Actividad de Aprendizaje 6-Análisis de datos(Proyecto de Aplicación)

Mónica Fernanda Rojas Celis

Yeimy Vanessa Ricardo Ballestas

Daniel Gómez Pérez

Servicio Nacional de Aprendizaje

Tecnología en Análisis y desarrollo de software – Ficha 2828523

Medellín, 10 de abril de 2025

Tabla de Contenido

1. Introducción

El conjunto de datos de Iris es un agrupación de mediciones de flores de iris que se utiliza en el aprendizaje automático y el análisis estadístico. Contiene información sobre la longitud y anchura de los pétalos y sépalos de cada flor, así como su especie. Algunas de sus características es que consta de 150 observaciones de flores de iris, incluye las especies Iris setosa, Iris virginica e Iris versicolor y se utiliza para construir modelos que puedan predecir la especie de una flor de iris.

Esta colección de datos ha sido ampliamente utilizada en el campo de la estadística y el aprendizaje automático; su creación fue en 1936 por el estadístico británico y biólogo Ronald Fisher. Esta base de datos es conocida por su utilidad en la clasificación de diferentes especies de plantas iris, existen tres tipos de plantas iris, esta base de datos nos brinda la cifra del largo y ancho, ya sea del pétalo o el sépalo, de esta manera buscamos ver la proporción de cada planta.

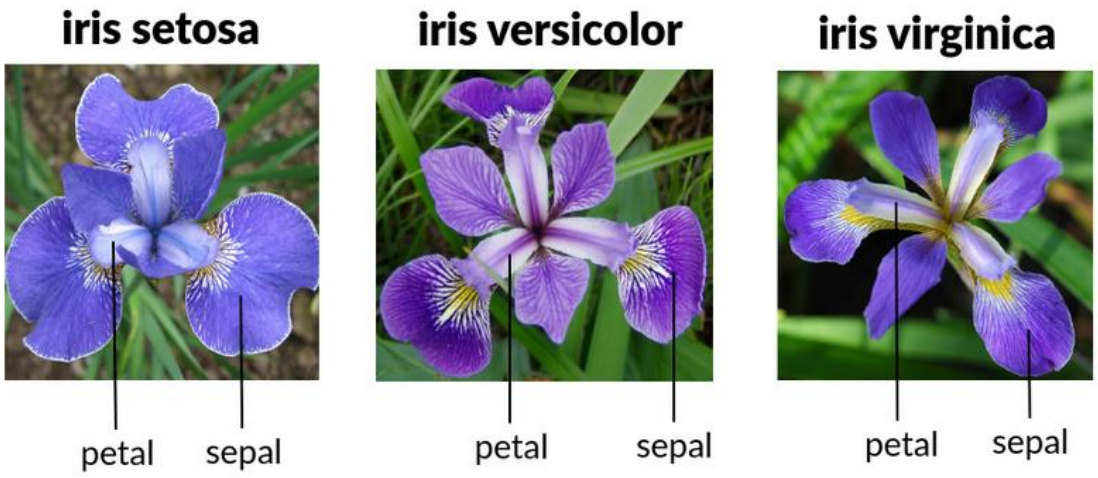


Imagen tomada de: https://www.kaggle.com/datasets/taweilo/iris-dataset-elarged-with-smote

Fisher, seleccionó tres especies de iris para incluir en la base de datos: Iris setosa, Iris versicolor e Iris virginica. las cuales fueron elegidas porque tenían grandes diferencias en sus características morfológicas. Para cada especie o tipo de flor, Fisher tomó medidas de cuatro características de las plantas iris como la longitud del sépalo, el ancho del sépalo, la longitud del pétalo y el ancho del pétalo en centímetros para facilitar comparar cada planta.

2. Objetivo del proyecto

El objetivo de este proyecto es utilizar el conjunto de datos de Iris para predecir la especie de la flor de Iris en función de las características de la flor, como el largo y ancho del sépalo y pétalo. Se utilizará un modelo de regresión logística, ya que es uno de los modelos más más utilizados para problemas de clasificación.

