UNIVERSIDAD CATÓLICA DE SAN PABLO
MAESTRÍA EN CIENCIA DE LA COMPUTACIÓN
SISTEMAS INTELIGENTES

Máquina de Vectores de Soporte (SVM)
Prof. Graciela Meza Lovón

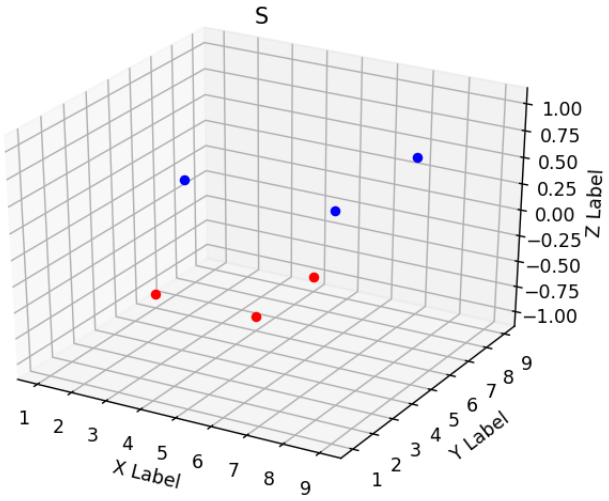
Palomino Paucar, Daniel Alfredo
Noviembre 26, 2018
https://bit.ly/2r7y556

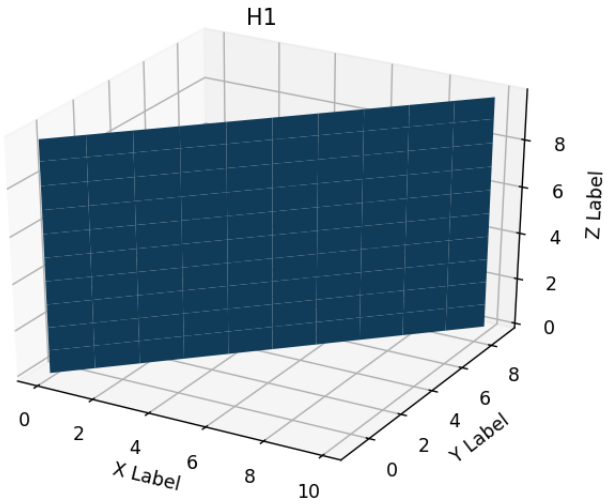
1. Preguntas de Teoría

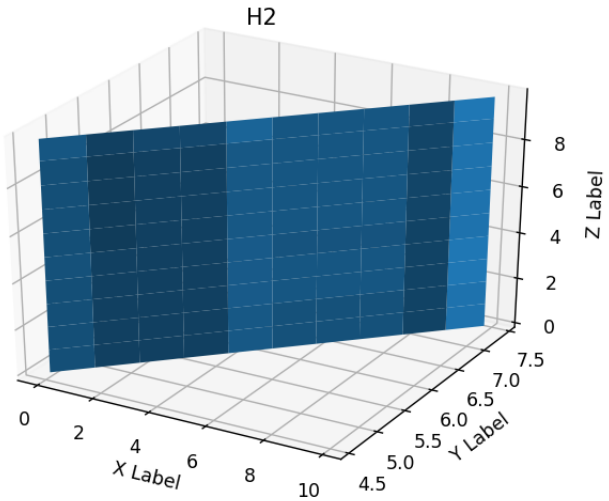
Sea el conjunto $S = ((1,6),-1),((4,9),-1),((4,6),-1),((5,1),1),((9,5),1),((9,1),1)$ **y un conjunto de cuatro hiperplanos** $H = H_1, H_2, H_3, H_4$ **definidos como:** $H_1 : x_1 - x_2 - 1 = 0$**,** $H_2 : 2x_1 - 7x_2 + 32 = 0$**,** $H_3 : \sqrt{\frac{1}{2}}x_1 - \sqrt{\frac{1}{2}}x_2 - \sqrt{\frac{1}{2}} = 0$**,** $H_4 : 2x_1 - 7x_2 - 32 = 0$

