Documentação experimento RecSys Challenge RL

Alex Echeverria
Heloisy Pereira

Introdução

O sistema de recomendação é um campo de pesquisa em constante evolução, que se dedica a fornecer sugestões personalizadas para os usuários, com base em seus interesses e comportamentos anteriores. Nesse sentido, a RecSys Challenge RL propõe a criação de um modelo de Deep Learning para geração de embeddings de itens, que será utilizado em um sistema de recomendação por similaridade. O objetivo é que seja utilizado o máximo de features e combinações possíveis na criação da representação do item (embedding), de modo a otimizar uma função de recomendação baseada em similaridade a partir dos embeddings criados.

Para o desafio, foi disponibilizado o conjunto de dados do Yelp, que contém informações sobre negócios, revisões, usuários, entre outros, em formato JSON. É necessário transformar o arquivo JSON em formato CSV, a fim de tornar o processo de criação de embeddings mais fácil. Além do formato CSV, também utilizamos o formato PARQUET.

Uma maneira de analisar as avaliações de restaurantes é por meio do uso de embeddings. Os embeddings são representações vetoriais densas de palavras, frases ou até mesmo usuários e itens, que são aprendidas por meio de técnicas de aprendizado de máquina.

No contexto de avaliações de restaurantes, podemos criar embeddings de usuários e embeddings de textos de reviews. Os embeddings de usuários representam a preferência de cada usuário por diferentes tipos de restaurantes, enquanto os embeddings de textos de reviews capturam as relações semânticas entre os termos utilizados nas avaliações.

Com essas representações vetoriais, podemos aplicar técnicas de aprendizado de máquina para realizar tarefas como classificação de sentimentos, recomendação de restaurantes e identificação de tópicos relevantes nas avaliações. Essa abordagem é útil para capturar relações não lineares entre usuários e restaurantes, que podem não ser facilmente identificáveis por meio de abordagens mais tradicionais.

Metodologia

Abordagem 1:

O primeiro notebook apresenta um processo de criação de embeddings de usuários para recomendação de negócios com base em suas avaliações. O processo é dividido em várias

etapas, que envolvem o carregamento de dados de avaliação, filtragem dos dados para obter os negócios relevantes, a criação de embeddings de usuários com base nas avaliações e a exportação dos embeddings para uso posterior em recomendações.

O código começa importando a biblioteca pandas e carregando os dados de avaliação de usuários e os conjuntos de dados do Yelp de revisão e usuário. Em seguida, ele filtra as avaliações relevantes para os negócios na lista de avaliações que devem ser recomendados e agrupa as avaliações por usuário e negócio.

Em seguida, ele define uma função que recebe uma lista e retorna os índices dos 5 maiores valores e aplica essa função à coluna de estrelas do dataframe de revisão para obter os usuários mais e menos bem avaliados para cada negócio. Em seguida, ele filtra os dados para obter apenas as avaliações relevantes para cada negócio.

Depois disso, o código define uma função que calcula os embeddings de usuário para cada negócio com base nas avaliações dos usuários e aplica essa função aos dados de revisão para criar os embeddings de usuário para cada negócio. Em seguida, ele exporta os embeddings de usuário para um arquivo parquet para uso posterior em recomendações.

No geral, o código realiza uma etapa importante na criação de um sistema de recomendação de negócios baseado em embeddings de usuário, e pode ser usado como ponto de partida para desenvolver sistemas de recomendação mais complexos e sofisticados.

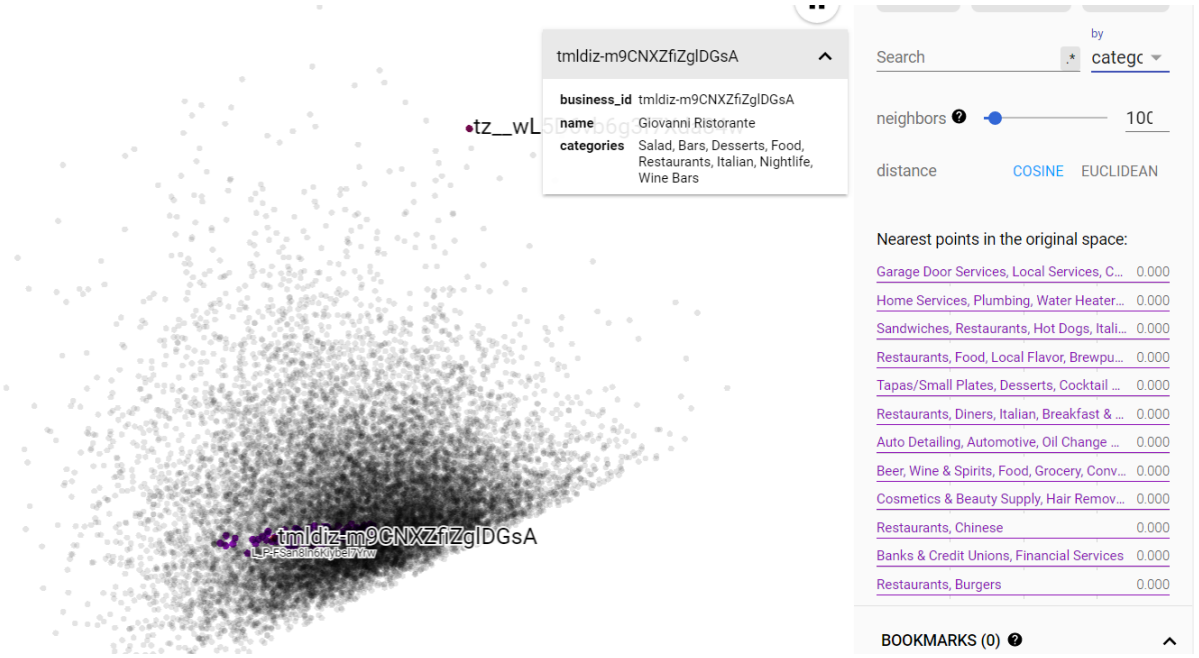
No segundo notebook, estão sendo carregados os dados de avaliação dos usuários, os dados de embeddings dos negócios e dos metadados relacionados a esses negócios. Em seguida, o notebook filtra os dados para conter apenas os negócios que possuem embeddings e os negócios de perfil dos usuários.

