

Act 14: Programando K-Nearest-Neighbor en Python

Patricio Ricardí

March 2025

¿Qué es K-NN?

Es un método de clasificación que predice la categoría de un dato basándose en:

- Los K datos más cercanos (vecinos)
- Una votación entre las clases de estos vecinos
- Ejemplo: Clasificar una fruta como "manzana" si sus 5 vecinos más cercanos son manzanas

Pasos realizados

1. **Cargar datos**: Usamos un archivo CSV con información para clasificar
2. **Preparar datos**:
 - Dividir en entrenamiento (80%) y prueba (20%)
 - Escalar características (normalizar valores)
3. **Entrenar modelo**:
 - Elegir $K = 5$ vecinos
 - Usar distancia euclidiana para medir cercanía
4. **Evaluar resultados**:
 - Matriz de confusión
 - Precisión, Recall y F1-score
 - Comparación con otros valores de K

Código simplificado

```
[language=Python, basicstyle=] Dividir datos from sklearn.model_selection import train_test_split
X_entrenamiento, X_prueba = train_test_split(datos, clases, test_size = 0.2)

Escalar características from sklearn.preprocessing import StandardScaler
escalador = StandardScaler() X_entrenamiento = escalador.fit_transform(X_entrenamiento)
X_prueba = escalador.transform(X_prueba)

Crear y entrenar modelo K-NN from sklearn.neighbors import KNeighborsClassifier
modelo_knn = KNeighborsClassifier(n_neighbors = 5) modelo_knn.fit(X_entrenamiento, y_entrenamiento)

Evaluar from sklearn.metrics import accuracy_score, classification_report
predicciones = modelo_knn.predict(X_prueba) print(f" Precisión : accuracy_score(y_prueba, predicciones) : .2f")
```

Resultados clave

- **Precisión:** 82% en datos de prueba
- **Matriz de confusión:**
 - 48 aciertos en clase 0 (de 60)
 - 34 aciertos en clase 1 (de 40)
- **Mejor K :** 5 vecinos mostró el mejor equilibrio
- F1-score promedio: 81%

Conclusiones

- Muy efectivo para problemas con fronteras de decisión complejas
- Ventajas:
 - Fácil de entender e implementar
 - Sin suposiciones sobre distribución de datos
- Desventajas:
 - Lento con datasets muy grandes
 - Sensible a datos no escalados
- Para mejorar:
 - Probar diferentes valores de K
 - Usar técnicas de reducción de dimensionalidad