Plataforma de recomendación de canciones
Proyecto Gestión de Base de Datos

Natalia Puello Acosta, Eliana Romero Leon, Ivan Trujillo

3 de abril de 2023

1. Descripción general de los data set

1.1. Tracks

La base de datos *tracks* cuenta con 586672 registros y 20 variables dentro de las cuales 8 son categóricas y 12 numéricas, del total de datos se encuentra que 300367 datos son valores nulos lo cual representa el 2.6 % del dataset, esto es importante para poder emplear técnicas estadísticas que nos permitan dar solución a dicho problema. En el Cuadro 1 se pueden observar las variables como por ejemplo el ID de la canción, nombre de esta, popularidad, duración, artista, entre otros, y con el fin de realizar un análisis general se especifica el tipo de variable, el rango en la cual se encuentra, problema detectado de dicha variable y una breve descripción.

Nombre	Tipo	Rango	Problemas	Descripción
ID	Categórica	-		Id de la canción
Name	Categórica	-	Completitud	Nombre de la canción
Popularidad	Numérico	0-100	Completitud	Popularidad de la canción
Duration_ms	Numérico	3344-5621218	Completitud	Duración de la canción en milisegundos
Explicit	Categórica	0, 1, -1		Indica si la canción es o no explícita
Artists	Categórica	-		Nombre del artista
Id_Artists	Categórica	-		Id del artista
Release_date	Categórica	-	Conformidad	Año de publicación de la canción
Danceability	Numérico	0-0.991	Completitud	Que tanto permite bailar la canción, valores cercanos a cero no es bailable, 1 más bailable
Energy	Numérico	0-1	Completitud	Medida perceptual de actividad e intensidad
Key	Numérico	0-11	Completitud	Pitch de la canción
Loudness	Numérico	-60-5.3	Completitud	Amplitud, se mide en db
Mode	Categórica	0, 1	Completitud	Modalidad(mayor o menor) de la canción
Speechness	Numérico	0-0.97	Completitud	Detecta presencia de palabras en la canción, valores cercanos a, 0 canciones sin palabras
Acousticness	Numérico	0-0.996	Completitud	Qué tanta confianza hay de que el track sea acústico (1 más certeza)
Instrumentalness	Numérico	0-1	Completitud	Predice si la canción tiene vocals, cercanos a 1 no vocals
Liveness	Numérico	0-1	Completitud	Detecta presencia de audiencia, más cercano a 1, tracks es en vivo
Valence	Numérico	0-1	Completitud	Qué tan positiva o feliz es la canción, 1 muy feliz
Tempo	Numérico	0-246	Completitud	Tempo estaimado en BPM
Time_signature	Categórica	0, 1, 3, 4, 5	Completitud	Cuantos Beats hay en cada bar

Cuadro 1: Descripción data set Tracks.

1.2. Artists

Como se presenta a continuación en el Cuadro 2 la base de datos *Artist* está compuesta por 5 variables. De las cuales, tres son categóricas y dos son numéricas.

Dado que el principal objetivo del presente trabajo es usar la información disponible para implementar un sistema de recomendación, se consideraron relevantes las variables de popularidad, seguidores y géneros esta última es la que mas preprocesamiento requirió dado los problemas de completitud y alta cardinalidad de categorías asociadas.

Nombre	Tipo	Rango	Problemas	Descripción
ID	Categórica	-		Id del artista
Name	Categórica	-		Nombrel artista
Followers	Numérico	0-78900234	Completitud	Seguidores del artista
Popularity	Numérico	0-100		Popularidad del artista
Genres	Categórica	-	Completitud, Alta heterogeneidad	Multiples géneros asociados al artista

Cuadro 2: Descripción data set Artists.

2. Análisis de calidad y procesos de preparación y limpieza de datos

2.1. Tracks

Inicialmente se eliminaron las filas cuyo nombre de canción se encontraba vacío, dado el objetivo del proyecto, se considera que el nombre de la canción es clave al momento de devolver la lista con las recomendaciones al usuario. Adicionalmente se unificó el formato de la variable *release_ date*, ya que se encontró que la fecha estaba en formatos diferentes (y/m/d ; m/d/y etc) se conservó sólo el año para todos para poder manejarlo con más facilidad más adelante.

