ML

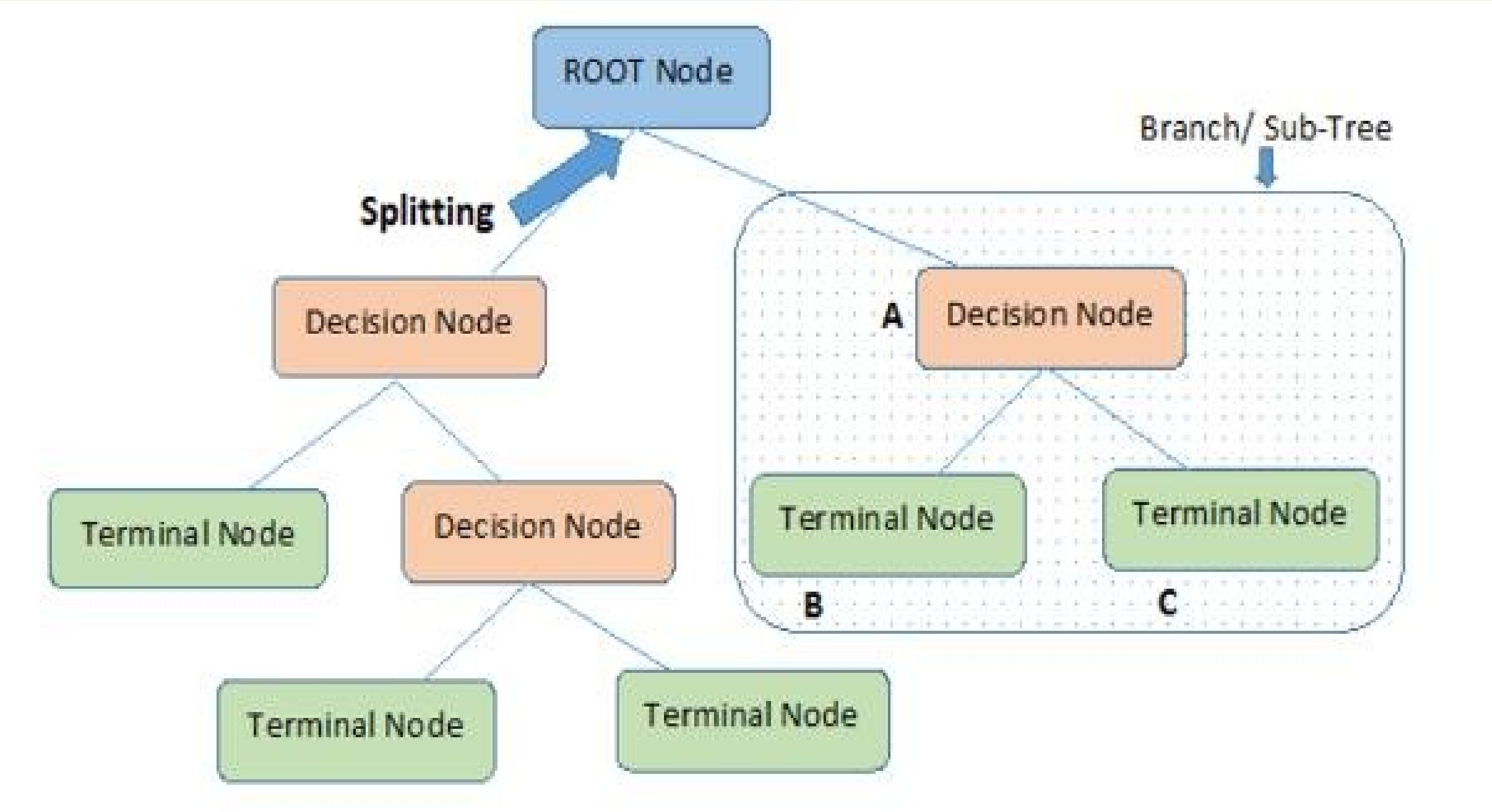
BY
Aradhana Behura
School Of Computer Engineering

