HO CHI MINH CITY UNIVERSITY TECHNOLOGY AND EDUCATION

◈◇◈◇◈

LECTURER

Le Van Vinh

Project 3

Decision Tree

GROUP MEMBERS:

Hồ Vũ Minh Đức - 17110022

Nguyễn Quang Minh - 17110051

SEMESTER: 1 – YEAR: 2020-2021

HO CHI MINH CITY - 2020

EVALUATION AND SCORE

EVALUATION:

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

SCORE:

WORD	NUMBER

INTRODUCTION

These day, modern world with the explosion of information technology, especially data, AI, NLP, Machine learning, Deep Learning concepts ... were born to analyze data, learn knowledge sources to help people make decisions from huge data sources.

This semester we are learning and acquiring knowledge about Decision Tree to learn more about machine learning.

This project helps us implement our knowledge about it.

During researching process, we also have some problems but luckily we can overcome the challenges. We are grateful for Mr. Vinh becase without his help and enthusiastic we may not achieve our sucess.

Thank You

Contents

I. Decision Tree

1.1 What is Decision Tree ?

In its simplest form, a decision tree is a type of flowchart that shows a clear pathway to a decision. In terms of data analytics, it is a type of algorithm that includes conditional ‘control’ statements to classify data. A decision tree starts at a single point (or ‘node’) which then branches (or ‘splits’) in two or more directions. Each branch offers different possible outcomes, incorporating a variety of decisions and chance events until a final outcome is achieved. When shown visually, their appearance is tree-like…hence the name!

Decision trees are extremely useful for data analytics and machine learning because they break down complex data into more manageable parts. They’re often used in these fields for prediction analysis, data classification, and regression. Don’t worry if this all sounds a bit abstract—we’ll provide some examples below to help clear things up. First though, let’s look at the different aspects that make up a decision tree.

1.2 What are the different parts of a Decision Tree ?

Decision trees can deal with complex data, which is part of what makes them useful. However, this doesn’t mean that they are difficult to understand. At their core, all decision trees ultimately consist of just three key parts, or ‘nodes’:

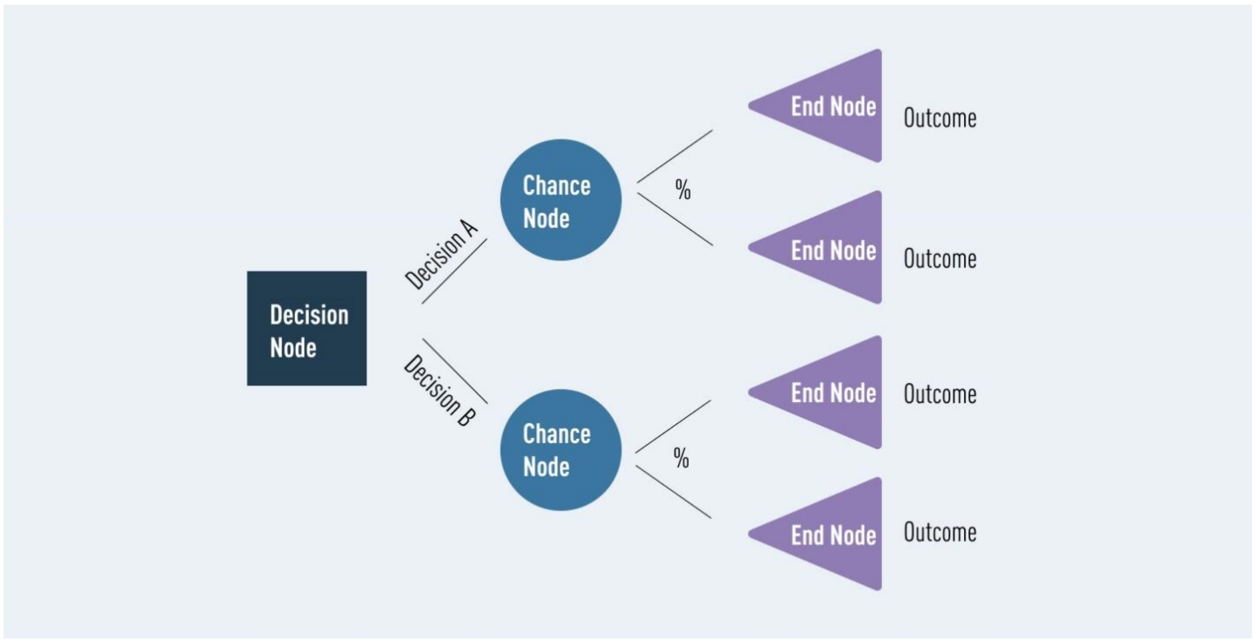
Connecting these different nodes are what we call ‘branches’. Nodes and branches can be used over and over again in any number of combinations to create trees of various complexity

