PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Inteligência Artificial e Aprendizado de Máquina

Frederico Comério

Conversão de Linguagem Natural em Operações de Análise de Dados com apoio de Modelos de Inteligência Artificial Generativa

Belo Horizonte
Agosto de 2024

Frederico Comério

Conversão de Linguagem Natural em Operações de Análise de Dados com apoio de Modelos de Inteligência Artificial Generativa

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Inteligência Artificial e Aprendizado de Máquina, como requisito parcial à obtenção do título de *Especialista*.

Belo Horizonte
Agosto de 2024

SUMÁRIO

1. Introdução

A Inteligência Artificial (IA) e o Aprendizado de Máquina (ML) têm transformado significativamente a maneira como interagimos com dados e informações. Em particular, os Large Language Models (LLMs) têm demonstrado um potencial sem precedentes na compreensão e geração de linguagem natural, criando oportunidades para o desenvolvimento de interfaces mais intuitivas em diversas áreas. Este trabalho explora a sinergia entre LLMs e bibliotecas de análise de dados, como o Pandas, com o objetivo de tornar a análise de dados acessível a um público mais amplo, inclusive àqueles sem conhecimentos técnicos específicos.

Além disso, a crescente popularização de interfaces baseadas em linguagem natural ressalta a importância de ampliar o acesso à análise de dados, permitindo que indivíduos sem formação técnica possam extrair insights valiosos em seus respectivos contextos. Ao integrar modelos LLM com ferramentas robustas como Pandas, este trabalho visa superar as barreiras técnicas tradicionalmente presentes na ciência de dados, democratizando o acesso a processos analíticos complexos. Esse avanço não só simplifica a interação com dados, como também abre novas possibilidades para a tomada de decisão informada em áreas como negócios, saúde e pesquisa acadêmica.

2. Descrição do Problema e da Solução Proposta

A análise de dados tradicionalmente exige conhecimento em linguagens de programação e bibliotecas específicas, criando uma barreira para usuários não técnicos. Essa limitação impede que muitos se beneficiem do poder da análise de dados para extrair insights valiosos em suas áreas de negócio.
Este projeto visa solucionar esse problema desenvolvendo uma interface de usuário baseada em chat que permite aos usuários realizar análises de dados complexas usando linguagem natural. Através da integração de um modelo LLM (como o GPT-4) com a biblioteca Pandas e PandasAI, a interface traduz as perguntas dos usuários em código executável, retornando os resultados de forma clara, concisa e amigável.

Objetivos do trabalho:
• Desenvolver uma interface web que permita o upload de datasets e interação com os dados carregados a partir de uma área de chat.
• Implementar um sistema de chat onde o usuário possa realizar perguntas sobre os dados carregados em linguagem natural.
• Utilizar um modelo de IA Generativa (LLM) capaz para converter essas perguntas em comandos pandas, realizando as operações de maneira assertiva e trazendo o resultado das operações de forma amigável para o usuário final de forma que mesmo um usuário sem o conhecimento técnico em programação, ciência de dados possa desfrutar do resultado das análises.

2.1 Ferramentas e Tecnologias Utilizadas

As atividades técnicas foram realizadas com apoio das seguintes tecnologias, ferramentas e bibliotecas:

Python (versão 3.10): Python foi escolhido pela sua facilidade de uso, vasta comunidade e riqueza de bibliotecas para ciência de dados e inteligência artificial. Sua sintaxe simples facilita o desenvolvimento rápido e permite que a aplicação seja acessível até mesmo para iniciantes.

Streamlit: O Streamlit foi adotado por permitir a criação rápida de interfaces web interativas sem a necessidade de conhecimentos profundos em desenvolvimento web. Isso é crucial para transformar um código de análise de dados em uma ferramenta acessível para usuários não técnicos, permitindo a interação direta com os dados.

PandasAI: O PandasAI foi utilizado por sua capacidade de integrar modelos de linguagem natural (LLMs) ao ambiente de manipulação de dados. Isso permite que consultas em linguagem natural sejam traduzidas em operações analíticas complexas no DataFrame, alinhando-se com o objetivo do trabalho de facilitar a análise de dados para usuários sem conhecimento técnico.

Pandas: Pandas é a base da manipulação de dados no projeto, fornecendo funcionalidades poderosas para leitura, transformação e visualização de dados. Sua integração com outras bibliotecas e sua ampla adoção no campo da análise de dados tornaram-no indispensável para o processamento e exibição dos datasets carregados na aplicação.