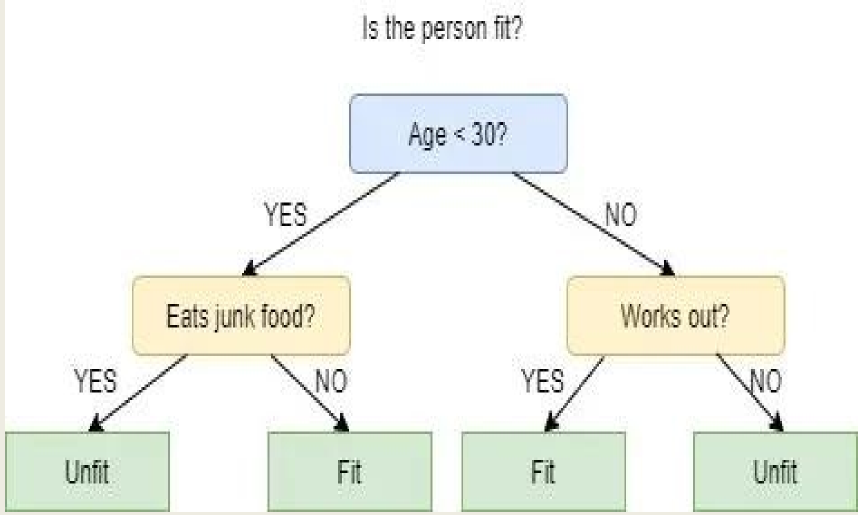
Decision Tree

- Decision Tree is a **Supervised learning technique** that can be used for **both classification and Regression problems**, but mostly it is preferred for solving Classification problems.
- It is a **tree-structured classifier**, where **internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions** and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a **graphical representation for getting all the possible solutions to a problem/decision based on given conditions.**
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

- **Reasons for using the Decision tree:**
- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Example





Decision Tree Terminologies

- **Root Node:** Root node is from where the **decision tree starts.** It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the **final output node,** and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the **process of dividing the decision node/root node into sub-nodes according to the given conditions.**
- **Branch/Sub Tree:** A tree formed by **splitting the tree.**
- **Pruning:** Pruning is the process of **removing the unwanted branches** from the tree.
- **Parent/Child node:** A node that is **divided into sub-nodes is known as a parent node, and the sub-nodes emerging from it are referred to as child nodes.** The **parent node represents a decision or condition, while the child nodes represent the potential outcomes or further decisions based on that condition.**

ID3 Algorithm

- ID3 stands for **Iterative Dichotomiser 3** and is named such because the algorithm **iteratively (repeatedly) dichotomizes(divides) features into two or more groups** at each step.

- **Invented by Ross Quinlan,** ID3 uses a top-down greedy approach to build a decision tree.

- In simple words, the **top-down approach means that we start building the tree from the top and the greedy approach means that at each iteration we select the best feature at the present moment to create a node.**

- **Most generally ID3 is only used for classification** problems with nominal features only.

Metrics in ID3

- **ID3 algorithm selects the best feature at each step while building a Decision tree.**

- So the answer to the question: 'How does ID3 select the best feature?' is that **ID3 uses Information Gain or just Gain to find the best feature.**

- **Information Gain calculates the reduction in the entropy and measures how well a given feature separates or classifies the target classes.**

- The **feature with the highest Information Gain is selected** as the best one.

- In simple words, **Entropy is the measure of disorder and the Entropy of a dataset is the measure of disorder in the target feature** of the dataset.

- In the case of binary classification (where the target column has only two types of classes) entropy is 0 if all values in the target column are homogenous(similar) and will be 1 if the target column has equal number values for both the classes.

Metrics in ID3

- We denote our dataset as S, entropy is calculated as:

$$\textbf{Entropy(S)} = - \sum p_i * \log_2(p_i) \text{ ; } i = 1 \text{ to } n$$

- where,
- n is the total number of classes in the target column (in our case n = 2 i.e YES and NO)
- p_i is the probability of class ‘i’ or the ratio of “number of rows with class i in the target column” to the “total number of rows” in the dataset.