3. Cargar la información (conjunto de datos)

3.1 Importamos las librerías necesarias

```
1  import pandas as pd
2  from sklearn import datasets
3  import numpy as np
4  import matplotlib.pyplot as plt
5  from sklearn.preprocessing import LabelEncoder
6  from sklearn.model_selection import train_test_split
7  from sklearn.linear_model import LogisticRegression
8  from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
9
```

3.2 Cargar y explorar el conjunto de datos

El conjunto de datos de Iris está disponible a través de la librería scikit-learn.

```
12   iris = datasets.load_iris()
13
```

Se convierte en un dataframe de pandas

```
15
16  df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
17  df['species'] = pd.Categorical.from_codes(iris.target, iris.target_names)
18
```

Se visualizan las primeras filas del conjunto de datos

```
20
21   print(df.head())
22
```

```
danie@LAPTOP-0MT3MCT1 MINGW64 ~/Downloads/conjunto iris
$ python app.py
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm) species
0                5.1               3.5                1.4               0.2  setosa
1                4.9               3.0                1.4               0.2  setosa
2                4.7               3.2                1.3               0.2  setosa
3                4.6               3.1                1.5               0.2  setosa
4                5.0               3.6                1.4               0.2  setosa
```

Se preprocesan los datos

```
25  X = df.drop('species', axis=1)
26  y = df['species']
```

```
30  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
31
32  print("Tamaño del conjunto de entrenamiento:", X_train.shape)
33  print("Tamaño del conjunto de prueba:", X_test.shape)
34
```

Se entrena el modelo de regresión logística para predecir las especies de Iris

```
35
36  logreg = LogisticRegression(max_iter=200)
37
38  logreg.fit(X_train, y_train)
39
40  y_pred = logreg.predict(X_test)
41
42  print("Predicciones:", y_pred[:5])
43
```

```
danie@LAPTOP-0MT3MCT1 MINGW64 ~/Downloads/conjunto iris
$ python app.py
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm) species
0                5.1               3.5                1.4              0.2  setosa
1                4.9               3.0                1.4              0.2  setosa
2                4.7               3.2                1.3              0.2  setosa
3                4.6               3.1                1.5              0.2  setosa
4                5.0               3.6                1.4              0.2  setosa
Tamaño del conjunto de entrenamiento: (120, 4)
Tamaño del conjunto de prueba: (30, 4)
Predicciones: ['versicolor' 'setosa' 'virginica' 'versicolor' 'versicolor']
```

3.3 Evaluar el modelo

Primero se calcula la precisión del modelo, que nos muestra la proporción de predicciones correctas.

```
45  accuracy = accuracy_score(y_test, y_pred)
46  print(f"Precisión del modelo: {accuracy:.4f}")
47
```

Luego se crea la matriz de confusión, que es la que nos muestra el número de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos.

```
48
49    conf_matrix = confusion_matrix(y_test, y_pred)
50    print("\nMatriz de Confusión:")
51    print(conf_matrix)
52
```

Y luego se muestra el informe de clasificación que nos proporciona métricas como la precisión, recall y F1-score para cada clase.

```
53
54    class_report = classification_report(y_test, y_pred)
55    print("\nInforme de Clasificación:")
56    print(class_report)
57
```

El resultado

```
danie@LAPTOP-0MT3MCT1 MINGW64 ~/Downloads/conjunto iris
$ python app.py
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm) species
0                5.1               3.5                1.4               0.2  setosa
1                4.9               3.0                1.4               0.2  setosa
2                4.7               3.2                1.3               0.2  setosa
3                4.6               3.1                1.5               0.2  setosa
4                5.0               3.6                1.4               0.2  setosa
Tamaño del conjunto de entrenamiento: (120, 4)
Tamaño del conjunto de prueba: (30, 4)
Predicciones: ['versicolor' 'setosa' 'virginica' 'versicolor' 'versicolor']
Precisión del modelo: 1.0000

Matriz de Confusión:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

```
Informe de Clasificación:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

3.4 Visualizar resultados

Podemos hacer una visualización de los resultados, por ejemplo, mostrando las predicciones frente a las clases reales usando un gráfico de dispersión.

Primero codificamos las etiquetas de las especies en números

```
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y_test)
```

Luego establecemos un mapa de colores basado en las especies

```
colors = ['red', 'green', 'blue']
species_labels = label_encoder.classes_
```