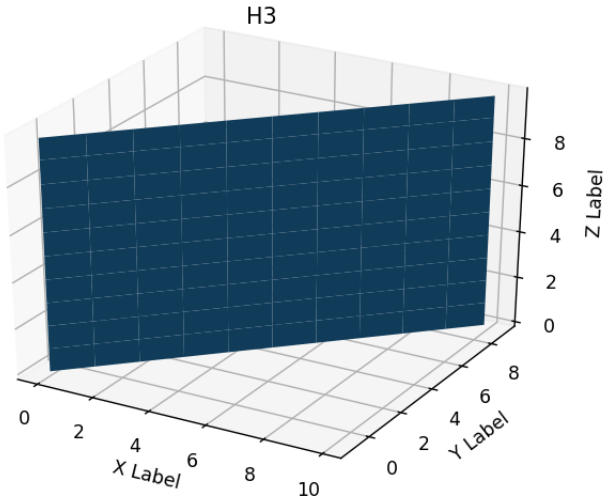
1. Usando cualquier lenguaje de programación grafique S, H_1, H_2, H_3 *y* H_4.

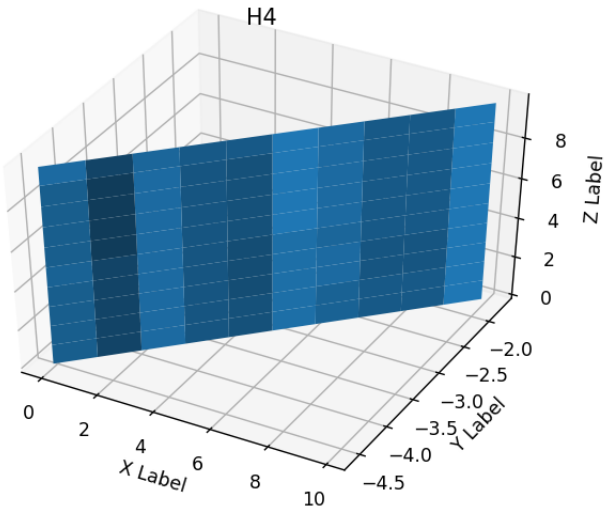
   Respuesta:






H2
X Label
Y Label
Z Label
0
2
4
6
8
10
4.5
5.0
5.5
6.0
6.5
7.0
7.5
0
2
4
6
8


H3
X Label
Y Label
Z Label
0
2
4
6
8
10
0
2
4
6
8
0
2
4
6
8



2. Encuentre los parámetros w y b que definen los hiperplanos $H_1,\ H_2,\ H_3\ y\ H_4$, y luego determine para $H_1,\ H_2,\ H_3\ y\ H_4$ si son hiperplanos de separación. Fundamente.

   Respuesta:

   Dadas la ecuación vectorial de un hiperplano: $W.X + b = 0$, se tiene al extender la ecuación: $w_1.x_1 + w_2.x_2 + w_3.w_3 + b = 0$.

   Dando la forma a la ecuaciones dadas:

   a) H_1 :

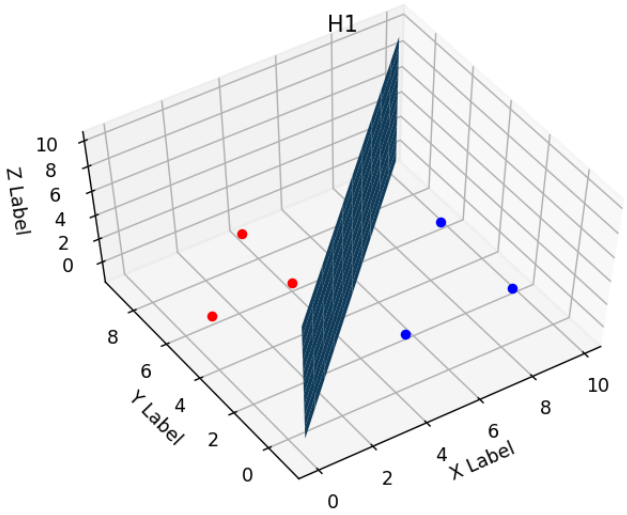
   $$x_1 - x_2 - 1 = 0$$

   $$(1,-1,0).X - 1 = 0$$

   Entonces:

   $$W = (1,-1,0) \wedge b = -1$$

   Graficando:



