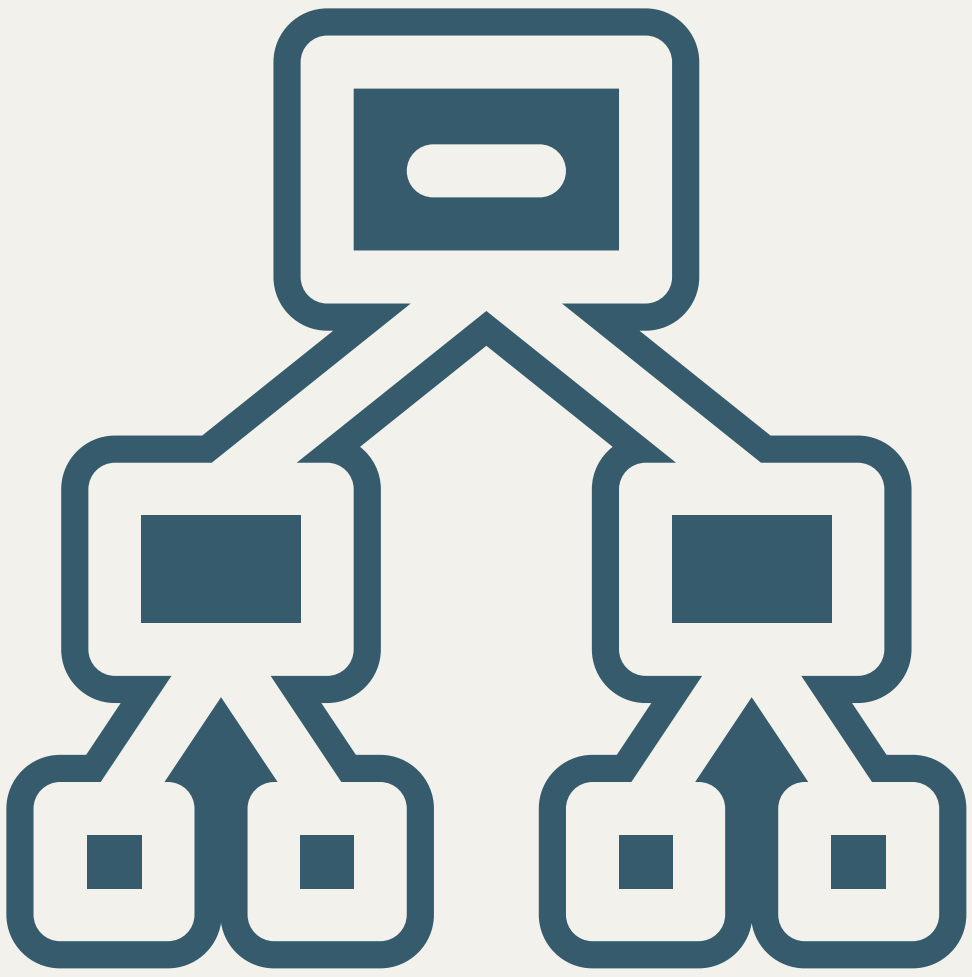
linkedin.com/in/vikrantkumar95

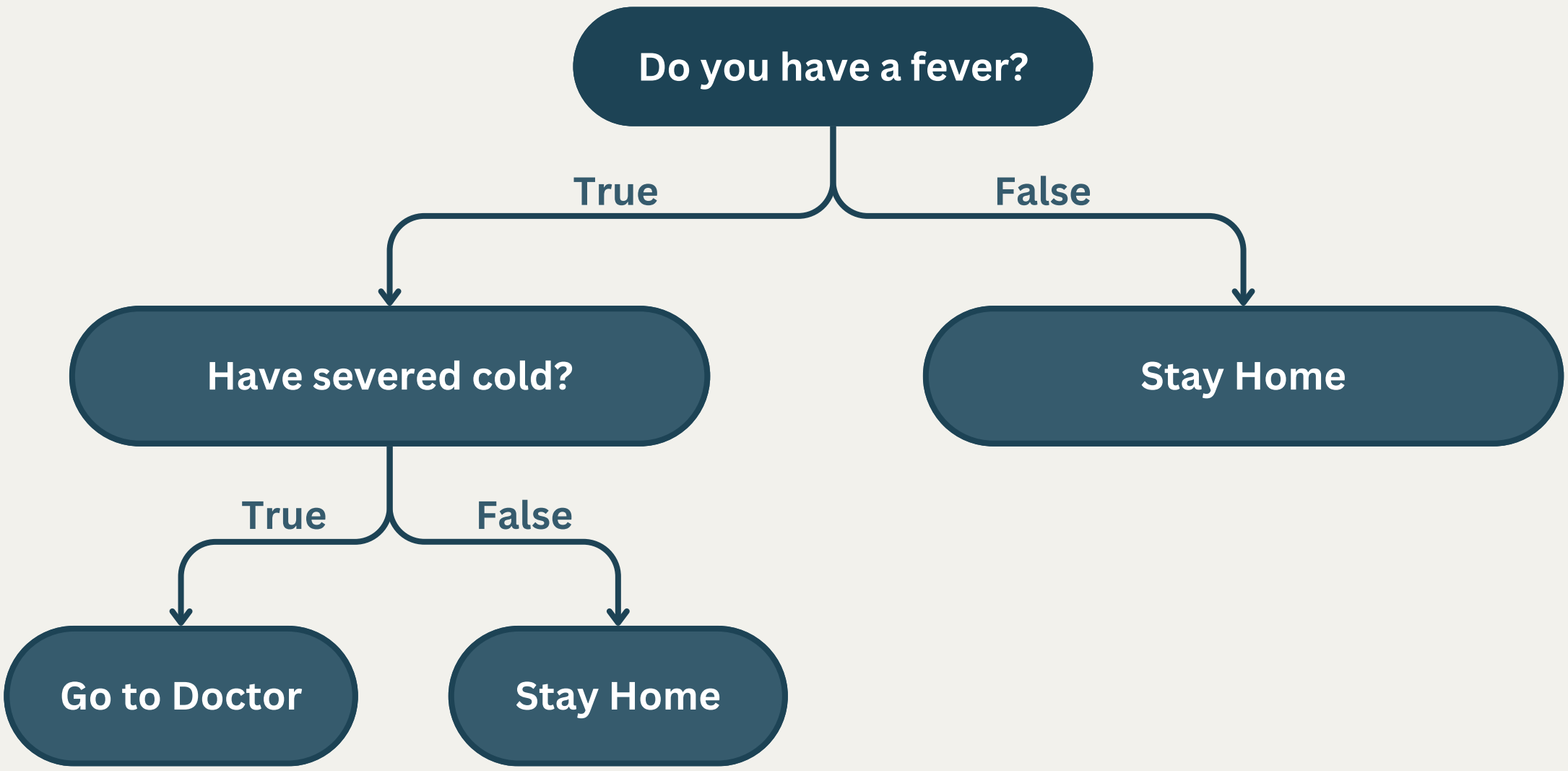
Decision Trees
Clearly Explained



So What Are Decision Trees?

A decision tree is a model used in machine learning that resembles a tree structure. It is a way of **making decisions based on rules**, which are derived from the data.

Say you wanted to decide if you need to go to the doctor or stay at home. Let’s make a Decision Tree to help you decide:

