Informe Ejecutivo – Proyecto VIII: Clustering y Reducción de Dimensionalidad con el Dataset de Setas

Fecha: 30/05/2025
Autores: Equipo de trabajo - Taller de IA y Machine Learning: Andreina Suescum y César Mercado.

Objetivo del Proyecto

Este proyecto tiene como objetivo aplicar técnicas de aprendizaje no supervisado (clustering) y reducción de dimensionalidad (PCA) sobre el conocido dataset de setas, con el fin de:

- Explorar la estructura interna de los datos sin usar etiquetas.
- Identificar agrupaciones naturales mediante KMeans.
- Comparar los resultados del clustering con un modelo de clasificación supervisado (Random Forest).
- Evaluar la calidad de los clusters utilizando métricas especializadas.

Dataset utilizado

- **Fuente:** Mushroom Dataset - Kaggle
- **Observaciones:** 8124 muestras de hongos
- **Características:** 22 variables categóricas (forma, color, olor, tallo, etc.)
- **Variable objetivo:** `class` (binaria: comestible `e` o venenoso `p`)

Tipo de problema

La variable objetivo `class` es de tipo categórico binario. Por lo tanto:

Se trata de un **problema de clasificación**, no de regresión.

Flujo de trabajo realizado

1. **Carga y exploración inicial**
    - Lectura del dataset completo desde CSV local.
    - Análisis de valores nulos y valores únicos por columna.

    - Validación de limpieza y consistencia.

2. **Preprocesamiento**
    - Uso de `LabelEncoder` para convertir todas las variables categóricas a formato numérico.
    - Separación entre variables predictoras (`X`) y variable objetivo (`y`).

3. **Reducción de dimensionalidad con PCA**
    - Aplicación de PCA para representar los datos en 2 dimensiones.
    - Visualización de los puntos en el plano PCA según la clase real (`class`).

4. **Clustering no supervisado con KMeans**
    - Evaluación del número óptimo de clusters con el método del codo.
    - Entrenamiento de KMeans con `k=9`.
    - Asignación de cada muestra a un cluster.
    - Visualización de la distribución de clases reales dentro de cada cluster con `catplot`.
    - Visualización en scatterplot PCA coloreando por cluster.

5. **Evaluación del clustering**
    - Se emplearon métricas objetivas:
        - Coeficiente de silueta (`silhouette_score`)
        - Índice de Rand Ajustado (`adjusted_rand_score`)
        - Información mutua normalizada (`normalized_mutual_info_score`)

6. **Comparación con modelo supervisado**
    - Entrenamiento de un clasificador Random Forest sobre los mismos datos.
    - Evaluación del modelo con `accuracy_score`, `classification_report` y `confusion_matrix`.
    - Comparación cualitativa y cuantitativa entre clustering y clasificación.

Resultados destacados

- PCA ha permitido reducir las +90 columnas codificadas a solo 2 dimensiones para análisis visual.
- KMeans logró separar de forma bastante clara las clases comestible y venenoso sin conocerlas.
- El modelo supervisado (Random Forest) alcanzó una precisión cercana al 100%.
- Las métricas de clustering muestran una **buena correspondencia entre clusters y etiquetas reales**, aunque no perfecta.

Conclusiones

- El dataset de hongos presenta una estructura clara que puede ser identificada sin necesidad de etiquetas.
- PCA ha facilitado la comprensión visual de los datos, revelando separaciones significativas.
- KMeans ha sido capaz de agrupar los hongos de forma bastante similar a su clasificación real, lo que lo convierte en una herramienta útil para tareas exploratorias.
- Random Forest ha confirmado el buen alineamiento entre clusters y clases reales, mostrando que muchas variables contienen información relevante.
- En problemas donde no se disponga de etiquetas, este enfoque sería muy útil para descubrir patrones ocultos y generar hipótesis.

Competencias trabajadas según la rúbrica

- ✔ Análisis exploratorio de datos categóricos
- ✔ Transformación de variables categóricas
- ✔ Aplicación de PCA y visualización en 2D
- ✔ Implementación de KMeans y evaluación con métricas avanzadas
- ✔ Comparación de resultados con modelos supervisados
- ✔ Identificación del tipo de problema (clasificación)
- ✔ Visualización clara con `matplotlib` y `seaborn`
- ✔ Control de versiones y estructura de trabajo colaborativo

---