Por otro lado, la variable *duration_ ms* se pasa de milisegundos a minutos para tener un valor más fácil de interpretar. Una vez realizada la conversión se encontró que hay canciones que duran desde 4 segundos hasta 93 minutos. Se hizo una prueba Shapiro Wilk para probar la normalidad de la variable y basado en esto determinar aproximadamente a cuántos valores estándar de la media se puede categorizar un valor como atípico. El p-value de la prueba fue cero, por lo tanto, la variable no se comporta normal (cosa que también podía ser vista en el histograma de la misma). Se clasificará como atípicos los valores que se encuentren a más de 7 desviaciones estándares de la media, se encuentran 1282 como se puede observar en la Figura 1. Sin embargo, se decide mantener todos los registros a pesar de su atipicidad, se considera que podrían ser útiles más adelantes para el modelo de recomendación.

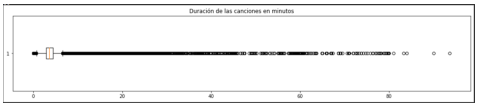


Figura 1: Valores atípicos de la variable tiempo.

Un común denominador del dataset que se logró identificar va relacionado a los valores vacíos para las variables cuantitativas *energy*, *danceability*, *key*, *loudness*, *mode*, *speechness*, *acousticness*, *instrumentalness*, *liveness*, *valence* y *tempo*, por lo cual procedimos a hacer una imputación usando el promedio de cada una de estas. Como último paso, se identifica que la variable *speechness* toma valores entre 0.66 y 1 si la canción está hecha completamente de palabras, 0.33 y 0.66 si tiene tanto música como palabras y menores a 0.33 si es en su mayoría música. Se decide convertir esta variable a categórica con los valores Alto SP, Medio SP y Bajo SP, para tener datos más interpretables.

2.2. Artists

A continuación se describe el tratamiento de la variable géneros(Genres); Primero se construyó un arreglo(diccionario) que contiene todos los posibles géneros musicales presente en la base de datos que son aproximadamente 5600, a partir de ahí se determinó una taxonomía para agrupar dichos géneros, utilizándose como referencia las siguientes categorías:*'unknown','disco','metal', 'blues', 'jazz', 'rock', 'rap', 'hip hop', 'reggae', 'pop', 'indie', 'ballad', 'folclor', 'folk', 'gospel', 'ska', 'punk', 'country', 'electronic','soul', 'opera', 'cumbia', 'techno', 'alternative', 'bolero', 'trap','vallenato', 'grunge','corrido','flamenco', 'trio', 'motivation', 'classical', 'instrumental', 'funk', 'hardcore', 'bachata','merengue', 'salsa', 'ranchera', 'orchestra', 'tango',*

'opera', 'son cubano', 'banda','percusion', 'samba', 'mambo' y *'other'*.

Con el objetivo de reducir la heterogeneidad de las categorías presentes en cada uno de los géneros por ejemplo; *rock en español, german rock, austrian rock* cada una de estas se mapeó a la jerarquía superior es decir al género *rock*. Cabe señalar que *Unknown*, and *other* son categorías especiales debido a que la primera indica la cantidad de registros en los cuales se desconoce el género porque no se registra y la última agrupa todos los géneros diferentes a los mencionados anteriormente. Para llevar a cabo dicha clasificación se utilizó **Fuzzy-string matching**, con el objetivo de poder hacer mejores clasificaciones, posteriormente se reemplazó en la base de datos cada uno de los géneros asociados a cada artista, por sus homólogos estandarizados. No obstante, la nueva clasificación no es suficiente para poder ser utilizado en un **modelo de similitudes**, además que en un artista particular pueden aparecer los diferentes géneros estandarizados por ejemplo *(rock,other,folk)* por lo que una selecciona aleatoria de un género representativo podría hacernos perder información relevante si selecciona la categoría 'other' para representarlo, por lo que se le dió una puntuación negativa para no ser seleccionada como género representativo, determinándose finalmente como representativo el primer género diferente a las etiquetas *other* y *unknown*.