- Information Gain for a feature column A is calculated as:

$$\textbf{IG(S, A)} = \textbf{Entropy(S)} - \sum((|S_v| / |S|) * \textbf{Entropy}(S_v))$$

- where S_v is the set of rows in S for which the feature column A has value v, $|S_v|$ is the number of rows in S_v and likewise |S| is the number of rows in S.

ID3 Steps

I. Calculate the Information Gain of each feature.

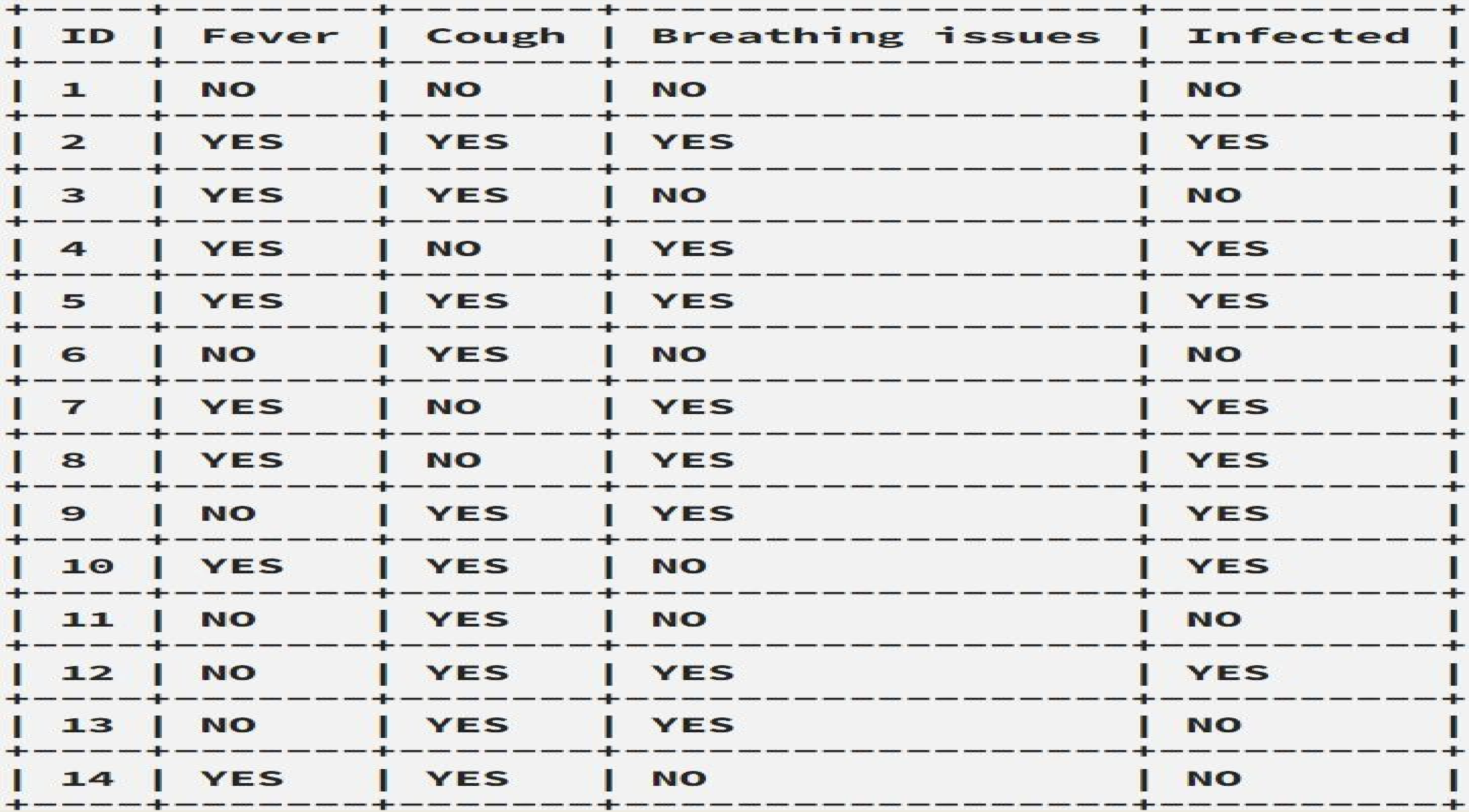
II. Considering that all rows don’t belong to the same class, split the dataset S into subsets using the feature for which the Information Gain is maximum.

