Arsal Abbasi[1]

Contents

Abstract: This paper gives a detailed introduction of decision trees, including their construction, types (regression and classification), and implementation in Jupyter Notebook with the scikit-learn library. Decision trees are an important tool in machine learning and data analysis, allowing models to effectively categorize and predict outcomes. For continuous or ordered discrete dependent variables, regression trees are employed, whereas classification trees are utilized for categorical dependent variables. The implementation of decision trees in Jupyter Notebook using scikit-learn provides an easy and accessible approach to study this powerful tool for data analysis and machine learning.
Keywords - *Decision trees*, *Regression trees, Classification trees*

[1] arsal.abbasi@stud.hshl.de

1 Motivation

Making the right decision is critical for success in all areas of work. People make decisions in various ways based on their experiences, new research, and personal judgment. The expectation is that with a rising number of solved instances and fresh research, judgements will become more dependable. However, this also means that individuals and groups making decisions face increased complexity because they can no longer handle massive amounts of data. As a result, reliable decision support strategies are required to assist people in making informed decisions. These strategies can help with data processing and provide better insights into various decision-making possibilities. Decisions play a very prominent and vital role in many areas and applications. Decision trees have been successfully used for many decision making purposes [Po02].

In the area of machine learning, decision trees are a strong and popular technique. They are a well-liked option for both novices and specialists due to their adaptability, intuitiveness, and simplicity of understanding. Decision trees are particularly helpful for managing huge datasets [KS08] with numerous characteristics and may be utilized for both classification and regression tasks.

Decision trees interpret-ability is one of its main benefits [KS08] . Decision trees, as opposed to other machine learning algorithms like neural networks or support vector machines, give a precise and succinct explanation of how they came to their conclusions. This interpretability can be crucial in industries like healthcare, where it's critical to be able to comprehend and articulate the rationale behind a diagnosis or recommended course of treatment.

The fact that decision trees can handle both category and numerical data is another benefit. Their suitability for a wide range of applications—from forecasting consumer behavior in marketing to seeing trends in financial data—results from this. Decision trees are resistant to noisy datasets because they can deal with missing values and outliers.

Determination trees do have certain drawbacks, though. The issue of overfitting, when the model grows too complicated and begins to match the noise in the data instead of the underlying patterns, is one of the main problems with decision trees. Poor performance on fresh, untested data may result from this.Several strategies have been developed to overcome this problem, such as pruning, which purges the tree of unneeded branches, or ensemble methods, which aggregate numerous decision trees to enhance performance.

Despite these difficulties, decision trees are still a common and useful technique in machine learning. They are extensively employed in industries including marketing, banking, and healthcare, and their study is still ongoing. Decision trees will undoubtedly continue to be a vital part of the toolkit for years to come as machine learning advances and expands.

2 General overview of Decision Trees

Decision trees is a supervised learning technique that has been widely used to develop classification models because they closely mirror human thinking and are simple to grasp [Ko13]. A decision tree is a method of categorizing data based on certain characteristics. It mimics a tree, with a root at the top and several branches leading to various outcomes. The initial point has no connections to the other points, whereas the other points have one link to the starting point. points having connections are referred to as internal or test nodes, whilst locations without connections are referred to as leaves. Each internal node divides the data into several groups based on a unique feature. This trait is generally dependent on a single property, such as color or size. If the characteristic is a number, such as height, the groups will be determined by a range of values [RM05].

Each leaf is allocated to a class that represents the best goal value. Alternatively, the leaf might include a probability vector showing the likelihood of the target characteristic having a specific value. Instances are identified by navigating them from the tree's root to a leaf, based on the results of the tests along the way [RM05].

Figure 1 represents one such example in which a decision tree reasons if a potential customer will respond to a direct mailing or not [RM05]. The classifier indicated in this example is a decision tree that lets an analyst forecast a potential customer's response by sorting them down the tree. When it comes to direct mailing, the decision tree also provides insight into the behavioral features of the entire group of prospective customers. Each node in the decision tree is labeled with the attribute being tested, and its accompanying branches are labeled with their respective values. As a result, the decision tree is useful in assessing data and forecasting customer behavior [RM05].

