Andrei dos Santos, Henrique Geribello Giabardo, Lucas Lopes de Paula Junior

Análise de Sentimento e Dashboard de preços de Produtos Obtidos por Web Scraping

São Paulo, SP

2023

Andrei dos Santos, Henrique Geribello Giabardo, Lucas Lopes de Paula Junior

Análise de Sentimento e Dashboard de preços de Produtos Obtidos por Web Scraping

Trabalho de conclusão de curso apresentado ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Orientador: Prof. Dr. Jorge Luís Risco Becerra

São Paulo, SP

2023

Sumário

1 Introdução

A extração de dados na internet é uma prática cada vez mais comum em diversos setores da sociedade, desde empresas que desejam monitorar seus concorrentes até pesquisadores que precisam coletar informações para seus estudos. O processo de coleta de dados em sites na web é conhecido como web scraping, que consiste na extração automatizada de informações em páginas da web por meio de software específico. No âmbito acadêmico, o tema de web scraping tem se mostrado relevante, tanto para estudantes de Engenharia de Computação, Ciência da Computação ou Sistemas de Informação (dentre outros cursos relacionados à área de tecnologia da informação) quanto para aqueles que desejam entender melhor as possibilidades oferecidas pela tecnologia. Em um trabalho de conclusão de curso, a análise do processo de web scraping pode ser uma oportunidade para compreender as principais tecnologias, técnicas e desafios envolvidos nesse processo.

1.1 Motivação

Vivemos em um cenário de Big Data e Machine Learning, onde a disponibilidade crescente de dados sobre produtos, demanda e feedback dos usuários na web oferece oportunidades inexploradas. No entanto, esses dados estão frequentemente dispersos, desorganizados e de difícil acesso, limitando seu potencial de criação de valor. Nesse contexto, a prática de web scraping se destaca como uma ferramenta essencial para transformar esses dados em informação e inteligência.

A busca por produtos em sites de e-commerce é uma tarefa laboriosa, e a necessidade de acompanhar a concorrência, os preços gerais dos produtos e a demanda é crucial para qualquer comerciante. No entanto, a falta de tecnologia acessível para esse acompanhamento deixa os competidores menores em desvantagem. Nosso projeto surge como uma resposta a essa discrepância, buscando equilibrar o jogo ao disponibilizar informações valiosas e permitir que mais negócios tomem decisões inteligentes com base em dados

1.2 Objetivos

O objetivo principal deste trabalho é proporcionar às empresas meios eficazes para avaliar estrategicamente seus produtos e os de seus concorrentes. Para isso, propomos a análise de preços e sentimentos dos consumidores. Através do acompanhamento constante, pretendemos oferecer insights sobre a dinâmica de demanda, oferta e feedback do comprador.

Disponibilizaremos essas informações em dashboards, não apenas sugerindo tomadas de decisão, mas também capacitando os clientes a criar suas estratégias de negócio com base em dados confiáveis

Assim, o sistema caracteriza-se por 3 partes principais: - Um Web Scrapper que reúne informação (preço e comentários dos produtos alvo) - Um sistema de analise de sentimento relacionado aos comentários - Um Dashboard que apresenta os dados coletados e os Insights gerados

A construção dessas 3 partes se fez pelo grupo, utilizando frameworks e bibliotecas disponíveis na internet, e algumas criações e integrações próprias.

Os benefícios principais para nossos usuários é um acompanhamento verídico dos preços dos produtos alvos, para entendimento de como está o estado atual do mercado e a concorrência. Esses dados permitem Insights, Inteligencia de Negócio e tomadas de decisões estratégicas.

Ao final do trabalho, espera-se que seja possível apresentar uma visão completa do processo de web scraping, desde a escolha das tecnologias até a aplicação prática em um projeto real. Além disso, será possível demonstrar a utilidade das informações coletadas e como elas podem ser utilizadas para aprimorar a tomada de decisão em diferentes setores da sociedade.

1.3 Justificativa

O cenário atual revela uma lacuna significativa entre grandes players e empresas menores no que diz respeito ao acompanhamento de dados.

Empresas robustas têm recursos e know-how para contratar os melhores desenvolvedores, enquanto concorrentes menores muitas vezes carecem dessa infraestrutura técnica. Nosso projeto não apenas equilibra essa disparidade, mas também preenche uma lacuna tecnológica, oferecendo uma solução acessível e eficaz para o acompanhamento estratégico de dados, vital para a tomada de decisões informadas.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte forma: Este trabalho está estruturado em capítulos que visam guiar o leitor através do desenvolvimento e resultados alcançados. A seguir, detalhamos a organização de cada capítulo:

Capítulo 1 - Introdução Este capítulo estabelece o contexto e a base para o desenvolvimento do trabalho. Inicia-se com a *Motivação*, apresentando o estado da arte resumido com base em trabalhos consultados, citando referências relevantes.

Em seguida, destacamos o *Objetivo*, oferecendo uma visão clara do propósito do trabalho. A *Justificativa* destaca a importância do projeto para a sociedade e comparações com trabalhos relevantes. A seção de *Organização do Trabalho* antecipa o leitor quanto à estrutura do documento.

Capítulo 2 - Aspectos Conceituais Este capítulo explora conceitos fundamentais relacionados ao desenvolvimento do trabalho. Com base em trabalhos consultados, fornecemos uma revisão bibliográfica detalhada sobre web scraping, análise de sentimentos e tecnologias relevantes.

Capítulo 3 - Metodologia do Trabalho Neste capítulo, detalhamos o processo de desenvolvimento do trabalho, abordando suas fases e especificidades. Citações de trabalhos consultados são incorporadas para embasar as escolhas metodológicas.

Capítulo 4 - Especificação de Requisitos Este capítulo define e descreve os requisitos do trabalho, estabelecendo as bases para o desenvolvimento subsequente. As escolhas metodológicas e citações de trabalhos consultados são incorporadas para fundamentar as decisões.

Capítulo 5 - Desenvolvimento do Trabalho Este capítulo descreve a transformação dos requisitos em produtos, abordando tecnologias utilizadas, projeto e implementação, bem como testes e avaliação. As seções seguem uma organização definida em conjunto com o orientador, destacando as decisões e justificativas.

Capítulo 6 - Considerações Finais Este capítulo engloba as conclusões do projeto de formatura, apresentando um balanço dos resultados atingidos e não atingidos, com justificativas. As contribuições do trabalho são destacadas, ressaltando o que foi efetivamente da autoria da equipe. Além disso, as perspectivas de continuidade são discutidas, apontando possíveis trabalhos futuros.

