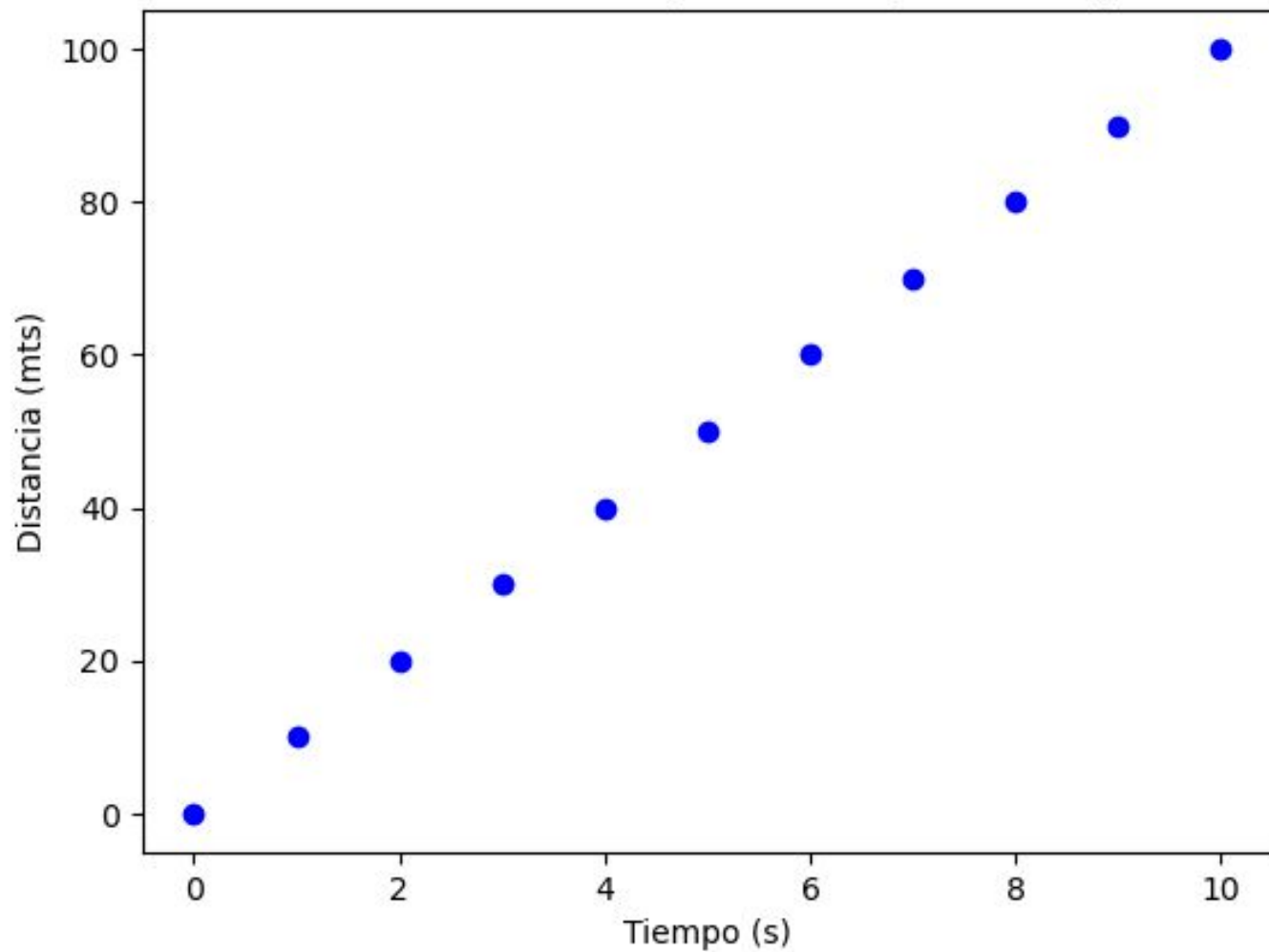


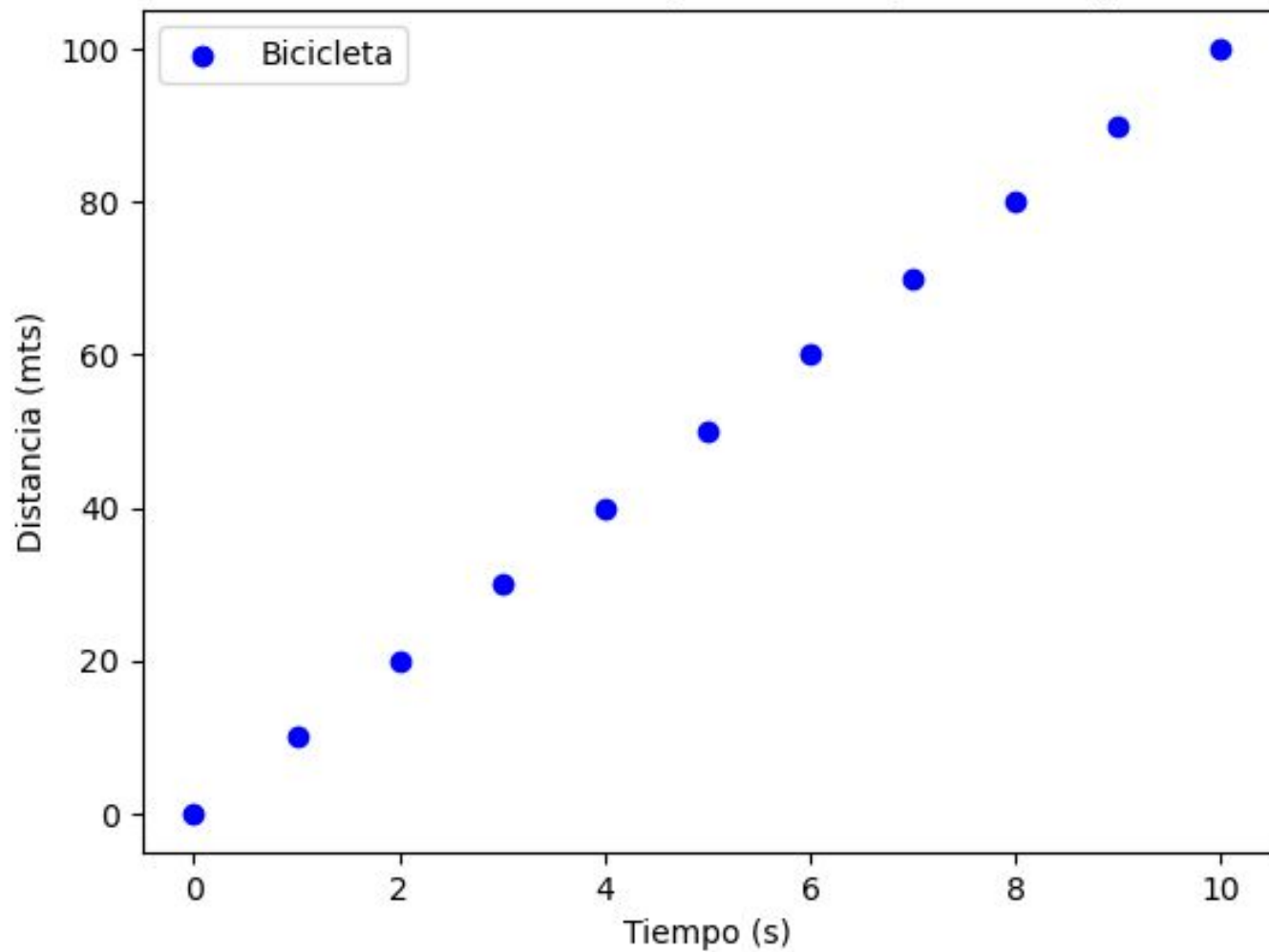


KNN

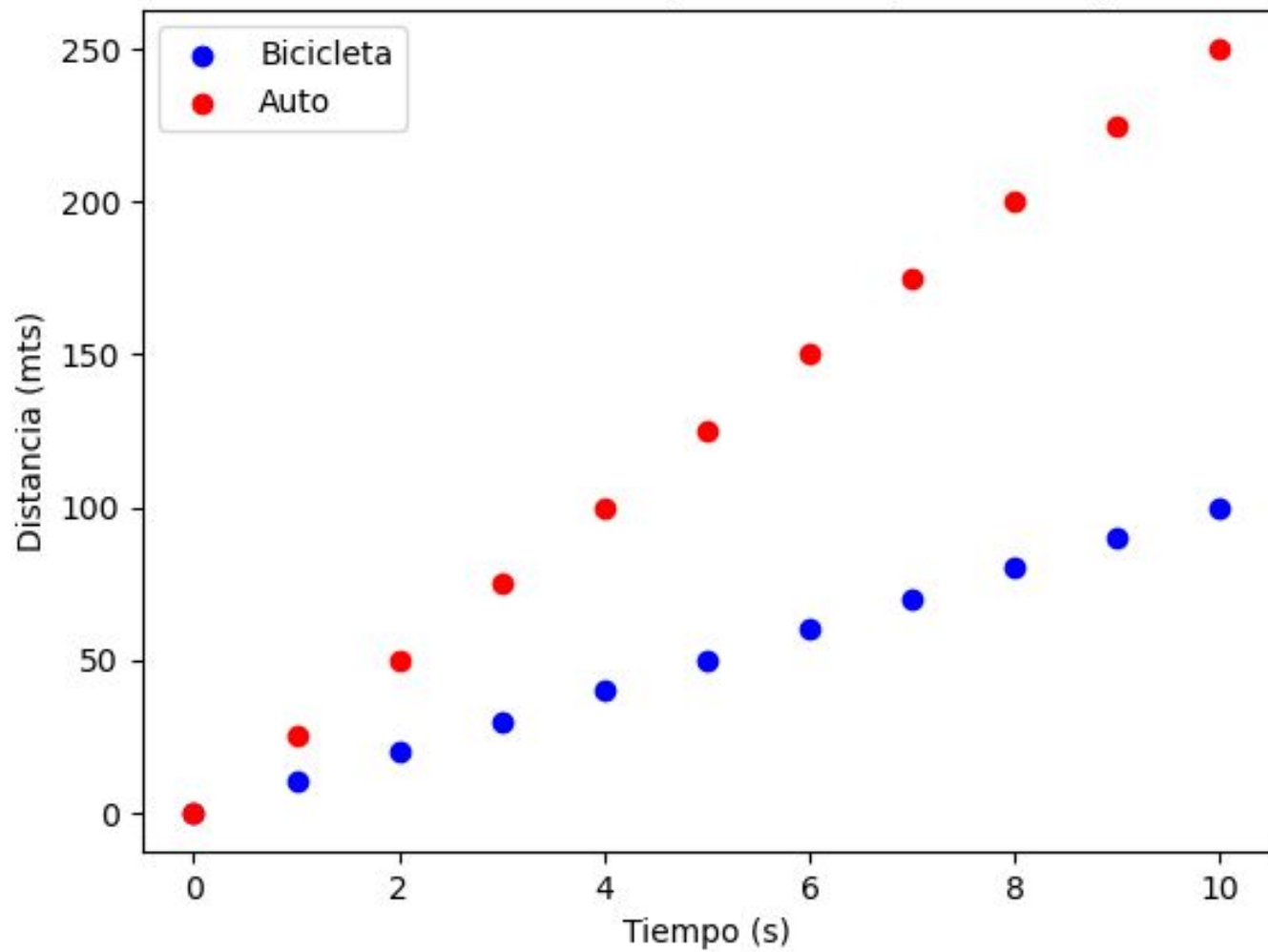
Movimiento de un objeto a lo largo del tiempo