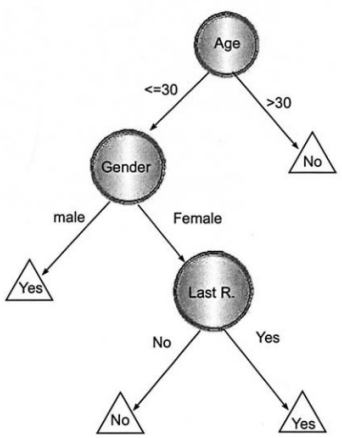

Fig. 1: Decision tree presenting a response to direct mailing [RM05]

The process of finding a solution using decision trees begins with the preparation of a set of solved cases. The entire set is then divided into (1) a training set for decision tree induction and (2) a testing set for checking the accuracy of an obtained answer [Po02]. A series of solved cases are generated in order to determine a solution using decision trees. The set is then separated into two sections: training and testing. The training set is used for creating the decision tree, whereas the testing set is used to validate the solution. Each example in the set includes input data (such as a car's color) and output data (such as whether the car is new or used). To form the decision tree, specific values are assigned to each input data (such as red or blue for color). Each input data is represented by an attribute node or a test node in the tree. The decision tree helps make decisions based on the input data.[Po02].

The decision trees are very easy to interpret [Po02]. Using the example given earlier in Fig.1, two things can be interpreted easily:

- Customers aged more than 30 years are not likely to respond to direct mailing.
- Customers who are less than 30 years and are male are likely to respond to direct mailing.

Using the same example, one thing can be said that as long as there is a mixture of data; male and female or last response, there is room to split further to try and get it to be only one category. This known as the purity of the node, an impure node can be branched further to improve purity. [Da].

There are methods for splitting decision trees. Some of which are entropy method, information gain method and gini index [Da]. All of these methods assist in finding the best attribute in order to do data splitting, this helps achieve accurate classification. The Entropy and Information Gain method focuses on a node's purity and impurity. The Gini Index calculates the likelihood of a random instance being misclassified when it is chosen at random. The lower the gini index, the smaller the chance of misclassification [Da]. Formula to find gini index is giving below, where j represents the number of classes and P(i) represents the number of observations in the node [Da].

$$Gini = 1 - \sum_{i=1}^{j} P(i)^2$$

2.1 Construction

A decision tree is created by reviewing a set of training objects using the "divide and conquer"method. The method begins by determining if all of the objects in the set belong to the same decision class. If they do, a tree consists of a single node, which is a leaf with the

necessary decision. If not, an attribute with at least two different decision classes is chosen, and the collection of objects is separated according to the category of the chosen attribute [Po02]. The chosen attribute becomes an attribute (test) node in a growing decision tree, and the inciting operation is repeated on the remaining objects with respect to the division for each branch from that node until a leaf (a decision) is reached. This procedure is repeated recursively, with each attribute chosen to maximize the separability of cases with various outcomes, until a tree is built that can aid in decision making depending on the qualities of new examples[Po02]. To help understand this phenomena better, imagine there are a bunch of examples with their respective outcomes, like whether a customer responds to direct mailing or not. A decision tree is a method for analyzing all of these samples and determining which features are most relevant in predicting the outcome. To construct a decision tree, first examine all of the examples to check if they have the same outcome. If they do, you just create a leaf node with that result. If the results of the outcomes differ, you select one attribute (such as age or gender) and see if it separates the examples into groups with distinct outcomes. If it does, you create a test node for that attribute and divide the examples accordingly. The process is then repeated for each group of samples with the remaining qualities until you reach a leaf node with a decision. With this approach, you can create a tree that can help you in making decisions depending on the characteristics of your examples.