Ao final deste trabalho, espera-se que o leitor tenha uma compreensão abrangente do desenvolvimento do projeto, desde a fundamentação teórica até a aplicação prática e os resultados alcançados.

Esperamos que a estrutura delineada proporcione uma compreensão abrangente do desenvolvimento do projeto, guiando o leitor de forma lógica e coerente pelos aspectos teóricos, metodológicos e práticos deste trabalho de conclusão de curso.

2 Aspectos Conceituais

2.1 Web Scraping: tecnologias, técnicas e problemáticas

O web scraping é uma técnica utilizada para extrair dados de sites da web de forma automatizada.

Para que essa técnica funcione de maneira eficiente, é necessário que sejam utilizadas tecnologias específicas e técnicas de programação adequadas, além de lidar com diversas problemáticas que podem surgir durante o processo.

Uma das tecnologias fundamentais para o web scraping é a realização de requisições HTTP, que permite que o programa obtenha o conteúdo da página desejada. Além disso, a linguagem de programação Python 3 é amplamente utilizada para o desenvolvimento de programas de scraping, principalmente pela facilidade de manipulação de dados. Por outro lado, a utilização de JavaScript é necessária em casos de páginas que possuam uma renderização dinâmica. Existem diversos frameworks e bibliotecas disponíveis para a realização do web scraping, como o Scrapy, que é um framework em Python para coleta de dados em larga escala. Outras bibliotecas amplamente utilizadas são o Selenium, para automação de testes em navegadores web, Requests, para realização de requisições HTTP em Python, Beautiful Soup 4, para fazer o parsing de HTML e XML, e Playwright e Puppeteer, que permitem a renderização dinâmica por navegador, headless ou não.

Entre as técnicas utilizadas no web scraping, utilizaremos a raspagem e parser HTML, que consistem em extrair e analisar as informações de uma página web. A renderização dinâmica por navegador também é uma técnica importante, uma vez que permite a obtenção de informações que só podem ser acessadas após a execução de scripts JavaScript, como scroll e acionamento de botões.

Contudo, o web scraping pode enfrentar diversas problemáticas durante o processo, como os bloqueios de IP, que podem ocorrer quando o site identifica que está sendo alvo de scraping e bloqueia o acesso do programa. Nesse sentido, é necessário utilizar técnicas como a rotação de proxies para evitar que isso aconteça. Outra questão importante é a utilização de user-agents, que ajudam a disfarçar a origem das requisições. Além disso, a utilização de seletores adequados para a extração de informações também é fundamental.

Em suma, o web scraping envolve uma série de tecnologias e técnicas que devem ser aplicadas de forma adequada para a obtenção eficiente de dados. Entretanto, o processo pode enfrentar diversas problemáticas, que precisam ser contornadas para que a coleta de informações seja bem-sucedida, e é o que visamos aprofundar no estudo e elaboração deste trabalho de conclusão de curso.

2.2 ETL: Extract, Transform and Load

A coleta de dados é uma tarefa fundamental em qualquer projeto de análise de dados. Através do web scraping, é possível extrair informações de diversas fontes, como sites e APIs, e transformá-las em dados úteis para análises e tomadas de decisão. No entanto, após a coleta de dados, é necessário garantir que esses dados sejam corretamente armazenados e estruturados para que possam ser facilmente consumidos e utilizados em análises posteriores. Nesse sentido, os pipelines de dados são uma solução eficaz para gerenciar e automatizar esse processo.

Os pipelines de dados são compostos por um conjunto de ferramentas e tecnologias que permitem a coleta, armazenamento, transformação e disponibilização de dados de maneira eficiente e segura.

Para garantir escalabilidade, segurança, disponibilidade e custos eficientes, utilizaremos uma provedora de Cloud Computing para hospedar a pipeline de dados desenvolvida neste trabalho. As principais provedoras de cloud utilizadas para a criação de pipelines de dados são a AWS (Amazon Web Services), GCP (Google Cloud Platform) e Azure (Microsoft Azure), que oferecem serviços e soluções para cada uma dessas etapas do processo.

Para a coleta de dados, é possível utilizar serviços como o AWS Lambda, GCP Cloud Functions e Azure Functions, que permitem a execução de códigos em resposta a eventos específicos, como a chegada de novos dados em uma determinada fonte. Isso torna o processo de coleta de dados mais automatizado e escalável, reduzindo a necessidade de intervenção manual.

Após a coleta de dados, é necessário estruturá-los e transformá-los em um formato adequado para análises. As principais ferramentas para essa etapa são o AWS Glue, GCP Dataflow e Azure Data Factory, que permitem a realização de processos ETL (Extração, Transformação e Carga) de forma automatizada e escalável. Essas ferramentas permitem que os dados sejam transformados em um formato padronizado e consistente, tornando-os mais fáceis de analisar. Já no armazenamento, há diferentes opções de banco de dados disponíveis em cada provedor de nuvem, tais como o Amazon RDS, o Google Cloud SQL e o Azure SQL. Após o armazenamento dos dados, é importante garantir que os mesmos estejam prontos para serem consumidos. Neste sentido, é preciso estruturar os dados de forma adequada, de acordo com o objetivo do projeto. Isso pode envolver a limpeza dos dados, a transformação de formatos, a agregação de informações, entre outras atividades.

Para tal, existem diversas ferramentas de ETL (Extract, Transform and Load) disponíveis no mercado. Entre elas, destacam-se o Apache NiFi, o Talend, o Apache Airflow e o AWS Glue.

Uma vez estruturados e preparados para uso, os dados podem ser consumidos por

meio de diferentes ferramentas de visualização e análise. Nesse contexto, é possível criar dashboards interativos e relatórios personalizados, que permitem a extração de insights valiosos para os negócios.

Entre as ferramentas de análise de dados mais utilizadas, destacam-se o Power BI, o Tableau e o Google Data Studio.

Em resumo, a criação de um fluxo ETL completo para o projeto de web scraping envolve diferentes etapas e ferramentas. É necessário selecionar os melhores serviços de nuvem, ferramentas de ETL e de visualização de dados para garantir que o pipeline de dados funcione adequadamente e que os resultados sejam alcançados com eficiência. Além disso, é preciso estar atento às problemáticas envolvidas no processo, como a segurança dos dados, o gerenciamento de recursos, o controle de qualidade e a escalabilidade do sistema.