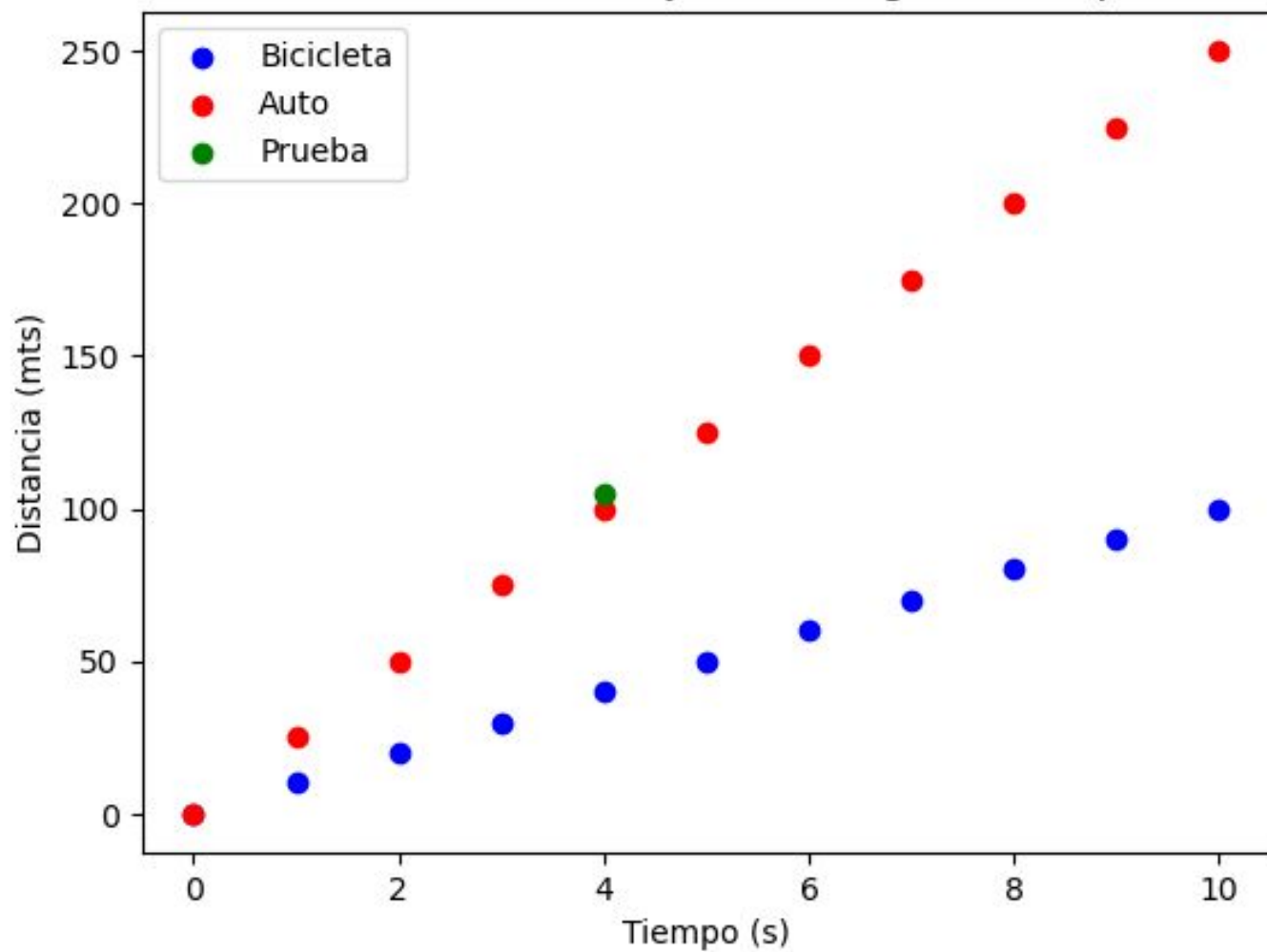
Movimiento de un objeto a lo largo del tiempo



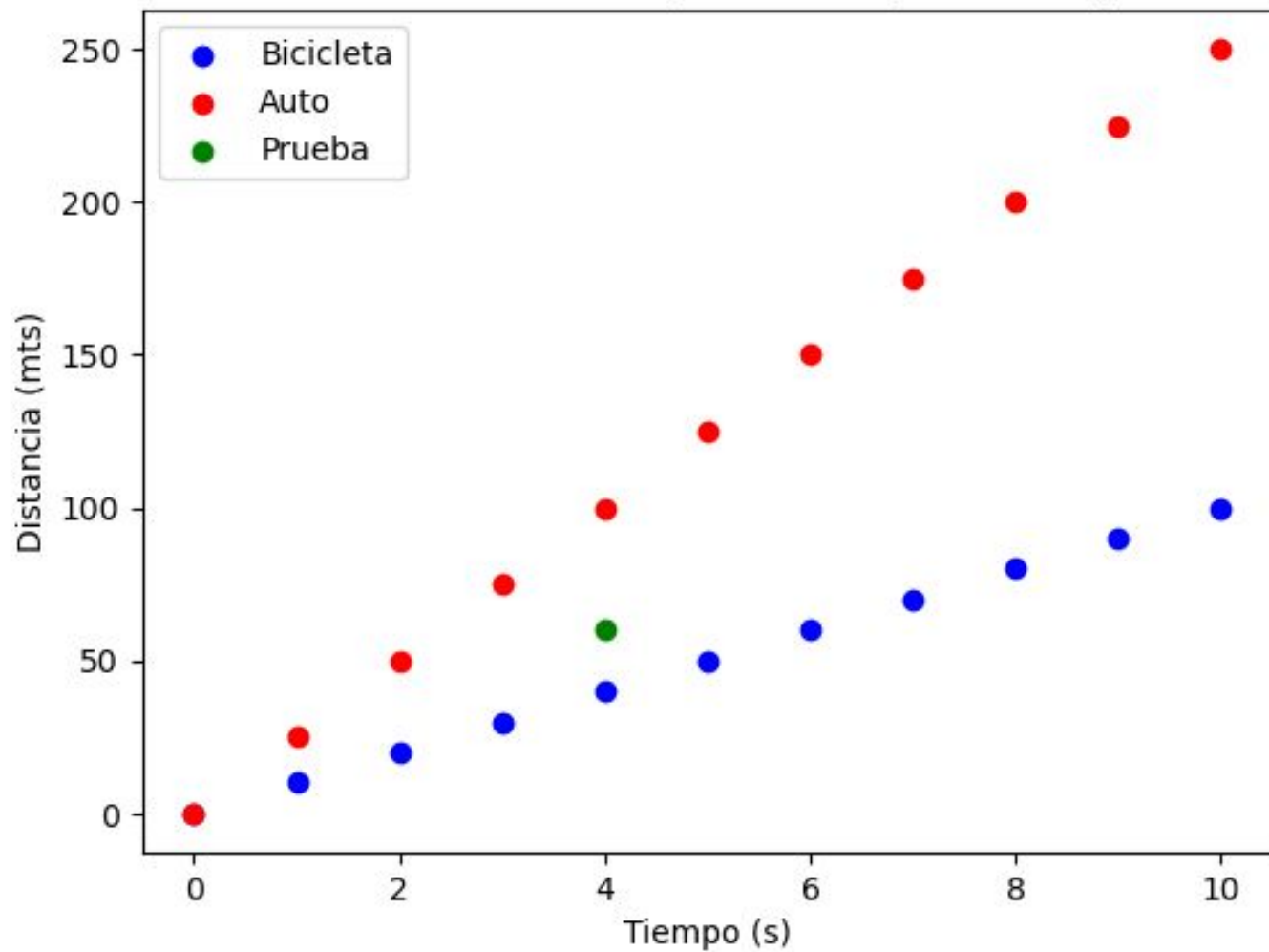
Movimiento de un objeto a lo largo del tiempo



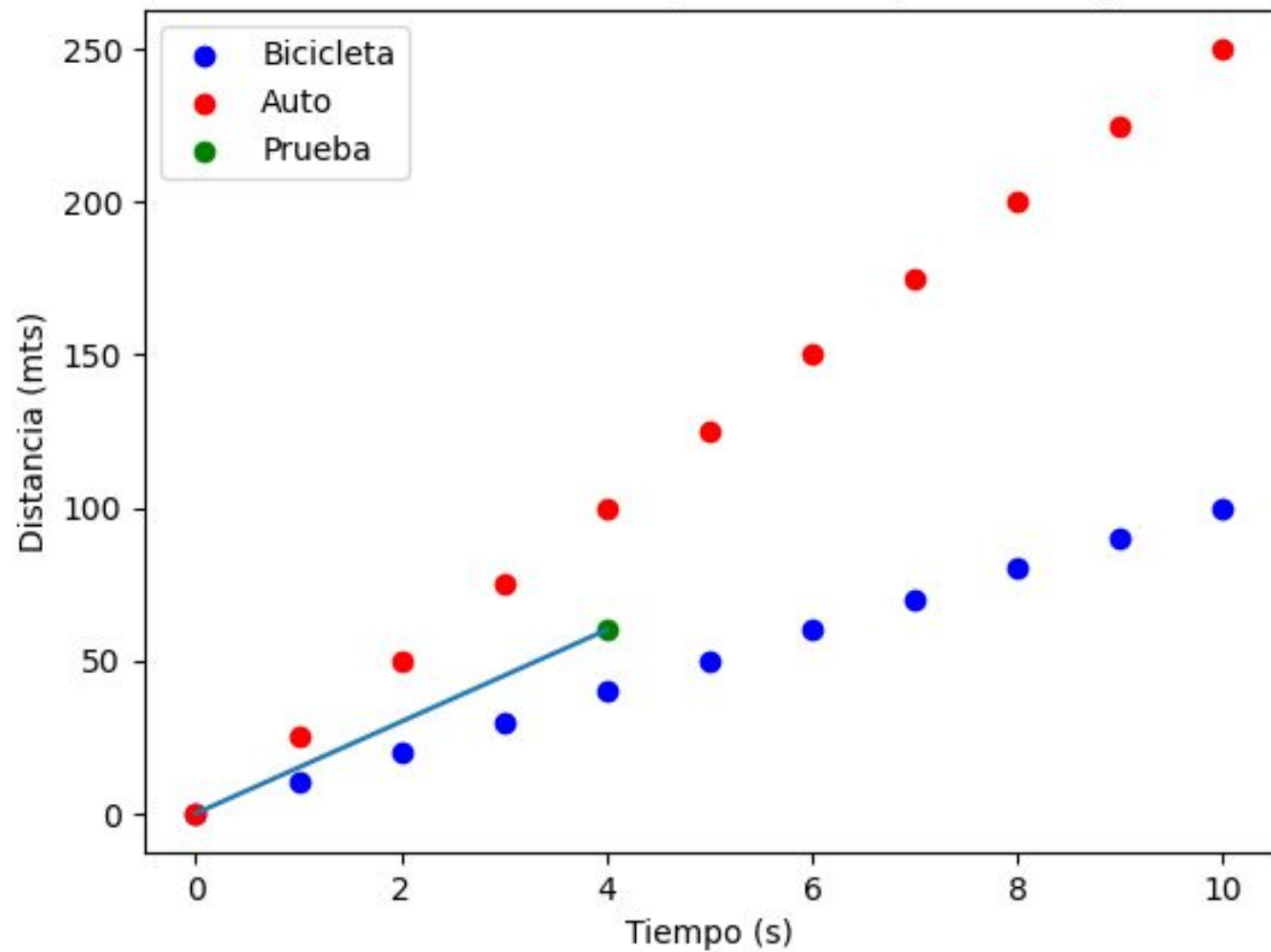
Movimiento de un objeto a lo largo del tiempo



