

RegresionLogistica - Velásquez Luna Elí Jafet

May 25, 2024

1 EJERCICIO

```
[29]: from sklearn.datasets import load_breast_cancer
      from sklearn.linear_model import LogisticRegression
      from sklearn.model_selection import train_test_split
      from sklearn.preprocessing import StandardScaler
      import pandas as pd
      import numpy as np

      tabla = load_breast_cancer(as_frame=True)

      # Separación de etiquetas
      X = tabla.data
      y = tabla.target
```

```
[30]: # Estandarizar valores de las variables
      scaler = StandardScaler()
      X = scaler.fit_transform(X)
```

1.1 Selección de 5 características de entrada

```
[31]: # Ver cuántas características tiene
      X.shape
```

```
[31]: (569, 30)
```

```
[22]: # Se seleccionan 5 al azar
      id_features = np.random.choice(30, 5, replace=False)
```

```
[52]: id_features
```

```
[52]: array([10, 27, 25, 18, 24])
```

```
[32]: # Se filtran los datos
      X = X[:,id_features]
```

1.2 Modelo

```
[33]: # Separación del dataset
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
```

```
[34]: # Definición del modelo y ajuste
modelo = LogisticRegression()
modelo.fit(X_train,y_train)
```

```
[34]: LogisticRegression()
```

```
[35]: # Predicción de etiquetas
y_pred = modelo.predict(X_test)
y_pred
```

```
[35]: array([1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0,
        1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1,
        1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1,
        1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1,
        0, 1, 1, 0])
```

Estas son las etiquetas predichas para los 0.2×569 datos de entrenamiento. En el siguiente ejercicio se usarán métricas para determinar qué tan buena fue la clasificación.

2 EJERCICIO 2: Métricas

- Calcular
- Interpretar

CLAVES:

1 - Benigno

0 - Maligno

2.1 Matriz de Confusión

```
[36]: from sklearn.metrics import confusion_matrix

# Matriz de Confusión
confusion_matrix(y_test,y_pred)
```

```
[36]: array([[40,  6],
        [ 1, 67]])
```

La matriz indica que hay 40 verdaderos negativos y 64 verdaderos positivos. Es decir, predijo correctamente 107 de los diagnósticos de cáncer de mama.

Mientras que hay considerablemente pocos falsos negativos y falsos positivos, con 1 y 6 respectivamente.

2.2 Exactitud (Accuracy)

```
[ ]: from sklearn.metrics import accuracy_score  
  
accuracy_score(y_test,y_pred)
```

```
[ ]: 0.9912280701754386
```

El modelo predijo correctamente las 2 categorías del dataset (cáncer de mama o no) en un 99.12%. Este es el porcentaje de valores en la diagonal, los que fueron categorizados correctamente)

2.3 Precisión (Precision)

```
[45]: from sklearn.metrics import precision_score  
  
precision_score(y_test,y_pred, pos_label=1)
```

```
[45]: 0.9178082191780822
```

Esta métrica indica que el 91.78% de las ocasiones cuando el modelo predice que el individuo está sano de cancer de mama, son correctas.

2.4 Exhaustividad (Recall)

```
[47]: from sklearn.metrics import recall_score  
  
recall_score(y_test,y_pred,pos_label=1)
```

```
[47]: 0.9852941176470589
```

Esta métrica indica que en el 98.53% de los casos, se predijo un cancer benigno cuando realmente era benigno, del total de muestras con resultado benigno.

2.5 F1-score

```
[51]: from sklearn.metrics import f1_score  
  
f1_score(y_test,y_pred,pos_label=1)
```

```
[51]: 0.9503546099290779
```

Es una métrica de equilibrio, en este caso para la clase 1 (cáncer benigno). Por lo que podría concluir que el modelo lo predice en un ~95.04%.