Visual Studio Code: O Visual Studio Code (VS Code) foi escolhido como ambiente de desenvolvimento integrado (IDE) por sua leveza, extensibilidade e suporte a diversas linguagens de programação. Ele é ideal para o desenvolvimento de projetos Python devido à sua vasta coleção de plugins específicos, como o suporte ao

Python e Jupyter Notebooks, que aceleram a produtividade e oferecem uma experiência rica de desenvolvimento.

Anaconda versão 23.1.0: O Anaconda foi adotado para gerenciamento de ambientes e pacotes, oferecendo uma solução robusta para gerenciar dependências complexas e garantir a reprodutibilidade do projeto. Com a versão 23.1.0, é possível criar ambientes Python específicos e instalar as bibliotecas necessárias de forma isolada, evitando conflitos de versões e facilitando a configuração do projeto tanto para desenvolvimento quanto para distribuição.

3. Coleta de Dados

Para demonstrar a funcionalidade da interface, utilizaremos o conjunto de dados "IMDB 5000 Movie Dataset" disponível no Kaggle. Este dataset contém informações sobre filmes, como título, gênero, diretor, orçamento, receita e avaliações.

Nome do dataset: IMDB 5000 Movie Dataset **Descrição: Conjunto de dados contendo informações sobre filmes do IMDB** **Link:** https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset		
Nome do Atributo	**Descrição**	**Tipo**
color	Indica se o filme é colorido ou preto e branco.	Texto
director_name	Nome do diretor do filme.	Texto
num_critic_for_reviews	Número de críticas feitas por críticos profissionais.	Numérico
duration	Duração do filme em minutos.	Numérico
director_facebook_likes	Curtidas na página do diretor no Facebook.	Numérico
actor_3_facebook_likes	Curtidas no Facebook do terceiro ator principal.	Numérico
actor_2_name	Nome do segundo ator principal.	Texto
actor_1_facebook_likes	Curtidas no Facebook do ator principal.	Numérico
gross	Receita bruta gerada pelo filme.	Numérico
genres	Gêneros aos quais o filme pertence.	Texto
actor_1_name	Nome do ator principal.	Texto
movie_title	Título do filme.	Texto
num_voted_users	Número de votos recebidos no IMDb.	Numérico
cast_total_facebook_likes	Total de curtidas no Facebook para todo o elenco.	Numérico

actor_3_name	Nome do terceiro ator principal.	Texto
facenumber_in_poster	Número de rostos visíveis no pôster do filme.	Numérico
plot_keywords	Palavras-chave que descrevem o enredo do filme.	Texto
movie_imdb_link	Link para o filme no IMDb.	Texto
num_user_for_reviews	Número de críticas feitas por usuários.	Numérico
language	Idioma do filme.	Texto
country	País de origem do filme.	Texto
content_rating	Classificação indicativa do filme.	Texto
budget	Orçamento do filme.	Numérico
title_year	Ano de lançamento do filme.	Numérico
actor_2_facebook_likes	Curtidas no Facebook do segundo ator principal.	Numérico
imdb_score	Pontuação do filme no IMDb.	Numérico
aspect_ratio	Proporção da tela do filme.	Numérico
movie_facebook_likes	Curtidas no Facebook da página do filme.	Numérico

Tabela 1: Fonte de dados IMDB 5000 - Conjunto de dados contendo informações sobre filmes do IMDB

4. Processamento/Tratamento de Dados

O processamento de dados será realizado de acordo com as etapas abaixo:

- O arquivo CSV será carregado em um DataFrame Pandas.
- Colunas irrelevantes para a demonstração, como links de mídia social, serão removidas.
- Dados faltantes serão tratados, seja através da remoção de linhas com dados ausentes ou imputação de valores.
- Os tipos de dados de cada coluna serão verificados e corrigidos se necessário.

```python
import pandas as pd

# Carregar o dataset
df = pd.read_csv('movie_metadata.csv')

# Remover colunas irrelevantes
df = df.drop(['movie_imdb_link', 'actor_1_facebook_likes',
              'actor_2_facebook_likes', 'actor_3_facebook_likes',
              'cast_total_facebook_likes', 'movie_facebook_likes'], axis=1)

# Tratar dados faltantes (exemplo: remover linhas com valores ausentes em 'gross')
df.dropna(subset=['gross'], inplace=True)
```

Figura 1: Limpeza de Dados com Python

5. Análise e Exploração dos Dados

Antes de prosseguir com a criação da interface, é recomendável explorar o conjunto de dados para entender suas características e identificar padrões relevantes. A análise exploratória de dados (EDA) guiará o desenvolvimento da interface e ajudará na formulação de perguntas relevantes para o modelo.

