Implementación del Algoritmo K-Nearest Neighbors (KNN) para dataset Wine

1. Algoritmo K-Nearest Neighbors (KNN)

El algoritmo K-Nearest Neighbors (KNN) es un algoritmo de aprendizaje supervisado utilizado para tareas de clasificación y regresión. Su funcionamiento se basa en que instancias similares suelen estar cerca en el espacio de características. Para predecir la clase de una nueva instancia, se calcula la distancia entre ésta y las instancias de entrenamiento y se seleccionan las k instancias más cercanas. La clase más común entre las k instancias se asigna a la nueva instancia. En este caso, se utilizó la distancia euclidiana para calcular la distancia entre instancias.

2. El Conjunto de Datos de Wine (Wine Dataset)

El dataset de Wine es un conjunto de datos clásico utilizado para probar algoritmos de clasificación. Contiene 178 instancias de vinos, cada una descrita por 13 características químicas diferentes, y cada vino pertenece a una de tres clases. Este dataset es suficientemente complejo para demostrar la utilidad de KNN en un entorno realista, además de ser un set de datos muy popular.

3. División de los Datos en Entrenamiento y Prueba (Train-Test Split)

Se utilizó una división de los datos en un conjunto de entrenamiento y un conjunto de prueba debido a que es un paso crucial para evaluar el rendimiento de un modelo de manera objetiva. En este caso, se usó una división de 75% para entrenamiento y 25% para prueba para que el modelo tenga suficientes datos para aprender y que el conjunto de prueba sea lo suficientemente grande para validar el rendimiento.

4. Semilla Aleatoria

Se utilizó una semilla aleatoria para garantizar la reproducibilidad de los resultados. Al fijar esta semilla, podemos obtener la misma división de datos cada vez que ejecutemos el código, lo que facilita la comparación de resultados

5. Estandarización

La estandarización es un paso muy importante para preparar los datos, especialmente para algoritmos basados en distancias como lo es KNN. Sin estandarización, las características con escalas muy grandes pueden dominar la distancia euclidiana y

afectar los resultados del modelo.. Al estandarizar las características, se asegura que todas tengan la misma escala y se evitan errores

6. Predicción con KNN (k=5)

El parámetro k representa el número de vecinos que se consideran para hacer una predicción. En este caso, k=5 se eligió para balancear el riesgo de overfitting (valores bajos de k) y underfitting (valores altos de k).

7. Matriz de Confusión

La matriz de confusión proporciona una comparación entre las etiquetas verdaderas y las predicciones del modelo.

En el caso de este modelo, la matriz de confusión es:

```
[[15  0  0]
 [ 1 16  1]
 [ 0  0 12]]
```

Esta matriz indica que:

- 15 vinos de la clase 0 fueron clasificados correctamente.
- 1 vino de la clase 1 fue incorrectamente clasificado como de clase 0.
- 16 vinos de la clase 1 fueron clasificados correctamente.
- 1 vino de la clase 1 fue incorrectamente clasificado como de clase 2.
- 12 vinos de la clase 2 fueron clasificados correctamente.

8. Precisión, Recall y F1 Score

- **Precisión (0.95):** Indica el porcentaje de predicciones correctas entre todas las predicciones realizadas para cada clase.
- **Recall (0.96):** Mide la capacidad del modelo para identificar correctamente todas las instancias de una clase particular.
- **F1 Score (0.96):** Es la media armónica de la precisión y el recall, por lo que equilibra ambas métricas.

En este caso, los resultados altos en precisión, recall y F1 Score indican que el modelo tiene un desempeño muy bueno en la clasificación de los vinos en las diferentes clases.

9. Conclusiones

El modelo KNN implementado muestra un alto rendimiento en la clasificación del dataset de vinos. El uso de estandarización y un valor adecuado de k ha permitido que el modelo haga predicciones precisas y balanceadas.

10. Recomendaciones a Futuro y Mejoras

Implementar un Grid Search para buscar los hiperparámetros (k) más óptimos para esta situación puede mejorar el rendimiento del modelo, tomando en cuenta que es una tarea muy intensiva y de alto requerimiento de poder computacional. Además, se puede considerar la validación cruzada para obtener una estimación más robusta del rendimiento del modelo.