get familiar with this algorithm.

ACKNOWLEDGMENT

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

REFERENCES

- [1] Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning." *Pattern recognition* 40.7 (2007): 2038-2048.