BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

ĐỒ ÁN III – MACHINE LEARNING

ĐỀ TÀI:

TÌM HIỂU VỀ DECISION TREE VÀ THUẬT TOÁN ID3

Giảng viên hướng dẫn: ThS. Trần Nhật Quang

Sinh viên thực hiện:

15110100 – Nguyễn Thiện Phúc

15110028 – Hoàng Phước Đại

Thành phố Hồ Chí Minh, tháng 7 năm 2020

MỤC LỤC

LỜI MỞ ĐẦU

Scikit-learn, Tensorflow từ lâu trở thành thư viện phổ biến trong lĩnh vực máy học và nghiên cứu dữ liệu. Vì vậy, việc tìm hiểu và áp dụng thư viện này cho các thuật toán máy học đã học là điều cần thiết cho sinh viên Công Nghệ Thông Tin ở các trường Đại Học.

Dù Scikit-learn cung cấp đầy đủ công cụ, việc hiểu ý nghĩa từng đoạn mã là cần thiết và cũng là mục tiêu quan trọng nhất khi hoàn tất đề tài. Chỉ khi chúng ta thực sự hiểu thì mới hoàn tất được những yêu cầu nâng cao hơn sau này. Ngoài ra, nhóm cũng đã tìm hiểu thêm các kiến thức bên ngoài về tối ưu hóa bộ tham chiếu, giúp kết quả đạt được tốt hơn.

Em mong rằng những trải nghiệm tìm tòi này có thể làm cơ sở cho công việc sau này. Khi tương lai, trí tuệ nhân tạo sẽ là một xu thế của cuộc cách mạng 4.0 trong và ngoài nước thì nguồn nhân lực công nghệ có hiểu biết về máy học chắc chắn có được lợi thế nhất định.

Cam kết không đạo văn

Định nghĩa: "Đạo văn ở mức độ nghiên cứu khoa học sinh viên được hiểu là: sử dụng công trình hay tác phẩm của người khác, lấy ý tưởng của người khác, sao chép nguyên bản từ ngữ của người khác mà không ghi nguồn, sử dụng cấu trúc và cách lý giải của người khác mà không ghi nhận họ, lấy những thông tin chuyên ngành mà không ghi rõ nguồn gốc." – theo bài "Lỗi đạo văn trong nghiên cứu khoa học và cách phòng tránh" - Cộng đồng sinh viên kinh tế nghiên cứu khoa học (RCES).

"Đạo văn là hành vi thiếu trung thực về mặt học thuật và vi phạm đạo đức nghiêm trọng. Ở cấp độ sinh viên, đạo văn sẽ khiến kết quả nghiên cứu bị hủy bỏ, tùy thuộc vào mức độ nghiêm trọng của hành vi. Ở cấp độ nghiên cứu chuyên nghiệp, người đạo văn có thể bị buộc thôi việc, thu hồi công trình đã công bố hoặc hủy bỏ chức danh." –theo

bài “Lỗi đạo văn trong nghiên cứu khoa học và cách phòng tránh“ - Cộng đồng sinh viên kinh tế nghiên cứu khoa học (RCES).

“Trong các trường hợp sau đây người viết **không** dẫn nguồn bị coi là đạo văn:

- Người viết trắng trợn sử dụng toàn bộ công trình, code của ai đó thành của mình.
- Người viết sao chép cách phân bố, bố cục của các đoạn văn từ một nguồn duy nhất mà không chỉnh sửa.
- Mặc dù người viết đã giữ lại các nội dung quan trọng của nguồn, nhưng người đó vẫn sửa lại một chút về “diện mạo” của bài viết đó bằng cách thay đổi từ khóa hay câu cú.
- Người viết dành thời gian để chú giải các nguồn khác nhau và nối chúng lại với nhau, thay vì dành nổ lực tương tự cho công việc của mình.
- Người viêt “mượn hầu như toàn bộ” các thành quả trước đó của chính mình để phục vụ cho bài viết mới. Không có nhiều sự khác nhau giữa bài mới và bài cũ.

