CAP 781
MACHINE LEARNING

Tanzeela Javid Kaloo (32638)
Assistant Professor
System And Architecture
Lovely Professional University

UNIT – II
Supervised Learning

Tanzeela Javid Kaloo (32638)
Assistant Professor
System And Architecture
Lovely Professional University

Content

- Regression,
- Linear Regression,
- Polynomial Regression,
- Classification,
- Logistic Regression,
- k-Nearest Neighbors (k-NN),
- Support Vector Machines (SVM),
- Decision Trees and Random Forests,
- Ensemble Methods,
- Bagging, Boosting,
- Model Evaluation Techniques,
- Cross Validation,
- Hyperparameter Tuning,
- Introduction to Scikit-learn,
- Hands-on with Real-world Datasets

Evaluation Metrics for Linear Regression

A variety of evaluation measures can be used to determine the strength of any linear regression model. These assessment metrics often give an indication of how well the model is producing the observed outputs.

The most common measurements are:

Mean Square Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$$

Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \widehat{Y_i}|$$

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{RSS}{n}} = \sqrt{\frac{\sum_{i=2}^{n}(y_i^{actual} - y_i^{predicted})^2}{n}}$$

(R-squared) $R^2 = 1 - \left(\frac{RSS}{TSS}\right)$

$$RSS = \sum_{i=2}^{n}(y_i - b_0 - b_1 x_i)^2$$

$$TSS = \sum(y - \overline{y_i})^2$$

Adjusted R-Squared Error $Adjusted\, R^2 = 1 - \left(\frac{(1-R^2).(n-1)}{n-k-1}\right)$

Model Evaluation Techniques - Classification

- Accuracy
- Precision
- Recall
- F1 – Score
- Sensitivity
- Specificity
- ROC (Receiver Operating Curve)
- AUC (Area Under the Curve)

Confusion Matrix

- **Confusion Matrix**: A matrix used to evaluate the performance of a classification model, showing the actual vs. predicted outcomes. For binary classification, it forms a **2x2 matrix**.
- **Example Confusion Matrix (for 165 samples)**:
  - **Predicted NO**:
    - Actual NO: **50**
    - Actual YES: **5**
  - **Predicted YES**:
    - Actual NO: **10**
    - Actual YES: **100**

- Key Terms:
  - **True Positives (TP)**: Predicted YES and actual is YES (100).
  - **True Negatives (TN)**: Predicted NO and actual is NO (50).
  - **False Positives (FP)**: Predicted YES but actual is NO (10).
  - **False Negatives (FN)**: Predicted NO but actual is YES (5).
- Accuracy Formula:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Samples}}$$

- Calculation:

$$\text{Accuracy} = \frac{100 + 50}{165} = 0.91$$

  - The model has an **accuracy of 91%**.

The confusion matrix helps visualize the performance in terms of correct predictions (on the diagonal) and errors (off the diagonal).

Classification Accuracy

- **Classification Accuracy**: Measures the ratio of correct predictions to the total number of input samples.
- **Formula**:

$$\text{Accuracy} = \frac{\text{No. of correct predictions}}{\text{Total number of input samples}}$$

- **Ideal Scenario**: Works well when classes are balanced, i.e., when there are equal numbers of samples for each class.
- **Class Imbalance Issue**: If class distribution is skewed (e.g., 90% class A, 10% class B), the model can achieve high accuracy by simply predicting the majority class, even though it might fail at correctly classifying the minority class.
- **Example**: A model trained with 90% of class A and 10% of class B will likely achieve 90% accuracy by predicting all samples as class A. However, if tested on a set with 60% class A and 40% class B, the accuracy drops to 60%.
- **Limitations**: Can create a misleading impression of good performance when the model fails to handle the minority class, leading to potential **False Positives**.

Logarithmic Loss

- **Log Loss (Logarithmic Loss)**: A performance metric that penalizes incorrect classifications, particularly **False Positives**, and is commonly used in multi-class classification.
- **Purpose**: The classifier assigns a probability to each class for all samples, and log loss measures the divergence between the predicted probabilities and the actual class labels.
- **Formula**:

$$\text{Logarithmic Loss} = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

- N: Number of samples
- M: Number of classes
- y_{ij}: Whether sample i belongs to class j (1 if true, 0 if false)
- p_{ij}: Predicted probability that sample i belongs to class j

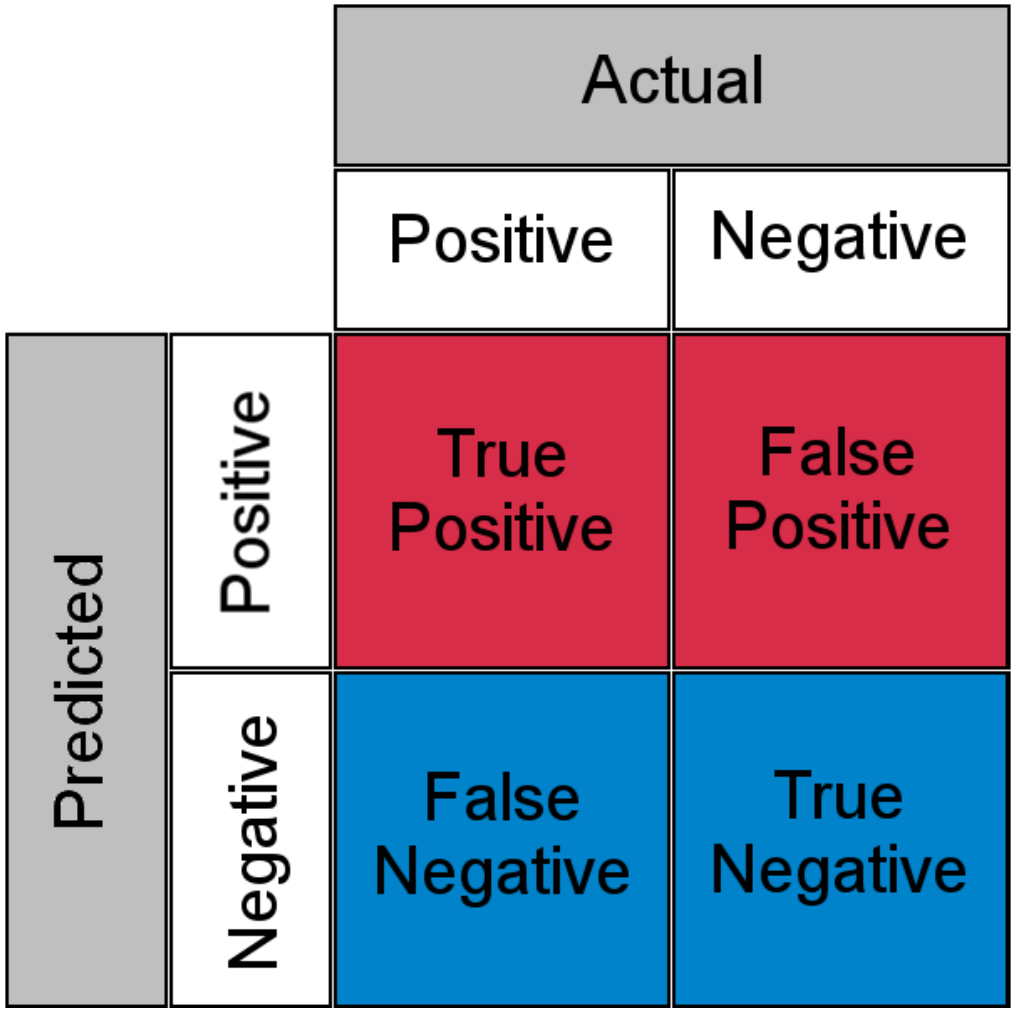
- **Range**:
  - The value of log loss lies between **0** and ∞.
  - **Closer to 0**: Indicates high accuracy.
  - **Farther from 0**: Indicates lower accuracy.
- **Key Insight**: Minimizing log loss improves classifier accuracy since it focuses on correctly assigning probabilities to classes.

Area Under Curve(AUC)

- **Area Under Curve (AUC)**: A widely used metric for binary classification. It represents the probability that a classifier will rank a randomly chosen positive example higher than a randomly chosen negative example.
- **True Positive Rate (TPR)**, also known as **Sensitivity**: The proportion of positive data points correctly classified as positive.
  - Formula:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

    - **TP**: True Positives
    - **FN**: False Negatives



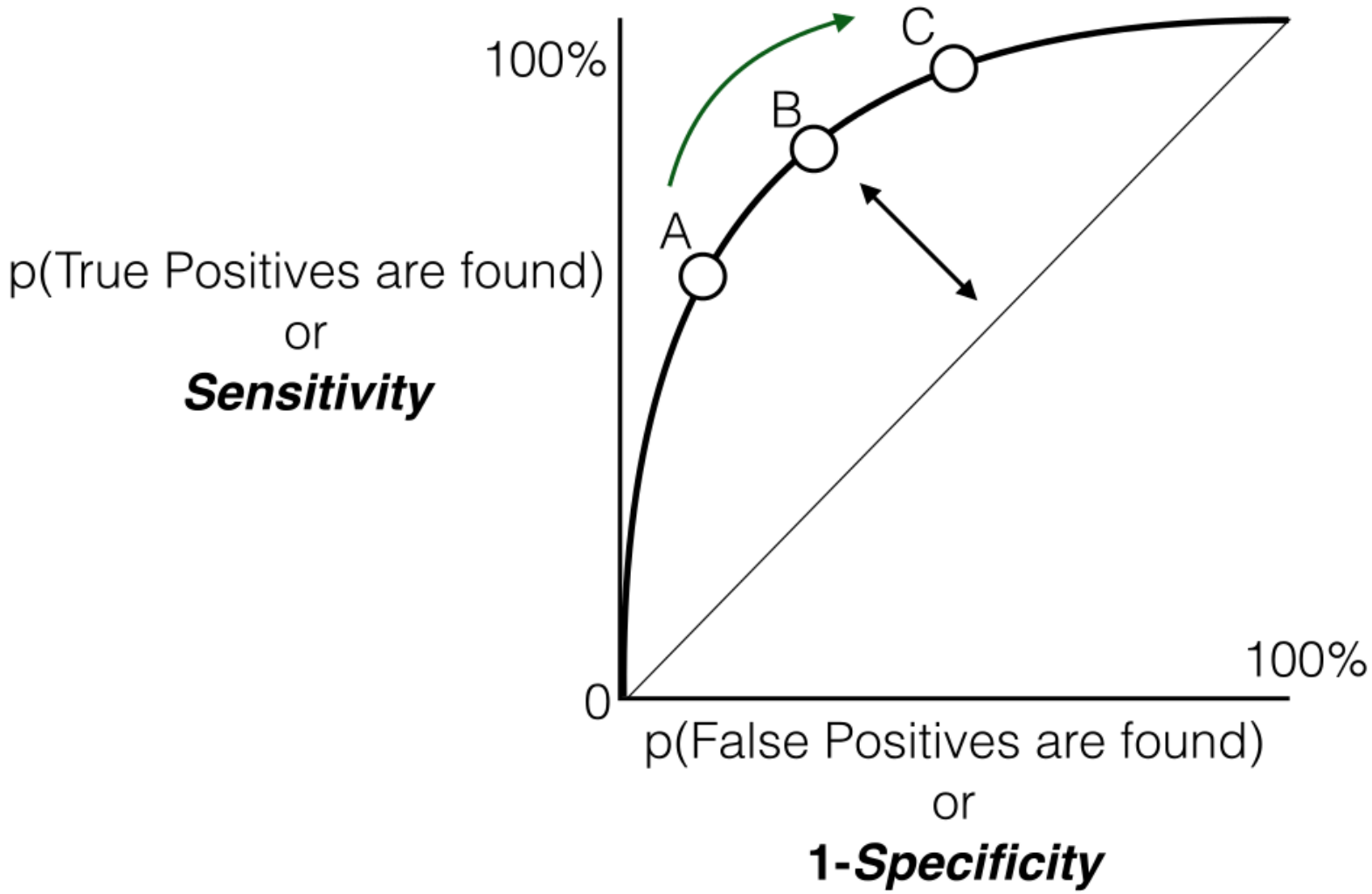
- **True Positive**: Actual Positive and Predicted as Positive
- **True Negative**: Actual Negative and Predicted as Negative
- **False Positive(Type I Error)**: Actual Negative but predicted as Positive
- **False Negative(Type II Error)**: Actual Positive but predicted as Negative

