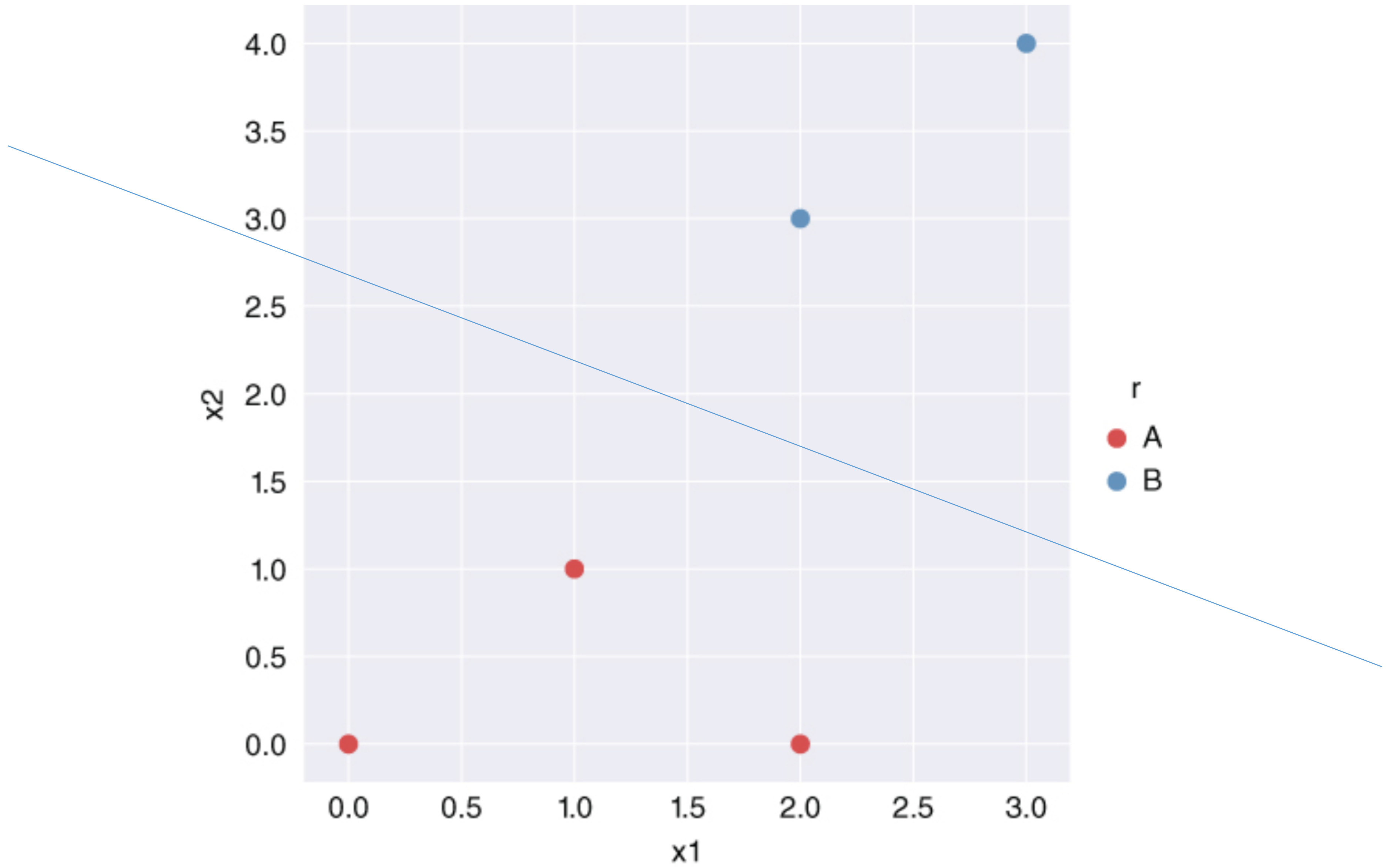
ITP 449

Support Vector Machines

Lecture 12

Support Vector Machines (SVMs)

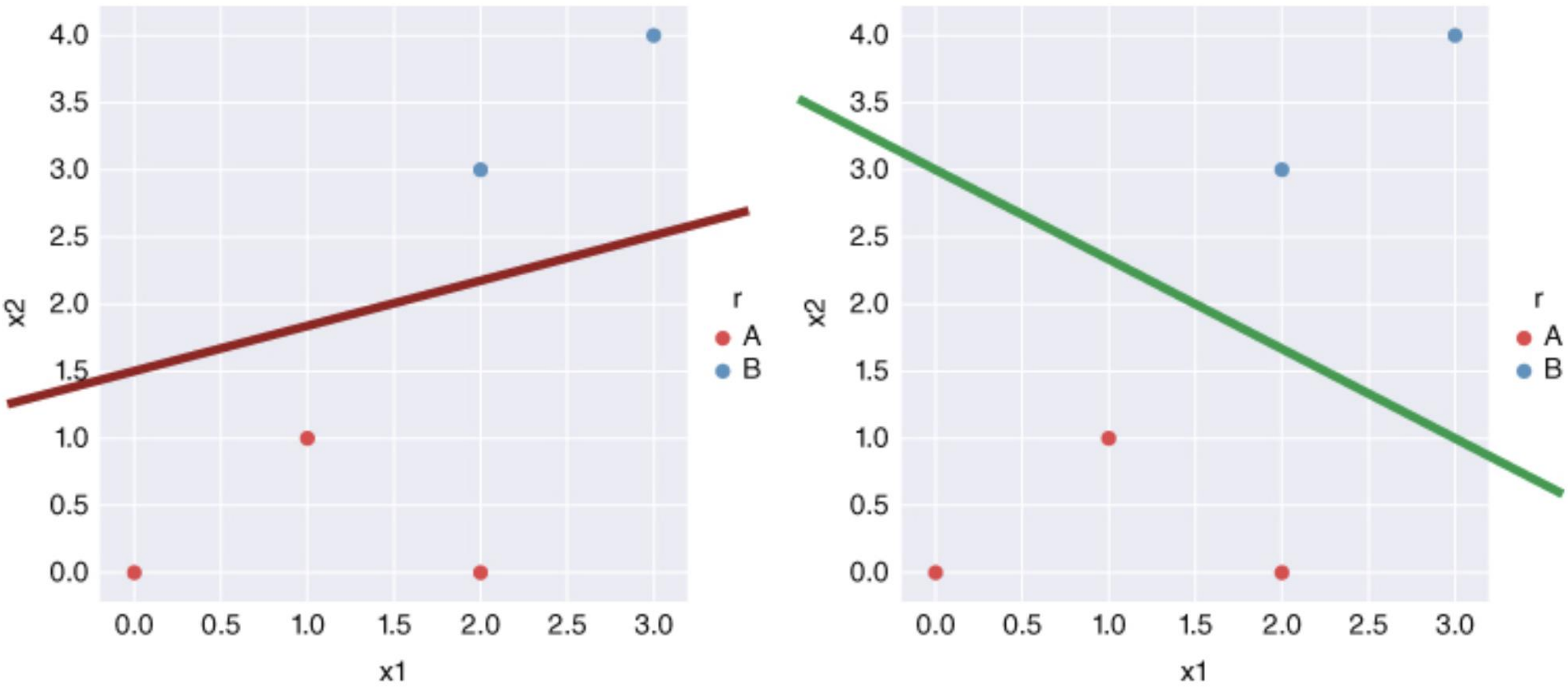
Used to predict categories and solve classification machine learning problems by defining a boundary between categories such that it maximizes the margin between classification groups.



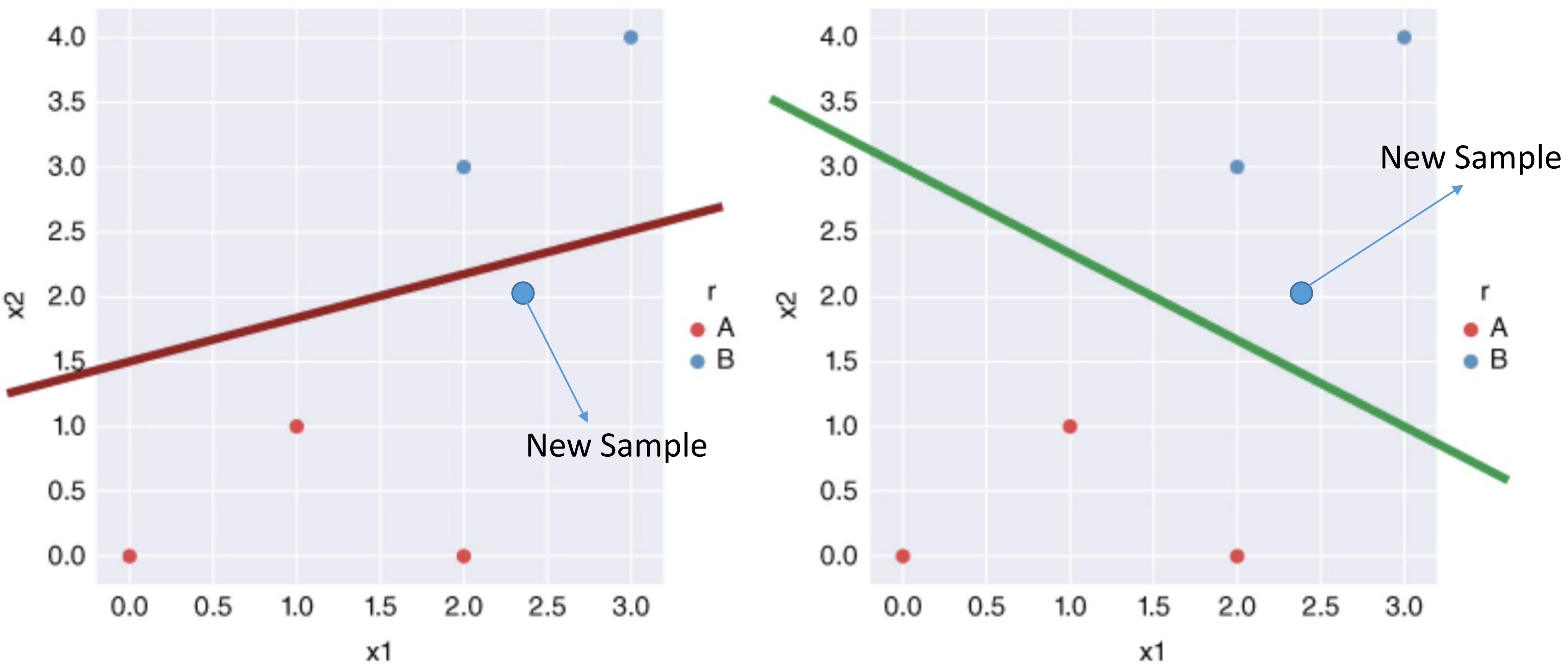
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
x2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x1
r
A
B

Problem: Draw a line (hyperplane) on the chart so as to distinctly classify the data points into red and blue classes.