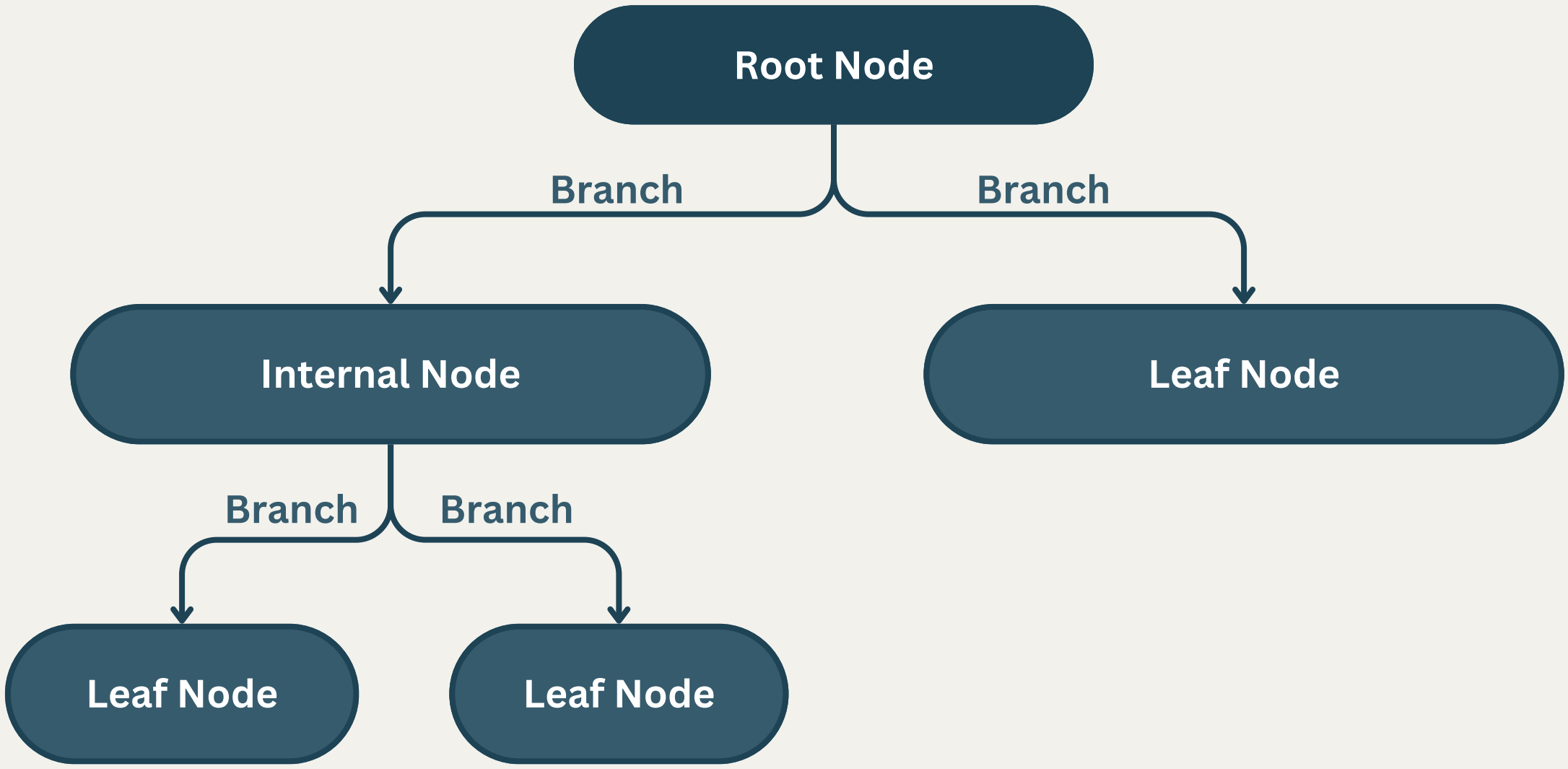


That’s a very simple Decision Tree that would allow you to decide if you need to go to the doctor or stay at home, based on two factors or features : Fever & Cold.

linkedin.com/in/vikrantkumar95

Anatomy of A Decision Tree

Let’s see what a Decision Tree is made of. Here is the previous tree, but with each component labeled:



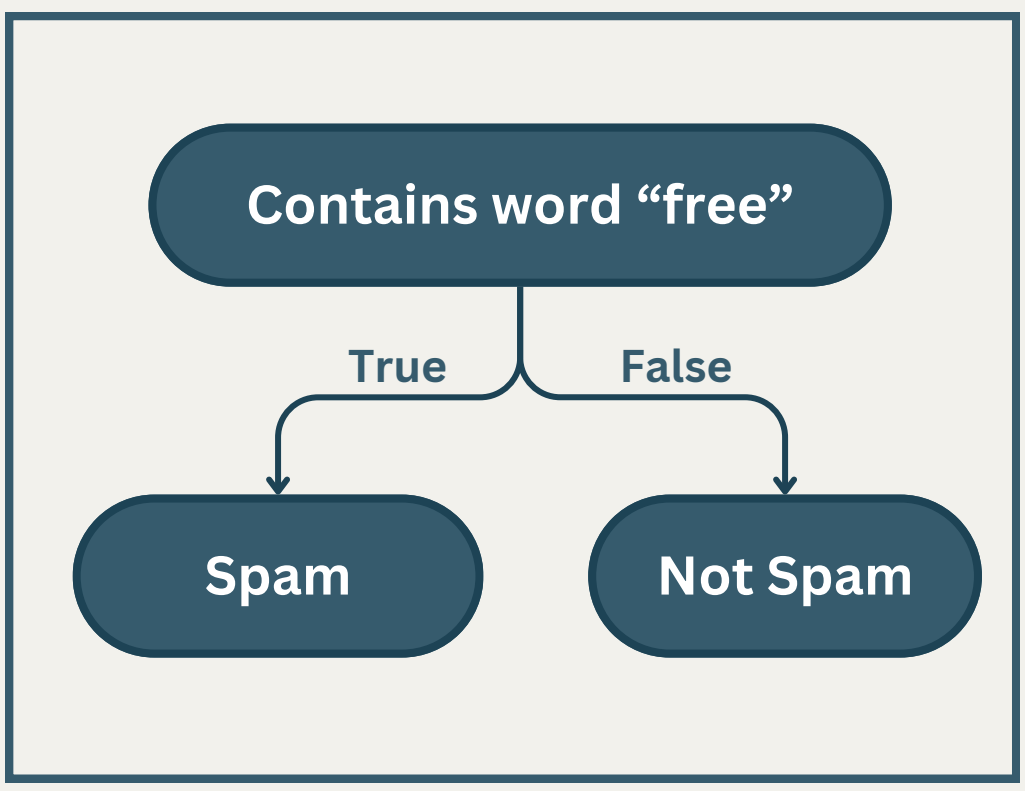
- **Root Node -** The very first node. It is from this node that the **initial decision** is made and the dataset begins to be split.
- **Internal Nodes** - Also known as Decision Nodes, these are the points where the **data is split further**. Internal nodes will have branches coming out of them.
- **Leaf Nodes** - Also known as Terminal Nodes, these nodes represent the **final output** or **decision**, with no further splitting.
- **Branches** - These are the lines connecting nodes, representing the **outcome of a test** and leading to the next internal node or a leaf.

Type of Decision Trees

There are two key types of Decision Trees: **Classification Trees** and **Regression Trees.**

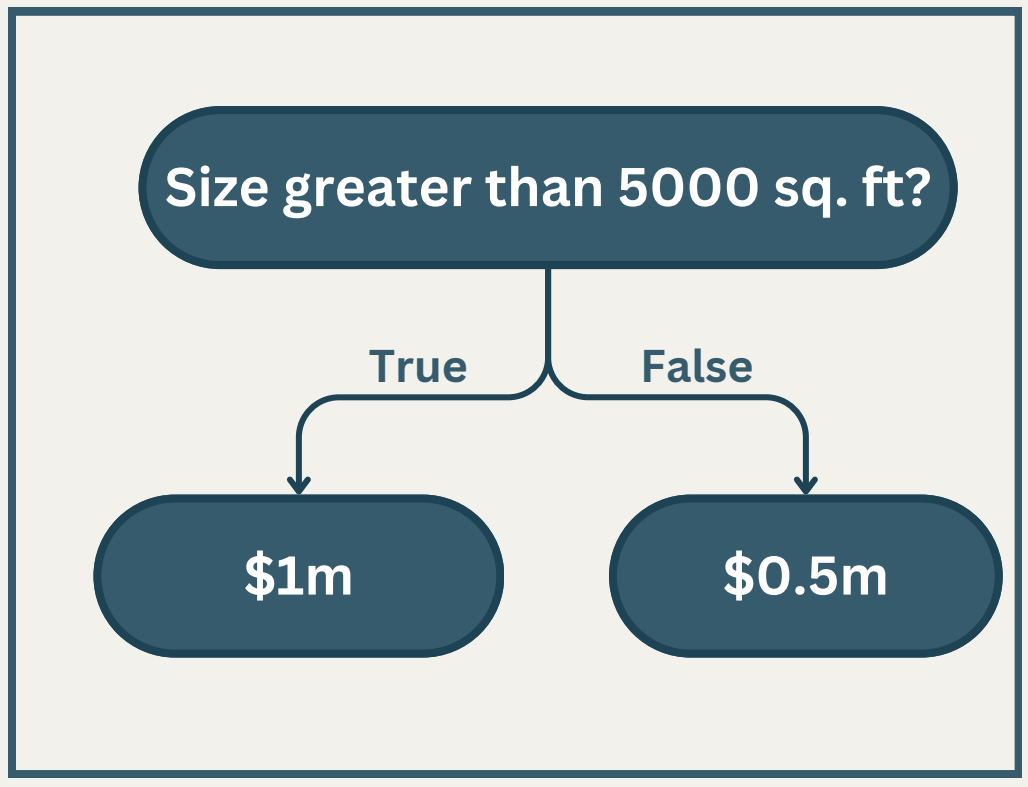
Below is an example of each:

Classification Tree



- Used for **categorical target variables**. They predict a discrete outcome (e.g., 'Spam' or 'Not Spam' in email filtering)

Regression Tree



- Used for **continuous target variables**. They predict a quantitative outcome (e.g., house prices or stock values).

