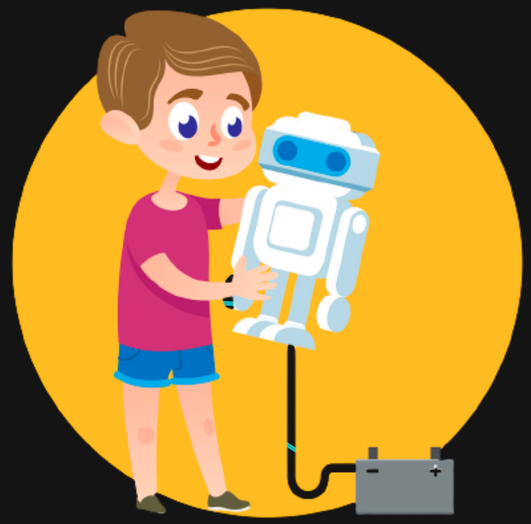
Information Guide

ML Algorithms
Explained to a 5 year old



Let's dumb it down!

Part 1

Bhavishya Pandit

Contents

What you need to know

Part - 1

Linear Regression

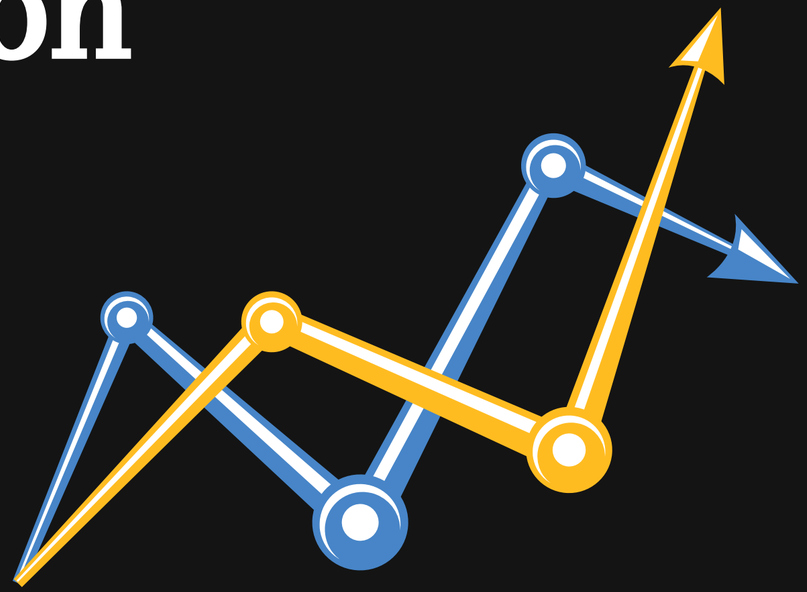
Linear regression is a way to make a line that best fits a bunch of points on a graph. Think of it like trying to connect dots on a piece of paper.

If we want to know how much a toy costs based on how many we buy, we can make a graph with on the vertical axis and on the horizontal axis.

Then we put dots on the graph for each time we buy a toy and how much it costs.

Next, we look at all the dots and try to make a straight line that goes through as many of the dots as possible.

This line is the "best fit" line and it can help us predict how much a toy will cost based on how many we want to buy.