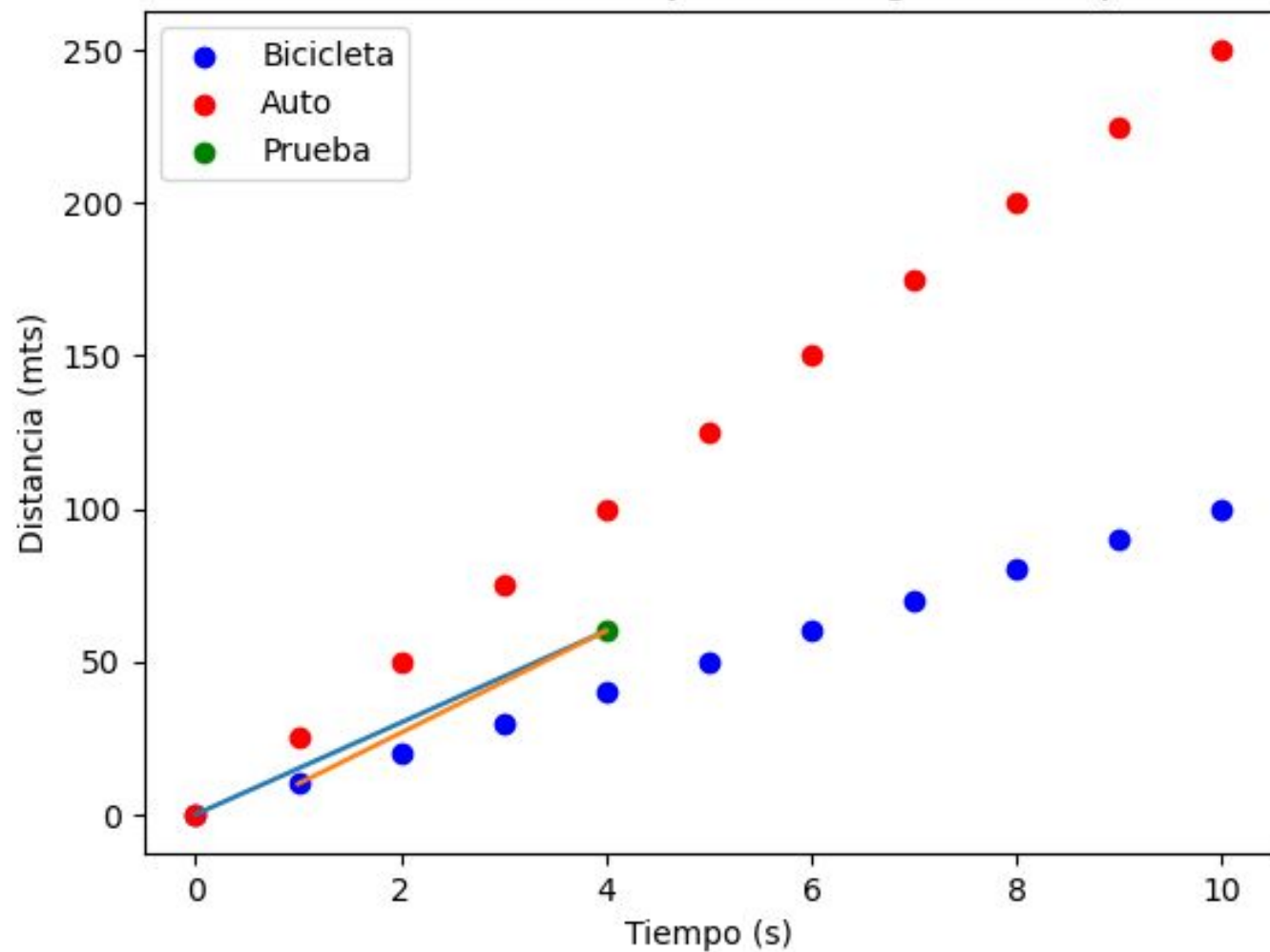
Movimiento de un objeto a lo largo del tiempo



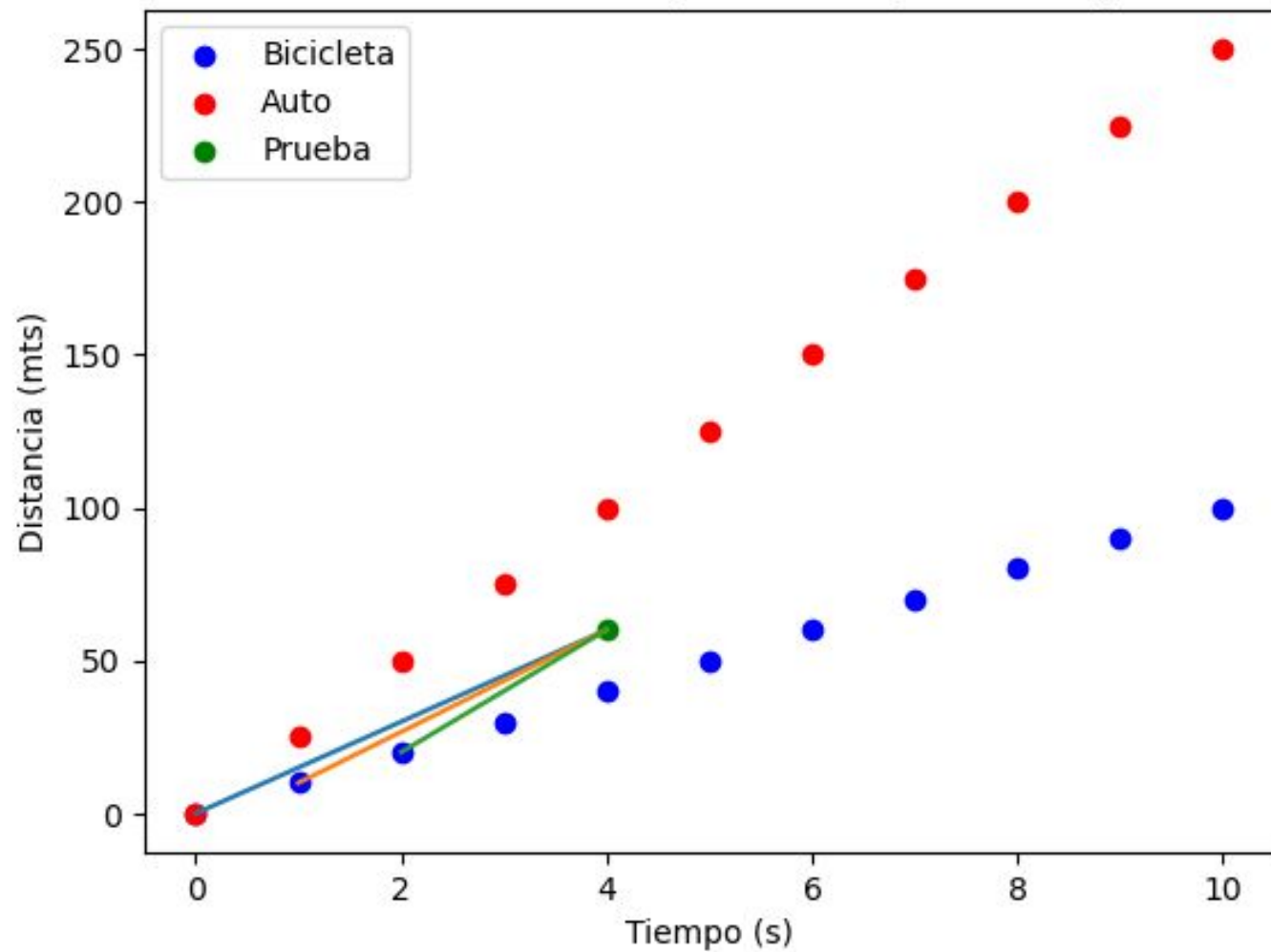
Movimiento de un objeto a lo largo del tiempo



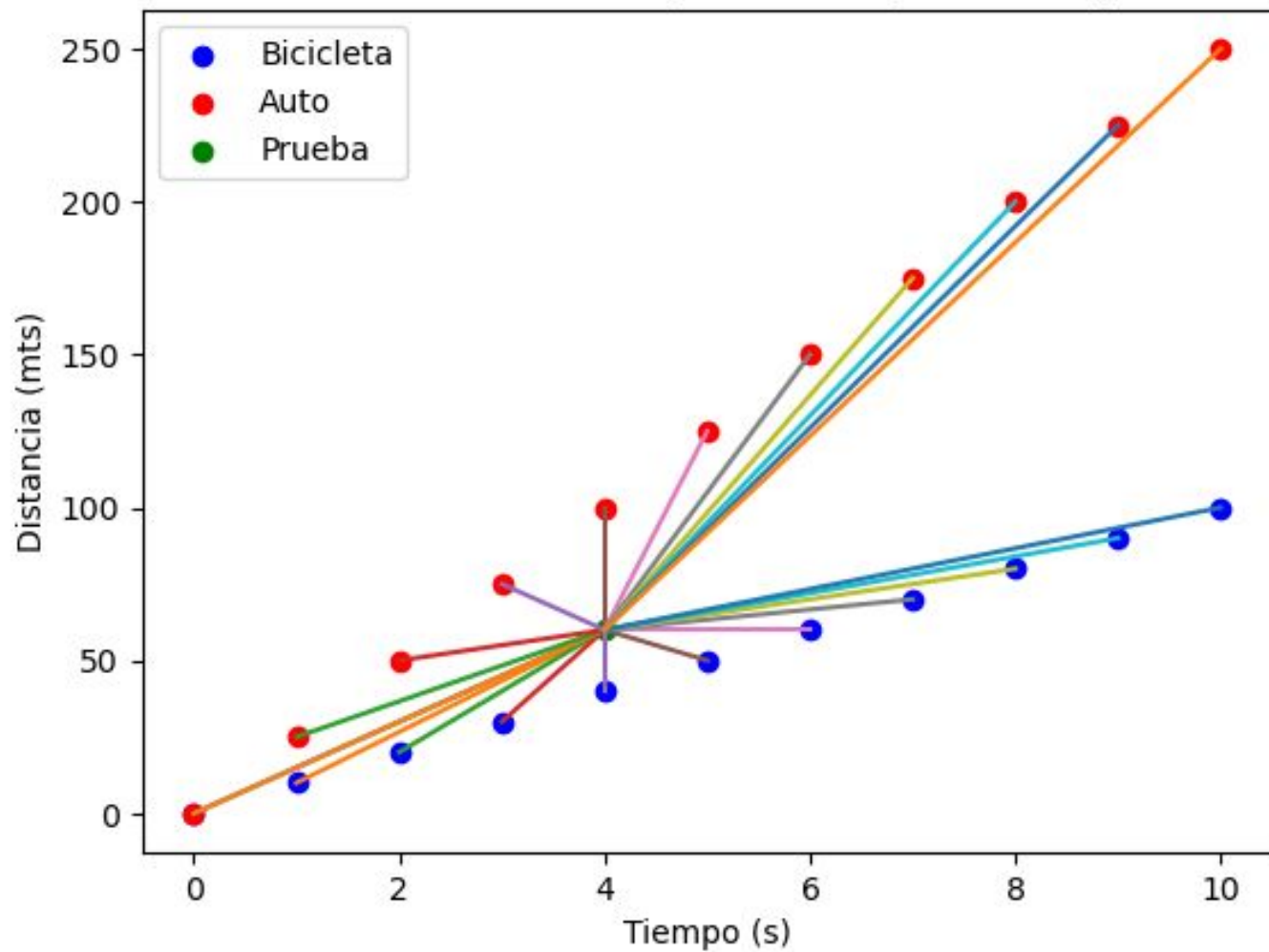
Movimiento de un objeto a lo largo del tiempo



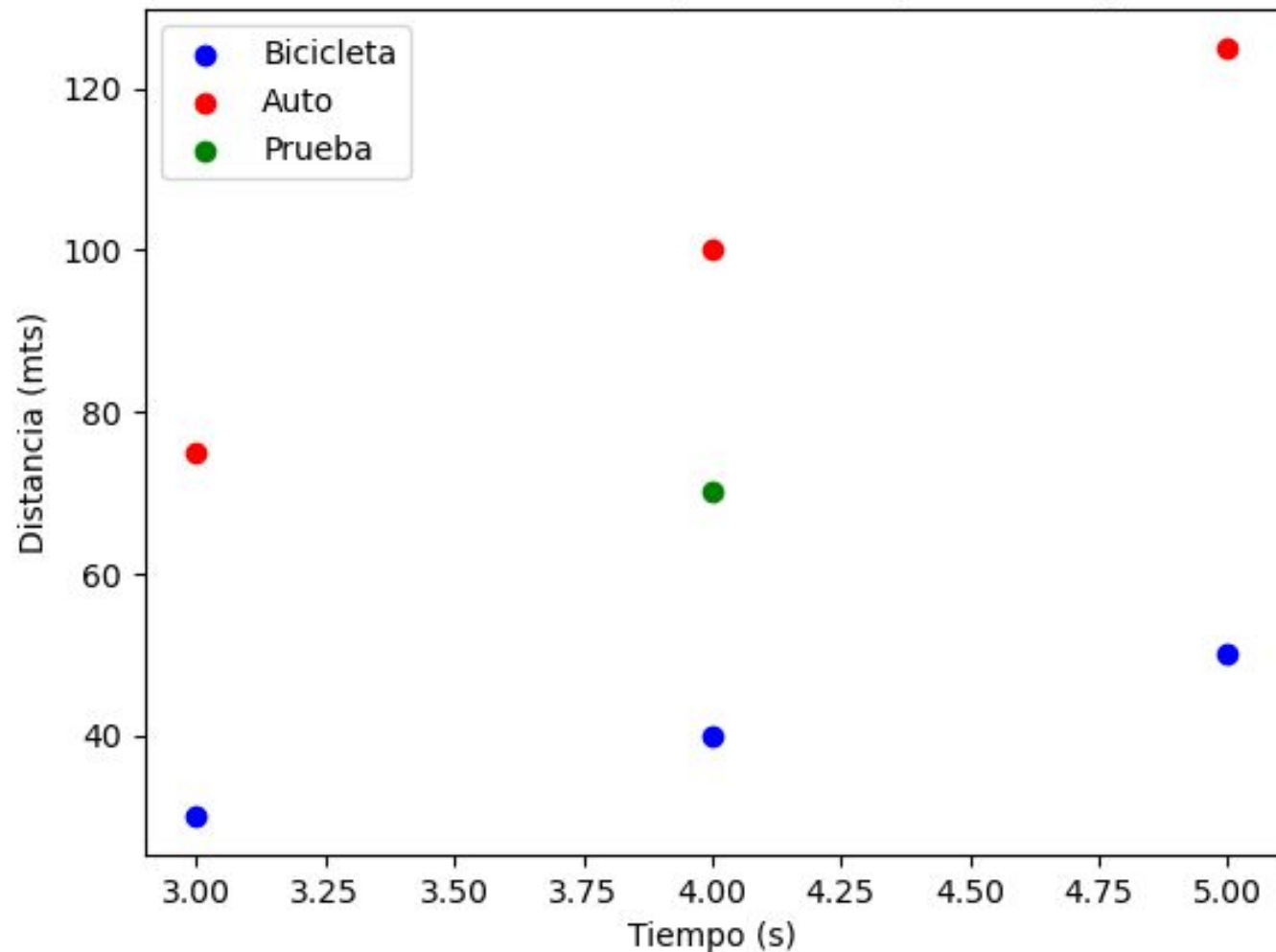
Movimiento de un objeto a lo largo del tiempo