The main difference between the two types of decision trees is the t**ype of outcome** (i.e the target variable) they try to predict. The criteria for splitting the data at each node also differs between the two. In this guide, we will focus on the **Classification Tree** (another guide for Regression Trees coming soon)

linkedin.com/in/vikrantkumar95

How Decision Trees Work

Starting from the root, the tree **splits** data based on **features**, aiming to segregate classes as distinctly as possible. Each split aims to create the **most homogeneous branches.**

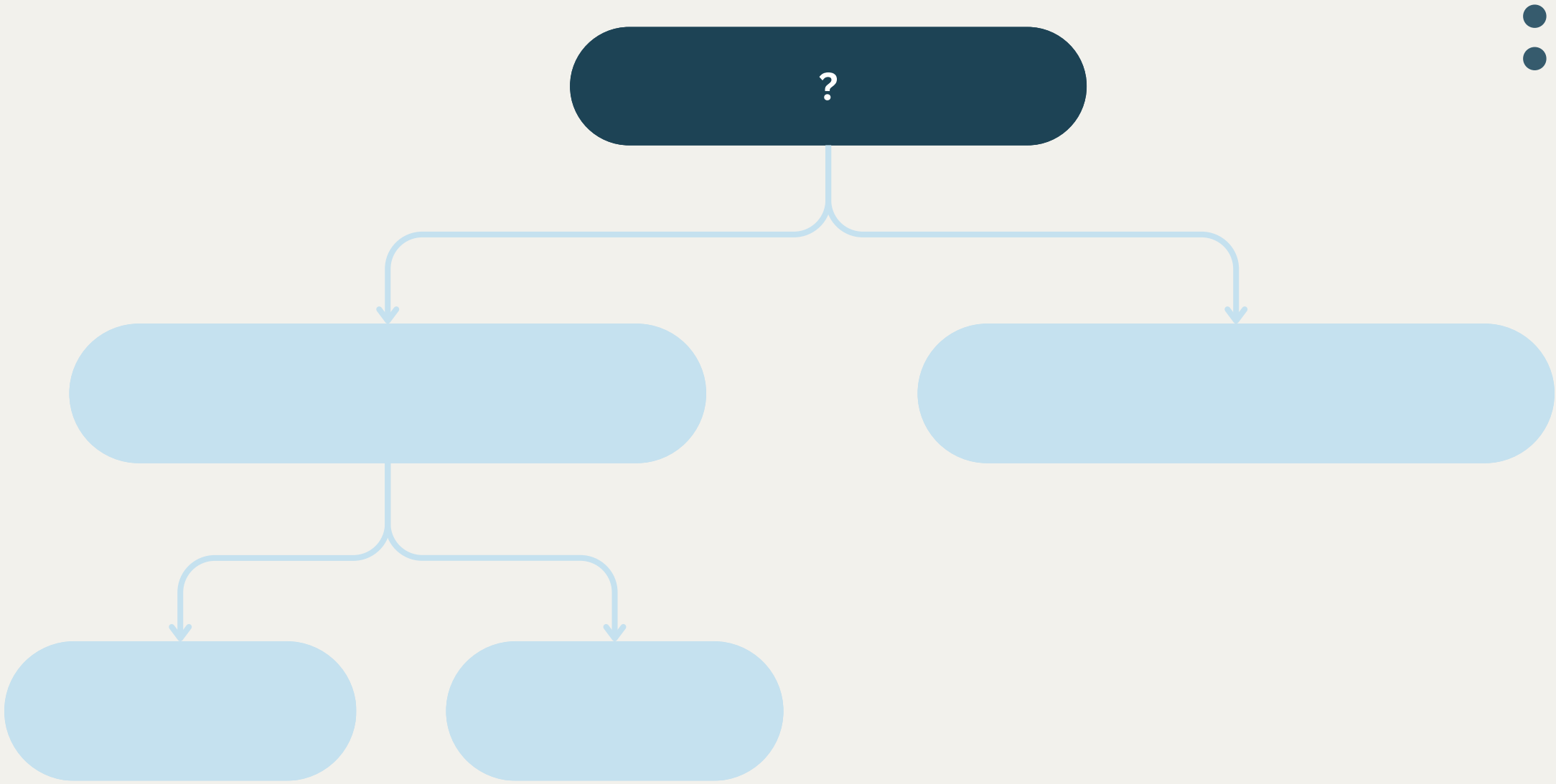
Let us take a sample data set. The goal is to train a Decision Tree to predict if a person will reach **On Time** (*target variable*) based on the **Weather**, type of **Transport** available and the **Distance** (these are the *features*).

S.No	Weather	Transport	Distance (km)	On Time
1	Sunny	Car	5	Yes
2	Rainy	Bus	10	No
3	Sunny	Bus	15	No
4	Rainy	Car	20	No
5	Sunny	Car	25	Yes
6	Rainy	Bus	30	No
7	Sunny	Bus	35	No

An interesting thing to note here is that the data consists of both numeric and categorical variables.

How Decision Trees Work

The first thing we have to decide is what will be the question at the **Root Node** (the first node) that creates the initial split of the dataset.



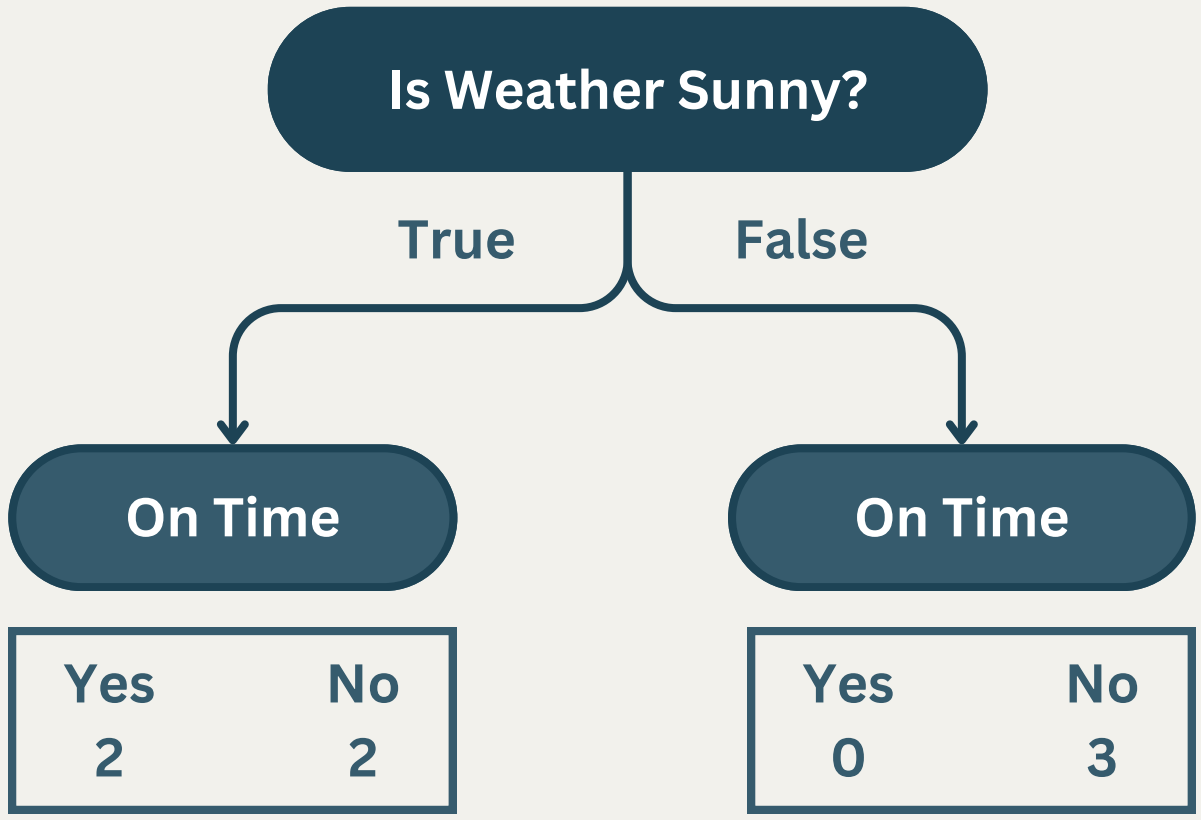
We will do this by looking at each of the three features. The feature that **best predicts the target variable independently** will be the one used to split the dataset in the root node.

So how do we check which feature best predicts the target variable? Let’s take a look in the next slide.

How Decision Trees Work

Let us first create a small tree using only the **Weather** feature.

