

CIn - Centro de Informática
UFPE

Gabriel Henrique Daniel da Silva

Recife, 04 de Julho de 2018

Metodologia

Ambiente de Desenvolvimento

Para o desenvolvimento do projeto, optamos por utilizar um ambiente de simples configuração, resposta rápida e com uma variedade grande de ferramentas já disponíveis para serem utilizadas.

Por isso optamos pela distribuição anaconda, que trás consigo a plataforma jupyter notebook que permite executar o código em células que funcionam como blocos de código interpretados em tempo real. A plataforma permite a utilização das linguagens Julia, Python e R.

Bibliotecas e Apis Utilizadas

Pandas: Biblioteca que facilita o uso de estrutura de dados. No caso do nosso projeto especificamente o uso de dataframes para observar melhor os atributos musicais retornados.

Spotipy: Biblioteca python para a API do spotify, facilitando e agilizando o desenvolvimento.

ScikitLearn: Biblioteca que traz uma série de ferramentas e algoritmos utilizados para aprendizagem de máquina. No nosso caso, utilizamos a mesma interessados especificamente nos algoritmos de clusterização, dos quais, utilizamos o K-means.

IPyWidgets: Biblioteca que fornece métodos simples

Desenvolvimento

IPyWidgets

`widgets.text(description, continuous_update)` : Exibe um campo de texto, permitindo que o usuário entre com o parâmetro (string) desejado.

`widgets.IntText(description, continuous_update)` : Exibe um campo de texto, semelhante ao método anterior `.text`. Porém recebendo um inteiro como parâmetro.

Spotipy

`prompt_for_user_token(username, scope, clientID, clientSecret, redirectUri)`: Método para autenticação do usuário diretamente com o spotify, permitindo livre acesso a todos os métodos da api (de acordo com as permissões recebidas no scope).

`user_playlist(user, idPlaylist)`: Recebe um usuário com o id de uma determinada playlist sua, e a retorna como um objeto.

`audio_features(idMusic)`: Recebe o id de uma determinada música e retorna um conjunto com uma série de atributos musicais relacionadas a mesma. Entre eles estão: Danceability, Acousticness, Duration, Energy, Instrumentalness, Loudness etc.

`user_playlist_create(idUser, namePlaylist, public)`: Realiza a criação da playlist de acordo com o nome dado pelo usuário passando o seu id, especificando se a playlist ficará pública ou privada.

`user_playlist_add_tracks(user, idPlaylist, listOfSongs)`: Adiciona uma lista de músicas selecionadas (`listOfSongs`) a uma determinada playlist de um usuário.

Pandas

`drop(columns)`: Recebe colunas que serão deletadas do dataframe em virtude de não serem úteis no momento.

`dataFrame()`: Realiza a criação de um objeto dataframe, bastante útil para visualizar e entender melhor os dados que foram recebidos em formato json, para decidir de que forma tratá-los ou utilizá-los.

SciKitLearn

`Kmeans(n_clusters)`: Seta o número de clusters que será utilizado pelo algoritmo de clusterização Kmeans.

`fit(dataFrame)`: Aplica o algoritmo de aprendizagem selecionado a um determinado dataframe passado como parametro.

`labels_`: Retorna os “resultados”, ou seja, os clusters a qual cada um dos elementos do dataframe foram atribuidos.

Referências

SciKitLearn Machine Learning in Python. Disponível em:
<<http://scikit-learn.org/stable/index.html>>. Acesso em: 26 de Maio de 2018

Pandas Powerfull Python Data Analysis Toolkit. Disponível em:
<<http://pandas.pydata.org/pandas-docs/stable/>>. Acesso em: 17 de Maio de 2018.

Jupyter Widgets. Disponível em:
<<https://ipywidgets.readthedocs.io/en/latest/>>. Acesso em: 25 de Junho de 2018.

Python lightweight library for the Spotify Web API. Disponível em:
<<http://spotipy.readthedocs.io/en/latest/>>. Acesso em: 17 de Maio de 2018.