What is Decision Tree

07 October 2022 10:02

- In General, Our algorithm is mostly real life based.
- It tends to do any of the two process such as,
  - It either minimize something
  - Or Maximize something

There are some limitations in the Linear regression and Logistic regression such as,

1. Relationship between independent and dependent variables have to be linear
2. Assumption of Parameters (Decision tree is non-parametric)
3. Treating Multicollinearity (That leads to loss of many information's)

Definition:

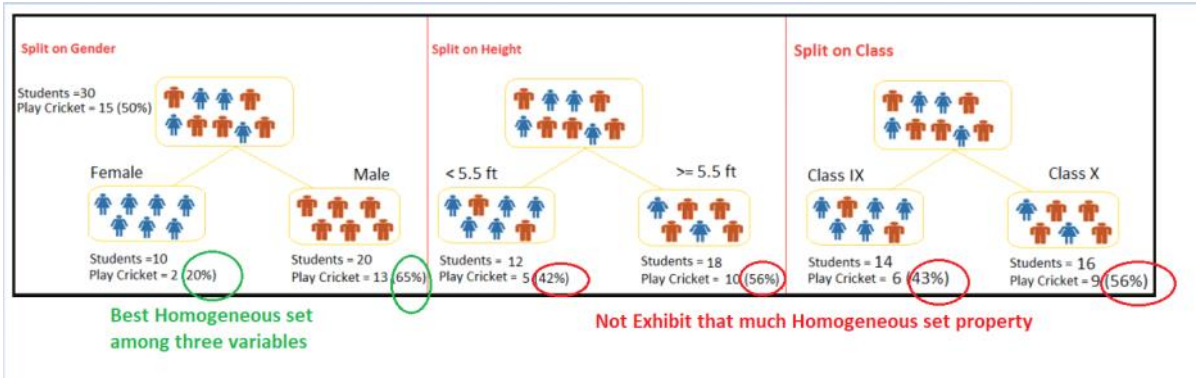
- Decision tree is a type of **supervised** learning algorithm
- It can be used to solve both **Regression** and **Classification** Problem
- Classification problems being put into practical application in most cases.
- It works for both **categorical** and **continuous** input and output variables.

Example:

- Let's say we have a sample of 30 students with three variables
  - Gender (Boy/Girl),
  - Class(IX/X) and
  - Height (5 to 6 ft).
- 15 out of these 30 play cricket in leisure time.
- Now, I want to create a model to predict who will play cricket during leisure period?

Note : In this problem, we need to segregate students who play cricket in their leisure time based on highly significant input variable among all three.

- This is where decision tree helps, it will segregate the students based on all values of three variable and identify the variable, which creates the best homogeneous sets of students (which are heterogeneous to each other).
- Observe the snapshot below to get insights about the homogeneous set from the three variables.



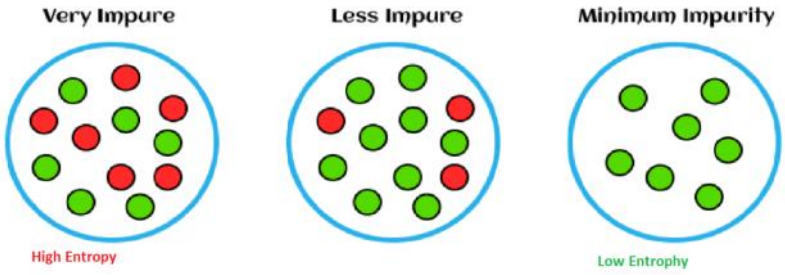
- In the snapshot above, you can see that variable Gender is able to identify best homogeneous sets compared to the other two variables.
- As mentioned above, decision tree identifies the most significant variable and its value that gives best homogeneous sets of population.

Now the question is, how does it identify the variable and the split?

- Before that we discuss about important concept "entropy"
- Then, define the terminologies in the decision tree
- And move on how the split is carried out.

Concept - **Entropy:**

- Entropy is defined as the randomness or measuring the disorder of the information being processed in Machine Learning.
- (or) we can say that entropy is the **machine learning metric** that **measures the unpredictability or impurity** in the system.
- It determines how a decision tree chooses to split data.



- Lower the entropy - It is easier to draw conclusion from a piece of information.
- Higher the entropy - It is difficult to draw any conclusion from that piece of information.

