Laboratory 2: Exploring Class Boundaries

Authors: Noemi Cicala, Mazzini Matteo

In this lab, we will explore how different models classify data and identify the parameters that directly impact their performance. By modifying these parameters, we can observe how the models adapt, altering their decision boundaries and reshaping class regions. For this purpose, we will work with a toy binary classification dataset to demonstrate these concepts.

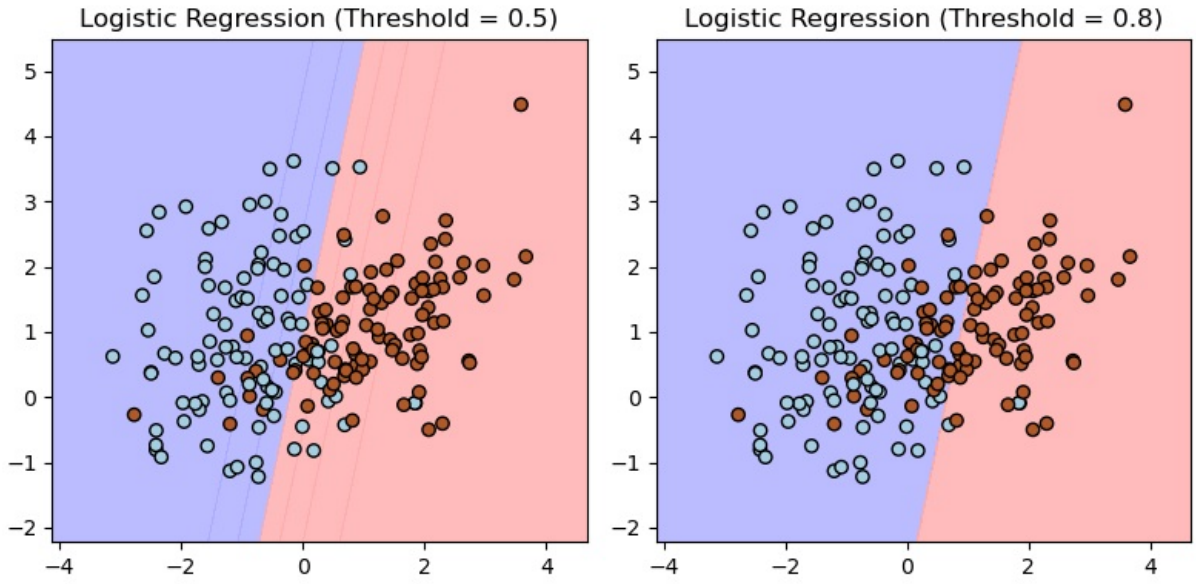
Linear Boundaries

We begin by exploring models that separate data by a straight line (or a hyperplane in higher dimensions), resulting in a linear decision boundary. These models are particularly effective when the relationship between the input features and class labels is inherently linear. The models we focus on are:

- Logistic Regression
- Linear SVM
- LDA

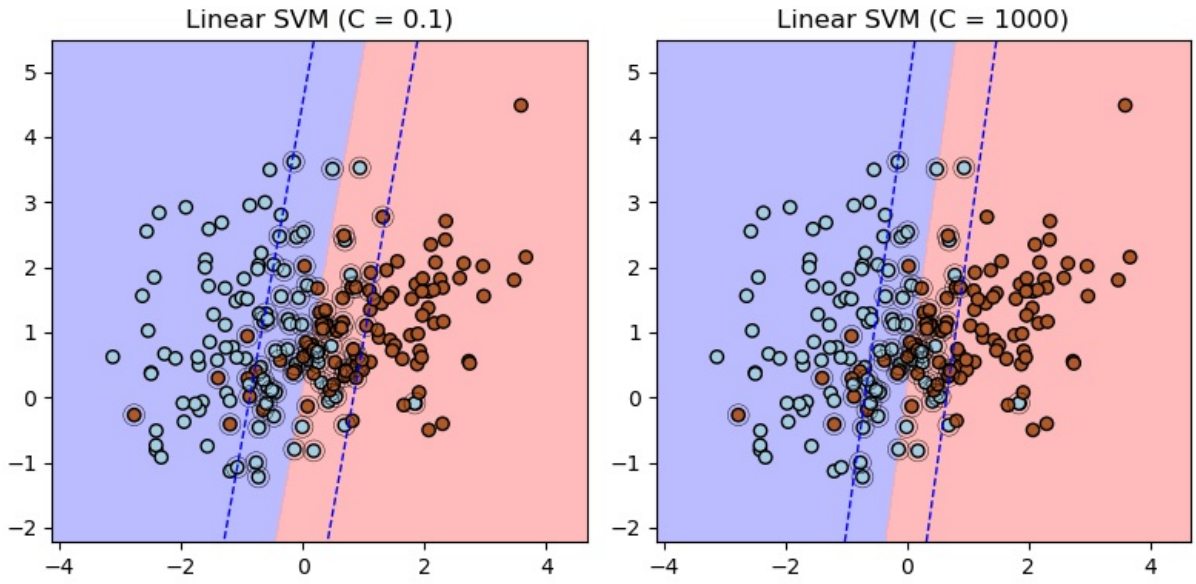
Logistic Regresion

This statistical model predicts the probability of a binary outcome based on a linear combination of input features, identifying a linear boundary by maximizing classification likelihood. Class output is determined by the probability threshold, which defaults to 0.5. Increasing the threshold shifts the boundary toward the positive class, shrinking its region, while decreasing it expands the positive class region



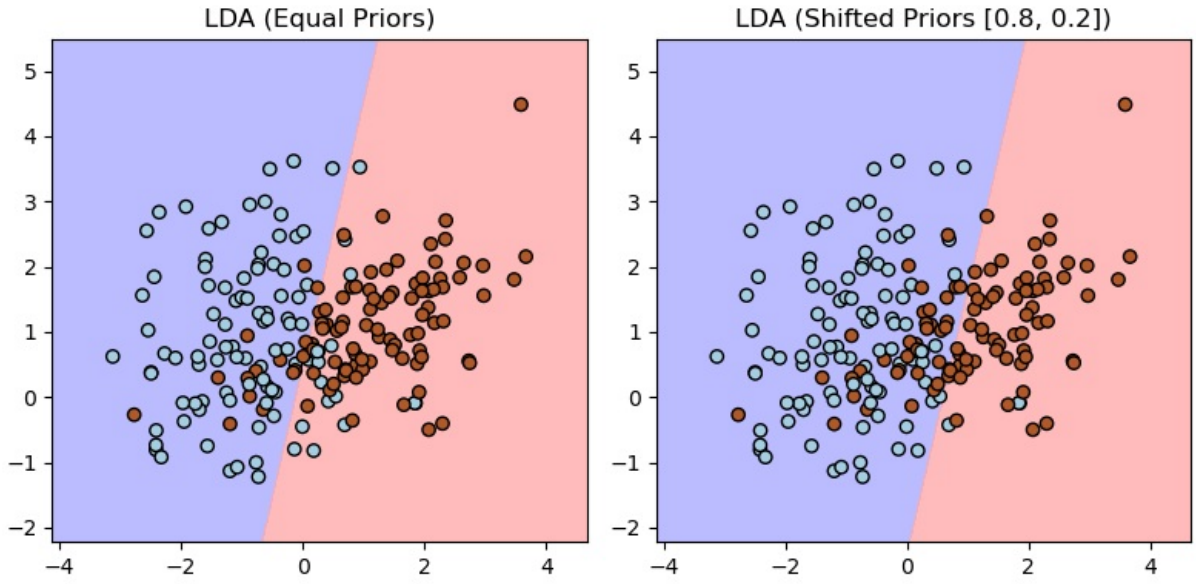
Linear SVM

This model identifies a linear boundary that maximizes the margin between classes, focusing on support vectors. The decision boundary is influenced by the Cparameter, which balances margin width and classification errors. A high C creates a boundary tightly fitted to the data, potentially overfitting, while a low C widens the margin, allowing more misclassifications and improving generalization.



LDA

This model identifies a linear boundary by modeling class distributions and assuming shared covariance. The decision boundary is influenced by class priors, which weight the importance of each class. Higher priors for one class shift the boundary toward the less probable class, expanding the dominant class region, while equal priors create a balanced separation.