Many such lines are possible. Find one that yields the maximum margin. The highest distance between the points.



4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
x2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x1
r
A
B
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
x2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x1
r
A
B



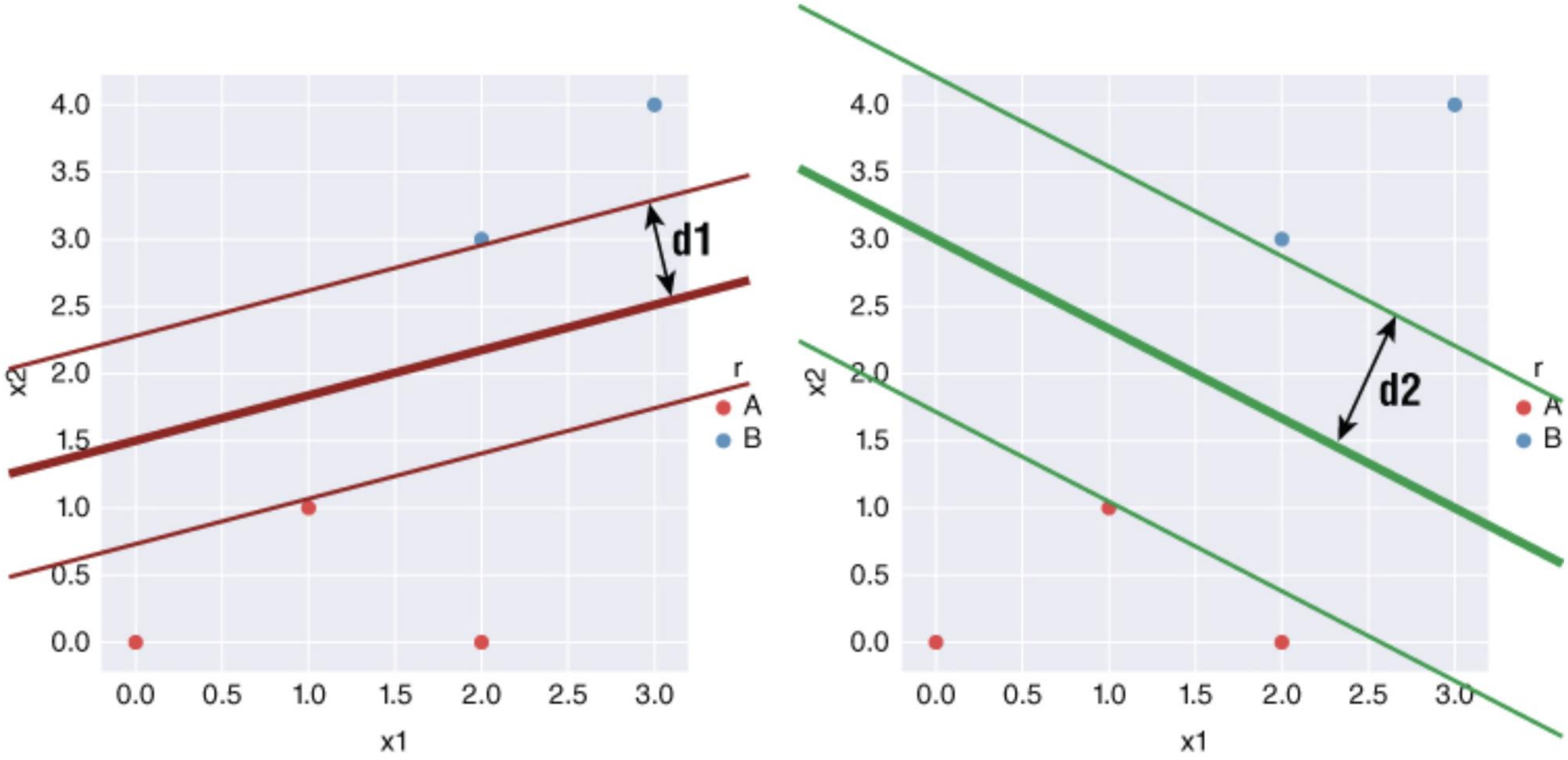
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
x2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x1
r
A
B
New Sample
New Sample

The bold line is the **hyperplane**. Also known as decision boundary.

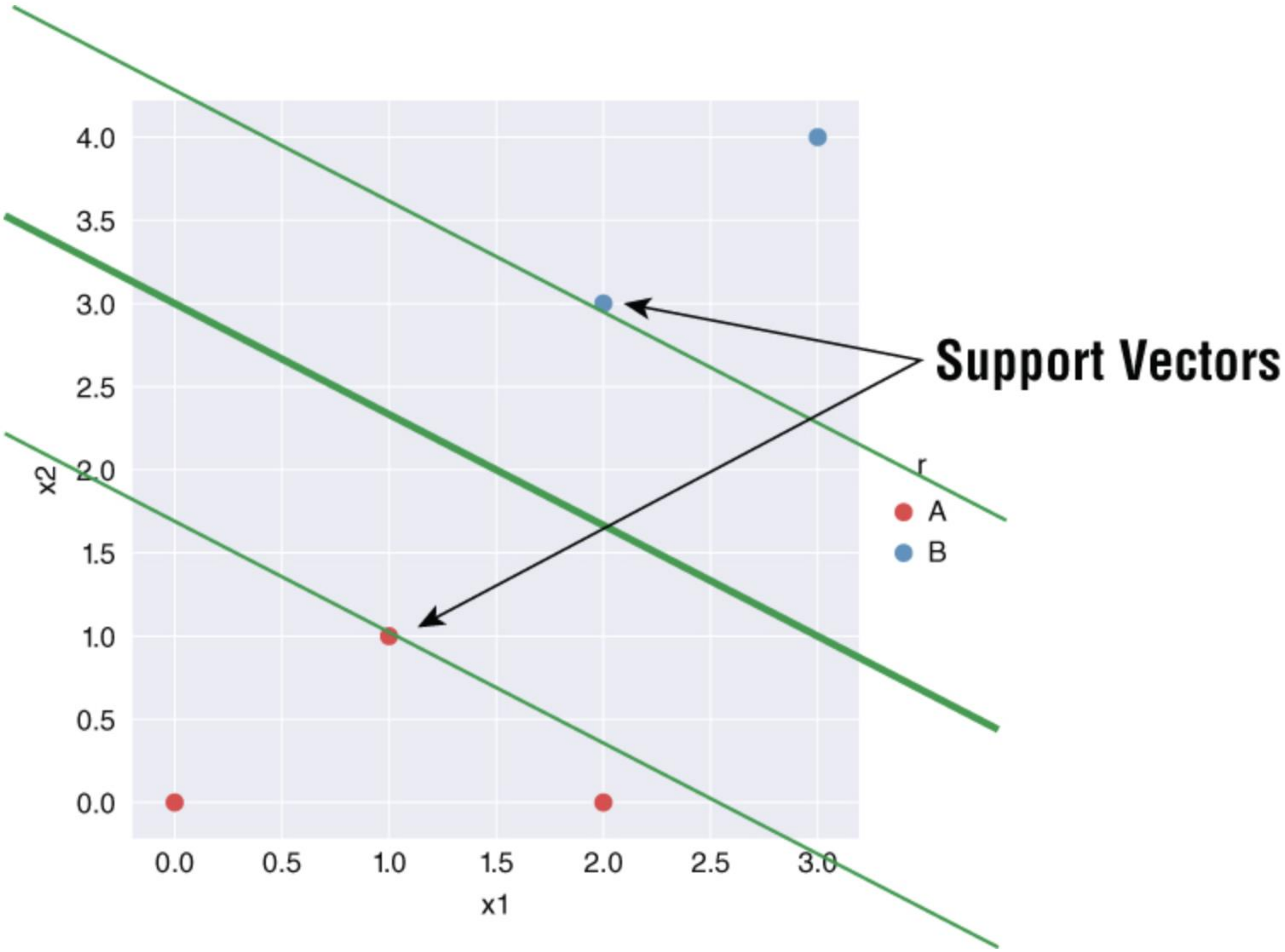
The reason to maximize the margin is that (new) points close to the decision boundary are not misclassified even if they are closer than the support vector. By having a large margin, you reduce this probability.

Support vectors are the points in each class that are close to the hyperplane.

