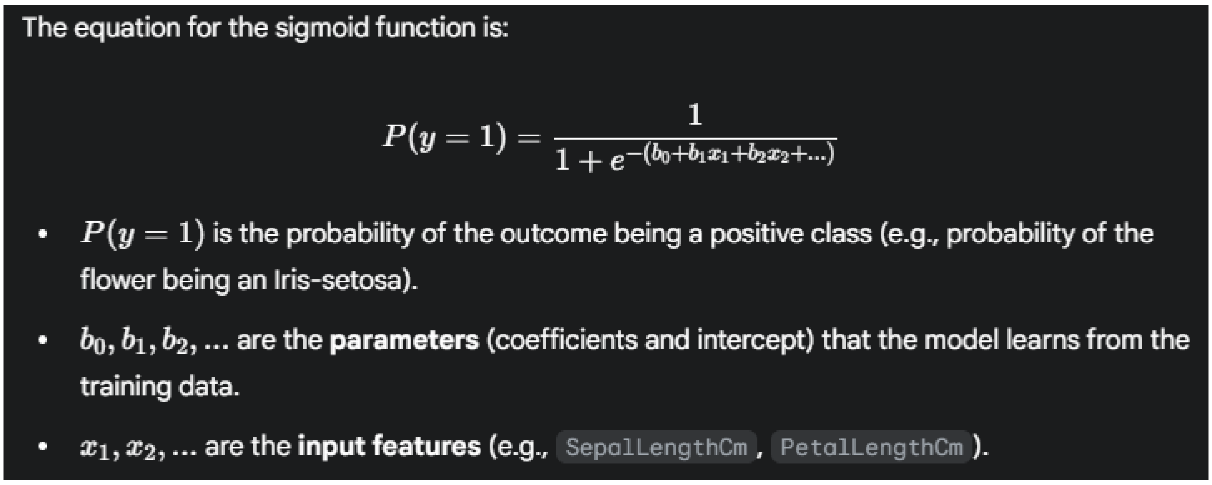
Logistic Regression

In this section, I am going to be talking about Logistic Regression a simple model which is used when you want to predict categorical variables. Like for example, If you want to predict between a cat and a dog.

The example I have used over here is in reference to the code - Iris Dataset where our aim is to predict based on several features what kind of a flower it is.

Logistic Regression uses something which is called as a Sigmoid Function which looks very fancy but is a very Simple Equation.



The equation for the sigmoid function is:

$$P(y=1) = \frac{1}{1+e^{-(b_0+b_1x_1+b_2x_2+\ldots)}}$$

- $P(y=1)$ is the probability of the outcome being a positive class (e.g., probability of the flower being an Iris-setosa).
- $b_0, b_1, b_2, \ldots$ are the **parameters** (coefficients and intercept) that the model learns from the training data.
- $x_1, x_2, \ldots$ are the **input features** (e.g., `SepalLengthCm`, `PetalLengthCm`).

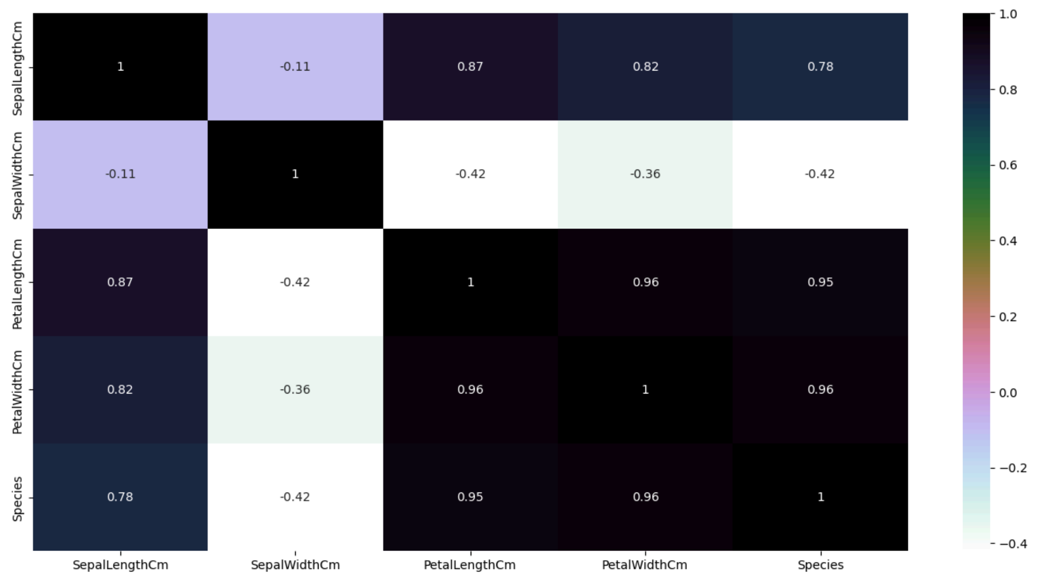
Look up SIgmoid Function to see how it graphs

The model learns the values of the parameters that maximize the probability of the observed outcomes. Once the parameters are learned, you can plug in the features of a new data point and get a probability. If this probability is above a certain threshold (usually 0.5), the model classifies it as the positive class; otherwise, it's the negative class.