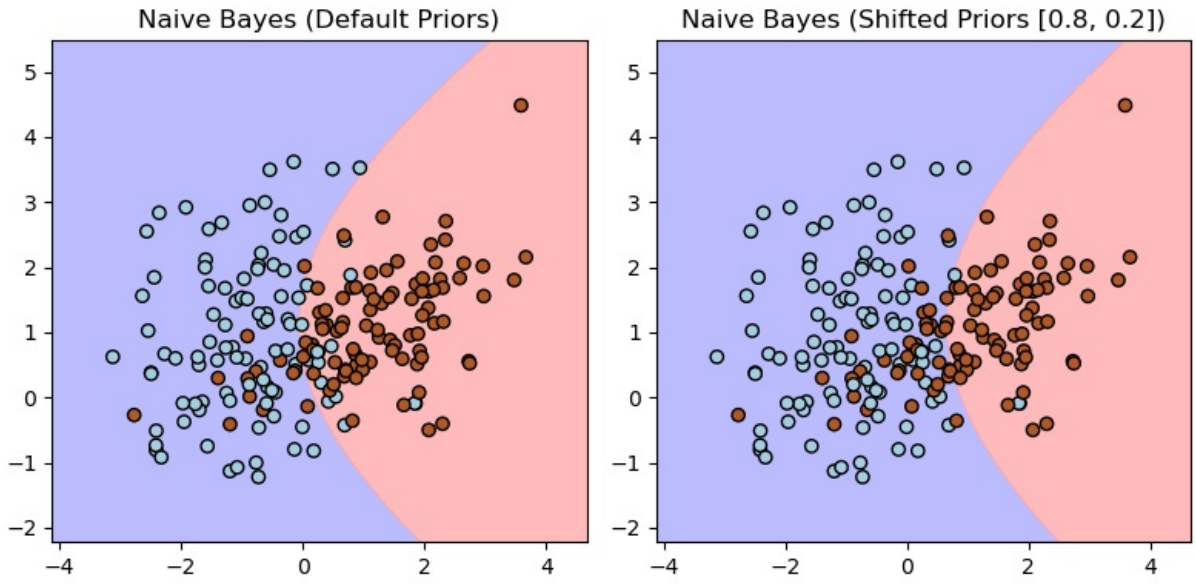
Non Linear Boundaries

We now focus on models capable of creating non-linear boundaries between classes. These models are often more effective than linear models when the relationship between variables is non-linear. However, their increased complexity can lead to overfitting if not carefully managed. The models include:

- Naive Bayes
- Quadratic Discriminant Analysis (QDA)
- SVM with RBF and Polynomial kernels
- K-Nearest Neighbors (KNN)
- Decision Trees (CART)
- Multilayer Perceptron (MLP)