2.2 Types

Over the years decision trees have gained a lot popularity and the use of decision in trees in supervised machine learning has increased, therefore numerous decision tree algorithms have been developed over the years [Ko13]. The most common types of decision trees are regression trees and classification trees. These types are used to solve regressing and classification problems. Classification and Regression Trees or CART for short is an acronym introduced by Leo Breiman to refer to Decision Tree algorithms that can be used for classification or regression predictive modeling problems [Br18].

2.2.1 Classification Trees

Classification trees are used for classification problems. In a classification problem, we get data on a variable Y with many values, as well as predictor variables X1, X2,... Xp. The goal is to find a model that can predict Y given new X values. One method is to split the space of X variables into k distinct sets (where k is the number of possible Y values), so that if a new collection of X values corresponds to one of these sets, we can expect that Y will take on a definite value. Traditional approaches such as linear discriminant analysis or nearest neighbor classification can be employed, however they may be difficult to comprehend if the predictor variables are many. Classification tree approaches, which split the data into rectangular sets depending on one predictor variable at a time, are an alternate approach.

This can make the sets simpler to comprehend, especially if the predictor variables are many [Lo11]. Figure 2 shows an example with three classes and two X variables. The left panel shows the data points and divisions, while the right panel depicts the decision tree structure. The tree structure has the benefit of being applicable to any number of variables, whereas the plot on the left can only handle two [Lo11].

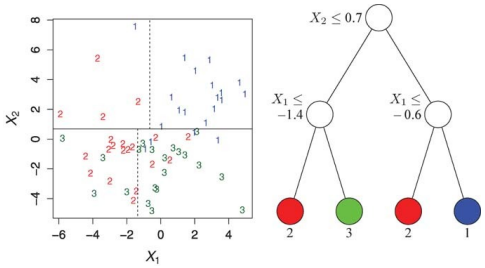

Fig. 2: Partitions (left) and decision tree structure (right) for a classification tree model [Lo11]

2.2.2 Regression Trees

Regression trees are like classification trees, the difference is that they do not have classes [Ti04]. In regression trees, the Y variable takes ordered values, and a regression model is fitted to each node to provide predicted Y values [Lo11]. To put in simpler words, regression trees help to predict a value that can be continuous or ordered (like a number), and predict the way we measure how well our prediction works is by looking at the difference between our predicted value and the actual value.

3 Implementation

In this section Decision Trees is implemented using the example data 'Iris'. The tool used in this implementation is Jupyter Notebook using scikit-learn library. Jupyter Notebook is a cell-based programming environment that allows you to write and execute code, see the output of the code directly below, include hyperlinks, take notes, and include images in between. In this environment, Python code can be executed, and other Python packages (pandas, plotly, numpy, and so on) can be imported to provide access to advanced data analysis and machine learning capabilities. Python code can be concealed behind interactive widgets, allowing students to work with powerful Python tools in a menu-based environment without ever seeing the code [BF21].

Iris data set has a total of six features, four of which are independent (SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalwidthCm) and one of which is dependent or target variable (Species). All independent features have non-zero float values, and the target variable has class labels: setosa, versicolor, and virginica. To visualize the data in scatter

graph Matplotlib and seaborn as a python library were used. The iris data is represented in figure 3 in scatter graph. 3 different classes are classified with different color and names.

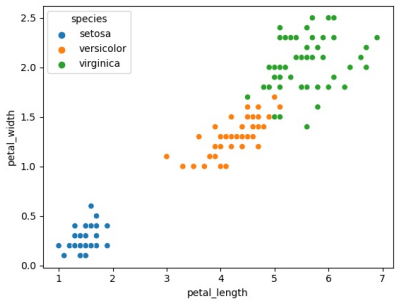

Fig. 3: Visualization of iris data

Figure 4 represents the implemented decision tree. The tree was visualized using Graphiz library. The data was split into two parts. 70% data was used for training and 30% was used for testing. The accuracy of the model was calculated to be 100%. The top node of the tree represents the complete data set, which is divided into two groups based on the "petal length (cm)"feature value. If the petal length is less than or equal to 2.45 cm, the species predicted is setosa; otherwise, the data set is divided further according on the "petal width (cm)"feature. If the petal width is less than or equal to 1.75 cm, the species is versicolor; otherwise, the species is virginica. The gini score is used to quantify the impurity of each subset and to guide the selection of features to split the data on, and each leaf node of the tree indicates a final conclusion on the species prediction.