Of course, there will be scenarios where it may not be possible to classify all points distinctly. There will be data points that end up in the opposite class. Instead of a straight line you could use a curved line to better separate the classes.



d2>d1


Support Vectors
x2
x1
r
A
B
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449_Fall2020 - in_class_coding.py

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

in_class_coding.py

```python
import pandas as pd

data = [[0, 0, 'A'], [1, 1, 'A'], [2, 3, 'B'],[2, 0, 'A'], [3, 4, 'B']]
df = pd.DataFrame(data)
df.columns = ['x1', 'x2', 'r']

print(df)
```

Run: in_class_coding

```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Cod
   x1  x2  r
0   0   0  A
1   1   1  A
2   2   3  B
3   2   0  A
4   3   4  B

Process finished with exit code 0
```

4: Run TODO 6: Problems Terminal Python Console Event Log

10:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

ITP449_Fall2020 - in_class_coding.py

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

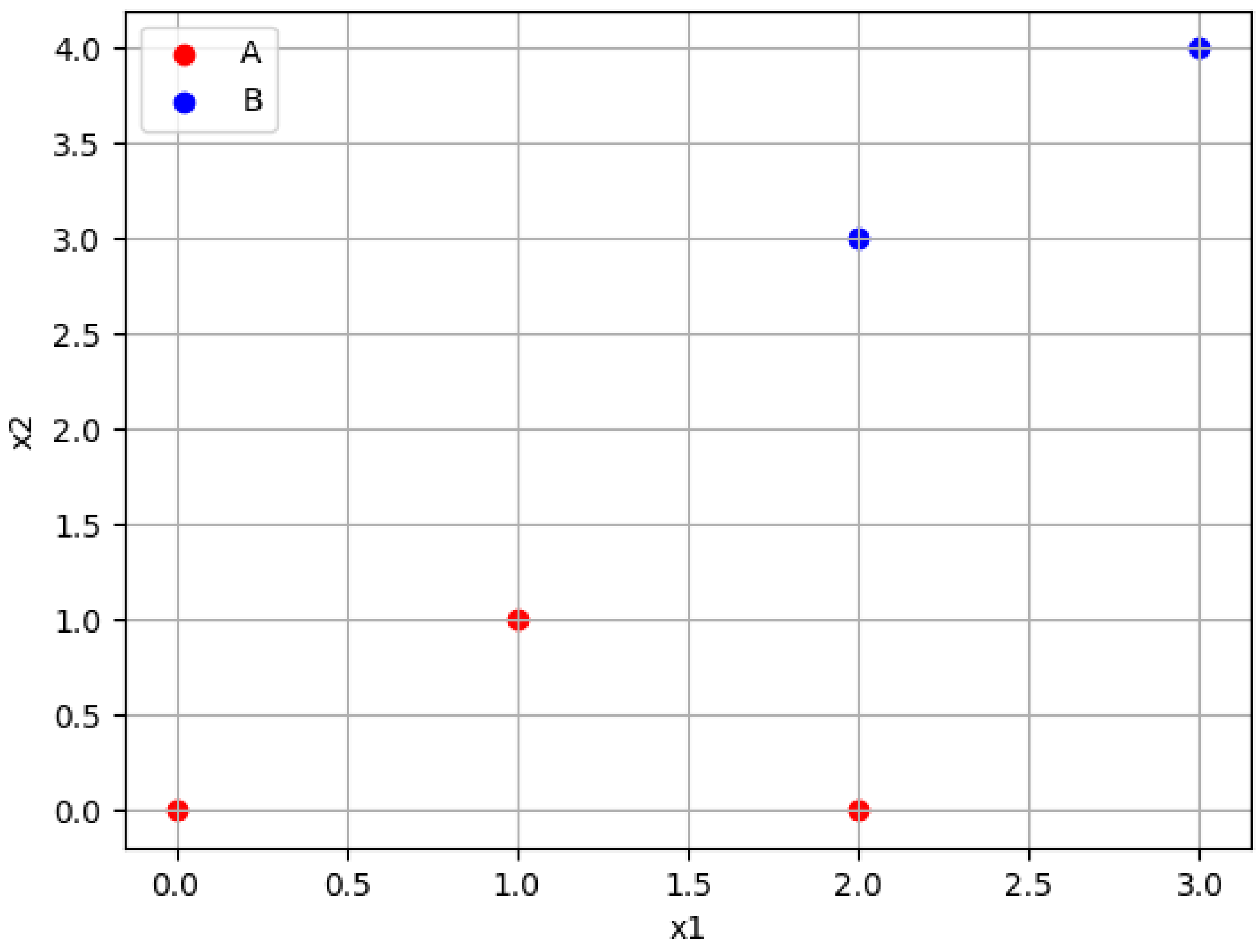
in_class_coding.py

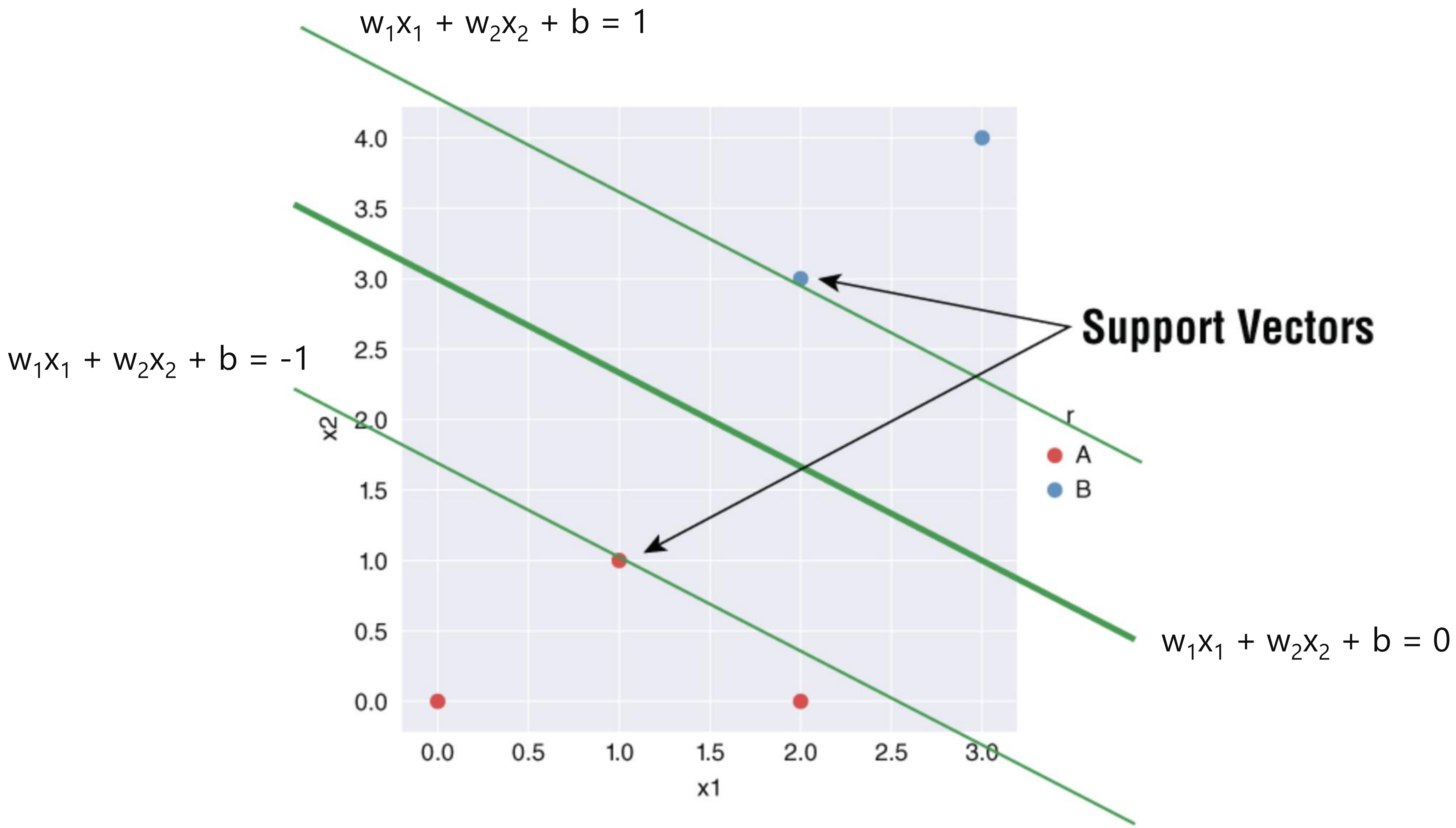
```python
import pandas as pd
import matplotlib.pyplot as plt

data = [[0, 0, 'A'], [1, 1, 'A'], [2, 3, 'B'],[2, 0, 'A'], [3, 4, 'B']]
df = pd.DataFrame(data)
df.columns = ['x1', 'x2', 'r']

plt.scatter(df.loc[df.r == 'A', 'x1'], df.loc[df.r == 'A', 'x2'], color='r', label='A')
plt.scatter(df.loc[df.r == 'B', 'x1'], df.loc[df.r == 'B', 'x2'], color='b', label='B')

plt.xlabel('x1')
plt.ylabel('x2')
plt.legend()
plt.grid()
plt.show()
```

Run: in_class_coding

```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Cod

Process finished with exit code 0
```

17:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


A
B
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
x2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x1



$w_1x_1 + w_2x_2 + b = 1$
$w_1x_1 + w_2x_2 + b = -1$
$w_1x_1 + w_2x_2 + b = 0$
Support Vectors
x2
x1
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
r
A
B

```python
from sklearn import svm

model = svm.SVC(kernel='linear')

X = df[['x1', 'x2']]
y = df['r']

model.fit(X, y)

print('Weights w1 and w2:', model.coef_)
print('Bias:', model.intercept_)
```

SVC = Support Vector Classification
Kernel = a technique to use higher dimensional space in order to make the classes separable (when they are not).
Options: linear, polynomial, sigmoid etc.

