**Actividad práctica: aplicación de kernels**

Una de las dudas habituales a la hora de trabajar con kernels es: ¿cuál de los existentes debemos considerar? La recomendación más generalizada consiste en empezar siempre por el kernel lineal (o bien emplear directamente la función LinearSVC), sobre todo si el volumen de datos es muy elevado. Para conjuntos de entrenamientos no excesivamente grandes se debe valorar, a continuación, el funcionamiento de kernels de base radial gaussiana. Y, finalmente, para datos muy específicos, se pueden plantear otros enfoques. En base a lo expuesto, procede a realizar la clasificación del dataset iris, considerando la longitud de los sépalos y las de los pétalos y empleando el kernel «linear», «rbf» y «poli». Determina la exactitud alcanzada en cada caso. Si puedes, incorpora a la función plot\_decision\_regions otros atributos para mejorar la salida de los gráficos; añade también qué estas representando en cada eje. Una vez realizado lo mencionado, comprueba tu respuesta.

“Copilot, haz codigo python Utilizando el dataset iris y el kernel lineal Linearsvc, y gausianos rbf y poli entrenar learn y mostrar graficamente con patlibplot plot\_decision\_regions el resultado de del dataset de entrenamiento y de test asó como la precision”

Una de las dudas habituales a la hora de trabajar con kernels es: ¿cuál de los existentes debemos considerar?

\* La recomendación más generalizada consiste en empezar siempre por el kernel lineal (o bien emplear directamente la función LinearSVC), sobre todo si el volumen de datos es muy elevado.

\* Para conjuntos de entrenamientos no excesivamente grandes se debe valorar, a continuación, el funcionamiento de kernels de base radial gaussiana.

Y, finalmente, para datos muy específicos, se pueden plantear otros enfoques.

En base a lo expuesto, procede a realizar la clasificación del dataset iris, considerando la longitud de los sépalos y las de los pétalos y empleando el kernel «linear», «rbf» y «poli». Determina la exactitud alcanzada en cada caso. Si puedes, incorpora a la función plot\_decision\_regions otros atributos para mejorar la salida de los gráficos; añade también qué estas representando en cada eje. Una vez realizado lo mencionado, comprueba tu respuesta.

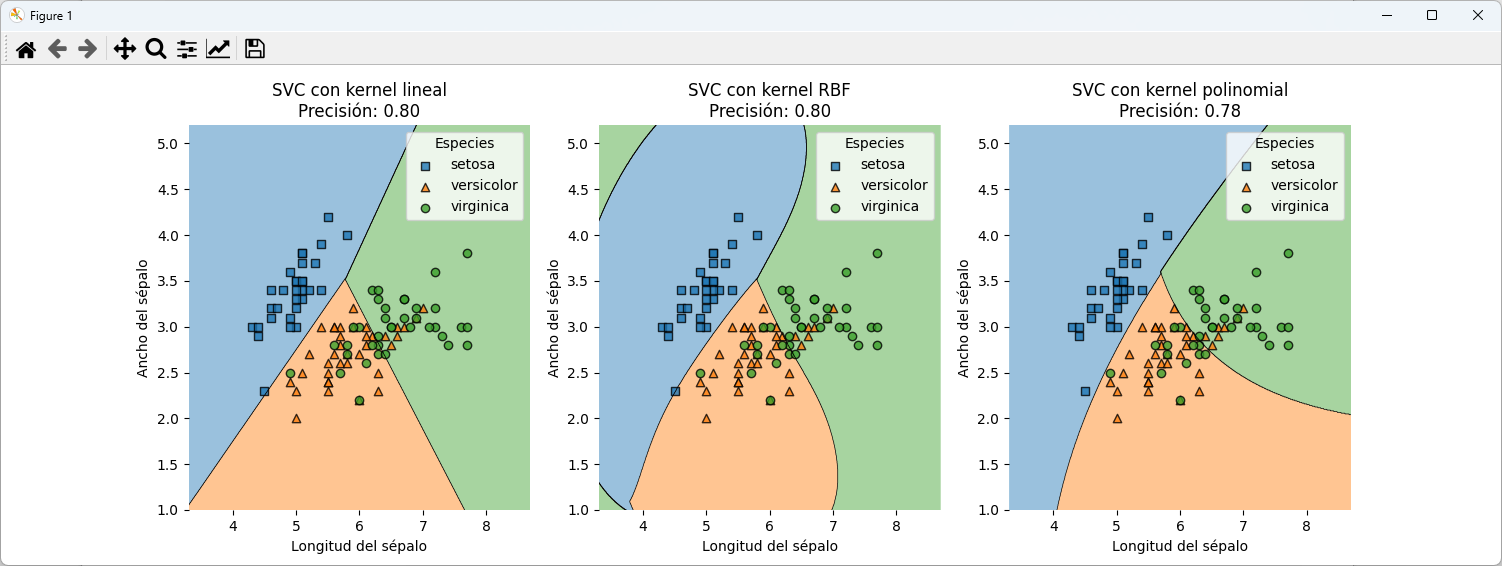
model1 = [

svm.SVC(kernel="linear", C=1.0),

svm.SVC(kernel="rbf", gamma=0.7, C=1.0),

svm.SVC(kernel="poly", degree=3, gamma="auto", C=1.0)

]

models = [

svm.SVC(kernel="linear", C=2.0),

svm.SVC(kernel="rbf", gamma=1.5, C=2.0),

svm.SVC(kernel="poly", degree=6, gamma="auto", C=2.0)

]