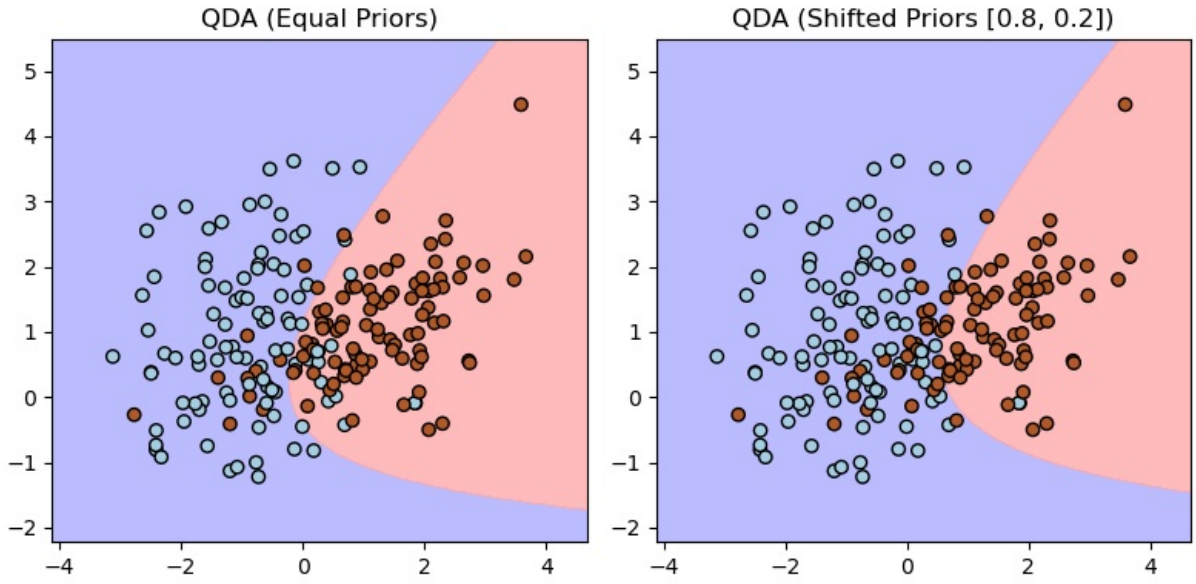
Naive Bayes

This model predicts class membership by modeling feature distributions independently for each class. The decision boundary is influenced by class priors, which adjust the weighting of each class. Higher priors for one class shift the boundary toward the less likely class, enlarging the dominant class region, while equal priors lead to a balanced separation.



QDA

This model identifies a quadratic boundary by modeling class distributions separately and allowing distinct covariances for each class. The decision boundary is influenced by class priors, which weight the importance of each class. Higher priors for one class shift the boundary toward the less probable class, expanding the dominant class region, while equal priors create a balanced separation.



Kernel SVMs

This model identifies a non-linear boundary by transforming data into higher-dimensional spaces using kernel functions, such as polynomial or RBF. In the transformed space, the model applies a linear classifier, which allows it to effectively separate classes that are not linearly separable in the original feature space.

- Polynomial Kernel: The boundary takes a global curved shape, whose complexity depends on the kernel degree. Higher degrees create more intricate, wavy boundaries, while lower degrees yield smoother ones. However, polynomial kernels are restricted to boundaries that can be represented by polynomial functions.
- RBF Kernel: The boundary forms localized, smooth, circular, or elliptical regions around data points, adapting to clusters in the input space. Unlike polynomial kernels, RBF kernels can model arbitrarily complex decision boundaries, making them more versatile for capturing intricate patterns.

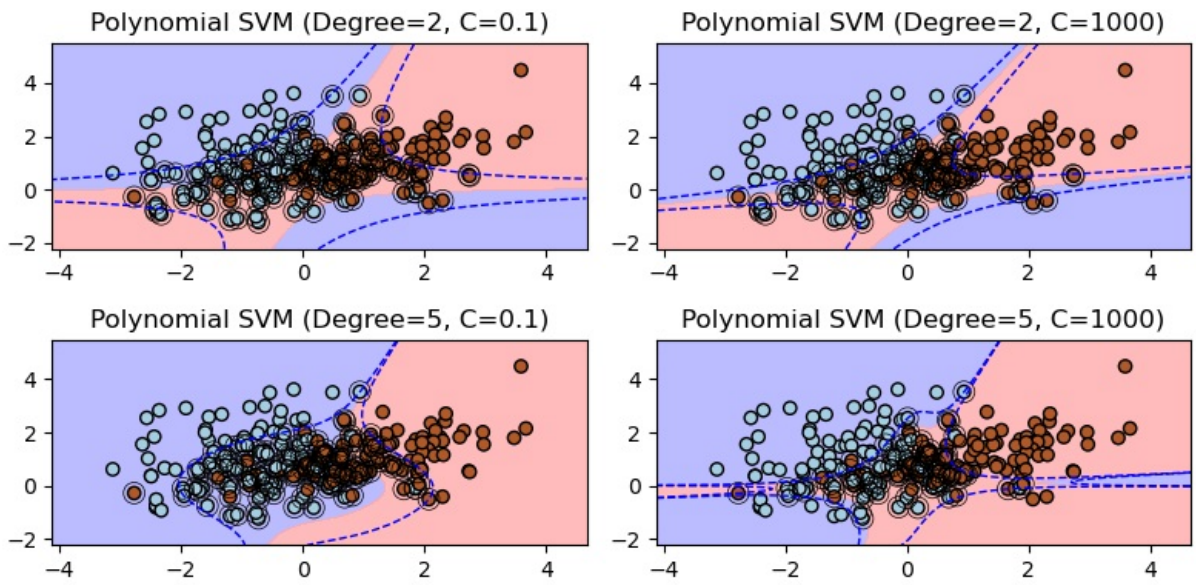
Parameters Influencing Decision Boundaries: The complexity of the decision boundary is influenced by two key parameters:

- *gamma* for RBF kernels: Controls the influence of each data point. Higher gamma values result in tighter, highly localized boundaries, increasing the risk of overfitting. Lower gamma values create broader, smoother boundaries that generalize better.
- Degree (d) for polynomial kernels: Determines the polynomial order. Higher degrees allow more complex, wavy boundaries, while lower degrees result in simpler, smoother shapes.