S.No	Weather	On Time
1	Sunny	Yes
2	Rainy	No
3	Sunny	No
4	Rainy	No
5	Sunny	Yes
6	Rainy	No
7	Sunny	No

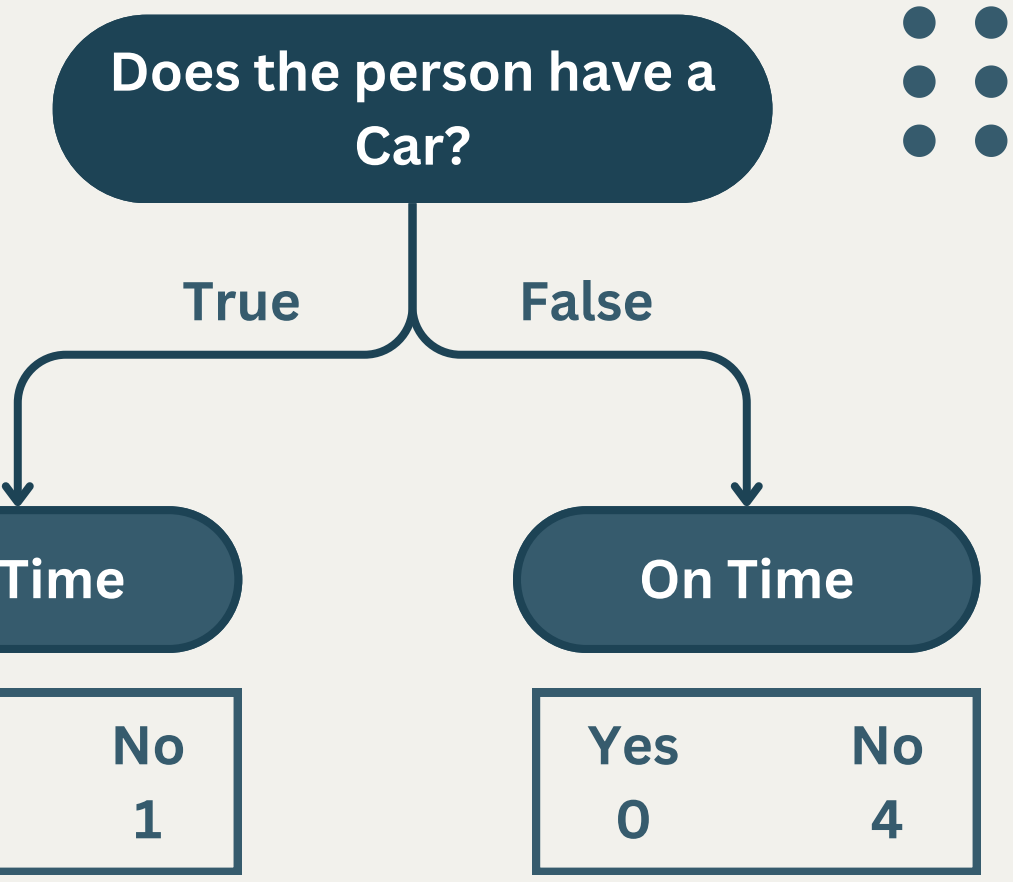


- We went from top to bottom through the data and bucketed each row into one of the nodes.
- For example, the first one is **Sunny**, so it goes to the left node after the initial split. Then we see that the target variable is **Yes**, so we increment the Yes count in the left now.
- Similarly we go through all the rows to build the small tree we see above.

How Decision Trees Work

Now we'll do the same with the **Transport** feature.

S.No	Transport	On Time
1	Car	Yes
2	Bus	No
3	Bus	No
4	Car	No
5	Car	Yes
6	Bus	No
7	Bus	No

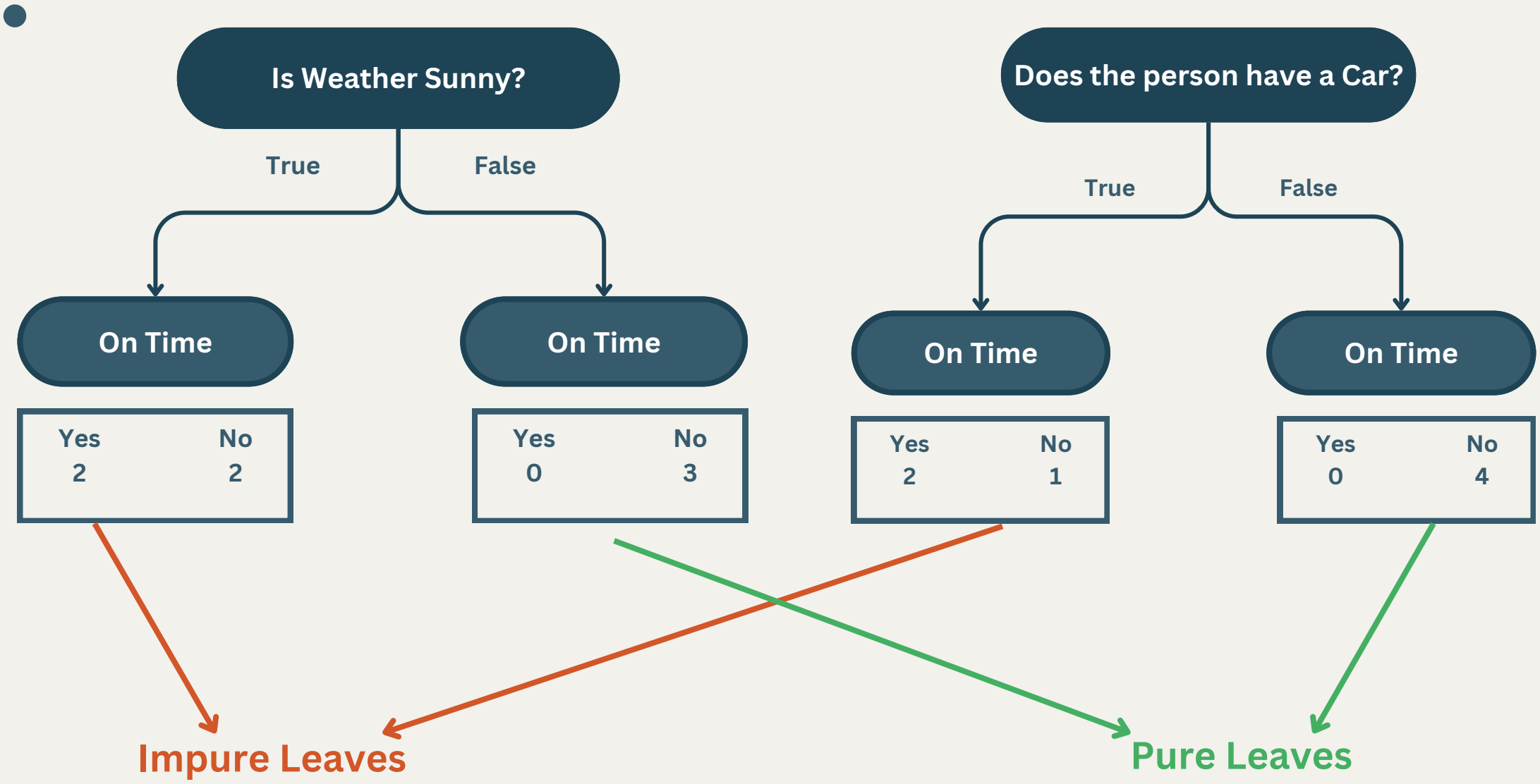


- We built a small tree for the Transport feature exactly how we built it previously for the Weather feature.
- For example, the first one is a **Car**, so it goes to the left node after the initial split. Then we see that the target variable is **Yes**, so we increment the Yes count in the left node.
- Similarly we go through all the rows to build the small tree we see above.

The question is: **Which one of the two trees created a better split?**

How Decision Trees Work

As we mentioned earlier, the goal of each split is to have homogenous branches. In order words, the more **pure** a leaf is after split, the better is the split.



- Both the trees have one impure leaf and one pure leaf. By **purity** we mean that **majority of the target variable belongs to one category**. In both the pure leaves we see that **No** is the only target variable, hence these leaves are a **100% pure leaves.**
- Both trees also have one impure leaf each. So how do we quantify which leaf is more impure? How do we quantify which tree as a whole is more impure? There are a bunch of metrics (Entropy, Chi-square etc) but the most famous one is **Gini Impurity.** This is what we'll be using in this guide.

linkedin.com/in/vikrantkumar95

How Decision Trees Work

Gini Impurity: It is a measure used in decision tree algorithms to quantify the likelihood of a specific instance being incorrectly classified if it were randomly chosen, based on the distribution of classes in the subset.

