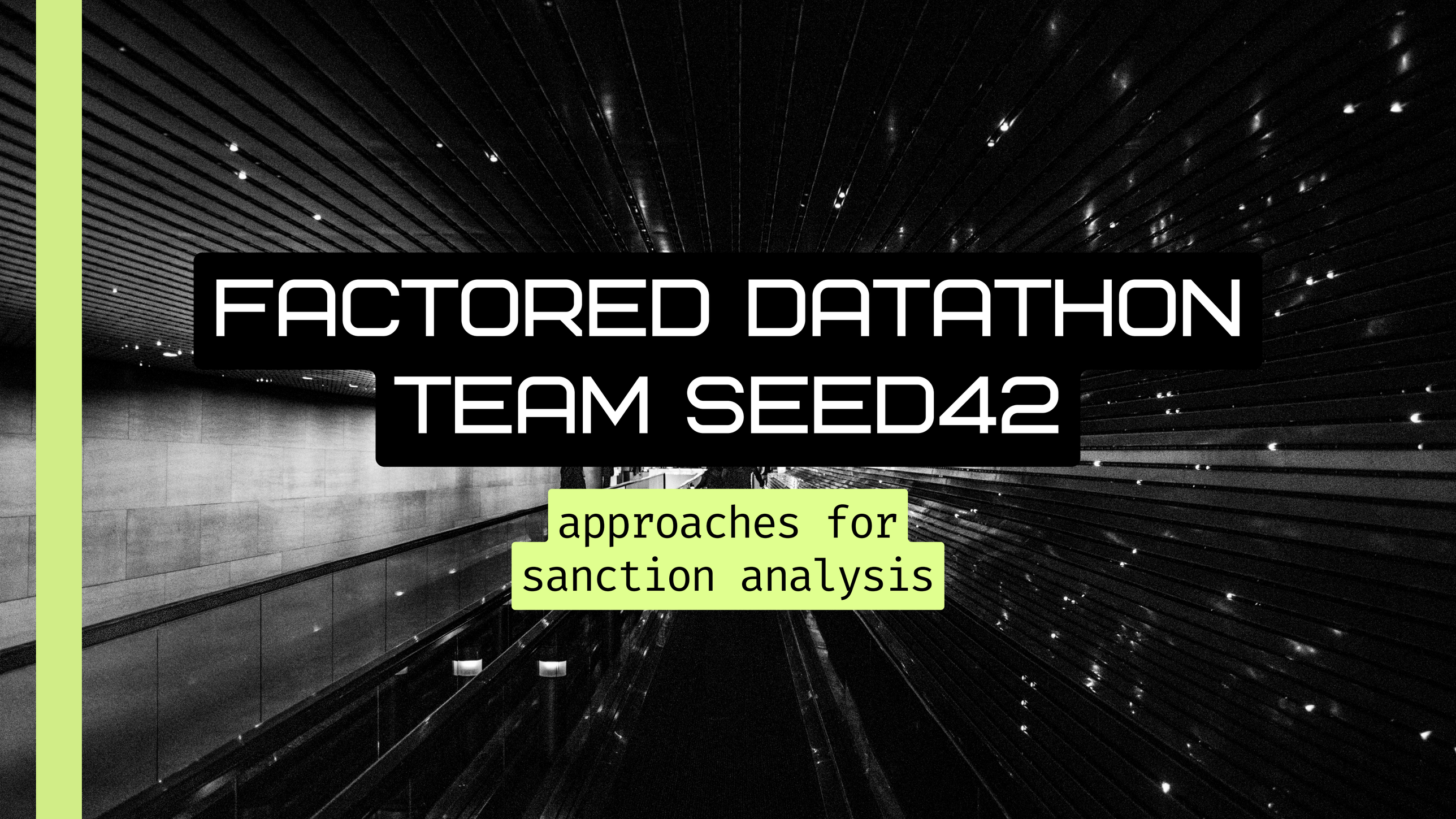

FACTORED DATATHON
TEAM SEED42
approaches for
sanction analysis

OUR APPROACH

Data Acquisition and Delivery

Utilizing a web scraper to automatically collect, organize, merge, and deliver data through the Spark framework.

Data Preparation and Initial Analysis

Preparing the data was essential to tailor it for our application development needs.
Initial analysis was made using Jupyter Notebooks over PySparkSQL.

Goldstein Scale Predictor API

A model capable of predicting the score was created and delivered to users who need to perform simulations and analysis over this metric.