El análisis de las variables categóricas se realizó a través de medidas de frecuencia relativa y absoluta, para las variables cuantitativas se utilizó la media y desviación estándar en caso de que dichas variables fueran normales y la mediana y el rango intercuantílico en el caso contrario, para evaluar la normalidad se utilizó el test de Shapiro-Wilk considerándose significancia estadística para valores de $P < 0{,}05$.

3. Análisis exploratorio de datos

3.1. Tracks

Basado en el perfilamiento de datos realizado se encontró que hay una alta correlación entre las variables energy, valence, danceability, acousticness y loudness. Por lo cual, se procede a calcular cada una de dichas relaciones, obteniendo una correlación de 0.48 entre danceability - valence, de 0.74 entre energy - loudness, de -0.69 entre energy - acousticness y finalmente de 0.5 entre loudness - acousticness. Adicionalmente, como se puede evidenciar en la Figura se grafica la correlación entre energy - loudness ya que obtuvo un mayor valor que las otras.

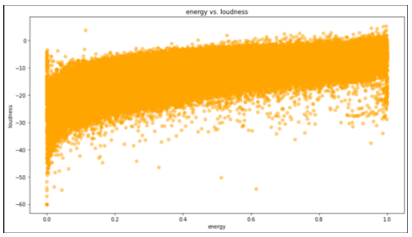


Figura 2: Correlación Energy - Loudness.

3.2. Artists

De acuerdo a lo observado en el Cuadro 3 para la base de datos *artist* se obtuvieron 1,162,095 registros no únicos, para los cuales se registra una media de 10,220 seguidores (Followers) con una mediana de 2.0 en popularidad (Popularity). Cabe destacar que aproximadamente, el 73.7 % de los géneros fueron clasificados dentro de la categoría 'Unknown', lo cual implica la evaluación de la presencia o no de un *sesgo de selección*, dado que esta información es relevante para determinar la capacidad de inferencia y evaluarse también la pertinencia misma de la taxonomía aquí empleada. No obstante, como se observa en el mismo cuadro, la siguiente categoría con mayor participación es *'Other'* con un 8,9 %, seguido de Pop, Indie, Rock, Metal y Rap con participaciones de 3.1 % , 1.8 % , 1.8 %, 1.3 % y 1.2 % respectivamente.

Cuadro 3: Artist

		Missing	Overall
n			1162095
Followers, mean (SD)		11	10220.7 (254399.5)
Popularity, median [Q1,Q3]		0	2.0 [0.0,13.0]
Main genre, n (%)	Alternative	0	3229 (0.3)
	Bachata		611 (0.1)
	Ballad		572 (0.0)
	Banda		1899 (0.2)
	Blues		3001 (0.3)
	Bolero		414 (0.0)
	Classical		10257 (0.9)
	Corrido		1356 (0.1)
	Country		2378 (0.2)
	Cumbia		1768 (0.2)
	Disco		1637 (0.1)
	Electronic		5689 (0.5)
	Flamenco		644 (0.1)
	Folclor		547 (0.0)
	Folk		8695 (0.7)
	Funk		2324 (0.2)
	Gospel		2536 (0.2)
	Grunge		520 (0.0)
	Hardcore		5355 (0.5)
	Hip hop		5832 (0.5)
	Indie		21408 (1.8)
	Instrumental		2154 (0.2)
	Jazz		8937 (0.8)
	Mambo		103 (0.0)
	Merengue		155 (0.0)
	Metal		15076 (1.3)
	Motivation		318 (0.0)
	Opera		640 (0.1)
	Orchestra		476 (0.0)
	Other		103931 (8.9)
	Percusion		427 (0.0)
	Pop		36334 (3.1)
	Punk		7431 (0.6)
	Ranchera		814 (0.1)
	Rap		13764 (1.2)
	Reggae		3096 (0.3)
	Rock		20692 (1.8)
	Salsa		423 (0.0)
	Samba		1107 (0.1)
	Ska		1543 (0.1)
	Son cubano		110 (0.0)
	Soul		2032 (0.2)
	Tango		482 (0.0)
	Techno		4115 (0.4)
	Trap		107 (0.0)
	Trio		424 (0.0)
	Unknown		856500 (73.7)
	Vallenato		232 (0.0)

4. Modelo de datos relacional de la base de datos PostgreSQL

Con los datos limpios se procede a hacer la carga de estos a Postgres. Primero se creó una base de datos en Postgres llamada "Proyecto" y se cargaron a estas 3 tablas: genres, artists y tracks.

