Classification with Decision Trees

Brian Kim

Motivation

Suppose a car company wants to decide which cars to buy ad time for in the next Super Bowl. They have a dataset of cars promoted in past Super Bowl ads, and information about the **miles per gallon** (`mpg`) and **horsepower** (`hp`) of each car. In addition, they know which ad campaigns were deemed **"Successful"** or **"Unsuccessful."** Which cars should the company buy ad time for?

Motivation

Suppose a car company wants to decide which cars to buy ad time for in the next Super Bowl. They have a dataset of cars promoted in past Super Bowl ads, and information about the **miles per gallon** (`mpg`) and **horsepower** (`hp`) of each car. In addition, they know which ad campaigns were deemed **"Successful"** or **"Unsuccessful."** Which cars should the company buy ad time for?

At its heart, this is a **prediction** problem. In order to **decide which cars to buy ad time for**, we want to use **data from previous ad campaigns** to make a **prediction** on which cars will have successful ad campaigns.

Prediction

To do this, we'll use a **machine learning** technique called **decision trees**. More specifically, this is a subset of machine learning called **supervised learning**.

Two branches of machine learning:

- **Unsupervised Learning** is when you have no **outcome variable** and are trying to create groupings based on the data.
- **Supervised Learning** is when you have an **outcome variable**, and you try to make a prediction about future data points.

What is Machine Learning?

Machine learning is when a machine (that is, a computer) **improves** on a **task** with respect to some **performance measure**.

In the case of **supervised learning** for Super Bowl car advertisements, we want to develop a way to **improve** how well we **predict future successful car advertisements**.

General Supervised Learning Framework

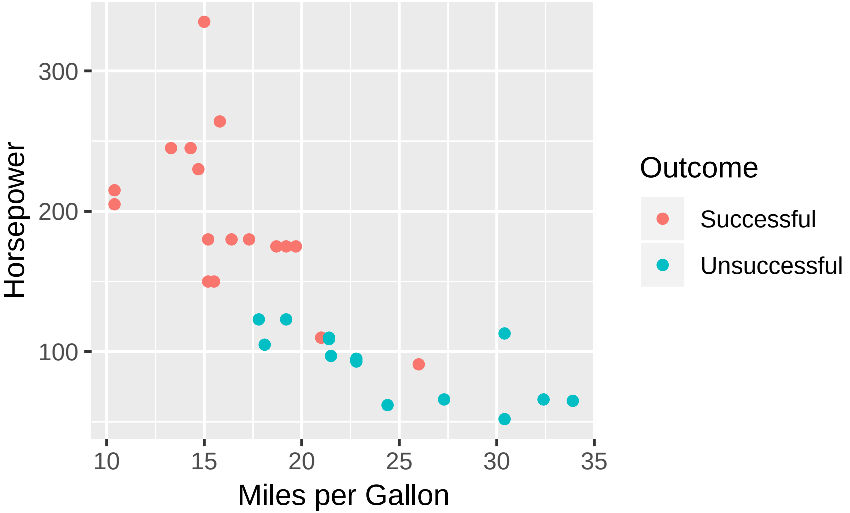
In supervised learning, we are focused on finding the relationship between a **label** y and **features** x.

$$y = f(x)$$

"Learning" is finding a function f that minimizes **future error** in recovering y.

In our case, we will use **decision trees** to find that function f to minimize future error when trying to determine **success of car advertisement campaigns** (y) based on **MPG and HP** (x)

