

Curso de IA

# Capítulo 5. Cuestionario

Para estudiantes

© 2023 SAMSUNG. Todos los derechos reservados.

La Oficina de Ciudadanía Corporativa de Samsung Electronics posee los derechos de autor de este documento.

Este documento es una propiedad literaria protegida por la ley de derechos de autor, por lo que está prohibida su reimpresión y reproducción sin permiso.

Para utilizar este documento fuera del plan de estudios de Samsung Innovation Campus, debe recibir el consentimiento por escrito del titular de los derechos de autor..

1. ¿Cuál de las siguientes declaraciones sobre el aprendizaje supervisado es incorrecta?

- 1 El aprendizaje supervisado consiste en realizar predicciones correctamente con datos dados a partir de datos de capacitación.
- 2 La precisión es la relación entre lo que el modelo clasifica como verdadero y lo que realmente clasifica como verdadero. Puede denominarse tasa de respuesta.
- 3 La tasa de recuperación es la relación entre lo que el modelo predice como verdadero y lo que es realmente verdadero.
- 4 La tasa de recuperación se calcula como  $\frac{tp}{tp+fp}$ . (donde tp es verdadero positivo y fp es falso positivo)

2. ¿Cuál es el método de estimación de los coeficientes de regresión en el análisis de regresión?

R= Método de mínimos cuadrados

3. ¿Cuál de las siguientes declaraciones sobre el sobreajuste es incorrecta?

- 1 Una característica de un patrón encontrado en un conjunto de datos específico que sólo existe en ese conjunto de datos, o una característica de un patrón que no se generaliza y sólo existe en unos datos específicos se denomina sobreajuste.
- 2 Los métodos de extracción utilizados para resolver el problema del sobreajuste incluyen el método holdout, el método de validación cruzada, el método bootstrap, etc.
- 3 En el método holdout, la validación cruzada se realiza dividiendo los datos de capacitación para el aprendizaje y la construcción del modelo y los datos de verificación para la evaluación del cumplimiento, y los resultados de los datos de verificación se utilizan únicamente para la medición del cumplimiento sin afectar al modelo.
- 4 El método de validación cruzada es un método para volver a seleccionar los datos de capacitación repetidamente y basado en la extracción de restauración, y es adecuado cuando la cantidad total de datos no es grande.

4. ¿Cuál de las siguientes afirmaciones sobre el modelo de árbol de decisión es incorrecta?
- 1 Los árboles de decisión se utilizan para la predicción de la quiebra de empresas, la predicción del precio máximo de las acciones, la predicción de la tasa de cambio y la predicción de las perspectivas económicas.
  - 2 El algoritmo CHAID (Detección Automática de Interacción Chi-cuadrado) es un algoritmo que realiza la separación utilizando el chi-cuadrado o la prueba F.
  - 3 El algoritmo CART (Árboles de clasificación y regresión) es un algoritmo que realiza la separación utilizando el índice de Gini, y 100 representa un nodo perfectamente puro entre 0 y 100.
  - 4 El índice de entropía del algoritmo C4.5 puede obtenerse utilizando el estadístico de prueba de razón de verosimilitud en la distribución polinómica y restando la entropía del nodo hijo de la entropía del nodo padre.
5. Cuando he entrenado un clasificador SVM utilizando el núcleo RBF, parece que no se ajusta al conjunto de capacitación. ¿Deberíamos aumentar gamma o disminuir gamma? ¿Qué pasa con C?
- R= Aumenta gamma para que el modelo se ajuste a los datos y aumentar C para reducir la tolerancia a errores, pero debemos tener en cuenta el sobre ajuste para no aumentarlos demasiado.
6. Utilice el algoritmo KNN para clasificar los datos del código siguiente y calcule la exactitud, la llamada y la precisión. Establezca el valor de adyacencia en 5.

```
importar numpy como np
desde sklearn.datasets importar load_breast_cancer
desde sklearn.model_selection importar train_test_split, GridSearchCV
desde sklearn.neighbors importar KNeighborsClassifier
desde sklearn importar métricas, preprocesamiento

# Cargar datos
data = load_breast_cancer()
X = data['data']
Y = 1 - data['target'] # 0 = 'benign' y 1 = malignant.

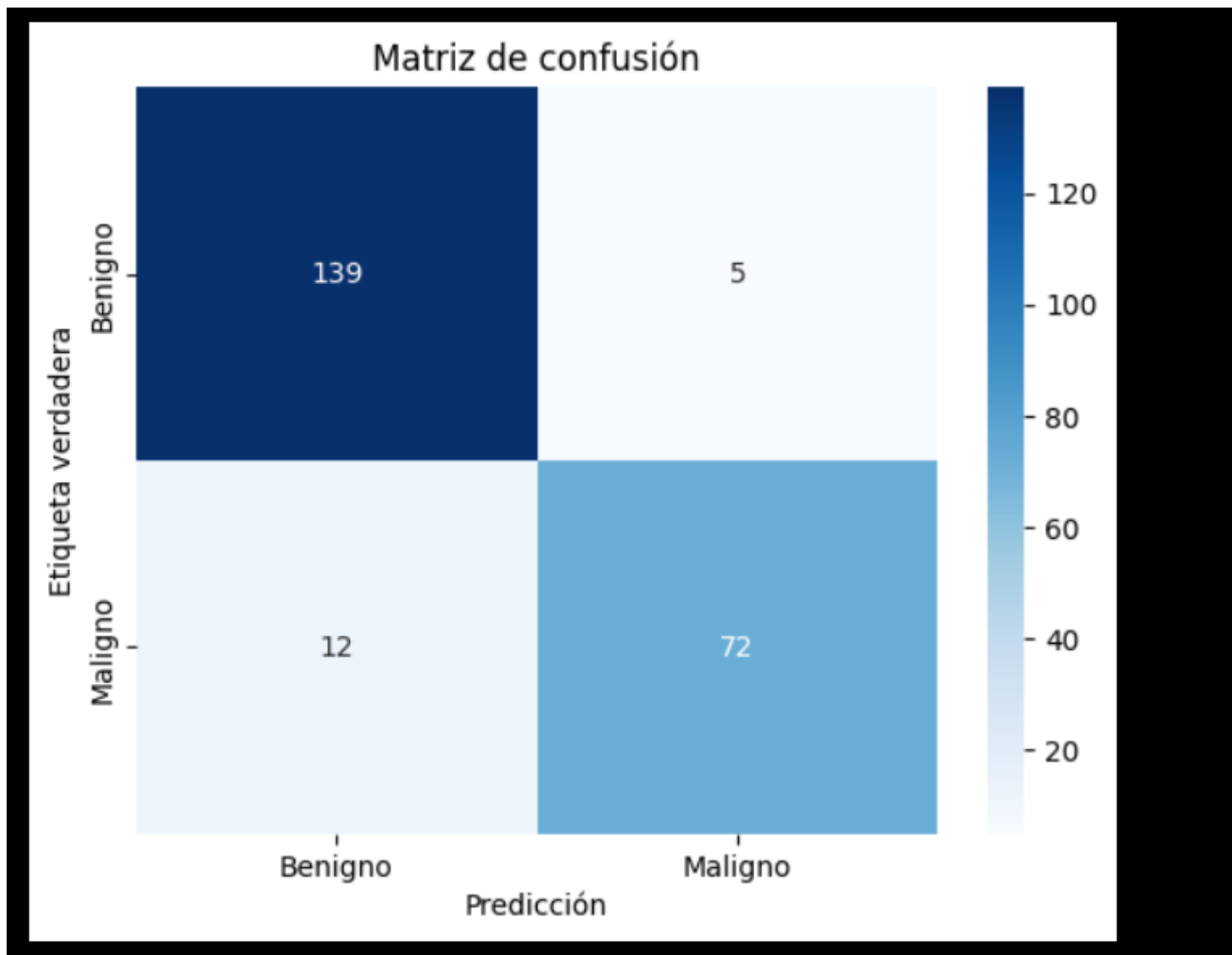
# Dividir el conjunto de datos en capacitación y pruebas.
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=1234)
```

```
> ✓
24] print("\nReporte de clasificación:")
    print(classification_report(Y_test, Y_pred, target_names=['Benigno','Maligno']))
```

