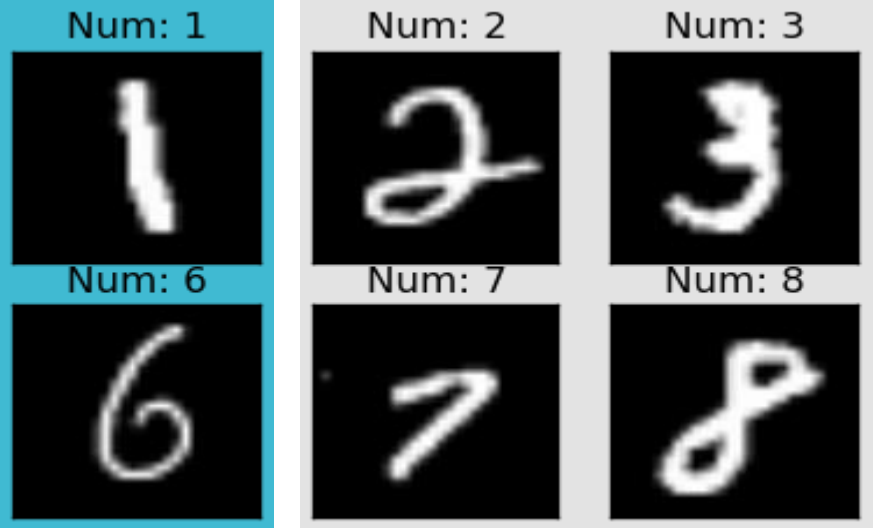
RECONOCIMIENTO DE IMAGENES (DIGITOS)