Trong các trường hợp sau đây người viết **có** dẫn nguồn bị coi là đạo văn:

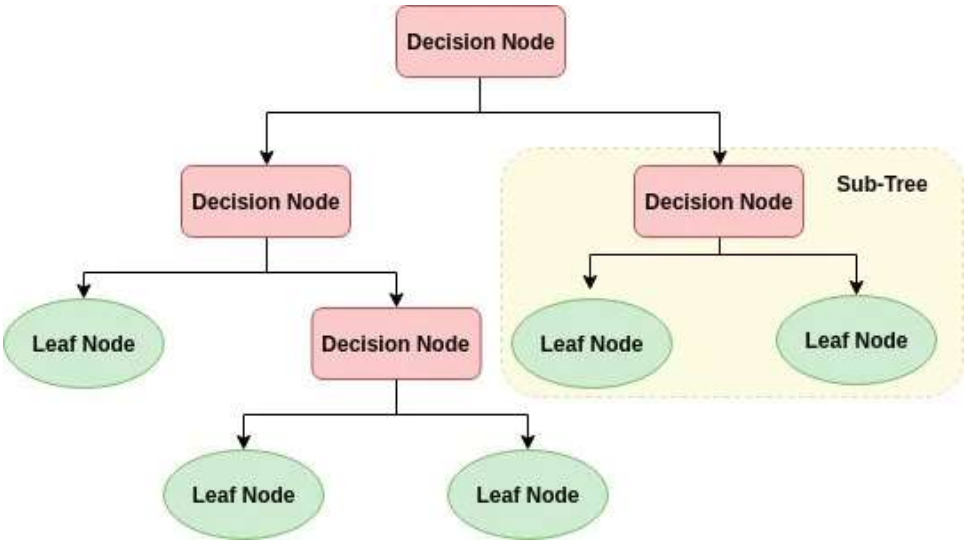
- Người viết dẫn tên tác giả nhưng lại sao lãng việc điền thông tin cụ thể để dẫn chứng về đoạn dẫn nguồn tham khảo như năm xuất bản, trang, chương, mục lục,…
- Người viết cung cấp thông tin sai sự thật về nguồn tham khảo, khiến đọc giả không tìm thấy được nguồn chính xác.
- Người viết có dẫn nguồn nhưng lại quên dấu trích dẫn dù đoạn đó được sao chép từng từ một hay gần như thế.
- Trong trường hợp này, người viết chỉ dẫn nguồn ở một vài nội dung tham khảo cơ bản. Mặt dù tiếp tục sử dụng các nội dung khác của cùng một nguồn này để viết bài nhưng người viết không tiếp tục trích dẫn. Bằng cách này, người đọc có thể bị “đánh lừa” bởi các trích dẫn nửa vời của người viết.”

Em cam kết không đạo văn trong bài báo cáo và code demo cho từng thuật toán.