Visualizamos las predicciones vs las clases reales

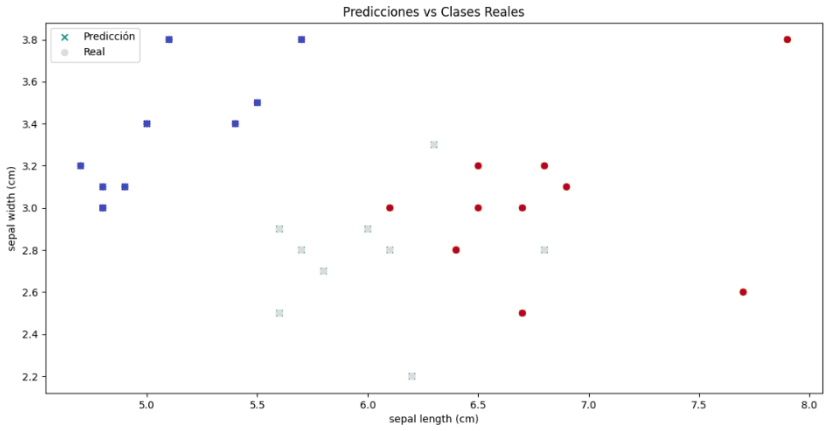
```
plt.figure(figsize=(8, 6))
```

Por último, usamos el valor codificado para asignar colores a las predicciones y las clases reales

```
plt.scatter(X_test.iloc[:, 0], X_test.iloc[:, 1], c=y_encoded, cmap='viridis', marker='x', label='Predicción')
plt.scatter(X_test.iloc[:, 0], X_test.iloc[:, 1], c=y_test.map({'setosa': 0, 'versicolor': 1, 'virginica': 2}),
            cmap='coolwarm', marker='o', label='Real')

plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.title("Predicciones vs Clases Reales")
plt.legend()
plt.show()
```

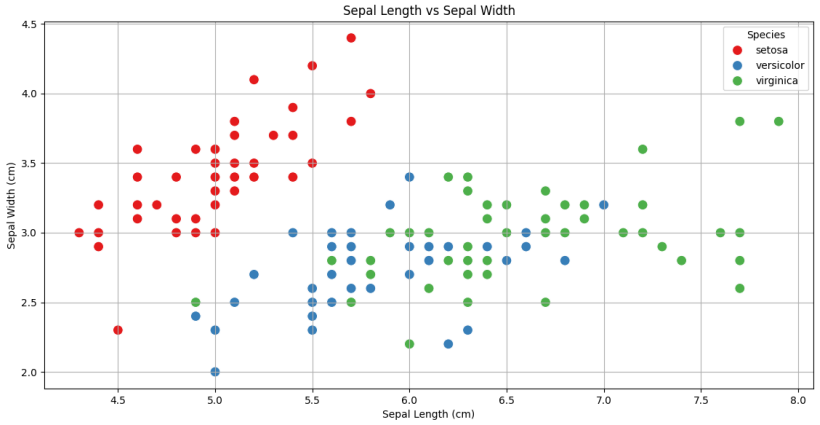
Nos genera un gráfico de dispersión que muestra las predicciones vs las clases reales



Generamos otro grafico de dispersión que visualiza la longitud del sépalo vs. ancho del sépalo.

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='sepal length (cm)', y='sepal width (cm)', hue='species', data=df, palette="Set1", s=100)
plt.title('Sepal Length vs Sepal Width')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.legend(title='Species')
plt.grid(True)
plt.show()
```

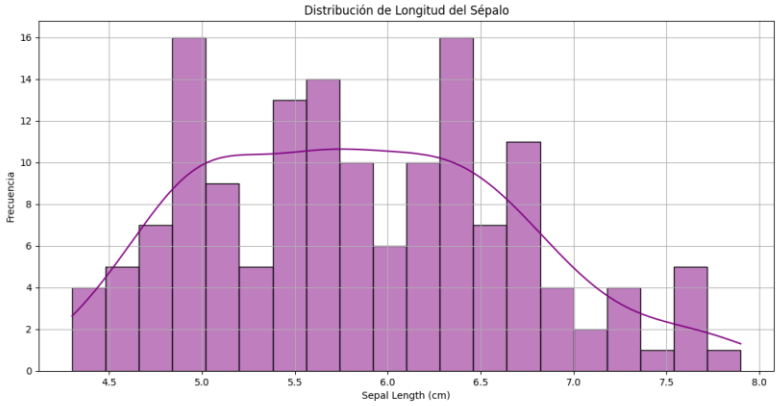
El resultado



Creamos un histograma de longitud del sépalo