That is the technical definition. We will delve further into it at a later time but for now all we need to understand is the formula:

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

Where p_i is the proportion of instances of class **i** in the subset

Let's try a couple of examples to get an intuition behind the formula. Here we have a total of **7 balls** that are split in different proportions in two sets. We calculate the Gini Impurity score of each set

Red	Blue	Green
0	3	4

$$Gini = 1 - \left(\frac{0}{7}\right) - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = 0.49$$

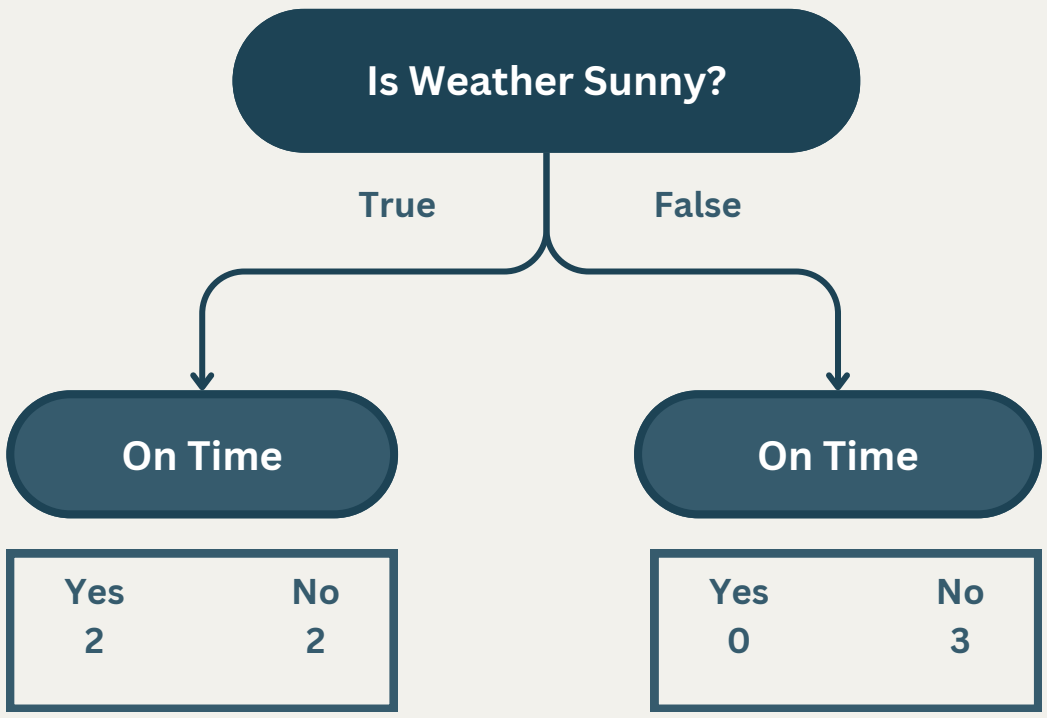
Red	Blue	Green
5	1	1

$$Gini = 1 - \left(\frac{5}{7}\right) - \left(\frac{1}{7}\right)^2 - \left(\frac{1}{7}\right)^2 = 0.45$$

The first example we get a gini impurity of 0.49 and for the second example we get 0.45. This means that the **second set** is **more pure**. The **closer to 0 the purer the set**. A homogenous set would have a 0 gini impurity (try and see how the formula would work if all 7 balls were red)

How Decision Trees Work

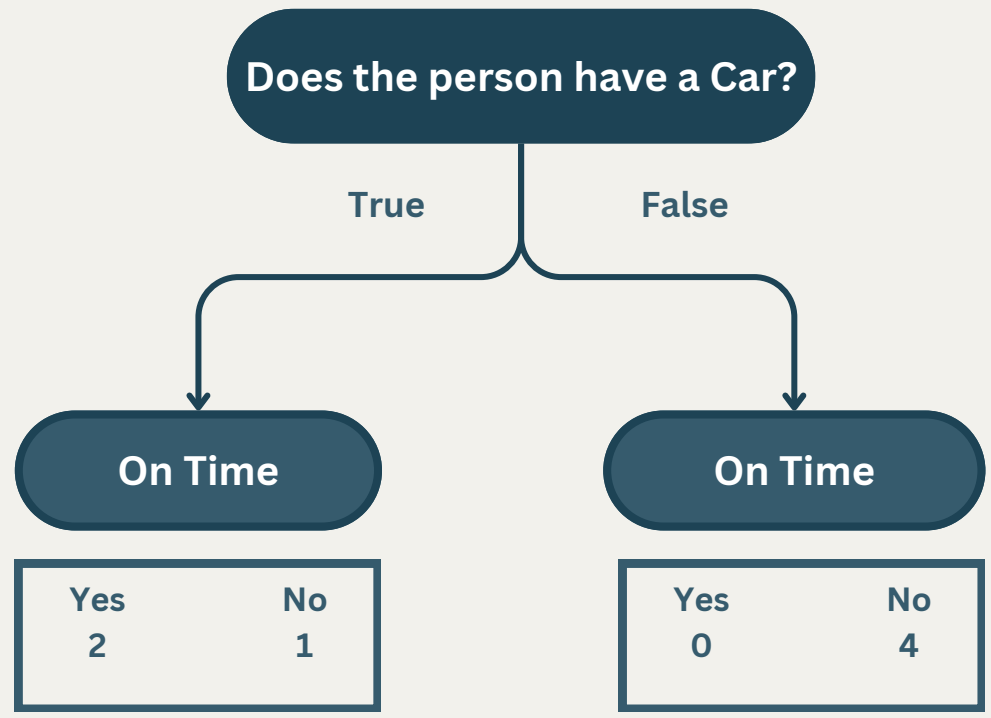
Now let us calculate the Gini Impurity score for the two trees we created:



Left Leaf $$Gini = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

Right Leaf $$Gini = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

Total $$Gini = \left(\frac{4}{4+3}\right)0.5 + \left(\frac{3}{4+3}\right)0 = 0.29$$



Left Leaf $$Gini = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 0.44$$

Right Leaf $$Gini = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$$

Total $$Gini = \left(\frac{3}{4+3}\right)0.44 + \left(\frac{4}{4+3}\right)0 = 0.21$$

- We calculate the Gini Impurity for each leaf first. Then we calculate the weighted average of the Gini Impurity of both the leaves to get the Gini Impurity of the entire tree (weighted by the total data points in the leaf over all of the data points).
- In this case we have a Gini Score of **0.29** for **Weather** and **0.21** for **Transport**

How Decision Trees Work

Now that we have the Gini Impurity scores for Weather and Transport, let's calculate the score for **Distance**. Now since this is a **numeric** feature, the calculation is a bit different.

First we sort the distance in ascending order and calculate the **average distance** between each **pairs of rows**.

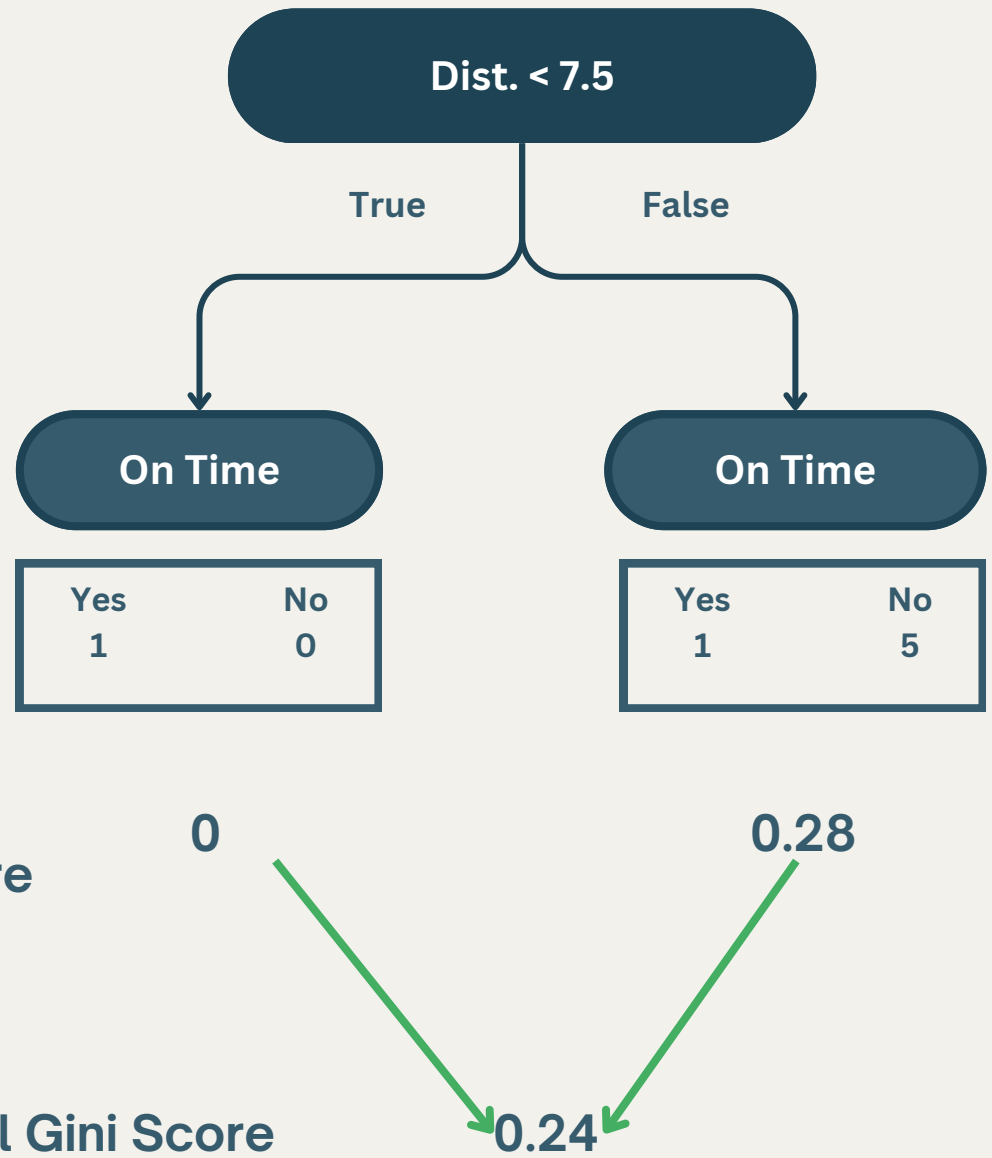
S.No	Distance (km)	On Time	Avg Dist.
1	5	Yes	-
2	10	No	7.5
3	15	No	12.5
4	20	No	17.5
5	25	Yes	22.5
6	30	No	27.5
7	35	No	32.5