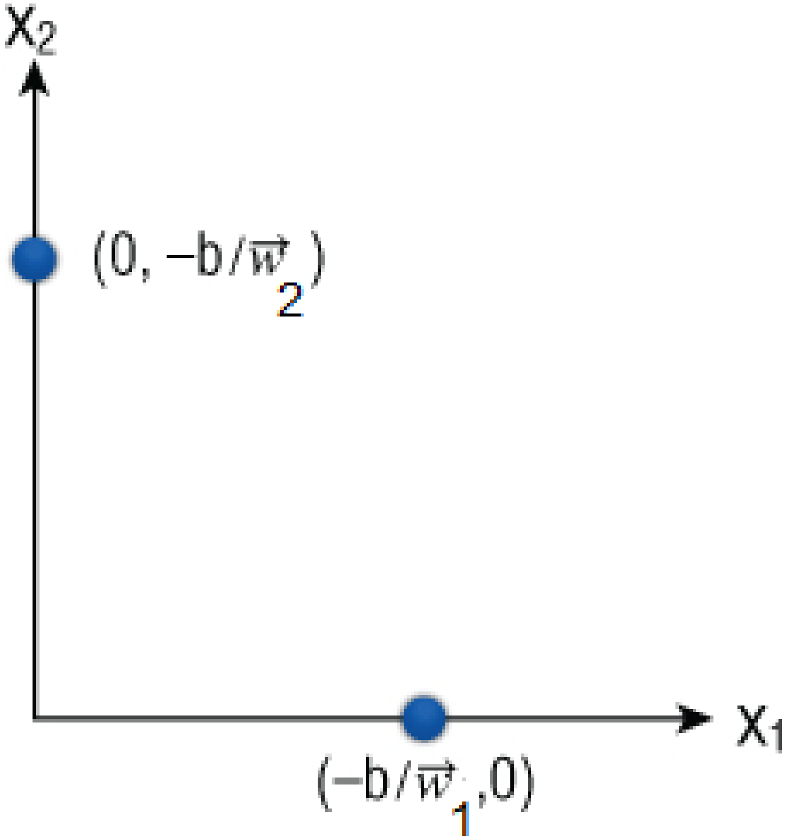
```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Cod
Weights w1 and w2: [[0.4 0.8]]
Bias: [-2.2]

Process finished with exit code 0
```

$$w_1x_1 + w_2x_2 + b = 0$$

$$w_1(0) + w_2x_2 + b = 0$$

$$w_2x_2 = -b$$

$$x_2 = -b / w_2$$



$$\text{slope} = (-b / w_2) / (b / w_1)$$
$$\text{slope} = -(w_1 / w_2)$$

$$w_1x_1 + w_2x_2 + b = 0$$

$$w_1x_1 + w_2(0) + b = 0$$

$$w_1x_1 = -b$$

$$x_1 = -b / w_1$$

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449_Fall2020 - in_class_coding.py

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

in_class_coding.py

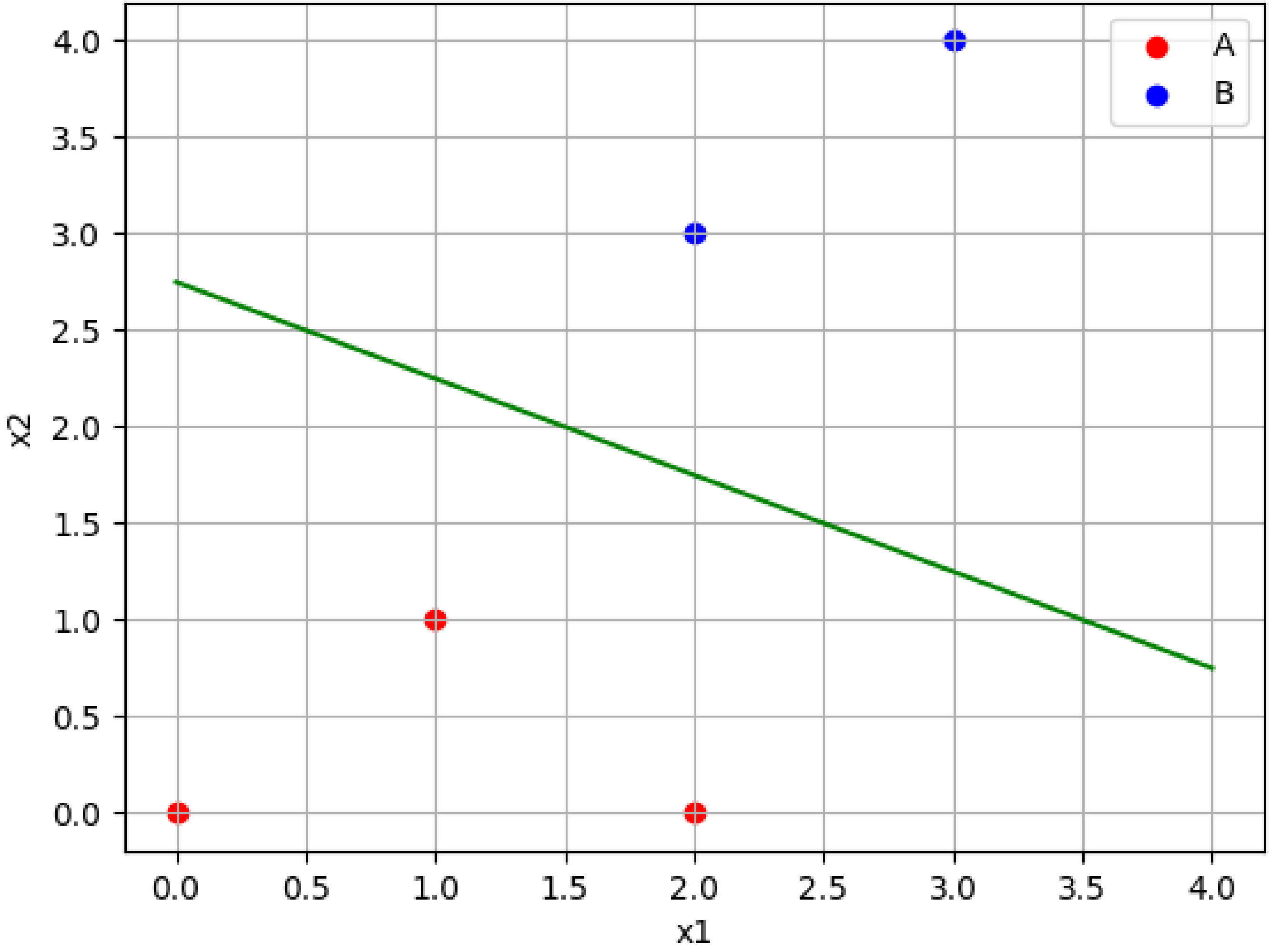
```python
print('Weights w1 and w2:', model.coef_)
print('Bias:', model.intercept_)

w = model.coef_[0]
slope = -w[0] / w[1]
b = model.intercept_[0]

xx = np.linspace(0, 4)
yy = slope * xx - (b / w[1])
plt.plot(xx, yy, color='g')
plt.show()
```

Run: in_class_coding

```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Cod
Weights w1 and w2: [[0.4 0.8]]
Bias: [-2.2]
```

4: Run TODO 6: Problems Terminal Python Console Event Log

4:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


A
B
x1
x2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

```python
xx = np.linspace(0, 4)
yy = slope * xx - (b / w[1])
plt.plot(xx, yy, color='g')
plt.show()

print(model.predict([[3, 3]])) # B
print(model.predict([[4, 0]])) # A
print(model.predict([[2, 2]])) # B
print(model.predict([[1, 2]])) # A
print(model.predict([[1.5, 1.5]])) # A
```

Run: in_class_coding

```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Cod
Weights w1 and w2: [[0.4 0.8]]
Bias: [-2.2]
['B']
['A']
['B']
['A']
['A']

Process finished with exit code 0
```

Kernels

Transform input data into the required form: linear, polynomial, and radial basis function (RBF). Nonlinear kernels used to separate categories with curved hyperplanes.

Regularization

Parameter C controls the tolerance of classification error. A high C will seek to classify all points correctly, narrowing the margin. A low C will aim for the widest margin possible but will result in some points being misclassified.

Demo IRIS

Do the following:

Import the Iris data into a DataFrame. Print the columns, data types and number of non-null rows. Perform any data wrangling that is required.

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

in_class_coding.py

```python
import pandas as pd
import os

os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
pd.set_option('display.max_columns', None)

irisDf = pd.read_csv('iris.csv')
print(irisDf.info())
```

Run: in_class_coding

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Sepal length   150 non-null    float64
 1   Sepal width    150 non-null    float64
 2   Petal length   150 non-null    float64
 3   Petal width    150 non-null    float64
 4   Species        150 non-null    object
dtypes: float64(4), object(1)
memory usage: 5.3+ KB
```

Do the following:

What are the species categories?

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449_Fall2020 - in_class_coding.py

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

in_class_coding.py

```python
import pandas as pd
import os

os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
pd.set_option('display.max_columns', None)