RAG Based Chat Bot Over the Data

A LLM was deployed using LangChain agents and tool to query the data and run analysis over it, answering the users questions with our chatbot

Infrastructure

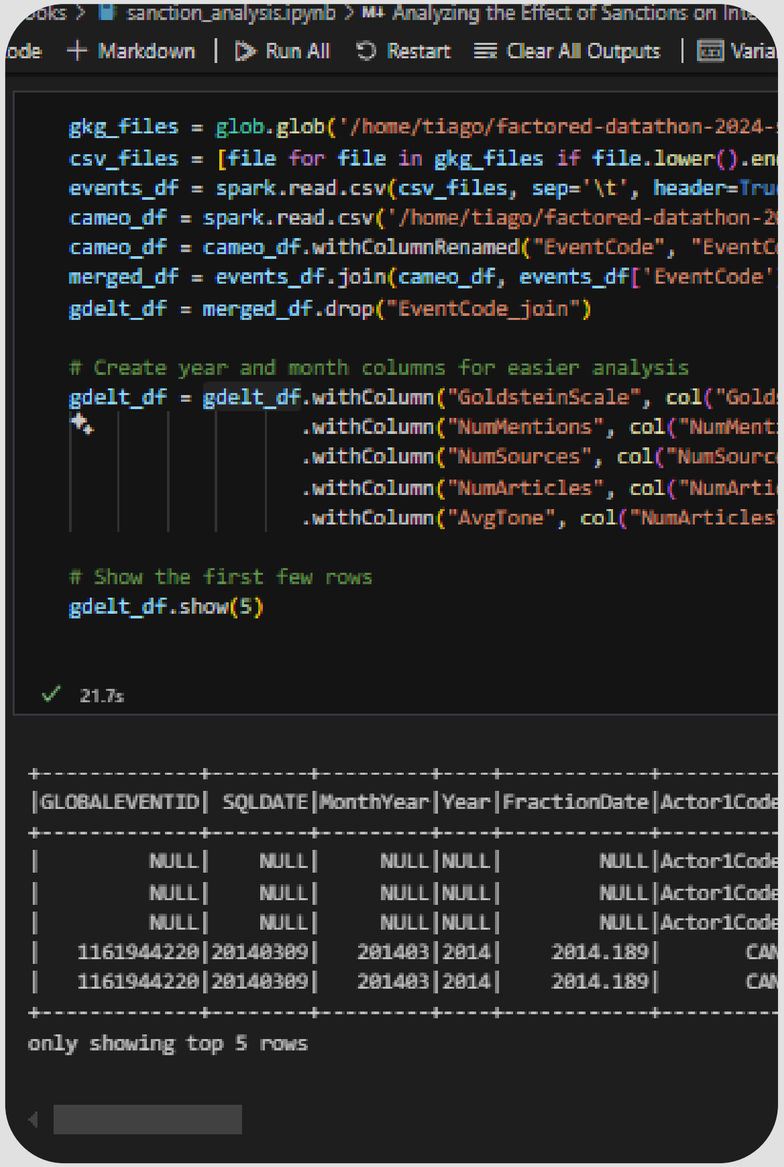
The initial version was developed on a simple architecture using Spark and accessed via Streamlit, APIs, or CLI. However, a more robust architecture has been planned for future implementation in the project. Currently, the project is only running locally.

EDA

Our data analysis focused on sanction events, particularly by combining GDELT databases with CAMEO codes.

The analysis covers data from January 1, 2023, to October 16, 2024, and includes:

- Geographic Distribution of Sanction-Related Events
- Temporal Trend of Sanction-Related Events
- Sentiment Analysis Related to Sanctions
- Identifying Key Actors in Sanction-Related Events
- Sentiment Analysis Over Time for Specific Countries
- Co-occurrence Analysis of Actors
- Temporal Analysis of Sanction Events by QuadClass
- Impact Analysis: Sanctions and Media Attention
- Analyzing the Effect of Sanctions on International Relations