Por lo tanto H_1: Sí es un hiperplano de separación.

b) H_2 :

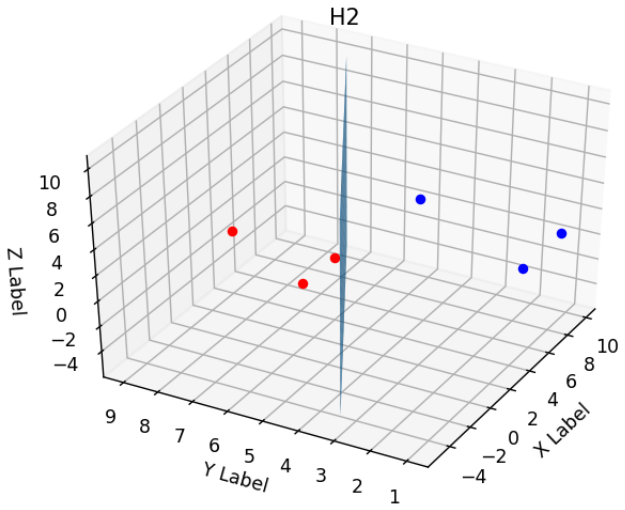
$$2x_1 - 7x_2 + 32 = 0$$

$$(2,-7,0).X + 32 = 0$$

Entonces:

$$W = (2,-7,0) \wedge b = 32$$

Graficando:



Por lo tanto H_2: Sí es un hiperplano de separación.

c) H_3 :

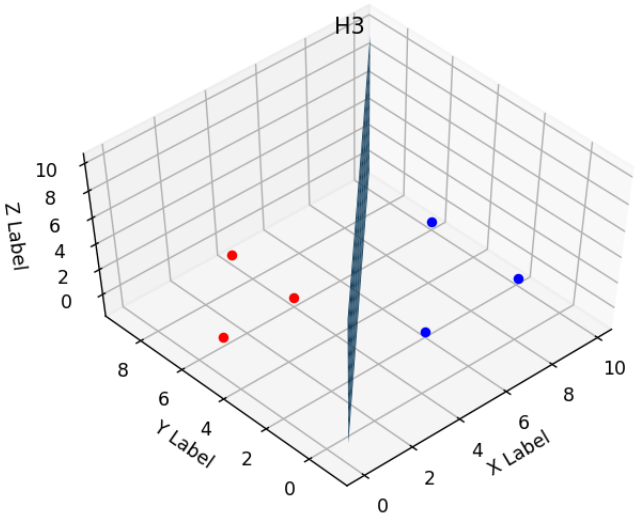
$$\sqrt{\frac{1}{2}}x_1 - \sqrt{\frac{1}{2}}x_2 - \sqrt{\frac{1}{2}} = 0$$

$$(1,-1,0).X - 1 = 0$$

Entonces:

$$W = (\sqrt{\frac{1}{2}}, -\sqrt{\frac{1}{2}}, 0) \wedge b = -\sqrt{\frac{1}{2}}$$

Graficando:



Por lo tanto H_3: Sí es un hiperplano de separación.

d) H_4 :

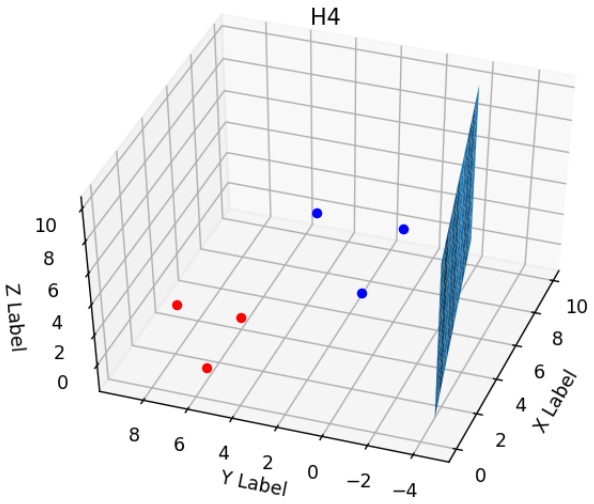
$$2x_1 - 7x_2 - 32 = 0$$

$$(2,-7,0).X - 32 = 0$$

Entonces:

$$W = (2,-7,0) \wedge b = -32$$

Graficando:



Por lo tanto H_4: No es un hiperplano de separación.