Example:
- Consider the flip of a coin
- Difficult to conclude what exactly the outcome
- There is 50% probability for each.
- Hence entropy is said to be high
- This is the essence in machine learning

Mathematical Formula to calculate the entropy:

$$E = -\sum_{i=1}^{N} P_i \log_2 P_i$$

- N - Total number of classes
- Pi = Probability of randomly selecting an example in class I
- Entropy always lies between 0 and 1 (It can be greater than 1 depends on the number of classes of the set).

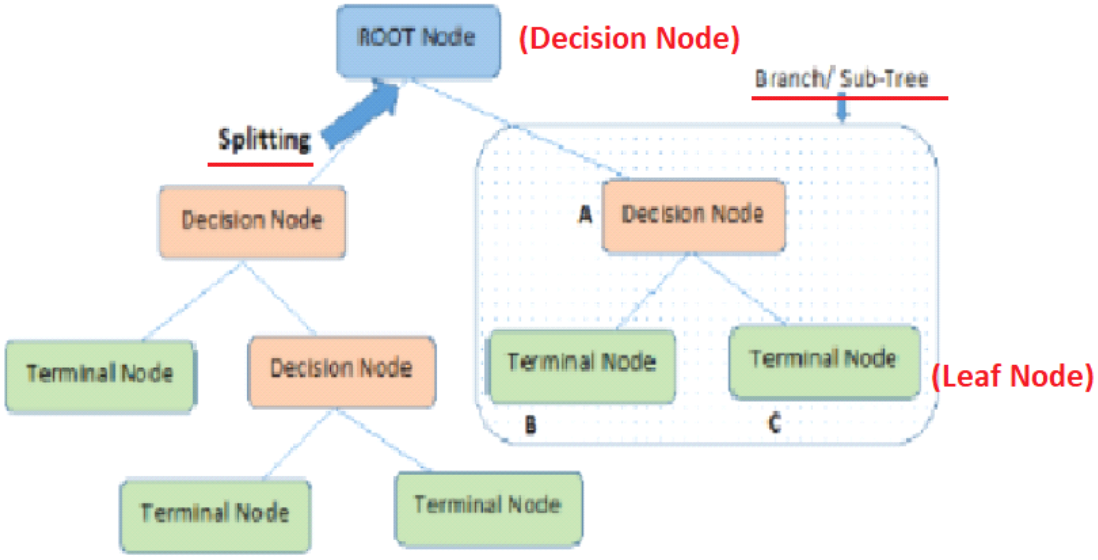
Interpreting the Value of entropy with example:
- Entropy = "0" means no impurity and not useful for learning
- Entropy = "1" means Dataset is good for learning.

Let consider the 2 scenario to illustrate how entropy values derived,

With 3 variables	With one variable
• Suppose we have a dataset having three colors of fruits as 2 red, 2 green, and 4 yellow. • Pr = (2/8), Pp = (2/8), Py = (4/8) be the probability to choosing red, green, yellow fruits. • Equation of entropy becomes **E=−(prlog2pr+pplog2pp+pylog2py)** $E = -(\frac{1}{4}\log_2(\frac{1}{4}) + \frac{1}{4}\log_2(\frac{1}{4}) + \frac{1}{2}\log_2(\frac{1}{2}))$ • **E = -(1/4*(-2)+1/4(-2)+1/2*(-1)** **E = -(-1/2-1/2-1/2)** **E = -(-1.5)= 1.5** • So, the **entropy will be 1.5** • As we discussed, entropy value can be greater than 1 when increase in the number of variables.	• Let consider the cases where all the observations belongs to the same class • Then equation becomes, • **E=−(1log21)** = 0

Important Terminology

10 October 2022 09:25



Root Node:

- It represents entire population or sample and this further gets divided into two sets.

Splitting:

- It is a process of dividing a Decision node/ Root node into two sub-nodes (Terminal nodes/ another sub-nodes).

Decision Node:

- When a sub-node splits into further sub-nodes, then it is called decision node.

Leaf/ Terminal Node:

- Output of decision nodes.
- Nodes do not split is called leaf or terminal node.

Pruning:

- When we remove sub-nodes of a decision node, this process is called pruning.
- You can say opposite process of splitting.

Branch / Sub-Tree:

- A sub-section of entire tree is called branch or sub-tree.

