Aprendizaje de Máquina

Machine Learning

Profesor: Víctor Viera Balanta

```
#importa los datasets de sklearn
#Autor Víctor Viera B.
from sklearn import datasets
#importo bibliotecas de datos
import pandas as pd
#importo biblioteca de vectores y matrices
import numpy as np
```

```
[2] #se cargan el dataset de digitos
    digits = datasets.load_digits()
    digits.data
```

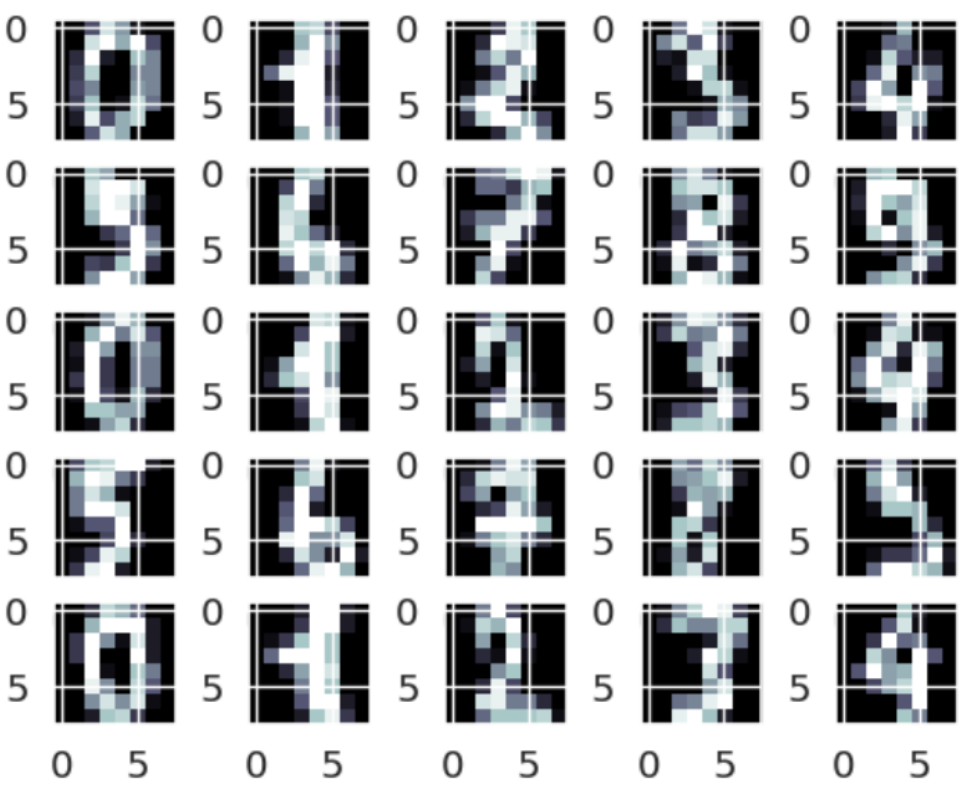
```
[66] #pixeles que conforman los datos
     print(digits.feature_names)
```

```
[67] #la forma de los datos
     digits.data.shape
```