GOLDENSTEIN SCALE PREDICTOR

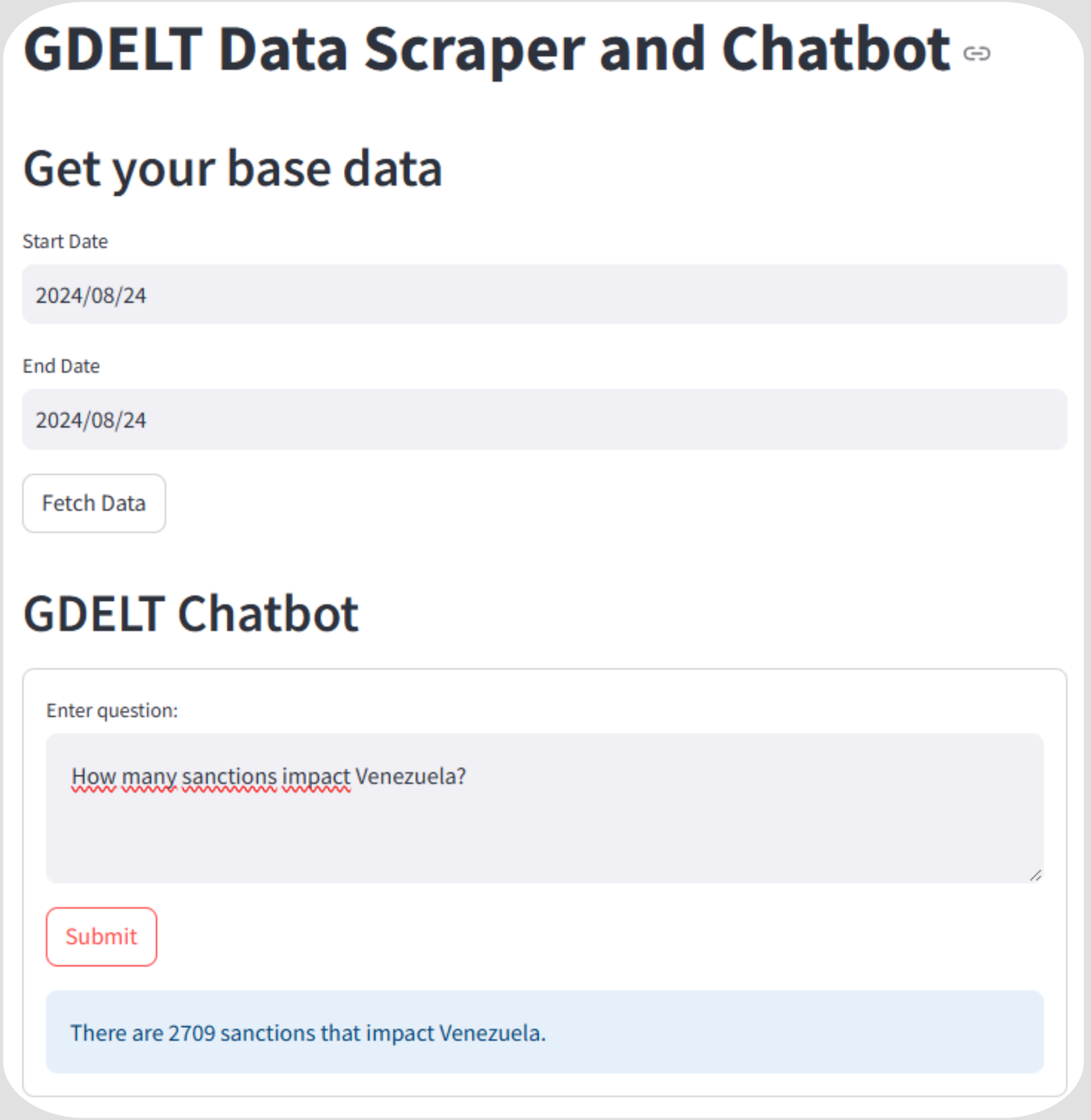
To help organizations simulate and predict the impact of sanctions on a global scale, we developed a Machine Learning model capable of predicting the Goldstein Scale using various key variables.