III. Make a decision tree node using the feature with the maximum Information gain.

IV. If all or most of the rows belong to the same class, make the current node as a leaf node with the class as its label.

V. Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

Example Dataset



```
+----+-----------+-----------+----------------------+--------------+
| ID | Fever     | Cough     | Breathing issues     | Infected     |
+----+-----------+-----------+----------------------+--------------+
| 1  | NO        | NO        | NO                   | NO           |
+----+-----------+-----------+----------------------+--------------+
| 2  | YES       | YES       | YES                  | YES          |
+----+-----------+-----------+----------------------+--------------+
| 3  | YES       | YES       | NO                   | NO           |
+----+-----------+-----------+----------------------+--------------+
| 4  | YES       | NO        | YES                  | YES          |
+----+-----------+-----------+----------------------+--------------+
| 5  | YES       | YES       | YES                  | YES          |
+----+-----------+-----------+----------------------+--------------+
| 6  | NO        | YES       | NO                   | NO           |
+----+-----------+-----------+----------------------+--------------+
| 7  | YES       | NO        | YES                  | YES          |
+----+-----------+-----------+----------------------+--------------+
| 8  | YES       | NO        | YES                  | YES          |
+----+-----------+-----------+----------------------+--------------+
| 9  | NO        | YES       | YES                  | YES          |
+----+-----------+-----------+----------------------+--------------+
| 10 | YES       | YES       | NO                   | YES          |
+----+-----------+-----------+----------------------+--------------+
| 11 | NO        | YES       | NO                   | NO           |
+----+-----------+-----------+----------------------+--------------+
| 12 | NO        | YES       | YES                  | YES          |
+----+-----------+-----------+----------------------+--------------+
| 13 | NO        | YES       | YES                  | NO           |
+----+-----------+-----------+----------------------+--------------+
| 14 | YES       | YES       | NO                   | NO           |
+----+-----------+-----------+----------------------+--------------+
```

Implementation of ID3 on Dataset

- The first step is to find the best feature i.e. the one that has the maximum Information Gain(IG).

- We’ll calculate the IG for each of the features now, but for that, we first need to calculate the entropy of S.

- From the total of 14 rows in our dataset S, there are 8 rows with the target value YES and 6 rows with the target value NO. The entropy of S is calculated as:

$$\textbf{Entropy(S)} = -\,(8/14) * \log_2(8/14) - (6/14) * \log_2(6/14) = 0.99$$

- **Note: If all the values in our target column are same the entropy will be zero (meaning that it has no or zero randomness).**

- We now calculate the Information Gain for each feature.

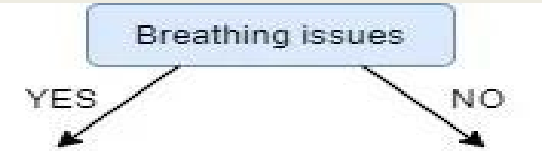
Implementation of ID3 on Dataset

- **IG calculation for Fever:**
- In this(Fever) feature there are 8 rows having value YES and 6 rows having value NO.
- In the 8 rows with YES for Fever, there are 6 rows having target value YES and 2 rows having target value NO.
- In the 6 rows with NO, there are 2 rows having target value YES and 4 rows having target value NO.
- $|S| = 14$
- For $v = \text{YES}$, $|S_v| = 8$
- **Entropy**$(S_v) = -(6/8) * \log_2(6/8) - (2/8) * \log_2(2/8) = 0.81$
- For $v = \text{NO}$, $|S_v| = 6$
- **Entropy**$(S_v) = -(2/6) * \log_2(2/6) - (4/6) * \log_2(4/6) = 0.91$