I. Thuật toán ID3 & Decision Tree

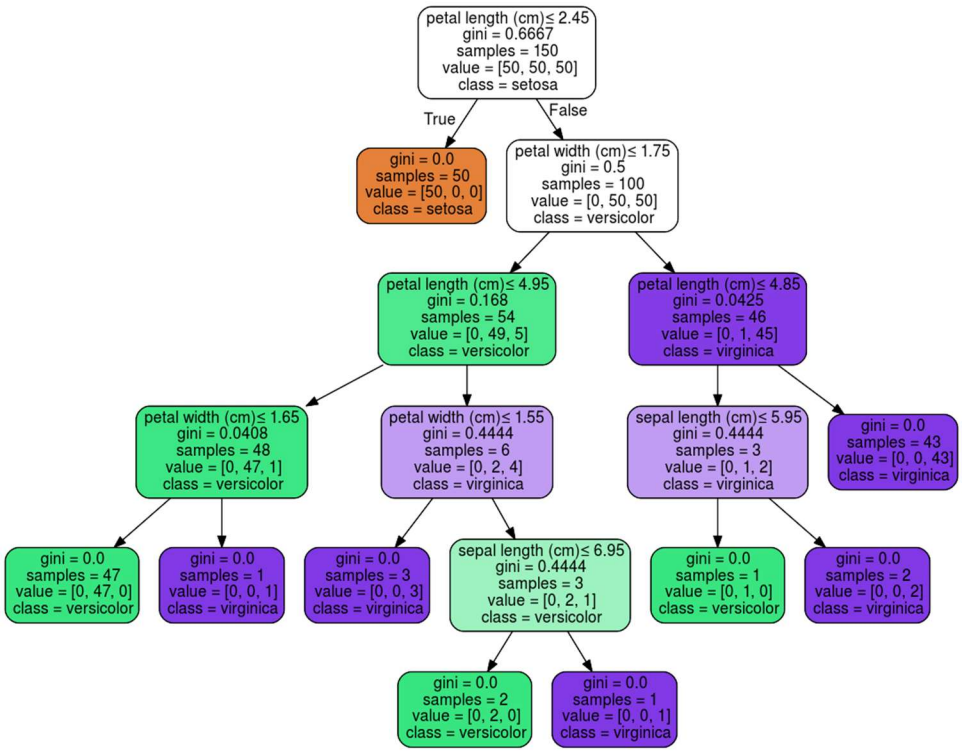
1. Giới thiệu về Decision Tree

Việc quan sát, suy nghĩ và ra các quyết định của con người thường được bắt đầu từ các câu hỏi. Machine Learning cũng có một mô hình ra quyết định dựa trên các câu hỏi. Mô hình này có tên là *cây quyết định* (*Decision Tree*).



Hình 1: *Mô hình Decision Tree*

Decision Node thể hiện một rule để phân nhánh data theo một thuộc tính, mỗi nhánh từ Decision Node là kết quả phân nhánh theo rule đó, Leaf Node là kết quả phân loại cuối cùng. Để hình dung rõ ràng, ta có ví dụ về Decision Tree trên Iris Dataset:



Hình 2: *Decision Tree with Iris Dataset using Sklearn*

Để xây dựng cấu trúc cây ở trên, thuật toán Decision Tree đơn giản sẽ bao gồm các bước sau:

- Chọn lựa thuộc tính của data để chia data sử dụng Attribute Selection Measures (ASM: Chỉ số đánh giá lựa chọn thuộc tính)
- Tạo decision node với feature và điều kiện ở trên.
- Phân nhánh data tạo các child node và lặp lại tiến trình ở trên cho đến khi một trong các điều kiện sau thoả mãn, ta sẽ có leaf node:
  - Tất cả data của node đều thoả mãn điều kiện của decision node.
  - Không có feature với điều kiện nào có thể được chọn nữa.
  - Không còn data nào thoả mãn điều kiện của decision node.

2. Entropy

Entropy là một khái niệm khá thông dụng được dùng trong nhiều lĩnh vực vật lý, toán học v.v… Trong lĩnh vực xử lý thông tin, nhà toán học Claude Shannon đã đưa

ra khái niệm entropy (Shannon entropy). Shannon entropy thể hiện mức độ hỗn loạn hay độ nhiễu của data. Với một hệ thống với N trạng thái có thể xảy ra thì công thức của Shannon entropy sẽ như ở dưới:

$$E(S) = -\sum_{i=1}^{N} p_i \log_2 p_i$$

Trong đó p_i là xác suất trạng thái thứ i xuất hiện ở trong hệ thống. Giá trị entropy càng cao thì độ hỗn loạn của hệ thống càng cao, còn nếu càng thấp thì hệ thống càng có trật tự. Và một hệ thống có trật tự cũng tương đương với việc data được phân nhánh một cách chuẩn xác.

3. Information Gain

Giá trị entropy thể hiện mức độ hỗn loạn của hệ thống, do đó khi entropy giảm hệ thống có trật tự hơn hay có thể nói có nhiều thông tin hơn. Do vậy, độ giảm của entropy được gọi là Information Gain và có công thức như sau:

$$Gain(Q) = S_0 - \sum_{i=1}^{q} \frac{N_i}{N} S_i$$

Trong đó:

- $Gain$ là giá trị Information Gain
- Q là điều kiện để chia data
- q là số nhóm sau khi chia
- N_i là số lượng data của mỗi nhóm

Nếu ta tiếp tục chia nhánh nhóm, cho tới khi mỗi nhóm chỉ có duy nhất một giá trị thì lúc đó entropy của các nhóm này sẽ đều là 0 (Do $\log_2 1 = 0$. Vậy là ta đã có được các Leaf Node và một Decision Tree dùng để dự đoán nhãn theo thứ tự.

