PROYECTO BIG DATA - VENTAS DE VIDEOJUEGOS SEGÚN SU CLASIFICACIÓN EN LA ESRB
PROGRAMA DE CIENCIAS DE LOS DATOS

Esteban Castillo Gamboa

Cargado y preprocesamiento

Las bases de datos se encuentran en formato csv por lo que se implementaron dos métodos para cargar cada base en un dataframe de Pyspark con su schema correspondiente.
Para la base de datos de ventas de videojuegos se implementó el siguiente esquema de Spark:

```
Ventas schema
StructType([
    StructField("Rank", IntegerType()),
    StructField("Name", StringType()),
    StructField("basename", StringType()),
    StructField("Genre", StringType()),
    StructField("ESRB_Rating", StringType()),
    StructField("Platform", StringType()),
    StructField("Publisher", StringType()),
    StructField("Developer", StringType()),
    StructField("VGChartz_Score", FloatType()),
    StructField("Critic_Score", FloatType()),
    StructField("User_Score", FloatType()),
    StructField("Total_Shipped", FloatType()),
    StructField("Global_Sales", FloatType()),
    StructField("NA_Sales", FloatType()),
    StructField("PAL_Sales", FloatType()),
    StructField("JP_Sales", FloatType()),
    StructField("Other_Sales", FloatType()),
    StructField("Year", FloatType()),
    StructField("Last_Update", StringType()),
    StructField("url", StringType()),
    StructField("status", IntegerType()),
    StructField("Vgchartzscore", FloatType()),
    StructField("img_url", StringType())
])
```

Para la base de datos ratings de videojuegos de la ESRB se implementó el siguiente esquema de Spark:

```
Ratings schema
StructType([
    StructField("title", StringType()),
    StructField("console", IntegerType()),
    StructField("alcohol_reference", IntegerType()),
    StructField("animated_blood", IntegerType()),
    StructField("blood", IntegerType()),
    StructField("blood_and_gore", IntegerType()),
    StructField("cartoon_violence", IntegerType()),
    StructField("crude_humor", IntegerType()),
    StructField("drug_reference", IntegerType()),
    StructField("fantasy_violence", IntegerType()),
    StructField("intense_violence", IntegerType()),
    StructField("language", IntegerType()),
    StructField("lyrics", IntegerType()),
    StructField("mature_humor", IntegerType()),
    StructField("mild_blood", IntegerType()),
    StructField("mild_cartoon_violence", IntegerType()),
    StructField("mild_fantasy_violence", IntegerType()),
    StructField("mild_language", IntegerType()),
    StructField("mild_lyrics", IntegerType()),
    StructField("mild_suggestive_themes", IntegerType()),
    StructField("mild_violence", IntegerType()),
    StructField("no_descriptors", IntegerType()),
    StructField("nudity", IntegerType()),
    StructField("partial_nudity", IntegerType()),
    StructField("sexual_content", IntegerType()),
    StructField("sexual_themes", IntegerType()),
    StructField("simulated_gambling", IntegerType()),
    StructField("strong_janguage", IntegerType()),
    StructField("strong_sexual_content", IntegerType()),
    StructField("suggestive_themes", IntegerType()),
    StructField("use_of_alcohol", IntegerType()),
    StructField("use_of_drugs_and_alcohol", IntegerType()),
    StructField("violence", IntegerType()),
    StructField("esrb_rating", StringType())
])
```

Para el preprocesamiento de los datos se implementaron las siguientes funciones en el módulo carga_datos.py:

1. ***drop_cols(sales_df)***: Elimina las columnas innecesarias de la base de datos de ventas para el entrenamiento del modelo.
2. ***cast_year_col(sales_df)***: Cambia el tipo de dato de la columna 'Year' de flotante a entero.
3. ***drop_global_sales_nulls(sales_df)***: Elimina las filas con valores nulos o ceros en la columna 'Global_Sales' ya que la predicción se basará en este valor y las entradas vacías no aportan nada al modelo.
4. ***group_sales_by_game(sales_df)***: Agrupa las ventas globales por nombre de juego. Esto, ya que las ventas vienen separadas por plataforma. Por lo que se tienen que unir todas las ventas para un mismo juego que haya salido en varias consolas o plataformas.
5. ***def normalize_col(df, col_name)***: Escala o normaliza los valores de la columna 'Global_Sales' para estandarizar los datos con el fin de evitar *overfitting.*
6. ***add_target_col(df, col_name)***: Crea y agrega la columna *target* al conjunto de datos. La columna *target* se llama Successful y es calculada analizando si las ventas globales de un videojuego son mayores o iguales al percentil 75 de todas las ventas globales. De ser mayores o iguales, se asigna un 1. De lo contrario se asigna un 0.