```
plt.figure(figsize=(10, 6))
sns.histplot(df['sepal length (cm)'], kde=True, bins=20, color='purple')
plt.title('Distribución de Longitud del Sépalo')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Frecuencia')
plt.grid(True)
plt.show()
```

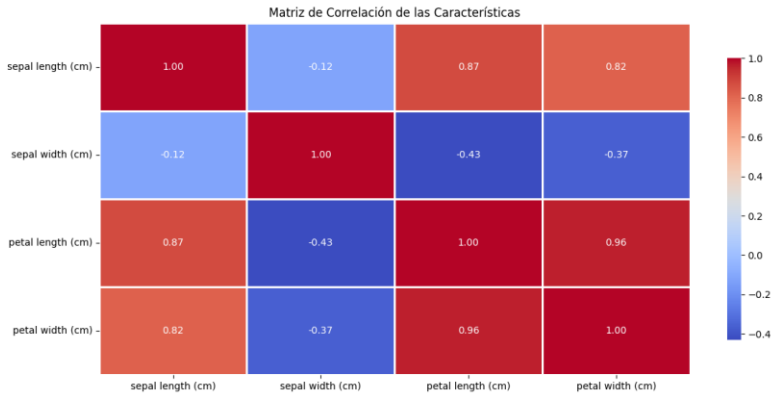
El resultado



Luego una matriz de correlación

```
118
119 correlation_matrix = df.iloc[:, :-1].corr()
120 plt.figure(figsize=(8, 6))
121 sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=1, cbar_kws={'shrink': 0.8})
122 plt.title('Matriz de Correlación de las Características')
123 plt.show()
124
```

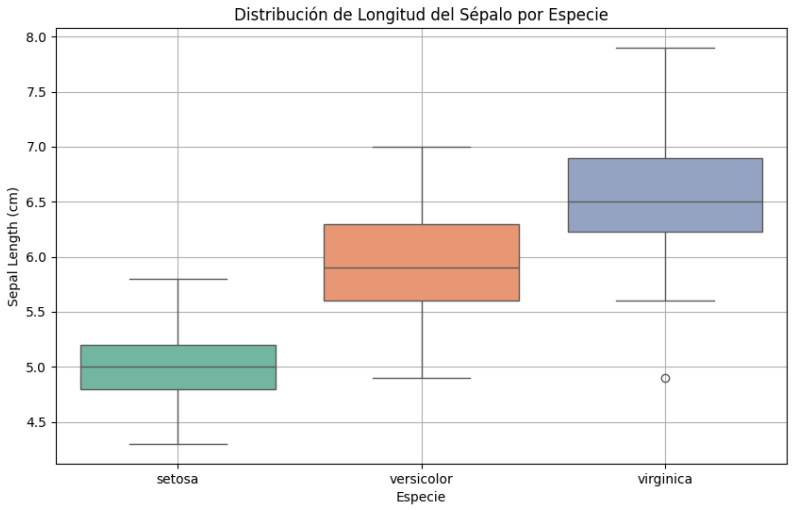
El resultado



Por último, un gráfico de caja (Box plot) para cada especie de longitud del sépalo

```
125
126 plt.figure(figsize=(10, 6))
127 sns.boxplot(x='species', y='sepal length (cm)', data=df, palette='Set2')
128 plt.title('Distribución de Longitud del Sépalo por Especie')
129 plt.xlabel('Especie')
130 plt.ylabel('Sepal Length (cm)')
131 plt.grid(True)
132 plt.show()
```

El resultado



4. Conclusión

En este proyecto se demuestra cómo construir y evaluar un modelo de predicción usando el conjunto de datos Iris y la regresión logística. De igual modo, se ha explorado el conjunto de datos, entrenado el modelo y evaluado su rendimiento con métricas apropiadas.

Es importante destacar que la base de datos iris se ha convertido en un recurso estándar y su creación ha sido de gran importancia para el desarrollo y la evaluación de algoritmos y técnicas de clasificación en el campo de la inteligencia artificial y la estadística.

Referencias

El dataset Iris | Interactive Chaos. (n.d.). https://interactivechaos.com/es/manual/tutorial-de-machine-learning/el-dataset-iris

Portilla, J. S. (2023, April 5). Clasificando el Dataset Iris con Regresión Logística. Joe Portilla. https://joeportilla.co/portfolio/iris-classification/

Python 3.13 documentation. (n.d.). Python Documentation. https://docs.python.org/es/3/

IRIS dataset - Classification. (2025, January 9). Kaggle. https://www.kaggle.com/datasets/taweilo/iris-dataset-elarged-with-smote

RPubs - Flores Iris. (n.d.). https://rpubs.com/Sebastian_Clavijo00/1051199