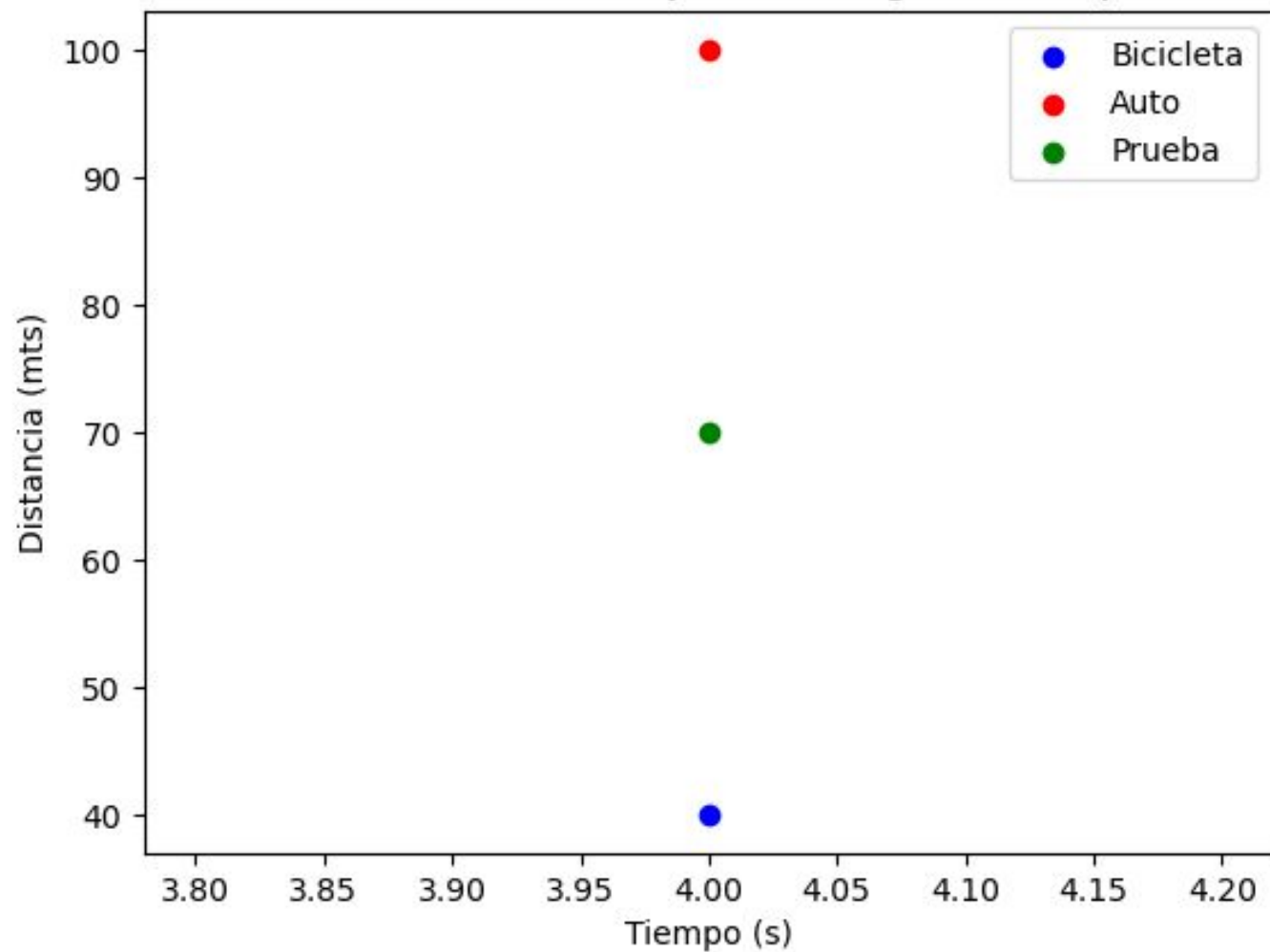
Movimiento de un objeto a lo largo del tiempo



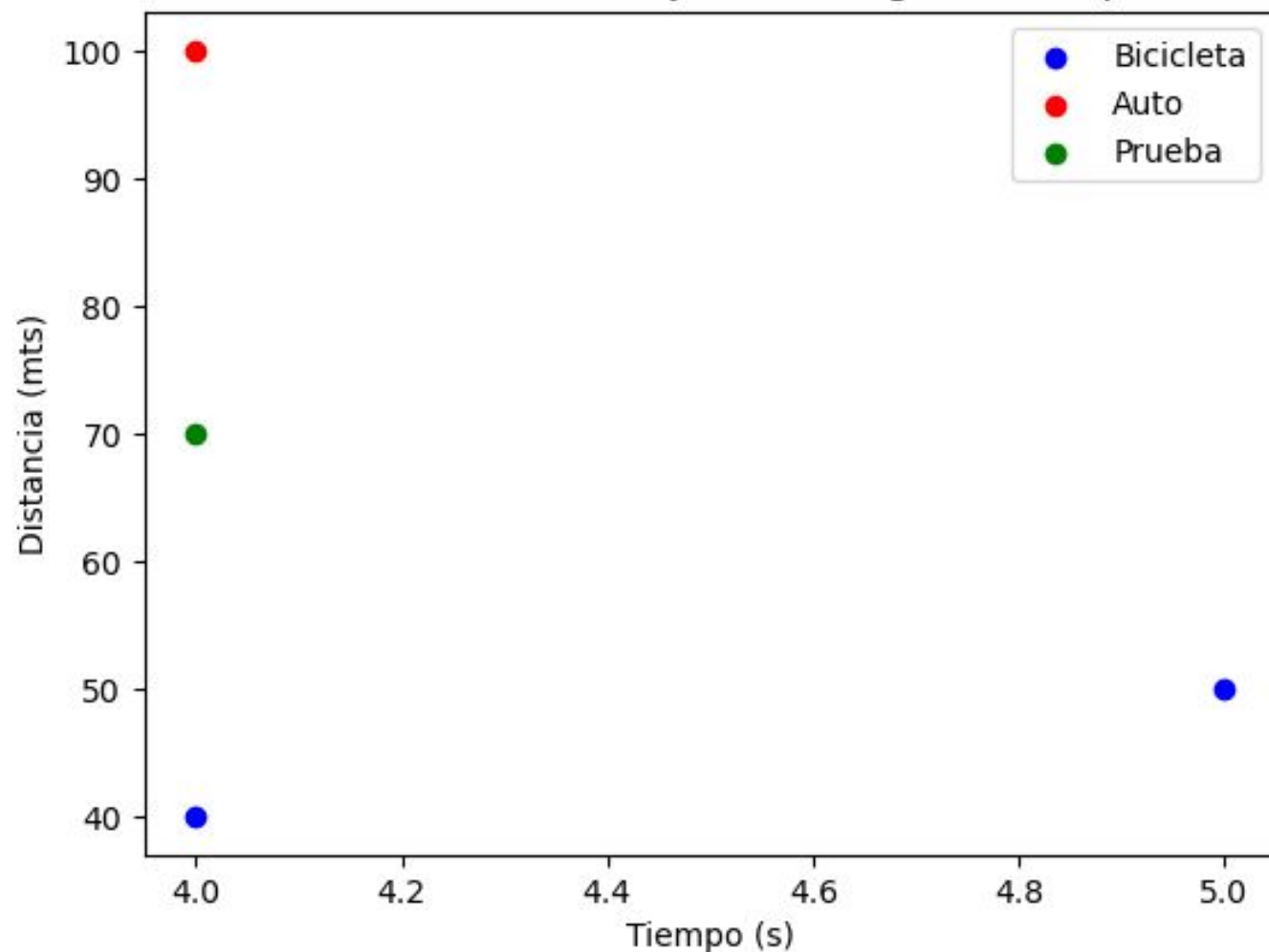
Movimiento de un objeto a lo largo del tiempo

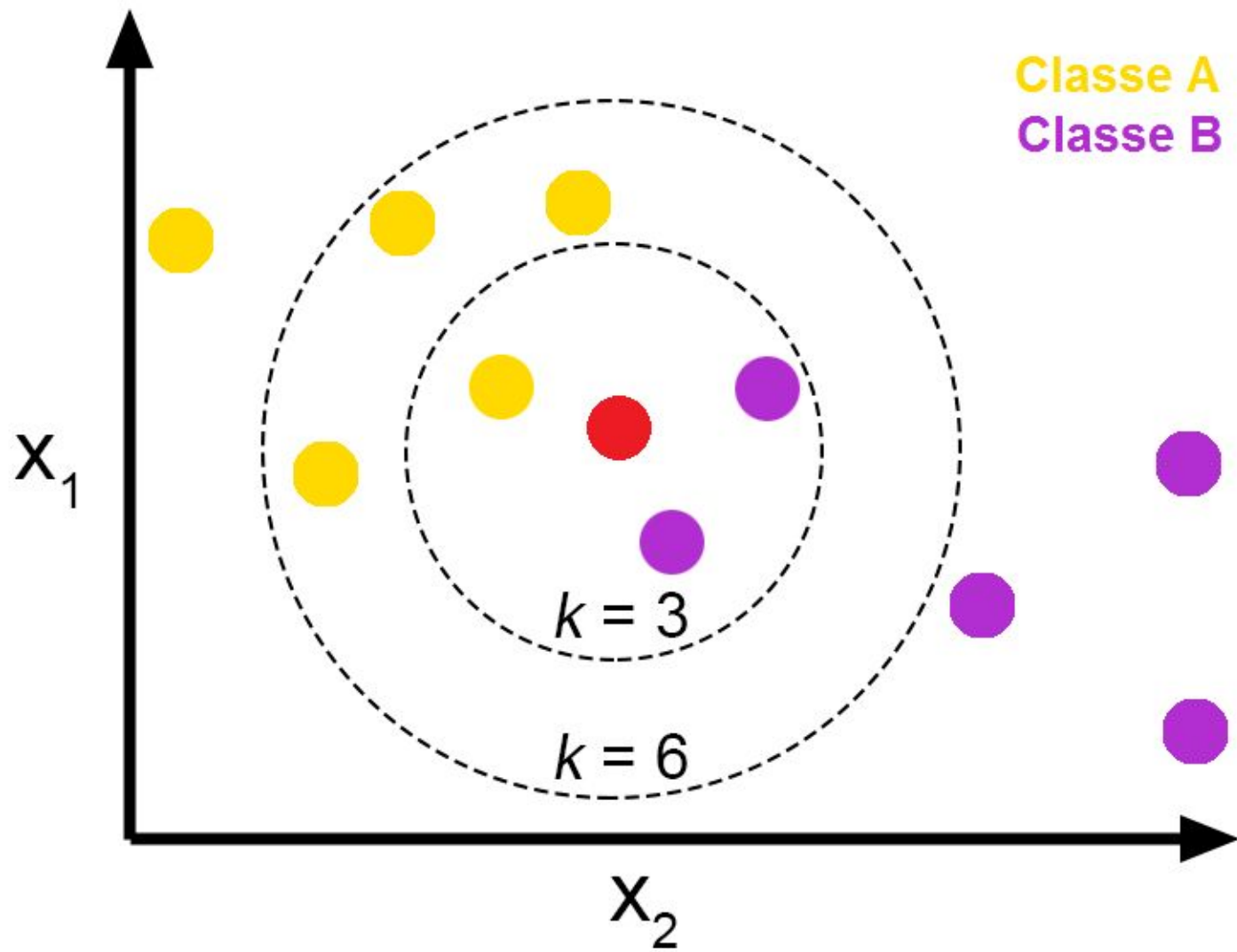


Movimiento de un objeto a lo largo del tiempo



Movimiento de un objeto a lo largo del tiempo





Métricas de distancia : Función de distancia : Distancia

Métricas de distancia : Función de distancia : Distancia

Distancia Euclidiana

Métricas de distancia : Función de distancia : Distancia

Distancia Euclidiana

$$\mathbf{x} = (x_1, x_2, \dots, x_d)^T$$

$$\mathbf{y} = (y_1, y_2, \dots, y_d)^T$$

Métricas de distancia : Función de distancia : Distancia

Distancia Euclidiana

$$\mathbf{x} = (x_1, x_2, \dots, x_d)^T$$

$$\mathbf{y} = (y_1, y_2, \dots, y_d)^T$$

$$\begin{array}{|c|} \hline x_1 \\ \hline x_2 \\ \hline \vdots \\ \hline x_d \\ \hline \end{array} \quad \begin{array}{|c|} \hline y_1 \\ \hline y_2 \\ \hline \vdots \\ \hline y_d \\ \hline \end{array}$$