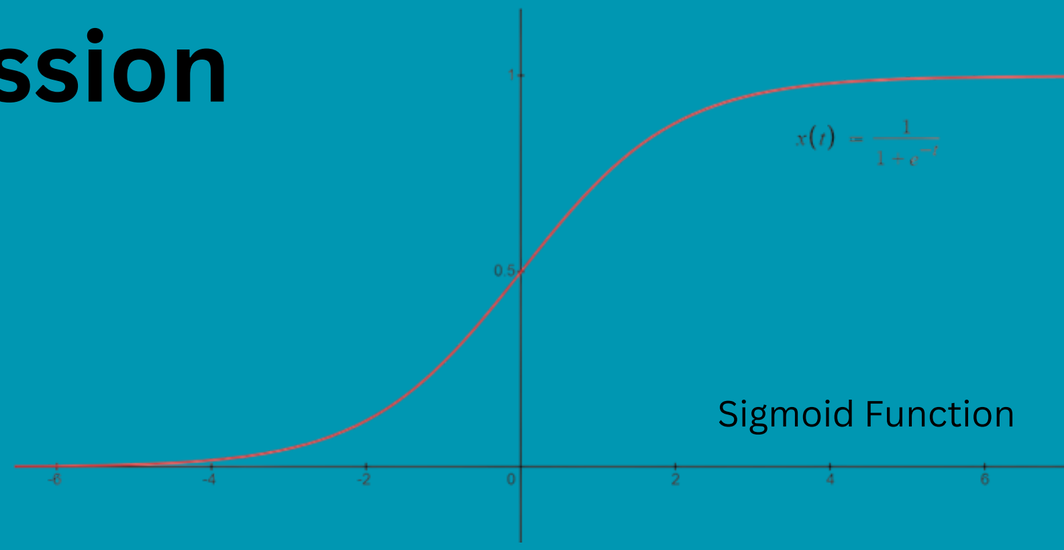


Mathematically, this line is represented by an equation called y = mx + b.

y is the cost of the toy.
x is the number of toys we want to buy.
m is the steepness of the line.
b is the point where the line crosses the vertical axis.

bhavishya-pandit

Logistic Regression



Sigmoid Function

Logistic Regression is a way to predict whether something is true or false (yes or no) based on some information.

Imagine we have a bag of toys and some of them are soft and some are hard. We want to predict if a toy is soft or hard based on its color.

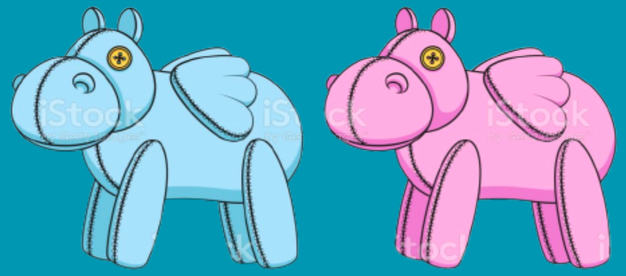
We have some toys that are pink and some that are blue.

We know that most pink toys are soft, while most blue toys are hard.

We can make a chart with on the vertical axis and on the horizontal axis to see if a toy is soft or hard based on its color.

Then we put dots on the chart for each toy we have.

We can use this information to predict if a new toy is soft or hard based on its color.



Mathematically, logistic regression uses an equation called the sigmoid function to predict the probability of something being true or false.

The sigmoid function is an S-shaped curve that maps any input value to a value between 0 and 1.

The equation for the sigmoid function is: $\sigma(z) = 1 / (1 + e^{(-z)})$.

z is the linear combination of the inputs and weights, plus a bias term.

w represents the weight of each input feature.

bhavishya-pandit

Support Vector Machine

Support Vector Machine (SVM) is a way to make a line that separates different groups of points on a graph. Think of it like putting a wall between different groups of toys.

If we want to know if a toy is a car or a truck, we can make a graph with the toy's speed on one axis and its weight on the other axis.

Then we put dots on the graph for each toy we have, with red dots for cars and blue dots for trucks.

Next, we try to find a line that separates the red dots from the blue dots as much as possible.

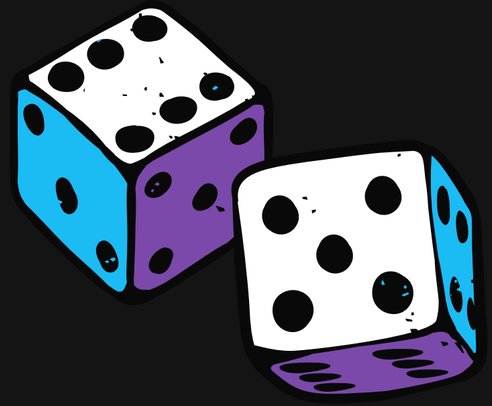
This line is called the "maximum margin" line and it helps us to predict if a new toy is a car or a truck based on its speed and weight.