La tabla Genres se crea a partir de los géneros que quedaron en la tabla artists (48 géneros diferentes), a estos se les asignó un identificador único (números del 1 al 48). En esta tabla el primary key es id del género.

En artists se remplazan los géneros por el identificador creado anteriormente y se cargó la tabla. En esta el primary key es el id del artista y se asignó una relación de llave foránea con la tabla genres (con los ids de género); también hay una llave foránea entre id de artista y id de artista en la tabla tracks.

Para tracks se tuvo que hacer un paso de limpieza adicional, se eliminaron los corchetes cuadrados y comillas que se encontraban en id artist y artist para que quedaran en el mismo formato que la tabla artista. La primary key es Id del tracks y tiene una relación de llave foránea con la tabla aristas en id de aristas.

Así se visualizan las tablas en Postgres. Hay una relación one to many (1:m) entre el id de la tabla género y el id género en la tabla artistas. Y una también one to many (1:m) entre el id del artista en artists y en tracks.

Para realizar el proceso de carga se debieron realizar los siguientes pasos:

- Se deben correr los scripts artists y tracks, estos hacen el proceso de extraer y transformar los datos originales. Donde se obtiene cvs tracks_new, remaster y lastbase.
- Correr el script intermedio llamado Intermedio Artist, de este se obtiene el csv artists_new, que contiene las variables originales del dataset artista pero con los géneros creados en la limpieza de los datos.
- Correr script Load SQL este se alimenta de los csv tracks_new y artists_new para hacer el cargue de las tablas a Postgres.

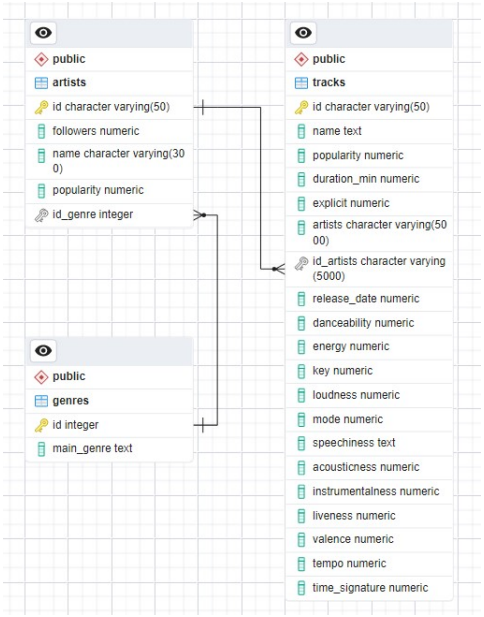


Figura 3: Esquema de relaciones de tablas en PostgreSQL.

	id [PK] integer	main_genre text
1	1	unknown
2	2	other
3	3	classical
4	4	pop
5	5	folk
6	6	electronic
7	7	instrumental
8	8	hip hop
9	9	rap

Figura 4: Evidencia #1 de la carga de tablas a PosgtreSQL.

	id [PK] character varying (50)	followers numeric	name character varying (300)	popularity numeric	id_genre integer
1	0001cekkfdEBoMlwVQvpLg	80.0	Jordan Colle	0	1
2	0001K3ON4ACv7hte9Jlc6o	1.0	Chris	0	1
3	0001wHqxbF2YYRQxGdbyER	6329.0	Motion Drive	19	2
4	0001ZVMPt41Vwzt1zsmuzp	13667.0	Thyro & Yumi	39	4
5	0002XY9y3JhjzTZqNCqEcv	1828.0	Katie Price & Peter Andre	0	1
6	00045gNg7mLEf9UY9yhD0t	829.0	Kubus & BangBang	13	8
7	0004m70bDmySiyX0mbJM...	0.0	Carla Sanders	0	1
8	0004mMWDlIBRhLMZP3p9...	4.0	Yung Danzy	0	1
9	00052qkj8osxzuMhyZA88Y	1.0	Stephen Olaussen	0	1

Figura 5: Evidencia #2 de la carga de tablas a PosgtreSQL.