3. En el conjunto H, ¿cuántos hiperplanos iguales existen?. En el caso de que existan ¿cuáles son éstos? Fundamente.

   Respuesta:

   Existen 2 hiperplanos iguales H_1 *y* H_3, dado que si a la ecuación de H_3 la dividimos entre $\sqrt{\frac{1}{2}}$ tendríamos la misma ecuación que H_1.

4. Calcule el margen τ para cada hiperplano de separación. Luego, suponga que el conjunto H contiene al hiperplano óptimo H^*, ¿cuál sería H^*? Fundamente.

   Respuesta:

   a) H_1:

   $$\tau_1 = 2{,}1213$$

   b) H_2:

   $$\tau_2 = 0{,}2742$$

   c) H_3:

   $$\tau_3 = 2{,}1213$$

   De acuerdo a los datos calculados, si el hiperplano H^* existiera en el conjunto H, este sería H_1 o H_3 (pues representan el mismo plano), dado que poseen el τ con mayor valor.

5. ¿Cuáles son los vectores de soporte del hiperplano H^* escogido en la pregunta anterior?. Fundamente. (No necesita encontrar los valores α)

   Respuesta:

   Los vectores de soporte del hiperplano óptimo elegido (H_1 o H_3)serían:

   $$(4,6,-1),\ (5,1,1),\ (9,5,1)$$

   pues se encuentran en el margen óptimo, teniendo cada vector el τ mínimo de 2,1313.

6. Demuestre la primera condición KKT, i.e. (Ec. 7 de las diapositivas) $\frac{\partial L}{\partial w}(w^*, b^*, \alpha) = w^* - \sum_{i=1}^{m} \alpha_i y(i) x^i$

   Respuesta:

   Sea:

   $$L = \frac{1}{2} W.W^T - \sum_{i=1}^{m} \alpha_i [y^{(i)}(W^T X^{(i)} + b) - 1]$$

   Expandiendo los términos:

   $$L = \frac{1}{2} W.W^T - \sum_{i=1}^{m} \alpha_i [y^{(i)} W^T X^{(i)}] - \sum_{i=1}^{m} \alpha_i [y^{(i)} b] + \sum_{i=1}^{m} \alpha_i$$

   Para encontrar los $W*, b^*, \alpha^*$ óptimo, necesitamos derivar respecto de los parámetros, lo cual se describe en la ecuación 1:

   $$\frac{\partial L(W^*, b^*, \alpha^*)}{\partial w_i} = 0, i = 1, ..., n$$

   Entonces, derivando L respecto de w_i tenemos:

   $$\frac{\partial L(W^*, b^*, \alpha^*)}{\partial w_i} = \frac{\partial(\frac{1}{2} W.W^T - \sum_{i=1}^{m} \alpha_i [y^{(i)} W^T X^{(i)}] - \sum_{i=1}^{m} \alpha_i [y^{(i)} b] + \sum_{i=1}^{m} \alpha_i)}{\partial w_i}$$

   Separando los términos de la derecha:

   $$\frac{\partial L(W^*, b^*, \alpha^*)}{\partial w_i} = \frac{\partial(\frac{1}{2} W.W^T)}{\partial w_i} - \frac{\partial(\sum_{i=1}^{m} \alpha_i [y^{(i)} W^T X^{(i)}])}{\partial w_i} - \frac{\partial(\sum_{i=1}^{m} \alpha_i [y^{(i)} b])}{\partial w_i} + \frac{\partial(\sum_{i=1}^{m} \alpha_i)}{\partial w_i}$$