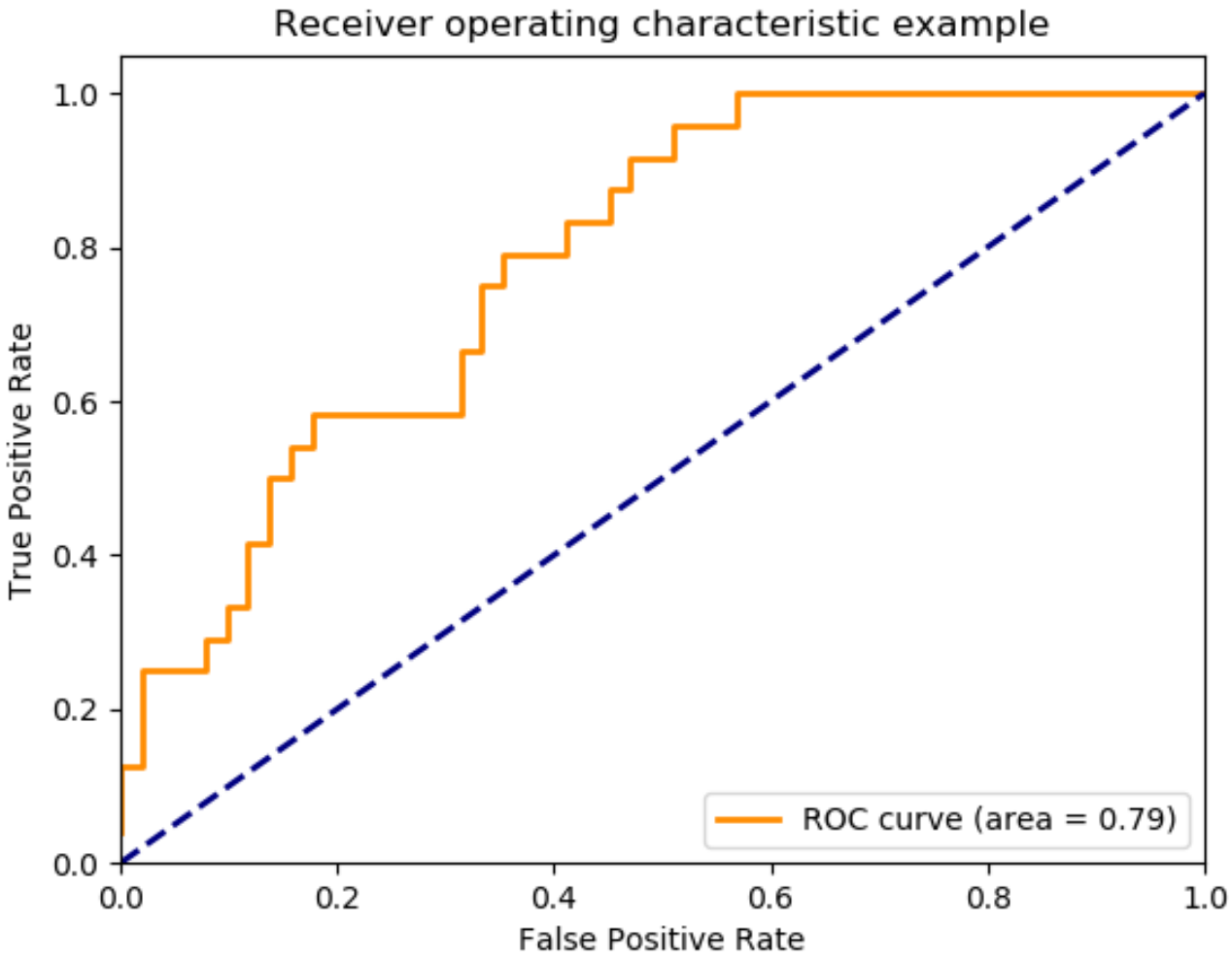
- **True Negative Rate (TNR)**, also known as **Specificity**: The proportion of negative data points correctly classified as negative.
  - Formula:

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \qquad \text{Specificity} = \frac{TN}{TN + FP} = 1 - \text{FPR}$$

    - **TN**: True Negatives
    - **FP**: False Positives
- **False Positive Rate (FPR)**: The proportion of actual negatives that are incorrectly identified as positives.
  - Formula:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$


100%
C
B
A
p(True Positives are found)
or
Sensitivity
0
100%
p(False Positives are found)
or
1-Specificity


Receiver operating characteristic example
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.79)

- **Range of TPR and FPR**: Both values lie between **0** and **1**.
- **AUC (Area Under the Curve)**: A curve plotted between the False Positive Rate (x-axis) and the True Positive Rate (y-axis) for all different data points, with values ranging from **0 to 1**.
    - **Greater AUC**: Indicates better model performance. An AUC close to **1** signifies that the model is good at distinguishing between positive and negative classes.

F1 Score

- **F1 Score**: The harmonic mean of **precision** and **recall**. It ranges from **0 to 1** and provides a balance between precision and recall, particularly useful when you want to find a model that does well on both metrics.
- **Precision**: Measures how many of the positive predictions made by the model are actually correct.
  - Formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

    - **TP**: True Positives
    - **FP**: False Positives

- **Recall**: Measures how many of the actual positive instances were correctly identified by the model.
  - Formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

    - **FN**: False Negatives
- **Precision vs Recall**:
  - **Lower recall, higher precision**: Results in high accuracy but misses many instances (e.g., may fail to detect a large number of true positives).
  - **Higher recall**: Captures more instances, but precision might decrease.

- F1 Score Formula:

$$F1 = 2 \times \frac{1}{\left(\frac{1}{\text{precision}} + \frac{1}{\text{recall}}\right)}$$

  - The **higher** the F1 score, the better the classifier's performance in balancing precision and recall.

1.4. Support Vector Machines

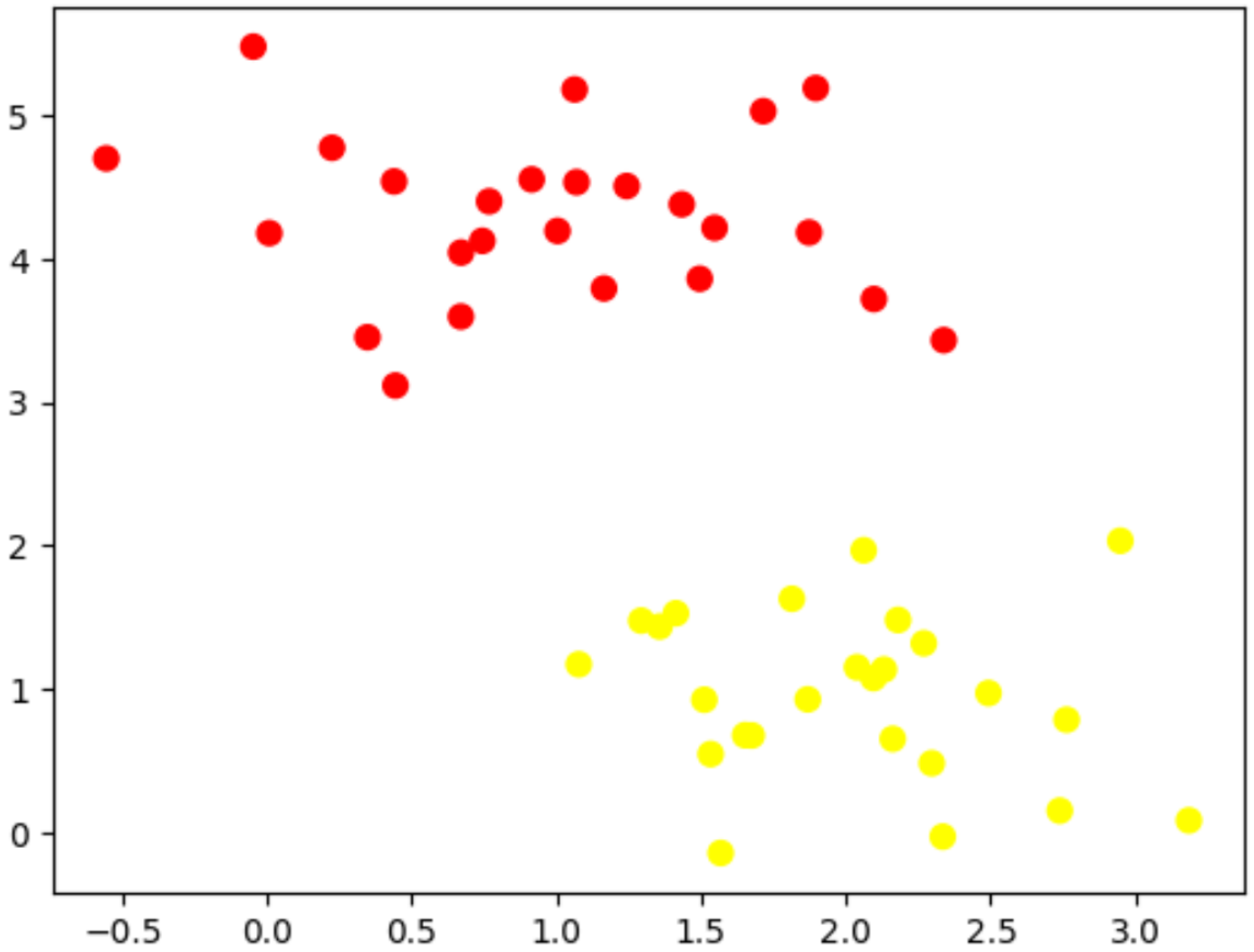
Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