O notebook então une os dados de embeddings dos negócios com os metadados desses negócios, para que seja possível realizar a recomendação baseada nos embeddings. Para isso, são utilizadas técnicas de filtragem colaborativa, em que a similaridade dos embeddings é usada para encontrar os negócios mais relevantes para o usuário.

Em seguida, o notebook salva os embeddings dos negócios e os metadados em um arquivo CSV e calcula os resultados da recomendação utilizando a biblioteca NMSLIB. Por fim, o notebook exporta os resultados em um arquivo CSV.

É importante notar que este é apenas um dos notebooks que compõem o projeto completo, e que outros notebooks contêm etapas adicionais de pré-processamento, limpeza e validação dos dados.

A visualização do espaço de embeddings gerados pela abordagem 1 é:



Abordagem 2:

Para essa atividade, geramos um motor de recomendação de negócios baseado na similaridade entre as avaliações que um usuário faz e as avaliações que um negócio possui.

Nos baseamos na ideia que as avaliações de um negócio sintetizam ideias do que a pessoa pensa. Ao processarmos essas informações, conseguimos identificar implicitamente os gostos e particularidades das pessoas, assim, realizando a semelhança entre o tipo de avaliação das pessoas e uma possível nova recomendação.

O trabalho consiste em:

- Filtramos os negócios (business_id) que aparecem do dataset de avaliação (eval_users.csv);

- Filtramos o Review.json para conter apenas avaliações em que os business_id aparecem no eval_dataset;

- Agrupamos as avaliações pelo business_id;

- Para cada business_id, pegamos as 5 melhores avaliações;

- Geramos embedding utilizando o modelo de linguagem BERT de cada uma das 5 avaliações, resultando em um embedding de 768 valores;

- Calculamos a média dos 5 embeddings, gerando um embedding que representa aquela empresa;

- Filtramos o dataset de Reviews por usuários contido em eval_dataset;