Mathematically, this line is represented by an equation called y = wx + b.

y is the prediction of the toy being a car or a truck.
x is a vector of the toy's speed and weight.
w is the weight of each feature (speed and weight) in the prediction.
b is the bias term.

bhavishya-pandit

Naive Bayes



Naive Bayes is a way to predict the likelihood of something happening based on some evidence.

Imagine that you want to know if it's going to rain today.

You know that when it's sunny, it rarely rains, but when it's cloudy, it's more likely to rain.

You can make a chart that shows how often it rains when it's sunny or cloudy.

You can use this information to predict if it will rain today based on the weather conditions.

bhavishya-pandit

Mathematically, Naive Bayes uses the Bayes theorem to calculate the probability of an event occurring given some evidence.

Bayes theorem states that the probability of an event A given some evidence B is equal to the probability of B given A, multiplied by the prior probability of A, divided by the prior probability of B.

The formula for Bayes theorem is: $P(A|B) = P(B|A) * P(A) / P(B)$.

$P(A)$ is the prior probability of A (the probability of A before we have any evidence).

$P(B|A)$ is the likelihood of B given A (the probability of observing the evidence B, given that A has occurred).

$P(B)$ is the prior probability of B (the probability of observing the evidence B).

To find the best prediction, Naive Bayes assumes that all features are independent of each other.

K-Nearest Neighbour

KNN (K-Nearest Neighbors) helps predict something based on the K closest data points.

For example, to predict if a fruit is an apple or orange, we can plot the fruits' weights and colors on a graph .

The K nearest neighbors of a new fruit will determine its classification.

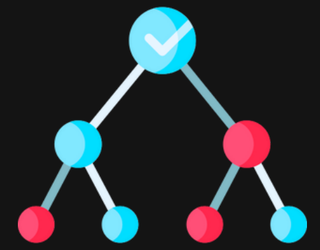
To calculate error, we use accuracy.

For instance, if we made 100 predictions and 80 of them are correct, the accuracy is 80%.

KNN is like asking friends for help to determine the type of fruit.

bhavishya-pandit

Decision Tree



A decision tree helps you make choices by asking yes/no questions.

Imagine you want to choose a pet.
😺 You can make a decision tree to help you decide what kind of pet you should get.

First, you can ask yourself: do I want a pet that's big or small?

If you say "small," then you can ask: do I want a pet that's quiet or noisy?

If you say "quiet," then you can ask: do I want a pet that's furry or not furry?

If you say "furry," then your decision tree tells you to get a cat! 😺

To calculate the error in decision trees, we can use a metric like entropy or Gini index.

Entropy measures the amount of information disorder in the data.

To calculate entropy, we sum the probability of each possible outcome multiplied by the log of that probability.

The Gini index measures how often a randomly chosen element would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset.
The lower the Gini index, the better the split.

bhavishya-pandit

Random Forest

Random forest is a group of decision trees that work together to make better predictions.

Imagine you want to predict if it will be a sunny day tomorrow.

You can make a decision tree to help you decide. You ask questions like "what was the weather like today?", "what is the forecast for tomorrow?", and "what is the temperature like?"

But decision trees can make mistakes! ⚠ They might be too simple, too complex, or too overfit.

That's why we can use a random forest. Instead of just one decision tree, we use many decision trees!

Each decision tree in the random forest makes its own prediction. The final prediction is the most popular prediction of all the decision trees.

So if most decision trees in the forest predict "sunny," then the final prediction will be "sunny"! ☀

To calculate error, we use metrics like accuracy or mean squared error.

For eg, if we have a dataset of people who want to buy a car, and we want to predict if they will buy a car or not, we can use a random forest.

Each decision tree will make its own prediction, and the final prediction will be the most popular one.

