

## Ejercicio 2 – Explicación del Notebook de Visualización 2D/3D

===== Este documento explica cada sección del notebook `ejercicio2.ipynb`, utilizado para la visualización y análisis de las trazas generadas en Unity para la Práctica 5 (APA - Battle City).

### ----- 1. Cargar Librerías

----- En la primera celda se importan las librerías necesarias: - pandas y numpy → Manejo de datos. - glob → Búsqueda de archivos CSV automáticamente. - matplotlib → Dibujar gráficos. - scikit-learn (PCA, StandardScaler) → Reducción de dimensionalidad. Estas librerías permiten trabajar de forma eficiente con datos numéricos y visualizar resultados. ----- 2. Cargar los CSV

generados en Unity ----- Se usa la función: `glob.glob("data/TankTraining*.csv")` para localizar automáticamente todos los archivos CSV generados en modo "Recorded". Cada archivo CSV contiene: - Variables de percepción del tanque. - Tiempo de captura del snapshot. - Acción ejecutada por el jugador. Después se concatenan todos los CSV en un único DataFrame usando: `data = pd.concat(dfs, ignore_index=True)` Esto permite trabajar con un dataset unificado y más grande.

### ----- 3. Separación de Features (X) y Etiquetas (y)

----- Se separan los datos en: - `**X**` → Todas las columnas excepto "time" y "action". Representan la percepción del entorno por parte del tanque. - `**y**` → La columna "action". Representa la clase/etiqueta a predecir (movimientos del jugador). Esta separación es esencial para aplicar técnicas de machine learning.

### ----- 4. Visualización de la Distribución de Clases

----- Se usa: `y.value_counts().plot(kind="bar")` Este gráfico de barras muestra cuántos ejemplos existen de cada acción: - Permite detectar desequilibrio entre clases. - Ayuda a identificar si el jugador repite demasiado ciertas acciones. - Evalúa la calidad del dataset previo al entrenamiento.

### ----- 5. Reducción de Dimensionalidad con PCA (2D)

----- El dataset tiene muchos atributos y no puede visualizarse directamente. Por eso se usa `**PCA**`, explicado en el Tema 6 de Data Mining. Pasos: 1. Escalar los datos con `StandardScaler` (PCA requiere datos normalizados). 2. Aplicar PCA a 2 componentes: `PCA(n_components=2)` 3. Dibujar los puntos en 2D coloreados según su clase. El resultado es un mapa visual donde cada punto representa un estado del juego y su color indica la acción tomada (UP, DOWN, LEFT, RIGHT...). ----- 6.

PCA en 3D ----- Similar al apartado anterior, pero usando: `PCA(n_components=3)` y graficando en un espacio tridimensional. Esto ofrece una representación más profunda de cómo se organizan los datos.

### ----- Conclusión

----- Este notebook cumple completamente los objetivos del Ejercicio 2: ✓ Combina todas las trazas generadas en Unity ✓ Muestra la distribución de clases ✓ Reduce dimensionalidad utilizando PCA ✓ Visualiza los datos en gráficos 2D y 3D Esto permite analizar si los datos son consistentes y sirven para entrenar un modelo MLP, que será el objetivo del siguiente ejercicio.