Técnicas de EDA:

- Estatísticas descritivas (média, mediana, desvio padrão) para variáveis numéricas.
- Distribuição de frequência para variáveis categóricas.
- Gráficos de dispersão, histogramas e boxplots para visualizar relações entre variáveis.

```
17  import matplotlib.pyplot as plt
18  import seaborn as sns
19
20  # Histograma da Pontuação IMDB
21  plt.figure(figsize=(10, 6))
22  sns.histplot(df['Pontuação IMDB'], kde=True)
23  plt.title('Distribuição da Pontuação IMDB')
24  plt.xlabel('Pontuação IMDB')
25  plt.ylabel('Frequência')
26  plt.show()
27
28  # Gráfico de dispersão entre Orçamento e Receita Bruta
29  plt.figure(figsize=(10, 6))
30  sns.scatterplot(x='budget', y='gross', data=df)
31  plt.title('Relação entre Orçamento e Receita Bruta')
32  plt.xlabel('Orçamento')
33  plt.ylabel('Receita Bruta')
34  plt.show()
```

Figura 2: Análise e exploração de dados com Pandas, Seaborn e MatPlotLib.

6. Preparação dos Dados para os Modelos de Aprendizado de Máquina

A preparação dos dados para o modelo LLM envolverá principalmente a criação de prompts eficazes que capturem a intenção do usuário e forneçam contexto suficiente para o modelo gerar o código Pandas correto.

Técnicas:

- Criação de templates de prompts: Definir estruturas de frases que guiam o usuário na formulação de perguntas compreensíveis para o modelo.
- Incorporação de informações contextuais: Incluir o nome do dataset, os nomes das colunas e os tipos de dados no prompt para o modelo.

A biblioteca PandasAI já traz consigo um prompt template adequado para o objetivo de realizar análise de dados em datasets no formato pandas, que foi utilizado como prompt template padrão para realizar todas as análises:

```
Analyze the data, using the provided dataframes (`dfs`).
1. Prepare: Preprocessing and cleaning data if necessary
2. Process: Manipulating data for analysis (grouping, filtering, aggregating, etc.)
3. Analyze: Conducting the actual analysis (if the user asks to plot a chart you must save it as an image in temp_chart.png and not show the chart.)
If the user requests to create a chart, utilize the Python matplotlib library to generate high-quality graphics that will be saved directly to a file.
At the end, return a dictionary of:
- type (possible values "string", "number", "dataframe", "plot")
- value (can be a string, a dataframe or the path of the plot, NOT a dictionary)
Examples:
    { "type": "string", "value": f"The highest salary is {highest_salary}." }
    or
    { "type": "number", "value": 125 }
    or
    { "type": "dataframe", "value": pd.DataFrame({...}) }
    or
    { "type": "plot", "value": "temp_chart.png" }
"""
```

Figura 3: Prompt Template (System Prompt) padrão da Biblioteca PandasAI

7. Aplicação de Modelos de Aprendizado de Máquina

O modelo LLM, como o GPT-4, será o principal componente de aprendizado de máquina deste projeto. Utilizaremos a API do OpenAI para acessar e interagir com o modelo.

```python
import openai
import pandas as pd

# Configurar a API Key do OpenAI
openai.api_key = "MINHA_API_KEY_DA_OPENAI"

def gerar_codigo_pandas(pergunta):
    prompt = f"""
    Dataset: IMDB Movie Dataset
    Tarefa: {pergunta}
    Colunas:
    - 'Título do Filme': string
    - 'Ano de Lançamento': inteiro
    - 'Gêneros': string (múltiplos gêneros separados por '|')
    """
    resposta = openai.Completion.create(
        engine="gpt-4o",
        prompt=prompt,
        max_tokens=500,
        temperature=0.1,
    )
    return resposta.choices[0].text.strip()

# Exemplo de uso
pergunta_usuario = "Quantos filmes de ação foram lançados em 1985?"
codigo_pandas = gerar_codigo_pandas(pergunta_usuario)

# Executar o código Pandas
resultado = eval(codigo_pandas)
print(resultado)
```

Figura 4: Testes de integração com a Biblioteca OpenAI para o linguagem de programação Python

8. Avaliação dos Modelos de Aprendizado de Máquina e Discussão dos Resultados

Neste capítulo, será apresentada a avaliação da eficácia do modelo de aprendizado de máquina utilizado no projeto, além de uma discussão dos resultados obtidos.

8.1 Metodologia de Avaliação

Para avaliar o desempenho do modelo de linguagem natural (LLM) integrado ao projeto, foi criado um caderno de testes contendo 10 perguntas extremamente específicas, desenhadas para exigir tanto o entendimento profundo de contexto quanto habilidades avançadas de filtragem e operações sobre o dataset. As perguntas foram formuladas para explorar diferentes nuances da análise de dados e verificar a capacidade do modelo em realizar operações complexas com precisão.