To calculate error, we can count the number of incorrect predictions and divide by the total number of predictions. The result is the accuracy of the random forest.

bhavishya-pandit

K-MEANS

K-means is a way to group things that are similar.

Imagine you have a bunch of toys and you want to put them in boxes with other toys that are similar.

You can use k-means to do this! First, you pick a number k, which is how many boxes you want.

Let's say you want 3 boxes.

Then you pick some toys and put them in one of the boxes.

Next, you look at all the toys and figure out which box each toy is closest to. You put each toy in the box that it's closest to.

You keep doing this until all the toys are in boxes with other toys that are similar.

To calculate error in k-means, we use a metric called "sum of squared distances."

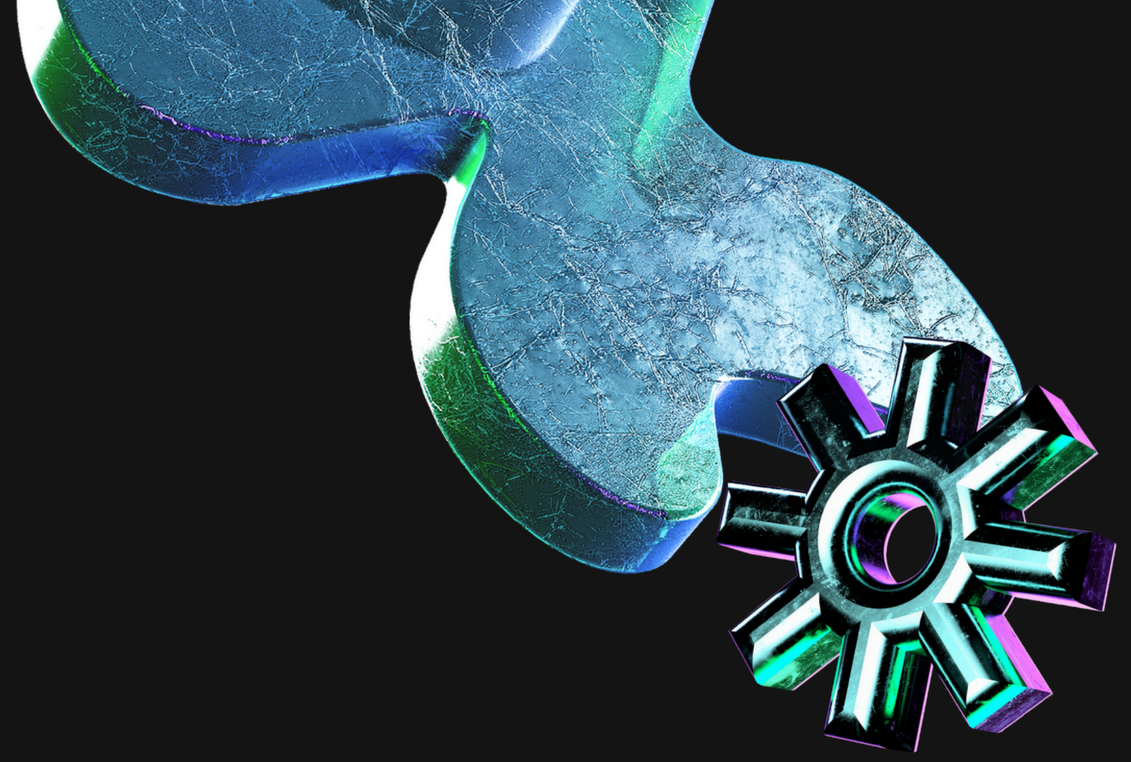
Here's how it works: for each toy in a box, you measure the distance between the toy and the center of the box.

Then you square the distance and add it up for all the toys in the box.

The sum of squared distances tells you how close the toys are to the center of the box.

You want the toys in each box to be as close to each other as possible, so you want the sum of squared distances to be as small as possible.

bhavishya-pandit



Do you have any questions?

Send it over LinkedIn or Twitter!
I hope you learned something new :)

bhavishya-pandit

BhavishyaP9