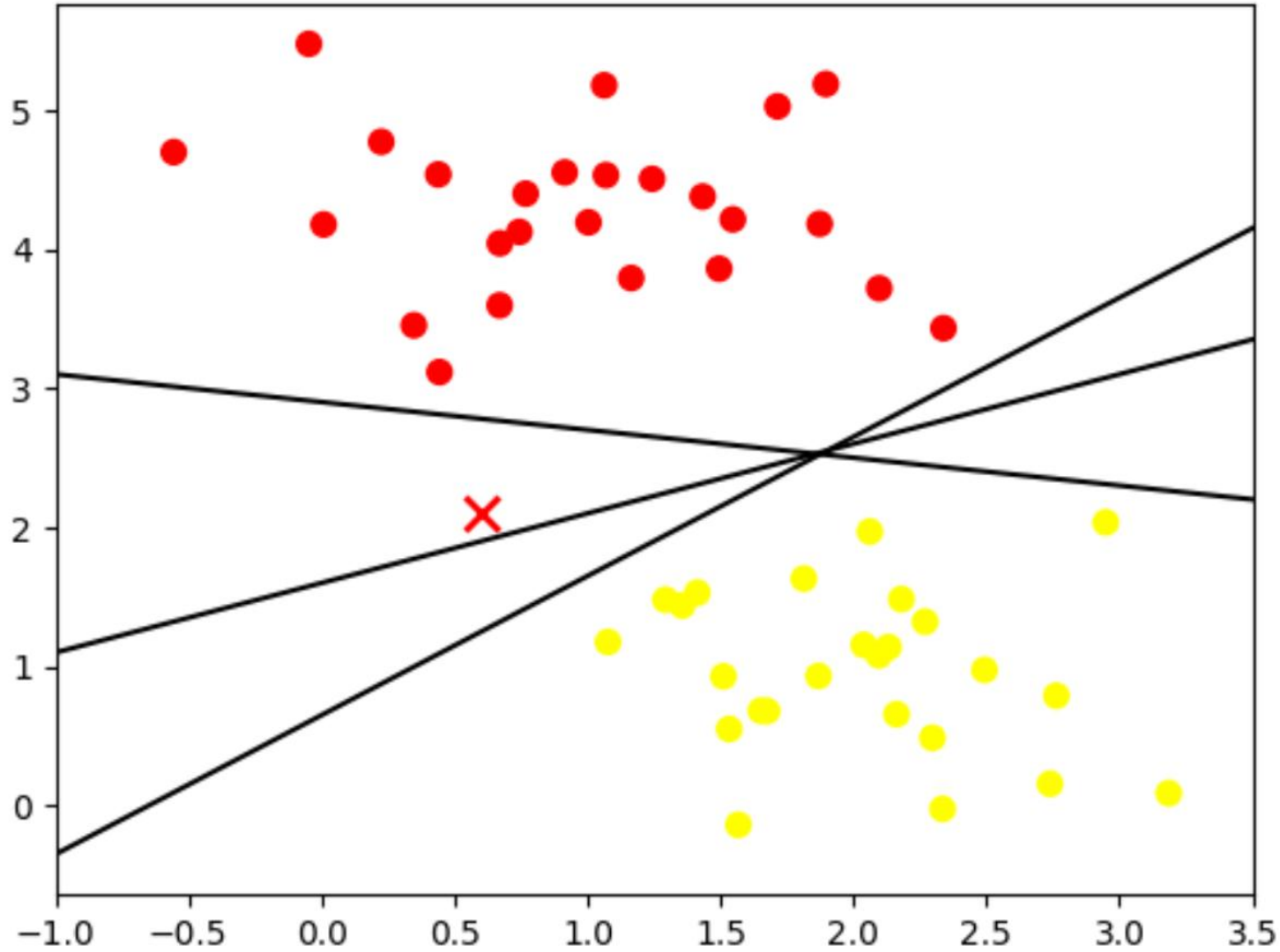
The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

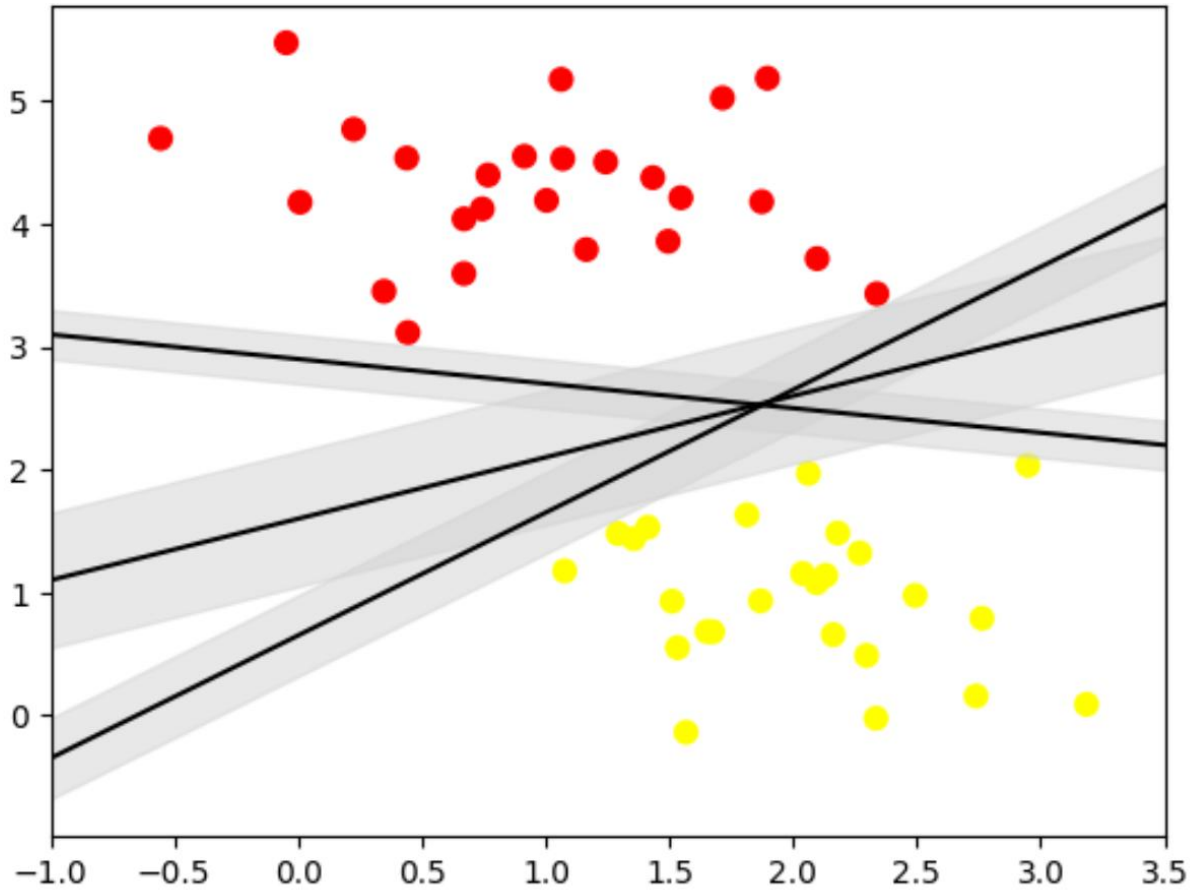
- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.


