**Tarea - Clustering & consulta ML**

**Presentado por:**

Hinara Pastora Sánchez Mata - *hisanchezm@unal.edu.co*

**Profesor:**

[Demetrio A Ovalle Carranza](mailto:dovalle@unal.edu.co)

Domingo 28 de Julio

****

**Universidad Nacional de Colombia**

**Facultad de Minas**

**Departamento de Ciencias de la computación y de la decisión**

**Ingeniería de sistemas e informática**

**2024**

1. En Machine Learning existen problemas característicos como clasificación, regresión, clustering y reducción de dimensionalidad. Para cada uno de estos problemas existen modelos con estructuras algorítmicas diferentes. Algunos ejemplos de estas estructuras para los problemas mencionados son:

* KNN
* Regresión lineal
* K-Means
* PCA

Para cada tipo de problema consulte otro algoritmo que sea de uso común. Explique brevemente el funcionamiento de los cuatro algoritmos dados en el enunciado y los cuatro investigados.  
  
**Clasificación**

* **KNN (K-Nearest Neighbors)**: Es un algoritmo de clasificación que asigna una etiqueta a un punto nuevo basado en las etiquetas de sus "k" vecinos más cercanos. Se basa en la idea de que puntos similares (vecinos cercanos) están en la misma categoría. Es un método sencillo pero efectivo, especialmente cuando el conjunto de datos no es muy grande.
* **Random Forest**: Es un conjunto de árboles de decisión. Cada árbol en el bosque vota por una clase y la clase con más votos se elige como la predicción final. Este algoritmo es robusto y maneja bien grandes conjuntos de datos con muchas características.

**Regresión**

* **Regresión lineal**: Este algoritmo intenta encontrar la línea recta que mejor se ajusta a un conjunto de datos. La idea es predecir el valor de una variable dependiente (y) basada en el valor de una variable independiente (x) usando la fórmula y = mx + b. Es simple y funciona bien para relaciones lineales.
* **Regresión polinómica**: Es una extensión de la regresión lineal, donde la relación entre la variable independiente y la dependiente se modela como un polinomio. Permite capturar relaciones más complejas que no son lineales.

**Clustering**

* **K-Means**: Este algoritmo agrupa datos en "k" clusters, donde cada punto pertenece al cluster con el centroide más cercano. El objetivo es minimizar la distancia dentro de cada cluster y maximizar la distancia entre los clusters. Es eficiente para conjuntos de datos grandes y es fácil de entender.
* **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: Se encarga de encontrar clusters basados en la densidad de puntos. Agrupa puntos cercanos que están en regiones densas y marca puntos en regiones dispersas como ruido. No requiere especificar el número de clusters de antemano y maneja bien datos con formas arbitrarias.

**Reducción de dimensionalidad**

* **PCA (Principal Component Analysis)**: Algoritmo que reduce la dimensionalidad de un conjunto de datos transformando las variables originales en un nuevo conjunto de variables no correlacionadas llamadas componentes principales. Estos componentes retienen la mayor cantidad de variabilidad posible de los datos originales. Es útil para simplificar y visualizar datos complejos.
* **t-SNE (t-Distributed Stochastic Neighbor Embedding)**: Es una técnica de reducción de dimensionalidad que se utiliza principalmente para la visualización de datos de alta dimensión. Convierte las similitudes entre puntos de datos en probabilidades y minimiza la divergencia entre las distribuciones de puntos en alta y baja dimensión. Es excelente para visualizar datos en 2D o 3D y descubrir patrones ocultos.

2. ¿Qué tipo de preprocesamiento debería hacer a un conjunto de datos si quisiera usar estos algoritmos y obtener un buen desempeño? (¿Qué puede ser común para todos y que es específico de cada uno?).

**Preprocesamiento común para todos los algoritmos**:

1. **Limpieza de datos**:

- Manejo de valores faltantes: Imputar valores faltantes o eliminar registros/columnas con valores faltantes.

- Eliminación de duplicados: Asegurarse de que no haya registros duplicados en el conjunto de datos.

- Corrección de errores: Identificar y corregir errores en los datos.

2. **Escalado de datos**:

- Normalización o estandarización: Escalar las características para que tengan una media de 0 y una desviación estándar de 1 (estandarización) o ajustar los valores a un rango de 0 a 1 (normalización).

3. **Codificación de variables categóricas**:

- One-Hot Encoding: Convertir variables categóricas en una serie de variables binarias.

- Label Encoding: Asignar un número entero a cada categoría.

4. **Eliminación de Outliers**:

- Identificar y tratar con valores atípicos que pueden afectar el desempeño de los algoritmos.

5. **División del conjunto de datos**:

- Train/Test Split: Separar los datos en conjuntos de entrenamiento y prueba para evaluar el rendimiento del modelo de manera justa.