7. ***join_dfs(df1, df2)***: Une el dataframe de las ventas con el de los ratings. Esta unión se hace a través de la columna 'Name' de el dataframe de ventas y la columna 'title' del dataframe de ratings.
8. ***write_df_in_db(df, table_name)***: Guarda un dataframe de PySpark en una base de datos de PostgreSQL.

El programa principal que llama a los métodos anteriores se encuentra en programa_principal.py

Para ejecutar esta sección del proyecto debe correr dentro del contenedor la instrucción: `spark-submit --driver-class-path postgresql-42.2.14.jar programa_principal.py`

Materialización en PostgreSQL

La materialización en la base de datos se realiza con la función ***write_df_in_db(df, table_name)*** expuesta en la sección anterior.

El esquema de bases en almacenadas en Postgres es el siguiente:

Ventas de videojuegos:

```
Ventas schema
StructType([
    StructField("Name", StringType()),
    StructField("Global_Sales", FloatType()),
    StructField("Genre", StringType()),
    StructField("Platform", StringType()),
    StructField("Publisher", StringType()),
    StructField("Developer", StringType()),
    StructField("Year", IntegerType()),
  ])
```

Ratings de la ESRB:

```
Ratings schema
StructType([
    StructField("title", StringType()),
    StructField("console", IntegerType()),
    StructField("alcohol_reference", IntegerType()),
    StructField("animated_blood", IntegerType()),
    StructField("blood", IntegerType()),
    StructField("blood_and_gore", IntegerType()),
    StructField("cartoon_violence", IntegerType()),
    StructField("crude_humor", IntegerType()),
    StructField("drug_reference", IntegerType()),
    StructField("fantasy_violence", IntegerType()),
    StructField("intense_violence", IntegerType()),
    StructField("language", IntegerType()),
    StructField("lyrics", IntegerType()),
    StructField("mature_humor", IntegerType()),
    StructField("mild_blood", IntegerType()),
    StructField("mild_cartoon_violence", IntegerType()),
    StructField("mild_fantasy_violence", IntegerType()),
    StructField("mild_language", IntegerType()),
    StructField("mild_lyrics", IntegerType()),
    StructField("mild_suggestive_themes", IntegerType()),
    StructField("mild_violence", IntegerType()),
    StructField("no_descriptors", IntegerType()),
    StructField("nudity", IntegerType()),
    StructField("partial_nudity", IntegerType()),
    StructField("sexual_content", IntegerType()),
    StructField("sexual_themes", IntegerType()),
    StructField("simulated_gambling", IntegerType()),
    StructField("strong_language", IntegerType()),
    StructField("strong_sexual_content", IntegerType()),
    StructField("suggestive_themes", IntegerType()),
    StructField("use_of_alcohol", IntegerType()),
    StructField("use_of_drugs_and_alcohol", IntegerType()),
    StructField("violence", IntegerType()),
    StructField("esrb_rating", StringType())
])
```

Nota: se almacena igual ya que no se necesita realizar ningún preprocesamiento.

