
RESULTADO DE APRENDIZAJE

RdA de la asignatura:

- **RdA 2:** Aplicar modelos de aprendizaje automático supervisado y no supervisado, así como su validación y optimización, en la resolución de problemas tanto reales como simulados.

RdA de la clase:

- Comprender los fundamentos del algoritmo kNN, sus ventajas, desventajas y parámetros principales.
- Implementar el algoritmo kNN en Python y evaluar su desempeño en un problema práctico.

INTRODUCCIÓN

Pregunta inicial: Imagina que estás organizando una fiesta y no sabes si alguien prefiere pizza o sushi. Si conoces las preferencias de sus amigos más cercanos, ¿podrías predecir su elección? ¿Cómo lo harías?

DESARROLLO

Actividad 1: Introducción al algoritmo kNN

En esta actividad, los estudiantes conocerán el algoritmo kNN, sus fundamentos, ventajas, desventajas y parámetros clave. Se iniciará con la visualización de un video y se continuará con una discusión guiada sobre los conceptos más importantes.

¿Cómo lo haremos?

- **Proyección de video:** Se presentará un video educativo que explique qué es el aprendizaje no supervisado.

Enlace al video: [K Nearest Neighbors](#).

- **Interacción con ChatGPT:** Después del video, cada estudiante formulará preguntas a ChatGPT relacionadas con los términos o conceptos nuevos.
- **Clase magistral:** Se explicará cómo funciona kNN en detalle:
 - Concepto de "vecinos más cercanos".
 - Parámetros importantes (valor de k, peso por distancia).
 - Ventajas y desventajas del algoritmo.

Verificación de aprendizaje: Se plantearán preguntas rápidas para los estudiantes:

- ¿Cómo decide kNN la clase de un punto nuevo cuando $k = 3$?
- Si $k = 1$, ¿cómo afecta esto al modelo en términos de sensibilidad al ruido?

Actividad 2: Implementación práctica del algoritmo kNN

En esta actividad, los estudiantes implementarán el algoritmo kNN utilizando Python. Se trabajará con un cuaderno de Jupyter que incluye un ejemplo guiado y un ejercicio práctico.

¿Cómo lo haremos?

- **Preparación:** Los estudiantes accederán a un cuaderno de Google Colab preparado para esta actividad. El cuaderno contiene un ejemplo básico de clasificación utilizando kNN.

Enlace al cuaderno: [10-kNN.ipynb](#).

- **Ejercicio guiado:** Los estudiantes ejecutarán y modificarán el siguiente código base:

```
# Importar librerías necesarias
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Cargar el dataset Iris
iris = load_iris()
X, y = iris.data, iris.target

# Dividir los datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Crear y entrenar el modelo kNN con k=3
modelo = KNeighborsClassifier(n_neighbors=3)
modelo.fit(X_train, y_train)

# Realizar predicciones y evaluar el modelo
y_pred = modelo.predict(X_test)
print("Precisión del modelo:", accuracy_score(y_test, y_pred))
```

- **Experimentación:** Los estudiantes modificarán el código para experimentar con:
 - Cambiar el valor de k y observar cómo afecta la precisión.
 - Probar diferentes métricas de distancia, como 'manhattan' o 'euclidean'.
 - Usar una proporción diferente en el conjunto de prueba (test_size).

EJERCICIO 1. Modifique el código para:

1. Establecer $k = 5$ y comparar los resultados con $k = 3$.

2. Utilizar la métrica de distancia 'manhattan' en lugar de 'euclidean'.
3. Cambiar el tamaño del conjunto de prueba a 40% y analizar el impacto en la precisión.

Verificación de aprendizaje:

- ¿Cómo afecta el valor de k la precisión del modelo?
- ¿Qué diferencias observaste al usar la métrica 'manhattan' en comparación con 'euclidean'?
- ¿Cómo influye el tamaño del conjunto de prueba en los resultados del modelo?

CIERRE

Tarea: Desarrollar los ejercicios planteados en el siguiente cuaderno y entregarlo por el aula virtual:

Enlace al cuaderno: [05-kNN.ipynb](#).

Pregunta de investigación:

1. ¿Qué sucede con el rendimiento de kNN cuando el tamaño del dataset crece significativamente, y cómo podrías abordar este problema?
2. ¿Qué tan sensible es kNN a datos con ruido o valores atípicos, y cómo podrías pre-procesar los datos para mitigar estos efectos?

Para la próxima clase: Visualizar el siguiente video sobre SVM:

Enlace al video: [¿Qué es una SVM?](#).