1.3 Parts of a Decision Tree

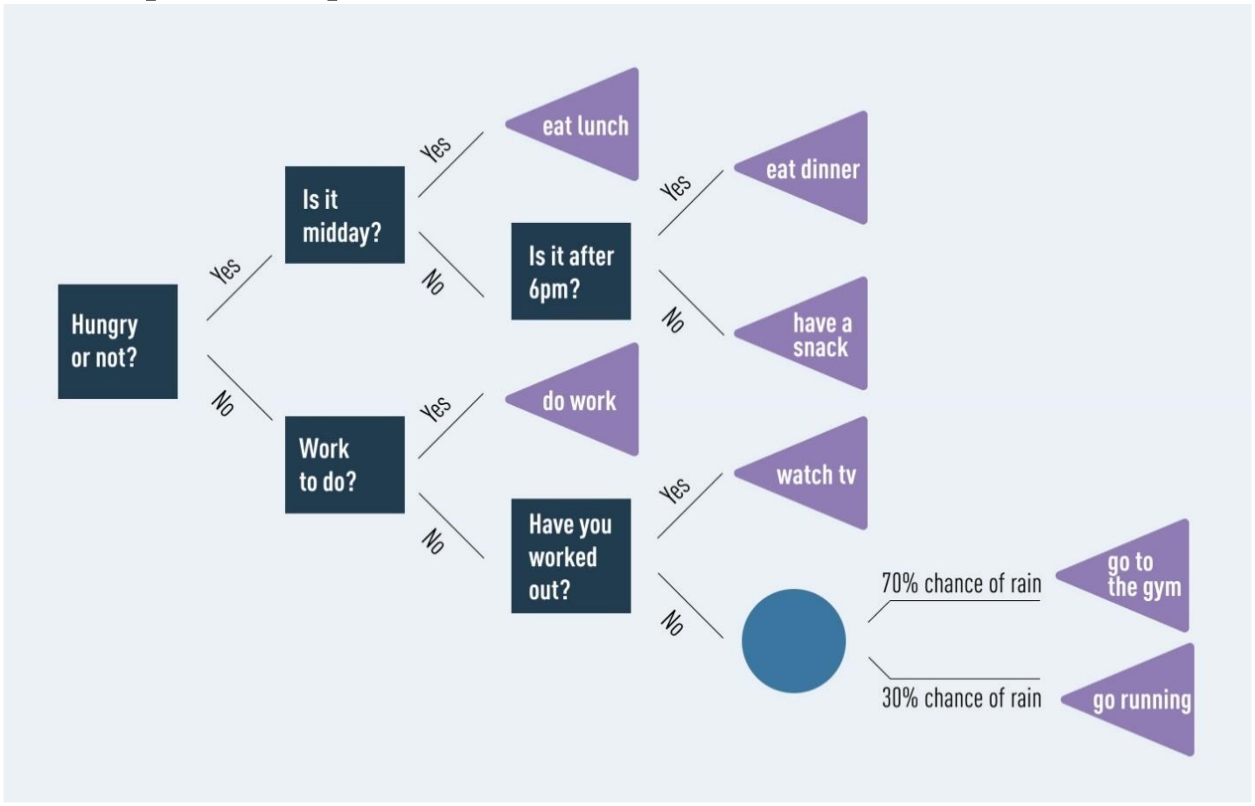
- Decision nodes: Representing a decision (typically shown with a square)