Métricas de distancia : Función de distancia : Distancia

Distancia Euclidiana

$$\mathbf{x} = (x_1, x_2, \dots, x_d)^T$$

$$\mathbf{y} = (y_1, y_2, \dots, y_d)^T$$

$$\begin{array}{|c|} \hline x_1 \\ \hline x_2 \\ \hline \vdots \\ \hline x_d \\ \hline \end{array} \quad \begin{array}{|c|} \hline y_1 \\ \hline y_2 \\ \hline \vdots \\ \hline y_d \\ \hline \end{array}$$

$$\sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

$$\mathbf{x} = (2, 3)$$

$$\mathbf{y} = (4, 6)$$

$$\mathbf{x} = (2, 3)^{\top}$$

$$\mathbf{y} = (4, 6)^{\top}$$

$$\mathbf{x} = (2, 3)^{\top}$$

$$\mathbf{y} = (4, 6)^{\top}$$

$$\mathbf{x} = \begin{vmatrix} 2 \\ 2 \end{vmatrix}$$

$$\mathbf{y} = \begin{vmatrix} 4 \\ 6 \end{vmatrix}$$

$$\mathbf{x} = (2, 3)^{\top}$$

$$\mathbf{y} = (4, 6)^{\top}$$

$$\mathbf{x} = \begin{vmatrix} 2 \\ 2 \end{vmatrix}$$

$$\mathbf{y} = \begin{vmatrix} 4 \\ 6 \end{vmatrix}$$

$$d(x, y) = \sqrt{(2 - 4)^2 + (2 - 6)^2}$$

$$\mathbf{x} = (2, 3)^{\top}$$

$$\mathbf{y} = (4, 6)^{\top}$$

$$\mathbf{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

$$d(x, y) = \sqrt{(2 - 4)^2 + (2 - 6)^2}$$

$$d(x, y) = \sqrt{20}$$

Iris - DataSet



Iris - setosa



Iris - versicolor



Iris - virginica

Datos del DataSet: ['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']

Clases: ['setosa' 'versicolor' 'virginica']

Características: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']

Datos del DataSet: `['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']`

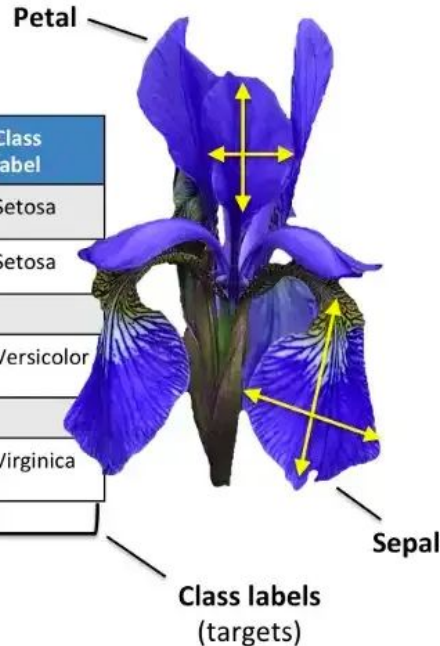
Clases: `['setosa' 'versicolor' 'virginica']`

Características: `['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']`

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)



```
from sklearn.datasets import load_iris
```

Datos del DataSet: `['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']`

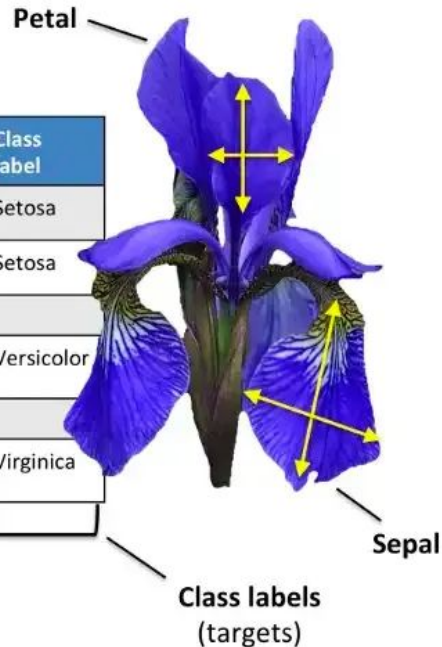
Clases: `['setosa' 'versicolor' 'virginica']`

Características: `['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']`

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)



```
from sklearn.datasets import load_iris
```

```
iris = load_iris()
```

Datos del DataSet: `['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']`

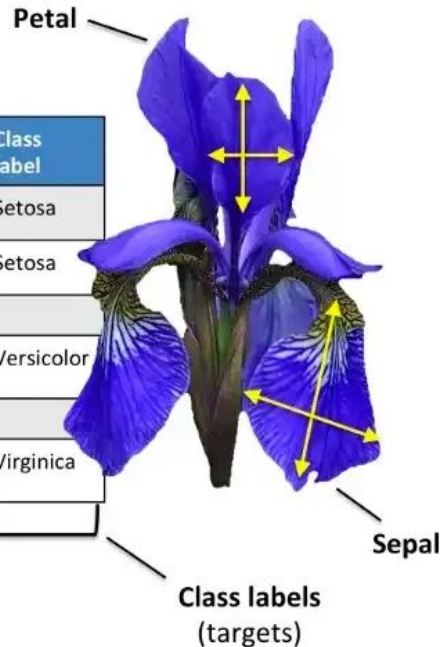
Clases: `['setosa' 'versicolor' 'virginica']`

Características: `['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']`

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)



```
from sklearn.datasets import load_iris
```

```
iris = load_iris()  
x_data = iris.data
```


Datos del DataSet: `['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']`

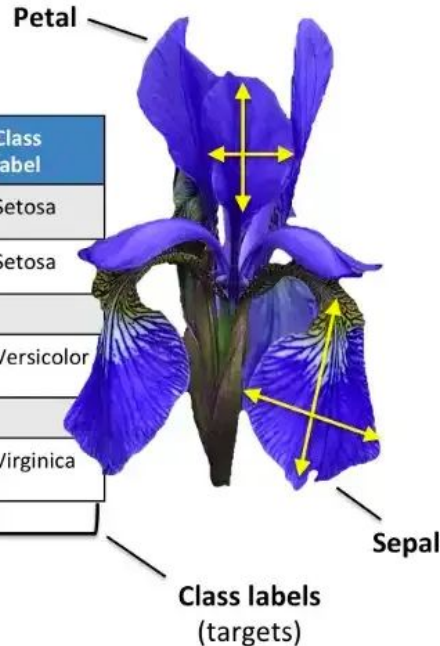
Clases: `['setosa' 'versicolor' 'virginica']`

Características: `['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']`

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)



```
from sklearn.datasets import load_iris
```

```
iris = load_iris()  
x_data = iris.data  
y_lbl = iris.target
```

Datos del DataSet: ['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']

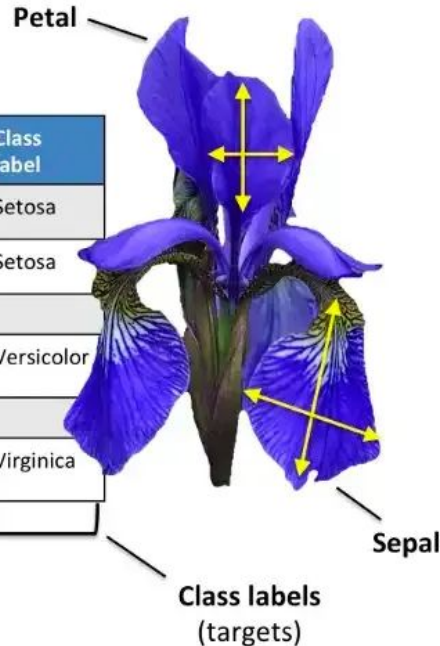
Clases: ['setosa' 'versicolor' 'virginica']

Características: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)



```
from sklearn.datasets import load_iris
```

```
iris = load_iris()  
x_data = iris.data  
y_lbl = iris.target  
classes = iris.target_names
```

Datos del DataSet: `['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']`

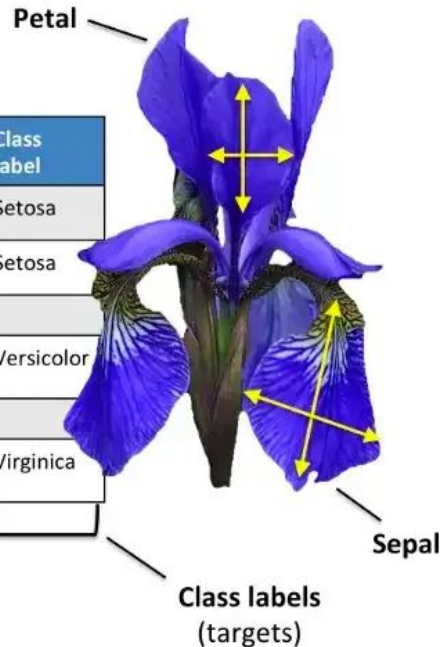
Clases: `['setosa' 'versicolor' 'virginica']`

Características: `['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']`

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)



```
from sklearn.datasets import load_iris
```

```
iris = load_iris()
x_data = iris.data
y_lbl = iris.target
classes = iris.target_names
characteristics = iris.feature_names
```


Datos del DataSet: `['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']`

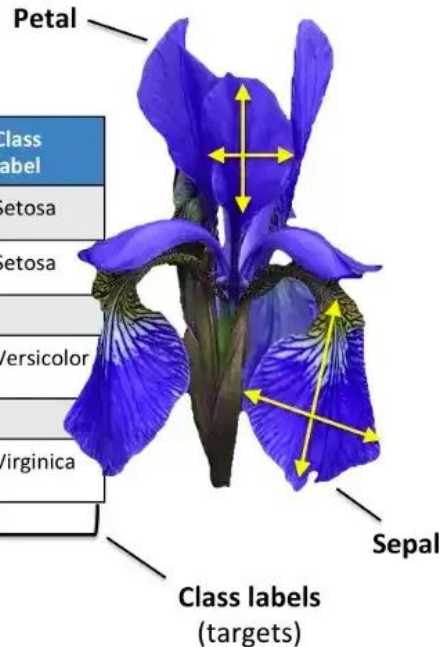
Clases: `['setosa' 'versicolor' 'virginica']`

Características: `['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']`

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)



```
from sklearn.datasets import load_iris
```

```
iris = load_iris()  
x_data = iris.data  
y_lbl = iris.target  
classes = iris.target_names  
characteristics = iris.feature_names
```

0 1 2
`['setosa' 'versicolor' 'virginica']`

Datos del DataSet: `['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module']`

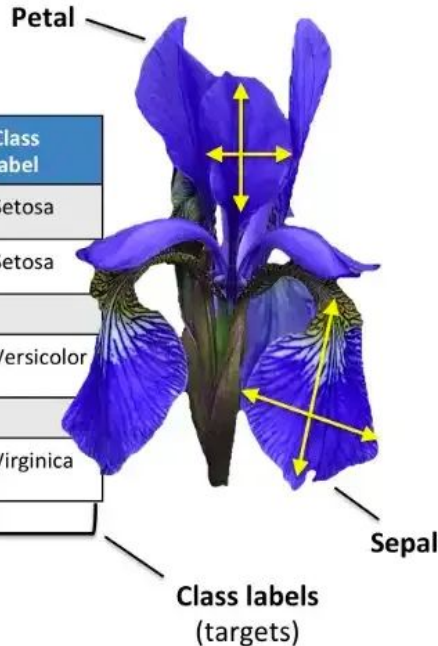
Clases: `['setosa' 'versicolor' 'virginica']`