4. Gini Impurity (Gini Index)

Đơn giản hơn so với Entropy và Information Gain, Gini Impurity là chỉ số thể hiện mức độ phân loại sai khi ta chọn ngẫu nhiên một phần tử từ tập data. Gini Impurity có công thức như sau:

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

Trong đó:

- $Gini$ là giá trị Gini Impurity
- i là số các lớp có trong tập data
- p_i là xác suất mà một phần tử ngẫu nhiên thuộc lớp i

5. Xây dựng Decision Tree dựa trên ID3 Algorithm

Decision tree được xây dựng dựa trên ID3 như sau:

- Thuật toán bắt đầu với nút gốc là S (thường là tập label).
- Trên mỗi lần lặp của thuật toán, nó lặp đi lặp lại thuộc tính không sử dụng của tập S và tính Entropy(H) và Information Gain(IG) của thuộc tính.
- Nó sẽ chọn thuộc tính có giá trị Entropy nhỏ nhất hoặc giá trị Information Gain lớn nhất.
- Tập S sau đó được phân chia theo thuộc tính được chọn để tạo ra một tập hợp con của dữ liệu.
- Thuật toán tiếp tục lặp lại trên mỗi tập hợp con, chỉ xem xét các thuộc tính chưa từng được chọn trước đó.

  Xét ví dụ ở phần sau, ta tính Entropy của Root-node:

  H(S) = 2.3906

  Của các Child-node:

  H(milk / S) = 1,4162

H(hair / S) = 1,5734

H(feathers / S) = 1,6602

…

Ta so sánh tất cả các Entropy vừa tính được thì:

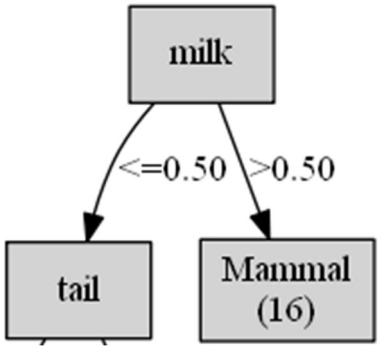
H(*milk/S****)*** < ***H(****tail/S****)*** < ***H(****breathes/S****)*** < ... < ***H(****amphibian/S****)***

Ta xét thuộc tính milk (vì Entropy nhỏ nhất), ta được:

H(S_0) = 2,384

H(S_1) = 0

Ta thấy *H(S_1) = 0* tức là dữ liệu tinh khiết. So với dữ liệu, ta thấy chỉ có nhóm Mammal có thuộc tính milk nên cây quyết định có hình dạng như sau:



Hình 3: *Milk Root-node và Child-node*

Ở Child-node thứ 2 ta đã xác định được label, còn ở Child-node thứ nhất ta tiếp tục xét thuộc tính có giá trị Entropy lớn thứ 2 là thuộc tính *tail.*

Cứ lặp đi lặp lại việc tính này cho đến khi hết các thuộc tính.

II. ID3 Estimator trên Python

1. Dataset

Bài này sẽ sử dụng Zoo Dataset [1] để training, dataset này gồm có 18 attribute & 7 class, thông tin về dataset như sau:

Class Information:

1 -- (41) aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sealion, squirrel, vampire, vole, wallaby, wolf

2 -- (20) chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren

3 -- (5) pitviper, seasnake, slowworm, tortoise, tuatara

4 -- (13) bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna

5 -- (4) frog, frog, newt, toad

6 -- (8) flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp

7 -- (10) clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

Attribute Information:

1 -- animal name: *Unique for each instance*

2 -- hair: *Boolean*

3 -- feathers: *Boolean*

4 -- eggs: *Boolean*

5 -- milk: *Boolean*

6 -- airborne: *Boolean*

7 -- aquatic: *Boolean*

8 -- predator: *Boolean*

9 -- toothed: *Boolean*

10 -- backbone: *Boolean*

11 -- breathes: *Boolean*

12 -- venomous: *Boolean*

13 -- fins: *Boolean*

14 -- legs: *Numeric (set of values: {0,2,4,5,6,8})*

15 -- tail: *Boolean*

16 -- domestic: *Boolean*

17 -- catsize: *Boolean*

18 -- type: *Numeric (integer values in range [1,7])*

Nhận xét: Tập data này đa số là dữ liệu số, khá thuận lợi cho việc training, mục tiêu của bài này là phân loại các *nhóm động vật* dựa trên *các thuộc tính của chúng* như: số chân, đẻ trứng hay con, vv… nên lúc train data, ta có thể bỏ cột giá trị tên động vật.

2. ID3 Estimator

Bài này sử dụng thư viện ID3 Estimator [2] của Daniel Pettersson & Otto Nordander, thông tin & các thuộc tính của Module này như sau:

`Id3Estimator()`

Parameters:

max_depth: int, optional - max depth of features.

min_samples_split: int, optional (default=2) - min samples to split on.

prune: bool, optional (default=False) - set to True to prune the tree.

gain_ratio: bool, optional (default=False) - use gain ratio on split calculations.

is_repeating: bool, optional (default=False) - use repeating features.

Methods:

`fit(X, y, check_input=True)`

Parameters:

`X:` Train data ở dạng mảng hoặc ma trận `[n_samples, n_features]`

`y:` Nhãn của dữ liệu ở dạng mảng (nhãn ở dạng Classification, số thực ở dạng Regression)

`check_input:` Kiểm tra giá trị đầu vào cho các thuộc tính số `(default=True)`

Returns:

`self:` Object - Returns self.

`predict(X)` – Dự đoán giá trị đầu ra y dựa trên giá trị đầu vào X.

Parameters:

`X:` Data ở dạng mảng hoặc ma trận `[n_samples, n_features_idx]`

Return:

`y: array of shape = [n_samples]`

3. Code thuật toán

```
from sklearn import metrics
from sklearn.model_selection import train_test_split
from id3 import Id3Estimator, export_graphviz
import pandas as pd
import numpy as np

data = pd.read_csv('zoo_dataset/zoo.csv')
label = pd.read_csv('zoo_dataset/class.csv')
dataset = data.merge(label, how='left', left_on='class_type',
                     right_on='Class_Number')
feature_names = ['hair', 'feathers', 'eggs', 'milk', 'airborne', 'aquatic', 'p
redator', 'toothed',
                 'backbone', 'breathes', 'venomous', 'fins', 'legs', 'tail', '
domestic']

X = dataset[feature_names]
y = dataset['Class_Type']

# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, train_size=.5, test_size=.5)

# Classifier
clf = Id3Estimator()
clf = clf.fit(X_train, y_train, check_input=True)

# Predict
pred = clf.predict(X_test)

# Accuracy Score
print('Accuracy Score: ', '{:.2%}'.format(metrics.accuracy_score(y_test, pred)
), '\n')

# Confusion matrix
print('Confusion matrix: \n',metrics.confusion_matrix(y_test, pred), '\n')

# Report
print('Classification report: \n',metrics.classification_report(y_test, pred))

# Graph Visualization
export_graphviz(clf.tree_, "out.dot", feature_names)
```