Parent Node:

- A node, which is divided into sub-nodes is called parent node of sub-nodes

Child Node:

- Sub-nodes are the child of parent node.
- i.e., Except for the root node, all other nodes are child nodes

How to Split

10 October 2022 09:25

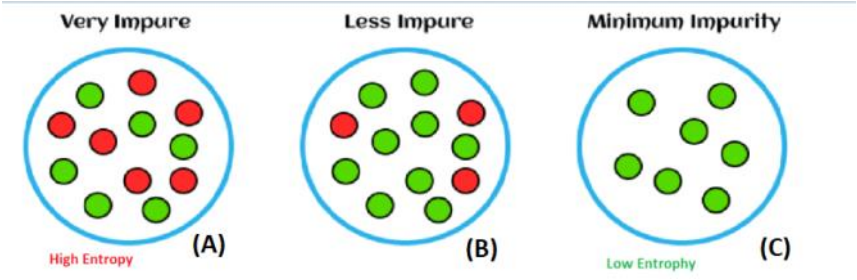
- The decision of making strategic splits heavily affects a tree's accuracy.
- The decision criteria is different for classification and regression trees.

- Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes.
- The creation of sub-nodes increases the homogeneity of resultant sub-nodes.
- In other words, we can say that purity of the node increases with respect to the target variable.
- At 1st Decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

Methods to determine best split:

1. Information Gain
2. Gini
3. Chi Square
4. Reduction in Variance
    - Information Gain and Gini not used for regression problems
    - Hence Reduction in variance used for regression problems

Information Gain



- Look at the image below and think which node can be described easily.
- I am sure, your answer is C because it requires less information as all values are similar.
- On the other hand, B requires more information to describe it and A requires the maximum information.
- In other words, we can say that C is a Pure node, B is less Impure and A is more impure.

Conclusion from the above statements:

- **Less impure node requires less information** to describe it. And, **more impure node** requires **more information.**

- **Information theory tries to measure and define this degree of disorganization in a system known as Entropy.**
  - If the sample is completely **homogeneous**, then the **entropy is zero**
  - if the sample is an equally divided (**50% – 50%**), it has **entropy of one**.

Formula for calculating Entropy:

$$\text{Entropy} = -p \log_2 p - q \log_2 q$$

Here,
p and q is probability of success and failure respectively in that node.

- Entropy is also used with categorical target variable.
- **It chooses the split which has lowest entropy compared to parent node and other splits.**
- The lesser the entropy, the better it is.
- The **decrease is entropy after split** is called **Information Gain.**

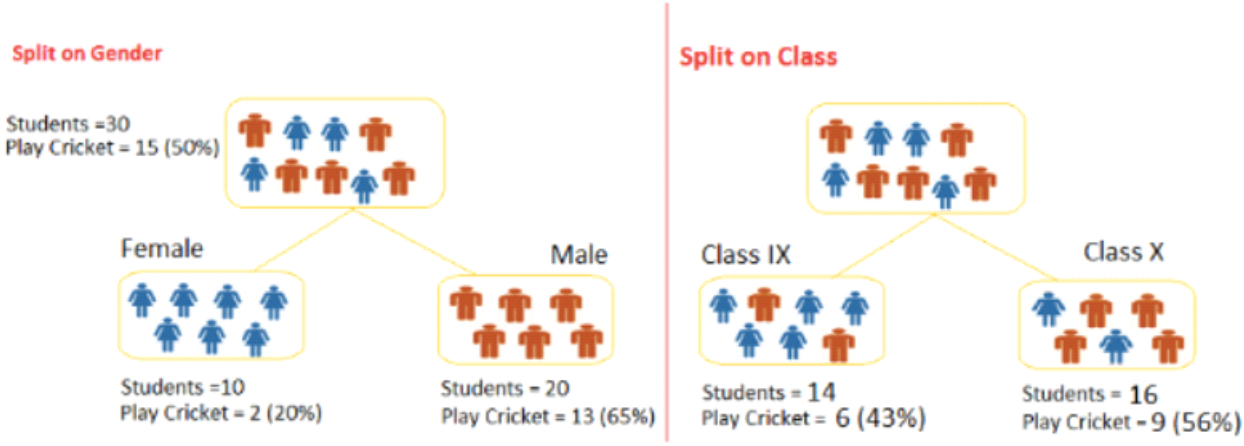
Steps to calculate "information Gain" for a split:

1. Calculate entropy of parent node.
2. Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.
3. Calculate the difference in entropy before and after split.