Características: `['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']`

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)



```
from sklearn.datasets import load_iris
```

```
iris = load_iris()  
x_data = iris.data  
y_lbl = iris.target  
classes = iris.target_names  
characteristics = iris.feature_names
```

```
0 1  
['sepal length (cm)', 'sepal width (cm)']  
  
2 3  
['petal length (cm)', 'petal width (cm)']
```

```
y_lbl = iris.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
y_lbl = iris.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
unique_values = np.unique(y_lbl)
```

```
y_lbl = iris.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
unique_values = np.unique(y_lbl)
```

```
print(f"Valores en y_lbl: {unique_values}")
```

Valores en y_lbl: [0 1 2]

```
y_lbl = iris.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
unique_values = np.unique(y_lbl)
```

```
print(f"Valores en y_lbl: {unique_values}")
```

Valores en y_lbl: [0 1 2]

```
lbl_class = [y_lbl[y_lbl == unique_values[i]] for i in unique_values]
```



```
y_lbl = iris.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
unique_values = np.unique(y_lbl)
```

```
print(f"Valores en y_lbl: {unique_values}")
```

Valores en y_lbl: [0 1 2]

```
lbl_class = [y_lbl[y_lbl == unique_values[i]] for i in unique_values]
total_class = [len(lbl) for lbl in lbl_class]
```

```
y_lbl = iris.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
unique_values = np.unique(y_lbl)
```

```
print(f"Valores en y_lbl: {unique_values}")
```

```
Valores en y_lbl: [0 1 2]
```

```
total_class = [np.count_nonzero(y_lbl == unique_values[i]) for i in unique_values]
```



```
y_lbl = iris.target
```


```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
unique_values = np.unique(y_lbl)
```

```
print(f"Valores en y_lbl: {unique_values}")
```

Valores en y_lbl: [0 1 2]

```
total_class = [np.count_nonzero(y_lbl == unique_values[i]) for i in unique_values]
```



```
y_lbl = iris.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
unique_values = np.unique(y_lbl)
```

```
print(f"Valores en y_lbl: {unique_values}")
```

Valores en y_lbl: [0 1 2]

```
total_class = [np.count_nonzero(y_lbl == unique_values[i]) for i in unique_values]
```

```
print(f"Numero de datos por clase:")
```

```
for i in range(len(total_class)):
    print(f"\t{total_class[i]} elementos en clase {i + 1}")
```

```
y_lbl = iris.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
unique_values = np.unique(y_lbl)
```

```
print(f"Valores en y_lbl: {unique_values}")
```

Valores en y_lbl: [0 1 2]

```
total_class = [np.count_nonzero(y_lbl == unique_values[i]) for i in unique_values]
```

```
print(f"Numero de datos por clase:")
```

Numero de datos por clase:

50 elementos en clase 1

50 elementos en clase 2

50 elementos en clase 3

```
for i in range(len(total_class)):
```

```
    print(f"\t{total_class[i]} elementos en clase {i + 1}")
```

```
x_data = iris.data
```

```
x_data = iris.data
```

```
[[5.1 3.5 1.4 0.2]  
 [4.9 3.  1.4 0.2]  
 [4.7 3.2 1.3 0.2]  
 [4.6 3.1 1.5 0.2]  
 [5.  3.6 1.4 0.2]  
 [5.4 3.9 1.7 0.4]  
 [4.6 3.4 1.4 0.3]  
 [5.  3.4 1.5 0.2]  
 [4.4 2.9 1.4 0.2]  
 [4.9 3.1 1.5 0.1]  
 [5.4 3.7 1.5 0.2]  
 [4.8 3.4 1.6 0.2]  
 [4.8 3.  1.4 0.1]
```

.

.

.

```
x_data = iris.data
```

```
[[5.1 3.5 1.4 0.2]  
 [4.9 3.  1.4 0.2]  
 [4.7 3.2 1.3 0.2]  
 [4.6 3.1 1.5 0.2]  
 [5.  3.6 1.4 0.2]  
 [5.4 3.9 1.7 0.4]  
 [4.6 3.4 1.4 0.3]  
 [5.  3.4 1.5 0.2]  
 [4.4 2.9 1.4 0.2]  
 [4.9 3.1 1.5 0.1]  
 [5.4 3.7 1.5 0.2]  
 [4.8 3.4 1.6 0.2]  
 [4.8 3.  1.4 0.1]  
 .  
 .  
 .
```

(150, 4)

```
x_data = iris.data
```

```
[[5.1 3.5 1.4 0.2]  
 [4.9 3.  1.4 0.2]  
 [4.7 3.2 1.3 0.2]  
 [4.6 3.1 1.5 0.2]  
 [5.  3.6 1.4 0.2]  
 [5.4 3.9 1.7 0.4]  
 [4.6 3.4 1.4 0.3]  
 [5.  3.4 1.5 0.2]  
 [4.4 2.9 1.4 0.2]  
 [4.9 3.1 1.5 0.1]  
 [5.4 3.7 1.5 0.2]  
 [4.8 3.4 1.6 0.2]  
 [4.8 3.  1.4 0.1]
```

(150, 4)

1.-Iris-Setosa

2.-Iris-Versicolor

3.-Iris-Virginica

```
x_1 = x_data[:50]  
x_2 = x_data[50:100]  
x_3 = x_data[100:]
```



```
x_1 = x_data[:50]  
x_2 = x_data[50:100]  
x_3 = x_data[100:]
```

```
steps = [sum(total_class[:i+1]) for i in range(len(total_class))]  
steps.append(0)  
x = [x_data[steps[i - 1]:steps[i]] for i in range(len(steps) - 1)]
```

```
x_1 = x_data[:50]  
x_2 = x_data[50:100]  
x_3 = x_data[100:]
```

```
steps = [sum(total_class[:i+1]) for i in range(len(total_class))]  
steps.append(0)  
x = [x_data[steps[i - 1]:steps[i]] for i in range(len(steps) - 1)]
```

```
print(f"Dimensión de x_1: {x_1.shape}")  
print(f"Dimensión de x_2: {x_2.shape}")  
print(f"Dimensión de x_3: {x_3.shape}")
```

✓ 0.0s

```
Dimensión de x_1: (50, 4)  
Dimensión de x_2: (50, 4)  
Dimensión de x_3: (50, 4)
```

```
x_1 = x_data[:50]
x_2 = x_data[50:100]
x_3 = x_data[100:]
```

```
steps = [sum(total_class[:i+1]) for i in range(len(total_class))]
steps.append(0)
x = [x_data[steps[i - 1]:steps[i]] for i in range(len(steps) - 1)]
```

```
print(f"Dimensión de x_1: {x_1.shape}")
print(f"Dimensión de x_2: {x_2.shape}")
print(f"Dimensión de x_3: {x_3.shape}")
```

✓ 0.0s

Dimensión de x_1: (50, 4)
Dimensión de x_2: (50, 4)
Dimensión de x_3: (50, 4)

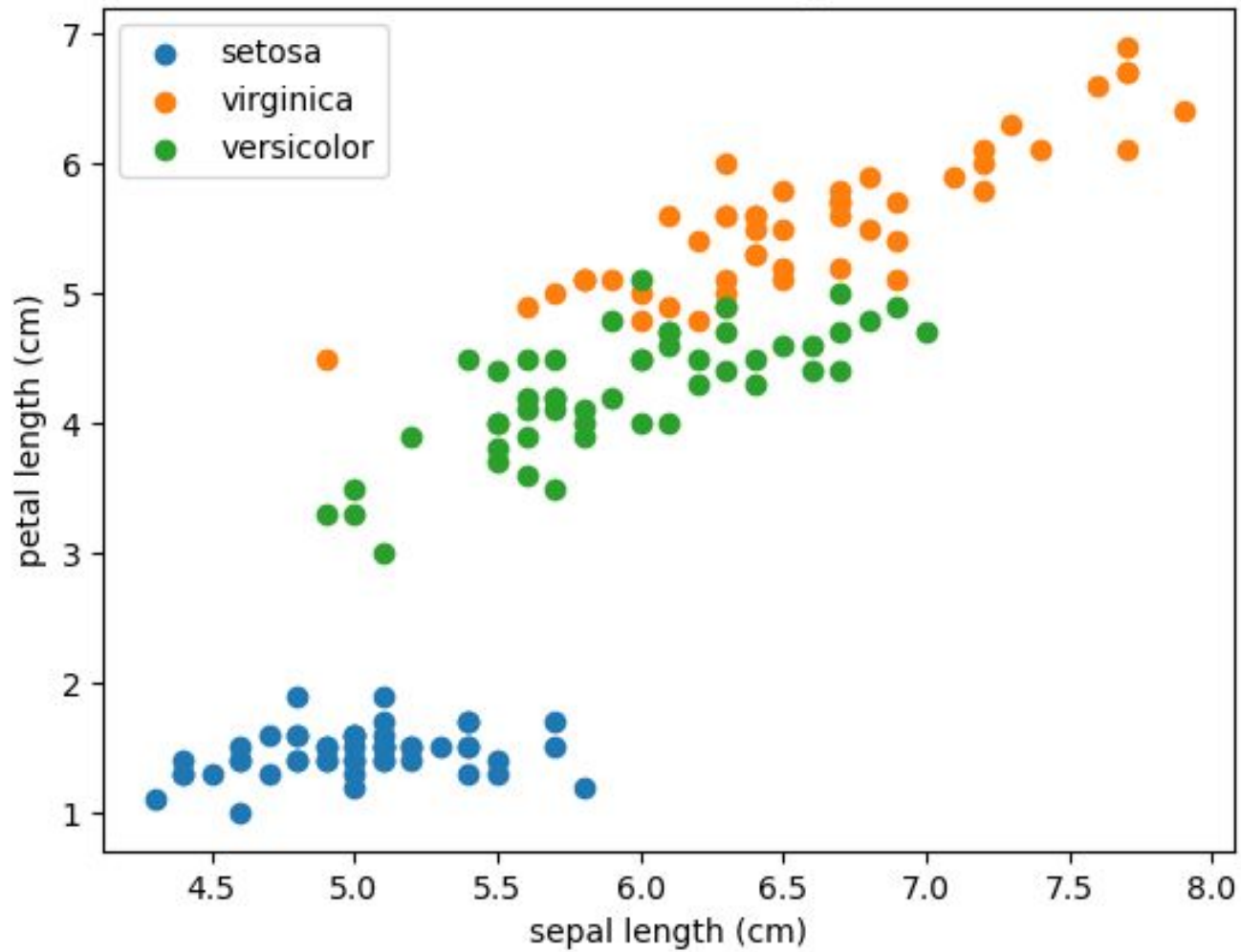
0	1
['sepal length (cm)', 'sepal width (cm)']	
2	3
['petal length (cm)', 'petal width (cm)']	

```
eje_x = 0  
eje_y = 2  
clase1 = 0  
clase2 = 2  
clase3 = 1
```

```
plt.scatter(x[clase1][:,eje_x], x[clase1][:,eje_y])  
plt.scatter(x[clase2][:,eje_x], x[clase2][:,eje_y])  
plt.scatter(x[clase3][:,eje_x], x[clase3][:,eje_y])
```

```
plt.title(f"Iris-{classes[clase1]} vs Iris-{classes[clase2]}")  
plt.xlabel(characteristics[eje_x])  
plt.ylabel(characteristics[eje_y])  
plt.legend([classes[clase1], classes[clase2], classes[clase3]])  
  
plt.show()
```

Iris-setosa vs Iris-virginica



QUERY

QUERY

TRAINING

QUERY

TRAINING

```
percent_query = 20
```


QUERY

TRAINING

```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
```

QUERY

TRAINING

```
percent_query = 20
```

```
nume_samples_query = int((len(x_data) * percent_query) / 100)
```

```
query_index_samples = np.random.choice(149, nume_samples_query, replace=False)
```

QUERY

TRAINING

```
percent_query = 20

num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

QUERY

TRAINING

```
percent_query = 20
```

```
nume_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, nume_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```



```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
```

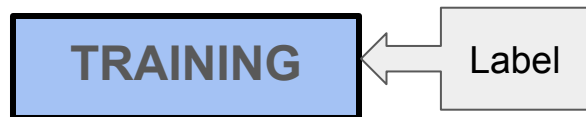


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```

$$Q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dots \\ q_n \end{bmatrix}$$



```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```

$$Q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dots \\ q_n \end{bmatrix} \quad T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ \dots \\ t_n \end{bmatrix}$$