5
4
3
2
1
0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0

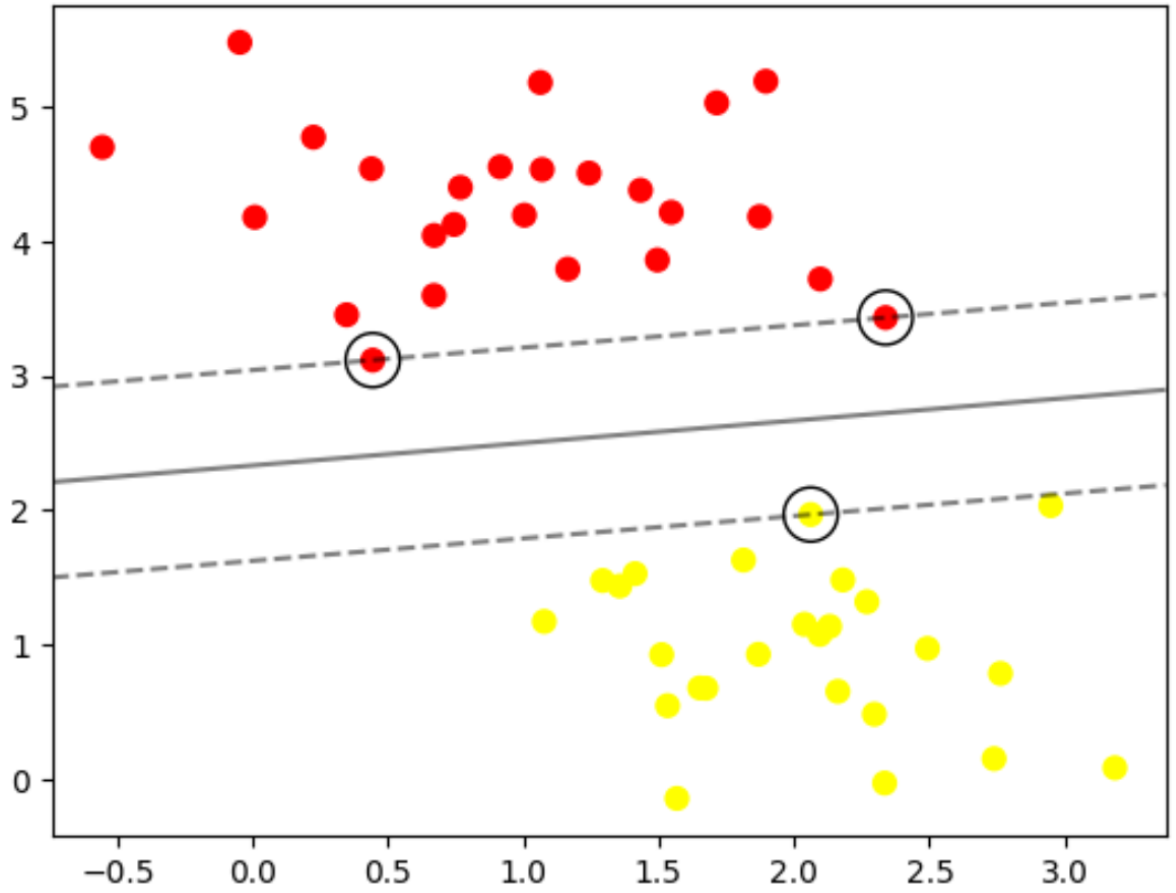

5
4
3
2
1
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

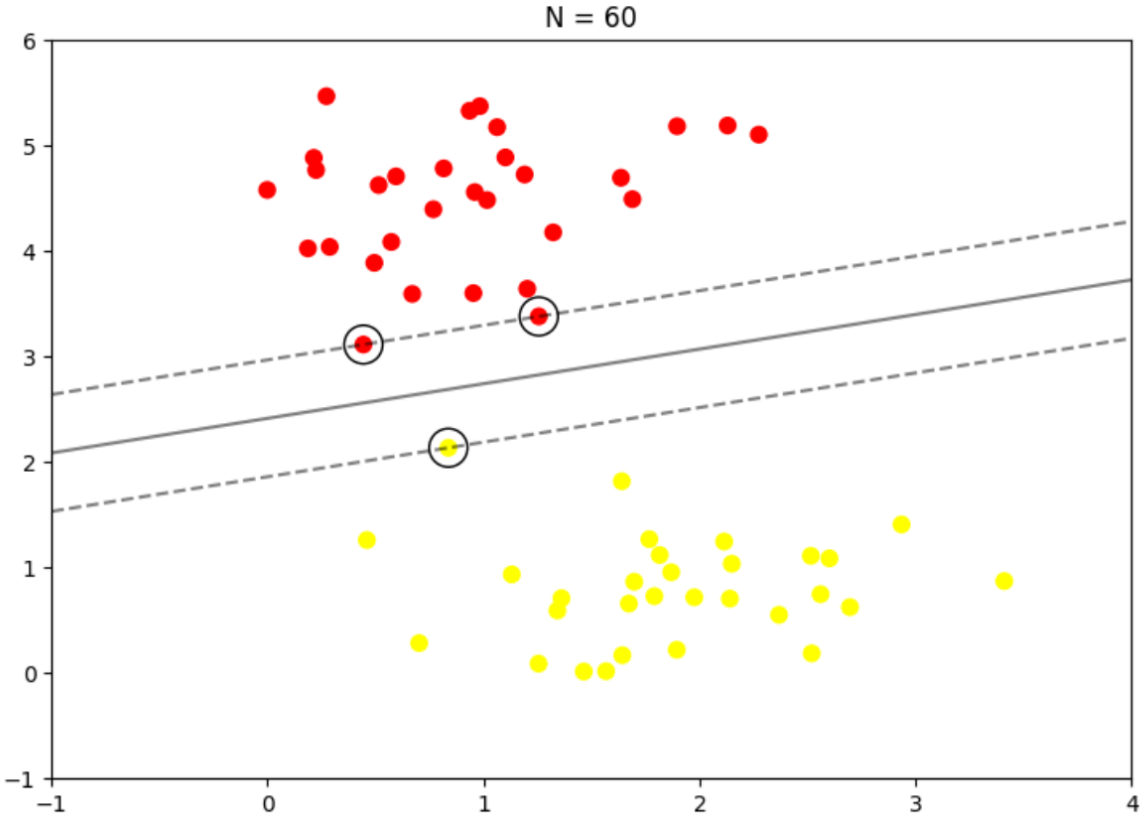
Support Vector Machines: Maximizing the Margin

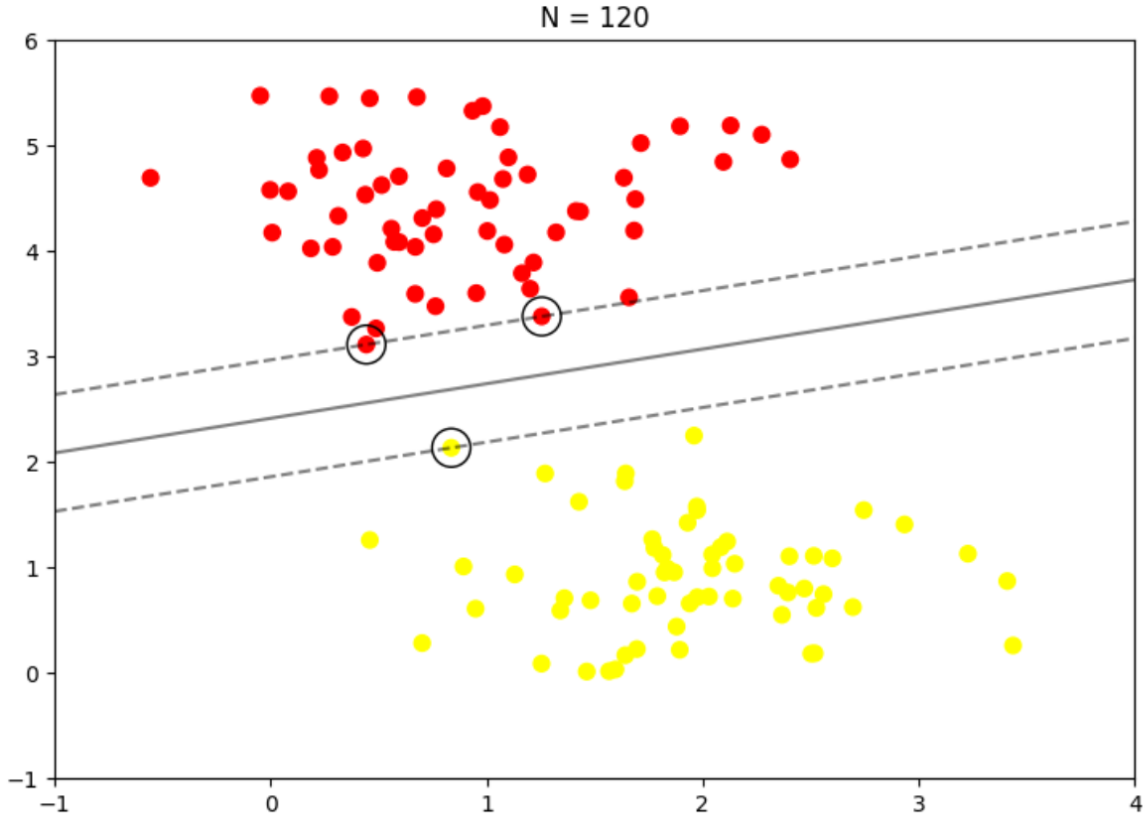
The line that maximizes this margin is the one we will choose as the optimal model.