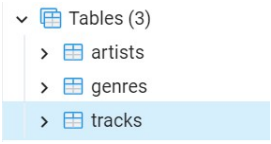


Figura 6: Evidencia #3 de la carga de tablas a PosgtreSQL.

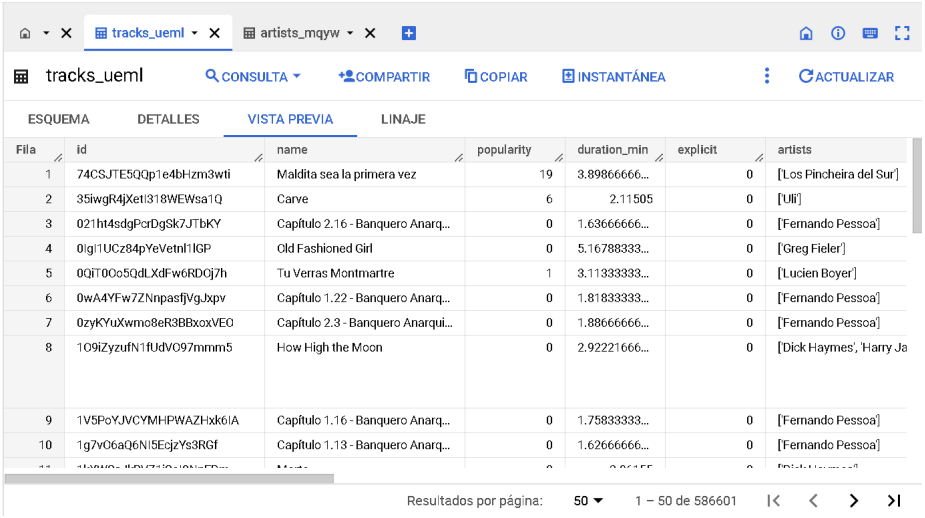
5. Esquema de la bodega de datos en BigQuery

En esta sección se describe brevemente el proceso de carga de los datos a BigQuery una vez finalizado el proceso de limpieza y transformación de estos. Para ello inicialmente se debió crear una cuenta en la plataforma, después se procedió a crear un proyecto llamado "Gestion_bases_datos".

Una vez creado el proyecto, se obtiene las credenciales de acceso en formato json la cual se utiliza en el script automatizado para realizar el proceso de carga. Cabe resaltar que fue necesario identificar los tipos de variables de cada Dataframe a subir ya que para ejecutar el código se deben describir una a una.

En la Figura 7 se puede observar la tabla de artistas y de tracks en la plataforma Google BigQuery. Adicionalmente en las Figuras 8 y 9 se evidencia la vista previa de la tabla tracks en la plataforma y parte del código de carga respectivamente.

Figura 7: Proyecto en BigQuery con tablas cargadas.



Fila	id	name	popularity	duration_min	explicit	artists
1	74CSJTE5QQp1e4bHzm3wti	Maldita sea la primera vez	19	3.89866666...	0	['Los Pincheira del Sur']
2	35iwgR4jXetI318WEWsa1Q	Carve	6	2.11505	0	['Uli']
3	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarq...	0	1.63666666...	0	['Fernando Pessoa']
4	0lgI1UCz84pYeVetnl1IGP	Old Fashioned Girl	0	5.16788333...	0	['Greg Fieler']
5	0QiT0Oo5QdLXdFw6RDOj7h	Tu Verras Montmartre	1	3.11333333...	0	['Lucien Boyer']
6	0wA4YFw7ZNnpasfjVgJxpv	Capítulo 1.22 - Banquero Anarq...	0	1.81833333...	0	['Fernando Pessoa']
7	0zyKYuXwmo8eR3BBxoxVEO	Capítulo 2.3 - Banquero Anarqui...	0	1.88666666...	0	['Fernando Pessoa']
8	1O9iZyzufN1fUdVO97mmm5	How High the Moon	0	2.92221666...	0	['Dick Haymes', 'Harry Ja
9	1V5PoYJVCYMHPWAZHxk6IA	Capítulo 1.16 - Banquero Anarq...	0	1.75833333...	0	['Fernando Pessoa']
10	1g7vO6aQ6NI5EcjzYs3RGf	Capítulo 1.13 - Banquero Anarq...	0	1.62666666...	0	['Fernando Pessoa']