The model, a GBTRegressor, serves as a robust baseline for further enhancements and refinements.

```
gbt = GBTRegressor(labelCol="GoldsteinScale", featuresCo

# Define the parameter grid
paramGrid = ParamGridBuilder() \
    .addGrid(gbt.maxDepth, [5, 10]) \
    .addGrid(gbt.maxIter, [50, 100]) \
    .addGrid(gbt.stepSize, [0.05, 0.1]) \
    .build()

# Define the evaluator
evaluator = RegressionEvaluator(labelCol="GoldsteinScale"

# Set up CrossValidator
crossval = CrossValidator(estimator=gbt,
                          estimatorParamMaps=paramGrid,
                          evaluator=evaluator,
                          numFolds=5)  # 5-fold cross-val

# Fit the CrossValidator to find the best model
cv_model = crossval.fit(validation_data)

# Get the best model
best_model = cv_model.bestModel

# Evaluate the best model on the test data
test_predictions = best_model.transform(test_data)

rmse_test = evaluator.evaluate(test_predictions)
print(f"Best Model RMSE on test data: {rmse_test}")

# Calculate MAPE on the test data
test_predictions = test_predictions.withColumn("absolute_
mape = test_predictions.selectExpr("avg(absolute_percenta
print(f"Best Model MAPE on test data: {mape}")

# Show sample predictions
test_predictions.select("prediction", "GoldsteinScale",
```

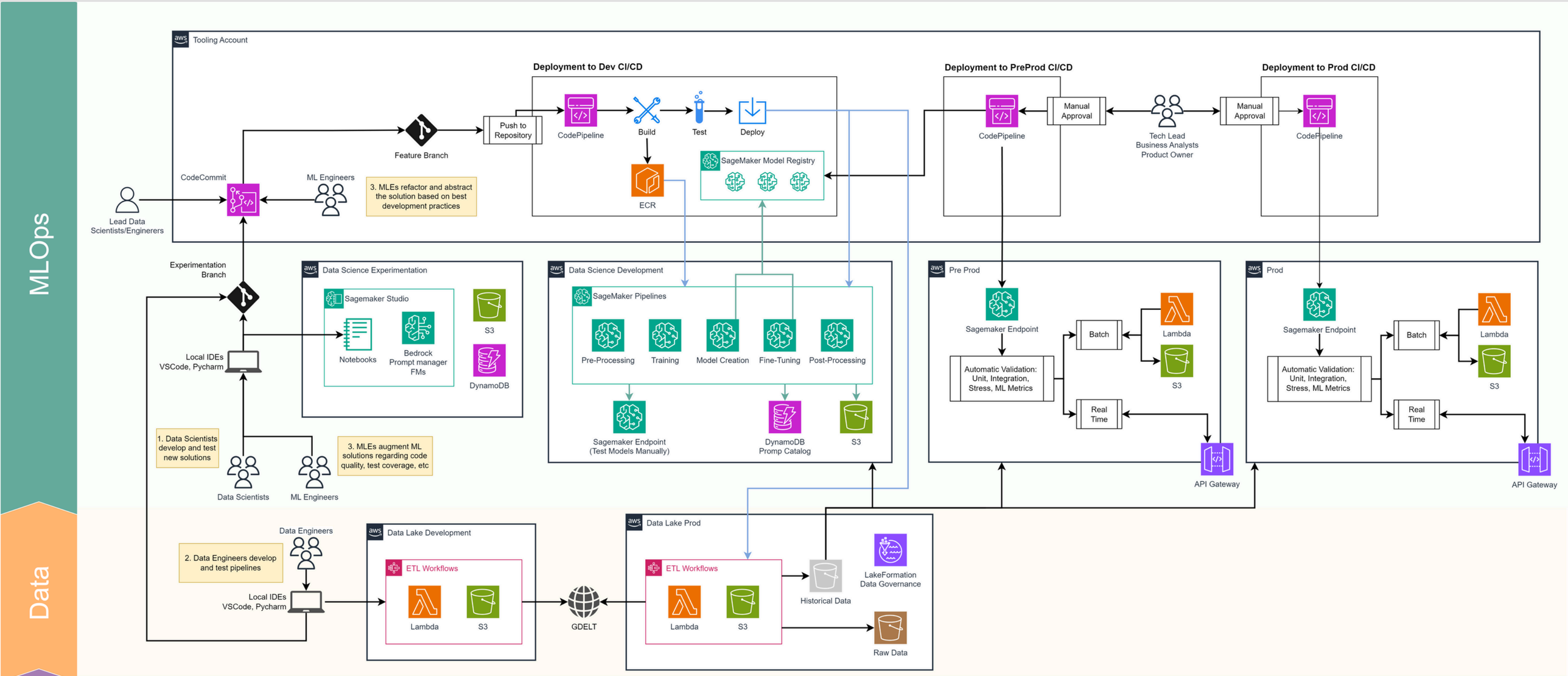
GDELT CHATBOT

Additionally, we have developed a chatbot using a Streamlit application to simplify data analysis and handling. This chatbot enables users to input a data range for analysis and ask questions to an LLM model, which utilizes Langchain Agents and Tools in a RAG-like structure for data analysis and reasoning.