We will now calculate the Gini Impurity for each of the Avg Distances.

How Decision Trees Work

We first take **7.5 Avg Distance** and create a Decision Tree

S.No	Distance (km)	On Time	Avg Dist.
1	5	Yes	-
2	10	No	7.5
3	15	No	12.5
4	20	No	17.5
5	25	Yes	22.5
6	30	No	27.5
7	35	No	32.5



- Since only 1 data point had a Distance less than 7.5, the left leaf only has one element. The right leaf has 6, consisting of 5 No & 1 Yes.
- We calculate the Gini Impurity score for both the leaves (0 & 0.28) and then calculate the weighted avg score for the whole tree (**0.24**).
- **We repeat this process for all the rows.**

How Decision Trees Work

We calculate the Gini Impurity for each row. The following are the results we get:

S.No	Distance (km)	On Time	Avg Dist.	Gini Impurity
1	5	Yes	-	
2	10	No	7.5	0.24
3	15	No	12.5	0.371
4	20	No	17.5	0.405
5	25	Yes	22.5	0.405
6	30	No	27.5	0.343
7	35	No	32.5	0.381

- We pick the **minimum Gini Impurity** from amongst all the rows and take that as the Gini Impurity Score for the **Distance** feature.

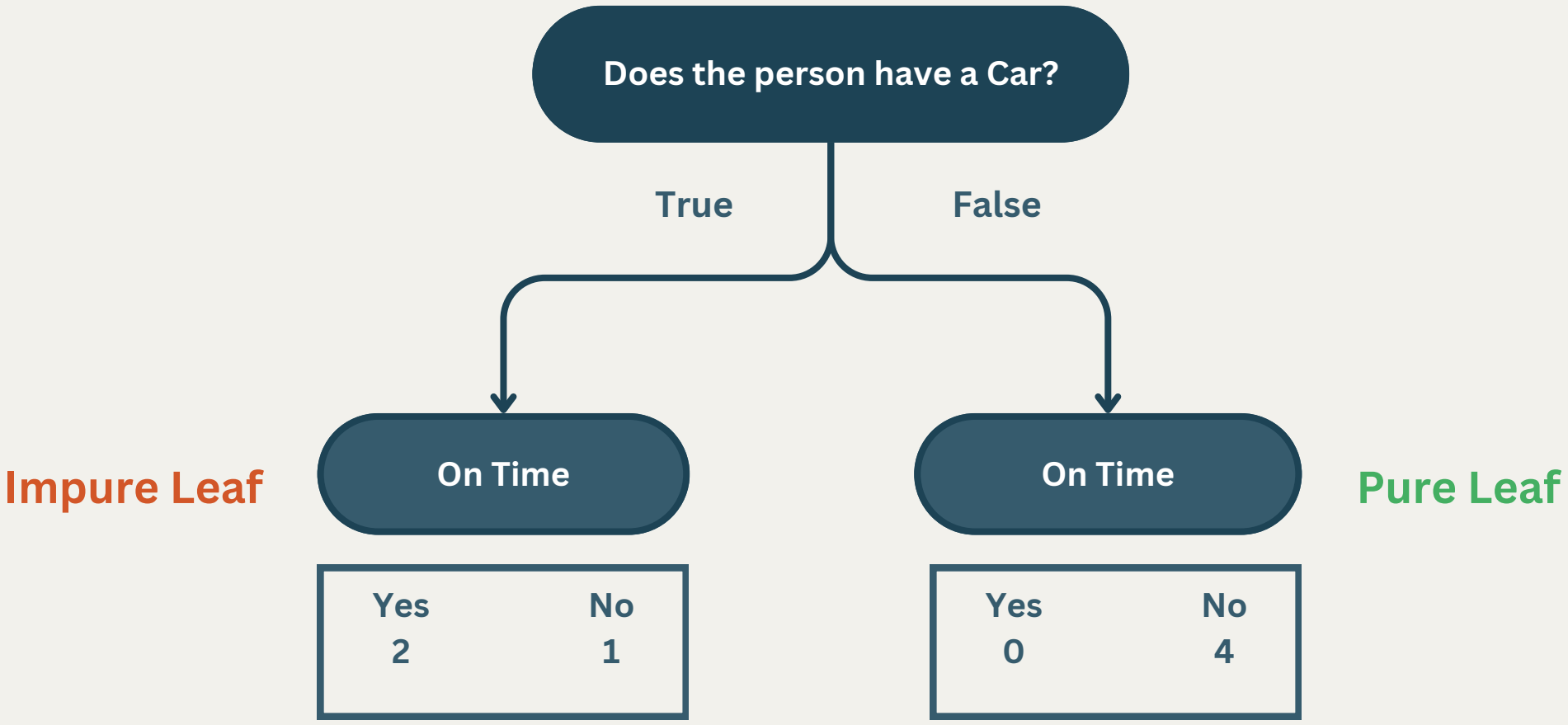
- So now we have **0.24** as the Gini Impurity Score for the **Distance** feature.

How Decision Trees Work

We have the following Gini Impurity Score for each of the three features:

Weather	Transport	Distance
0.29	**0.21**	0.24

Since **Transport** has the lowest Gini Impurity score, we **split** the **Root Node** on that feature.