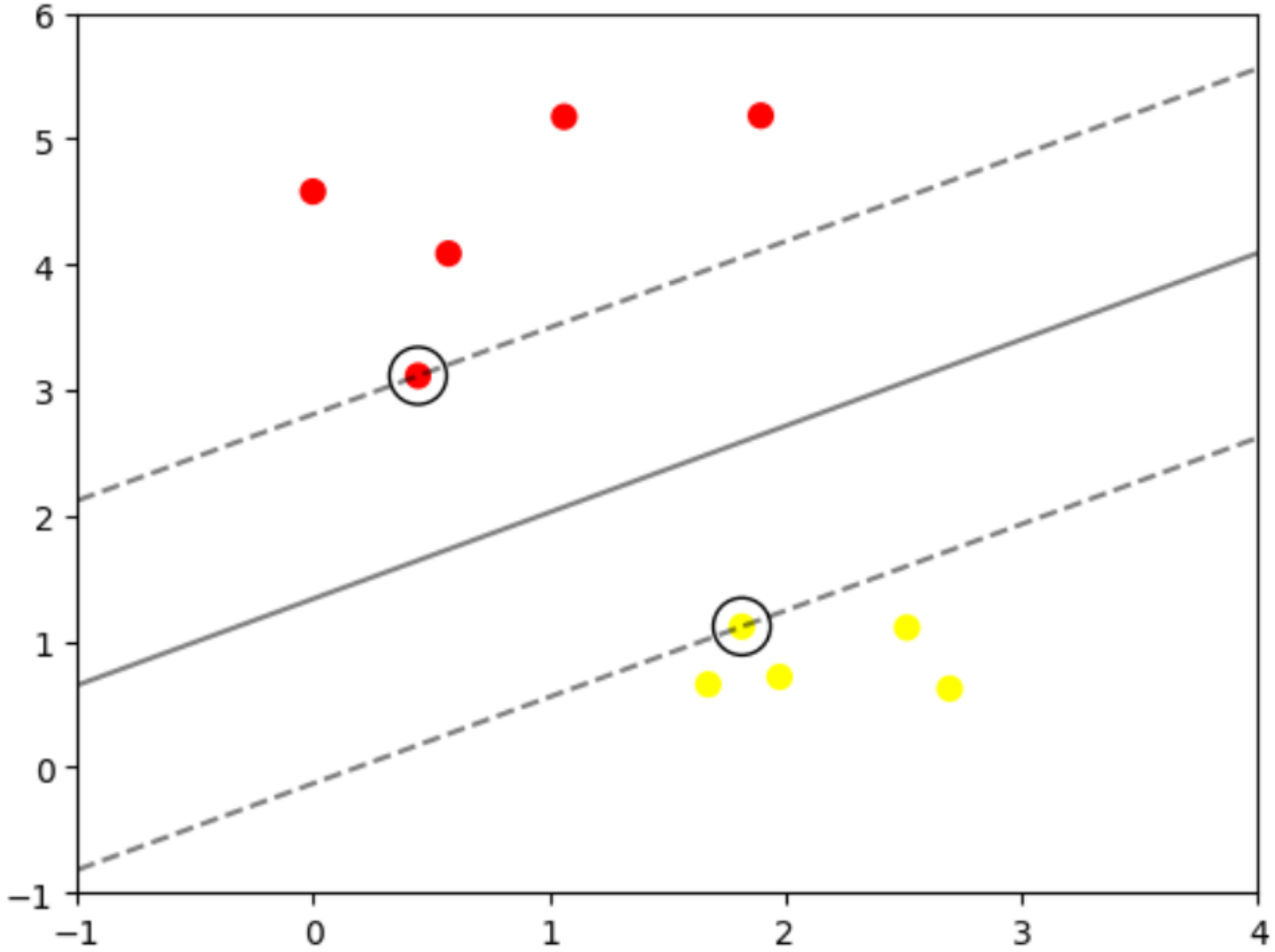


Support Vectors

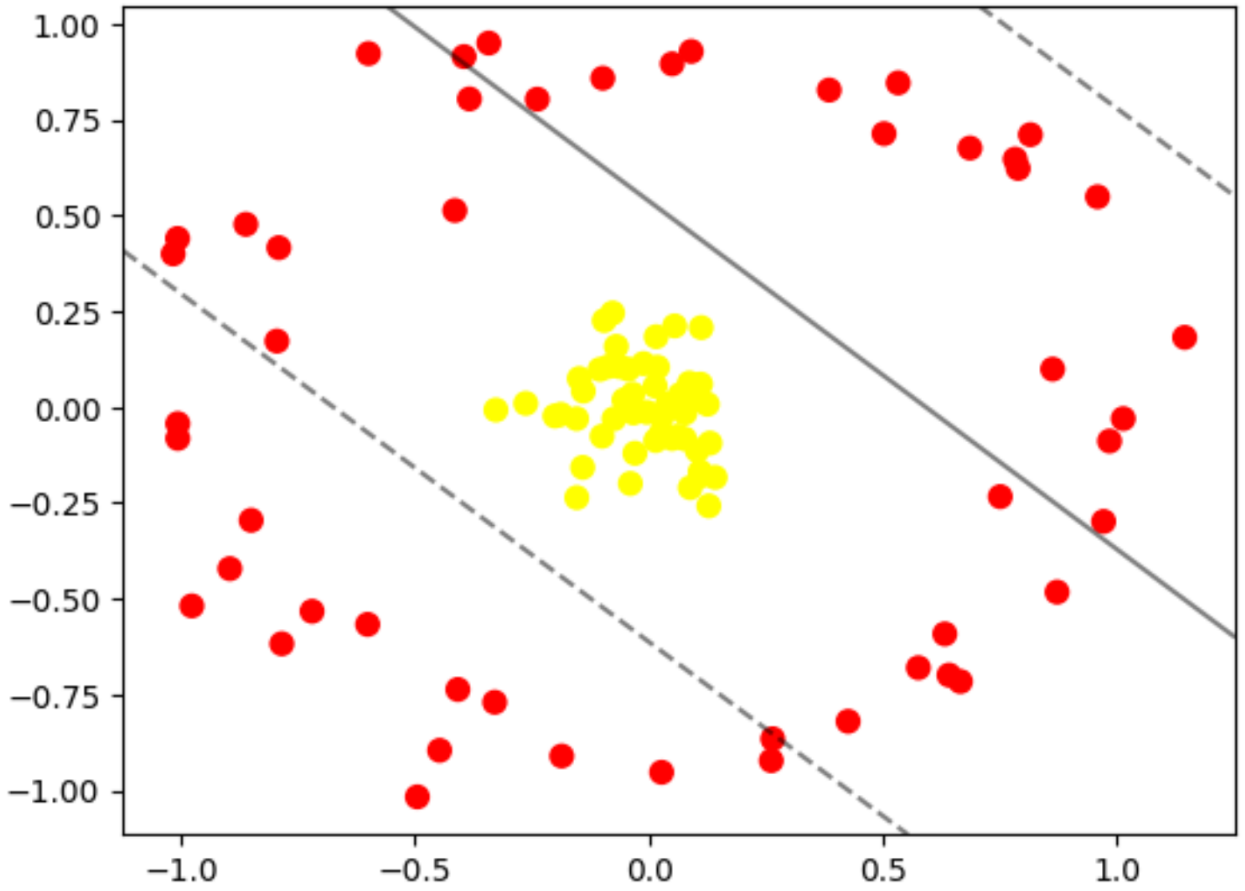



N = 60
6
5
4
3
2
1
0
−1
−1
0
1
2
3
4

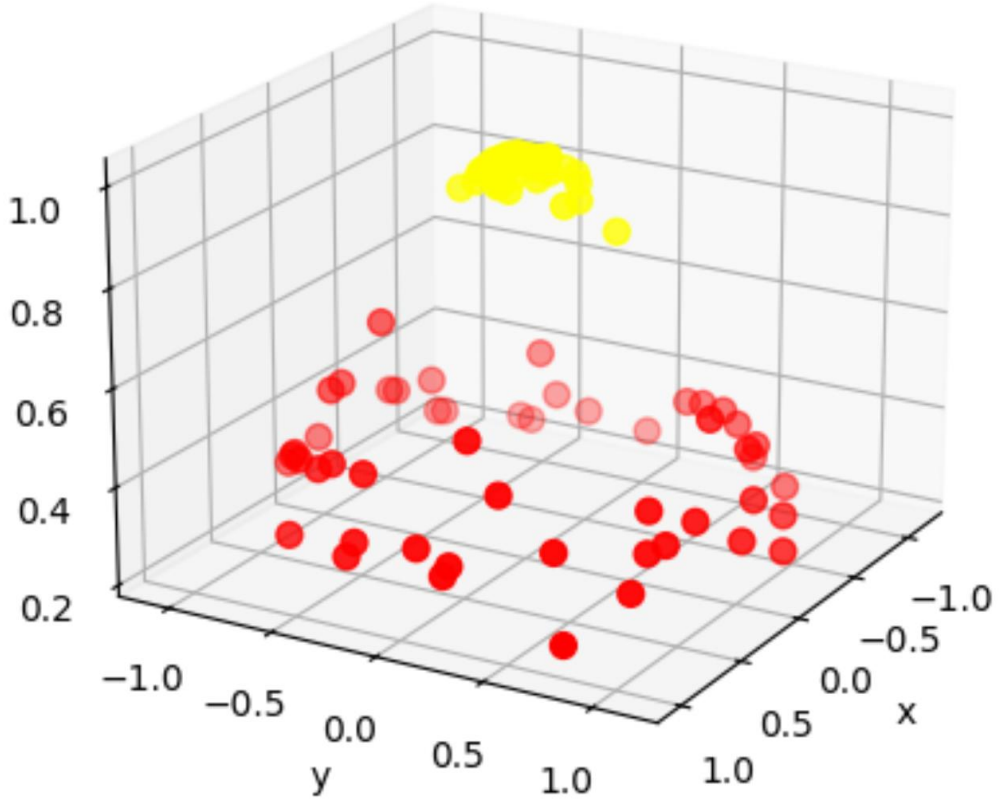

N = 120
6
5
4
3
2
1
0
−1
−1
0
1
2
3
4


6
5
4
3
2
1
0
−1
−1
0
1
2
3
4

Beyond Linear Boundaries: Kernel SVM



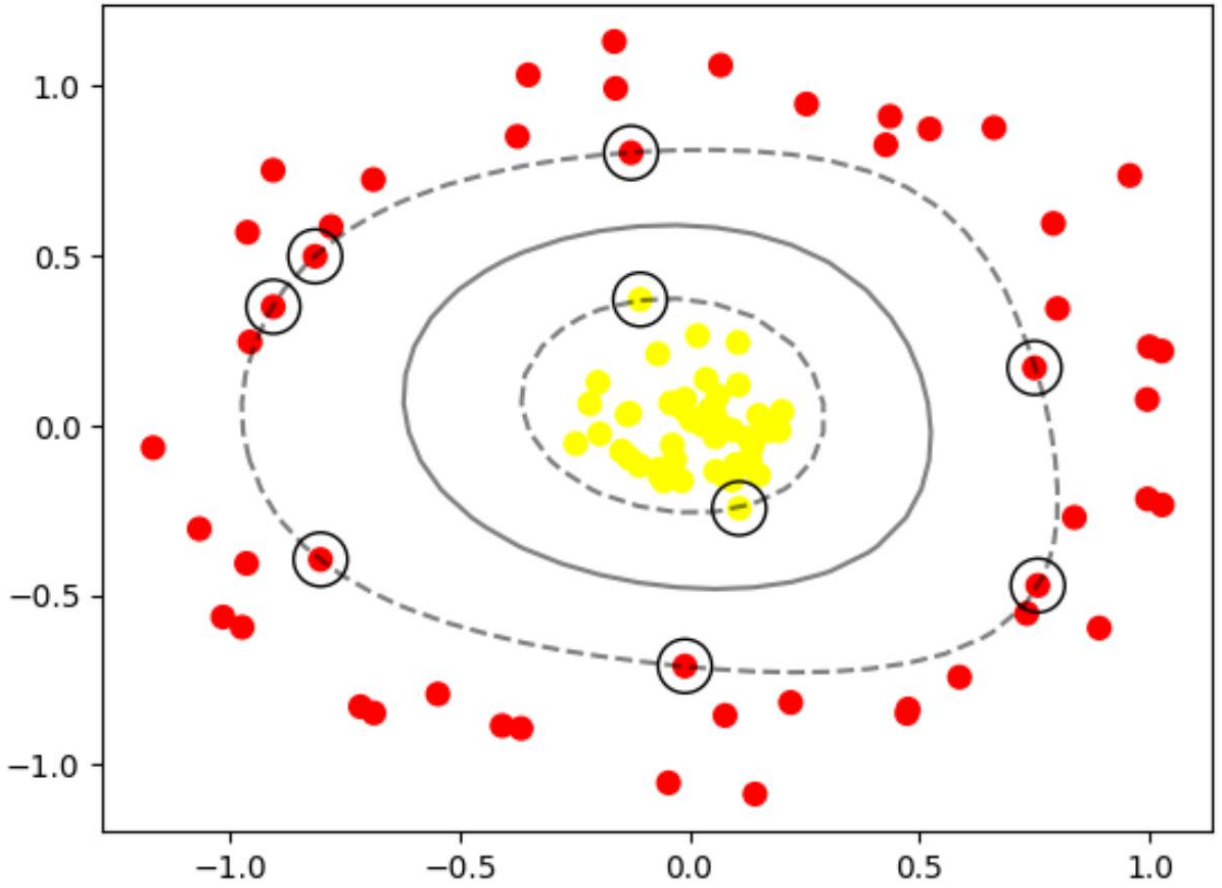
Radial Basis Function (RBF)