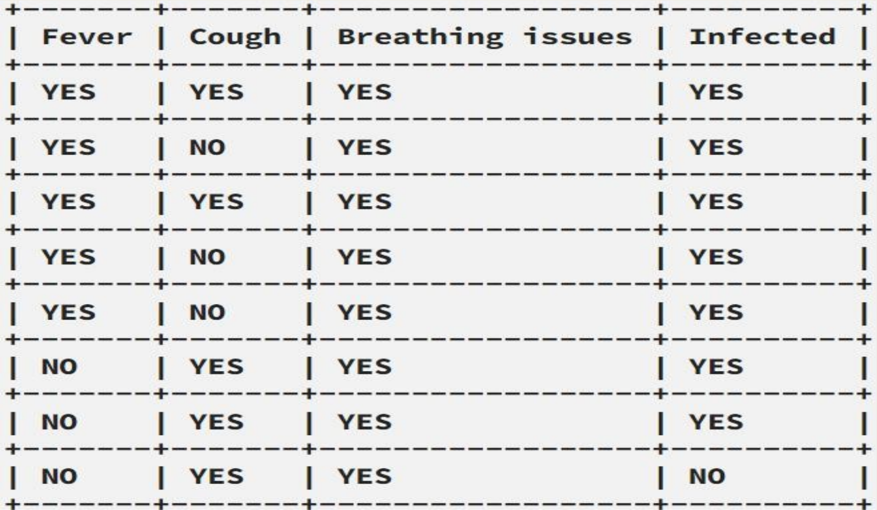
- **# Expanding the summation in the IG formula:**
- **IG**$(S, \text{Fever}) = \text{Entropy}(S) - (|S_{YES}| / |S|) * \text{Entropy}(S_{YES}) - (|S_{NO}| / |S|) * \text{Entropy}(S_{NO})$
- $\therefore$ **IG**$(S, \text{Fever}) = 0.99 - (8/14) * 0.81 - (6/14) * 0.91 = 0.13$

Implementation of ID3 on Dataset

- Next, we calculate the IG for the features "Cough" and "Breathing issues".
- **IG(S, Cough) = 0.04**
- **IG(S, BreathingIssues) = 0.40**
- Since the feature Breathing issues have the highest Information Gain it is used to create the root node.
- Hence, after this initial step our tree looks like this:



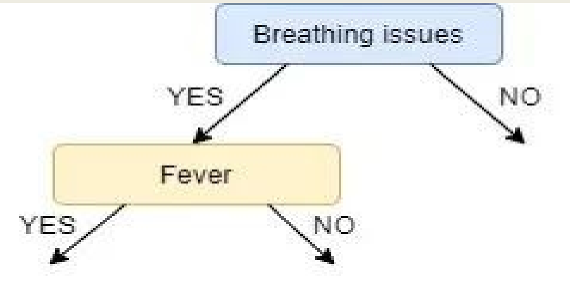
- Next, from the remaining two unused features, namely, Fever and Cough, we decide which one is the best for the left branch of Breathing Issues.
- Since the left branch of Breathing Issues denotes YES, we will work with the subset of the original data i.e the set of rows having YES as the value in the Breathing Issues column. These 8 rows are shown below:



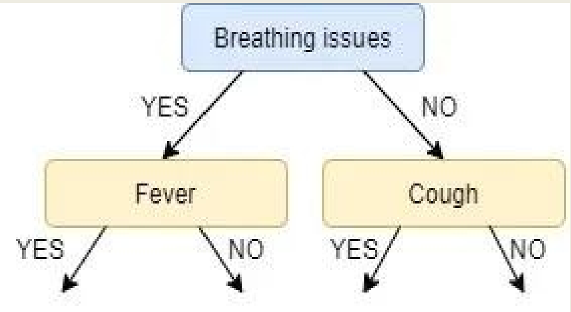
Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES
NO	YES	YES	YES
NO	YES	YES	YES
NO	YES	YES	NO