2.3 Governança e Monetização de dados

A governança e a monetização de dados são temas cada vez mais relevantes na era da transformação digital e do grande volume de informações gerado pela internet. Quando se trata de web scraping, esses temas ganham ainda mais importância, uma vez que a extração de dados pode ser utilizada para diversos fins, desde análises de mercado até a criação de modelos de negócios baseados em dados.

A governança de dados refere-se à gestão e ao controle dos dados gerados por uma organização, garantindo a qualidade, a segurança, a conformidade e a utilização ética das informações. No contexto de web scraping, a governança de dados envolve questões como a escolha das fontes de dados, a legalidade da extração, a proteção dos dados pessoais e a conformidade com as leis de proteção de dados, como a LGPD – Lei Geral de Proteção de Dados. Para garantir a governança dos dados coletados por meio de web scraping, é preciso adotar boas práticas de segurança, como a criptografia dos dados, a gestão de acesso e a monitorização de possíveis violações. Além disso, é preciso estabelecer políticas claras de uso e de compartilhamento dos dados, garantindo que as informações sejam utilizadas de maneira ética e responsável.

Já a monetização de dados refere-se à transformação dos dados em valor econômico, por meio da sua utilização em serviços ou produtos. No contexto de web scraping, a monetização de dados pode ocorrer de diversas formas, como a venda dos dados coletados, a utilização dos dados para a criação de modelos de negócios baseados em dados ou a sua utilização em serviços de análise e de consultoria.

Para a monetização de dados por meio de web scraping, é importante escolher as fontes de dados adequadas, que possam gerar informações relevantes para o mercado. Além disso, é preciso garantir a qualidade e a precisão dos dados coletados, para que os

insights gerados sejam confiáveis e possam agregar valor aos negócios.

No entanto, é preciso estar atento às questões éticas envolvidas na monetização de dados. O uso indiscriminado de dados pessoais ou a utilização dos dados de maneira não autorizada podem gerar problemas de privacidade e prejudicar a imagem da empresa. Por isso, é fundamental adotar políticas transparentes e éticas de monetização de dados, garantindo a proteção da privacidade dos indivíduos e o uso responsável das informações coletadas.

Em resumo, a governança e a monetização de dados são temas fundamentais quando se trata de web scraping. É preciso adotar boas práticas de governança de dados, garantindo a legalidade, a segurança e a conformidade com as leis de proteção de dados. Além disso, é importante explorar as oportunidades de monetização de dados, mas sempre garantindo a ética e a responsabilidade na utilização das informações coletadas.

Visamos utilizar estes conceitos na elaboração do nosso TCC e principalmante na criação de um contexto prático para ele, agregando valor não só aos dados mas sim aos usuário que os utilizem, podendo até gerar valor econômico.

2.4 Análise de Sentimentos

A análise de sentimentos é uma técnica de processamento de linguagem natural que tem como objetivo determinar a polaridade de um texto, ou seja, se ele é positivo, negativo ou neutro. Essa técnica pode ser utilizada em diversos tipos de dados, como avaliações de produtos, posts em redes sociais, comentários em fóruns, entre outros.

No contexto do nosso TCC sobre web scraping, a análise de sentimentos será uma das técnicas utilizadas para avaliar a qualidade dos produtos de algumas marcas coletados.

A análise de sentimentos por si só pode ser uma tarefa muito complexa de inteligência artificial e exigir uma quantidade de dados muito grande para treinamento do modelo de linguagem. Além disso, existem SaaSs (*Softwares as a Service*) em diversas provedoras de Cloud disponíveis para utilizarmos por meio de APIs, como a *Cloud Natural Language* do *Google Cloud* ou *Amazon Comprehend* da *Amazon Web Service* que realizam esse tipo de tarefa.

Tendo ambos aspectos em vista, utilizaremos um desses serviços, a determinar de acordo com a que melhor se encaixa na nossa aplicação para a análise de sentimento e apresenta custos mais atraentes.

No entanto, a análise de sentimentos também apresenta algumas problemáticas. Uma das principais é a ambiguidade do texto, que pode levar a diferentes interpretações dependendo do contexto. Por exemplo, a frase "esse celular é tão grande quanto um tijolo"pode ser interpretada como positiva ou negativa dependendo da perspectiva do

usuário. Uma análise correta da resposta da API utilizada determinará a polaridade do comentário específico.

Outra problemática é a falta de contexto. Muitas vezes, um texto pode ser considerado negativo em relação a um produto específico, mas positivo em relação a outro. Por exemplo, um comentário que reclama da duração da bateria de um smartphone pode ser negativo em relação a esse aspecto específico, mas positivo em relação às demais características do aparelho.

Para isso, utilizaremos o conceito de entidade presente nas APIs supracitadas. A partir deste conceito, podemos determinar para a entidade "duração da bateria"que a análise de sentimento é, por exemplo, negativa, mas para a entidade "construção"a avaliação pode ser positiva.

2.5 Dashboards e SelfService BI

A criação de um dashboard ou frontend com gráficos interativos é uma parte fundamental do nosso TCC sobre web scraping com ETL em nuvem e análise de sentimentos. O objetivo principal é transformar os dados coletados em insights valiosos que possam ser facilmente compreendidos pelos usuários finais. Para isso, é necessário utilizar as tecnologias mais adequadas para criar visualizações intuitivas e interativas.

Uma das principais tecnologias utilizadas para a criação de dashboards e frontends é o JavaScript, em especial as bibliotecas e frameworks como React, Vue.js, Angular e D3.js. Essas ferramentas permitem criar interfaces de usuário dinâmicas e responsivas, além de possibilitar a criação de gráficos interativos, como gráficos de linha, barras, pizza, scatterplot, mapa de calor, entre outros.

No entanto, a escolha da ferramenta mais adequada depende das necessidades específicas do projeto. No contexto do nosso TCC, podemos utilizar o React como biblioteca principal para construir os componentes visuais, combinado com o D3.js para a criação de gráficos mais complexos e interativos.

Além disso, podemos utilizar outras ferramentas para a criação de dashboards, como o Tableau, Power BI, Google Data Studio, entre outros. Essas plataformas permitem a integração de diferentes fontes de dados, a criação de dashboards com diversos tipos de gráficos e a possibilidade de compartilhar com outros usuários.