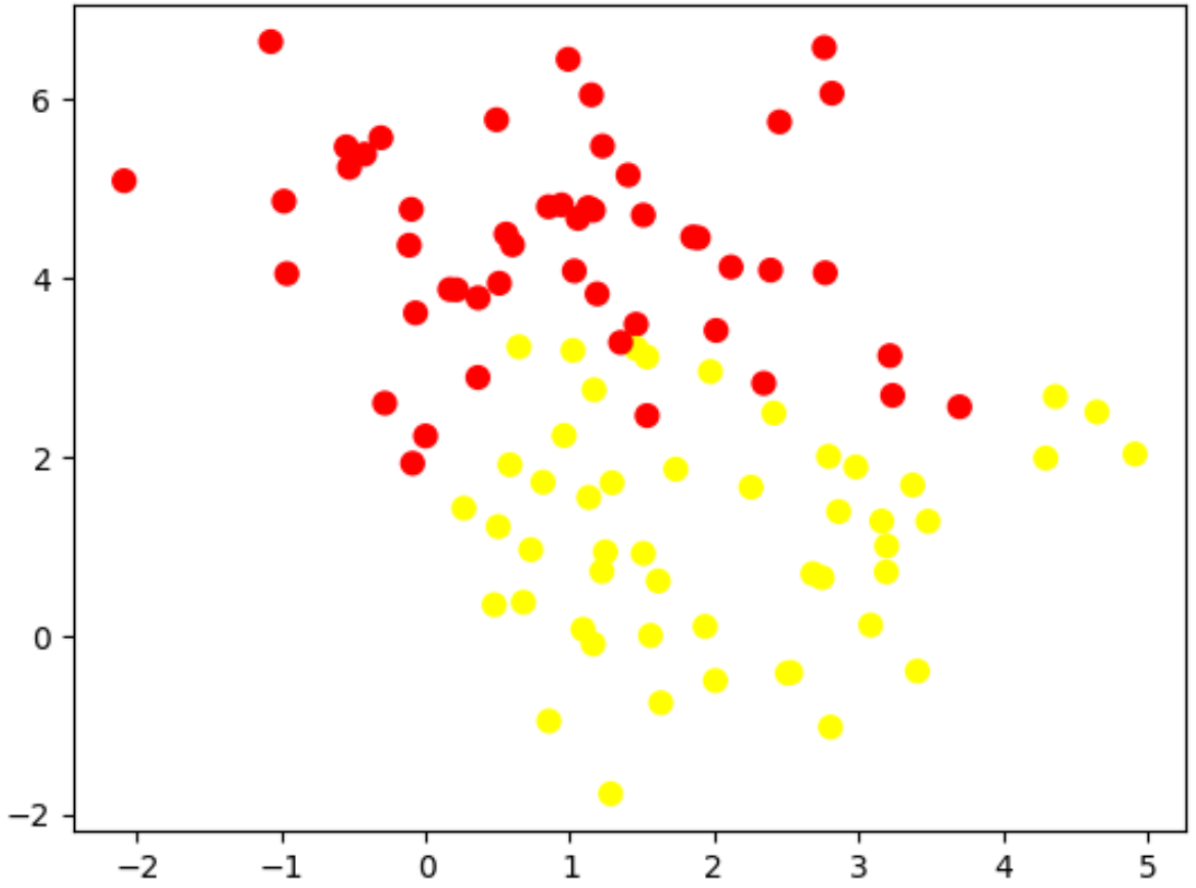
Kernel Transformation

- The need to make a choice of kernel function is a problem.
- Need to automatically find the best basis functions to use.
- One strategy
  - To compute a basis function centered at every point in the dataset.
  - This type of basis function transformation is known as a ***kernel transformation***.
  - A potential problem with this strategy—projecting N points into N dimensions—is that it might become very computationally intensive as N grows large.
- kernel trick, a fit on kernel-transformed data can be done implicitly—that is, without ever building the full N-dimensional representation of the kernel projection.
- This kernel trick is built into the SVM, and is one of the reasons the method is so powerful.

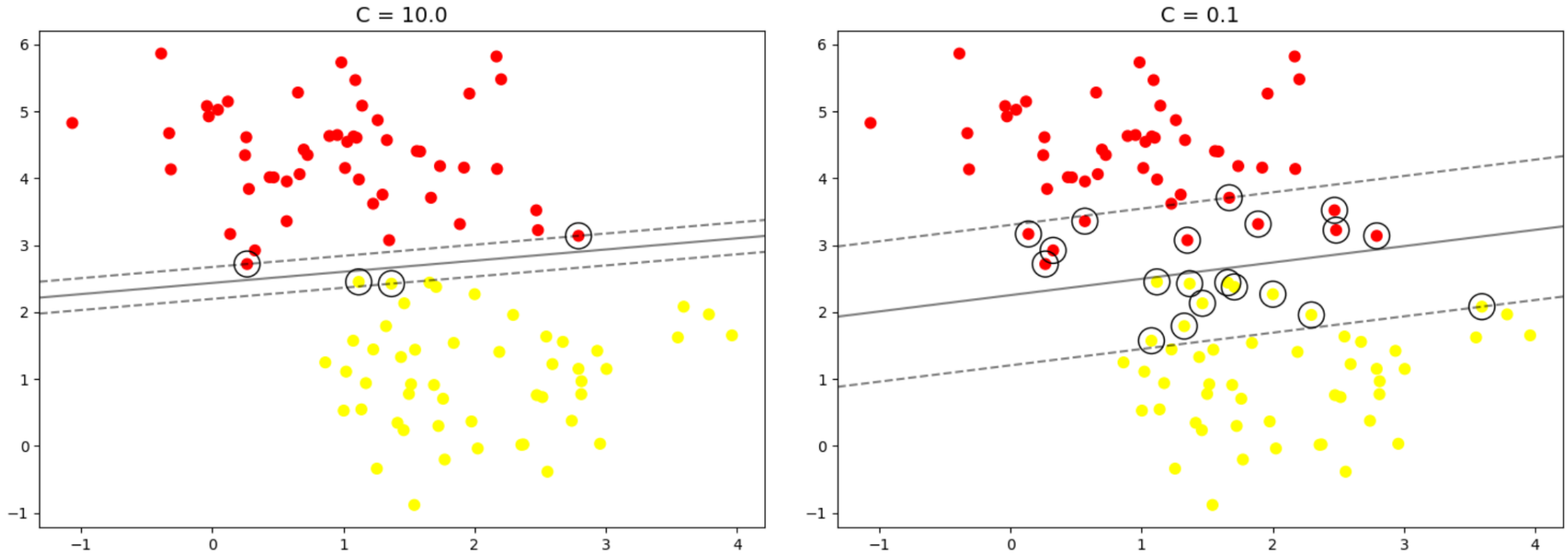
RBF Kernel



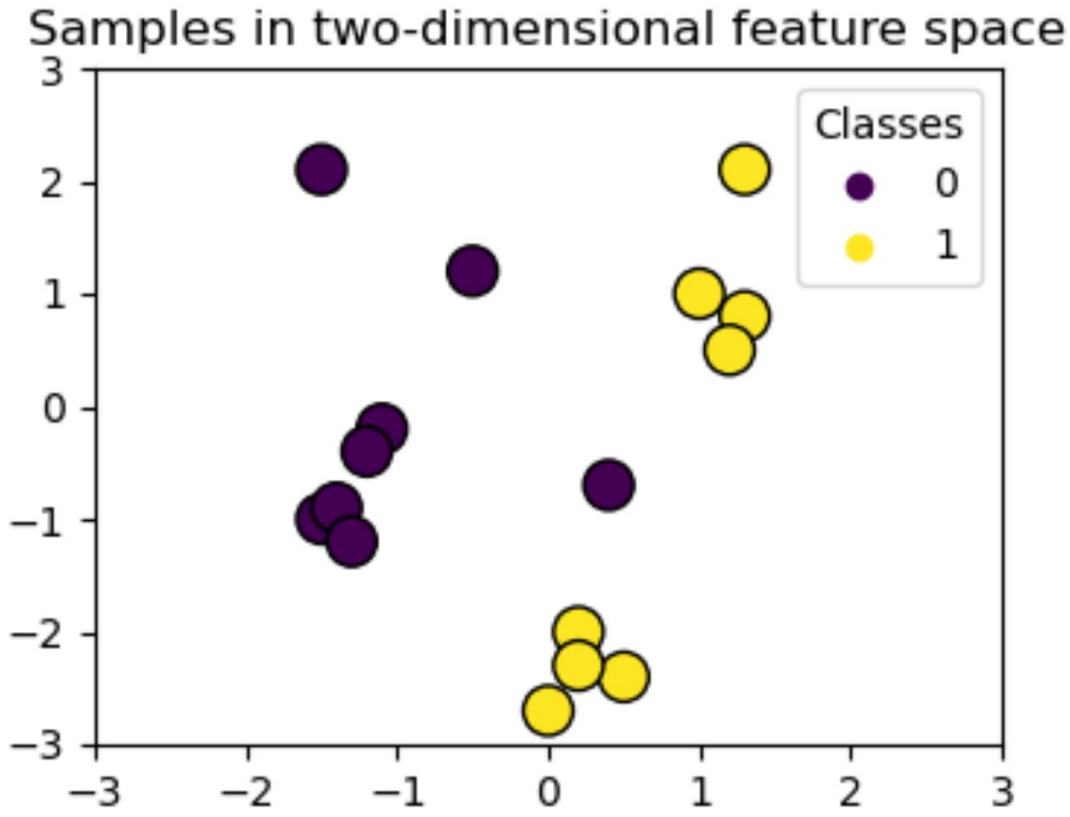
Tuning the SVM: Softening Margins



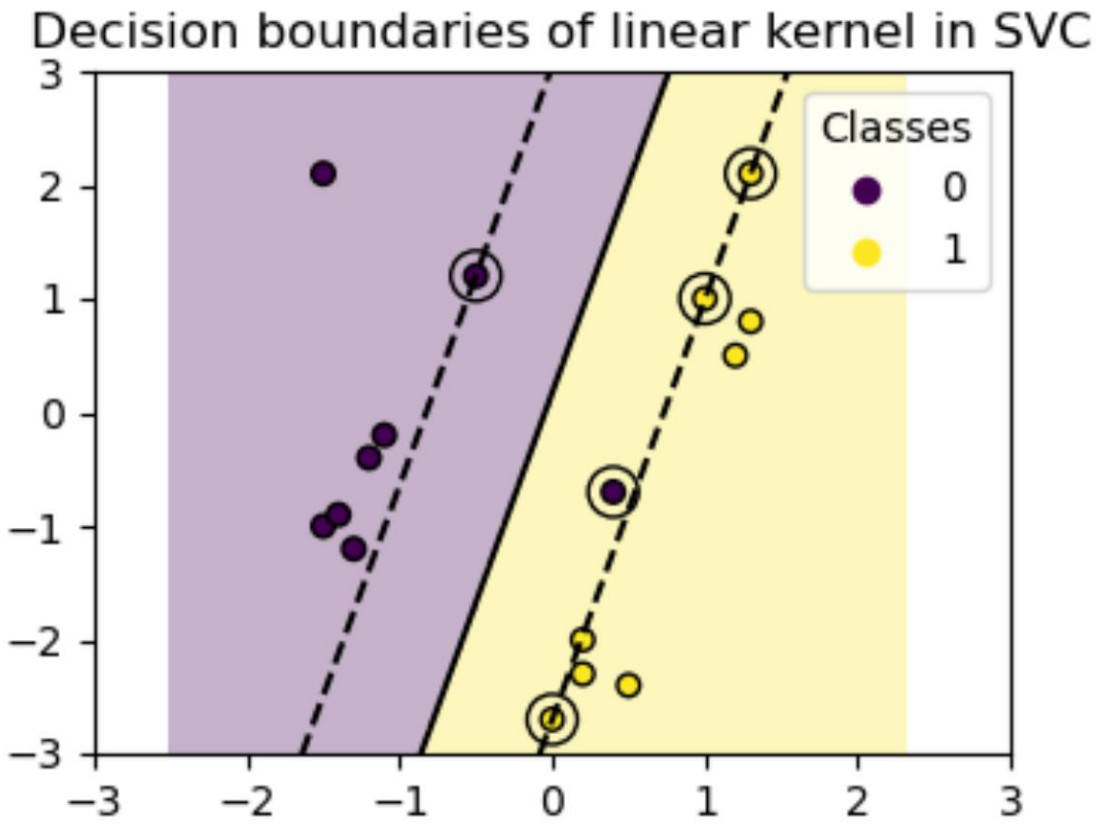
Tuning Parameter ‘C’