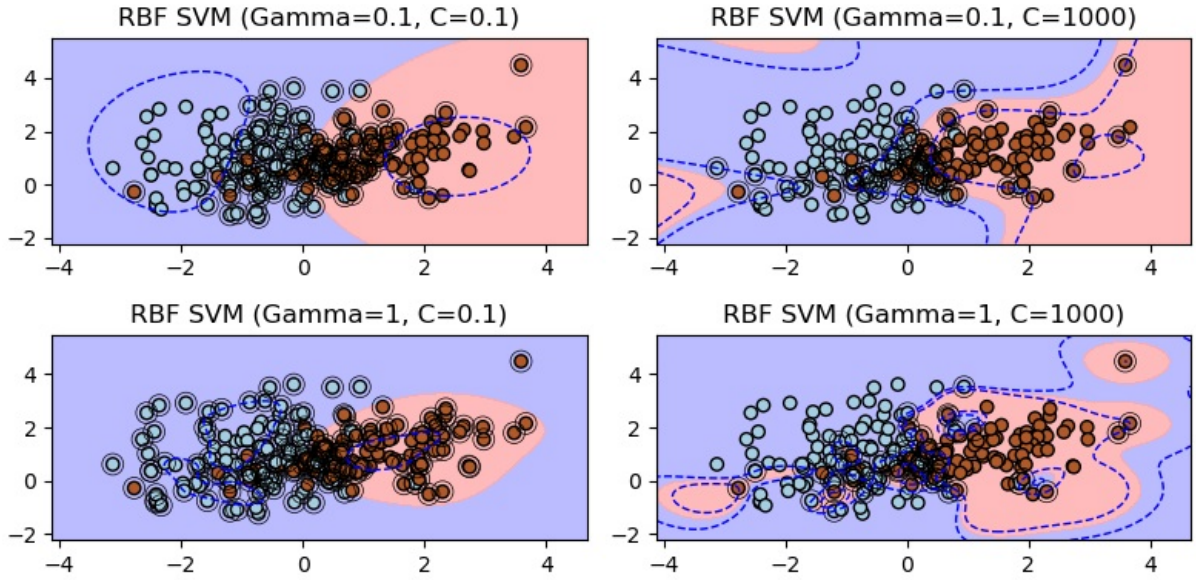
Additionally, as in the linear svm the C parameter affects the decision boundary for both kernels by controlling the trade-off between margin size and misclassification.

- High C: Produces a tighter boundary that focuses on minimizing classification errors, but it risks overfitting the data.
- Low C: Creates a wider margin, tolerating more misclassifications to improve generalization.

Polynomial SVM



RBF-SVM



K-NN

This model identifies non-linear decision boundaries based on the distribution of the training data and classifies points using the k-nearest neighbors. The decision boundary is shaped by several factors, including the value of k, the distance metric, and the weighting of neighbors.