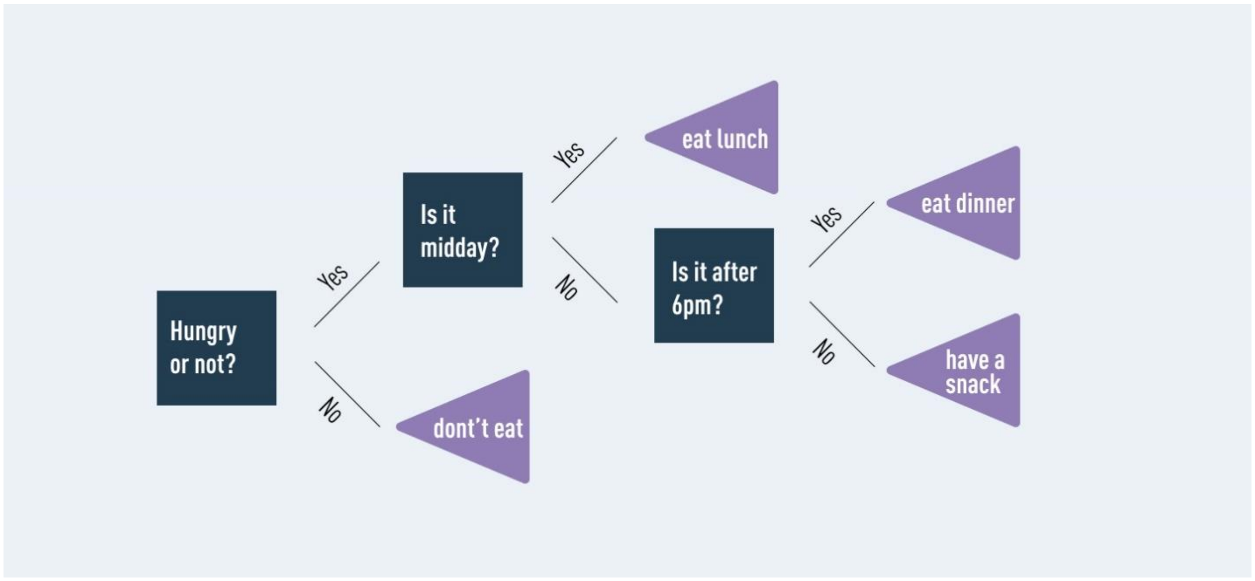
- Chance nodes: Representing probability or uncertainty (typically denoted by a circle)
- End nodes: Representing an outcome (typically shown with a triangle)

Luckily, a lot of decision tree terminology follows the below tree analogy, which makes it much easier to remember!



1.4 An example of a simple decision tree





II. Going Deeper

1.1 Pros and cons of decision tree

Used effectively, decision trees are very powerful tools. Nevertheless, like any algorithm, they're not suited to every situation. Here are some key advantages and disadvantages of decision trees.

Advantages of decision tree

- Good for interpreting data in a highly visual way.
- Good for handling a combination of numerical and non-numerical data.
- Easy to define rules, e.g. 'yes, no, if, then, else…'
- Requires minimal preparation or data cleaning before use.
- Great way to choose between best, worst, and likely case scenarios.
- Can be easily combined with other decision-making techniques.

Disadvantages of decision tree

- Overfitting (where a model interprets meaning from irrelevant data) can become a problem if a decision tree's design is too complex.
- They are not well-suited to continuous variables (i.e. variables which can have more than one value, or a spectrum of values).
- In predictive analysis, calculations can quickly grow cumbersome, especially when a decision path includes many chance variables.
- When using an imbalanced dataset (i.e. where one class of data dominates over another) it is easy for outcomes to be biased in favor of the dominant class.
- Generally, decision trees provide lower prediction accuracy compared to other predictive algorithms.