What does b0,b1 represent

“These determine the "weight" or importance of each feature in predicting the class. A large positive coefficient for `PetalLengthCm` would mean that longer petals significantly increase the probability of a flower belonging to a specific species.”

So according to the correlation matrix below , since petal Length and Petal Width had higher correlations, Higher weights are given to them and I have also checked by considering 2 main columns and all the columns and higher accuracy was given to the former.



The C Parameter: The Rule-Follower 👮(More Intutive)

Imagine you're a teacher trying to draw a line to separate two groups of students, the "tall" group and the "short" group, on a playground.

- The **students** are your **data points**.
- The **line** you draw is your **decision boundary**.
- Your goal is to draw a line that separates the two groups.

Low C Value: The Strict Teacher 🙅

A model with a **low C value** is like a **strict teacher**. This teacher has a firm rule: "Keep your line simple and straight." This teacher heavily penalizes any line that wiggles or bends just to perfectly separate a few outliers.

- The strict teacher's line might misclassify one or two students, but it's a simple, general rule that works well for the whole class.
- **Result:** A simpler model that generalizes well but might have slightly lower accuracy on the training data. This helps prevent **overfitting**.

High C Value: The Lenient Teacher 🧑‍🏫

A model with a **high C value** is like a **lenient teacher**. This teacher says, "I want my line to be perfect. No student should be misclassified!" They will bend and twist the line to make sure every single student on the playground is on the correct side.

- This lenient teacher's line is very complex and might only work for this specific group of students. If a new student joins, the complex line might not work as well.

- **Result:** A complex model that has high accuracy on the training data but might **overfit** and perform poorly on new, unseen data.

In short, the **C parameter is a control dial for how much the model should prioritize a simple, general line versus a complex, highly accurate line.** Low C favors simplicity, while high C favors accuracy on the training data.

The C Parameter (Regularization) - Lil Mathematical

C is the inverse of the regularization strength. This parameter directly controls the trade-off between fitting the training data well and keeping the model's parameters small to prevent **overfitting**.

- **Small C value** (e.g., 0.01): A small C value means a **strong regularization**. It adds a large penalty for large coefficients. The model is forced to keep the coefficients close to zero, which makes the decision boundary simpler and less sensitive to individual data points or noise. This helps the model generalize better to new, unseen data, which is especially useful when dealing with a small dataset.
- **Large C value** (e.g., 100): A large C value means a **weak regularization**. The model is less penalized for having large coefficients. It will try to fit the training data as closely as possible, which can lead to a complex decision boundary that perfectly classifies the training data but might perform poorly on the test set.

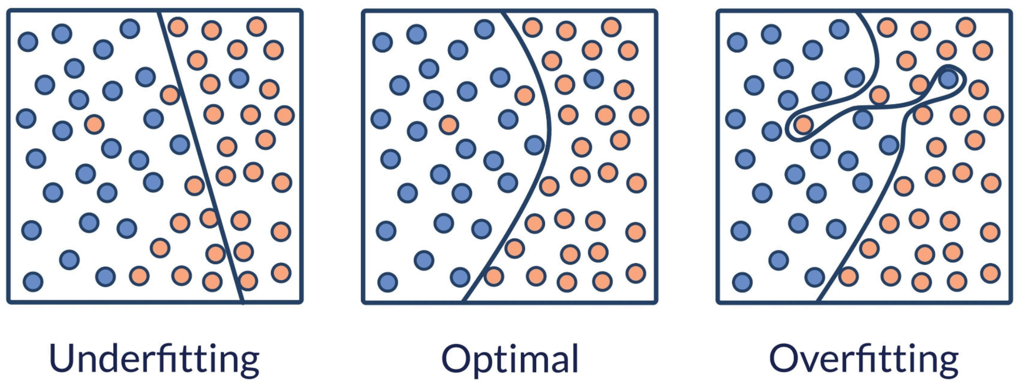
Mathematical Connection: The cost function that Logistic Regression minimizes is:

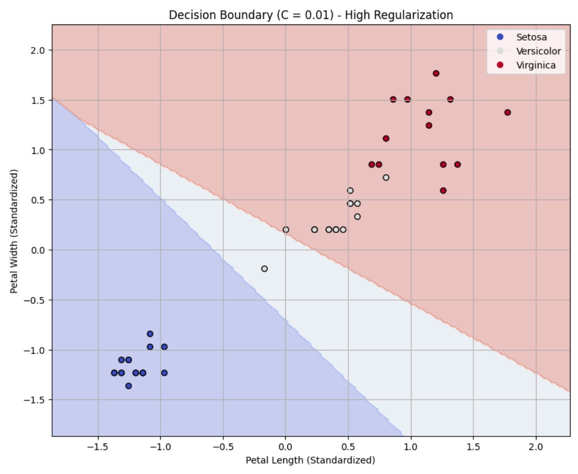
$$J(\theta) = \underbrace{-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]}_{\text{Logistic Loss}} + \underbrace{\frac{1}{2C}||\theta||^2}_{\text{Regularization Term}}$$

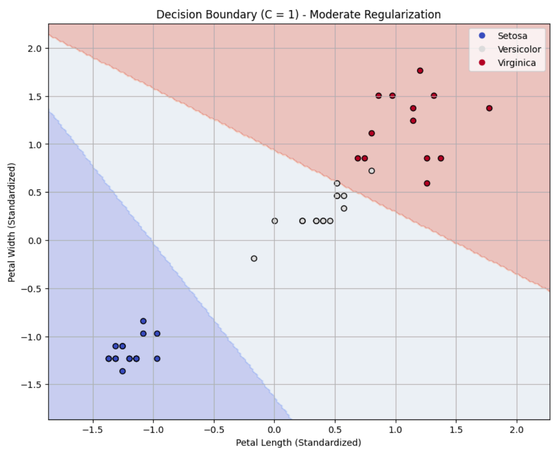
In Our Dataset, We have seen that High C Has caused an Accuracy score of 0.53 and Lower C has caused an Accuracy of 0.97

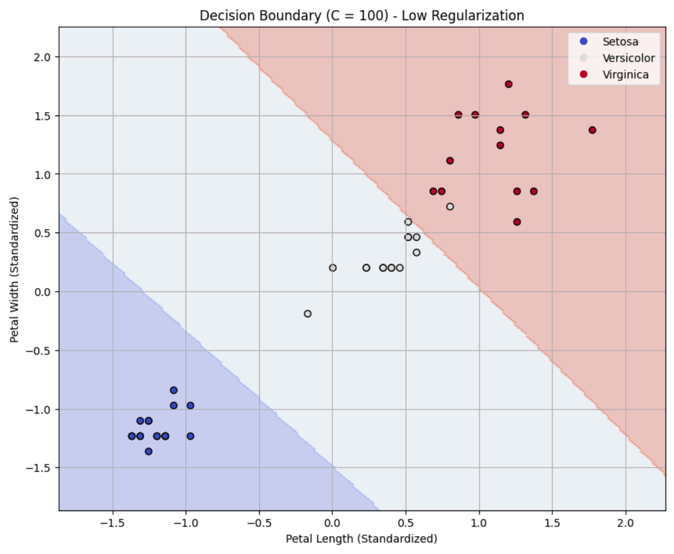
When regularization is too strong (low C), the model's coefficients are pushed towards zero. This can cause the model to underfit the data. Underfitting occurs when the model is too simple to capture the underlying patterns in the data, leading to a poor performance on both the training and testing sets.

A high C allows the model to be more flexible. It can assign larger coefficient values to the features, which enables it to draw a more complex decision boundary that better separates the data points. For the Iris dataset, which is relatively clean and linearly separable (especially Petal Length and Width), this flexibility allows the model to achieve high accuracy. However, in noisier datasets, this same flexibility could lead to overfitting, where the model learns the noise in the training data and performs poorly on new data.


Underfitting
Optimal
Overfitting


Decision Boundary (C = 0.01) - High Regularization
Setosa
Versicolor
Virginica
Petal Width (Standardized)
Petal Length (Standardized)


Decision Boundary (C = 1) - Moderate Regularization
Setosa
Versicolor
Virginica
Petal Width (Standardized)
Petal Length (Standardized)


Decision Boundary (C = 100) - Low Regularization
Setosa
Versicolor
Virginica
Petal Width (Standardized)
Petal Length (Standardized)