#cross\_val\_score(svm**.**SVC(kernel**=**'linear',C**=**10,gamma**=**'auto'),iris**.**data, iris**.**target, cv**=**5)

#cross\_val\_score(svm**.**SVC(kernel**=**'rbf',C**=**10,gamma**=**'auto'),iris**.**data, iris**.**target, cv**=**5) for no linear, complex grupos, like a circle inside another circle

**import** pandas **as** pd

**from** sklearn.datasets **import** load\_iris

iris **=** load\_iris()

A close up of a flower

Description automatically generated with medium confidence

iris**.**feature\_names

['sepal length (cm)',

'sepal width (cm)',

'petal length (cm)',

'petal width (cm)']

iris**.**target\_names

array(['setosa', 'versicolor', 'virginica'],

dtype='<U10')

df **=** pd**.**DataFrame(iris**.**data,columns**=**iris**.**feature\_names)

df**.**head()

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** |
| --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 |

df['target'] **=** iris**.**target

df**.**head()

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** |
| --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 |

df[df**.**target**==**1]**.**head()

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** |
| --- | --- | --- | --- | --- | --- |
| **50** | 7.0 | 3.2 | 4.7 | 1.4 | 1 |
| **51** | 6.4 | 3.2 | 4.5 | 1.5 | 1 |
| **52** | 6.9 | 3.1 | 4.9 | 1.5 | 1 |
| **53** | 5.5 | 2.3 | 4.0 | 1.3 | 1 |
| **54** | 6.5 | 2.8 | 4.6 | 1.5 | 1 |

df[df**.**target**==**2]**.**head()

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** |
| --- | --- | --- | --- | --- | --- |
| **100** | 6.3 | 3.3 | 6.0 | 2.5 | 2 |
| **101** | 5.8 | 2.7 | 5.1 | 1.9 | 2 |
| **102** | 7.1 | 3.0 | 5.9 | 2.1 | 2 |
| **103** | 6.3 | 2.9 | 5.6 | 1.8 | 2 |
| **104** | 6.5 | 3.0 | 5.8 | 2.2 | 2 |

df['flower\_name'] **=**df**.**target**.**apply(**lambda** x: iris**.**target\_names[x])

df**.**head()

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **flower\_name** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 | setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 | setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 | setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 | setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 | setosa |

df[45:55]

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **flower\_name** |
| --- | --- | --- | --- | --- | --- | --- |
| **45** | 4.8 | 3.0 | 1.4 | 0.3 | 0 | setosa |
| **46** | 5.1 | 3.8 | 1.6 | 0.2 | 0 | setosa |
| **47** | 4.6 | 3.2 | 1.4 | 0.2 | 0 | setosa |
| **48** | 5.3 | 3.7 | 1.5 | 0.2 | 0 | setosa |
| **49** | 5.0 | 3.3 | 1.4 | 0.2 | 0 | setosa |
| **50** | 7.0 | 3.2 | 4.7 | 1.4 | 1 | versicolor |
| **51** | 6.4 | 3.2 | 4.5 | 1.5 | 1 | versicolor |
| **52** | 6.9 | 3.1 | 4.9 | 1.5 | 1 | versicolor |
| **53** | 5.5 | 2.3 | 4.0 | 1.3 | 1 | versicolor |
| **54** | 6.5 | 2.8 | 4.6 | 1.5 | 1 | versicolor |

df0 **=** df[:50]

df1 **=** df[50:100]

df2 **=** df[100:]

**import** matplotlib.pyplot **as** plt

**%matplotlib** inline

**Sepal length vs Sepal Width (Setosa vs Versicolor)**

plt**.**xlabel('Sepal Length')

plt**.**ylabel('Sepal Width')

plt**.**scatter(df0['sepal length (cm)'], df0['sepal width (cm)'],color**=**"green",marker**=**'+')

plt**.**scatter(df1['sepal length (cm)'], df1['sepal width (cm)'],color**=**"blue",marker**=**'.')

Chart, scatter chart

Description automatically generated

**Petal length vs Pepal Width (Setosa vs Versicolor)**

plt**.**xlabel('Petal Length')

plt**.**ylabel('Petal Width')

plt**.**scatter(df0['petal length (cm)'], df0['petal width (cm)'],color**=**"green",marker**=**'+')

plt**.**scatter(df1['petal length (cm)'], df1['petal width (cm)'],color**=**"blue",marker**=**'.')

Chart, scatter chart

Description automatically generated

**Train Using Support Vector Machine (SVM)**

**from** sklearn.model\_selection **import** train\_test\_split

X **=** df**.**drop(['target','flower\_name'], axis**=**'columns')

y **=** df**.**target

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.2)

**from** sklearn.svm **import** SVC

model **=** SVC()

model**.**fit(X\_train, y\_train)

model**.**score(X\_test, y\_test)

0.93333333333333335

model**.**predict([[4.8,3.0,1.5,0.3]])

array([0])

**Tune parameters**

**1. Regularization (C)**

97

model\_C **=** SVC(C**=**1)

model\_C**.**fit(X\_train, y\_train)

model\_C**.**score(X\_test, y\_test)

97

0.933

106

model\_C **=** SVC(C**=**10)

model\_C**.**fit(X\_train, y\_train)

model\_C**.**score(X\_test, y\_test)

106

0.966

**2. Gamma**

103

model\_g **=** SVC(gamma**=**10)

model\_g**.**fit(X\_train, y\_train)

model\_g**.**score(X\_test, y\_test)

103

0.90

**3. Kernel**

104

model\_linear\_kernal **=** SVC(kernel**=**'linear')

model\_linear\_kernal**.**fit(X\_train, y\_train)

model\_linear\_kernal**.**score(X\_test, y\_test)

105

0.966