Nombres: Andres Leon, Samuel Tarazona

Materia: Introducción a Machine Learning – 802

Docente: Alexander Espinosa Garcia

Actividad: Clasificación de plantas usando regresión lineal

Enlace Github:
https://github.com/03SamuelTarazona/MachineLearning/blob/main/clasificacionIris.py

Descripción del Procedimiento

1. **Carga del dataset Iris**
    - Se utilizó el conjunto de datos **Iris** que viene incluido en la librería scikit-learn.
    - Este dataset contiene 150 muestras de flores divididas en **tres especies**: *Setosa*, *Versicolor* y *Virginica*.
    - Cada flor tiene **4 características** medidas en centímetros:
        - Longitud del sépalo.
        - Ancho del sépalo.
        - Longitud del pétalo.
        - Ancho del pétalo.
2. **Exploración inicial de los datos**
    - Los datos se organizaron en un **DataFrame de Pandas** para facilitar su análisis.
    - Se observaron las primeras filas para confirmar que los valores eran correctos y que cada fila correspondía a una flor con sus características y su especie.
3. **Preparación de los datos**
    - Se definieron las **variables de entrada (X)** que corresponden a las características de las flores.
    - Se definió la **variable objetivo (y)** que indica a qué especie pertenece cada flor.

- Los datos se dividieron en dos grupos:
    - **Entrenamiento (80%)**: para que los modelos aprendan.
    - **Prueba (20%)**: para evaluar qué tan bien funciona el modelo con datos nuevos.

4. **Entrenamiento de modelos**
   Se probaron tres algoritmos distintos, todos de scikit-learn:
    - **Regresión Lineal básica**: aunque no está pensada directamente para clasificación, se usó como experimento. Para convertir sus resultados en clases, se redondearon los valores al número entero más cercano (0, 1 o 2).
    - **RidgeClassifier**: es una variante de la regresión lineal adaptada para clasificación y con mejor estabilidad.
    - **Regresión Logística**: modelo especialmente diseñado para clasificación, que calcula la probabilidad de que una flor pertenezca a cada clase y elige la más probable.
5. **Evaluación de resultados**
    - Se midió la **precisión (accuracy)**, que indica el porcentaje de aciertos del modelo.
    - Se generaron reportes que muestran:
        - **Precisión (precision)**: cuántas predicciones positivas fueron correctas.
        - **Cobertura (recall)**: cuántas de las flores reales de una clase fueron correctamente identificadas.
        - **F1-score**: un equilibrio entre precisión y cobertura.

Los resultados obtenidos fueron:

- **Regresión Lineal básica**: 100% de aciertos.
- **RidgeClassifier**: 90% de aciertos.
- **Regresión Logística**: 100% de aciertos.

6. **Interpretación de coeficientes**

- Con el modelo de Regresión Logística se analizaron los coeficientes que indican qué tan importantes son las características para distinguir cada especie.
- Por ejemplo:
  - El **largo y ancho del pétalo** son los que más ayudan a diferenciar *Versicolor* y *Virginica*.
  - El **ancho del sépalo** tiene mayor peso en la especie *Setosa*.

7. **Visualización**
   - Se realizó una gráfica en 2D usando dos características (petal length y petal width).
   - Los puntos se colorearon según la especie a la que pertenecen.
   - La gráfica muestra cómo las tres especies se agrupan de forma clara y distinta, lo cual explica por qué los modelos logran clasificar con tanta precisión.

Descripción del Algoritmo

El algoritmo principal implementado fue la **Regresión Logística**, utilizando la librería scikit-learn. A continuación, se explica paso a paso el código empleado:

1. Importación de librerías

```
import pandas as pd
from sklearn import datasets
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
```

Aquí se cargan las librerías necesarias:

- **pandas** para organizar los datos en tablas.
- **datasets** de scikit-learn para acceder al dataset Iris.
- **LogisticRegression** para crear el modelo de clasificación.
- **train_test_split** para dividir los datos en entrenamiento y prueba.

- **classification_report** y **accuracy_score** para evaluar los resultados del modelo.

2. Carga del dataset Iris

```
iris = datasets.load_iris()
X = iris.data
y = iris.target
```

En este fragmento:

- iris.data contiene las 4 características (largo y ancho de sépalos y pétalos).
- iris.target contiene la clase a la que pertenece cada flor (0 = Setosa, 1 = Versicolor, 2 = Virginica).

3. División de datos en entrenamiento y prueba

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

el dataset se divide:

- **80% para entrenamiento** (X_train, y_train).
- **20% para prueba** (X_test, y_test).
  Esto asegura que el modelo se entrene con un grupo de datos y se evalúe con otro diferente.

4. Creación y entrenamiento del modelo

```
modelo = LogisticRegression(max_iter=200)
modelo.fit(X_train, y_train)
```

Explicación:

- Se crea el modelo de **Regresión Logística** con max_iter=200 para asegurar que el algoritmo tenga suficientes iteraciones para converger.
- Con fit(), el modelo aprende a partir de los datos de entrenamiento, ajustando sus parámetros internos.

5. Predicciones

```
y_pred = modelo.predict(X_test)
```

Aquí el modelo usa lo aprendido para predecir las clases de las flores en el conjunto de prueba.

6. Evaluación del modelo

```
print("Precisión:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Se muestran métricas de desempeño:

- **Precisión (accuracy):** porcentaje total de aciertos.
- **Reporte de clasificación:** incluye precisión, recall y F1-score por cada especie de flor.

En este caso, el modelo alcanzó un **100% de aciertos**.

7. Interpretación de coeficientes

```
print("Coeficientes del modelo:", modelo.coef_)
print("Intercepto:", modelo.intercept_)
```

Estos valores indican la importancia de cada característica en la clasificación.

- Se observó que el **largo y ancho del pétalo** son las características más relevantes para diferenciar *Versicolor* y *Virginica*.
- El **ancho del sépalo** es más importante para identificar *Setosa*.

Ventajas del algoritmo de Regresión Logística:

- Es rápido y eficiente.
- Funciona muy bien con problemas de clasificación como el dataset Iris.
- Permite interpretar qué características son más importantes para diferenciar entre las clases.

Conclusión

El experimento demostró que la **Regresión Logística** es el modelo más adecuado para este problema, logrando un **100% de aciertos** en la clasificación de las flores Iris.
Aunque la Regresión Lineal básica también alcanzó buenos resultados en este caso, la Regresión Logística es más confiable porque está diseñada específicamente para clasificación.

Además, el análisis de los coeficientes confirmó que las características más determinantes son el **largo y ancho del pétalo**, seguidos por algunas medidas del sépalo.