Test DataTeam
Braulio E. Franco Garcia



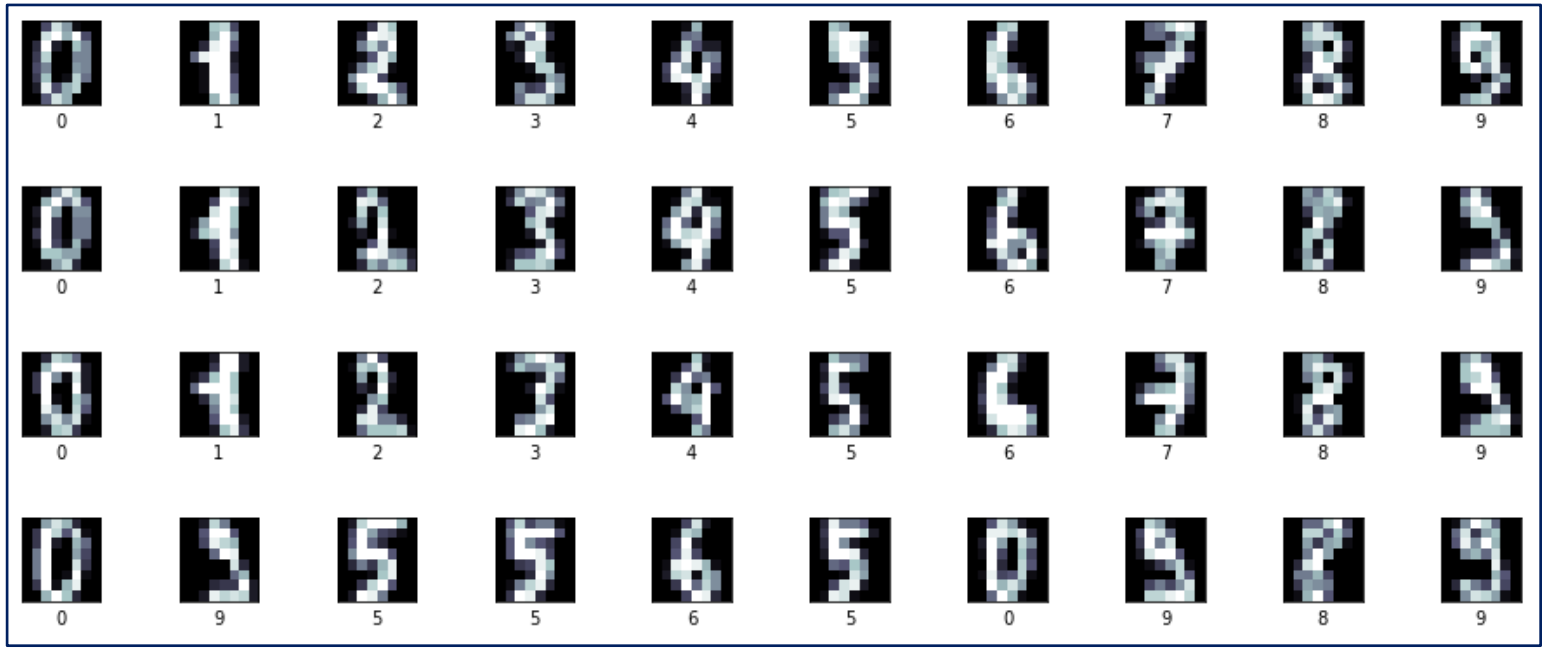
DATOS

Total 1797 imagenes de 8x8 pixeles.
Etiquetas del 0 al 9
Rango de pixeles 0 – 16

```
#descarga del data set
from sklearn.datasets import load_digits
digits = load_digits()
#descripcion del data set
digits.DESCR
#estructura y etiquetas de imagenes
digits.images.shape
for num in digits.target_names:
    print(num)
```

```
#Visualizacion de 40 imagenes con
su etiqueta
fig = plt.figure(figsize = (12,12))
fig.subplots_adjust(left=0, right=1,
bottom=0,top=1,
hspace=1,wspace=1)
for i in range(40):
    im = fig.add_subplot(10,10,i+1)
    im.imshow(digits.images[i],
cmap=plt.cm.bone)

im.set(xticks=[],yticks=[],xlabel=dig
its.target_names[digits.target[i]])
```

Visualizan 40 imagenes.



Separacion de dataset

Conjunto de entrenamiento (80%)
Conjutno de prueba (20%)

```
from sklearn.model_selection import train_test_split #forma aleatoria

X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target,
test_size=0.2, random_state=42)
```

ENTRENAMIENTO DE LA MAQUINA DE SOPORTE VECTORIAL

Se eligió este algoritmo debido a diversas referencias que indicant que es de los mejores para clasificar imagenes.
Kernel = linear por overfitting.

```
#entrenamiento del algoritmo Maquina de Soporte Vectorial
from sklearn.svm import SVC #clasificacion

#se toma un kernel lineal
svc = SVC(kernel="linear", class_weight = "balanced")
svc.fit(X_train, y_train)
```

VALIDACION

Validacion cruzada K-fold (5)

```
from sklearn.model_selection import cross_validate
from scipy.stats import sem
from sklearn.model_selection import cross_val_score

scores = cross_val_score(svc, X_train, y_train, cv=5)
print(scores)
```

[0.97569444 0.97222222 0.96515679 0.9825784 0.95818815]

```
print(np.mean(scores))
```