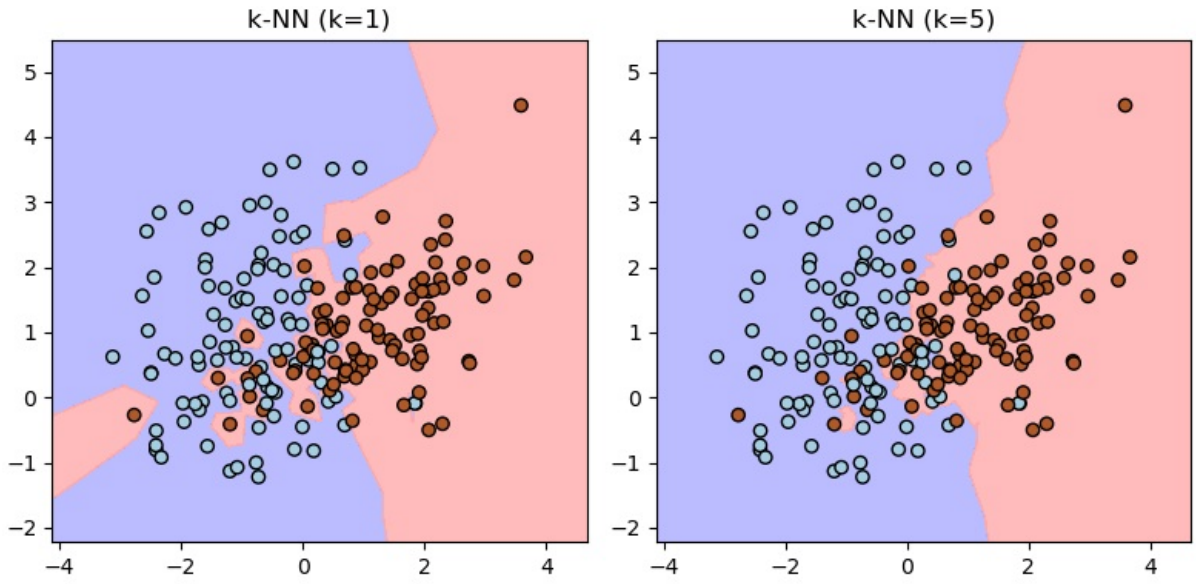
Effect of k (Number of Neighbors):

- A smaller k creates highly irregular, jagged boundaries that closely follow the training data, making the model sensitive to noise and prone to overfitting. k =1 classifies al the points as in the training dataset, thereofre no training error and overfitted model.
- A larger k results in smoother, more generalized boundaries, as decisions are based on a larger set of neighbors, reducing sensitivity to noise but potentially underfitting.

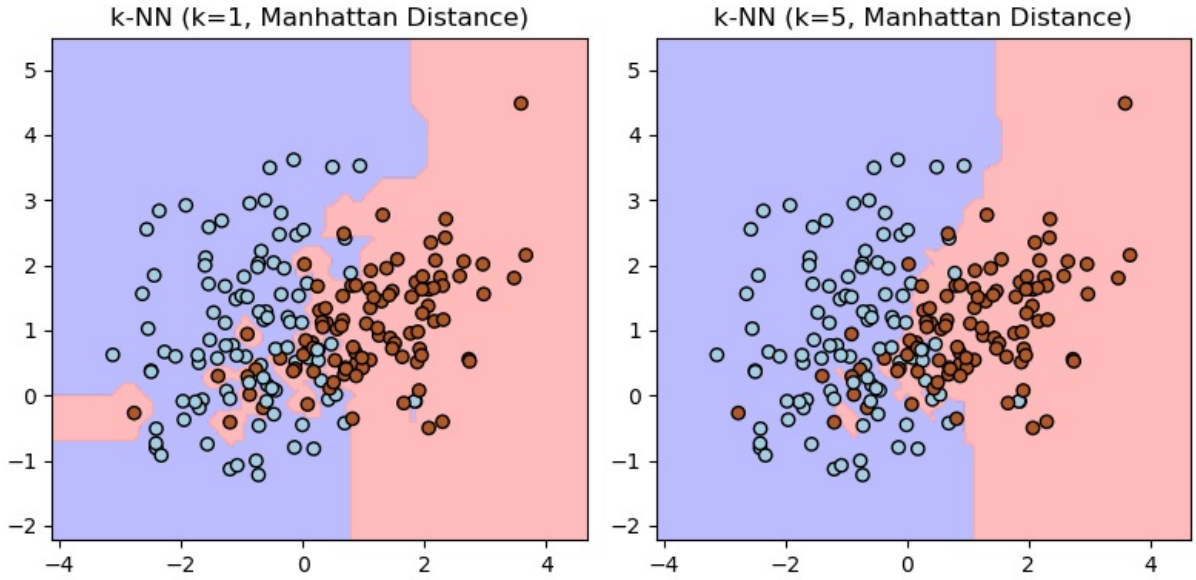
The choice of distance metric determines how neighbors are defined and influences the shape of decision regions:

- Euclidean Distance: Forms circular or spherical regions around data points.
- Manhattan Distance: Forms more rectangular and less smoother regions.
- Other Metrics: Metrics like Minkowski or cosine similarity result in other varied boundary shapes.

k = 1 and k = 5 with Euclidean Distance



k = 1 and k = 5 with Manhattan Distance



CART Decision Trees

This model identifies axis-aligned, rectangular boundaries by recursively splitting the feature space into regions based on feature thresholds. The decision boundary is influenced by the depth of the tree and the impurity measure used for splitting.