1.2 What are decision tree used for ?

Despite their drawbacks, decision trees are still a powerful and popular tool. They're commonly used by data analysts to carry out predictive analysis (e.g. to develop operations strategies in businesses). They're also a popular tool for machine learning and artificial intelligence, where they're used as training algorithms for supervised learning (i.e. categorizing data based on different tests, such as 'yes' or 'no' classifiers.)

Broadly, decision trees are used in a wide range of industries, to solve many types of problems. Because of their flexibility, they're used in sectors from technology and health to financial planning. Examples include:

A technology business evaluating expansion opportunities based on analysis of past sales data.
A toy company deciding where to target its limited advertising budget, based on what demographic data suggests customers are likely to buy.
Banks and mortgage providers using historical data to predict how likely it is that a borrower will default on their payments.
Emergency room triage might use decision trees to prioritize patient care (based on factors such as age, gender, symptoms, etc.)
Automated telephone systems guiding you to the outcome you need, e.g. 'For option A, press 1. For option B, press 2', and so on.
As you can see, there many uses for decision trees!.

1.3 Demo

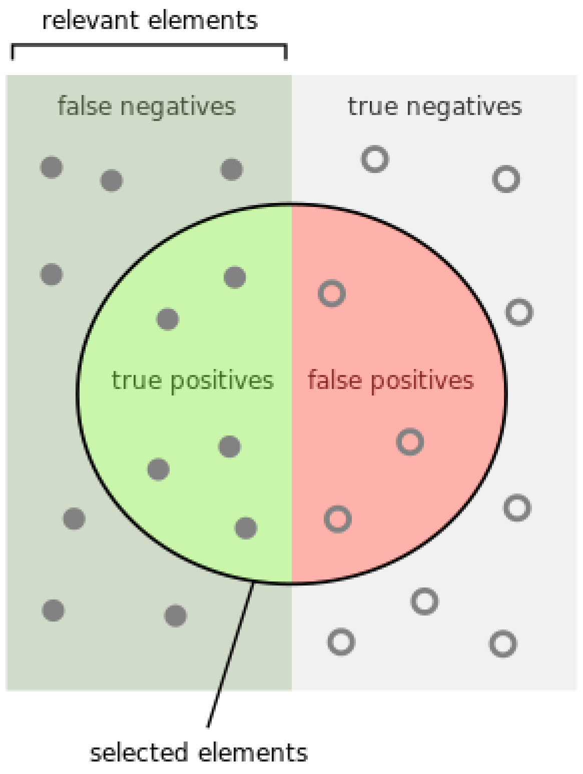
We take an example from Github
https://colab.research.google.com/github/Anny8910/Decision-Tree-Classification-on-Diabetes-Dataset/blob/master/Diabetes_set_(Decision_tree).ipynb

III. PRECISION AND RECALL ON EXCEL

1. Code for Precision And Recall

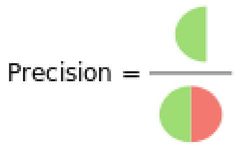
```
figure

plot(recall,precision)

grid on

title(sprintf('Average Recall/Precision = %.1f',ap))

numData = xlsread('precisionAndRecall.xlsx')

TNNumber = numData(:,1);

FNNumber = numData(:,2);

FPNumber = numData(:,3);

TPNumber = numData(:,4);

Precision = (TPNumber)/(TPNumber + FPNumber)

Recall = (TPNumber) / (TPNumber + FNNumber)

[Precision, Recall] = xlswrite('precisionAndRecallResult.xlsx')
```