Visualizing the Data

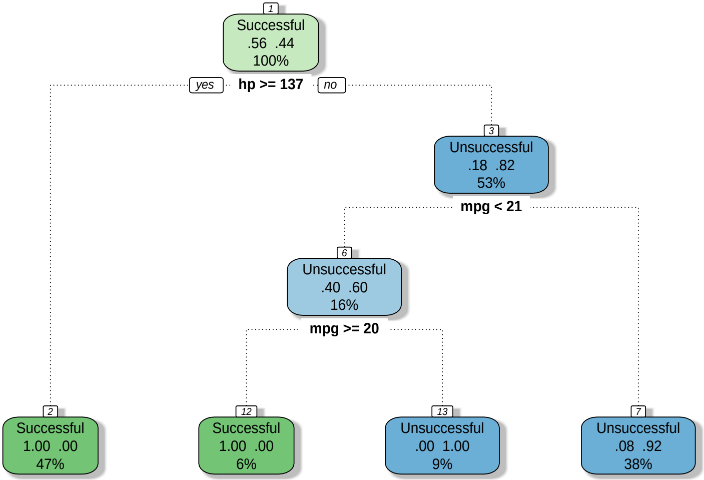


How it Works

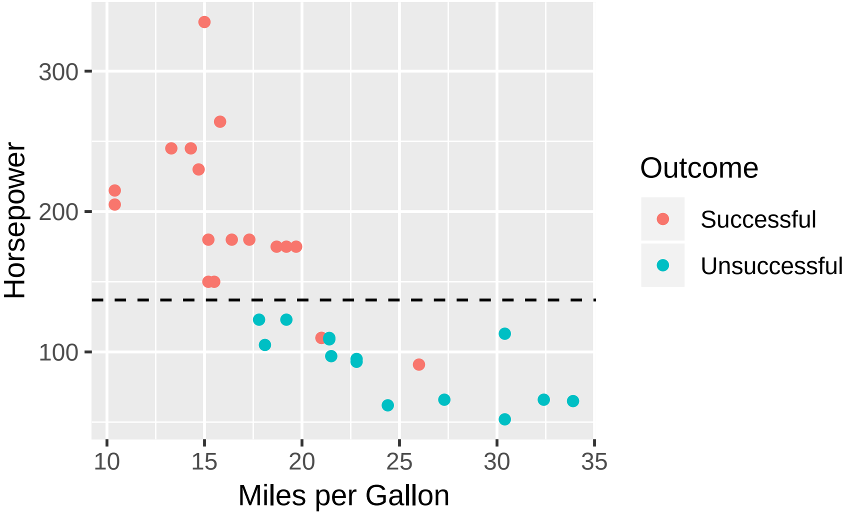
Rough outline of the tree building algorithm

- Consider each feature (in this case, horsepower and MPG).
- Order the values in each feature.
- Compute a measure of how well a partition would split the data for each possible break between values.
- Make the split that maximizes this value.
- Repeat until you reach a stopping criteria.

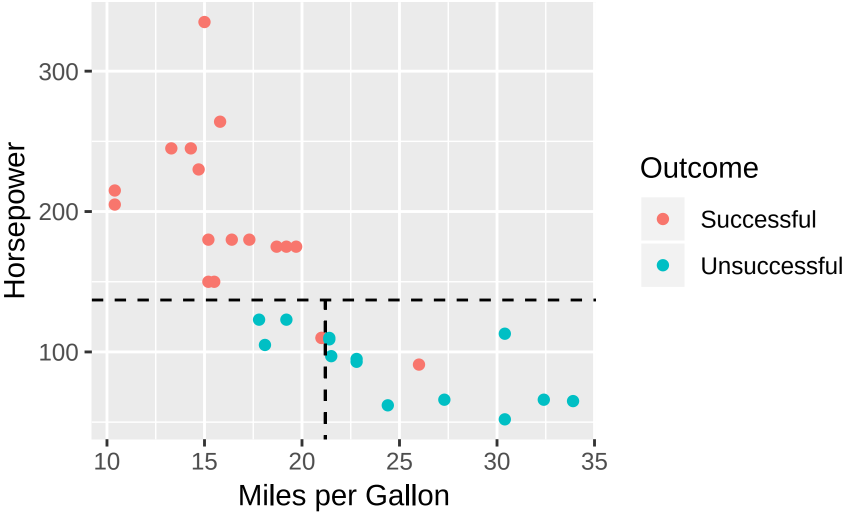
Tree Diagram



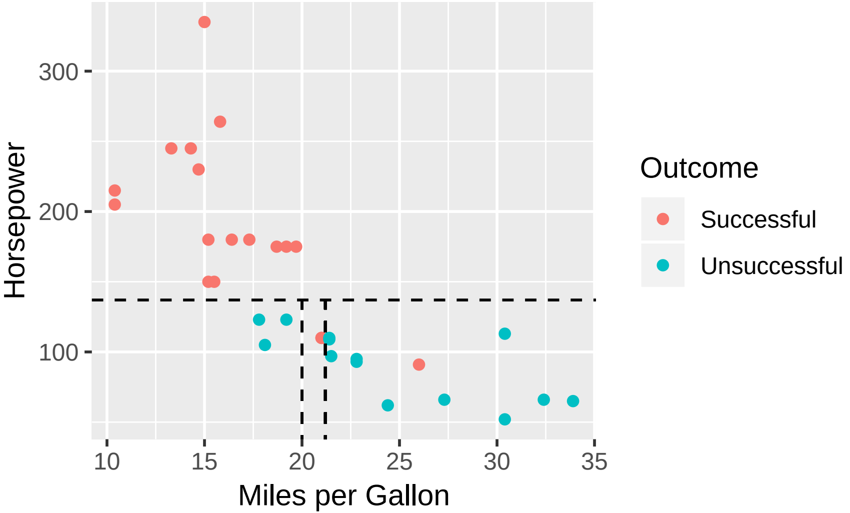
Partitioning



Partitioning



Partitioning



Tuning a Tree

Note that there are multiple places where we need to make decisions on parameters for building a tree.

- Measure for optimal splitting (Gini impurity, information gain).
- Minimum node size for stopping splits.
- Max depth

How should we determine each of these parameters?

By using a **training** set and a **testing** set.