Para garantir a governança dos dados e a segurança da informação, é importante considerar o uso de autenticação de usuários, criptografia de dados e políticas de acesso e privacidade. Além disso, é importante pensar em formas de monetização dos dados coletados, seja através de venda de insights para empresas interessadas ou de outros modelos de negócio.

Dessa forma, a criação de um dashboard ou frontend com gráficos interativos é uma etapa importante para transformar os dados coletados em informações valiosas para os usuários finais.

Com o uso das tecnologias adequadas e a consideração de aspectos de governança e monetização de dados, podemos garantir a qualidade e utilidade do resultado final do nosso TCC.

2.6 E-commerces e comércio na Web

O comércio eletrônico, ou e-commerce, constitui uma modalidade de negócios que se destaca pela sua dinâmica intensa e pela vastidão de dados que permeiam esse ecossistema digital. No âmago do e-commerce, está a realização de transações comerciais por meio da internet, viabilizando a compra e venda de uma gama diversificada de produtos e serviços. Esse ambiente é caracterizado por sua natureza dinâmica, refletida na constante evolução das plataformas digitais, nas estratégias de marketing e na própria preferência do consumidor.

A dinamicidade do e-commerce é amplificada pela abundância de dados gerados diariamente. Milhões de transações, avaliações de produtos, interações dos consumidores e informações de mercado fluem incessantemente pelas plataformas online. Este volume massivo de dados apresenta um desafio significativo para os gestores do e-commerce, exigindo o uso eficaz de ferramentas de análise de dados e tecnologias como machine learning para extrair insights valiosos que impulsionem a tomada de decisões estratégicas.

A complexidade do acompanhamento no e-commerce reside na natureza multifacetada dos produtos. Um único item pode ter diversas variações, desde diferentes cores e tamanhos até modelos ou versões específicas. Essas variações, muitas vezes, resultam em valores distintos, impulsionados por estratégias de precificação dinâmica, promoções temporárias e condições de mercado em constante mudança. O desafio está em acompanhar essas flutuações de preço e disponibilidade, garantindo precisão e atualização constante para proporcionar aos consumidores uma visão completa e precisa dos produtos oferecidos.

Além disso, o e-commerce não apenas oferece produtos, mas também serviços associados, como frete, seguro, garantias estendidas, entre outros. A gestão eficiente desses serviços, alinhada à rápida adaptação às demandas e expectativas do mercado, representa outro aspecto desafiador. Estratégias de logística, parcerias estratégicas e a oferta de uma experiência de compra fluida são cruciais para atender às crescentes expectativas dos consumidores online.

Em resumo, o e-commerce é um ecossistema repleto de oportunidades, mas também de desafios. Sua dinâmica intensa, a enorme quantidade de dados envolvidos, a complexidade

na gestão de produtos e serviços, e a necessidade de adaptação constante fazem do e-commerce um campo empolgante, porém complexo, onde a agilidade e a inovação são essenciais para o sucesso.

3 Metodologia do trabalho

A metodologia adotada para o desenvolvimento deste projeto segue uma abordagem estruturada, dividida em cinco fases distintas, que são descritas a seguir.

3.1 Fase 0: Ideação

Na fase inicial, foi realizado um processo de ideação que resultou na concepção do projeto, considerando a demanda por análise de dados em e-commerces. Nesta etapa, foram definidos os objetivos do sistema, o escopo do web scraper e da análise de sentimento, bem como a identificação dos stakeholders envolvidos.

3.2 Fase 1: Levantamento de Requisitos e Visões de Arquitetura

A segunda fase concentrou-se no levantamento detalhado dos requisitos funcionais e não funcionais, considerando o desenvolvimento do web scraper para coleta de dados em e-commerces, a integração com a API de análise de sentimento do Google e a construção do Dashboard SelfService. As visões de arquitetura foram delineadas, abrangendo distribuição, infraestrutura, informação, computacional e tecnologia.

3.3 Fase 2: Busca por tecnologias que atendem aos requisitos

Com base nos requisitos específicos do projeto, a terceira fase envolveu uma pesquisa aprofundada para a escolha das tecnologias mais adequadas. Isso incluiu a seleção de linguagens de programação, bibliotecas e frameworks para a implementação eficaz do web scraper, da análise de sentimento e do Dashboard SelfService.

3.4 Fase 3: Desenvolvimento e Execução

A implementação prática do projeto ocorreu nesta fase, com o desenvolvimento do web scraper para coleta de dados em tempo real nos e-commerces identificados. A integração com a API de análise de sentimento da Google foi realizada, e o Dashboard SelfService foi construído para proporcionar uma visualização intuitiva e eficaz dos dados. O grupo se dividiu igualmente para desenvolver a solução de ponta a ponta.

3.5 Fase 4: Coletas de Dados e Conclusões

A última fase do projeto concentra-se na coleta efetiva de dados por meio do web scraper em operação. Os resultados são analisados, comparados com as expectativas iniciais e conclusões são extraídas em relação à eficácia do sistema em fornecer dados valiosos para análise de BI. Além disso, nesta fase, a documentação final do projeto é elaborada, preparando o terreno para futuras melhorias e expansões. Foi nessa etapa que o documento que você tem em mãos foi desenvolvido.

4 Especificação de Requisitos

4.1 Visão de Arquitetura: Engenharia

Distribuição: Teremos uma distribuição de dos dados no modelo B2B (business to business) - é um termo de vendas que representa negócios realizados por empresas para outras empresas. Ao contrário do B2C, que representa acordos de uma pessoa jurídica para outra física. No nosso modelo de negócios, entregaremos os dados previamente acordados 1x por semana, ou então no prazo acordado com o cliente.

Infraestrutura: Permitiremos que o nosso Bot Web Scrapper rode num servidor nosso. Contaremos com nossa API que processa o sentimento das mensagens por conexão à AI do Google. Disponibilizaremos as informações através do Dashboard SelfService desenvolvido.

4.2 Visão de Arquitetura: Empresa

Nosso objetivo como empresa: providenciar ao cliente os melhores dados já tratados para que seu BI se debruce neles. Nosso escopo: Ser dona do processo de webscrapping e agregar valor com analise de sentimento Nossas Politicas: Respeito à LGPD e fornecimento dos dados aos clientes

Abaixo uma figura de um diagrama mostrando como é o fluxo no ponto de vista da nossa empresa cliente.