Example to calculate the "Information Gain" :

Referring to example used above,

- where we want to segregate the students based on target variable (playing cricket or not).
- In the snapshot below, we split the population using two input variables Gender and Class studying.
- Now, we want to identify which split is producing more homogeneous sub-nodes using Information Gain.



Step 01: "Entropy of Parent node"

- Probability of students playing cricket = 15/30
- Probability of students not playing cricket = 15/30
- Entropy for parent node = - (15/30) log(15/30) – (15/30) log(15/30) = 1
- "1" here shows that our node is "impure node"

Step 02: Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.

Split on Gender:

- **Probability** of Female who plays the cricket = 2/10
- Probability of Female who not plays the cricket = 8/10
- Probability of Male who plays the cricket = 13/20
- Probability of Male who not plays the cricket = 7/20

- **Entropy** for Female node = -(2/10) log(2/10) – (8/10) log(8/10) = **0.72**
- Entropy for Male node = -(13/20) log(13/20) – (7/20) log(7/20) = **0.93**
- Entropy for split Gender = **Weighted entropy of sub-nodes** = (10/30)0.72 + (20/30)0.93 = **0.86**

Split on Class:

- **Probability** of Class IX playing Cricket =
- Probability of Class IX Not playing Cricket =
- Probability of Class X playing cricket =
- Probability of Class X Not playing cricket =

- **Entropy** for Class IX node, -(6/14) log(6/14) – (8/14) log(8/14) = 0.99
- Entropy for Class X node, -(9/16) log(9/16) – (7/16) log(7/16) = 0.99
- Entropy for split Class = **weighted entropy of sub-nodes =** (14/30)0.99 + (16/30)0.99 = 0.99

Step-03 "Calculate the difference in entropy before and after split."

- **Information Gain for split Gender,**
  = Entropy before split - Entropy after split
  = 1 - 0.86 = 0.14

- **Information Gain for split Class** = Entropy before split - Entropy after split
  = 1 - 0.99 = 0.01

Decision:

- Above, we can see that Information Gain for split on Gender is the highest among all,
- so the tree will split on Gender.

Gini

- Gini says, if we select two items from a population at random then,
- **if the population is pure**, they must be of **same class and probability** for this is **1.**

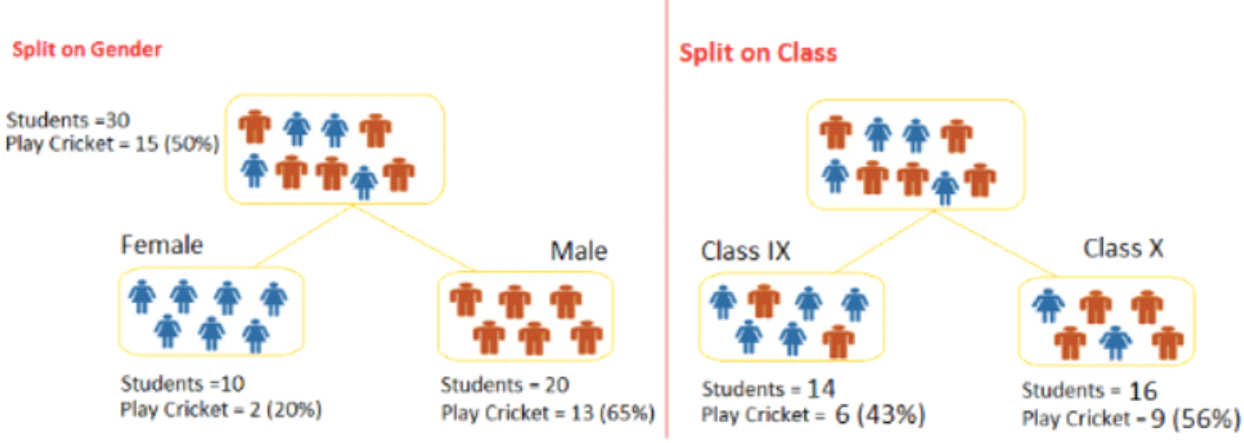
Properties for Gini:

1. It works with **categorical target variable** "Success" or "Failure".
2. It performs **only binary splits**.
3. **Higher** the value of Gini higher the **homogeneity**.
4. CART (Classification and Regression Tree) uses Gini method to create binary splits.