   El tercer y cuarto término de la derecha no depende de w_i, por lo tanto al derevirase desapareceran:

$$\frac{\partial L(W^*, b^*, \alpha^*)}{\partial w_i} = \frac{\partial(\frac{1}{2} W.W^T)}{\partial w_i} - \frac{\partial(\sum_{i=1}^{m} \alpha_i [y^{(i)} W^T X^{(i)}])}{\partial w_i}$$

Finalmente calculando la derivada parcial:

$$\frac{\partial L}{\partial w}(w^*, b^*, \alpha) = w^* - \sum_{i=1}^{m} \alpha_i y(i) x^i = 0$$

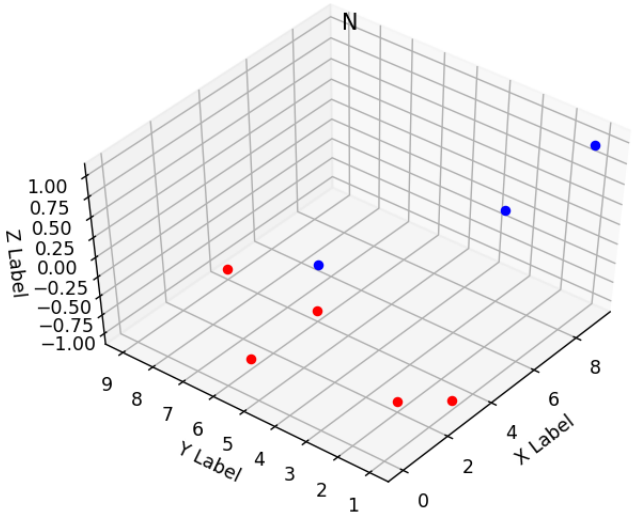
Quedando demostrado la ecuación 7 de las diapositivas.

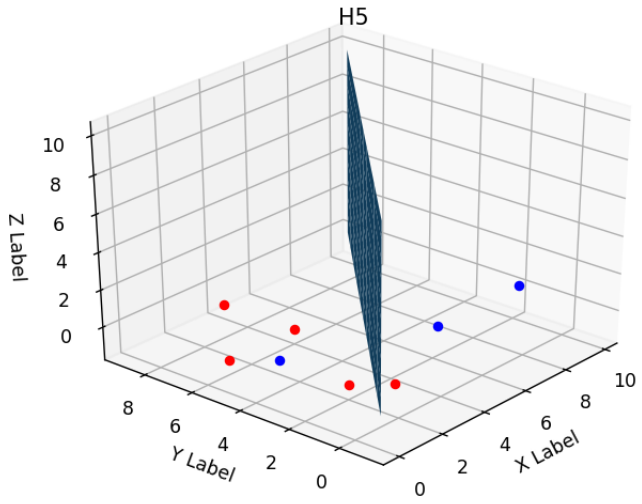
Sea el conjunto $N = ((1,6),-1),((4,9),-1),((4,6),-1),((5,1),1),((9,1),1),((0,3),1),((2,2),-1),((3,1),-1)$ **y el hiperplano** H_1 **definido anteriormente.**

7. Usando cualquier lenguaje de programación grafique N y H_5.

   Respuesta:

   Graficando:





8. Identifique los ejemplos que son separables y los que no lo son. Luego, determine los ejemplos que son clasificados correctamente y los que no.

   Respuesta:

   Para que un vector sea separable debe cumplir:

$$y_i(W^T X) \geq 1$$

   Por lo tanto haciendo los cálculos para los ejemplos del espacion N y el hiperplano H_5:

   a) $((1,6),-1)$: 5

   b) $((4,9),-1)$: 5

   c) $((4,6),-1)$: 2

   d) $((5,1),1)$: 4

   e) $((9,1),1)$: 8

   f) $((0,3),1)$: -3

   g) $((2,2),-1)$: 0

   h) $((3,1),-1)$: -2

Por lo tanto: Los ejemplos separables son:
$((1,6),-1),((4,9),-1),((4,6),-1),((5,1),1),((9,1),1)$

y los no separables son:
$((0,3),1),((2,2),-1),((3,1),-1)$

Así mismo, los ejemplos clasificados correctamente son:
$((1,6),-1),((4,9),-1),((4,6),-1),((5,1),1),((9,1),1),((2,2),-1)$

y los no clasficados correctamente son:
$((0,3),1),((3,1),-1)$

9. Calcule la holgura de los ejemplos no separables.

   Respuesta:

   Dado que:

$$y_i(W^T X) \geq 1 - \epsilon_i$$

   Se tendrá para los ejemplos no separables $((0,3),1),((2,2),-1),((3,1),-1)$:

   a) $((0,3),\ 1)$: $\epsilon_i = 4$

   b) $((2,2),-1)$: $\epsilon_i = 1$

   c) $((3,1),-1)$: $\epsilon_i = 3$

2. PREGUNTAS DE INVESTIGACIÓN

1. Explique el significados de la constante C en el término $C\sum_{i=1}\epsilon_i$ que se agrega a la función objetivo en el caso de ejemplos casi linealmente separables. Luego, explique la influencia de C en la capacidad de generalización de una SVM.

   Respuesta:

   El parámetro C está asociado al mejoramiento de la generalización del sistema evitando el overfitting, es análogo al parámetro de regularización usando en redes neuronales.

   El parámetro C le dice al sistema SVM que tanto quiere que se evite el error de clasficación (misclassifying) por cada ejemplo de entrenamiento.

   Para valores grandes de C, la optimización elegirá un hiperplano de margen angosto si es que ese hiperplano realiza un mejor trabajo en clasificar correctamente todos los ejemplos de entrenamiento.

   Por el contrario, un valor muye pequeño de C causará que sistema busque un hiperplano de margen amplio, incluso si ese hiperplano clasifica incorrectamente más puntos.

2. Describa el significado del parámetro γ en el kernel gaussiano. Luego, explique la influencia de γ en la capacidad de generalización de una SVM.

   Respuesta:

   El parámetro γ es el parámetro libre de la función base del radio gausiano.

   Un valor pequeño de γ significa una función gaussiana con una varianza grande, así que la influencia de x_j es más grande hacia x_i. Es decir, si x_j es un vector de soporte, un valor pequeño de γ implica que la clase de este vector de soporte tendrá influencia en decidir la clase del vector x_i incluso si la distancia entre ellos es grande.

   Por el contrario, si γ tiene un valor muy grande entonces la varianza será pequeña, lo que implica que el vector de soporte no tendrá una influencia importante sobre vectores más alejados.

3. IMPLEMENTACIÓN

1. Usando Scikit-learn de Python, implemente (comente su código) una svm que clasifique el conjunto de datos (por definir).

   Implementación:

   Como no se ha incluído el dataset, he propuesto trabajar el presente ejercicio con el siguiente dataset (Iris Flower):
   `https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data`

   Lectura de Dataset

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

url = "https://archive.ics.uci.edu/ml/machine-learning-databases"
      "/iris/iris.data"
# Asignar columnas al Dataset para su posterior lectura
colnames = ['sepal-length', 'sepal-width',
            'petal-length', 'petal-width', 'Class']
# Lectura del Dataset usando pandas
irisdata = pd.read_csv(url, names=colnames)
```

   Exploración de Datos

```
# Dimensiones del Dataset
irisdata.shape
# Descripci n del Dataset
irisdata.head()
```

	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Preprocesamiento

```
# Separaci n de las columnas del dataset para crear los valores de entrada
# y el valor de salida
X = irisdata.drop('Class', axis=1)
y = irisdata['Class']
# Separaci n del dataset en valores de entranemiento y testeo
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

Entrenamiento

```
from sklearn.svm import SVC
# Elecci n del Kernel: linear
svclassifier = SVC(kernel='linear')
# Ejecuci n del entrenamiento
svclassifier.fit(X_train, y_train)
```

Predicción y Validación

```
# Predicci n usando los valores de testeo
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
# Creaci n de la matriz de confusi n
print(confusion_matrix(y_test,y_pred))
# Creaci n de reporte de resultados
print(classification_report(y_test,y_pred))
```

```
[[13  0  0]
 [ 0  8  0]
 [ 0  0  9]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        13
Iris-versicolor       1.00      1.00      1.00         8
 Iris-virginica       1.00      1.00      1.00         9

    avg / total       1.00      1.00      1.00        30
```

2. Experimente y muestre resultados usando diferentes valores para los parámetros de los kernels: lineal, polinomial, gaussiano, y el parámetro C. Los resultados deben ser mostrados en el documento pdf.

