Práctica: Evaluación de la Calidad de un Modelo de Machine Learning

Objetivo:

- Entrenar un modelo de regresión logística en el conjunto de datos Iris y evaluar su precisión.

Herramientas Utilizadas:

- Python
- Scikit-Learn

Pasos:

Importar Librerías:

- Asegúrate de tener instalada la biblioteca Scikit-Learn (`scikit-learn`). Puedes instalarla usando `pip install scikit-learn`.

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

Cargar y Explorar los Datos:

- Utiliza el conjunto de datos Iris, que está disponible en Scikit-Learn.

```
iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Entrenar un Modelo:

- Entrena un modelo de regresión logística.

```
model = LogisticRegression(random_state=42)

model.fit(X_train, y_train)
```

Realizar Predicciones:

- Realiza predicciones en el conjunto de prueba.

y_pred = model.predict(X_test)

Evaluar la Precisión:

- Calcula la precisión del modelo.

accuracy = accuracy_score(y_test, y_pred)

print(f'Precisión del modelo: {accuracy:.2f}')

- Explora otras métricas como la matriz de confusión y el informe de clasificación para obtener más detalles sobre el rendimiento del modelo.

 # Matriz de confusión

conf_matrix = confusion_matrix(y_test, y_pred)

print('Matriz de Confusión:')

print(conf_matrix)

 # Informe de clasificación

class_report = classification_report(y_test, y_pred, target_names=iris.target_names)

print('Informe de Clasificación:')

print(class_report)

Explica en tus palabras la evaluación del presente modelo de acuerdo a la precisión

Explica en tus palabras la evaluación del modelo de acuerdo a la matriz de confusión