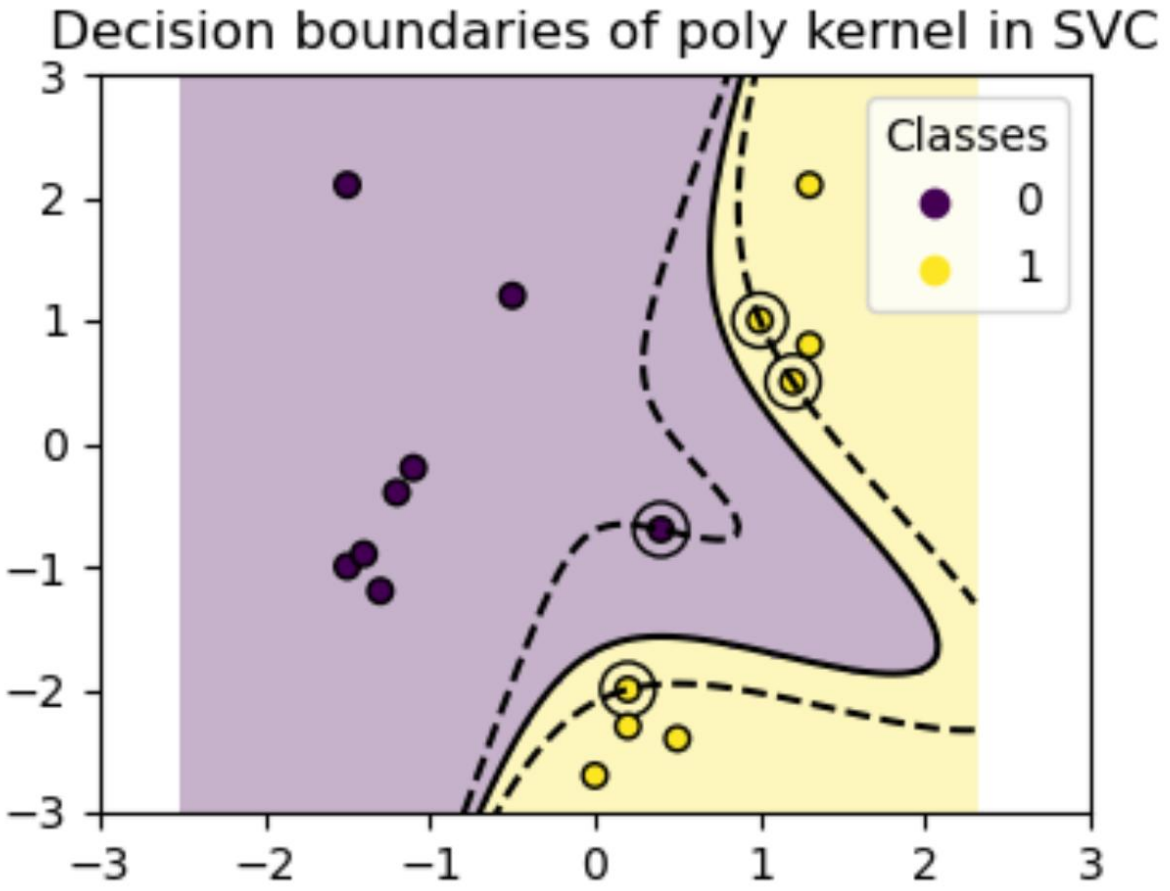
Classification Boundaries With Different SVM Kernels