irisDf = pd.read_csv('iris.csv')
print(irisDf.info())
print(irisDf.Species.unique())
```

Run: in_class_coding

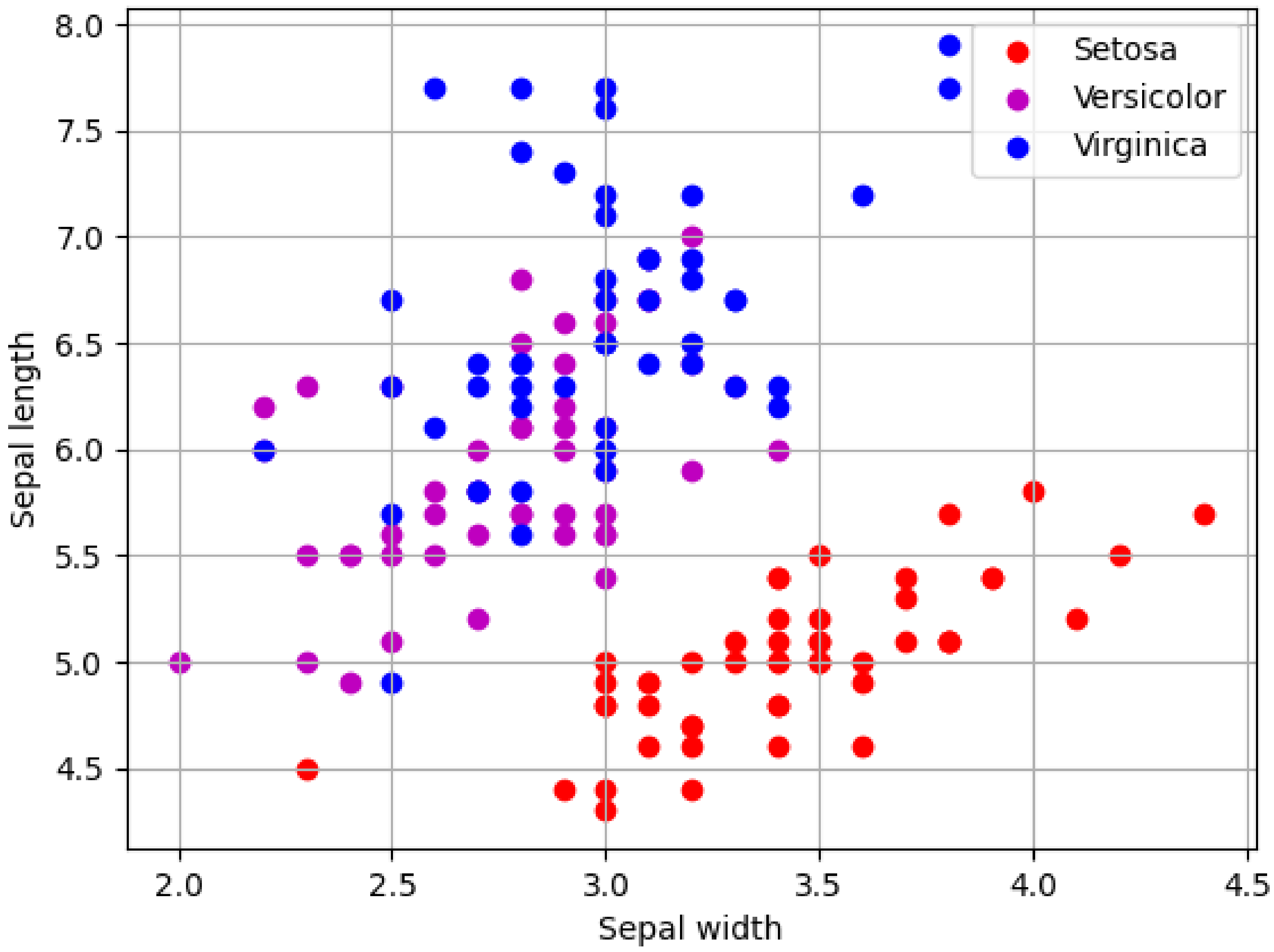
```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Sepal length  150 non-null    float64
 1   Sepal width   150 non-null    float64
 2   Petal length  150 non-null    float64
 3   Petal width   150 non-null    float64
 4   Species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 5.3+ KB
None
['I. setosa' 'I. versicolor' 'I. virginica']
```

4: Run TODO 6: Problems Terminal Python Console Event Log

1:87 CRLF UTF-8 4 spaces Python 3.8 (Class)

Do the following:

Plot a scatterplot of the sepal_width and sepal_length, assigning different colors to the species categories. What is your observation?



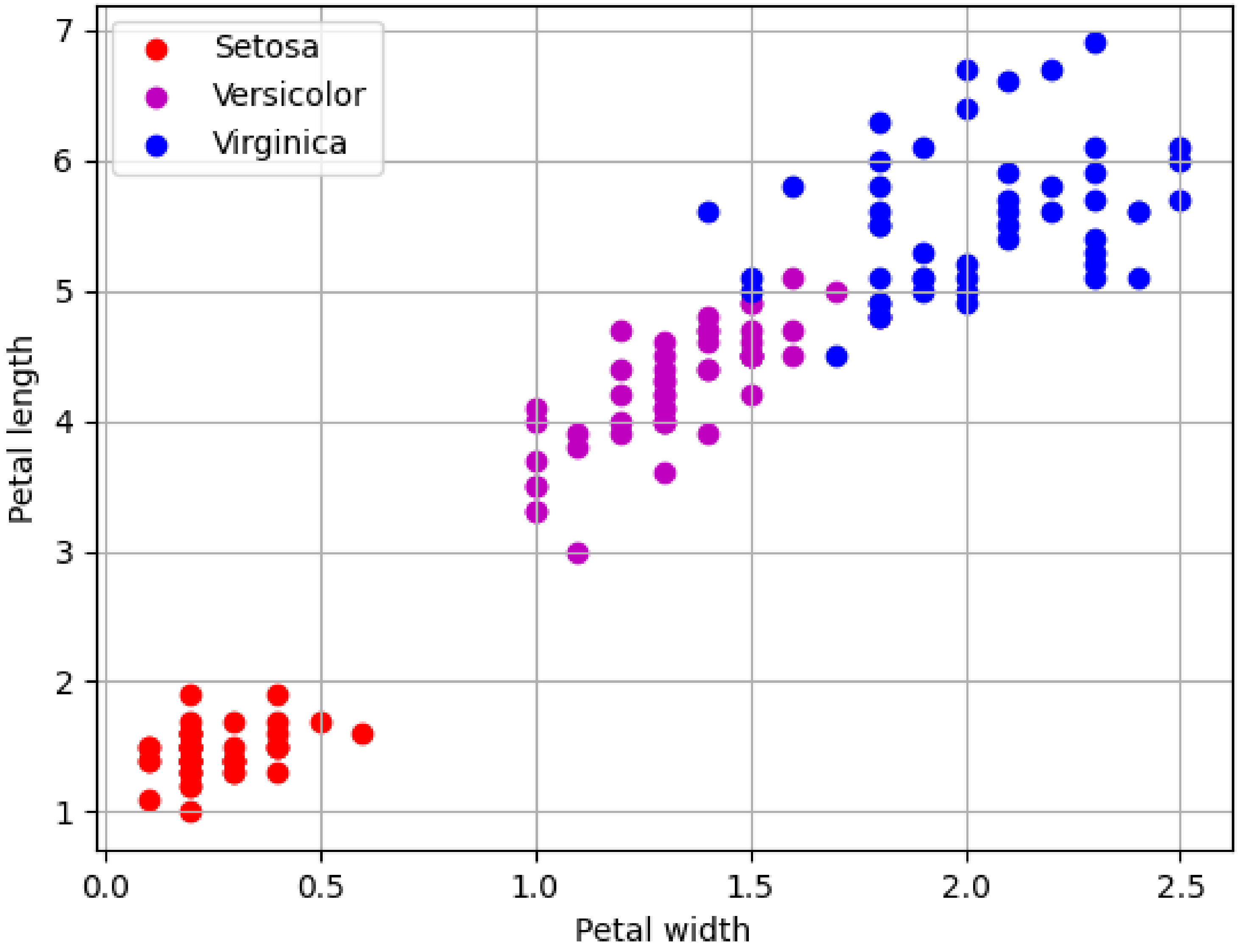
Setosa
Versicolor
Virginica
Sepal length
Sepal width
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.0
2.5
3.0
3.5
4.0
4.5

```python
plt.scatter(irisDf.loc[irisDf.Species == 'I. setosa', 'Sepal width'],
            irisDf.loc[irisDf.Species == 'I. setosa', 'Sepal length'],
            color='r', label='Setosa')

plt.scatter(irisDf.loc[irisDf.Species == 'I. versicolor', 'Sepal width'],
            irisDf.loc[irisDf.Species == 'I. versicolor', 'Sepal length'],
            color='m', label='Versicolor')

plt.scatter(irisDf.loc[irisDf.Species == 'I. virginica', 'Sepal width'],
            irisDf.loc[irisDf.Species == 'I. virginica', 'Sepal length'],
            color='b', label='Virginica')

plt.xlabel('Sepal width')
plt.ylabel('Sepal length')
plt.legend()
plt.grid()
plt.show()
```

Process finished with exit code 0

Do the following:

Plot a scatterplot of the petal_width and petal_length, assigning different colors to the species categories. What is your observation?


Setosa
Versicolor
Virginica
Petal length
Petal width
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449_Fall2020 - in_class_coding.py

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

in_class_coding.py

```python

plt.scatter(irisDf.loc[irisDf.Species == 'I. setosa', 'Petal width'],
            irisDf.loc[irisDf.Species == 'I. setosa', 'Petal length'],
            color='r', label='Setosa')

plt.scatter(irisDf.loc[irisDf.Species == 'I. versicolor', 'Petal width'],
            irisDf.loc[irisDf.Species == 'I. versicolor', 'Petal length'],
            color='m', label='Versicolor')

plt.scatter(irisDf.loc[irisDf.Species == 'I. virginica', 'Petal width'],
            irisDf.loc[irisDf.Species == 'I. virginica', 'Petal length'],
            color='b', label='Virginica')

plt.xlabel('Petal width')
plt.ylabel('Petal length')
plt.legend()
plt.grid()
plt.show()

```

Run: in_class_coding

4: Run TODO 6: Problems Terminal Python Console Event Log

CRLF UTF-8 4 spaces Python 3.8 (Class)

Do the following:

Split the data into training and testing sets. Train a model using all four features.
Plot the confusion matrix.
Print the accuracy score.

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449_Fall2020 - in_class_coding.py

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

in_class_coding.py

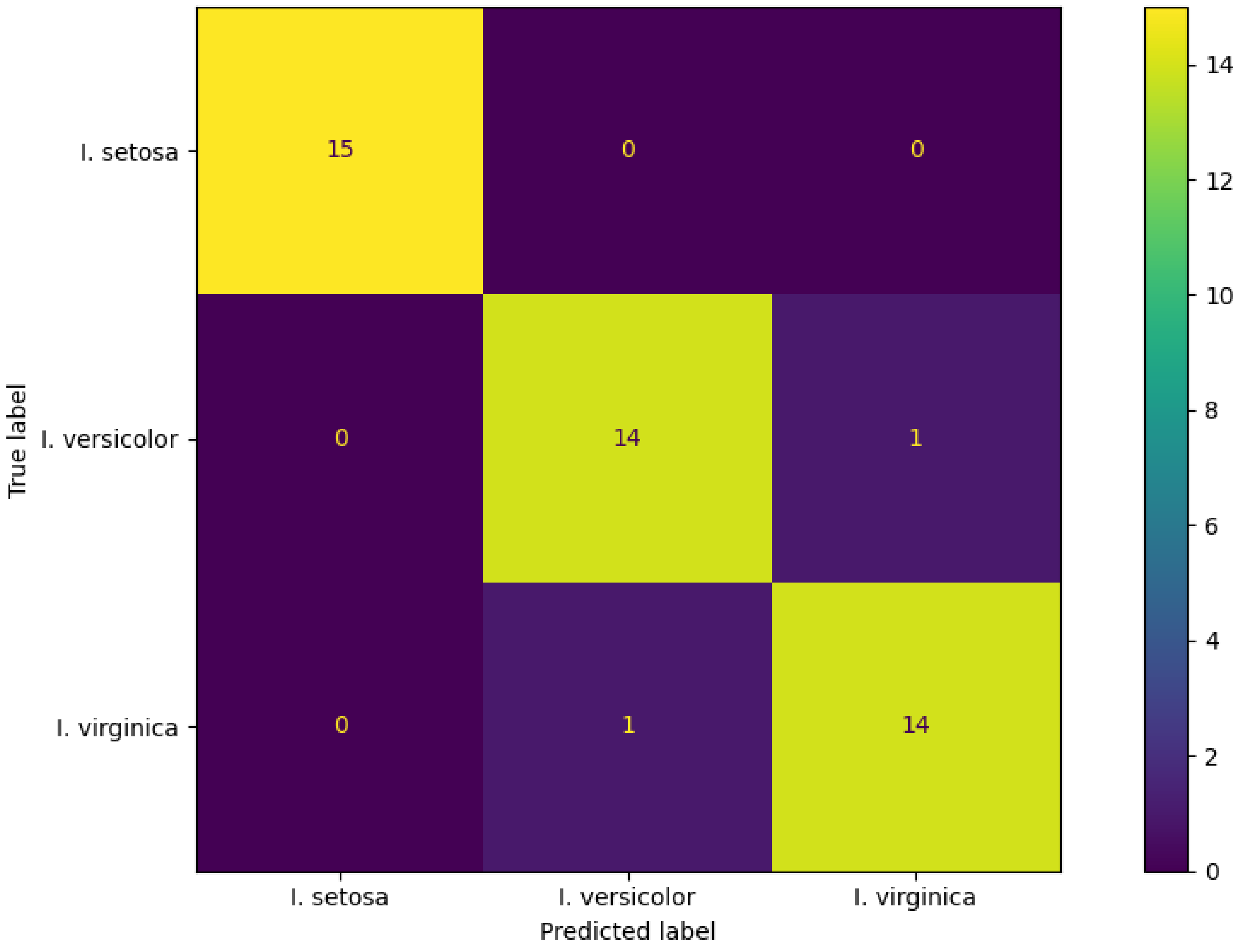
```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=2020, stratify=y)

