IIC2413 – Bases de Datos

Control III
Data Science - NoSQL

Entrega

El plazo para la entrega es Viernes 26 de Noviembre, hasta las 23:59. La entrega debe hacerse subiendo un jupyter notebook (`.ipynb`) a canvas con los análisis pedidos en el enunciado y que se pueda ejecutar teniendo el archivo de la base de datos en la misma carpeta. Si necesitas agregar algo más puedes subir un `.zip` con todos los archivos, pero debes incluir un `README` en markdown explicando a qué corresponde cada archivo.

Introducción

El Billboard 200 es un ranking semanal que contiene los álbumes más populares en USA. En este control usaremos el historial del ranking entre el 5/1/1963 y el 19/1/2019, para hacer algunos análisis interesantes y practicar nuestras recientemente adquiridas habilidades de Pandas, MongoDB y SQL.

Los datos que usaremos en este control son los siguientes:

- `billboard-200.db`: Es una base de datos con dos tablas: `albums`, que contiene la semana, artista, nombre del álbum, la posición en el ranking, cantidad de canciones y duración en milisegundos, y la tabla `acoustic_features` que tiene descriptores acústicos de algunas de las canciones de los albumes del ranking, obtenidos a partir de la api de spotify. Además de eso la tabla tiene el id de spotify, nombre de la canción, nombre del álbum, artista, id del album en spotify y fecha de lanzamiento. Para más info ver acá.
- Un set de datos con $\sim 50K$ letras de canciones obtenidas a partir de la API de Genius. Estas están disponibles para ser consultadas remotamente conectándote a una instancia de MongoDB que levantamos (las letras tienen un índice de texto).

1. Instrucciones para manejo de datos y consultas

Queremos que uses la librería sqlite3 de Python para consultar la base de datos de billboard (`billboard-200.db`) para luego procesar los resultados con pandas, de forma similar a como lo hicimos en la guía de data science. Por otro lado, para consultar las letras debes conectarte a nuestro servidor de mongo usando la librería pymongo y el siguiente snippet:

```
from pymongo import MongoClient

uri = "mongodb://alumnoXX:XX@gray.ing.puc.cl/control3?authSource=perm"
client = MongoClient(uri)
db = client.get_database('control3')
collection = db.lyrics
```

Donde XX es tu RUT con dígito verificador y guión (sin puntos). Intenta ser cuidadoso al consultar a Mongo y hacer el control con anticipación para que no tengamos problemas de alta carga en el servidor. Decidir qué cosas calculas con SQL, cuáles con Mongo y cuales con pandas es decisión tuya.

2. Tareas a realizar

En esta sección se describen las tareas a realizar para el control. Hacer un análisis preliminar de los datos (ej: revisar si hay nulos o ruido) antes de completar lo que se pide es responsabilidad tuya. Puedes usar cualquier librería de visualización de python para generar los gráficos.

2.1. Análisis Billboard

1. Antes de hacer cualquier análisis queremos saber la completitud de los datos, para eso queremos calcular para cada semana en el periodo, cuantos discos de los 200 totales se encuentran en la tabla. Calcula el % para cada semana y haz un linechart con la semana en el eje X y el porcentaje en el Y.

2. Ahora queremos saber cuales son los artistas más importantes en el período (según billboard (?)) para eso ve cuales fueron los 10 artistas que más veces aparecieron en el ranking.

3. Más importante, nos interesa saber cual es el mejor álbum de la historia... o algo así. Para eso cuenta las apariciones de cada álbum en el ranking. Muestra los 10 que más aparecieron. Adicionalmente nos interesa saber como distribuye este valor, para ver eso haz un histograma donde la cantidad de apariciones está en el eje X.

2.2. Análisis descriptores acústicos

1. En primer lugar debemos entender como distribuyen los descriptores acústicos obtenidos a partir de la api de Spotify, para eso haz un histograma de los descriptores: `acousticness`, `danceability`, `energy`, `instrumentalness`, `liveness`, `loudness`, `speechiness`, `valence` (uno por descriptor). Comenta lo que veas, ¿hay alguna tendencia en los tracks que aparecen en el ranking?.

2. El análisis anterior es interesante pero podemos avanzar un paso más considerando las veces que cada canción estuvo en el ranking. Repite los histogramas anteriores pero esta vez si una canción estuvo 10 veces en el ranking entonces sus descriptores deben estar 10 veces en el histograma. Comenta lo que veas, ¿Hay alguna tendencia?, ¿Se ven diferencias respecto a los histogramas anteriores?.

3. Calcula la correlación entre los descriptores. Comenta.

4. Como el ranking es a nivel de álbum, queremos también tener los descriptores a nivel de álbum. Genera estos descriptores tomando la media de los descriptores de todas las canciones del álbum. Calcula la correlación de estos descriptores con la cantidad de apariciones de cada álbum en el billboard. Para los 2 con mayor correlación haz un scatter plot para visualizar.

5. La columna `key` representa la tonalidad en que esta la canción. Construye un pie chart en que se muestra que % de las canciones en el ranking están en cada tonalidad. Hint: Los enteros mapean a notas según como se describe en la documentación de la api.

2.3. Análisis Letras

1. Analizaremos si la temática de las letras afecta el ranking y los descriptores. Para eso obtiene todas las canciones en el dataset que tengan las palabras `love`, `happy` o `joy` (u otras similares que considere relevantes) y todas las que tengan `hate`, `sad` o `cry` (u otras similares que considere relevantes). Usando estos 2 conjuntos de canciones construye 2 conjuntos mutualmente exclusivos (con intersección vacía). ¿Hay alguna diferencia respecto a como se comportan los descriptores acústicos en estos dos conjuntos? ¿Y las posiciones en el ranking? Comente y además sugiera una forma de construir de mejor manera este análisis.