Figura 8: Tabla Tracks cargada en BigQuery.

```
credentials = service_account.Credentials.from_service_account_file("/content/gestion-bases-datos-5cb2829adf4e.json", scopes=["https:/

[11] client = bigquery.Client(credentials=credentials, project=credentials.project_id)

# Creating the job config
job_config = bigquery.LoadJobConfig(
    schema=[
        # Supported datatypes: https://cloud.google.com/bigquery/docs/reference/standard-sql/data-types
        bigquery.SchemaField("id", bigquery.enums.SqlTypeNames.STRING), #revisar string
        bigquery.SchemaField("name", bigquery.enums.SqlTypeNames.STRING),
        bigquery.SchemaField("popularity", bigquery.enums.SqlTypeNames.INT64),
        bigquery.SchemaField("duration_min", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("explicit", bigquery.enums.SqlTypeNames.INT64),
        bigquery.SchemaField("artists", bigquery.enums.SqlTypeNames.STRING),
        bigquery.SchemaField("id_artists", bigquery.enums.SqlTypeNames.STRING),
        bigquery.SchemaField("release_date", bigquery.enums.SqlTypeNames.INT64),
        bigquery.SchemaField("danceability", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("energy", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("key", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("loudness", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("mode", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("speechiness", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("acousticness", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("instrumentalness", bigquery.enums.SqlTypeNames.FLOAT64),
        bigquery.SchemaField("liveness", bigquery.enums.SqlTypeNames.FLOAT64),
```

Figura 9: Parte del código de carga a BigQuery.

6. Arquitectura general del la solución

6.1. Primera arquitectura

En la Figura 10 se puede evidenciar un diagrama en bloques el cual representa el proceso de extracción, transformación y carga tanto a PostgreSQL como BigQuery. Cabe resaltar que en los bloques de tranformación se evidencian los pasos más importantes de este proceso.

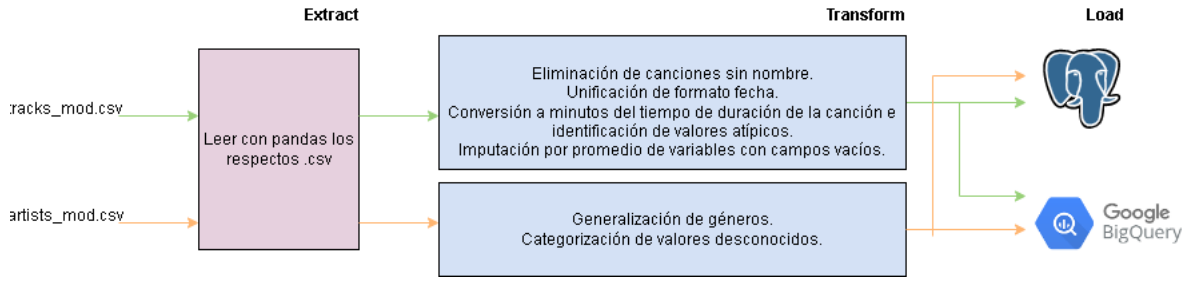


Figura 10: Diagrama de arquitectura ETL.