from sklearn import svm

model = svm.SVC() # default kernel=rbf
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

from sklearn import metrics

print('Accuracy =', metrics.accuracy_score(y_test, y_pred))
metrics.plot_confusion_matrix(model, X_test, y_test)
plt.show()
```

Run: in_class_coding

```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Cod
Accuracy = 0.9555555555555556

Process finished with exit code 0
```

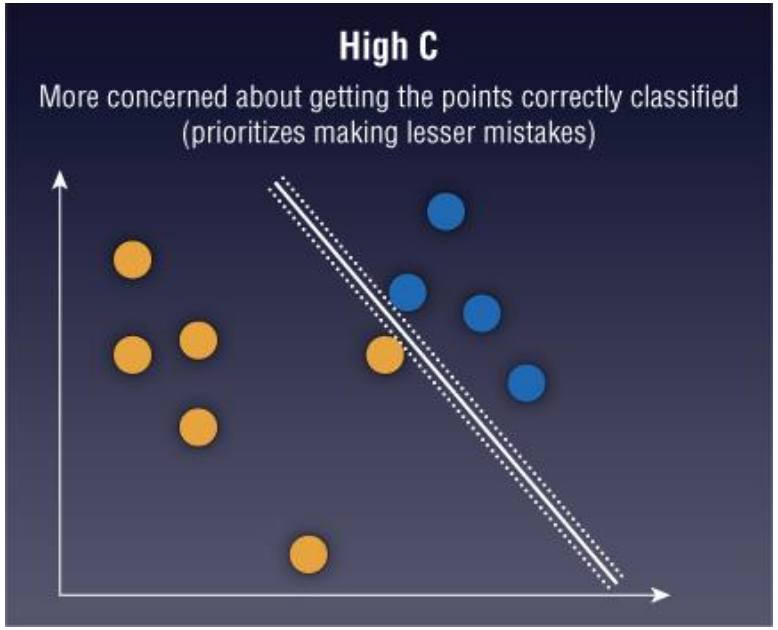
4: Run TODO 6: Problems Terminal Python Console Event Log

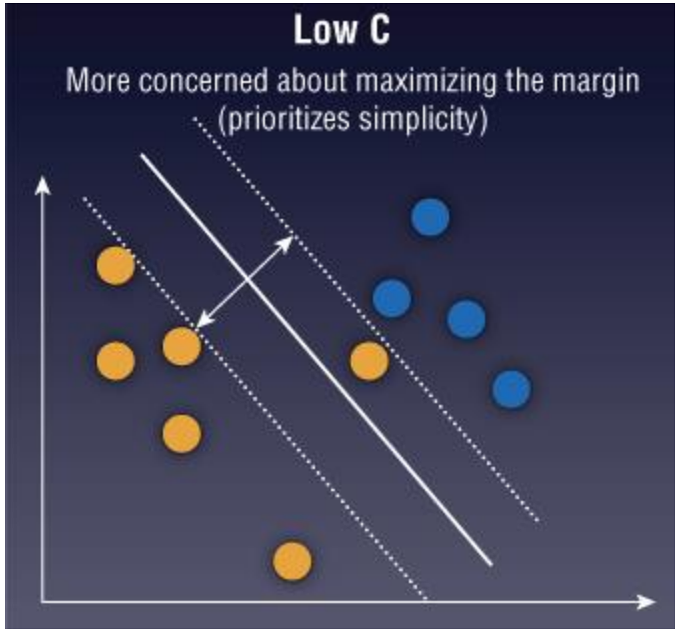
66:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


I. setosa
I. versicolor
I. virginica
True label
15
0
0
0
14
1
0
1
14
I. setosa
I. versicolor
I. virginica
Predicted label
0
2
4
6
8
10
12
14

Regularization

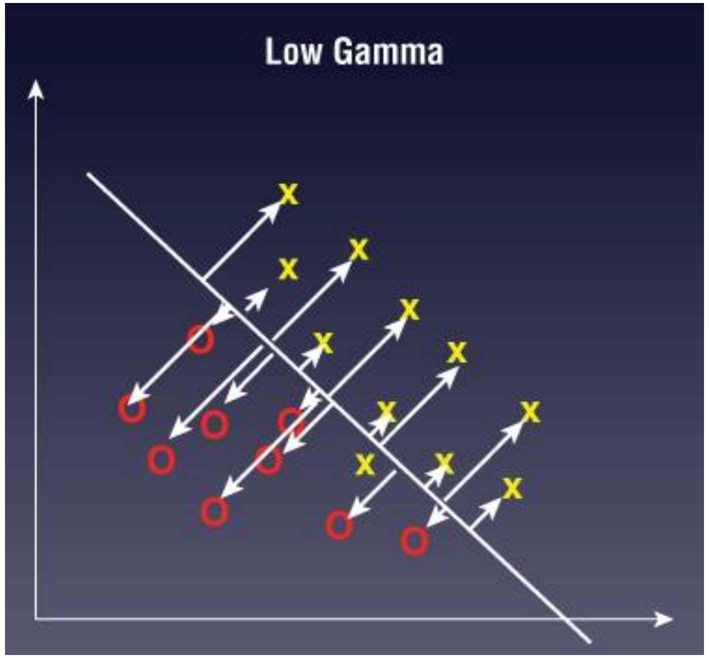
Parameter C controls the tolerance of classification error. A high C will seek to classify all points correctly, narrowing the margin. A low C will aim for the widest margin possible but will result in some points being misclassified.