0.9707680023

REPORTE DE CLASIFICACION

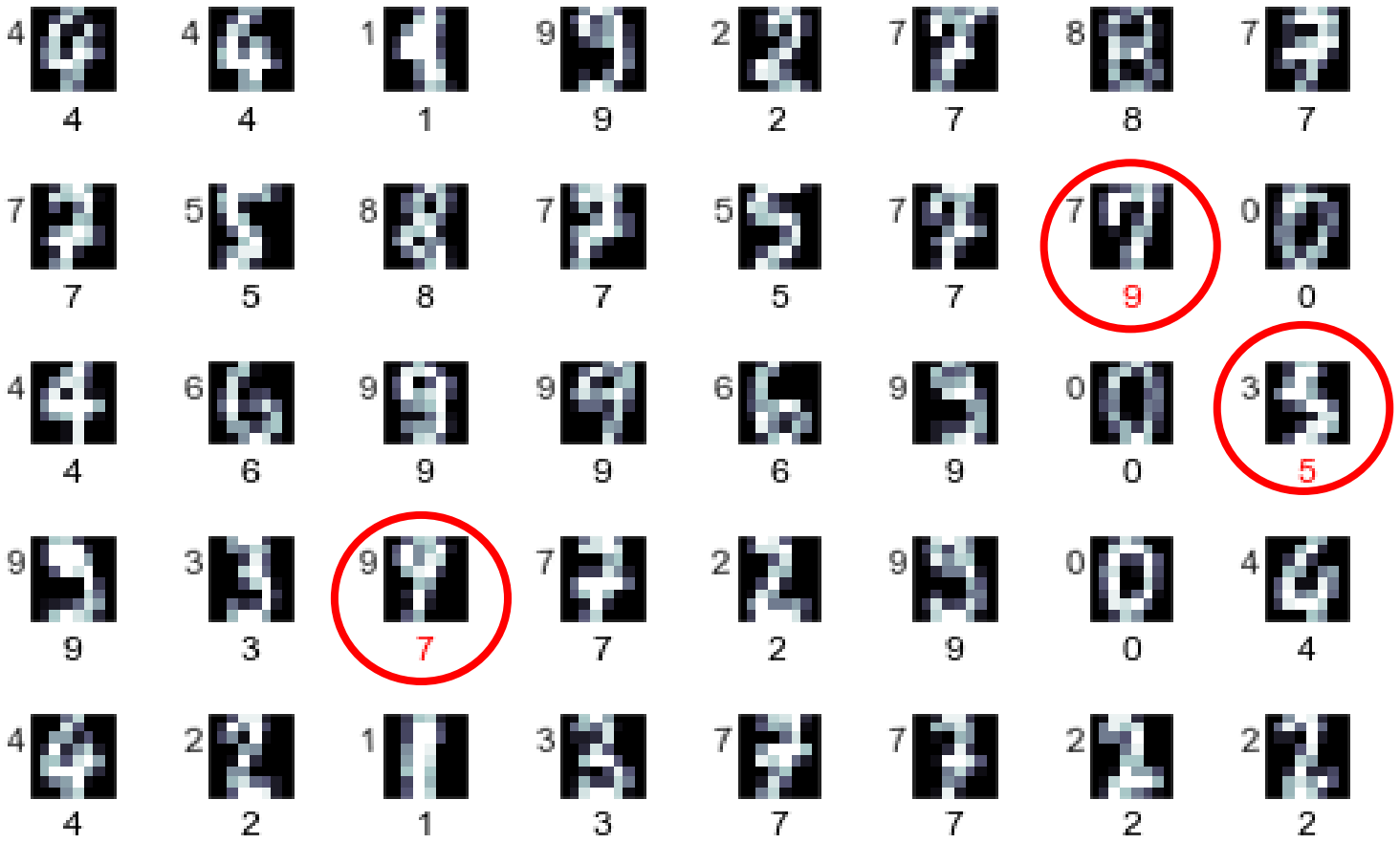
```
#reporte de clasificacion
y_pred = svc.predict(X_test)
test_accuracy =
accuracy_score(y_test,
y_pred)

print('Exactitud en el
conjunto de prueba:
{}'.format(test_accuracy))

print(classification_report(y_t
est, y_pred))
```

Exactitud en el conjunto de prueba: 0.9777777777777777

	precision	recall	f1-score	support
0	1.00	1.00	1.00	33
1	0.97	1.00	0.98	28
2	1.00	1.00	1.00	33
3	0.97	0.94	0.96	34
4	0.98	0.98	0.98	46
5	0.96	1.00	0.98	47
6	1.00	1.00	1.00	35
7	0.97	0.97	0.97	34
8	1.00	0.97	0.98	30
9	0.95	0.93	0.94	40
accuracy			0.98	360
macro avg	0.98	0.98	0.98	360
weighted avg	0.98	0.98	0.98	360