Base con los datos cruzados:

```
Sales_rating schema
    StructField("Name", StringType()),
    StructField("Global_Sales", FloatType()),
    StructField("Global_Sales_scaled", FloatType()),
    StructField("Genre", StringType()),
    StructField("Platform", StringType()),
    StructField("Publisher", StringType()),
    StructField("Developer", StringType()),
    StructField("Year", IntegerType()),
    StructField("console", IntegerType()),
    StructField("alcohol_reference", IntegerType()),
    StructField("animated_blood", IntegerType()),
    StructField("blood", IntegerType()),
    StructField("blood_and_gore", IntegerType()),
    StructField("cartoon_violence", IntegerType()),
    StructField("crude_humor", IntegerType()),
    StructField("drug_reference", IntegerType()),
    StructField("fantasy_violence", IntegerType()),
    StructField("intense_violence", IntegerType()),
    StructField("language", IntegerType()),
    StructField("lyrics", IntegerType()),
    StructField("mature_humor", IntegerType()),
    StructField("mild_blood", IntegerType()),
    StructField("mild_cartoon_violence", IntegerType()),
    StructField("mild_fantasy_violence", IntegerType()),
    StructField("mild_language", IntegerType()),
    StructField("mild_lyrics", IntegerType()),
    StructField("mild_suggestive_themes", IntegerType()),
    StructField("mild_violence", IntegerType()),
    StructField("no_descriptors", IntegerType()),
    StructField("nudity", IntegerType()),
    StructField("partial_nudity", IntegerType()),
    StructField("sexual_content", IntegerType()),
    StructField("sexual_themes", IntegerType()),
    StructField("simulated_gambling", IntegerType()),
    StructField("strong_janguage", IntegerType()),
    StructField("strong_sexual_content", IntegerType()),
    StructField("suggestive_themes", IntegerType()),
    StructField("use_of_alcohol", IntegerType()),
    StructField("use_of_drugs_and_alcohol", IntegerType()),
    StructField("violence", IntegerType()),
    StructField("esrb_rating", StringType(),
    StructField("Successful", IntegerType()),
```

Nota: La columna 'Successful' se utilizará como la variable *target*.'

Modelos de predicción

Para esta sección se utilizó un notebook de Jupyter llamado **Esteban_Castillo_Proyecto.ipynb**. Para lograr correr Jupyter Notebooks en el contenedor se utiliza el Dockerfile proveído para la lección 5 y la tarea 3 del curso. Para configurar el contenedor se deben correr los siguientes comandos:

1. `docker build --tag proyecto .`
2. `docker run -i -t proyecto /bin/bash`

El objetivo predictivo es la clasificación de los juegos como exitosos o no en el mercado global basado en las clasificaciones que le asigna la organización ESRB.
Para clasificar los videojuegos se desarrollaron dos modelos:

1. **Regresión logística**: se escogió este modelo ya que es utilizado para problemas de clasificación.
2. **Árbol de decisiones**: se eligió ya que se adapta muy bien al problema de clasificación binaria.

Antes de entrenar el modelo se revisaron los valores atípicos de la columna 'Global_Sales_scaled' y se eliminaron. Porteriormente, se utilizó la función de PySpark **VectorAssembler** para generar el vector de las features para alimentar el modelo. Las columnas utilizadas fueron:

1. 'Global_Sales_scaled'
2. 'alcohol_reference'
3. 'animated_blood'
4. 'blood'
5. 'blood_and_gore'
6. 'cartoon_violence
7. 'crude_humor'
8. 'drug_reference'
9. 'fantasy_violence'
10. 'intense_violence'
11. 'language'
12. 'lyrics'
13. 'mature_humor
14. 'mild_blood'
15. 'mild_cartoon_violence'
16. 'mild_fantasy_violence'
17. 'mild_language'
18. 'mild_lyrics'
19. 'mild_suggestive_themes'
20. 'mild_violence'
21. 'no_descriptors'
22. 'nudity'
23. 'partial_nudity'
24. 'sexual_content'
25. 'sexual_themes'
26. 'simulated_gambling'
27. 'strong_janguage'
28. 'strong_sexual_content'
29. 'suggestive_themes'
30. 'use_of_alcohol'
31. 'use_of_drugs_and_alcohol'
32. 'violence'

Una vez generado el dataframe con los vectores features, se procedió a separar los datos en proporción 70/30 para generar los conjuntos de entrenamiento y prueba respectivamente.

Para entrenar ambos modelos se usó la columna 'Features' como el parámetro FeaturesCol y la columna 'Successful' como el parámetro labelCol. Luego se procedió a entrenar el modelo con el conjunto de datos de entrenamiento y a generar los objetos de validación cruzada. Finalmente se evalúa el modelo con el conjunto de datos de prueba.

Análisis de resultados

Al evaluar ambos modelos, se obtiene una precisión de 0,81 para el modelo de regresión logística mientras que el modelo de árboles de decisión muestra una precisión de 0,96. Adicionalmente, se calculó el área bajo la curva del precisión-recall donde el modelo de regresión logística obtuvo un 0,78 mientras que el modelo de árbol de decisiones obtuvo un 0,99. Por estas razones se considera el árbol de decisiones como el modelo superior para realizar predicciones sobre este conjunto de datos.