Steps to calculate Gini for a split:

1. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p^2+q^2).
2. Calculate Gini for split using weighted Gini score of each node of that split

Example to identify the best split for student:



Step-01: "Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p^2+q^2)."

Split on Gender:

- **Gini** for sub-node **Female** = (0.2)(0.2)+(0.8)(0.8)=0.68
- **Gini** for sub-node **Male** = (0.65)(0.65)+(0.35)(0.35)=0.55

Split on Class:

- **Gini** for sub-node Class IX = (0.43)(0.43)+(0.57)(0.57)=0.51
- **Gini** for sub-node Class X = (0.56)(0.56)+(0.44)(0.44)=0.51
- Calculate **weighted Gini for Split Class** = (14/30)0.51+(16/30)0.51 = **0.51**

Step-02: "Calculate Gini for split using weighted Gini score of each node of that split":

- Calculate weighted Gini for Split Gender = (10/30)0.68+(20/30)0.55 = 0.59
- Calculate weighted Gini for Split Class = (14/30)0.51+(16/30)0.51 = 0.51

Decision:

Above, you can see that Gini score for split on Gender is higher than split on Class, hence, the node split will take place on Gender.

Gini Impurity:

- which is determined by subtracting the gini value from 1
- So mathematically we can say,

Gini Impurity = 1 - Gini

Implementation

10 October 2022 09:25

- The file **daily_weather.csv** is a comma-separated file that contains weather data.
- The weather station is equipped with sensors that capture weather-related measurements such as **air temperature, air pressure, and relative humidity**.
- Data was collected for a period of three years, from September 2011 to September 2014, to ensure that sufficient data for different seasons and weather conditions is captured.

Problem: Use morning sensor signals as features to predict whether the humidity will be high at 3pm.

1. Importing Libraries and Dataset

```
[1]  import pandas as pd
     from sklearn.metrics import accuracy_score, auc
     from sklearn.model_selection import train_test_split
     from sklearn.tree import DecisionTreeClassifier
```

```
[2]  from google.colab import drive
     drive.mount('/content/drive')

     Mounted at /content/drive
```

```
[4]  data = pd.read_csv('/content/drive/MyDrive/Almabetter/Module 04 ML/Source files for Decision Tree/daily_weather.csv')
```

2. Data Understanding

```
data.columns
```

```
Index(['air_pressure_9am', 'air_temp_9am', 'avg_wind_direction_9am',
       'avg_wind_speed_9am', 'max_wind_direction_9am', 'max_wind_speed_9am',
       'rain_accumulation_9am', 'rain_duration_9am', 'relative_humidity_9am',
       'high_humidity_3pm'],
      dtype='object')
```

- Just viewing the values count from the high humidity column which we going to take as dependent variable

```
data['high_humidity_3pm'].value_counts()
```

```
0    535
1    529
Name: high_humidity_3pm, dtype: int64
```

3. Data Cleaning

```
[6] data.shape #With Null values
```

```
(1095, 10)
```

Removing the rows which contains the null values

```
[ ] data.dropna(inplace=True)
```

```
[ ] data.shape # Without Null values
```

```
(1064, 10)
```

4. Creating Dependent and independent variables

```
[ ] dependent_variable = 'high_humidity_3pm'
```

```
[ ] independent_variables = ['air_pressure_9am', 'air_temp_9am', 'avg_wind_direction_9am',
       'avg_wind_speed_9am', 'max_wind_direction_9am', 'max_wind_speed_9am',
       'rain_accumulation_9am', 'rain_duration_9am', 'relative_humidity_9am']
```

```
[ ] X = data[independent_variables]
```

```
[ ] y = data[dependent_variable]
```

5. Fitting to train the dataset

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=324)
```

```
humidity_classifier = DecisionTreeClassifier(criterion='entropy', max_leaf_nodes=10, random_state=0)
humidity_classifier.fit(X_train, y_train)
```

criterion :

- This parameter determines how the impurity of a split will be measured.
- The **default value is "Gini"** but you can also use "entropy" as a metric for impurity.
- splitter: This is how the decision tree searches the features for a split.

max_leaf_nodes – Maximum number of leaf nodes a decision tree can have.
max_features – Maximum number of features that are taken into the account for splitting each node.

6. Predict on Test dataset