Complete code of implementation can be viewed here.

4 Conclusion

In conclusion, decision trees are an effective tool for classification and regression applications. They produce a tree-like structure by recursively partitioning the data based on the predictor variables. Each tree leaf node is a prediction for the target variable. Decision trees are particularly useful for dealing with enormous data sets since they can handle both categorical and continuous data. They are also easily interpret able, making the decision-making process understandable. Overall, decision trees are a useful addition to the machine learning toolbox, and they can be implemented with a variety of tools, including scikit-learn in Python.

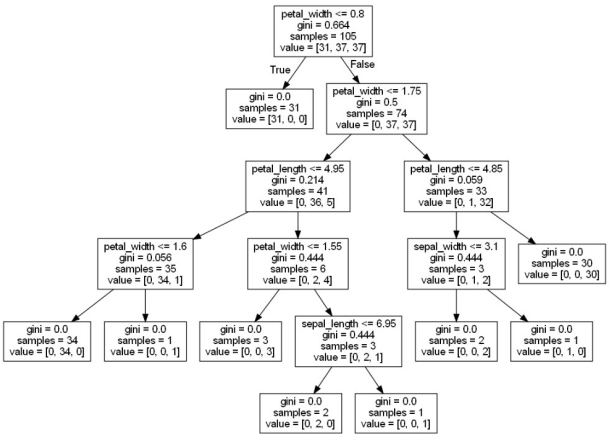

Fig. 4: Implemented decision tree

5 Declaration of Originality

I, Arsal Abbasi, herewith declare that I have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. I agree that my work may be checked by a plagiarism checker.

Date&Place - Arsal Abbasi

Bibliography

[BF21] Biehler, Rolf; Fleischer, Yannik: Introducing students to machine learning with decision trees using CODAP and Jupyter Notebooks. Teaching Statistics, 43:S133–S142, 2021.

[Br18] Brownlee, Jason: , How to implement the decision tree algorithm from scratch in python, 2018.

[Da] Dash, Shailey: , Decision Trees Explained—Entropy, Information Gain, Gini Index, CCP Pruning.. — towardsdatascience.com. https://towardsdatascience.com/decision-trees-explained-entropy-information-gain-gini-index-ccp-pruning-4d78070db36c. [Accessed 27-May-2023] gives a thorough overview of fundamental ideas linked to decision trees. It goes into the mathematical foundations and practical aspects of developing decision tree models, including entropy, information gain, the Gini index,

and cost-complexity pruning. The essay is a great resource for learning about the many components and approaches involved in decision tree creation and optimization.

[Ko13] Kotsiantis, Sotiris B: Decision trees: a recent overview. Artificial Intelligence Review, 39:261–283, 2013.

[KS08] Kingsford, Carl; Salzberg, Steven L: What are decision trees? Nature biotechnology, 26(9):1011–1013, 2008.

[Lo11] Loh, Wei-Yin: Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery, 1(1):14–23, 2011.

[Po02] Podgorelec, Vili; Kokol, Peter; Stiglic, Bruno; Rozman, Ivan: Decision trees: an overview and their use in medicine. Journal of medical systems, 26:445–463, 2002.

[RM05] Rokach, Lior; Maimon, Oded: Decision trees. Data mining and knowledge discovery handbook, pp. 165–192, 2005.

[Ti04] Timofeev, Roman: Classification and regression trees (CART) theory and applications. Humboldt University, Berlin, 54, 2004.