Implementation of ID3 on Dataset

- Next, we calculate the IG for the features Fever and Cough using the subset S_{BY} (Set Breathing Issues Yes)
- **Note:** For IG calculation the Entropy will be calculated from the subset S_{BY} and not the original dataset S.
- **IG(S_{BY}, Fever) = 0.20**
- **IG(S_{BY}, Cough) = 0.09**
- **IG of Fever is greater than that of Cough, so we select Fever as the left branch** of Breathing Issues.
- Our tree now looks like this:



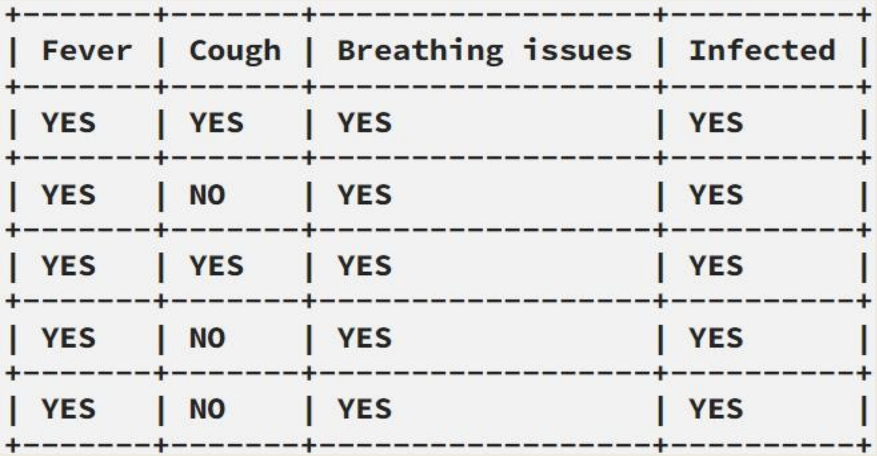
- Next, we find the feature with the maximum IG for the right branch of Breathing Issues. But, since there is **only one unused feature left we have no other choice but to make it the right branch of the root node.**
- So our tree now looks like this:



- There are no more unused features, so we stop here and jump to the final step of creating the leaf nodes.

Implementation of ID3 on Dataset

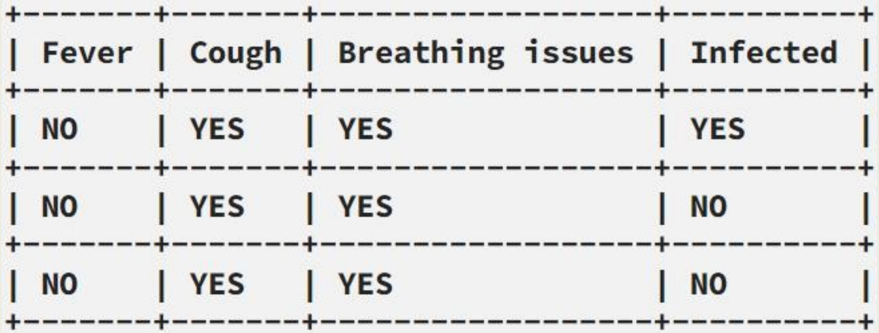
- ❖ For the left leaf node of Fever, we see the subset of rows from the original data set that has Breathing Issues and Fever both values as YES.



```
+-------+-------+------------------+----------+
| Fever | Cough | Breathing issues | Infected |
+-------+-------+------------------+----------+
| YES   | YES   | YES              | YES      |
+-------+-------+------------------+----------+
| YES   | NO    | YES              | YES      |
+-------+-------+------------------+----------+
| YES   | YES   | YES              | YES      |
+-------+-------+------------------+----------+
| YES   | NO    | YES              | YES      |
+-------+-------+------------------+----------+
| YES   | NO    | YES              | YES      |
+-------+-------+------------------+----------+
```

- ❖ Since **all the values in the target column are YES, we label the left leaf node as YES, but to make it more logical we label it Infected.**

- ❖ Similarly, for the right node of Fever we see the subset of rows from the original data set that have Breathing Issues value as YES and Fever as NO.