6.2. Arquitectura final

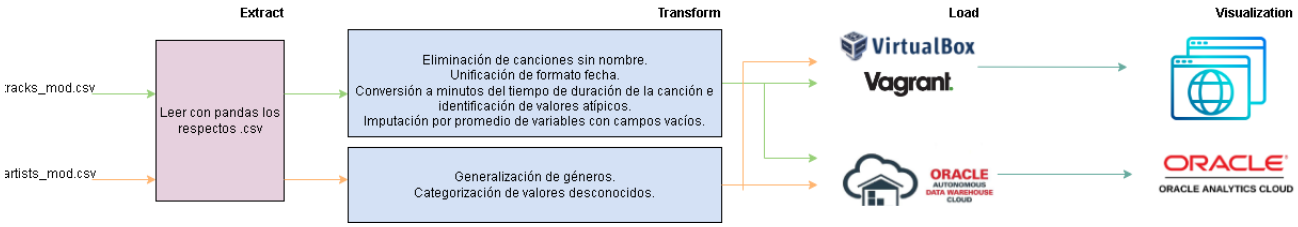


Figura 11: Diagrama de arquitectura final.

Para la creación del dashboard de este proyecto se hizo uso de dos herramientas de Oracle, Autonomous Data Warehouse y Oracle Analytics Cloud. Antes de hacer el dashboard, lo primero que se hizo fue subir los data sets de artistas y canciones, limpios y con todas las transformaciones mencionadas en los apartados anteriores, al Data Warehouse de Oracle aprovisionado con 1 OCPU y 1TB de memoria. Esta es una herramienta de Oracle optimizada para cargas de trabajo analíticas que ayuda a incrementar rendimiento y reducir costos operativos. Con la información en la nube, hicimos la conexión con el visualizador Oracle Analytics Cloud.

Para el desarrollo del aplicativo web se utilizó virtualbox y vagrant para no tener problemas de compatibilidad, el archivo que describe dicha virtualización con las librerías necesarias para la implementación del aplicativo es:

```
Vagrant.configure("2") do |config|

    config.vm.box = "ubuntu/bionic64"

    config.vm.provision "shell", inline: <<-SHELL
      sudo apt-get update
      sudo apt-get install -yq \
```

```
      build-essential \
      tree \
      python-dev \
      python3-minimal \
      python3-pip \
      python3-dev \
      python3-venv
    sudo apt-get update
    pip3 install --upgrade pip
    pip install pandas matplotlib scikit-learn flask pickle5 annoy
  SHELL

  config.vm.network :forwarded_port, guest: 5000, host: 1234, host_ip: "127.0.0.1"
end
```

Lo que implica que se puede acceder al aplicativo de recomendación a través de un navegador con la siguiente dirección (posteriormente se haya ejecutado en terminal el archivo app.py)

```
http://0.0.0.0:1234/
```

6.2.1. Workflow

Posterior a activar el entorno de desarrollo por medio de:

```
vagrant up
vagrant ssh
```

Y estar en la carpeta compartida la máquina local y virtual en

```
cd /vagrant
```

Se ejecuta

```
python3 training.py
python3 app.py
```

Esto permitirá acceder al aplicativo que se está ejecutando en la url anteriormente mencionada.

El *framework* usado para el desarrollo del aplicativo es **flask**, debido a que es ligero y se encuentra documentación acerca de la implementación de modelos **ML**.

7. Sistema de recomendación

Con el objetivo de que el usuario, tenga una mejor experiencia de uso del aplicativo se ha desarrollado un prototipo básico con el siguiente esquema desplegado en WEB:

El usuario observará en la interfaz las d canciones con mas valor de un conjunto f de características como *popularity,danceability*, posteriormente de acuerdo a las *preferencias* de los usuarios podrán ver un conjunto de k canciones recomendadas basadas en similitud.

Popularity	**Danceability**
Song A	Song 1
Song B	Song 2
Song C	Song 3

Cuadro 4: Interfaz web

El usuario puede dar clic sobre cualquier canción listada en la interfaz y obtener una recomendación de k canciones, por ejemplo:

Recommended song One - ARTIST A	Reccomended Song Two- - ARTIST B	Recommended Song Three - ARTIST C

Cuadro 5: Canciones recomendadas

Puede observar un ejemplo de como funciona en el siguiente link ejemplo.

7.1. Caracteristicas de entrenamiento

Para el entrenamiento, como referencia se utilizó el siguiente vector de caracteristicas; *'popularity','danceability', 'energy','loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'time signature'* para el entrenamiento se dejó el hiperparámetro igual a 10 como viene por defecto en el ejemplo de la documentación.