```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```

$$\begin{array}{|c|} \hline q_1 \\ \hline q_2 \\ \hline q_3 \\ \hline \dots \\ \hline q_n \\ \hline \end{array} \qquad \begin{array}{|c|} \hline t_1 \\ \hline t_2 \\ \hline t_3 \\ \hline \dots \\ \hline t_n \\ \hline \end{array}$$

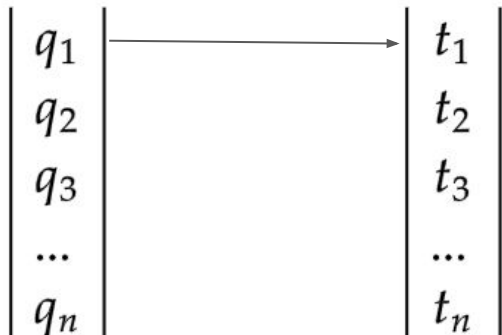


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```





```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
```

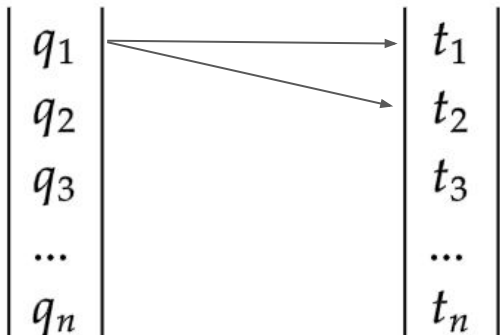
```
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
```

```
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
```

```
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



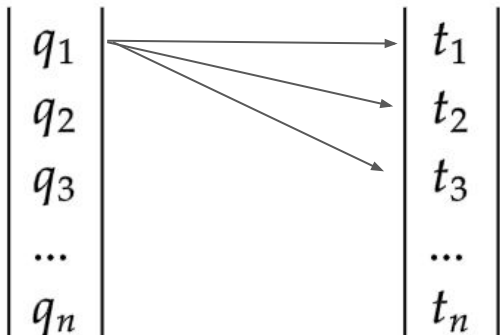


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



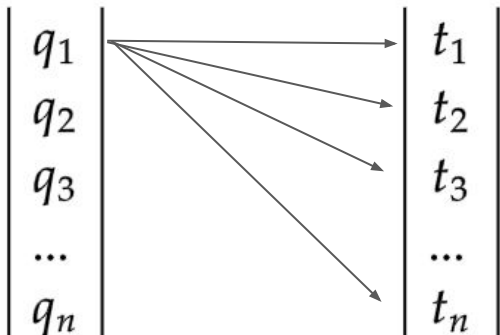


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



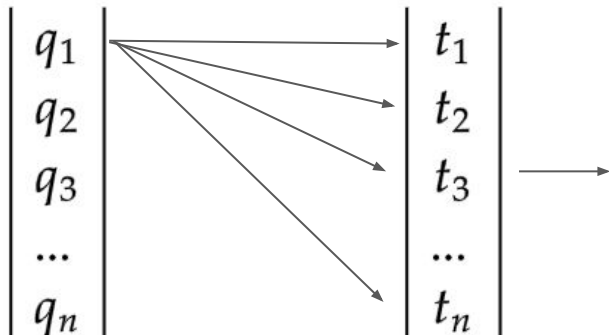


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



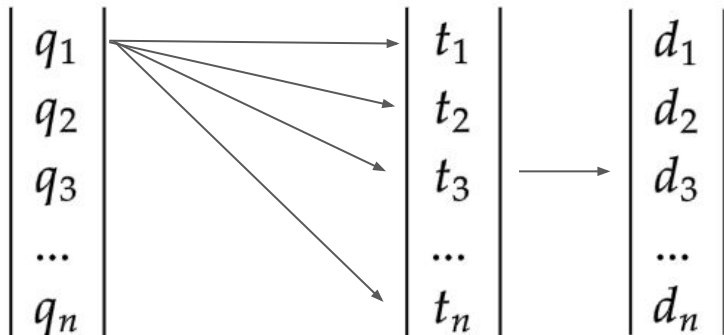


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```





```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
```

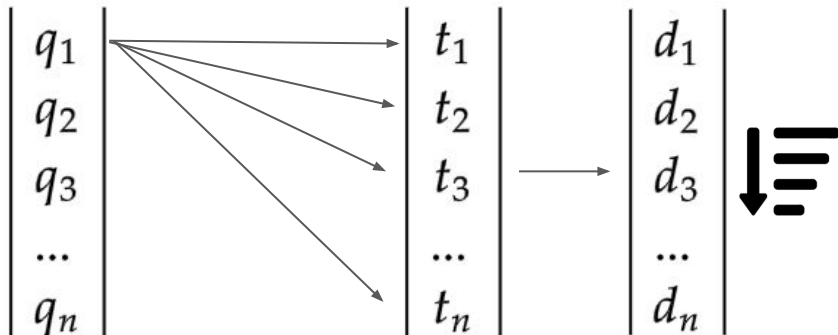
```
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
```

```
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
```

```
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



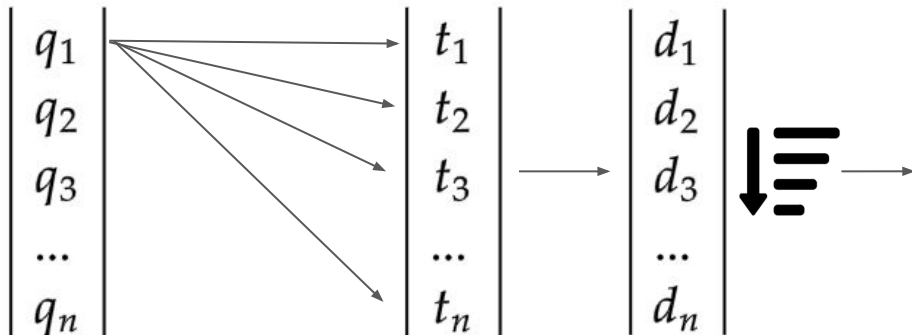


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



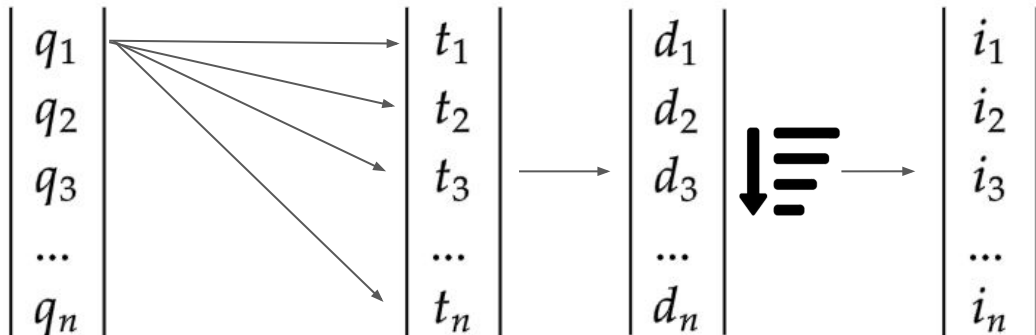


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



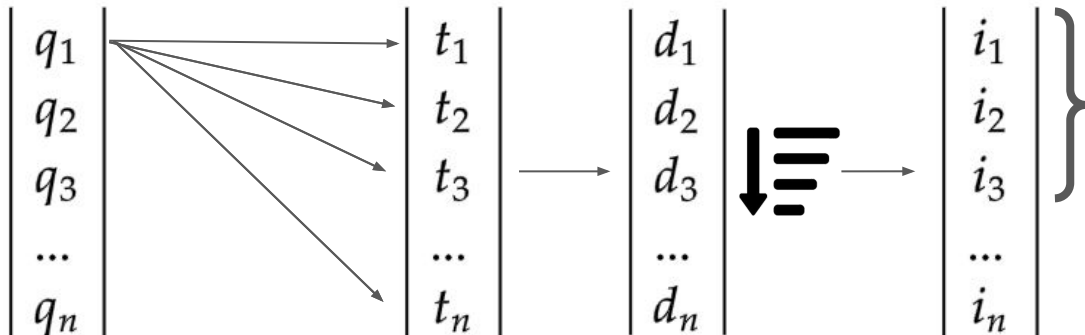


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



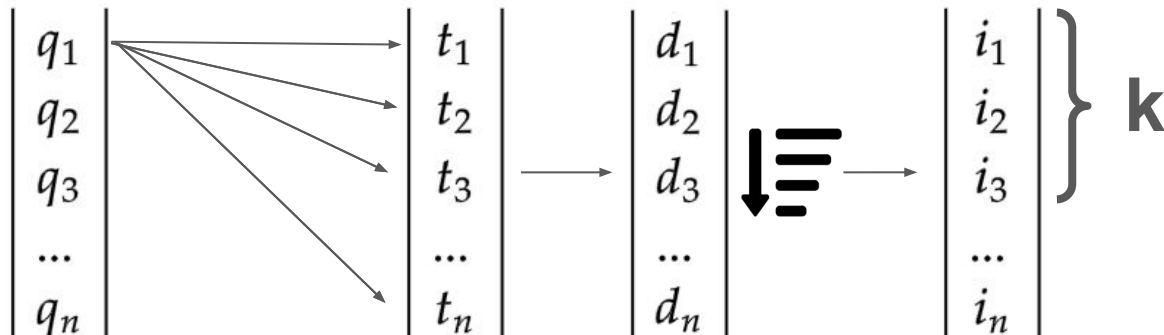


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```





```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
```

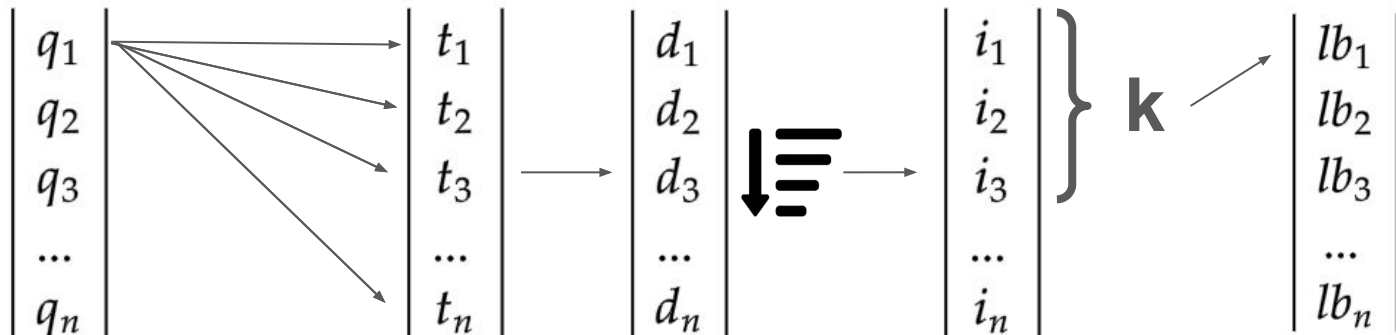
```
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
```

```
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
```

```
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```





```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
```

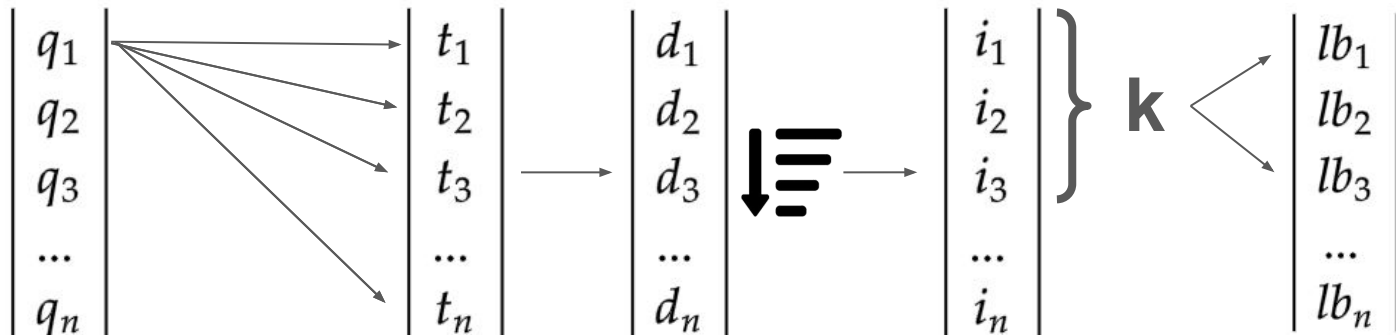
```
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
```