2. How to calculate precision and recall



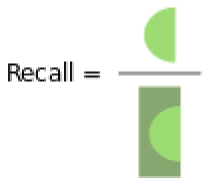
- **Calculate the precision**

$$Precision = \frac{tp}{tp + fp}$$



- **Calculate the recall**

$$Recall = \frac{tp}{tp + fn}$$



3. Result of experience

		Predicted	Actual	TN	FN	FP	TP
1.40325	1.630282	2	2	0	0	0	1
2.375414	0.333976	2	2	0	0	0	1
-0.69414	1.07507	2	2	0	0	0	1
1.64663	0.591603	2	2	0	0	0	1
1.239074	1.227641	2	2	0	0	0	1
0.019234	0.549755	1	2	0	1	0	0
0.674806	1.367474	2	2	0	0	0	1
1.256968	1.554522	2	2	0	0	0	1
3.683798	2.283916	2	2	0	0	0	1
3.077078	0.854407	2	2	0	0	0	1
-0.01242	-0.60377	1	1	1	0	0	0
3.276193	0.370308	2	2	0	0	0	1
1.544053	2.015946	2	2	0	0	0	1
0.952709	0.195884	1	2	0	1	0	0
1.536057	1.720715	2	2	0	0	0	1
0.846275	1.093037	2	2	0	0	0	1
0.906892	2.077522	2	2	0	0	0	1

2.117273	-0.47067	2	2	0	0	0	1
2.056776	0.851726	2	2	0	0	0	1
2.062894	0.094116	2	2	0	0	0	1
1.503623	3.181006	2	2	0	0	0	1
0.094385	1.618914	2	2	0	0	0	1
1.537929	2.034229	2	2	0	0	0	1
2.222676	0.206365	2	2	0	0	0	1
1.36667	0.648538	2	2	0	0	0	1
1.77602	0.795648	2	2	0	0	0	1
1.545164	1.823818	2	2	0	0	0	1
0.772419	0.791596	1	2	0	1	0	0
1.220404	1.526156	2	2	0	0	0	1
0.409538	-0.53886	1	1	1	0	0	0
1.666297	0.734613	2	2	0	0	0	1
0.139697	0.38231	1	2	0	1	0	0
0.198347	-0.18279	1	2	0	1	0	0
0.392876	1.380981	2	2	0	0	0	1
-1.20821	1.211488	2	1	0	0	1	0
2.078785	1.02511	2	2	0	0	0	1

1.243893	-0.00026	2	2	0	0	0	1
0.433804	1.845619	2	2	0	0	0	1
2.027724	1.262635	2	2	0	0	0	1
-0.28364	0.7757	1	2	0	1	0	0
0.923318	1.017167	2	2	0	0	0	1
0.818915	0.803503	1	2	0	1	0	0
1.239405	-0.31266	2	2	0	0	0	1
1.234644	0.785762	2	2	0	0	0	1
0.35134	0.376475	1	2	0	1	0	0
0.977462	0.265595	1	2	0	1	0	0
0.876341	0.132699	1	2	0	1	0	0
1.47078	0.599832	2	2	0	0	0	1
1.819949	-0.50198	2	2	0	0	0	1
1.831955	1.723172	2	2	0	0	0	1
0.35226	1.390045	2	2	0	0	0	1
1.058019	0.984979	2	2	0	0	0	1
0.089412	0.973922	1	2	0	1	0	0
0.164874	0.401377	1	2	0	1	0	0
0.994863	1.764014	2	2	0	0	0	1

2.149473	0.900087	2	2	0	0	0	1
0.422751	0.464102	1	2	0	1	0	0
1.278534	2.013539	2	2	0	0	0	1
0.830812	0.831422	1	2	0	1	0	0
1.838017	0.558228	2	2	0	0	0	1
0.183202	0.779685	1	2	0	1	0	0
1.024418	0.364055	2	2	0	0	0	1
1.414395	0.159904	2	2	0	0	0	1
1.825458	2.8945	2	2	0	0	0	1
2.158159	2.241623	2	2	0	0	0	1
1.064448	1.230651	2	2	0	0	0	1
-0.11869	0.057161	1	1	1	0	0	0
0.443274	0.350899	1	2	0	1	0	0
0.203814	0.867599	1	2	0	1	0	0
2.762843	1.593562	2	2	0	0	0	1
0.538299	0.000997	1	2	0	1	0	0
1.561058	-0.7474	2	2	0	0	0	1
0.855686	-0.08682	1	2	0	1	0	0
1.666458	1.250133	2	2	0	0	0	1