Um exemplo de pergunta presente no caderno de testes é: "Qual foi o filme de maior orçamento do diretor James Cameron?". Esta questão exige que o modelo compreenda o contexto envolvendo a relação entre "diretor" e "orçamento", além de realizar operações de filtragem e ordenação sobre o dataset para encontrar a resposta correta.

Figura 5: Protótipo respondendo pergunta de usuário já com interface gráfica implementada.

```
Code running:
```
def analyze_data(dfs: list[pd.DataFrame]) ->dict:
    df = dfs[0]
    df['budget'] = df['budget'].astype(float)
    james_cameron_movies = df[df['director_name'] == 'James Cameron']
    highest_budget_movie = james_cameron_movies[james_cameron_movies['budget'] == james_cameron_movies['budget'].max()]
    highest_budget_movie_title = highest_budget_movie['movie_title'].values[0]
    return {'type': 'string', 'value': f'The movie with the highest budget directed by James Cameron is {highest_budget_
movie_title}.'}
```
2024-08-26 16:02:00.544 Answer: {'type': 'string', 'value': 'The movie with the highest budget directed by James Cameron
 is Avatar\xa0.'}
2024-08-26 16:02:00.547 Executed in: 10.932517051696777s
The movie with the highest budget directed by James Cameron is Avatar .
```

Figura 6: Prompt do usuário sendo traduzido em comando na biblioteca PANDAS pelo backend.

8.2 Resultados Obtidos

O modelo utilizado, o GPT-4 da OpenAI, apresentou um desempenho exemplar, respondendo corretamente a todas as perguntas do caderno de testes, atingindo um score de 100% de precisão. Este resultado demonstra que o modelo não só foi capaz de entender o contexto das perguntas como também executou operações complexas de forma eficaz sobre os dados fornecidos.

Os testes envolveram uma variedade de desafios, desde a identificação de padrões até a extração e combinação de informações específicas em múltiplas colunas, destacando a robustez e flexibilidade do GPT-4 para aplicações em análise de dados.

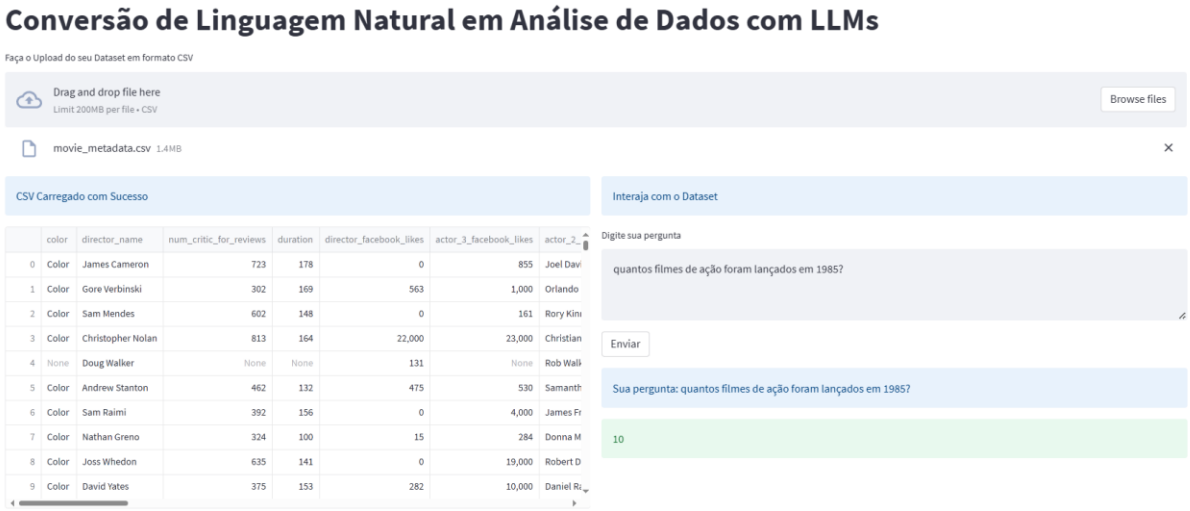

Figura 7: Outro exemplo de interação do usuário final com a interface

```
    """
    action_movies_1985 = dfs[0][(dfs[0]['title_year'] == 1985) & dfs[0]['genres'].str.contains('Action')]
    num_action_movies_1985 = action_movies_1985.shape[0]
    return {'type': 'number', 'value': num_action_movies_1985}
    ```
2024-08-26 16:20:23.966 Answer: {'type': 'number', 'value': 10}
2024-08-26 16:20:23.967 Executed in: 0.035000324249267585
10
```