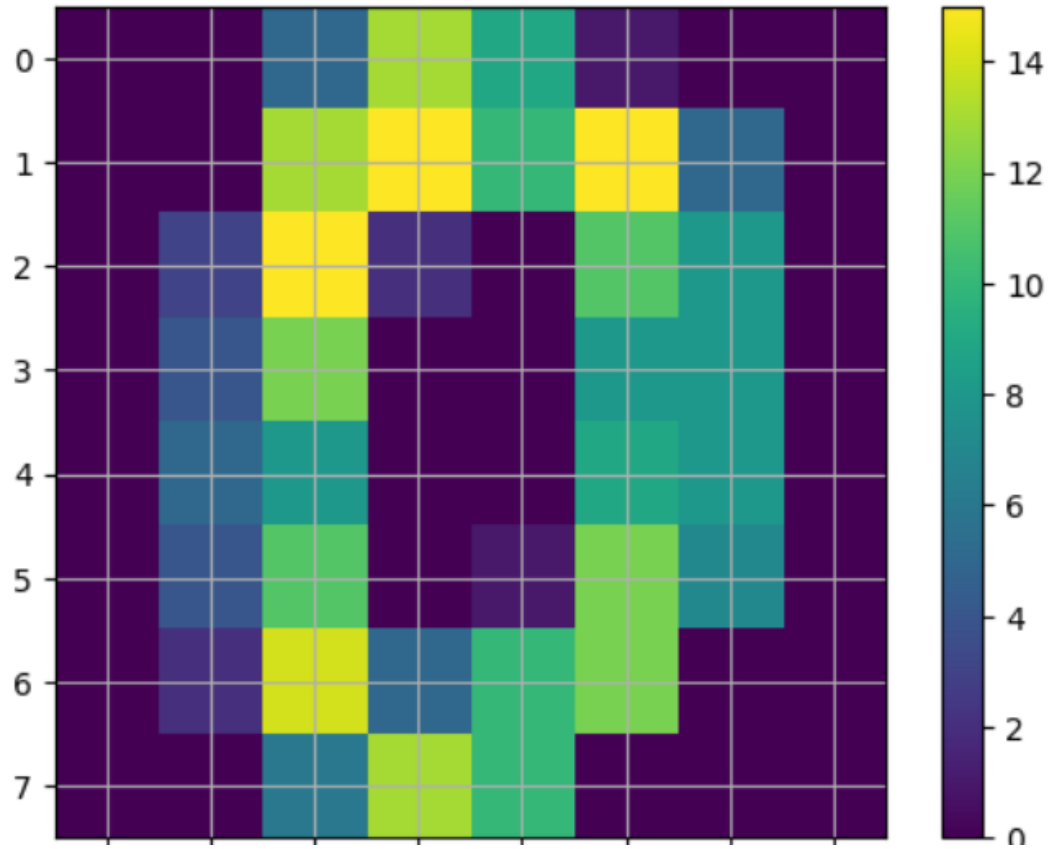
```
[68] #variable dependiente
     digits.target
```

```
[72] #datos de una imagen
     digits.data[0]
```

```python
#se imprimen 5 x 5 datos
import matplotlib.pyplot as plt
fig, ax = plt.subplots(5, 5)
for i, axi in enumerate(ax.flat):
    axi.imshow(digits.images[i], cmap='bone')
```



```python
#se visualizan 2 dígitos
plt.figure()
for i in range(0,2):
  plt.imshow(digits.images[i])
  plt.colorbar()
  plt.grid(True)
  plt.show()
```




```python
[10] #se dividen los datos en prueba y entrenamiento
     from sklearn.model_selection import train_test_split
     train, Xtest, ytrain, ytest = train_test_split(digits.data, digits.target, stratify=digits.target, random_state=42)
```

```python
[14] #se llama al modelo de Knn
     from sklearn.neighbors import KNeighborsClassifier
     neigh = KNeighborsClassifier(n_neighbors=10)
```

```
[13] #se asignan la variables X y
     X= train
     y= ytrain
```