0.426363	1.293515	2	2	0	0	0	1
-0.0517	1.33876	2	2	0	0	0	1
-0.06678	0.902287	1	2	0	1	0	0
1.366145	1.137767	2	2	0	0	0	1
0.866969	0.642885	1	2	0	1	0	0
0.85296	1.646516	2	2	0	0	0	1
2.064483	-0.02127	2	2	0	0	0	1
1.218688	1.341272	2	2	0	0	0	1
1.148358	0.363468	2	2	0	0	0	1
2.190774	0.748835	2	2	0	0	0	1
0.396651	1.414588	2	2	0	0	0	1
1.522468	1.779318	2	2	0	0	0	1
1.626316	0.161771	2	2	0	0	0	1
0.817214	1.945494	2	2	0	0	0	1
1.161753	1.495107	2	2	0	0	0	1
0.125617	0.949101	1	2	0	1	0	0
0.139035	0.853584	1	2	0	1	0	0
1.078656	0.836795	2	2	0	0	0	1
1.541691	0.772669	2	2	0	0	0	1

2.939118	1.017284	2	2	0	0	0	1
0.499832	1.038468	2	2	0	0	0	1
1.140498	1.619547	2	2	0	0	0	1
0.938129	2.145233	2	2	0	0	0	1
-0.44977	1.350186	2	2	0	0	0	1
0.670775	0.842715	1	2	0	1	0	0
-0.34601	1.468893	2	2	0	0	0	1
-0.90839	-0.96003	1	1	1	0	0	0
-1.51488	-1.47424	1	1	1	0	0	0
-0.52539	-0.79425	1	1	1	0	0	0
-0.84647	-0.66151	1	1	1	0	0	0
-0.93241	-0.57113	1	1	1	0	0	0
-0.74238	-1.34558	1	1	1	0	0	0
-0.8693	-0.77531	1	1	1	0	0	0
-1.47074	-0.94968	1	1	1	0	0	0
-1.08117	-0.58697	1	1	1	0	0	0
-1.07303	-0.73192	1	1	1	0	0	0
-1.26601	-0.55106	1	1	1	0	0	0
-0.15895	-1.06597	1	1	1	0	0	0

-1.43786	-1.0736	1	1	1	0	0	0
-1.24191	-0.49611	1	1	1	0	0	0
-1.356	-2.06183	1	1	1	0	0	0
-1.58711	-1.25229	1	1	1	0	0	0
-1.09612	-1.6353	1	1	1	0	0	0
-1.13704	-1.19129	1	1	1	0	0	0
-0.23496	-0.67566	1	1	1	0	0	0
-1.12451	-0.58714	1	1	1	0	0	0
-1.53211	-1.50747	1	1	1	0	0	0
-0.19827	-1.23553	1	1	1	0	0	0
-0.38266	-0.93149	1	1	1	0	0	0
-1.11481	-1.14593	1	1	1	0	0	0
-1.75308	-0.84909	1	1	1	0	0	0
-1.22231	-0.80003	1	1	1	0	0	0
-1.07797	-1.46498	1	1	1	0	0	0
-0.86197	-1.08842	1	1	1	0	0	0
-1.13058	-2.06605	1	1	1	0	0	0
-0.77829	-0.42732	1	1	1	0	0	0
-0.80405	-1.31455	1	1	1	0	0	0

-1.62534	-1.60192	1	1	1	0	0	0
-1.47398	-1.12697	1	1	1	0	0	0
-1.37055	-1.71432	1	1	1	0	0	0
-1.25391	-1.01043	1	1	1	0	0	0
-1.16029	-1.28033	1	1	1	0	0	0
-0.99377	0.088889	1	1	1	0	0	0
-2.51459	-0.43077	1	1	1	0	0	0
-1.22851	-2.24844	1	1	1	0	0	0
-0.37878	-0.77934	1	1	1	0	0	0
-1.53335	-1.69907	1	1	1	0	0	0
-0.53314	-1.12753	1	1	1	0	0	0
-0.82484	-0.9178	1	1	1	0	0	0
-1.0145	-0.62613	1	1	1	0	0	0
-0.90877	-1.13652	1	1	1	0	0	0
-1.78253	-0.21185	1	1	1	0	0	0
-1.04227	-1.24047	1	1	1	0	0	0
-0.19803	-0.83624	1	1	1	0	0	0
-0.95083	-0.66763	1	1	1	0	0	0
-0.97931	-0.95741	1	1	1	0	0	0

