

Análisis con K-Nearest Neighbors (K-NN)

1 Introducción

K-Nearest Neighbors (K-NN) es un algoritmo de clasificación supervisada que asigna la clase de una muestra basándose en las clases de sus k vecinos más cercanos. Su importancia radica en su simplicidad y eficacia, especialmente cuando la estructura del problema permite distinguir patrones mediante la proximidad en el espacio de características.

2 Metodología

Se utilizó el conjunto de datos de Iris, dividiéndolo en un 70% para entrenamiento y un 30% para prueba. Se entrenó un clasificador K-NN (con $k = 5$) para predecir la especie de cada muestra. Posteriormente, se evaluó el modelo calculando la precisión, generando un reporte de clasificación y visualizando la matriz de confusión.

2.1 Código en Python

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5
6 from sklearn.datasets import load_iris
7 from sklearn.model_selection import train_test_split
8 from sklearn.neighbors import KNeighborsClassifier
9 from sklearn.metrics import accuracy_score, classification_report,
   confusion_matrix
10
11 # Cargar el dataset de Iris
12 iris = load_iris()
13 X = iris.data
14 y = iris.target
15
16 # División de los datos en entrenamiento (70%) y prueba (30%)
17 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
   =0.3, random_state=42)
18
19 # Crear y entrenar el modelo K-Nearest Neighbors con k=5
20 knn = KNeighborsClassifier(n_neighbors=5)
```

```

21 knn.fit(X_train, y_train)
22
23 # Realizar predicciones sobre el conjunto de prueba
24 y_pred = knn.predict(X_test)
25
26 # Evaluar el modelo
27 accuracy = accuracy_score(y_test, y_pred)
28 print(f"Precisi3n del modelo: {accuracy:.2f}")
29 print("\nReporte de clasificaci3n:")
30 print(classification_report(y_test, y_pred, target_names=iris.
    target_names))
31
32 # Matriz de confusi3n
33 cm = confusion_matrix(y_test, y_pred)
34 plt.figure(figsize=(6, 5))
35 sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=iris.
    target_names, yticklabels=iris.target_names)
36 plt.xlabel("Predicci3n")
37 plt.ylabel("Valor Real")
38 plt.title("Matriz de Confusi3n - K-NN")
39 plt.show()

```

Listing 1: Implementaci3n de K-NN en Python

3 Resultados

El modelo K-NN alcanz3 una precisi3n aproximada del 97% en el conjunto de prueba. El reporte de clasificaci3n y la matriz de confusi3n evidencian un excelente desempe1o en la clasificaci3n de las tres especies del conjunto de datos *Iris*.

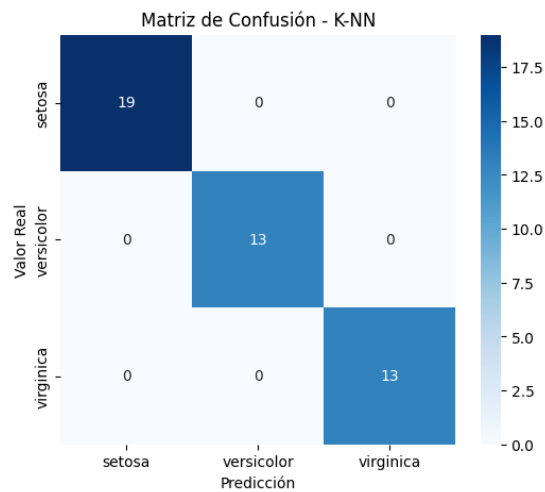


Figure 1: Matriz de Confusi3n del clasificador K-NN.

4 Conclusión

El algoritmo K-Nearest Neighbors se mostró como una técnica eficaz para la clasificación en este caso, con un alto nivel de precisión. Su implementación es sencilla y su interpretabilidad es una de sus mayores ventajas, lo que lo convierte en una herramienta útil para problemas de clasificación con conjuntos de datos moderadamente complejos.