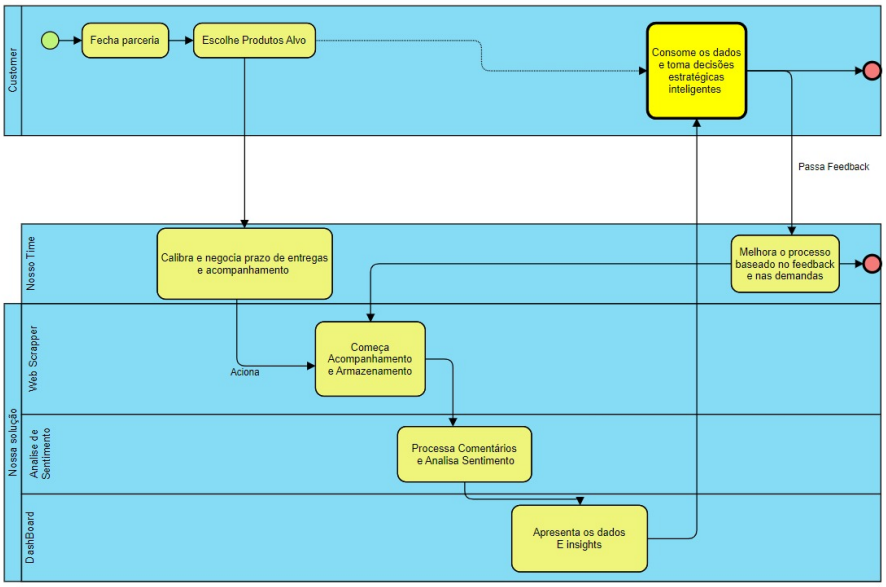

Figura 1 – Diagrama BPMN

4.3 Visão de Arquitetura: Informação

Informação manipulada é aquela que os clientes solicitaram. Preço dos produtos + análise dos comentários nos e-commerces que buscaremos

4.4 Visão de Arquitetura: Computacional

1 - Aplicação de Web Scrapping rodando em servidor e armazenando os dados 2 - Leitura de sentimento feita sobre os dados com IA 3 - Representação dessa geração de Insights em uma dashboard pro usuário

4.5 Visão de Arquitetura: Técnologia

Tecnologia: Bot Python + Biblioteca de IA + React Código: Próprio, usando soluções já existentes Produtos: WebScrapper + IA + DashBoard como FrontEnd

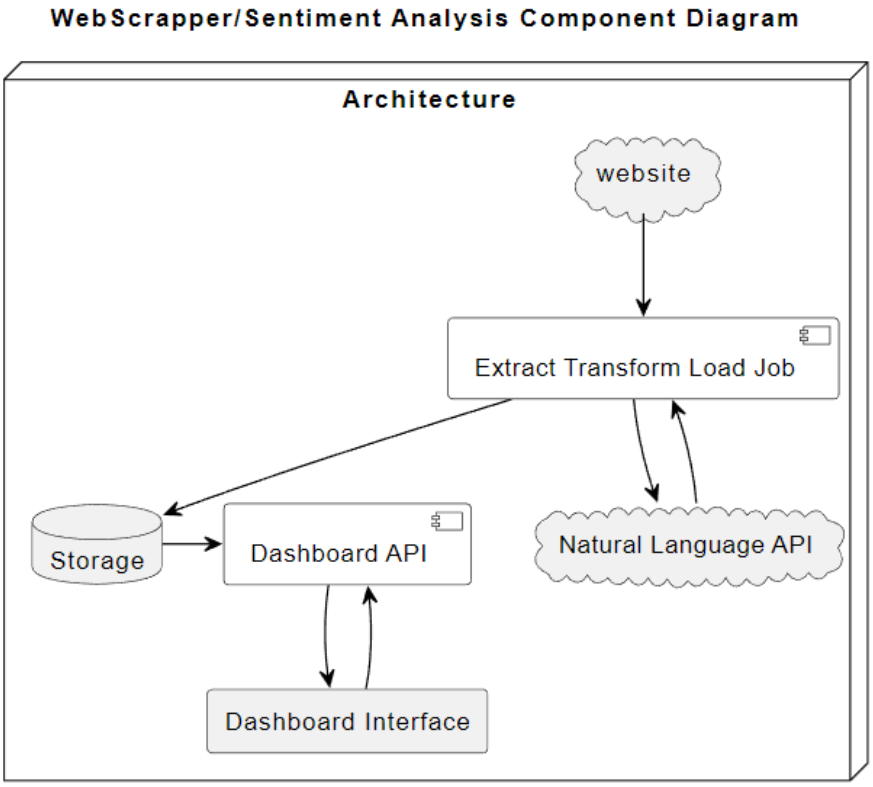

Figura 2 – Arquitetura Simplificada

4.6 Requisitos Funcionais

1. Web Scrapping:

- O sistema deve ser capaz de realizar web scrapping em sites específicos de e-commerce.

- Deve extrair informações relevantes, como preço dos produtos e comentários dos clientes.

2. Processamento de Sentimento:

- A análise de sentimento deve ser realizada em tempo real.

- A análise de sentimento deve extrair sentimento a respeito de cada entidade dos comentários.

3. Armazenamento de Dados:

- Os dados coletados pelo web scraper devem ser armazenados de forma segura em um bucket em nuvem.

- Deve ser possível acessar e recuperar esses dados para análises futuras.

4. Distribuição de Dados:

- Os dados processados devem ser distribuídos para os clientes no modelo B2B.

- A entrega deve ocorrer semanalmente ou conforme acordado com o cliente.

5. Dashboard SelfService:

- Desenvolver um Dashboard SelfService para que os clientes possam visualizar as informações de maneira intuitiva.

- As informações devem ser apresentadas de forma clara e organizada.

6. Conformidade com LGPD:

- O sistema deve seguir as diretrizes e regulamentações da Lei Geral de Proteção de Dados (LGPD).

-Garantir que os dados dos clientes sejam tratados com respeito à privacidade e segurança.

4.7 Requisitos Não Funcionais

1. Desempenho:

- A aplicação de web scrapping deve ter um desempenho eficiente, garantindo a coleta rápida e precisa de dados.

- O tempo de resposta do sistema como um todo deve ser otimizado para garantir uma experiência ágil ao usuário.

2. Segurança:

- Implementar medidas de segurança robustas para proteger os dados coletados e armazenados.

- Garantir que a comunicação entre os componentes da arquitetura seja segura.

3. Disponibilidade:

- Garantir alta disponibilidade para o web scraper, API de processamento de sentimento e dashboard.

- Minimizar o tempo de inatividade para garantir acesso contínuo aos dados.