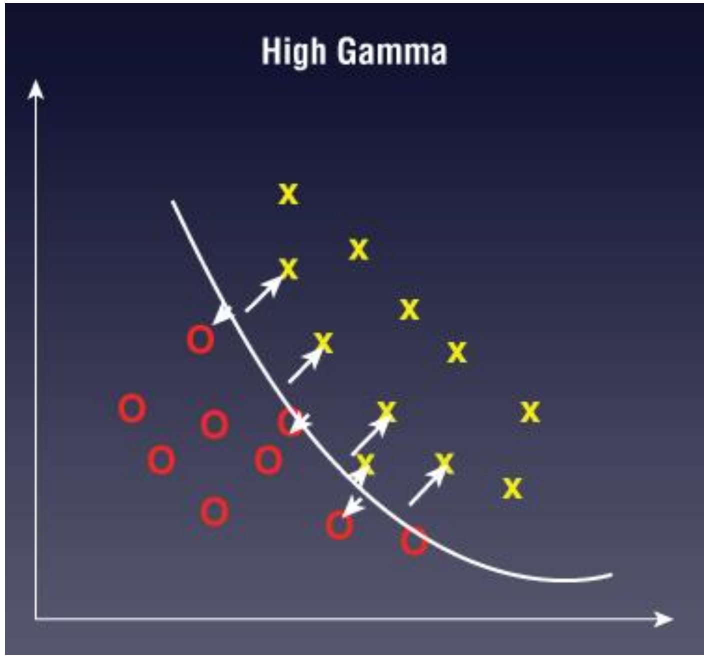




Gamma

Defines how far the influence of a single training example reaches. A low Gamma indicates that every point has a far reach while a high Gamma indicates that points closest to the decision boundary have a close reach.





Linear Kernel

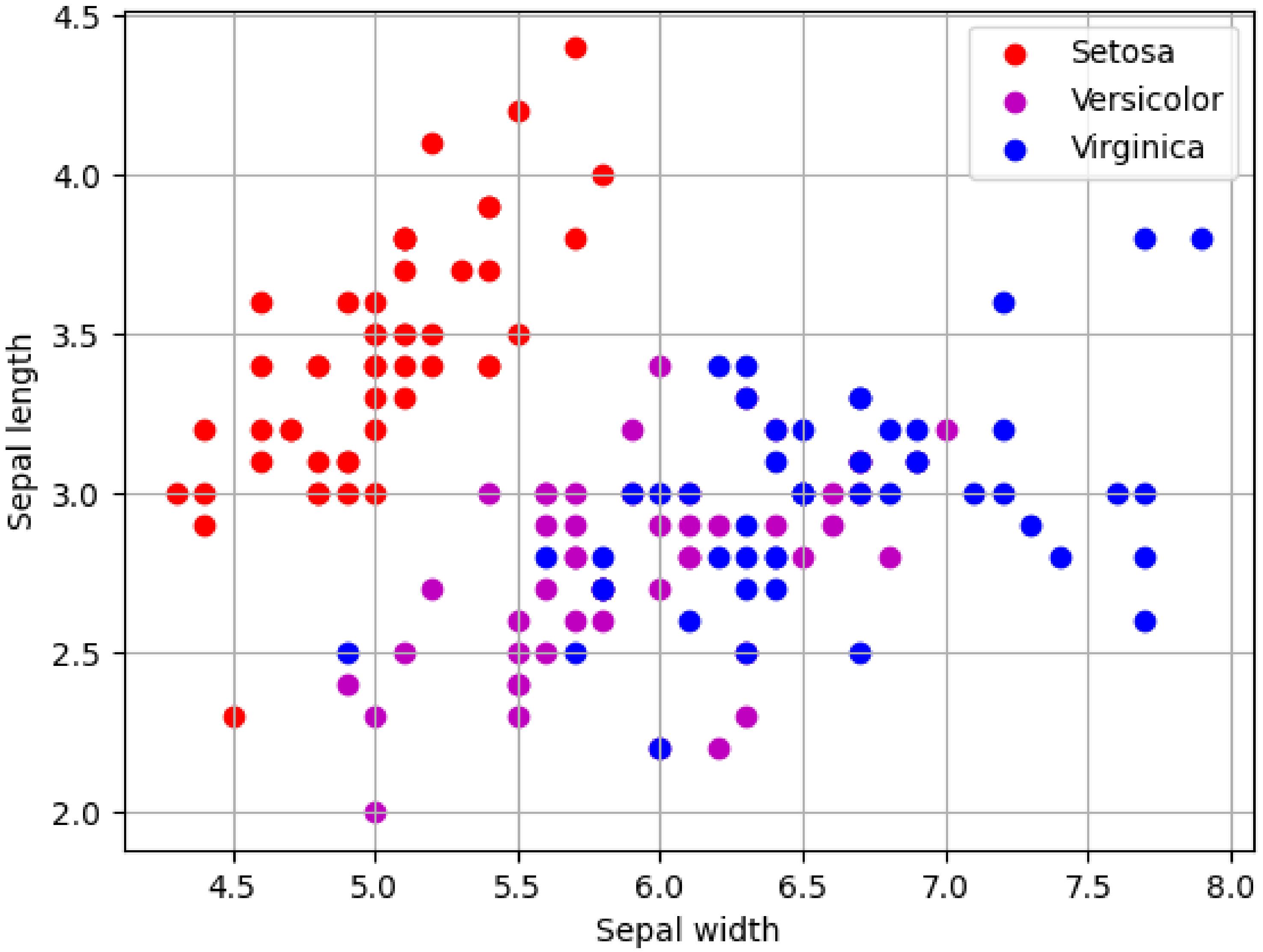
Uses a straight line to separate points for classification.

Polynomial Kernel

Uses polynomial curves to separate points for classification where a polynomial of degree 1 is essentially a linear kernel.

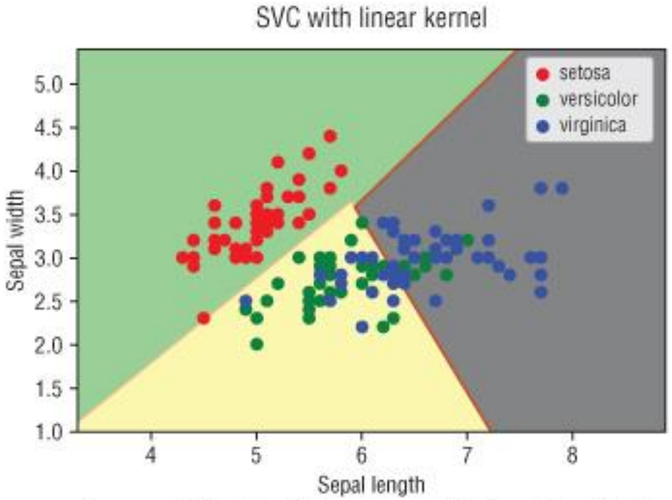
Radial Basis Function (RBF) Kernel

Gives value to each point based on its distance from the origin or a fixed center, commonly on a Euclidean space.



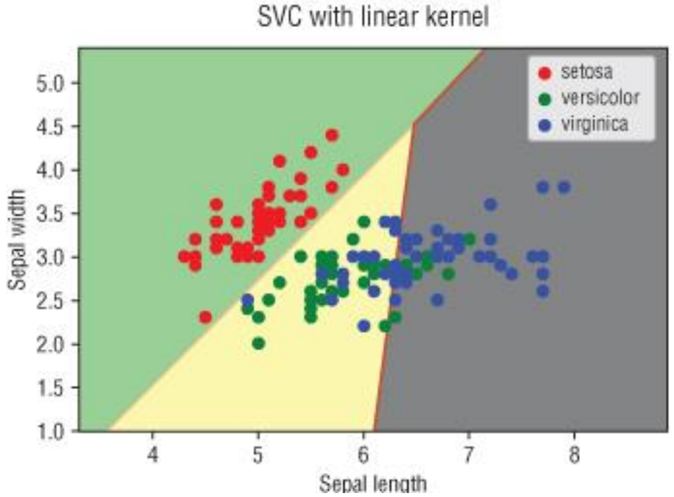
Setosa
Versicolor
Virginica
Sepal length
Sepal width
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

Default C
C=1

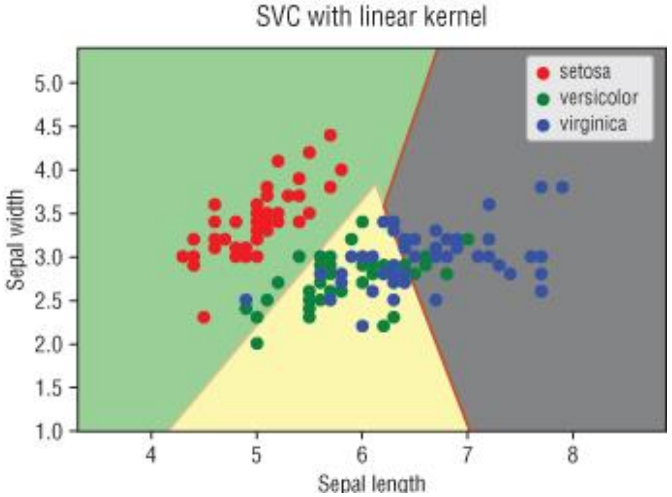


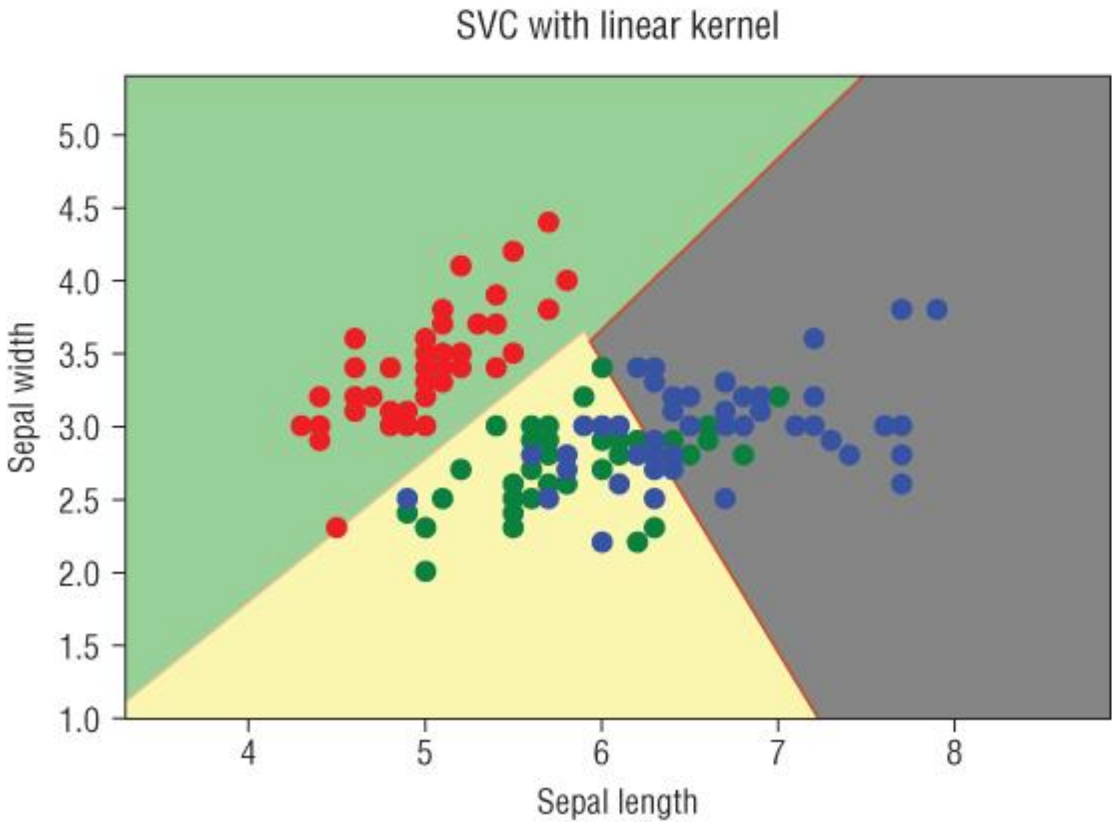

