Real-time BI Satisfaction System with Chatbots and Web Scraping.

Presented by Ahmed Hamza Gouissem

Research and Analysis:

A Real-time BI dashboard is a visualization tool that uses data updated at any moment, allowing us to have proactive monitoring. It helps executives make informed and timely decisions, which saves the company time and money.

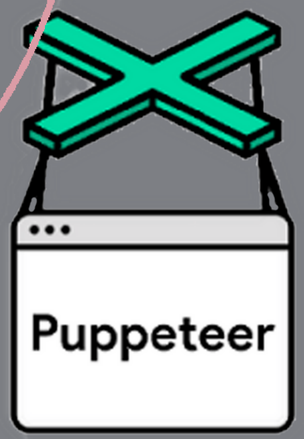
To enhance the database, we should use Web Scraping technologies. However, it could cause legal problems due to the regulations of various websites that do not condone any scraping activity. To avoid this problem, we need access to the “robots.txt” file or read the regulation terms of the website.

Creating a chatbot will ease the comprehension of the site and help the user change the visualization of the graphs to their specific needs. Plus, it will take any user feedback to enhance their experience in the platform.

Technology Stack Recommendations:

Web Scrapping

This represents the first step of the workflow: Data Collecting.
We have a range of scraping technologies to choose from: if we are working on a website constructed by HTML, we should use Beautiful Soup. However, due to the limitations of the technology, we have two other advanced options: Puppeteer (JavaScript) and Selenium (Python, Java). There is not much of a difference between the two: selenium supports various browsers (Puppeteer navigates only in Google Chrome and Chromium), so it depends more on the necessities of the enterprise.

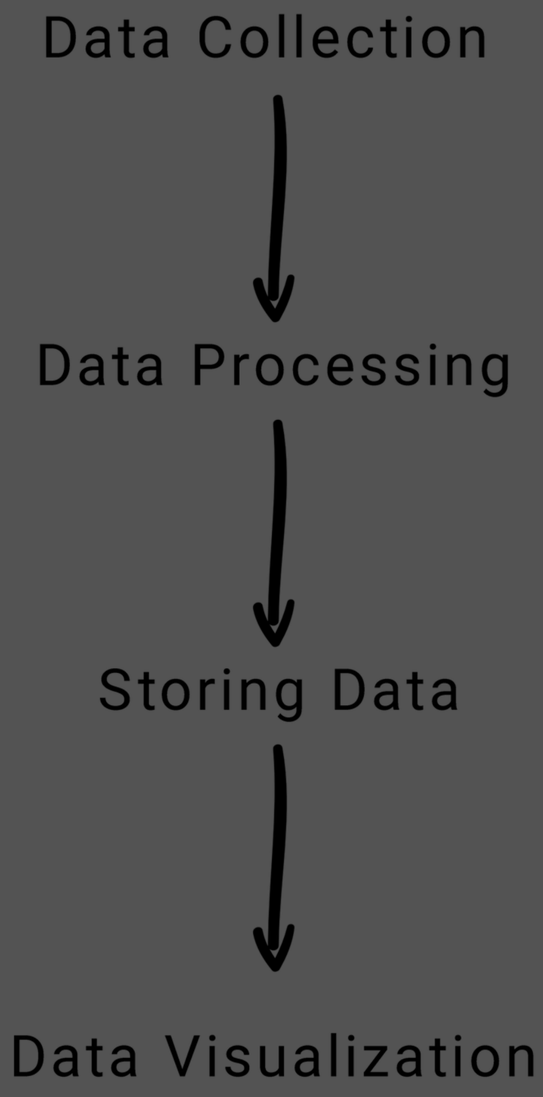


Beautiful Soup

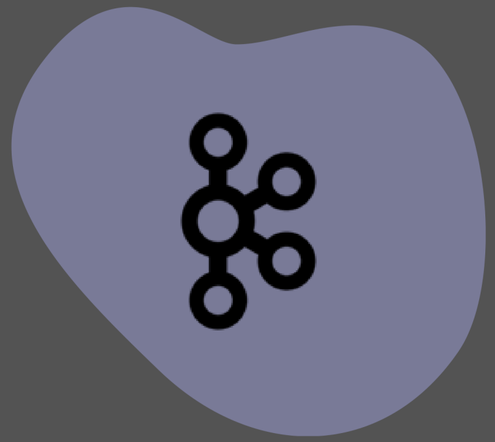
Se

Real-time Dashboard

The Workflow of building a real-time dashboard consists of collecting data at the beginning by scrapping it from websites. Then, process the data and clean it to store it in a dynamic database, which means a NoSQL database. Lastly, visualize the data based on the needs.
We distinguish from that we will need many frameworks and technologies.