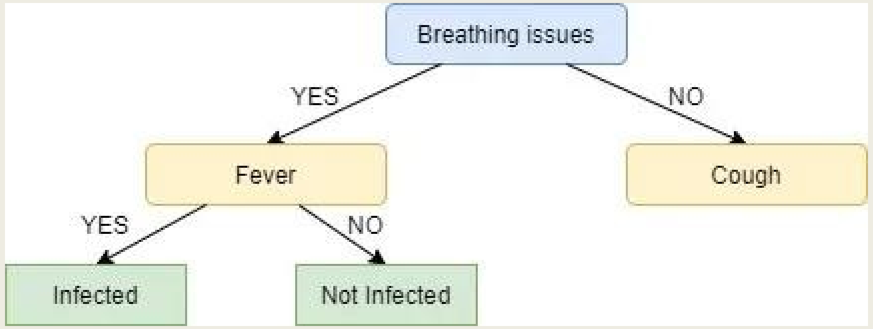


```
+-------+-------+------------------+----------+
| Fever | Cough | Breathing issues | Infected |
+-------+-------+------------------+----------+
| NO    | YES   | YES              | YES      |
+-------+-------+------------------+----------+
| NO    | YES   | YES              | NO       |
+-------+-------+------------------+----------+
| NO    | YES   | YES              | NO       |
+-------+-------+------------------+----------+
```

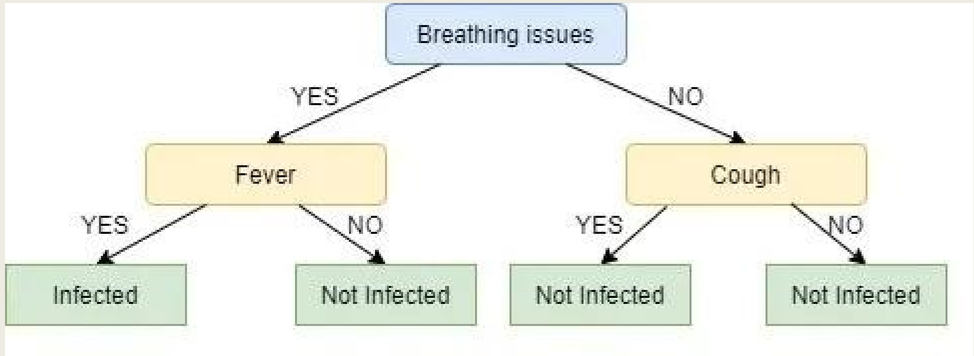
- ❖ Here not all but **most of the values are NO, hence NO or Not Infected** becomes our right leaf node.

Implementation of ID3 on Dataset

- Our tree, now, looks like this:



- We repeat the same process for the node Cough, however here **both left and right leaves turn out to be the same i.e. NO or Not Infected** as shown below:



- The **right node of Breathing issues is as good as just a leaf node with class ‘Not infected’**. This is one of the Drawbacks of ID3, it doesn’t do pruning.

- **Pruning is a mechanism that reduces the size and complexity** of a Decision tree by **removing unnecessary nodes.**

- Another drawback of ID3 is overfitting or high variance i.e. it learns the dataset it used so well that it fails to generalize on new data which **can be resolved using the Random Forest algorithm.**