-1.36708	-0.55952	1	1	1	0	0	0
-1.01541	-0.83839	1	1	1	0	0	0
-0.88383	-1.39207	1	1	1	0	0	0
-0.78681	-1.90269	1	1	1	0	0	0
-1.1864	-0.0707	1	1	1	0	0	0
-1.11823	-1.30227	1	1	1	0	0	0
0.011845	-0.94832	1	1	1	0	0	0
-2.12918	-0.71842	1	1	1	0	0	0
0.114723	-0.9432	1	1	1	0	0	0
-0.83122	-1.45236	1	1	1	0	0	0
-0.49997	-1.23386	1	1	1	0	0	0
-1.83208	-1.06244	1	1	1	0	0	0
-1.29502	-0.26052	1	1	1	0	0	0
-1.13903	-1.43041	1	1	1	0	0	0
-0.78864	-0.60767	1	1	1	0	0	0
-1.8351	-0.84569	1	1	1	0	0	0
-0.76418	-1.11693	1	1	1	0	0	0
-1.60642	-1.52849	1	1	1	0	0	0
-0.9669	-1.14207	1	1	1	0	0	0

-0.67382	-1.04335	1	1	1	0	0	0
-0.83647	-1.7347	1	1	1	0	0	0
-0.45868	-0.90391	1	1	1	0	0	0
-0.49696	-1.41115	1	1	1	0	0	0
-1.32545	-1.04712	1	1	1	0	0	0
-0.87147	-0.83189	1	1	1	0	0	0
-1.47219	-1.45233	1	1	1	0	0	0
-1.66089	-1.14413	1	1	1	0	0	0
-0.53759	-0.82497	1	1	1	0	0	0
-0.99998	-1.91793	1	1	1	0	0	0
-1.02746	-0.48201	1	1	1	0	0	0
-0.54444	0.212231	1	1	1	0	0	0
-0.70271	-0.5203	1	1	1	0	0	0
-0.8249	-1.15789	1	1	1	0	0	0
-0.37487	-0.78569	1	1	1	0	0	0
-0.53511	-1.51799	1	1	1	0	0	0
-0.88012	-0.06107	1	1	1	0	0	0
-1.34518	-0.52965	1	1	1	0	0	0
-1.32578	-0.60633	1	1	1	0	0	0

-0.40395	-1.43794	1	1	1	0	0	0
-1.80592	-0.84003	1	1	1	0	0	0
-1.01223	-1.27915	1	1	1	0	0	0
-1.97442	-1.15571	1	1	1	0	0	0
-0.48975	-1.285	1	1	1	0	0	0
-0.56914	-1.51287	1	1	1	0	0	0
-0.99942	-1.45437	1	1	1	0	0	0
-1.03542	-1.10495	1	1	1	0	0	0
-2.24314	-1.84943	1	1	1	0	0	0
-0.70941	-0.6962	1	1	1	0	0	0
-2.09622	-1.0589	1	1	1	0	0	0
-2.15964	-0.65042	1	1	1	0	0	0
Total				103	24	1	72

Actual class	**Predicted class**	
	Negative	Positive
Negative	TN	FP
Positive	FN	TP

- Precision's result

$$\boldsymbol{Precision} = \frac{tp}{tp + fp} = \frac{72}{72 + 1} = 0.986$$

- Recall's result

$$\boldsymbol{Recall} = \frac{tp}{tp + fn} = \frac{72}{72 + 24} = 0.75$$

IV. Preferences

- https://careerfoundry.com/en/blog/data-analytics/what-is-a-decision-tree/
- https://github.com/Anny8910/Decision-Tree-Classification-on-Diabetes-Dataset