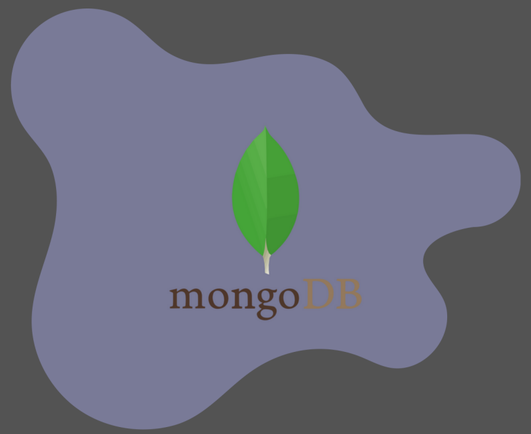


Real-time Dashboard



Data Processing

Apache Kafka, Apache Flink,
Apache Spark



Data Storing

MongoDB, Cassandra

Data Visualization

Plotly, Matplotlib, Seaborn

Chat Bot

Building a chatbot consists of many aspects. Therefore, there are multiple technologies to use. I will focus only on the developing perspective of making a chatbot:

- Natural Language Processor (NLP) is a crucial aspect when creating a chatbot for understanding and interpreting user input. We can use NLP tools and frameworks such as Dialogflow and RASA.
- For the chatbot deployment, we should consider cloud services such as AWS and Microsoft Azure due to the scalability factor that we will need for the bot.
- Several frameworks and platforms provide tools and APIs for building chatbots, hence Botpress and Bot Framework SDK.

Ensuring Trust and Reliability: Security Measures and Testing Strategies for Real-Time BI Dashboard Development

Security and Compliance Guidelines:

Of course, any web application that uses user data must secure them, so we need to encrypt the data and minimize it. We also need to respect laws such as the Personal Information Protection and Electronic Documents Act (PIPEDA), the Children's Online Privacy Protection Act (COPPA), and the General Data Protection Regulation (GDPR).

User Training Plan

For any new feature, especially if it is an non ordinary, we could do a quick tutorial like in games: showcasing the feature and what it does and if the user wants to try it or not.

Deployment and Monitoring Approach

We need to build the necessary features that any user of any level will need so that we can deploy it at the beginning and then we add feature by feature with the modifications based on the feedback of the users on the platform.

Testing Strategy

First, we need to test the functionality of the platform itself by testing the accuracy of data and its proper integration. Then, we test the performance: load testing, response time, and data refresh rates. After that, we do a security test such as SSL and Encryption. Usability Testing comes after the security test.

Demo

Demo

I build a prototype of this project. First, I scraped data from the Wikipedia page on the “largest companies by revenue” since Wikipedia content is released under the Creative Commons Attribution-ShareAlike License (CC BY-SA). I used Beautiful Soup since it is an HTML page, and it isn’t big data to collect. Then, I cleaned it and stored it in a MongoDB database.

In another Python script, I extracted the data from the database, processed it, created graphs with it, and displayed the graphs in the web application using Streamlit for building the web application and Plotly for the visualization. I added two forms to add and delete data and rerun the program as soon as one of the forms has been submitted. Otherwise, the web application will rerun every 200 seconds.

Finally, I tried to add a chatbot: we used Rasa to build the chatbot. However, I faced a lot of problems so I left the code as a comment.

Demo

As I mentioned, it is just a prototype: the program still lacks multiple features, like security checks, UI/UX components, etc. The chatbot NLP/NLU is still in development.
The program is still messy, so It needs another month to be cleaner, and hopefully, the program can be ready to use.

Thanks!