

**Primeiro Projeto de Big Data and Cloud Computing:**

**Sistema de Recomendação de Filmes**

**Professor:** Eduardo Marques

**Tiago Coelho:** up201604170

**Edson Ferreira:** up201607407

**Porto, 04/2020**

**Introdução**

Para este projeto foi-nos proposta a elaboração de um sistema de recomendação de filmes através de uma pesquisa por palavras-chave. O trabalho proposto consiste na leitura, no seu subsequente pré-processamento e armazenamento naGoogle Cloud Storage, de dados não tratados dos conjuntos fornecidos pelo professor baseados na informação contida em MovieLens e IMDB. Posteriormente, os dados pré-processados serão convertidos em tabelas BigQueryatravés de uma *Loader Cloud Function* ativada por um sistema de *publish/subscribe.* Por fimo sistema de recomendações funciona através de uma *Search Cloud Function* que está encarregue das consultas às tabelas BigQueryde modo a retornar a lista de resultados pretendida.Foi nos também sugerido implementar diferentes estratégias para além da pesquisa por palavras-chave (trabalho extra) como a procura baseada em pesos e o índice de Jaccard.

Para toda a implementação foi utilizado o Google Colab para a programação em Python, o Apache Spark para a manipulação e criação dos conjuntos de dados, SQL para realizar as consultas às tabelas armazenadas, assim como as tecnologias da Google Cloud previamente mencionadas.

**Trabalho Realizado**

Das tarefas propostas conseguimos realizar na totalidade a recomendação com base em palavras-chave assim como a recomendação de acordo com um índice de similaridade a um identificador de filme fornecido, tendo deste modo completado todo o trabalho obrigatório assim como uma das sugestões do trabalho extra. Consideramos ainda a implementação das restantes sugestões de trabalho extra, mas não conseguimos implementá-las na sua totalidade antes do prazo de entrega. Como tal, decidimos excluir essas abordagens do nosso trabalho final.

Entre os conjuntos de entrada fornecidos conseguimos processar todos eles na recomendação por palavras-chave, no entanto para a recomendação por índice de similaridade, devido ao grande número de entradas da tabela criada e às limitações de memória do Google Colab não foi possível processar os conjuntos large4 e large5.

Este trabalho foi realizado no Google Cloud Project com o id cobalt-catalyst-231416 sendo que os BigQuery DataSets gerados possuem os mesmos nomes dos DataSets de input.

**Pré-Processamento de Dados**

A pesquisa por palavras-chave requereu a criação de uma Dataframecom entradas movieId, word, TF-IDF de modo a conseguirmos obter o grau de importância de uma palavra para um filme específico no contexto de todos os filmes fornecidos:

* Através de uma consulta SQL a uma tabela implementada pelo professor com as palavras associadas a cada filme, obtivemos uma tabela que representa o número de vezes em que uma palavra aparece associada a cada filme (tabela de frequências).
* Utilizando transformações da DataframeAPI obtivemos o número de filmes assim como a frequência da palavra com maior ocorrência para cada filme e, com base nesses resultados e na tabela anterior, criamos uma nova coluna na tabela que corresponde à frequência de um termo (TF).
* Calculamos o número de filmes em que cada palavra aparece (DF) e com base nesses valores e no número total de filmes criamos uma coluna na tabela com a frequência de documentos inversa (IDF), utilizando mais uma vez a Dataframe API.
* Por fim criamos uma nova coluna na tabela resultante do produto das colunas TF e IDF e aplicamos as restrições do valor mínimo de TF-IDF à tabela resultante.

A pesquisa por índice de similaridade requereu a criação de uma outra Dataframe com entradas movieId1, movieId2, JI de modo a conseguirmos obter o grau de similaridade entre todos os filmes de acordo com os utilizadores que os avaliaram:

* Através de consultas SQL obtivemos uma tabela com a união da informação distinta sobre o userId e movieId proveniente das tabelas ratings e tags.
* Utilizando a Dataframe API realizamos um join no userId da tabela anterior com ela própria de acordo com a restrição movieId1 < movieId2 de modo a obter apenas combinações únicas.
* De seguida fazemos um count agrupado por movieId1, movieId2 de forma a derivar um número de intersecções entre cada par de filmes.
* Para cada entrada da tabela resultante criamos duas novas colunas com o número de utilizadores que avaliaram o movieId1, movieId2.
* Por fim, criamos uma nova coluna com o índice de similaridade (Jaccard Index) que resulta da divisão do número de intersecções entre movieId1, moviedId2 pela soma do número de utilizadores que avaliaram o moviedId1 ou o movieId2.

**Carregamento Das Tabelas BigQuery**

Após o carregamento das tabelas pré-processadas para o Google Cloud Storage, foi necessário implementar uma Google Cloud Function que lesse os ficheiros armazenados e criasse tabelas BigQuery com base nos mesmos:

* À semelhança do método fornecido pelo professor para criar a tabela com os dados sobre os filmes, criamos dois métodos que exercem a mesma função para a tabela tfidf e a tabela do índice de similaridade, que apenas diferem no schema da tabela e nos tipos dos campos que a preenchem.
* De seguida criamos a Google Cloud Function que corre este código, sendo que esta é ativada através de um sistema *publish/subscribe*.

**Desenvolvimento Da Função de Busca**

Por fim, após a criação das tabelas BigQuery foi necessário implementar uma Google Cloud Function que permita fazer as consultas necessárias, de acordo com o operador dado, com o objetivo de retornar as recomendações esperadas:

* À semelhança do método fornecido pelo professor para listar a tabela com os dados sobre os filmes criamos dois métodos que exercem a mesma função para a tabela tfidf e a tabela do índice de similaridade que apenas diferem na tabela a qual a consulta é realizada.
* Para a pesquisa por palavras-chave criamos um novo método que processa as palavras fornecidas no formato correto e executamos uma consulta SQL a junção da tabela dos filmes com a tabela tfidf de modo a obtermos a lista de filmes de acordo com as palavras fornecidas por ordem decrescente de valor TF-IDF médio.
* Para a pesquisa por índice de similaridade executamos uma consulta SQL à tabela do índice de similaridade para obtermos um movieId de acordo com o movieId fornecido como argumento. Esse movieId é utilizado para recolher a informação sobre o filme que este representa na tabela dos filmes. Retornamos uma tabela por ordem decrescente de índice de similaridade.