7.2. Desventajas y cosas por mejorar

Debido a que el algoritmo está basado en KNN el sistema se ve sujeto al problema de la dimensionalidad por lo que al aumentar las características de entrenamiento el concepto de similitud desaparece lo cual no permite una diferenciación correcta, no por que no sean diferentes si no debido efecto que tiene un mayor número de dimensiones sobre la distancia (*sparsity*). Se recomienda utilizar un algoritmo más robusto, así como un método de almacenamiento del registro previo si el usuario ya ha ingresado previamente al aplicativo.

8. Dashboard

A continuación, se describirán las secciones que se pueden encontrar en dashboard.

- Vista general de Tracks: En esta sección se puede encontrar el total de canciones en la base de datos, la popularidad promedio de las canciones y su duración promedio en minutos.
- Vista general de Artists: Se puede observar la cantidad total de artistas, una cuenta de la popularidad, seguidores de todos los artistas y promedio de la popularidad.
- Top 10 de artistas con mayor popularidad: esta es una visual muy importante para el negocio ya que permite conocer quienes dominan en popularidad en la plataforma. A partir de esto se podrían crear estrategias para recomendar canciones similares, pero con artistas que tienen baja popularidad, con el fin de impulsar el crecimiento de otros artistas no tan reconocidos.
- Top 10 de géneros con mayor cantidad de seguidores: en este se pueden ver los géneros que corresponden a los artistas con más seguidores en la plataforma. Se analiza que el género Pop domina, con casi 50 % del total.
- Top 10 de géneros por popularidad: Este gráfico se puede contrastar con el anterior y una conclusión importante para el negocio sería ser un poco más específicos a la hora de clasificar los géneros de las canciones, dado que a pesar de ser pop el género más seguido.
- Popularidad de las canciones a partir del año de lanzamiento: Esta gráfica permite identificar de qué periodo de tiempo le gustan más las canciones a los usuarios de la app. En este caso, se puede ver que las canciones más populares son del 2016 al 2022, es decir, los usuarios disfrutan más de canciones nuevas. Dependiendo de la estrategia que quiera lanzar el negocio se pueden tomar decisiones. Si se quiere mantener esta tendencia de escuchar canciones nuevas, la empresa debe estar a la vanguardia y constantemente añadiendo canciones que acaban de ser lanzadas, con el fin de no perder el interés de los usuarios.

- Popularidad dependiendo del Speechiness: Este boxplot permite entender que tanta presencia de palabras hay en una canción contra su nivel de popularidad. Se puede ver que a los usuarios, en promedio, les gustan más las canciones con baja y media presencia de palabras. Esto debe ser tenido en cuenta en el futuro para añadir canciones que tengan más que todo medio y bajo SP, suponiendo que el objetivo sea seguir añadiendo canciones que se adapten a los gustos de los usuarios.
- Porcentaje de popularidad dependiendo del Spreechiness: Esta gráfica es solo una extensión de lo mencionado en el punto anterior, esta muestra los porcentajes exactos.
- Histogramas: Se grafican los valores de energía y popularidad, con el fin de tener un mejor entendimiento del comportamiento de estas variables.

Indicadores importantes para el negocio:

- Géneros con mayor cantidad de seguidores
- Géneros más populares
- Artistas con mayor popularidad

9. Repositorio y video

En el siguiente repositorio (Github) se encuentran los archivos, que sustentan el proceso de desarrollo anteriormente descrito (click aquí). Adicionalmente, se realizo un video final donde se muestran los resultados obtenidos tanto de la aplicación como del dashboard, en el siguiente link se puede ver. (click aquí).

Referencias

[1] J. Parra, "Análisis exploratorio y análisis confirmatorio de datos," *Espacio abierto*, vol. 11, no. 1, pp. 115–24, 2002.

[2] F. M. Ocaña Peinado, "Análisis descriptivo y exploratorio de datos Análisis Exploratorio de Datos," *Ugr.Es*, p. 25, 2021.

[1] [2]