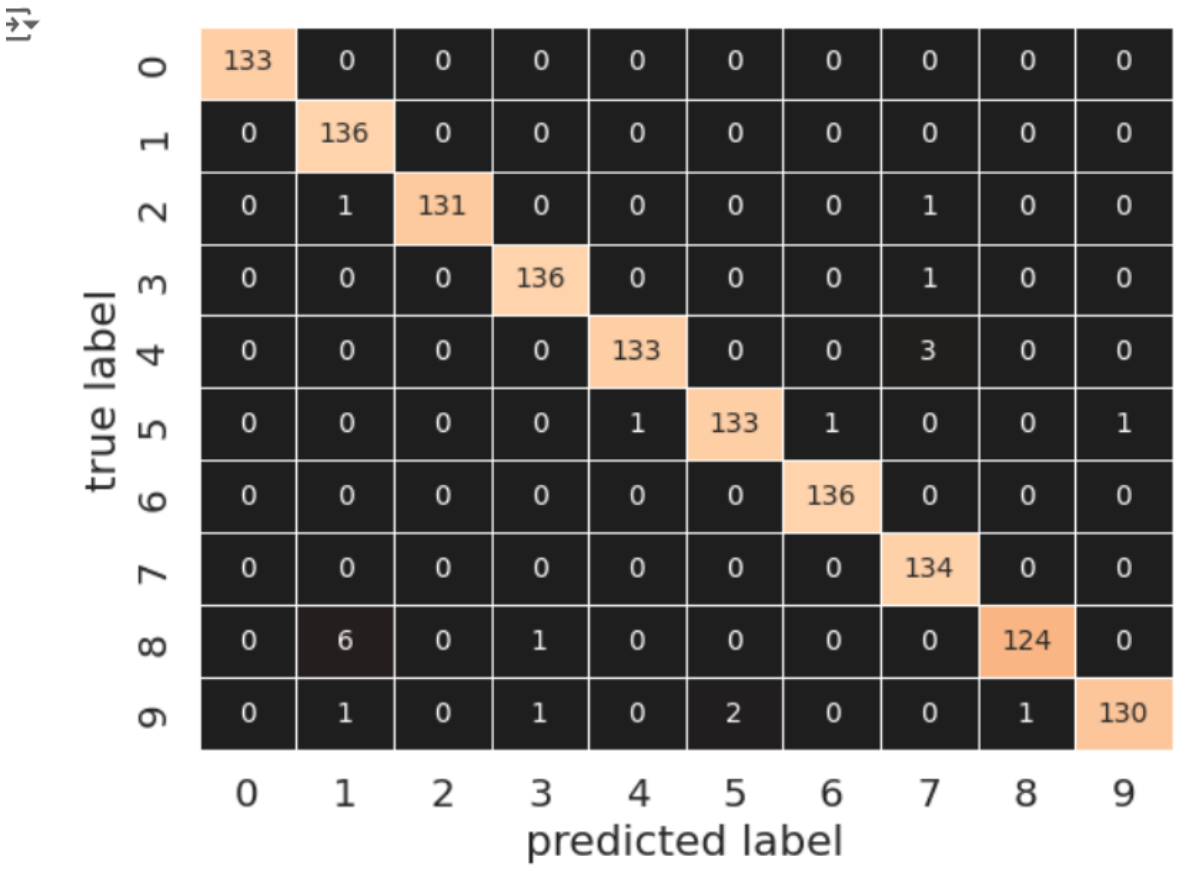
```
     # se entrena el modelo
     neigh.fit(X, y)
```

```
[16] # se realiza la predicción con los datos de entrenamiento
     ypred=neigh.predict(train)
```

```
[37] #se pinta la matriz de confusión
     from sklearn.metrics import confusion_matrix,classification_report
     mx=confusion_matrix(ytrain,ypred)
     mx
```

```
[38] #Graficamos la matriz de confusión
     #para graficar datos
     import matplotlib.pyplot as plt
     import seaborn as sn
     sn.set(font_scale=1.3) # for label size
     #mx=datos(matrix de confusión)
     #annot=True si se escribe del dato en cada celda
     #annot_kws= datos en cada celda y tamaño de la fuente
     #fmt= formato del tipo de dato (f(float),g(general),d(decimal))
     #cbar barras a la derecha de la matriz
     #center tiene que ver con el color (100)
     sn.heatmap(mx, annot=True, annot_kws={"size": 10},fmt='g',center=0,linewidths=0.5,cbar=False) # font size
     plt.ylabel('true label')
     plt.xlabel('predicted label');

     plt.show()
```




```
#se imprine el classification report
print(classification_report(ytrain, ypred))
```