4. Compatibilidade:

- Assegurar que a aplicação seja compatível com diferentes navegadores para o Dashboard SelfService.

- Garantir compatibilidade com sistemas operacionais diversos para o servidor e componentes principais.

5. Manutenibilidade:

- Desenvolver o sistema de forma modular e documentada para facilitar a manutenção e futuras atualizações.

- Garantir que novas fontes de dados possam ser adicionadas de maneira eficiente.

6. Usabilidade:

- O Dashboard SelfService deve ser intuitivo e fácil de usar, mesmo para usuários não técnicos.

- Fornecer suporte adequado para que os clientes possam explorar e interpretar os dados facilmente.

7. Compliance:

- Assegurar que o sistema esteja em conformidade com as políticas e regulamentações internas da empresa, além das leis de proteção de dados vigentes.

5 Desenvolvimento do Trabalho

5.1 Tecnologias Utilizadas

As tecnologias utilizadas em nosso projeto incluem, mas não se limitam a

Scrapper:

- Scrapy

- Playwright

- Python

- Docker

- Cloud Storage (Google Cloud)

Análise de sentimento:

- Python

- Flask - biblioteca em Python utilizada para exportar uma API

- Cloud Storage (Google Cloud)

- Natural Language AI (Google Cloud)

- Google Cloud Translation

Dashboard: - Grafana

5.2 Detalhamento técnico do Scrapper

O processo de ETL (Extract, Transform, Load) é essencial para coletar, transformar e armazenar dados provenientes da extração de informações de páginas da web. Esse sistema pode ser dividido em 3 componentes principais:

Extract (Extração): Ele é responsável por coletar dados a partir das páginas páginas HTML na web. O objetivo é extrair informações de preços, descrições e comentários de compradores do produto para análise posterior.

Transform (Transformação): Durante a transformação, os dados extraídos passam por processos que os preparam para análise. Isso envolve limpeza de dados, conversão de formatos, e qualquer manipulação necessária para adequar os dados às necessidades específicas do usuário ou do sistema.

Load (Carregamento): Na fase de carregamento, os dados transformados são arma-

zenados em um local apropriado para acesso futuro. No nosso caso, escolhemos um bucket em nuvem para isso

Primeiramente, os preços e descrições são extraídos da página de busca de produtos, como a seguir:

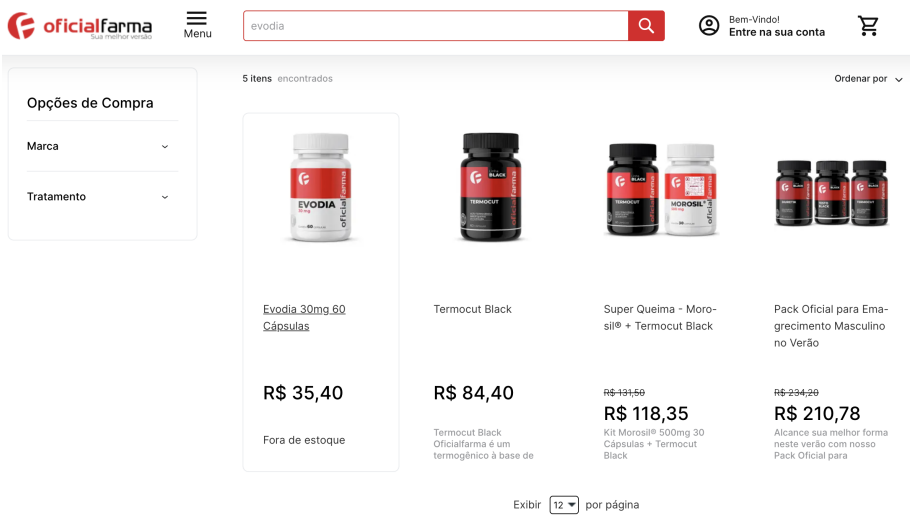

Figura 3 – Página de Busca por medicamento Evodia na loja Oficial Farma

O sistema realiza a extração de comentários em uma etapa separada, pois isso envolve acessar o link obtido na etapa anterior, além de realizar renderização dinâmica para acessar os comentários. como na página a seguir:

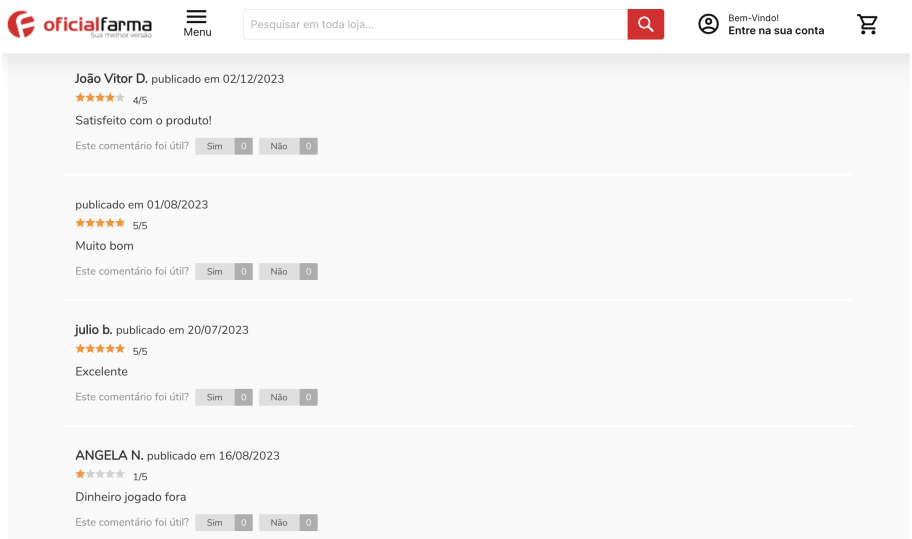

Figura 4 – Página de Descrição e Comentários do medicamento Evodia na loja Oficial Farma

Comentários extraídos:

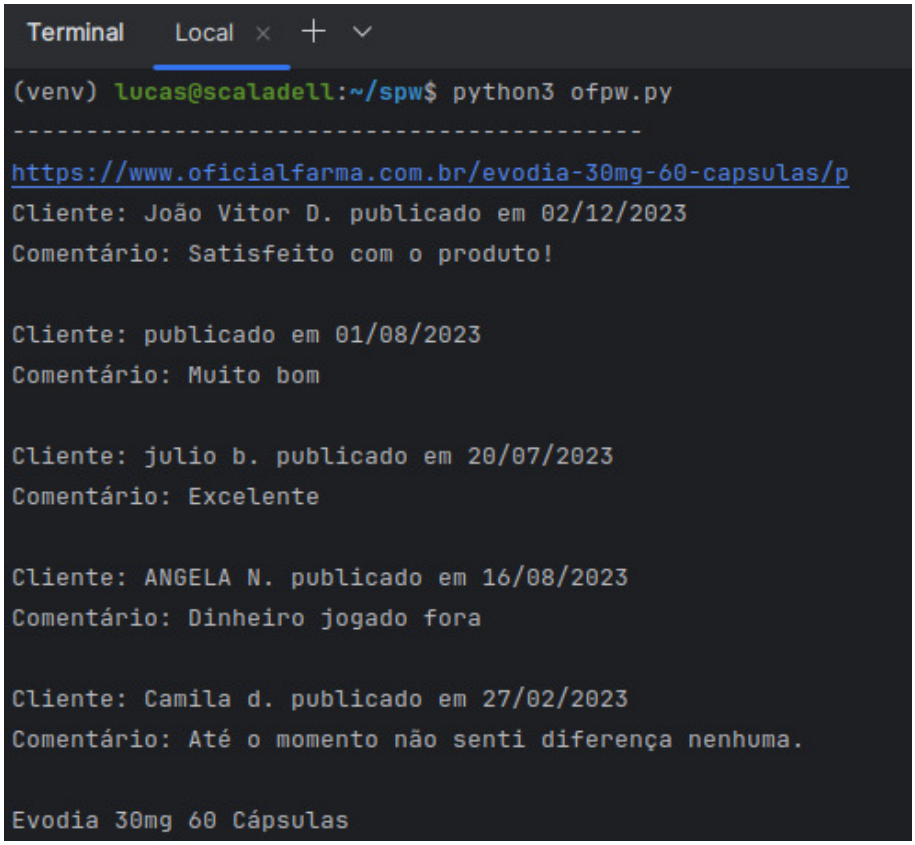

Figura 5 – Comentários extraídos por meio de scraping

O sistema de Scraper foi desenvolvido em cima do framework Scrapy. Ele é um framework de scraping em Python que simplifica a extração de dados de websites de maneira estruturada. Construído sobre o Twisted, um framework assíncrono para Python, Scrapy oferece uma arquitetura robusta para a construção de spiders, facilitando a criação de robôs de scraping.

Porém, somente o Scrapy não é suficiente para lidar com renderização dinâmica, especialmente no momento de extrair os comentários dos produtos, que pode ter alguma interação de scroll ou botão para mostrar mais resultados. Para a renderização dinâmica, utilizamos o Playwright que, por sua vez, é uma ferramenta de automação de browser compatível com várias linguagens de programação, incluindo Python. Essa ferramenta possibilita o controle e a interação programática com navegadores, tornando-a útil para scraping em sites com conteúdo dinâmico ou interativo.

Python foi escolhida pois é uma linguagem de programação de alto nível amplamente utilizada em diversas áreas, incluindo desenvolvimento web e automação. Sua simplicidade, legibilidade e ampla oferta de bibliotecas são ideais para o desenvolvimento de web scrapers.

Para resolver problemas de compatibilidade e possibilitar o processo de conteinerização, utilizamos Docker, uma plataforma de conteinerização, facilita a empacotação e distribuição de aplicativos em ambientes isolados chamados containers. Essa abordagem

proporciona consistência e facilidade de implantação em diferentes ambientes, sendo útil para encapsular o ambiente de execução do web scraper.

Por fim, os dados extraídos são convertidos para um formato CSV(Comma Separated Value) e armazenados em bucket no Cloud Storage, um serviço oferecido pelo Google Cloud Platform, permite o armazenamento e recuperação escaláveis e seguros de dados.

5.3 Detalhamento técnico da API de Análise de Sentimento

O fluxo geral consiste em exportar os dados armazenados no Cloud Storage por meio da API desenvolvida com Flask. Os comentários são enviados para a Google Cloud Translation para tradução para o inglês, a análise de sentimento de entidade é feita por meio da Natural Language AI e posterior tradução de volta para o português utilizando novamente a Google Cloud Translation. Os resultados são então estruturados no formato a ser consumido pelo Dashboard, onde cada entidade possui informações sobre score e magnitude. Esse processo integrado proporciona uma solução abrangente para analisar e estruturar dados provenientes de scraping para uso em um frontend.

Utilizando os comentários extraídos do produto Evodia raspado anteriormente, podemos fazer a análise de sentimento das entidades:

```
➜  ~ curl http://127.0.0.1:5000 | json_pp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   437  100   437    0     0     34      0  0:00:12  0:00:12 --:--:--    94
[
   {
      "name" : "produtos",
      "sentiment" : {
         "magnitude" : 0.699999988079071,
         "score" : 0.699999988079071
      }
   },
   {
      "name" : "Product",
      "sentiment" : {
         "magnitude" : 0.899999976158142,
         "score" : 0.899999976158142
      }
   },
   {
      "name" : "Product",
      "sentiment" : {
         "magnitude" : 0.899999976158142,
         "score" : 0.899999976158142
      }
   },
   {
      "name" : "Dinheiro",
      "sentiment" : {
         "magnitude" : 0.300000011920929,
         "score" : -0.300000011920929
      }
   },
   {
      "name" : "diferença",
      "sentiment" : {
         "magnitude" : 0,
         "score" : 0
      }
   }
]
```

Figura 6 – Análise de sentimento dos comentários

Como pode ser observado pela imagem, o sistema identifica que comentários que não se refiram a uma entidade específica estejam falando a respeito do produto em si, caracterizados com o nome "Product"no nosso exemplo, mas que pode ser parametrizado a depender do produto em questão. Além disso, devemos notar que a tradução pode não ser perfeita. Neste caso, podemos executar um passo final de encontrar sinônimos e acumular o sentimento de diversos comentários em uma análise só, por meio de uma média aritmética entre as avaliações.

Sobre seu sentimento, o score indica a positividade do comentário(neste caso negativa) e a magnitude indica a força geral da emoção sobre a entidade.

Para desenvolver este sistema, optamos pela mesma linguagem de programação que o Scraper, para facilitar manutenção. Utilizamos também Flask, uma biblioteca em Python, utilizada para criar APIs de maneira eficiente e simples.