✓ 0.0s

Reporte de clasificación:

	precision	recall	f1-score	support
Benigno	0.92	0.97	0.94	144
Maligno	0.94	0.86	0.89	84
accuracy			0.93	228
macro avg	0.93	0.91	0.92	228
weighted avg	0.93	0.93	0.92	228



```

C:\Users\oscar\Desktop\UPIITA\10mo semestre\SAMSUNG\Ejercicios\5.4_SIC_DT.ipynb
+ Code + Markdown

import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics, preprocessing

import pandas as pd
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
# Cargar datos
data = load_breast_cancer()

[17] ✓ 0.0s Python

X = data['data']
Y = 1 - data['target'] # 0 = 'benign' y 1 = malignant.

# Dividir el conjunto de datos en capacitación y pruebas.
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=1234)

[18] ✓ 0.0s Python

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

[19] ✓ 0.0s Python
    
```

```

acces = []
k_grid = range(1,100,1)
for k in k_grid:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, Y_train)
    Y_pred = knn.predict(X_test)
    acces.append(accuracy_score(Y_test,Y_pred))

# Selección de k óptimo y evaluación final
optimal_k = k_grid[np.argmax(acces)]
print(f"El valor óptimo de k es: {optimal_k}")
#para el ejercicio nos piden k5 entonces...
ejerk=5

#reemplazamos ejer k por optimal_k para tener el mejor numero de vecinos y scalar a los entrenamientos
#knn = KNeighborsClassifier(n_neighbors=optimal_k)
#knn.fit(X_train_scaled, Y_train)
#Y_pred = knn.predict(X_test_scaled)
knn = KNeighborsClassifier(n_neighbors=ejerk)
knn.fit(X_train, Y_train)
Y_pred = knn.predict(X_test)

# Matriz de confusión
conf_matrix = confusion_matrix(Y_test, Y_pred)

```

El valor óptimo de k es: 3



```

print("\nReporte de clasificación:")
print(classification_report(Y_test, Y_pred, target_names=['Benigno','Maligno']))
exactitud = metrics.accuracy_score(Y_test, Y_pred)
precision = metrics.precision_score(Y_test, Y_pred)
llamada = metrics.recall_score(Y_test, Y_pred)
print("exactitud= ",exactitud)
print("precision= ",precision)
print("llamada= ",llamada)

```

[35] ✓ 0.0s Python

...

Reporte de clasificación:				
	precision	recall	f1-score	support
Benigno	0.92	0.97	0.94	144
Maligno	0.94	0.86	0.89	84
accuracy			0.93	228
macro avg	0.93	0.91	0.92	228
weighted avg	0.93	0.93	0.92	228

exactitud= 0.9254385964912281  
 precision= 0.935064935064935  
 llamada= 0.8571428571428571