**Memoria**

ML: Tienda de Juegos

Las herramientas principales que se ocuparon en el ML

* Python 3.11.5
* Kaggle: Datasets
* Librerías Panda, Matplotlib, Folium, Plotly, Seaborn, numpy, sklearn, DecisionTreeRegressor, KNeighborsClassifier, RandomForestClassifier
* ChatGPT 4.0

Busque el tema el Kaggle en el siguiente enlace:

[https://www.kaggle.com/datasets/andrewmvd/board-games](https://www.kaggle.com/datasets/jahaidulislam/significant-earthquake-dataset-1900-2023).

En el proyecto de Machine Learning, se realizaron los siguientes pasos

1. Limpieza de Datos y Homologación:

La primera fase se centró en la limpieza exhaustiva de los datos, abordando valores nulos y asegurando la homologación de variables para tener un conjunto de datos completo y coherente. Este paso es esencial para establecer una base sólida antes de la exploración y modelado.

1. Análisis Exploratorio de Datos (EDA):

Luego, se realizó un análisis exploratorio de datos para comprender la estructura y la naturaleza de la información. Este análisis proporcionó una visión detallada de las variables presentes, posibles patrones y relaciones, lo que ayudó a tomar decisiones informadas en las etapas posteriores del proyecto.

1. Definición del Objetivo:

Se estableció claramente el objetivo del proyecto: clasificar juegos de mesa en cuatro categorías principales para optimizar el trabajo de la tienda de juegos.

1. Aplicación de Machine Learning:

Con el entendimiento profundo de los datos y el objetivo claro, se procedió a la implementación de modelos de Machine Learning. Se probaron tres modelos diferentes: Decision Tree, Random Forest y KNN, como parte del proceso de evaluación para determinar el más efectivo en función de las métricas de rendimiento.

En conjunto, la combinación de estos métodos y técnicas contribuyó a la implementación eficiente y estructurada del código, permitiendo un análisis detallado de los datos sobre terremotos.

1. Evaluación y Selección del Modelo:

En modelo RandomForestClassifier

* 1. Inicialización del Modelo (rf\_clf):
* Se crea una instancia del RandomForestClassifier.
  1. División de Datos con Estratificación:
* Los datos se dividen en conjuntos de entrenamiento y prueba utilizando la función train\_test\_split.
* La estratificación se utiliza para asegurar una distribución proporcional de las clases en ambos conjuntos.
  1. Aplicación de SMOTE (Oversampling):
* Se utiliza la técnica de SMOTE (Synthetic Minority Over-sampling Technique) para abordar el desbalance de clases en el conjunto de entrenamiento. SMOTE genera muestras sintéticas para las clases minoritarias para equilibrar la distribución.
  1. Conteo de Frecuencias después de Aplicar SMOTE:
* Se imprime el conteo de las frecuencias de las clases después de aplicar SMOTE, para verificar el equilibrio.
  1. GridSearchCV para Buscar Mejores Hiperparámetros:
* Los parámetros incluyen el número de árboles (n\_estimators), la profundidad máxima de los árboles (max\_depth), el número mínimo de muestras requeridas para dividir un nodo interno (min\_samples\_split), y el número mínimo de muestras requeridas para ser una hoja (min\_samples\_leaf).
* Se ajusta el modelo a diferentes combinaciones de hiperparámetros y utiliza validación cruzada (cv=5) "Cross Validación" para evaluar el rendimiento.
  1. Mejor Modelo:
* Se obtiene el mejor modelo identificado por GridSearchCV con best\_estimator\_.
  1. Evaluación del Modelo en el Conjunto de Prueba:
* Se realiza la predicción en el conjunto de prueba utilizando el mejor modelo.
* Se calcula la precisión (accuracy) y se imprime el informe de clasificación (classification\_report).

**Resumen:**

Después de analizar detenidamente los resultados obtenidos con los tres modelos (KNN, RandomForest, DecisionTreeClassifier), llego a la conclusión de que el modelo RandomForest es la opción más adecuada.

RandomForest:

Exactitud (Accuracy): 0.7852

Aspectos Destacados:

Mayor precisión en comparación con KNN (0.7670) y DecisionTreeClassifier (0.7299).

Equilibrio notable entre precisión, recall y F1-score en diversas clases, indicando un rendimiento general sólido.

Alta precisión y recall, especialmente para la clase crítica (Seccion Strategy).

RandomForest, al ser un modelo de conjunto, construye múltiples árboles de decisión y combina sus predicciones, proporcionando robustez y mejor capacidad de generalización.

La mayor exactitud y el rendimiento equilibrado refuerzan la confianza en la capacidad del modelo para realizar predicciones precisas.