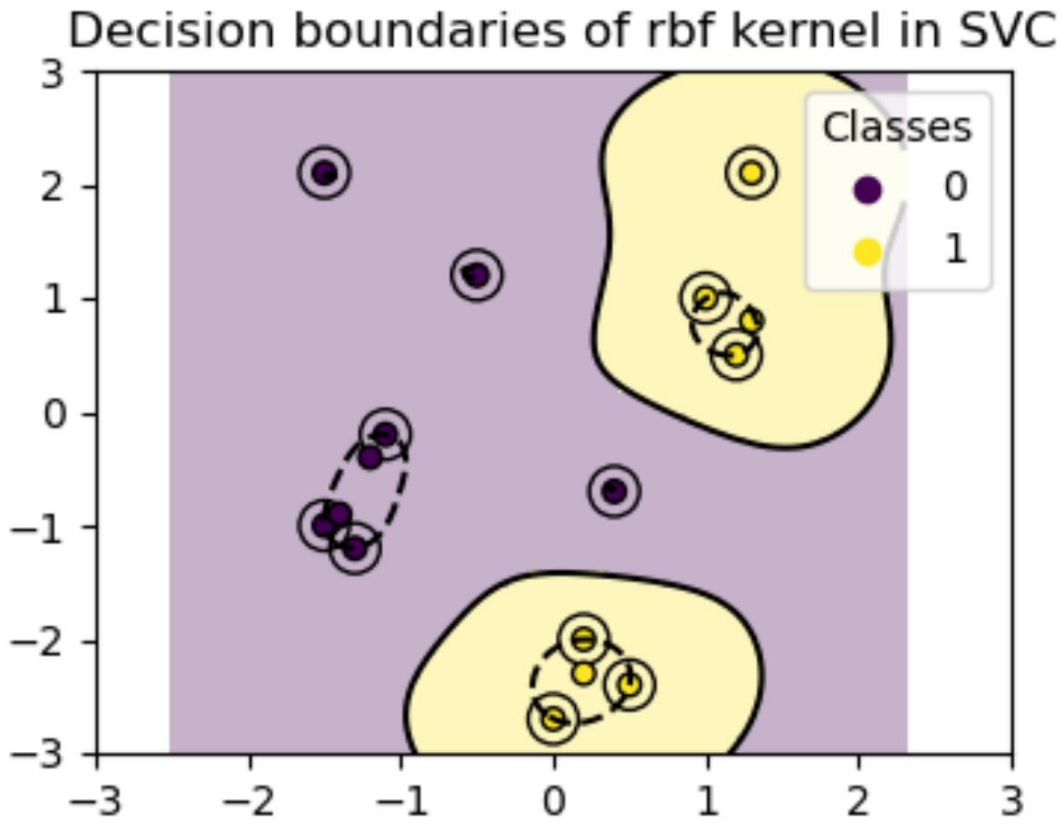
Linear Kernel



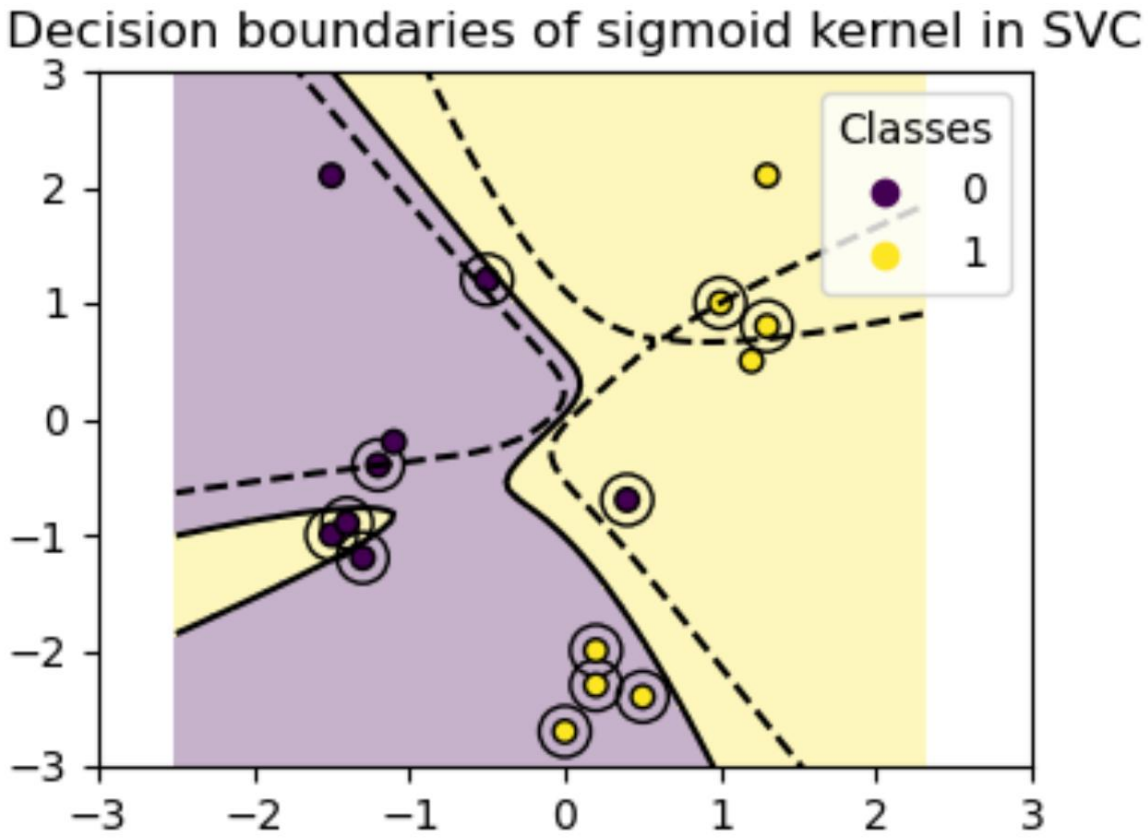
Polynomial Kernel

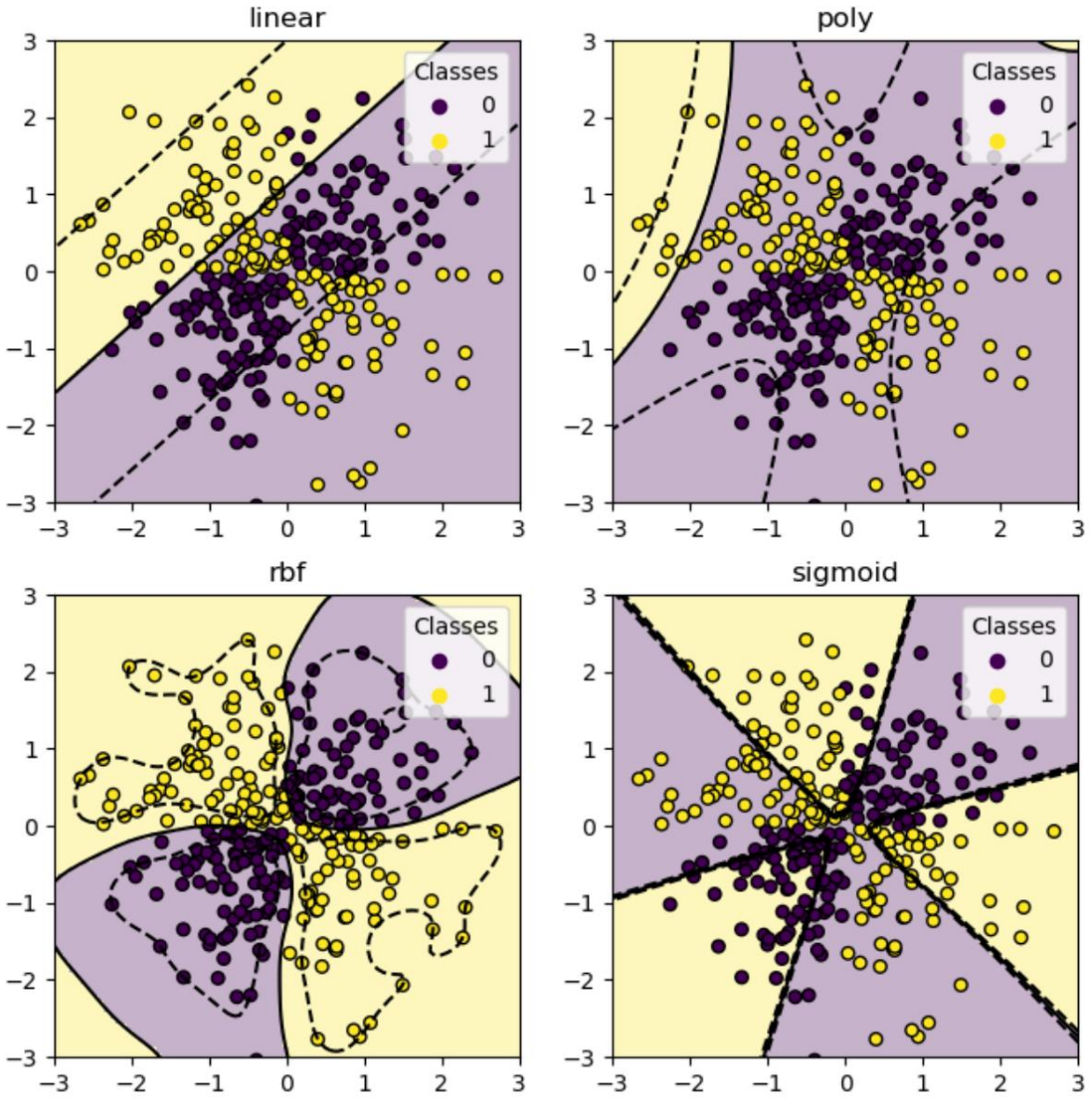


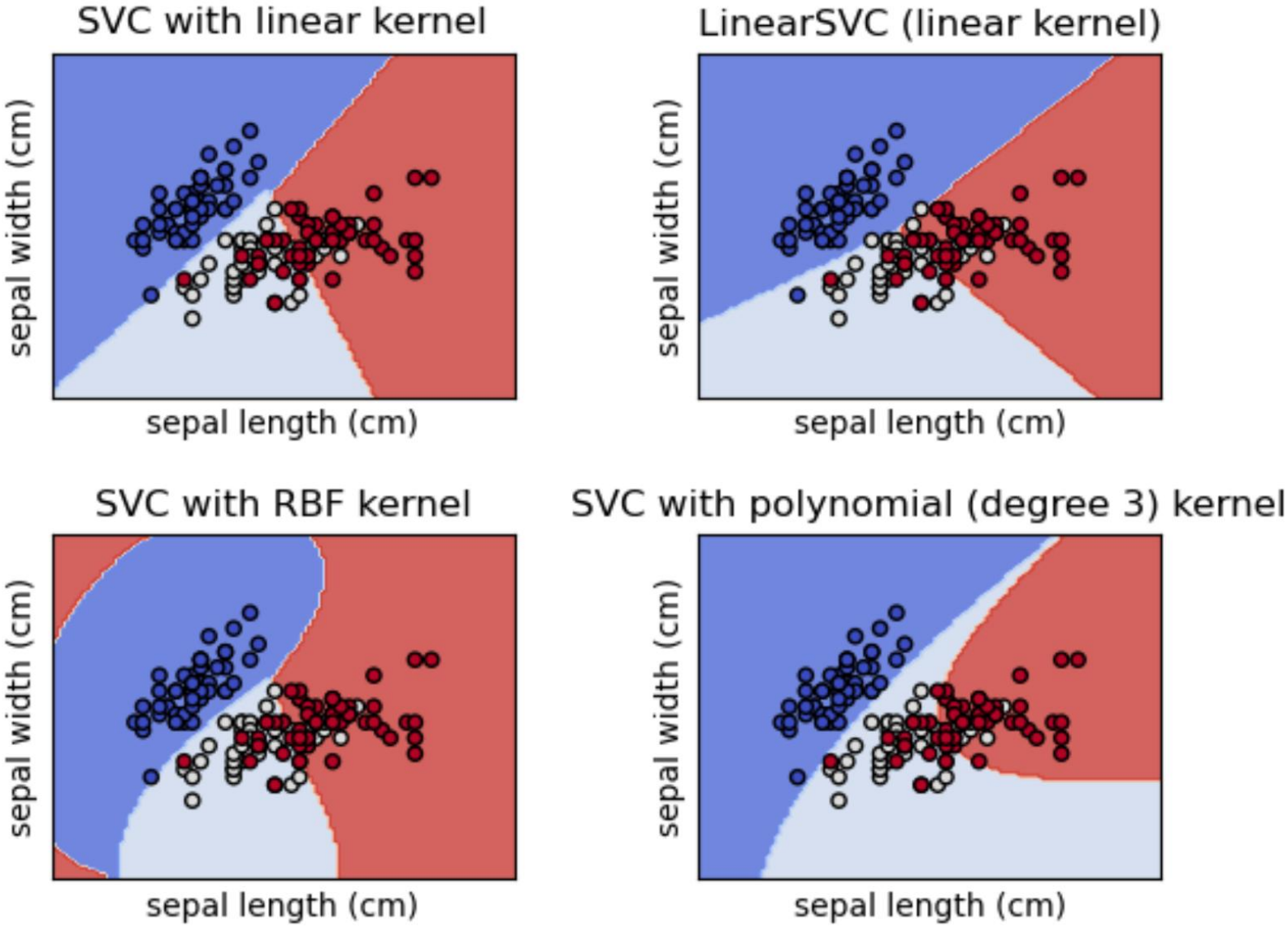
RBF Kernel


Decision boundaries of rbf kernel in SVC
Classes
0
1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

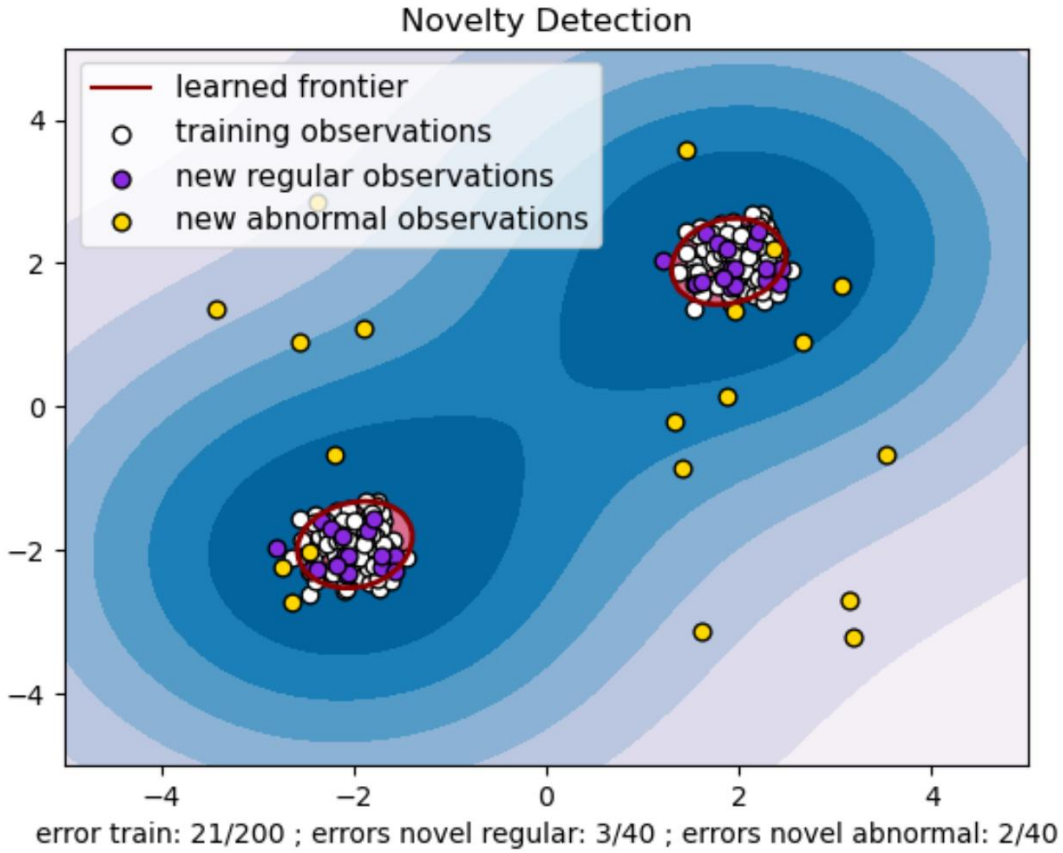
Sigmoid Kernel




linear
poly
rbf
sigmoid
Classes
0
1


SVC with linear kernel
LinearSVC (linear kernel)
SVC with RBF kernel
SVC with polynomial (degree 3) kernel
sepal width (cm)
sepal length (cm)

One-class SVM With Non-linear Kernel (RBF)



One-Class SVM versus One-Class SVM using Stochastic Gradient Descent