**Figura 6:** Prompt do usuário sendo traduzido em comando na biblioteca PANDAS pelo backend.

O caderno de testes completo, contendo todas as perguntas e respostas geradas durante a avaliação, está disponível na tabela abaixo e também se encontra no repositório Github deste trabalho (vide capítulo 10 – Links). Este recurso é importante para garantir a transparência do processo de avaliação e para permitir que outros pesquisadores ou desenvolvedores possam realizar suas próprias análises:
```

Pergunta	Resposta Esperada	Resposta Encontrada pelo Modelo	Resultado
Qual o filme de maior receita (gross) do diretor Steven Spielberg?	E.T. the Extra-Terrestrial	E.T. the Extra-Terrestrial	Certo
Quantos filmes de comédia foram lançados em 1995?	25	25	Certo
Qual é a duração média dos filmes de Christopher Nolan?	140.25	140.25	Certo
Qual o filme de maior orçamento com a classificação indicativa R (Restricted)?	The Host	The Host	Certo
Qual o filme com maior pontuação IMDb no gênero Drama?	The Shawshank Redemption	The Shawshank Redemption	Certo
Qual o ano com o maior número de filmes lançados?	2009	2009	Certo
Qual o país com maior número de filmes no gênero Ação?	Estados Unidos	Estados Unidos	Certo
Qual o filme de maior duração dirigido por Quentin Tarantino?	The Hateful Eight	The Hateful Eight	Certo
Qual o orçamento médio dos filmes de ficção científica lançados nos anos 2000?	$131,782,068.12	$131,782,068.12	Certo
Quantos filmes lançados em 2010 têm uma pontuação IMDb acima de 8.0?	6	6	Certo

Tabela 2: Caderno de Testes para avaliação da acurácia do modelo

9. Conclusão

Este trabalho evidenciou o grande potencial da combinação de modelos de linguagem natural (LLMs) com bibliotecas de análise de dados para o desenvolvimento de interfaces de usuário intuitivas e acessíveis. A interface de chat criada permite que usuários sem conhecimento técnico realizem análises em datasets complexos utilizando linguagem natural, democratizando o acesso a insights valiosos e promovendo a inclusão de novos públicos no processo de tomada de decisões baseada em dados.

No entanto, algumas limitações ainda precisam ser consideradas. A precisão da interface está diretamente ligada à capacidade do LLM de interpretar corretamente as intenções dos usuários e gerar código Pandas apropriado. Atualmente, esses modelos ainda podem apresentar falhas, especialmente diante de perguntas complexas ou formuladas de maneira ambígua.

Para trabalhos futuros, é recomendada a exploração de integrações com outras bibliotecas de visualização de dados, visando a apresentação dos resultados de forma mais intuitiva e interativa. Além disso, é importante investigar métodos para aumentar a resiliência do modelo frente a erros gramaticais e linguagem informal. Outra frente de aprimoramento é a incorporação de mecanismos de feedback contínuo por parte dos usuários, permitindo ajustes e melhorias progressivas no sistema, garantindo uma experiência cada vez mais eficaz e precisa.

10. Links

Link para o repositório deste trabalho:
https://github.com/fredcobain/tcc_pos_ia_pucmg

11. Referências

DataCamp. Pandas Tutorial: DataFrames in Python. 2023. Disponível em: <https://www.datacamp.com/tutorial/pandas-tutorial-dataframe-python/>. Acesso em: 05 ago. 2024.

Kaggle. IMDB 5000 Movie Dataset. 2017. Disponível em: <https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset/> Acesso em: 12 ago. 2024.

Python Software Foundation. Python 3.10 Documentation. 2021. Disponível em: <https://docs.python.org/3.10/>. Acesso em: 18 jul. 2024.

Streamlit Inc. Streamlit Documentation. 2023. Disponível em: <https://docs.streamlit.io/>. Acesso em: 20 jul. 2024.

PandasAI Documentation. 2023. Disponível em: <https://github.com/gventuri/pandas-ai/>. Acesso em: 22 jul. 2024.

The Pandas Development Team. Pandas Documentation. 2023. Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 24 jul. 2024.

Microsoft. Visual Studio Code Documentation. 2023. Disponível em: <https://code.visualstudio.com/docs/>. Acesso em: 26 jul. 2024.

Anaconda Inc. Anaconda Documentation. 2023. Disponível em: <https://docs.anaconda.com/>. Acesso em: 28 jul. 2024.

PYTHON. Disponível em: <https://www.python.org/>. Acesso em: 04 de jul. 2024