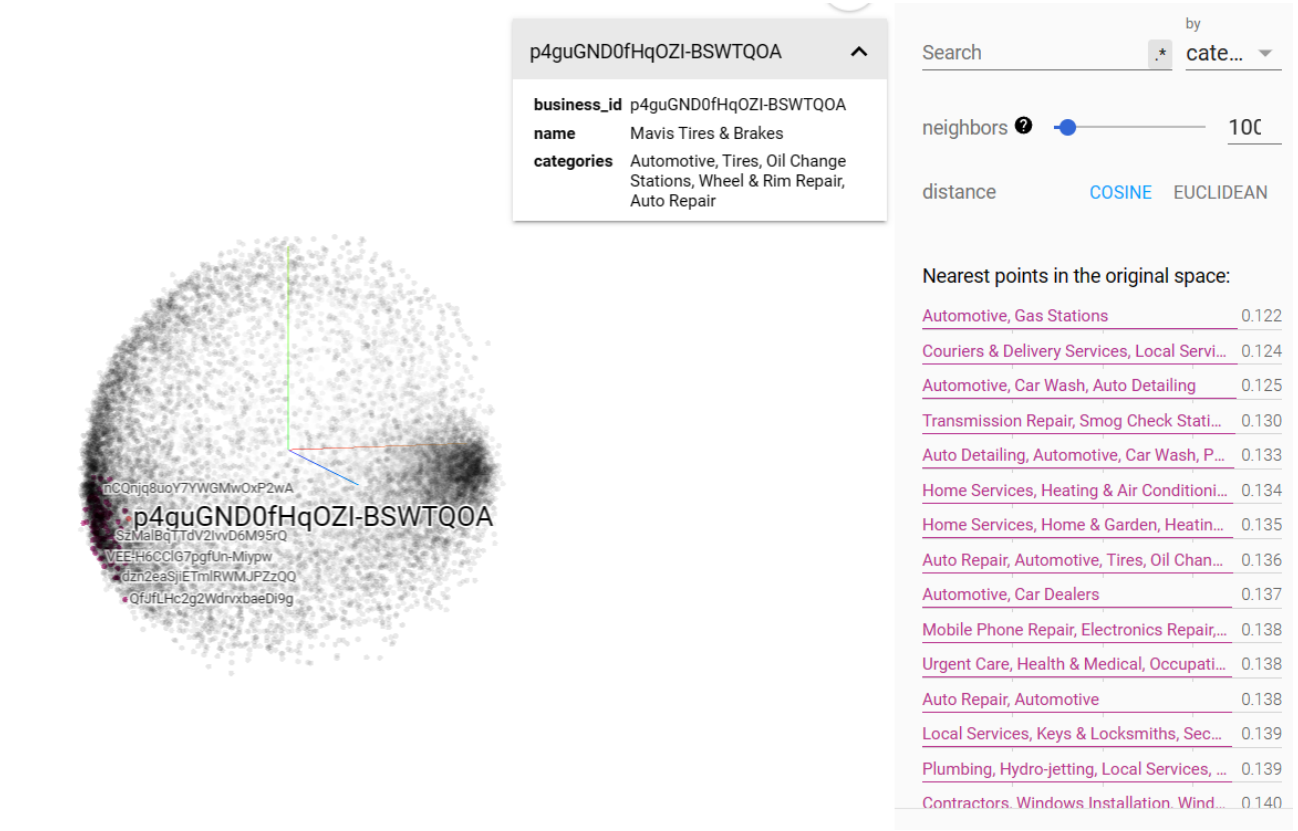
- Para cada usuário, pegamos as 5 melhores avaliações que ele já fez;

- O texto dessas avaliações é processado como já mencionado anteriormente, realizando a média dos embeddings retornados;

- De posse dos embeddings de cada user e dos negócios, basta substituirmos o embedding de user_perfil com o embedding particular do user_id do dataset de eval;

- Com isso, gera-se os embeddings necessários para a avaliação.

A visualização do espaço gerado pelos embeddings é:



Alterações feitas nos datasets

- df bussiness
  Colunas dropadas:
  - name: vários nomes podem se repetir. Apesar de não termos muitos nomes repetidos, temos alguns. Descartou-se para garantir que não tenhamos problemas com isso.
  - adress: preferimos dropar
  - city: foi dropado pois há muitas cidades, porém trabalharemos com raio de acordo com latitude/longitude.

- state: apesar de ter poucos estados, foi dropado pois trabalharemos com raio de acordo com latitude/longitude.
- postal_code: não julgamos necessário
- attributes: dropamos pois 77.35% dos atributos dos dicionários são nulos
- hours: preferimos dropar

- df review
  Colunas dropadas:
  - review_id: preferimos dropar, é apenas um identificador
  - date: preferimos dropar

  Colunas alteradas (escala logarítmica, depois transformado para int32):
  - useful
  - funny
  - cool

- df user:
  Colunas dropadas:
  - name: preferimos dropar
  - yelping_since: usamos numa engenharia de features, depois dropamos
  - friends: preferimos dropar
  - elite: preferimos dropar
  - average_stars: usamos numa engenharia de features, depois dropamos

  Colunas alteradas (escala logarítimica e/ou transformando tempo em dias e/ou criando feature e/ou passando para float32):
  - review_count
  - account_age
  - useful
  - funny
  - cool
  - fans
  - chato
  - compliment_hot
  - compliment_more
  - compliment_profile
  - compliment_cute
  - compliment_list
  - compliment_note
  - compliment_plain
  - compliment_cool
  - compliment_funny
  - compliment_writer
  - compliment_photos

Resultados:

Vale ressaltar que o resultado final no código de avaliação da abordagem 1 conseguiu bater o baseline, porém não se saiu melhor que a outra abordagem testada. Tendo como NDCG@5: 0.5338634141138902 e NDCG@10: 0.5854002296262076. Diante disso, não utilizamos os embeddings e os metadados dessa abordagem.

A segunda abordagem conseguiu bater bem no baseline!

Segundo a métrica de avaliação NDGC@5 e NDCG@10, os valores que obtivemos com essa abordagem foram de 0.55 e 0.59, respectivamente.

Os baselines de recomendação aleatória ou BERT embedding usando a feature ‘categories’ é de apenas (0.45, 0.51) e (0.50, 0.56), respectivamente.

Próximas iterações do trabalho consistem em testar novas formas de interpretar o texto, inserir as piores avaliações também em conta e usar mais features também, para a geração de embeddings.