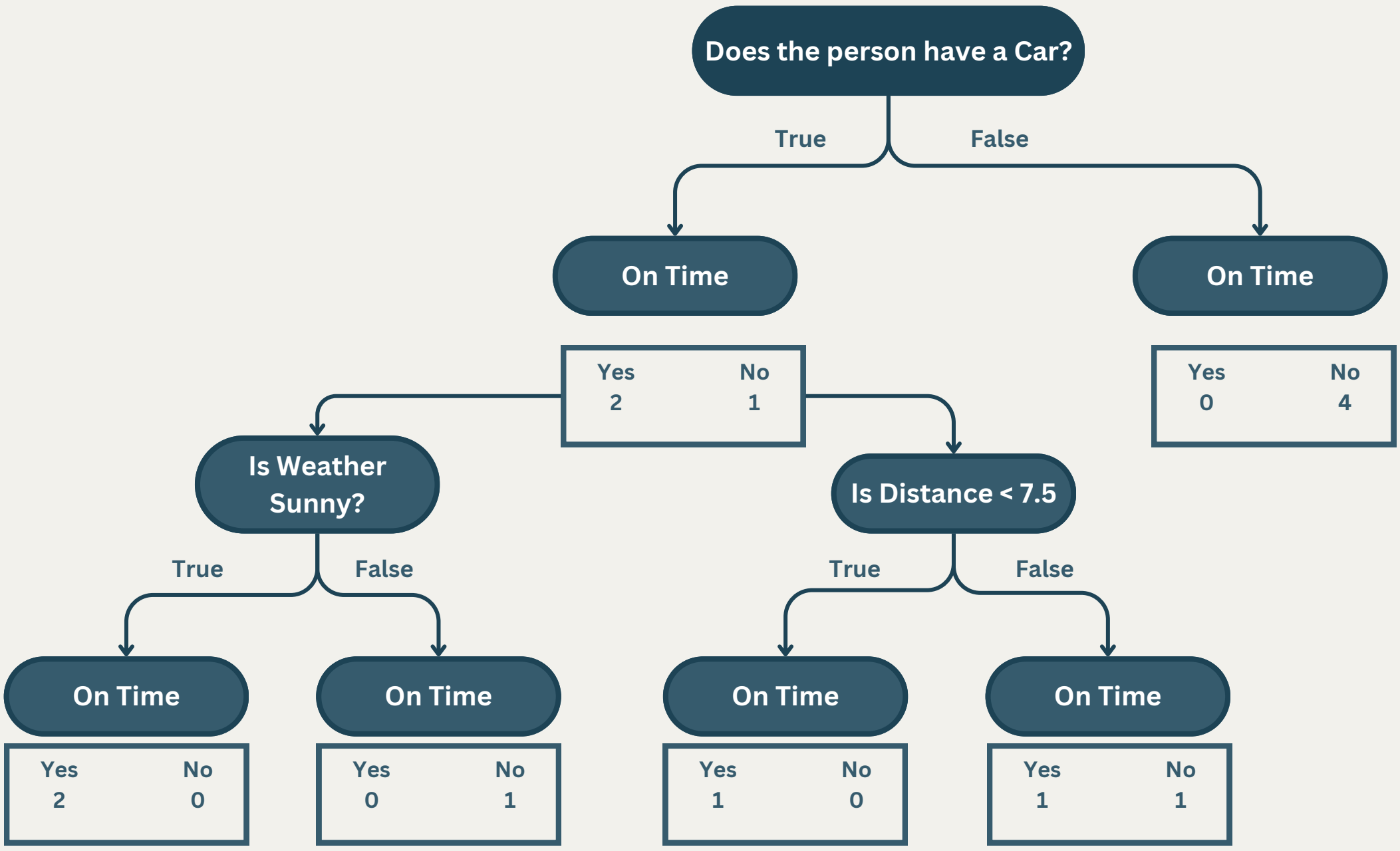


- After the first split we see that the Right Leaf is pure and does not need to be split further
- The Left Leaf however is impure. Let us see if we can split it further.

How Decision Trees Work

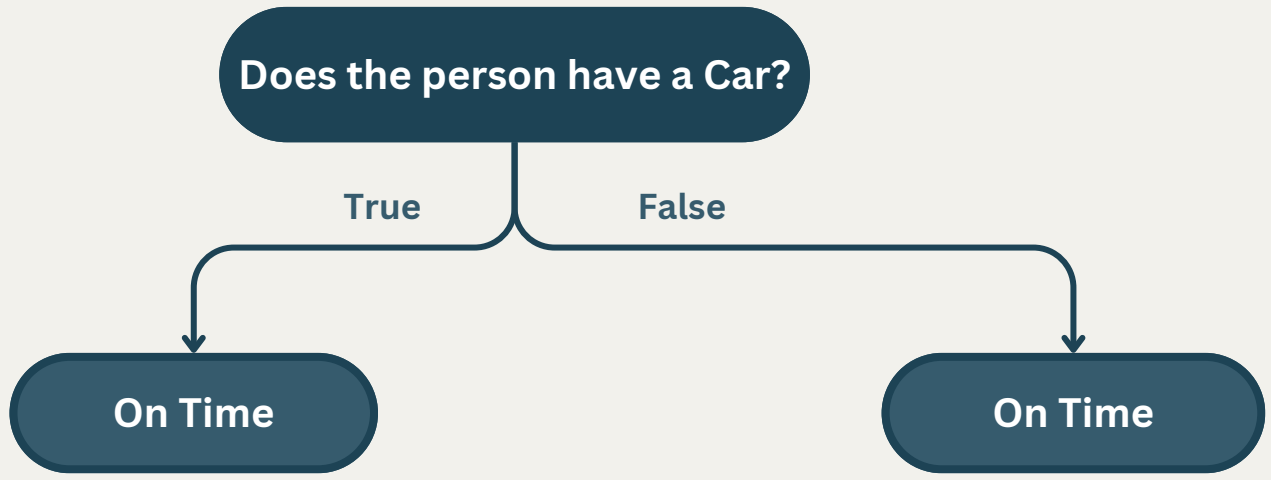
Since we are only looking to split the Left Node (people who have a Car), we only take the data with **Car** as the **Transport**.

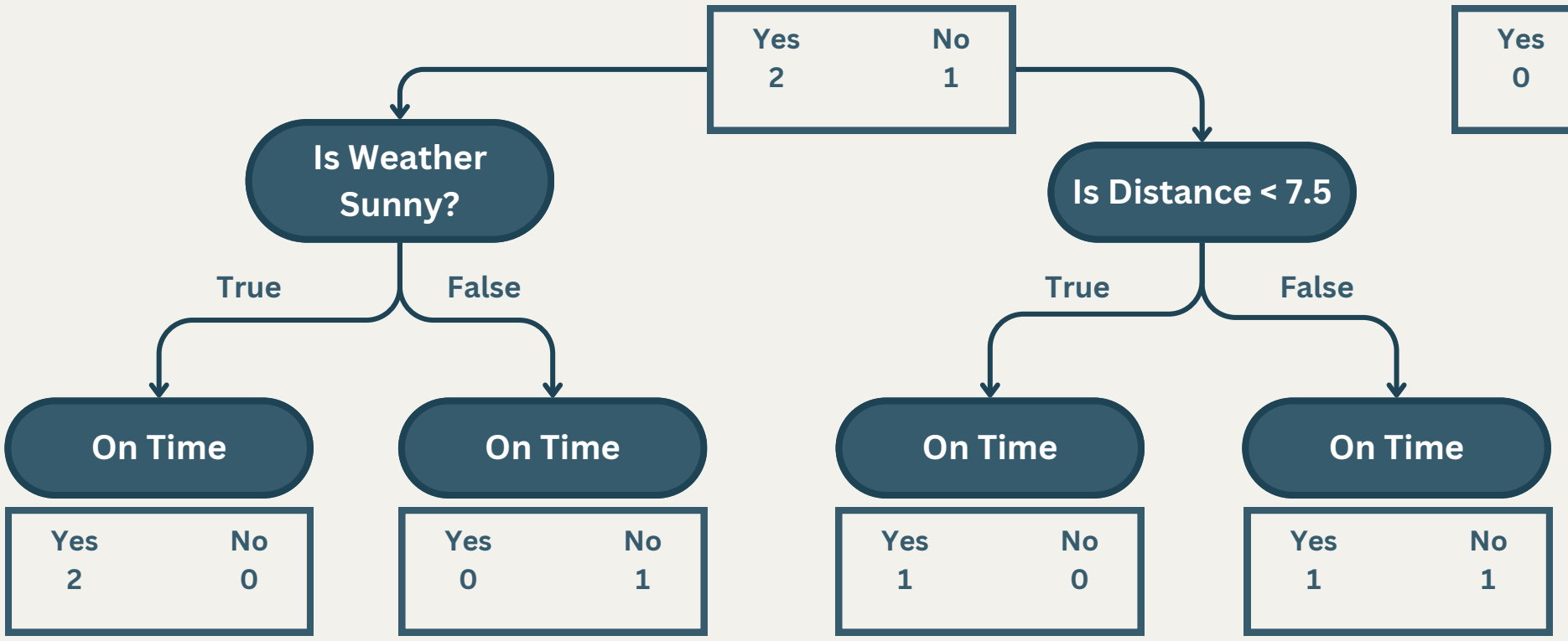
S.No	Weather	Transport	Distance (km)	On Time
1	Sunny	Car	5	Yes
4	Rainy	Car	20	No
5	Sunny	Car	25	Yes



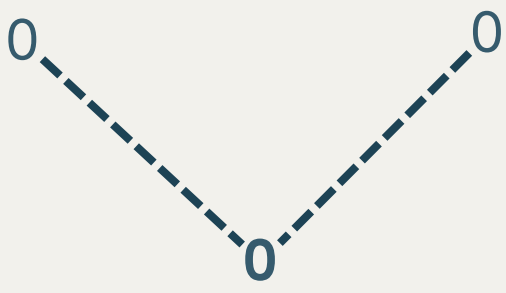
Let’s calculate the Gini Impurity for each of the two possible splits