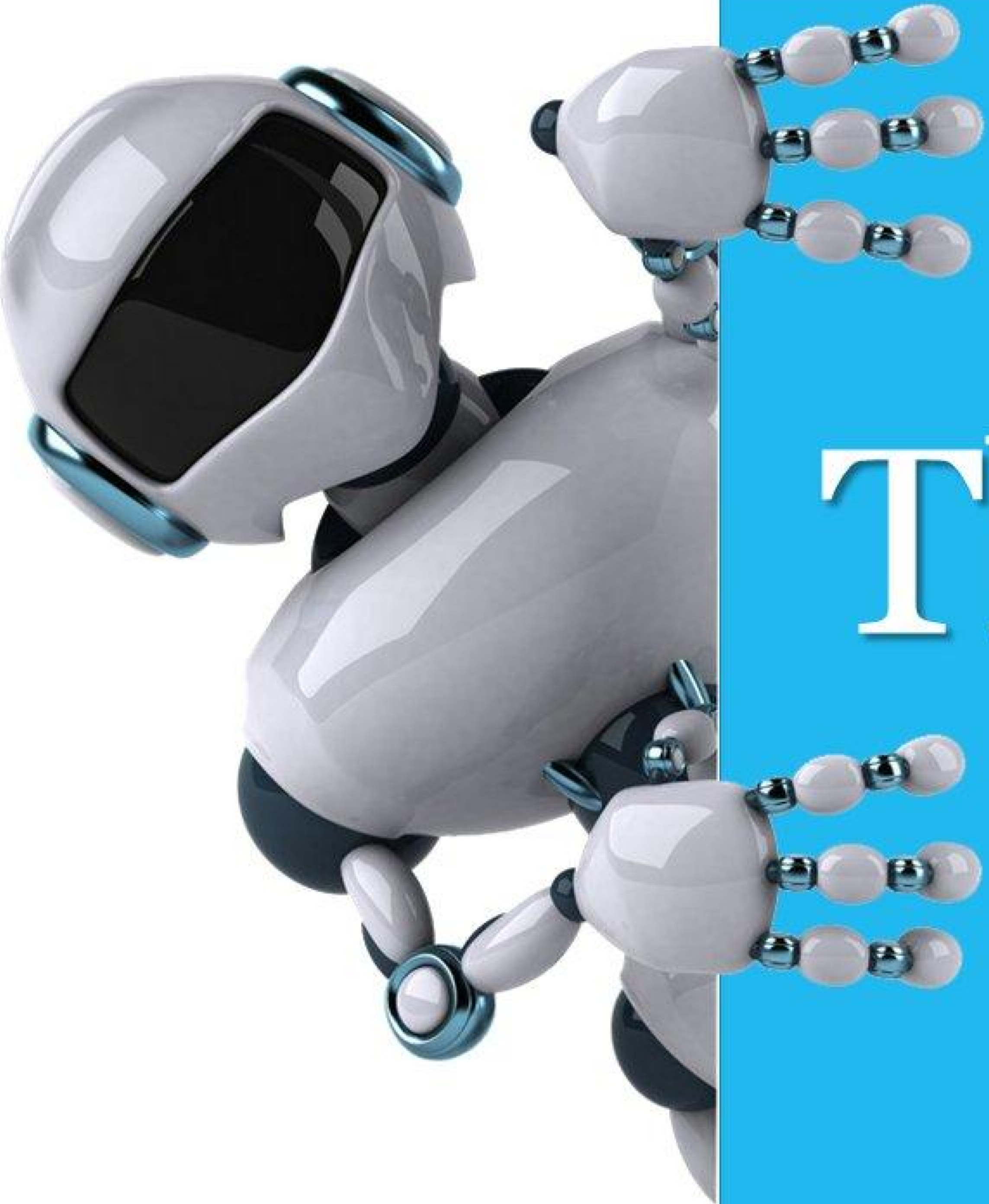
Advantages and Disadvantages of the Decision Tree

❖ Advantages of the Decision Tree

1. It is **simple to understand** as it follows the same process which a human follow while making any decision in real-life.
2. It can be **very useful for solving decision-related problems.**
3. It helps to think about **all the possible outcomes for a problem.**
4. There is **less requirement of data cleaning compared to other algorithms.**

❖ Disadvantages of the Decision Tree

5. The decision tree **contains lots of layers, which makes it complex.**
6. It may have an **overfitting issue**, which **can be resolved using the Random Forest algorithm.**
7. For more class labels, the **computational complexity of the decision tree may increase.**
8. It may **contain some unnecessary nodes** which can be **solved by prunning.**



Thank you