**Preprocesamiento específico para cada algoritmo**:

1. **KNN (K-Nearest Neighbors)**:

- Escalado de características: Muy importante porque KNN es sensible a las distancias.

- Reducción de dimensionalidad: Utilizar PCA para reducir el número de características y acelerar el proceso.

2. **Regresión Lineal**:

- Escalado de características: Aunque no siempre es necesario, puede ayudar con la convergencia de algoritmos de optimización.

- Transformación de características: Crear nuevas características basadas en relaciones no lineales o polinómicas.

- Eliminación de multicolinealidad: Detectar y tratar con características altamente correlacionadas.

3. **K-Means**:

- Escalado de características: Necesario para asegurar que todas las características contribuyan equitativamente a la distancia.

- Determinación del número de clusters: Utilizar el método del codo o la silueta para determinar el número óptimo de clusters.

4. **PCA (Principal Component Analysis)**:

- Escalado de características: Es crucial porque PCA busca maximizar la varianza y características con diferentes escalas pueden dominar.

- Centralización de datos: Restar la media de cada característica para centrar los datos en el origen.

5. **Random Forest**:

- Manejo de valores faltantes: Random Forest puede manejar valores faltantes, pero es mejor imputarlos.

- Codificación de variables categóricas: One-Hot Encoding para variables categóricas.

- Reducción de dimensionalidad: No es estrictamente necesario, pero puede acelerar el proceso y mejorar la interpretabilidad.

6. **Regresión Polinómica**:

- Creación de características polinómicas: Generar características basadas en polinomios de las variables originales.

- Escalado de características: Ayuda en la convergencia de los algoritmos de optimización.

7. **DBSCAN**:

- Escalado de características: No es tan crítico como en K-Means, pero puede ayudar.

- Eliminación de ruido y outliers: Pre-procesar para eliminar datos claramente erróneos o ruidosos que podrían afectar el algoritmo.

8. **t-SNE**:

- Escalado de características: Aunque t-SNE no es tan sensible al escalado como otros algoritmos, puede ayudar a mejorar la convergencia y el rendimiento.

- Reducción preliminar de dimensionalidad: Utilizar PCA antes de aplicar t-SNE para reducir la dimensionalidad inicial a algo manejable (e.g., 50 dimensiones).

3. Investigue cómo implementar la técnica de clustering aglomerativo y úsela para encontrar una segmentación del dataset Iris. Explique su razonamiento para la elección de la altura del dendrograma y compare los resultados con los obtenidos usando k-means (¿se obtuvieron resultados similares?, si hay diferencias ¿a qué podrían deberse?).

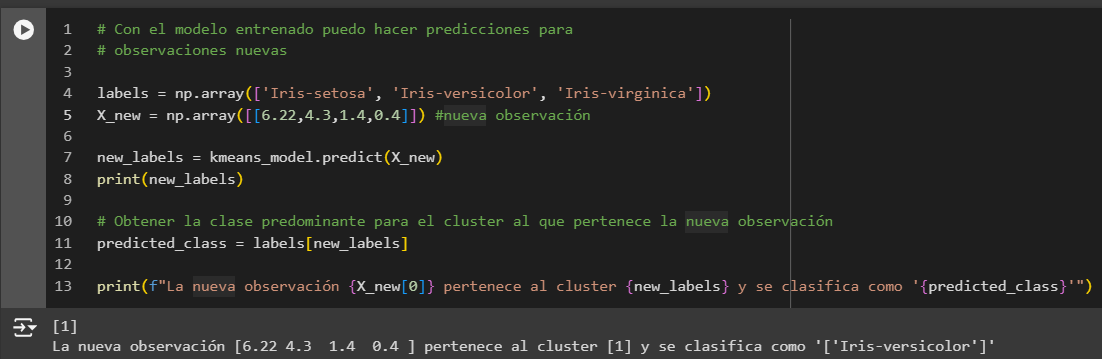
**Implementación de clustering aglomerativo en el Dataset Iris**

El clustering aglomerativo es una técnica jerárquica que agrupa los datos en una jerarquía de clusters. Vamos a utilizar el dataset Iris, que contiene 150 muestras de flores de iris con cuatro características: longitud y ancho del sépalo, y longitud y ancho del pétalo. Este dataset también incluye la especie de cada muestra, que usaremos para comparar la efectividad del clustering.

**Véase el código y la implementación en el notebook**

4. Realice una predicción para una flor con las siguientes características:

sepal-length = 6.22, sepal-width = 4.3, petal-length = 1.4, petal-width = 0.4. ¿A que especie pertenece?.



Como podemos ver en la imagen esta nueva flor pertenece al cluster 1 que hace referencia a la especie Iris Versicolor