Natural Language AI, parte do Google Cloud, é empregado para realizar a análise de sentimento nos comentários extraídos. Essa funcionalidade proporciona insights sobre a polaridade e magnitude dos sentimentos expressos nos textos.

Google Cloud Translation, outra ferramenta poderosa do Google Cloud, é utilizada para traduzir os comentários para o inglês, uma vez que a funcionalidade de análise de entidade do Natural Language AI não suporta português. Após a análise de sentimento em inglês, os resultados são traduzidos de volta para o português.

5.4 Detalhamento técnico do DashBoard

Para a visualização de dados de uma forma eficiente e customizável, escolhemos Grafana, uma plataforma open source de observabilidade e análise de dados que fornece ferramentas avançadas para visualização e monitoramento de métricas, logs e dados de séries temporais. Projetada para ser altamente modular e extensível, Grafana é amplamente utilizada na construção de dashboards interativos e personalizáveis que permitem aos usuários visualizar e analisar dados provenientes de diversas fontes.

Os exemplos a seguir são gráficos de histórico de preços de diferentes medicamentos na loja Oficial Farma:

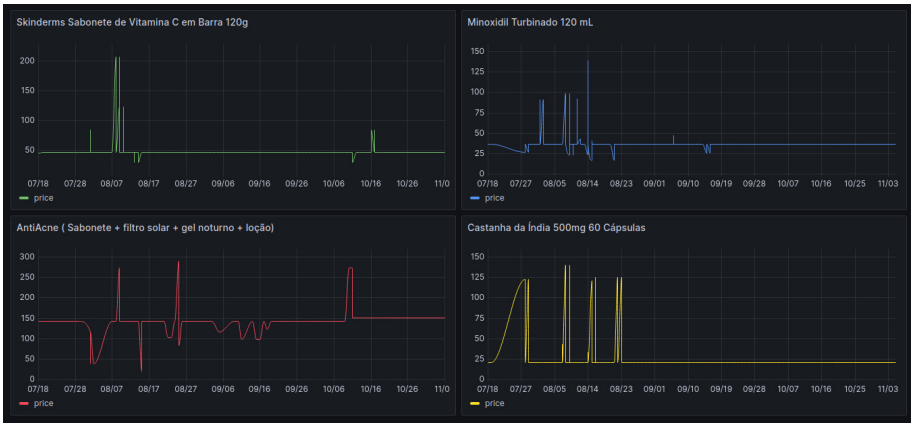

Figura 7 – Gráficos de Preços de produtos da Oficial Farma no Grafana

Alguns motivos fazem a escolha do Grafana ideal para o projeto:

- Grafana oferece uma variedade de opções de visualização de dados, incluindo gráficos de séries temporais, tabelas, medidores, e mapas. Isso permite representar os dados extraídos de maneira clara e eficiente.

- É altamente flexível e suporta integração com uma ampla gama de fontes de dados, incluindo bancos de dados relacionais, NoSQL, serviços em nuvem e APIs. Isso facilita a conexão com o Cloud Storage do Google para acessar os dados armazenados.

- A capacidade do Grafana de se integrar facilmente a APIs é crucial no contexto de uma aplicação que extrai dados raspados. A API desenvolvida em Flask pode ser conectada ao Grafana para buscar e exibir os dados em tempo real.

- Grafana permite a criação de dashboards interativos, onde os usuários podem explorar os dados, ajustar filtros e visualizar diferentes aspectos dos resultados. Isso proporciona uma experiência de usuário dinâmica e personalizável.

- Grafana é uma plataforma de código aberto, o que significa que você pode personalizá-la de acordo com suas necessidades específicas e contribuir para o desenvolvimento da comunidade.

De modo geral, a escolha do Grafana como frontend oferece uma combinação de flexibilidade, capacidade de visualização, integração e suporte à comunidade, tornando-o uma escolha sólida para apresentar os dados extraídos de forma acessível e interativa.

6 Considerações Finais

6.1 Conclusões do Projeto de Formatura

O projeto de formatura em Engenharia de Computação representa uma jornada de exploração e implementação no campo da análise de sentimentos e web scraping aplicados ao monitoramento de preços de produtos. Ao testar o projeto com foco em farmácias, evidenciamos sua capacidade de cumprir objetivos específicos e agregar valor. No entanto, alguns desafios foram enfrentados ao longo do caminho.

Enfrentamos um imprevisto ao lidar com a API Cloud Natural AI, que não possui a funcionalidade de análise de sentimento de entidade para a língua portuguesa. Para solucionar esse imprevisto, utilizamos a API Google Cloud Translation. No entando, devemos notar que a tradução pode não ser perfeita. Além disso, podemos executar um passo final de encontrar sinônimos e acumular o sentimento de diversos comentários em uma análise só, por meio de uma média aritmética entre as avaliações.

6.2 Contribuições

As contribuições deste projeto são multifactadas e refletem a dedicação e expertise da equipe. Em primeiro lugar, a implementação bem-sucedida da extração de dados por web scraping e a análise de sentimentos proporcionaram uma visão profunda sobre a dinâmica de preços de produtos e a satisfação do cliente.

A superação da barreira linguística, inicialmente restrita ao inglês, reforça a capacidade de adaptação da equipe e a busca contínua por soluções eficientes. Essa experiência contribuiu significativamente para a compreensão das nuances e desafios envolvidos na aplicação prática de tecnologias como web scraping e análise de sentimentos em contextos diversificados.

6.3 Perspectivas de Continuidade

O projeto abre portas para futuras explorações e aprimoramentos. Para uma continuidade bem-sucedida, sugerimos a expansão das funcionalidades para integrar mais fontes de dados e otimizar a análise de sentimentos em diferentes idiomas.

A incorporação de inteligência artificial avançada pode aprimorar ainda mais a precisão da análise de sentimentos, proporcionando insights mais ricos. Além disso, a

exploração de setores adicionais, além das farmácias, pode revelar aplicações valiosas em mercados diversos.

Vale lembrar que páginas na web não são estáticas, o que significa que podem sofrer alterações ao longo do tempo, o que faz com que o scraper tenha que ser modificado para se adequar a essas alterações, sendo um trabalho constante.

Em última análise, as perspectivas de continuidade incluem o refinamento constante do sistema, adaptando-se às demandas em evolução do cenário tecnológico e empresarial. O trabalho desenvolvido estabelece uma base sólida para futuras iterações, proporcionando uma plataforma robusta para inovação contínua.