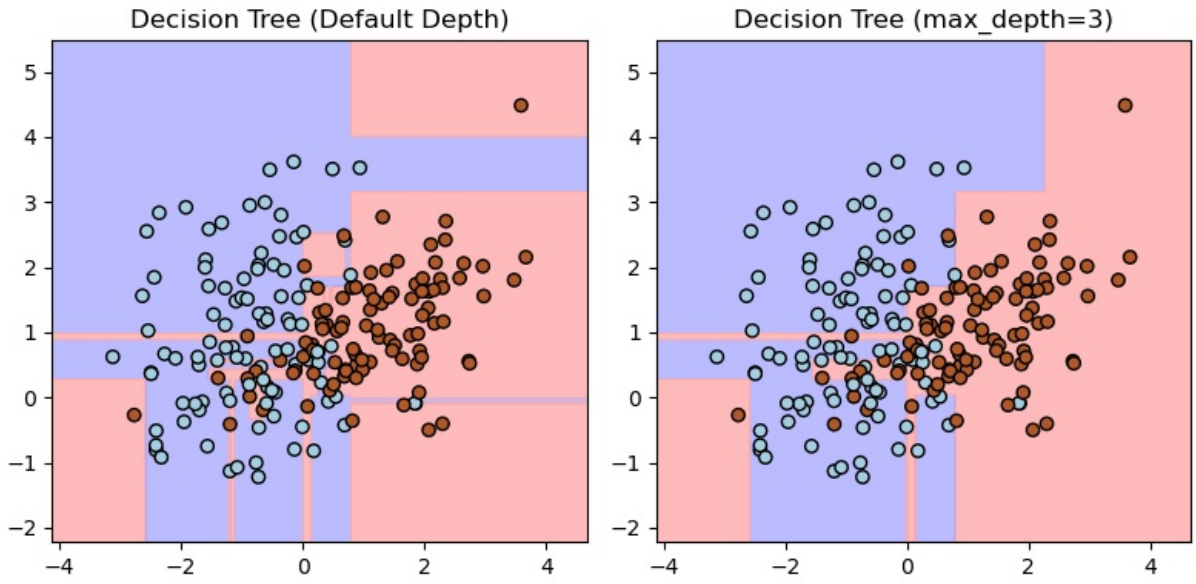
Tree Depth:

- A shallow tree results in fewer splits, creating coarse, simple boundaries that may underfit the data.
- A deeper tree introduces more splits, creating finer, more detailed regions. However, overly deep trees can overfit the training data, leading to highly fragmented boundaries.

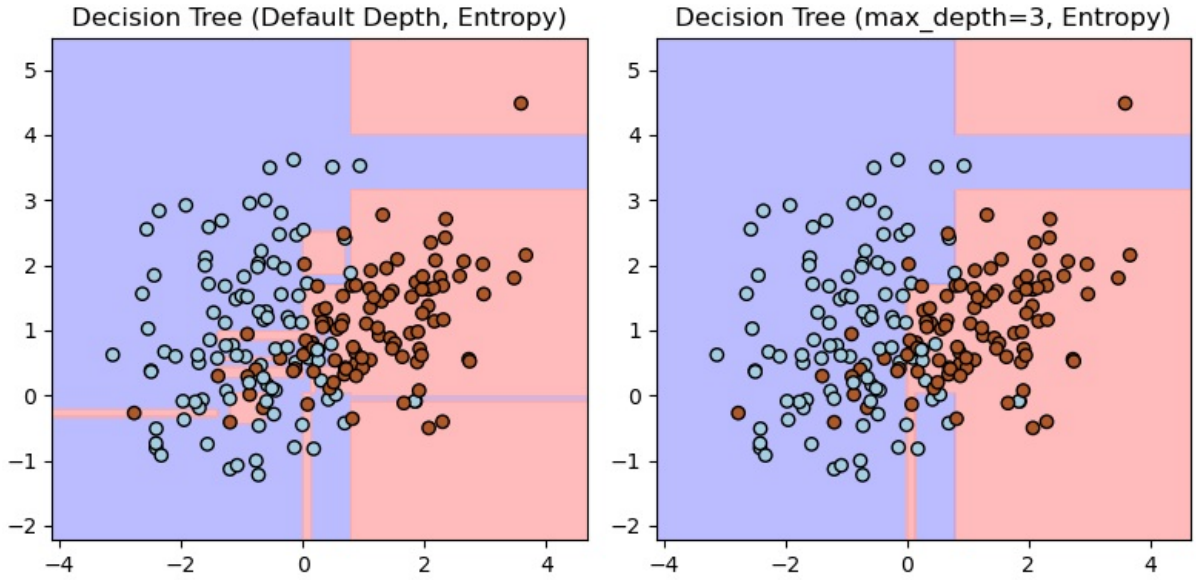
Impurity Measures:

- Gini Index prioritizes splits that maximize class separation with a focus on majority class dominance.
- Entropy prioritizes splits that maximize information gain, often creating more balanced splits.

Default and reduced depth with Gini impurity



Default and reduced depth with entropy impurity



MLP with one hidden layer

This model's decision boundaries are shaped by the number of neurons in the hidden layer and the activation function, as these determine how the feature space is partitioned into regions. According to the Universal Approximation Theorem, an MLP with one hidden layer and a non-linear activation function can approximate any continuous function, allowing it to represent arbitrarily complex boundaries given enough neurons.

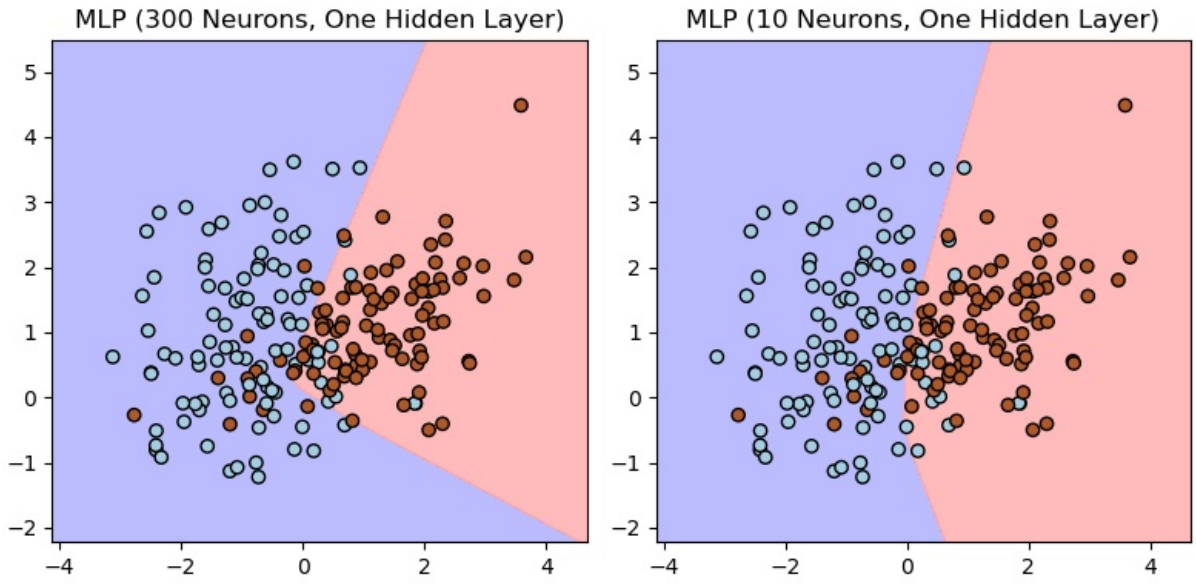
Number of Neurons:

- Fewer neurons: Result in simple, coarse boundaries that may underfit, as the model lacks sufficient capacity to capture complex patterns.
- More neurons: Allow the model to create complex, fine-grained boundaries, potentially matching highly intricate patterns. However, excessive neurons can lead to overfitting, where the model learns noise in the training data.

Activation Functions:

- ReLU: Produces piecewise linear boundaries defined by combinations of linear hyperplanes, leading to sharp, angular decision regions.
- Tanh: Generates smooth, non-linear boundaries with gradual transitions, making it better suited for capturing complex, smooth patterns.

MLP with hidden 300 and 10 hidden neurons and Relu activation function



MLP with hidden 300 and 10 hidden neurons and tanh activation function