INFRASTRUCTURE

Although for this first version we opted for a local and simplified infrastrucutre, we propose the following design to make this solutions to production.

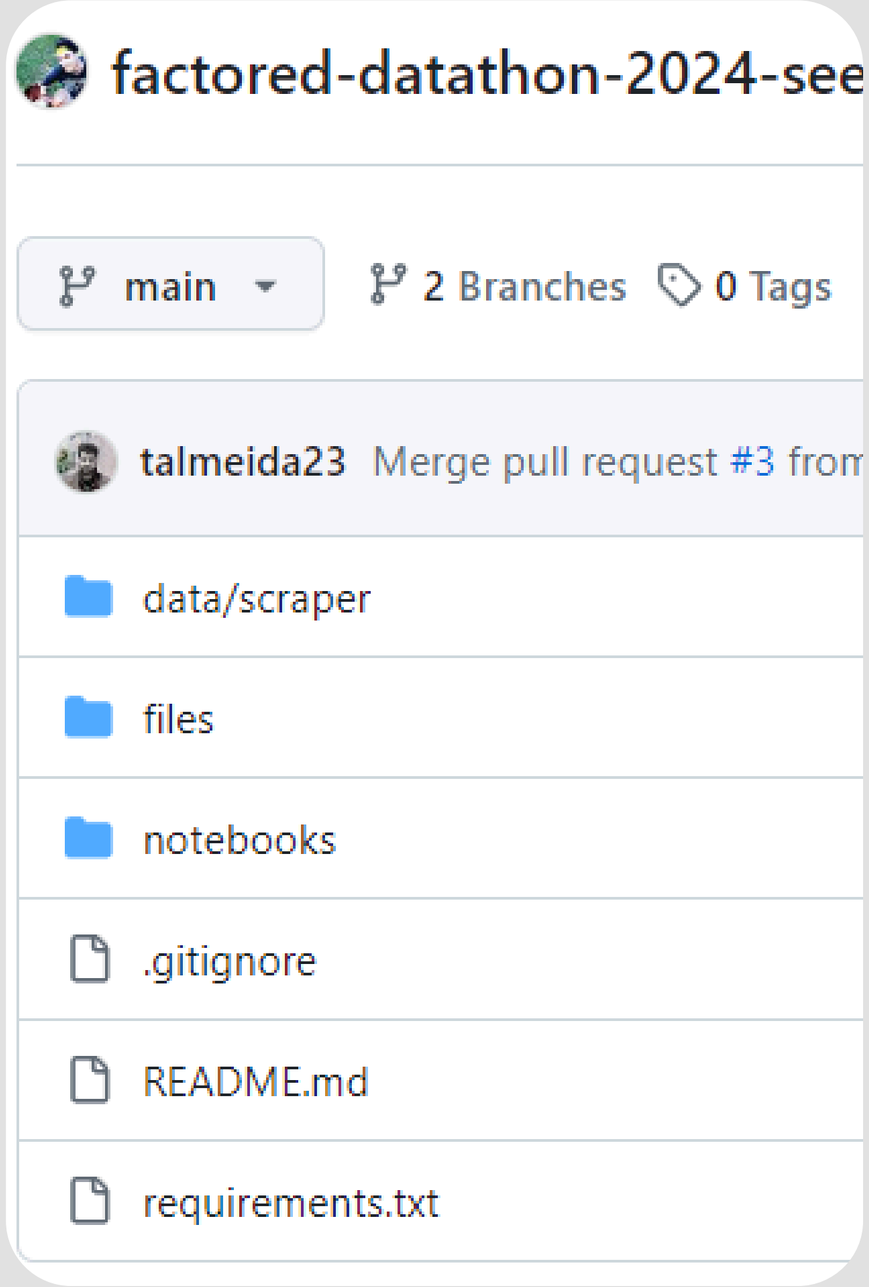


OUR GITHUB REPOSITORY

Our github repository can be acessed through the following link and contain the content below

https://github.com/glev1/factored-datathon-2024-seed42

Content:
app - Our Streamlit and API application
notebooks - Our Jupyter EDA notebooks
packages - The libraries we built to support our development
README.md - Our general documentation
requirements.txt - The Python packages we used




THANK YOU
team seed42