```
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
```

```
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



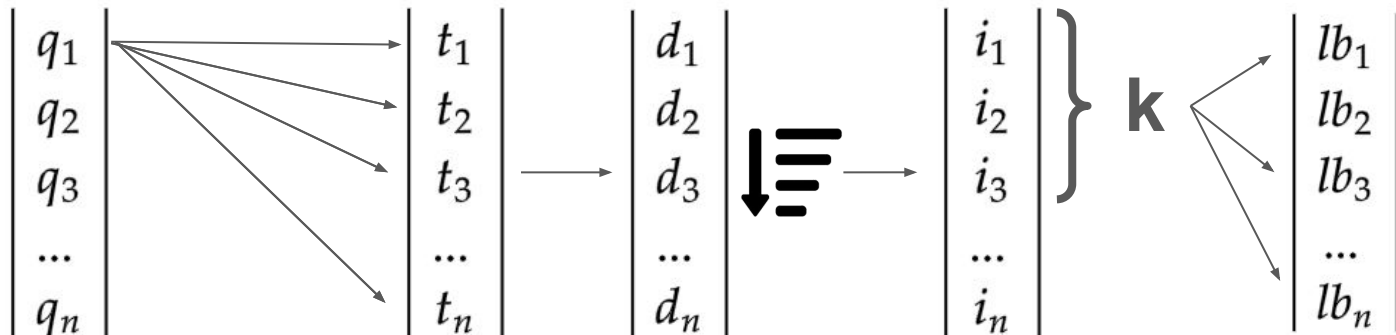


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



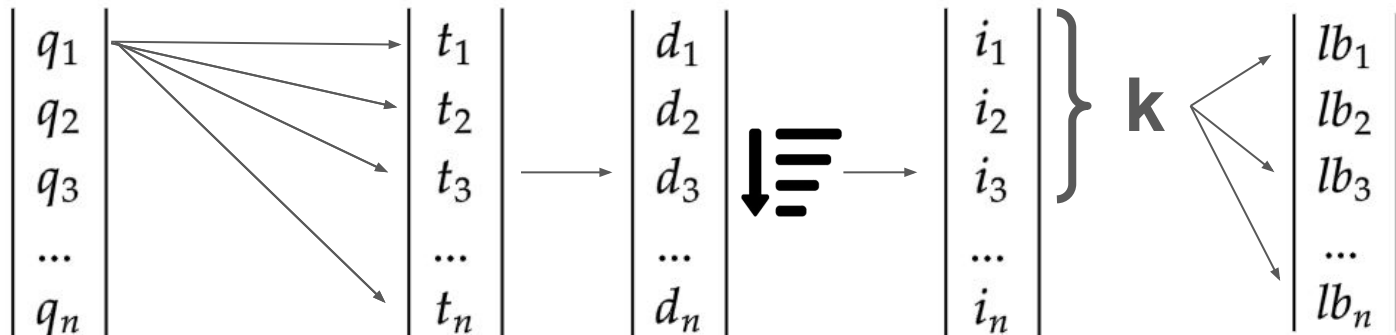


```
percent_query = 20
```

```
num_samples_query = int((len(x_data) * percent_query) / 100)
query_index_samples = np.random.choice(149, num_samples_query, replace=False)
query_samples = x_data[query_index_samples]
```

```
training_samples = np.delete(x_data, query_index_samples, axis=0)
```

```
query_sample_lbl = y_lbl[query_index_samples]
training_sample_lbl = np.delete(y_lbl, query_index_samples, axis=0)
```



**Voto
Mayoritario**

Código kNN

Código kNN

```
k = 5  
predictions = []
```

Código kNN

```
k = 5  
predictions = []  
for i in range(len(query_samples)):
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = ((query_samples[i][0] - training_samples[j][0]) ** 2 +
                     (query_samples[i][1] - training_samples[j][1]) ** 2 +
                     (query_samples[i][2] - training_samples[j][2]) ** 2 +
                     (query_samples[i][3] - training_samples[j][3]) ** 2)
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
```


Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
    distancias = np.array(distancias)
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
    distancias = np.array(distancias)
    sort_index = np.argsort(distancias)
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
    distancias = np.array(distancias)
    sort_index = np.argsort(distancias)
    k_index = sort_index[:k]
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
    distancias = np.array(distancias)
    sort_index = np.argsort(distancias)
    k_index = sort_index[:k]
    k_labels = training_sample_lbl[k_index]
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
    distancias = np.array(distancias)
    sort_index = np.argsort(distancias)
    k_index = sort_index[:k]
    k_labels = training_sample_lbl[k_index]

    vals, counts = np.unique(k_labels, return_counts=True)
```

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
    distancias = np.array(distancias)
    sort_index = np.argsort(distancias)
    k_index = sort_index[:k]
    k_labels = training_sample_lbl[k_index]

    vals, counts = np.unique(k_labels, return_counts=True)
    mode_label = vals[np.argmax(counts)]
```


Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    distancias = []
    for j in range(len(training_samples)):
        distancia = 0
        for k in range(len(query_samples[i])):
            distancia += (query_samples[i][k] - training_samples[j][k]) ** 2
        distancia = distancia ** .5
        distancias.append(distancia)
    distancias = np.array(distancias)
    sort_index = np.argsort(distancias)
    k_index = sort_index[:k]
    k_labels = training_sample_lbl[k_index]

    vals, counts = np.unique(k_labels, return_counts=True)
    mode_label = vals[np.argmax(counts)]
    predictions.append(mode_label)
```

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & q_{n4} \end{pmatrix}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & q_{n4} \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & q_{n4} \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ \vdots & \vdots & \vdots & \vdots \\ t_{m1} & t_{m2} & t_{m3} & t_{m4} \end{pmatrix}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & q_{n4} \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ \vdots & \vdots & \vdots & \vdots \\ t_{m1} & t_{m2} & t_{m3} & t_{m4} \end{pmatrix}$$

$$\sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & q_{n4} \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ \vdots & \vdots & \vdots & \vdots \\ t_{m1} & t_{m2} & t_{m3} & t_{m4} \end{pmatrix}$$

$$\sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{11} & q_{12} & q_{13} & q_{14} \\ \vdots & \vdots & \vdots & \vdots \\ q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & q_{n4} \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ \vdots & \vdots & \vdots & \vdots \\ t_{m1} & t_{m2} & t_{m3} & t_{m4} \end{pmatrix}$$

$$\sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{11} & q_{12} & q_{13} & q_{14} \\ \vdots & \vdots & \vdots & \vdots \\ q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix} -$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & q_{n4} \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ \vdots & \vdots & \vdots & \vdots \\ t_{m1} & t_{m2} & t_{m3} & t_{m4} \end{pmatrix}$$

$$\sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{11} & q_{12} & q_{13} & q_{14} \\ \vdots & \vdots & \vdots & \vdots \\ q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix} - \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ \vdots & \vdots & \vdots & \vdots \\ t_{m1} & t_{m2} & t_{m3} & t_{m4} \end{pmatrix}$$

$$\begin{pmatrix} q_{11} - t_{11} & q_{12} - t_{12} & q_{13} - t_{13} & q_{14} - t_{14} \\ q_{11} - t_{21} & q_{12} - t_{22} & q_{13} - t_{23} & q_{14} - t_{24} \\ \vdots & \vdots & \vdots & \vdots \\ q_{11} - t_{m1} & q_{12} - t_{m2} & q_{13} - t_{m3} & q_{14} - t_{m4} \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} = \mathbf{Z}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [] & \dots & [] \\ [] & [] & \dots & [] \\ [] & [] & \dots & \vdots \\ [] & [] & \dots & [] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [] \\ [] & [] & \dots & [] \\ [] & [] & \dots & \vdots \\ [] & [] & \dots & [] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [s_{1m}] \\ [] & [] & \dots & [] \\ [] & [] & \dots & \vdots \\ [] & [] & \dots & [] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [s_{1m}] \\ [s_{21}] & [] & \dots & [] \\ [] & [] & \dots & \vdots \\ [] & [] & \dots & [] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [s_{1m}] \\ [s_{21}] & [s_{22}] & \dots & [] \\ [] & [] & \dots & \vdots \\ [] & [] & \dots & [] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [s_{1m}] \\ [s_{21}] & [s_{22}] & \dots & [s_{2m}] \\ [] & [] & \dots & \vdots \\ [] & [] & \dots & [] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [s_{1m}] \\ [s_{21}] & [s_{22}] & \dots & [s_{2m}] \\ \dots & \dots & \dots & \vdots \\ [s_{m1}] & [] & \dots & [] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [s_{1m}] \\ [s_{21}] & [s_{22}] & \dots & [s_{2m}] \\ \dots & \dots & \dots & \vdots \\ [s_{m1}] & [s_{m2}] & \dots & [] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ \boxed{z_{m1} & z_{m2} & z_{m3} & z_{m4}} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & \boxed{z_{m1}} \\ z_{12} & z_{22} & \dots & \boxed{z_{m2}} \\ z_{13} & z_{23} & \dots & \boxed{\vdots} \\ z_{14} & z_{24} & \dots & \boxed{z_{m4}} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [s_{1m}] \\ [s_{21}] & [s_{22}] & \dots & [s_{2m}] \\ \dots & \dots & \dots & \vdots \\ [s_{m1}] & [s_{m2}] & \dots & [s_{mm}] \end{pmatrix}$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \bullet \begin{pmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{13} & z_{23} & \dots & \vdots \\ z_{14} & z_{24} & \dots & z_{m4} \end{pmatrix} = \mathbf{S} = \begin{pmatrix} [s_{11}] & [s_{12}] & \dots & [s_{1m}] \\ [s_{21}] & [s_{22}] & \dots & [s_{2m}] \\ \dots & \dots & \dots & \vdots \\ [s_{m1}] & [s_{m2}] & \dots & [s_{mm}] \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

$$S = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
```

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
```

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{11} & q_{12} & q_{13} & q_{14} \\ \vdots & \vdots & \vdots & \vdots \\ q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
```

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{11} & q_{12} & q_{13} & q_{14} \\ \vdots & \vdots & \vdots & \vdots \\ q_{11} & q_{12} & q_{13} & q_{14} \end{pmatrix} - \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ \vdots & \vdots & \vdots & \vdots \\ t_{m1} & t_{m2} & t_{m3} & t_{m4} \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
    S = Z @ Z.T
```

$$\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix} \cdot \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{pmatrix}^T$$

$$S = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
    S = Z @ Z.T
    distancia_S = np.diag(S) ** (1 / 2)
```

$$S = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
    S = Z @ Z.T
    distancia_S = np.diag(S) ** (1 / 2)
    sort_index = np.argsort(distancia_S)
```

$$S = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
    S = Z @ Z.T
    distancia_S = np.diag(S) ** (1 / 2)
    sort_index = np.argsort(distancia_S)
    k_index = sort_index[:k]
```

$$S = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
    S = Z @ Z.T
    distancia_S = np.diag(S) ** (1 / 2)
    sort_index = np.argsort(distancia_S)
    k_index = sort_index[:k]
    k_labels = training_sample_lbl[k_index]
```

$$\mathbf{S} = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
    S = Z @ Z.T
    distancia_S = np.diag(S) ** (1 / 2)
    sort_index = np.argsort(distancia_S)
    k_index = sort_index[:k]
    k_labels = training_sample_lbl[k_index]

    vals, counts = np.unique(k_labels, return_counts=True)
```

$$S = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
    S = Z @ Z.T
    distancia_S = np.diag(S) ** (1 / 2)
    sort_index = np.argsort(distancia_S)
    k_index = sort_index[:k]
    k_labels = training_sample_lbl[k_index]

    vals, counts = np.unique(k_labels, return_counts=True)
    mode_label = vals[np.argmax(counts)]
```


$$S = \begin{pmatrix} X & s_{21} & \dots & s_{m1} \\ s_{12} & X & \dots & s_{m2} \\ s_{13} & s_{23} & \dots & \vdots \\ s_{14} & s_{24} & \dots & X \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} ** .5$$

Código kNN

```
k = 5
predictions = []
for i in range(len(query_samples)):
    Q = np.tile(query_samples[i,:],(training_samples.shape[0],1))
    Z = Q - training_samples
    S = Z @ Z.T
    distancia_S = np.diag(S) ** (1 / 2)
    sort_index = np.argsort(distancia_S)
    k_index = sort_index[:k]
    k_labels = training_sample_lbl[k_index]

    vals, counts = np.unique(k_labels, return_counts=True)
    mode_label = vals[np.argmax(counts)]
    predictions.append(mode_label)
```