```
y_predicted = humidity_classifier.predict(X_test)
```

7. Evaluating the Metrics

- First we just view the y-predicted and y-test

```
[ ] y_predicted[:10]

    array([0, 0, 1, 1, 1, 1, 1, 0, 1, 1])
```

```
y_test[:10]

456     0
845     0
693     1
259     1
723     1
224     1
300     1
442     0
585     1
1057    1
Name: high_humidity_3pm, dtype: int64
```

- Now, check on the accuracy using the code,

```
[ ] accuracy_score(y_test, y_predicted) * 100

    90.9090909090909
```

8. Visualizing the Decision Tree

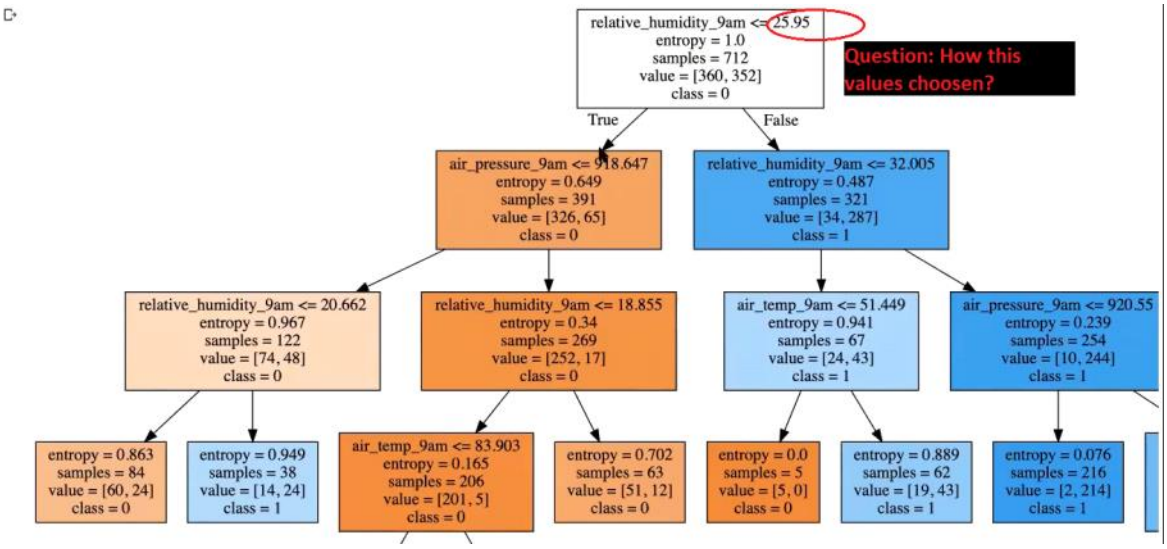
```
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import tree
from IPython.display import SVG
from graphviz import Source
from IPython.display import display
```

- One needs to install **pydotplus** and **graphviz**.
- These can be installed with the package manager.
- Here, we used only graphviz

- Graphviz is a tool that is used for drawing graphics; it takes help from dot files.
- Pydotplus is a module to graphviz's dot language.

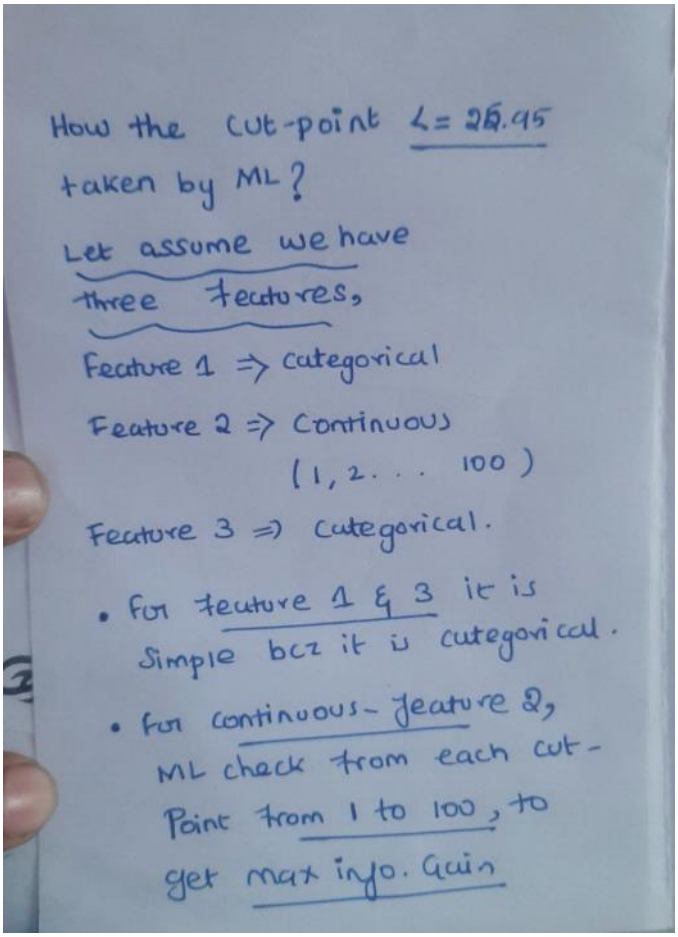
- What is **IPython** display in Python?
  - IPython (Interactive Python) is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history.
- **SVG**
  - short for **scalable vector graphic** , is a standard graphics type used for rendering two-dimensional images.

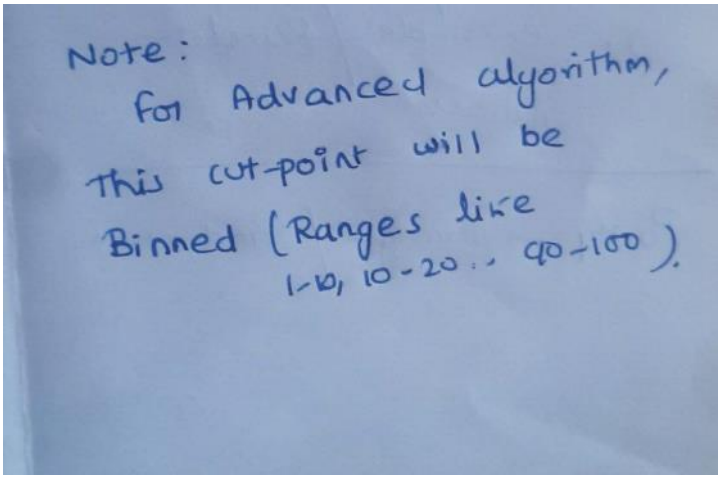
```
graph = Source(tree.export_graphviz(humidity_classifier, out_file=None
          , feature_names=X_train.columns, class_names=['0', '1']
          , filled = True))
display(SVG(graph.pipe(format='svg')))
```



Explanation:

How the cutting point on the Each Node chosen by the ML?





How the decision tree can be controlled by using parameters?

- We can control the number of observations that each node must have for further split