How Decision Trees Work





Gini Impurity

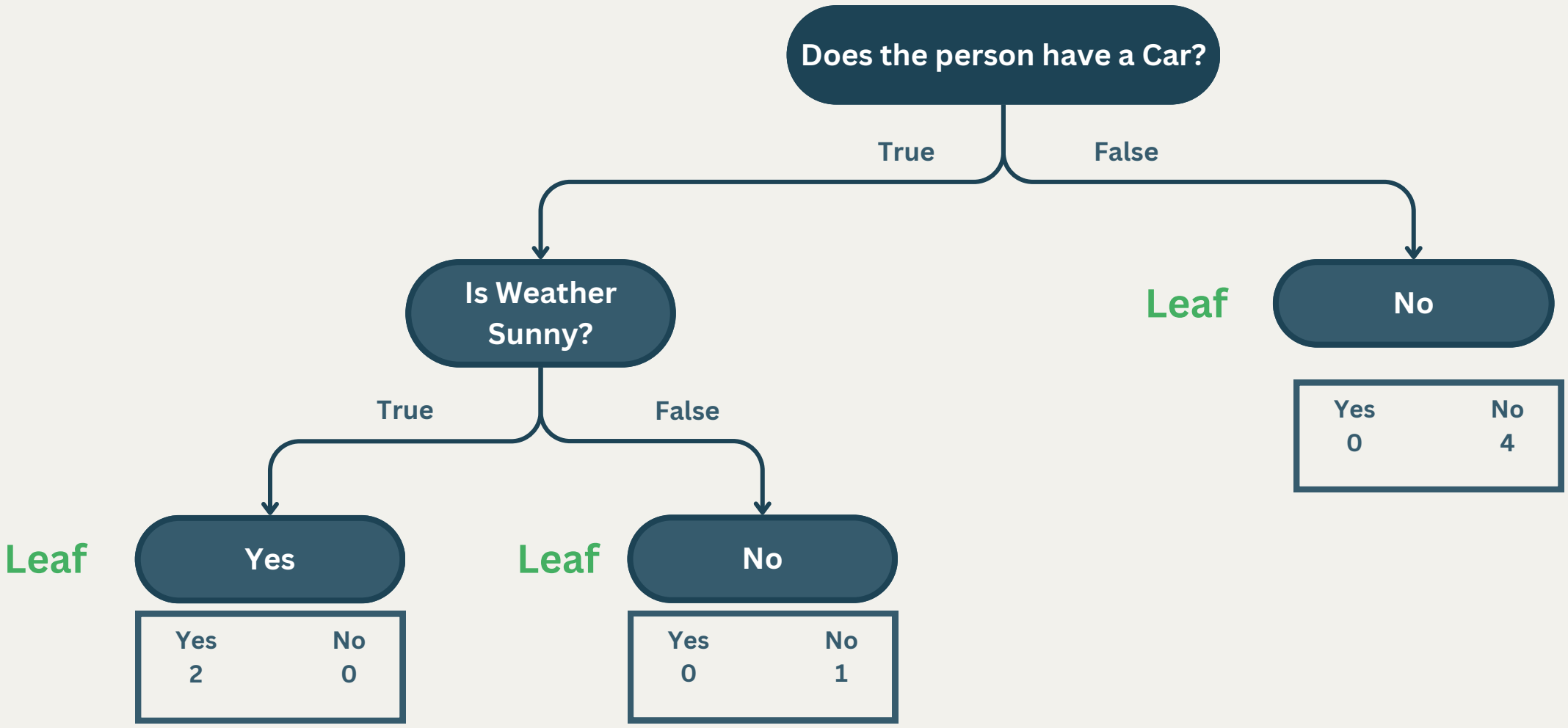


Total Gini Impurity

- We see that if we were to split further by **Distance** we would get a Gini Impurity Score of 0.33
- We also see that if we were to split further by **Weather** we would get a Gini Impurity Score of 0 (perfect split)
- We will then split by **Weather** as that has the **lower Gini Impurity Score**

linkedin.com/in/vikrantkumar95

How Decision Trees Work



Hurray! This is the **final tree**!

- We have 3 leaf nodes in the end. Each of the Leaf Node is **assigned value based** on the **majority class** (We have 2 leaves with No as the majority and 1 leaf with Yes as the majority).
- Notice that the **Distance** feature was not used in this tree. This is a common occurrence. Not all features will always be used when training a Decision Tree. The **use of features in a decision tree is determined by how well they contribute to splitting the dataset.**
- Also notice that the middle Leaf Node has only 1 data point. This makes us doubtful if the model will generalise well (i.e give accurate predictions on unseen data). In our example, we have likely **overfit** the model.

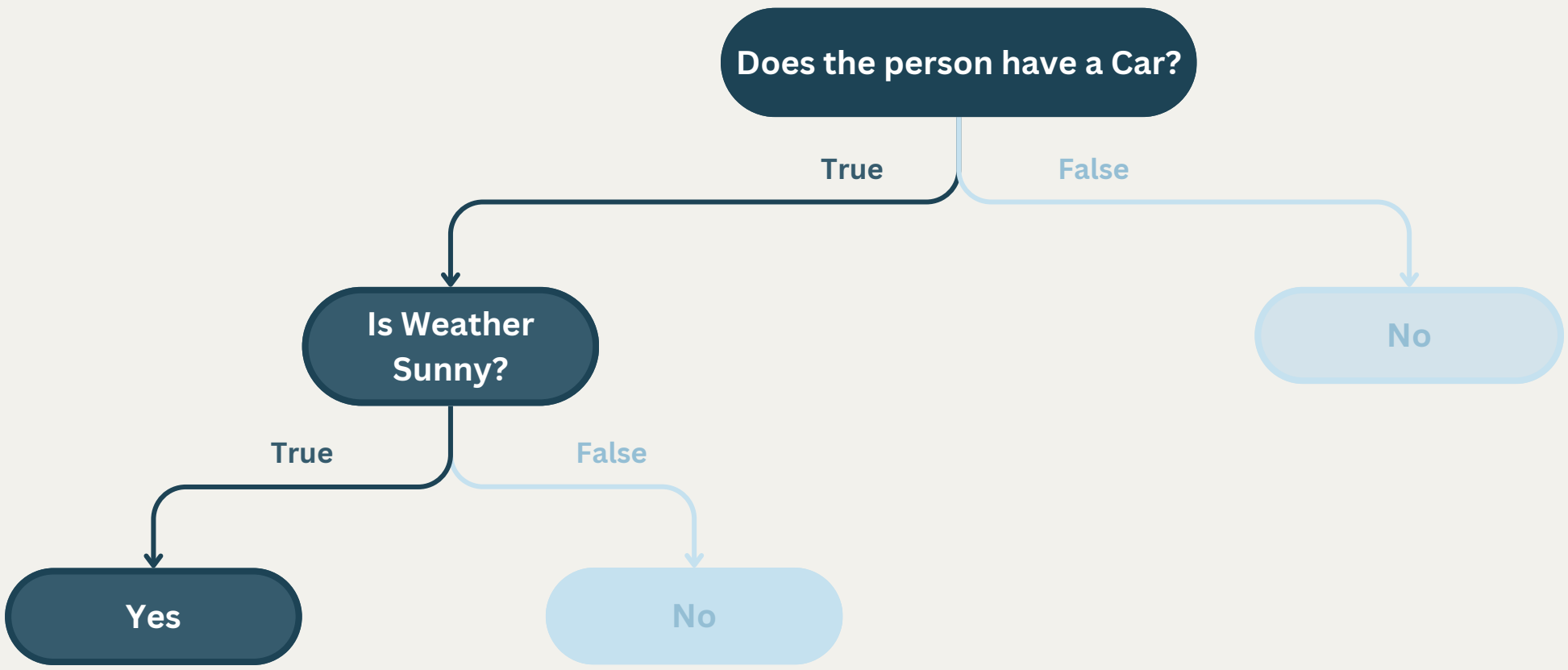
linkedin.com/in/vikrantkumar95

Making Predictions with the Tree

Suppose we had to predict if the following person will make it to their destination on time:

S.No	Weather	Transport	Distance (km)	On Time
1	Sunny	Car	10	?

We would go from top to bottom and see which Leaf we end up in:



S.No	Weather	Transport	Distance (km)	On Time
1	Sunny	Car	10	Yes

Based on our Decision Tree, the person would indeed **reach on time!**

What's Next?

Today, we've unlocked the basics of Decision Trees, a powerful tool in data science and machine learning. As we continue our journey, look forward to diving deeper into:

- **Advanced Algorithms**: Explore variations like Random Forests and Boosted Trees, and understand how ensemble methods enhance prediction accuracy.
- **Feature Engineering**: Learn the art of crafting powerful features and the impact of feature selection on the performance of Decision Trees.
- **Handling Overfitting**: Delve into strategies like pruning, cross-validation, and setting optimal tree complexities to build more generalizable models.
- **Hybrid Models:** Discover how Decision Trees are combined with other algorithms to solve complex problems in innovative ways.
- **Real-World Applications**: See Decision Trees in action across various domains, from financial risk assessment to medical diagnosis.

linkedin.com/in/vikrantkumar95

Enjoyed reading?

Follow for everything Data and AI!

linkedin.com/in/vikrantkumar95