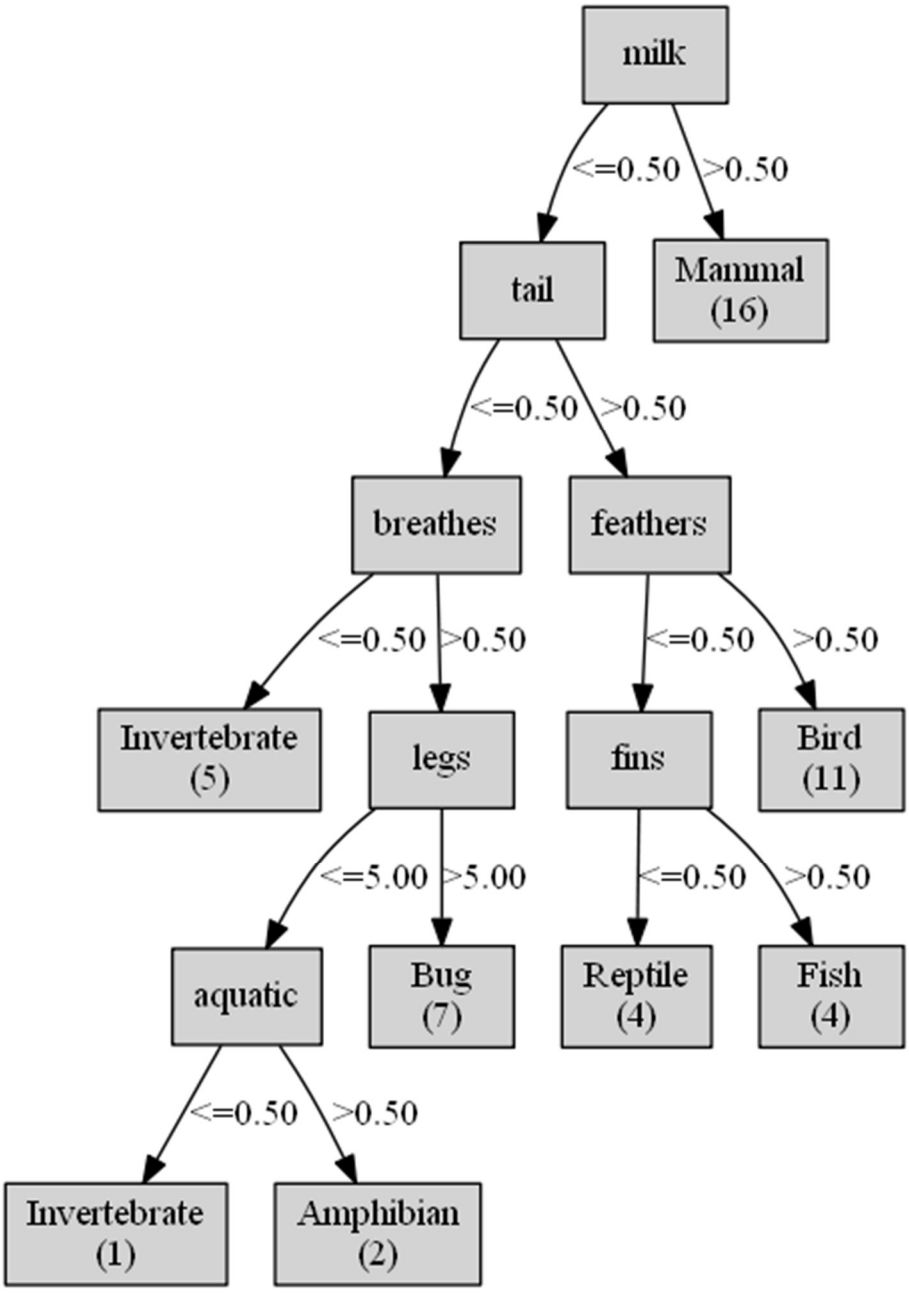
4. Kết quả

```
Classification report:
               precision    recall  f1-score   support

   Amphibian       0.50      1.00      0.67         1
        Bird       1.00      1.00      1.00        10
         Bug       0.80      1.00      0.89         4
        Fish       1.00      1.00      1.00         5
Invertebrate       1.00      1.00      1.00         4
      Mammal       1.00      1.00      1.00        24
     Reptile       1.00      0.33      0.50         3

    accuracy                           0.96        51
   macro avg       0.90      0.90      0.87        51
weighted avg       0.97      0.96      0.96        51
```

Hình 4: *Classification Report*



Hình 5: *Xuất ra cây của data*

III. Tài liệu tham khảo

[1] Zoo Data Set - Machine Learning Repository. Retrieved from https://archive.ics.uci.edu/ml/datasets/Zoo

[2] ID3 Estimator - Daniel Pettersson & Otto Nordander. Retrieved from https://svaante.github.io/decision-tree-id3/