- We can control the number of leaf in the decision tree
- We can say the maximum information gain that allows to further split
- We can say the maximum depth of the decision tree

Advantages & Disadvantages

10 October 2022 09:26

Easy to Understand:

- Decision tree output is very easy to understand even for people from non-analytical background.
- It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.

Useful in Data exploration:

- Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables.
- With the help of decision trees, we can **create new variables / features** that has better power to predict target variable.
- It can also be used in data exploration stage. For example, we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable. **Less data cleaning required**: It requires less data cleaning compared to some other modeling techniques. It is **not influenced by outliers** and missing values to a fair degree.

Non Parametric Method:

- Decision tree is considered to be a non-parametric method.
- This means that decision trees have **no assumptions** about the space distribution and the classifier structure.

Disadvantages

Over fitting:

- As decision tree is a greedy algorithm, which tree grows it reduces at each node takes 1, that create the problem of overfitting
- Over fitting is one of the most practical difficulty for decision tree models.
- This problem gets solved by setting constraints on model parameters and pruning.

  3 ways to avoid over-fitting:
  - Control using the hyper-parameter by don't let the tree grow much enough
  - Pruning (based on CV, Penalty, Levels)
  - Ensemble and Random Forest.

Note:

- Strategies like ensuring each leaf node as one pure class that will not help

to reduce the overfitting in decision tree.

- Purity = Homogeneous / Randomness