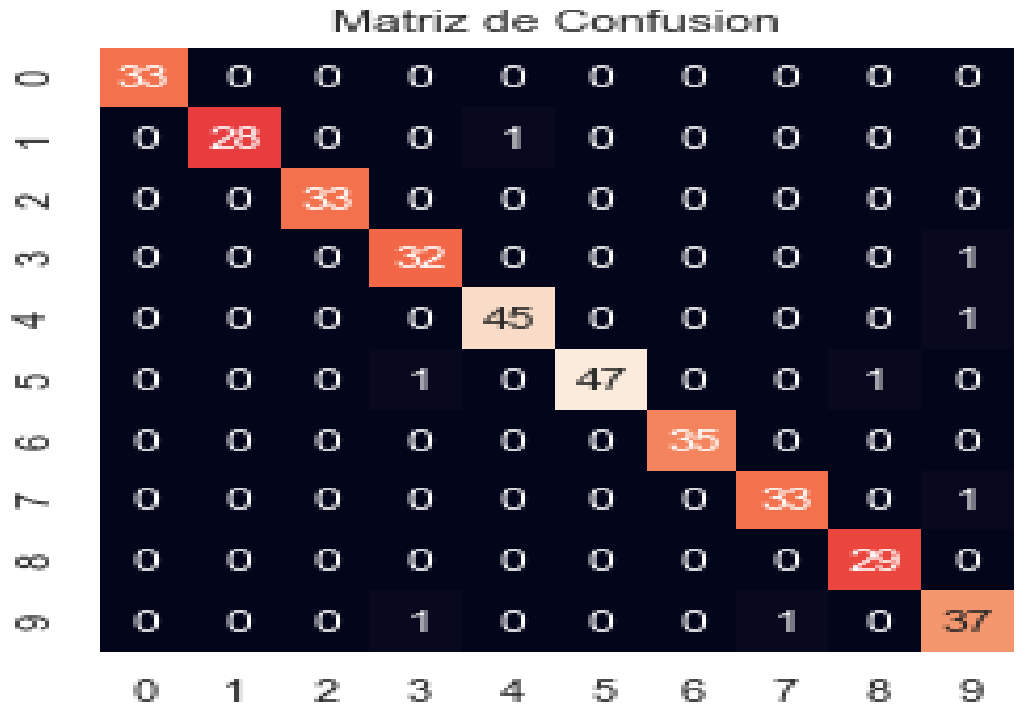
MUESTREO DE PREDICCIÓN

```
fig = plt.figure(figsize = (12,12))
fig.subplots_adjust(left=0, right=1,
bottom=0,top=1,
hspace=1,wspace=1)

for i in range(200):
    im = fig.add_subplot(15,15,i+1)

im.imshow(X_test[i].reshape(8,8),
cmap=plt.cm.bone)
    im.set(xticks=[],yticks=[])

im.set_xlabel(digits.target_names[
y_pred[i]],
                color = 'black' if y_pred[i]
== y_test[i] else 'red')

im.set_ylabel(digits.target_names[
y_test[i]],rotation= math.pi)
```



MATRIZ DE CONFUSIÓN

```
mat = confusion_matrix(y_test, y_pred)

import seaborn as sns; sns.set()
plt.title('Matriz de Confusion')
sns.heatmap(mat.T, square=True, annot=True, fmt="d", cbar=False,
        xticklabels=digits.target_names, yticklabels=digits.target_names)
```



¿QUÉ MÁS?

Saber el valor de una única imagen, saber cuáles imágenes corresponden a un determinado dígito.

Se puede abordar el problema con otro algoritmo de clasificación como Naive Bayes, modificar el Kernel, o implementar Redes neuronales.

UNA IMAGEN

Implementación con Naive Bayes

```
#arreglos auxiliares
X, y = digits["data"], digits["target"]

#valores booleanos segun sea o no 7.
y_train_7 = (y_train == 7)
y_test_7 = (y_test == 7)

#entrena algoritmo Naive Bayes (valores discretos)
from sklearn.naive_bayes import MultinomialNB
mnb = MultinomialNB()
mnb.fit(X_train, y_train_7)

digito = X[17]
mnb.predict([digito])
array([ True])
```

UN DIGITO

Implementacion con Naive Bayes

```
y_pred = []
for dig in X_test:
    y_pred = np.append(y_pred, mnb.predict([dig]))
```

VALIDACION

Reporte de clasificación

```
print(classification_report(y_test_7,y_pred))
```

	precision	recall	f1-score	support
False	1.00	0.98	0.99	326
True	0.87	0.97	0.92	34
accuracy		0.98	360	
macro avg	0.93	0.98	0.95	360
weighted avg	0.98	0.98	0.98	360

Matriz de confusion

```
print(pd.crosstab(y_test_si,y_pred_si,rownames=['True'],colnames=['Predicted'],margins=True))
```

Predicted	nosiete	siete	All
True			
nosiete	321	5	326
siete	1	33	34
All	322	38	360

GRACIAS