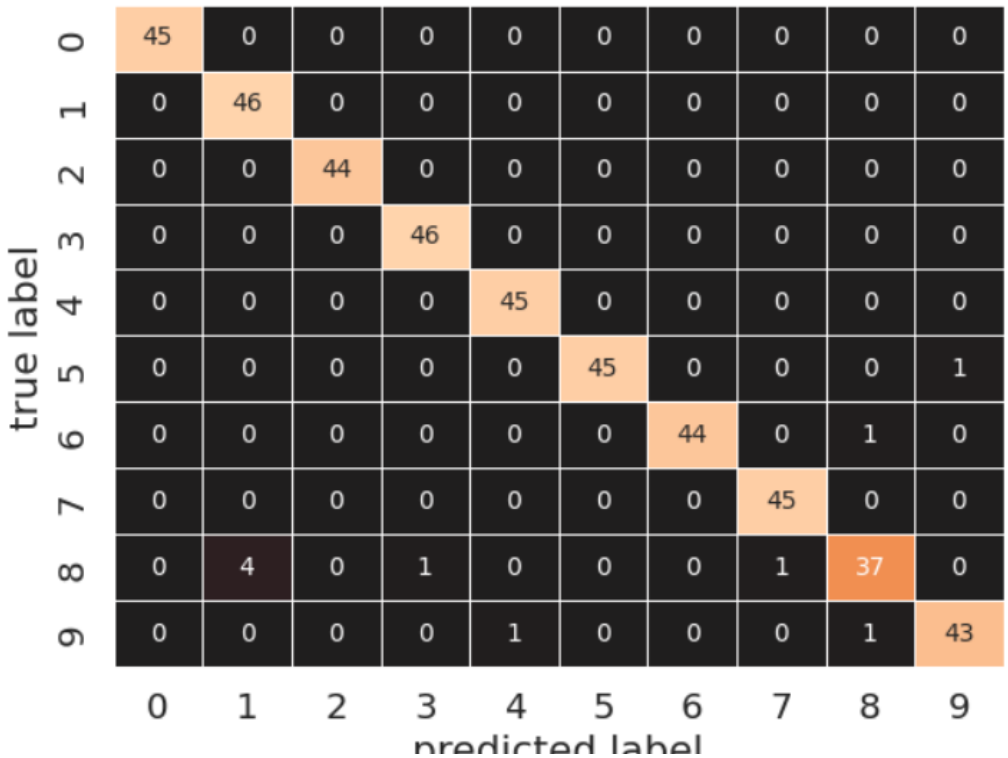
[26]
```
#se realiza la predicción con los datos de prueba
ypredTest=neigh.predict(Xtest)
```

[39]
```
# se calcula la matriz de confusión
mx2=confusion_matrix(ytest, ypredTest)
mx2
```

```
array([[45,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 46,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 44,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 46,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0, 45,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0,  0, 45,  0,  0,  0,  1],
       [ 0,  0,  0,  0,  0,  0, 44,  0,  1,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 45,  0,  0],
       [ 0,  4,  0,  1,  0,  0,  0,  1, 37,  0],
       [ 0,  0,  0,  0,  1,  0,  0,  0,  1, 43]])
```

```python
#Graficamos la matriz de confusión prueba
#para graficar datos
import matplotlib.pyplot as plt
import seaborn as sn
sn.set(font_scale=1.3) # for label size
#mx=datos(matrix de confusión)
#annot=True si se escribe del dato en cada celda
#annot_kws= datos en cada celda y tamaño de la fuente
#fmt= formato del tipo de dato (f(float),g(general),d(decimal))
#cbar barras a la derecha de la matriz
#center tiene que ver con el color (100)
sn.heatmap(mx2, annot=True, annot_kws={"size": 10},fmt='g',center=0,linewidths=0.5,cbar=False) # font size
plt.ylabel('true label')
plt.xlabel('predicted label');

plt.show()
```

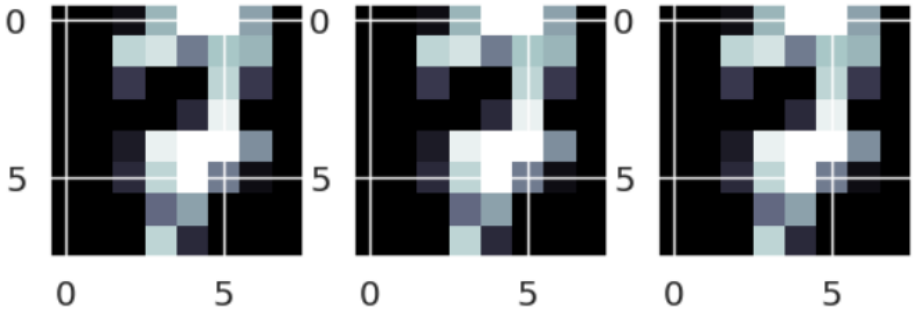



```
[42] print(classification_report(ytest, ypredTest))
```

```
#imágenes de8x8, la imagen No 300
digits.images[300].shape
```

```
(8, 8)
```

```
[61] #se imprime la imagen
import matplotlib.pyplot as plt
fig, ax = plt.subplots(1, 3)
for i, axi in enumerate(ax.flat):
    axi.imshow(digits.images[300], cmap='bone')
```



[76]
```
#se realiza la predicción de un dato nuevo
ypredNumero=neigh.predict([digits.data[300]])
ypredNumero
```

```
array([7])
```

```
#se imprime el dato nuevo
plt.figure()
plt.imshow(digits.images[300])
plt.colorbar()
plt.grid(True)
plt.show()
```