```
(array([0, 1, 2]), array([50, 53, 47]))
```

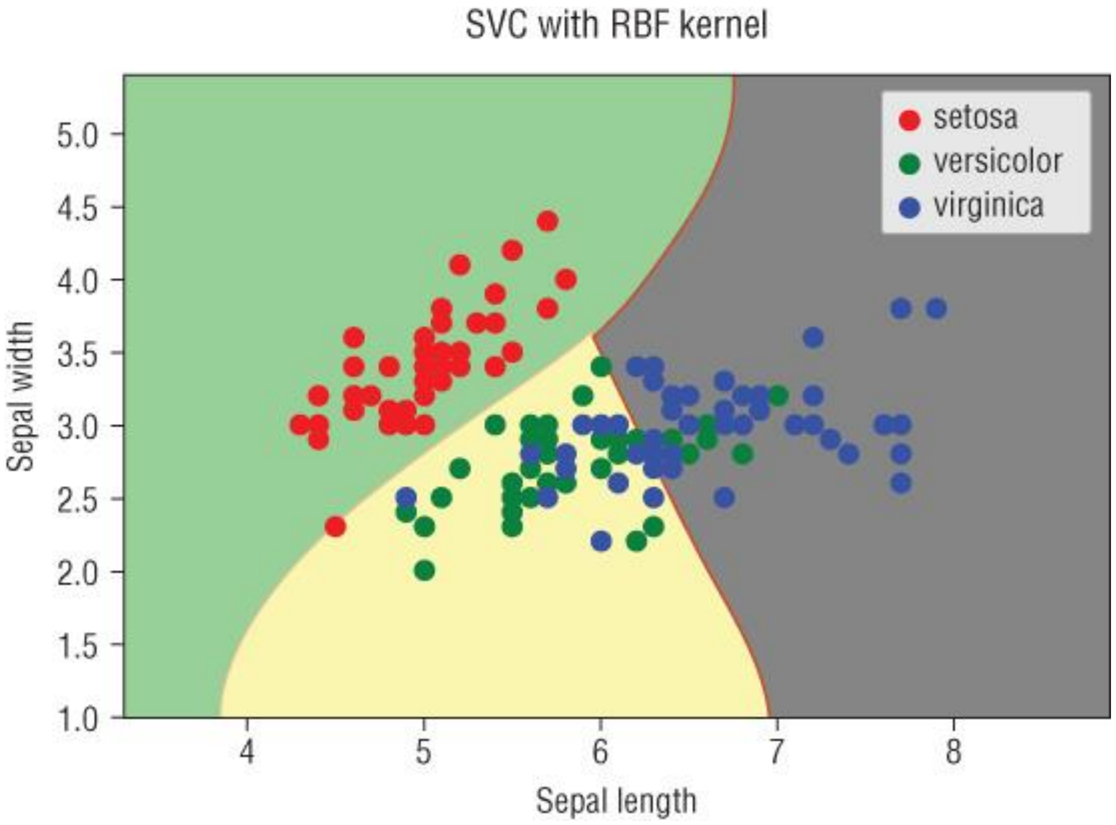
High C
$C=10^{10}$

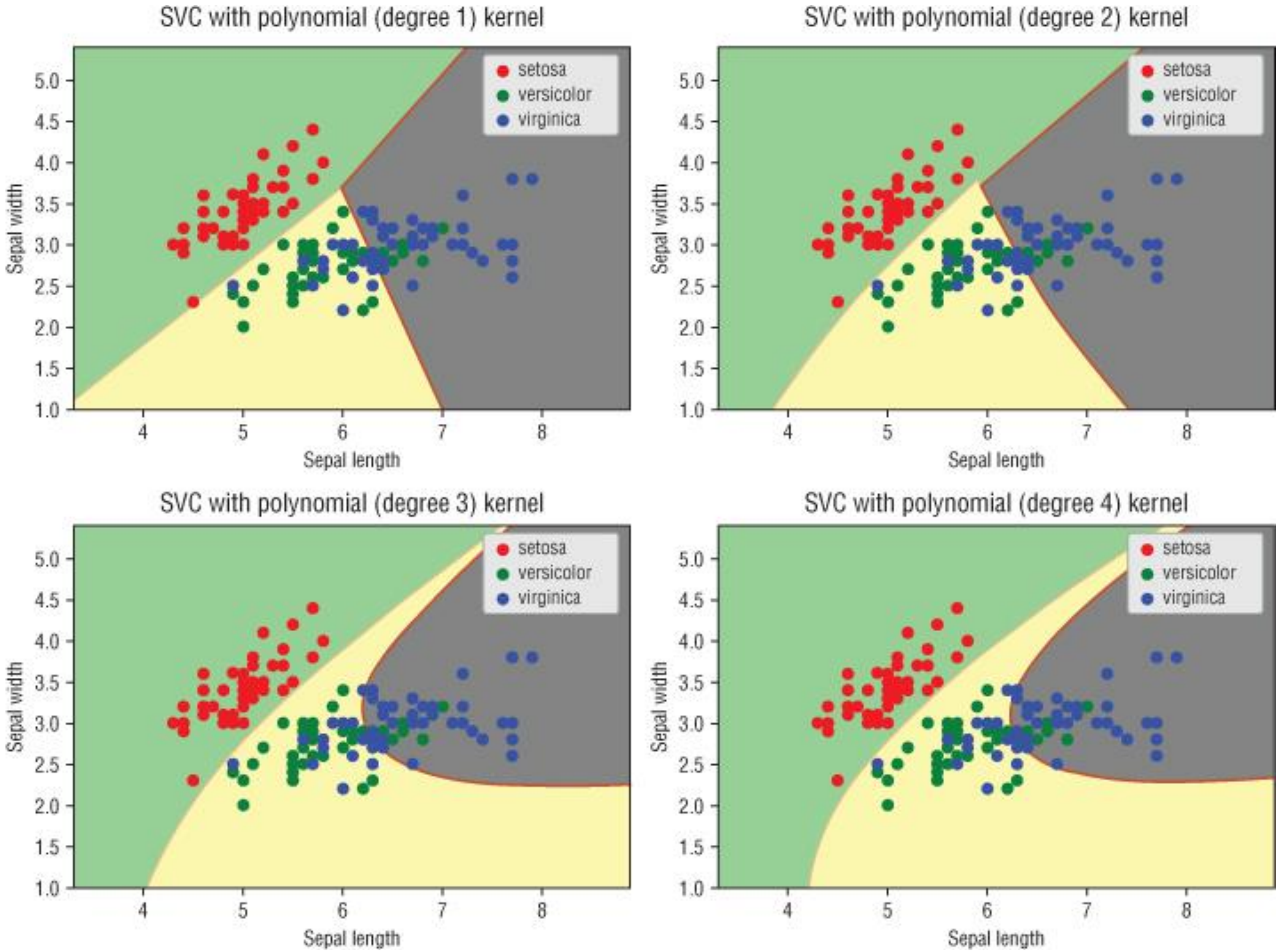


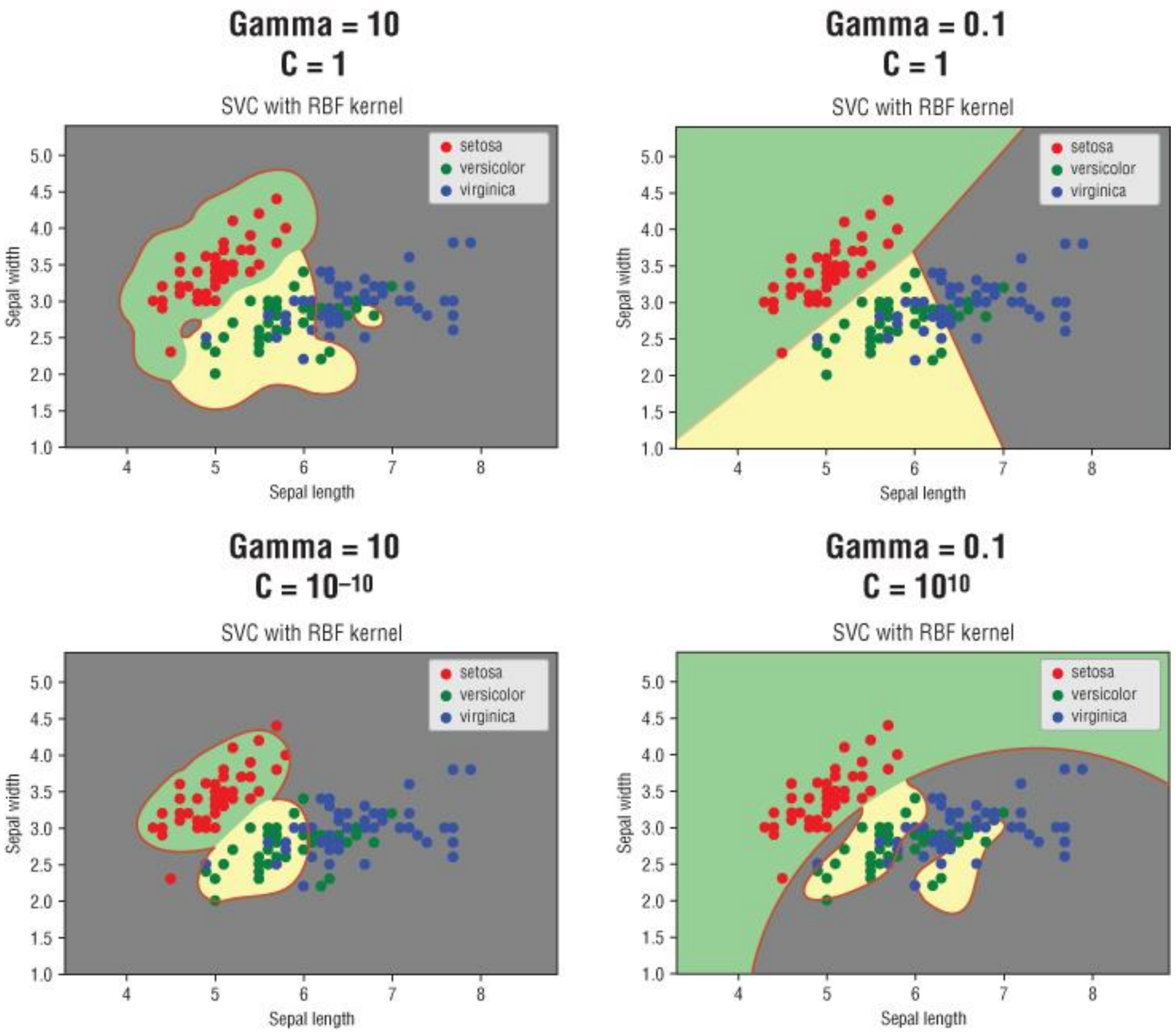
Low C
$C=10^{-10}$




SVC with linear kernel
Sepal width
Sepal length
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8


SVC with RBF kernel
setosa
versicolor
virginica
Sepal width
Sepal length
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8


SVC with polynomial (degree 1) kernel
SVC with polynomial (degree 2) kernel
SVC with polynomial (degree 3) kernel
SVC with polynomial (degree 4) kernel
setosa
versicolor
virginica
Sepal width
Sepal length
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8


Gamma = 10
C = 1
SVC with RBF kernel
setosa
versicolor
virginica
Sepal width
Sepal length
Gamma = 0.1
C = 1
SVC with RBF kernel
Gamma = 10
C = 10–10
SVC with RBF kernel
Gamma = 0.1
C = 1010
SVC with RBF kernel