Implementación:

Como no se ha incluído el dataset, he propuesto trabajar el presente ejercicio con el siguiente dataset (Iris Flower): `https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data`

Lectura de Dataset

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

url = "https://archive.ics.uci.edu/ml/machine-learning-databases"
      "/iris/iris.data"
# Asignar columnas al Dataset para su posterior lectura
colnames = ['sepal-length', 'sepal-width',
            'petal-length', 'petal-width', 'Class']
# Lectura del Dataset usando pandas
irisdata = pd.read_csv(url, names=colnames)
```

Exploración de Datos

```
# Descripci n del Dataset
bankdata.head()
```

	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Preprocesamiento

```
# Separaci n de las columnas del dataset para crear los valores de entrada
# y el valor de salida
X = irisdata.drop('Class', axis=1)
y = irisdata['Class']
# Separaci n del dataset en valores de entranemiento y testeo
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

Lineal

Entrenamiento

```
from sklearn.svm import SVC
# Elecci n del Kernel: linear Y par metro C=2.0
svclassifier = SVC(kernel='linear', C=2.0)
# Ejecuci n del entrenamiento
svclassifier.fit(X_train, y_train)
```

Predicción y Validación

```
# Predicci n usando los valores de testeo
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
# Creaci n de la matriz de confusi n
print(confusion_matrix(y_test,y_pred))
# Creaci n de reporte de resultados
print(classification_report(y_test,y_pred))
```

```
[[11  0  0]
 [ 0 11  0]
 [ 0  0  8]]
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        11
Iris-versicolor       1.00      1.00      1.00        11
 Iris-virginica       1.00      1.00      1.00         8

    avg / total       1.00      1.00      1.00        30
```

Polinomial

Entrenamiento

```
from sklearn.svm import SVC
# Elecci n del Kernel: poly Y par metro C=0.5
svclassifier = SVC(kernel='poly', degree=8, C=0.5)
# Ejecuci n del entrenamiento
svclassifier.fit(X_train, y_train)
```

Predicción y Validación

```
# Predicci n usando los valores de testeo
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
# Creaci n de la matriz de confusi n
print(confusion_matrix(y_test,y_pred))
# Creaci n de reporte de resultados
print(classification_report(y_test,y_pred))
```

```
[[11  0  0]
 [ 0 11  0]
 [ 0  0  8]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        11
Iris-versicolor       1.00      1.00      1.00        11
 Iris-virginica       1.00      1.00      1.00         8

    avg / total       1.00      1.00      1.00        30
```

Gaussiano

Entrenamiento

```
from sklearn.svm import SVC
# Elecci n del Kernel: rbf
svclassifier = SVC(kernel='rbf')
# Ejecuci n del entrenamiento
svclassifier.fit(X_train, y_train)
```

Predicción y Validación

```
# Predicci n usando los valores de testeo
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
# Creaci n de la matriz de confusi n
print(confusion_matrix(y_test,y_pred))
# Creaci n de reporte de resultados
print(classification_report(y_test,y_pred))
```

```
[[11  0  0]
 [ 0 10  1]
 [ 0  0  8]]
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	11
Iris-versicolor	1.00	0.91	0.95	11
Iris-virginica	0.89	1.00	0.94	8
avg / total	0.97	0.97	0.97	30

3. Dentro de la sección de Implementación incluya una subsección donde indique las instrucciones para ejecutar el código.

   Implementación:

   La presente implementación ha sido realizada en python 3.6 usando anaconda y jupyter-lab como herramientas de virtualización y visualización.

   El notebook desarrollado que incluye todos los pasos descritos, así como los datasets, ha sido publicado en mi cunta de github:

   `https://github.com/dpalominop/SistemasInteligentes/blob/master/semana7/homework/svm.ipynb`