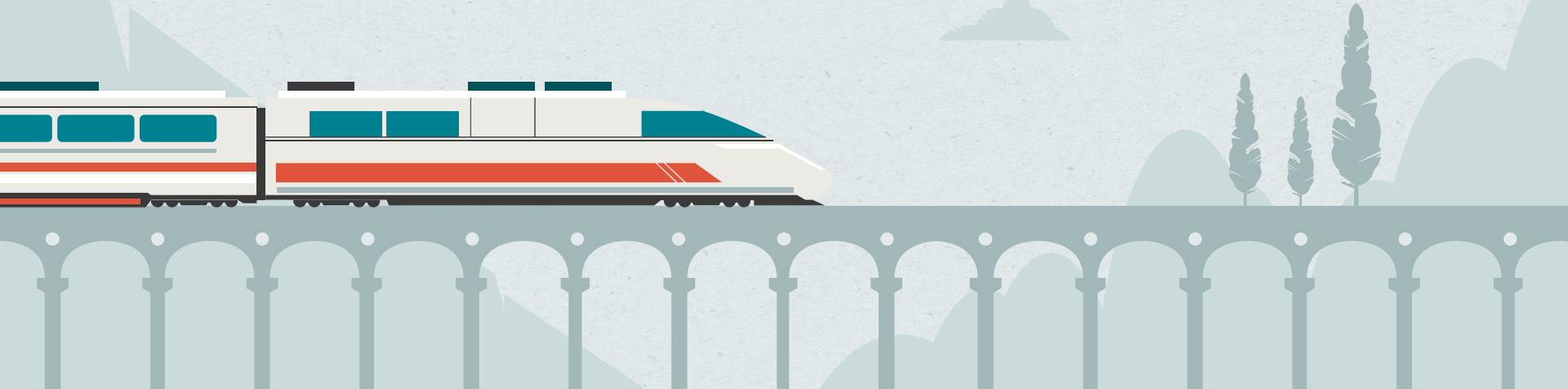
Optimizing Public Transit using MBTA Data

Group8
Aishwarya Rauthan, Atharva Lokhande, Lila Su, Neeharika Kamireddy

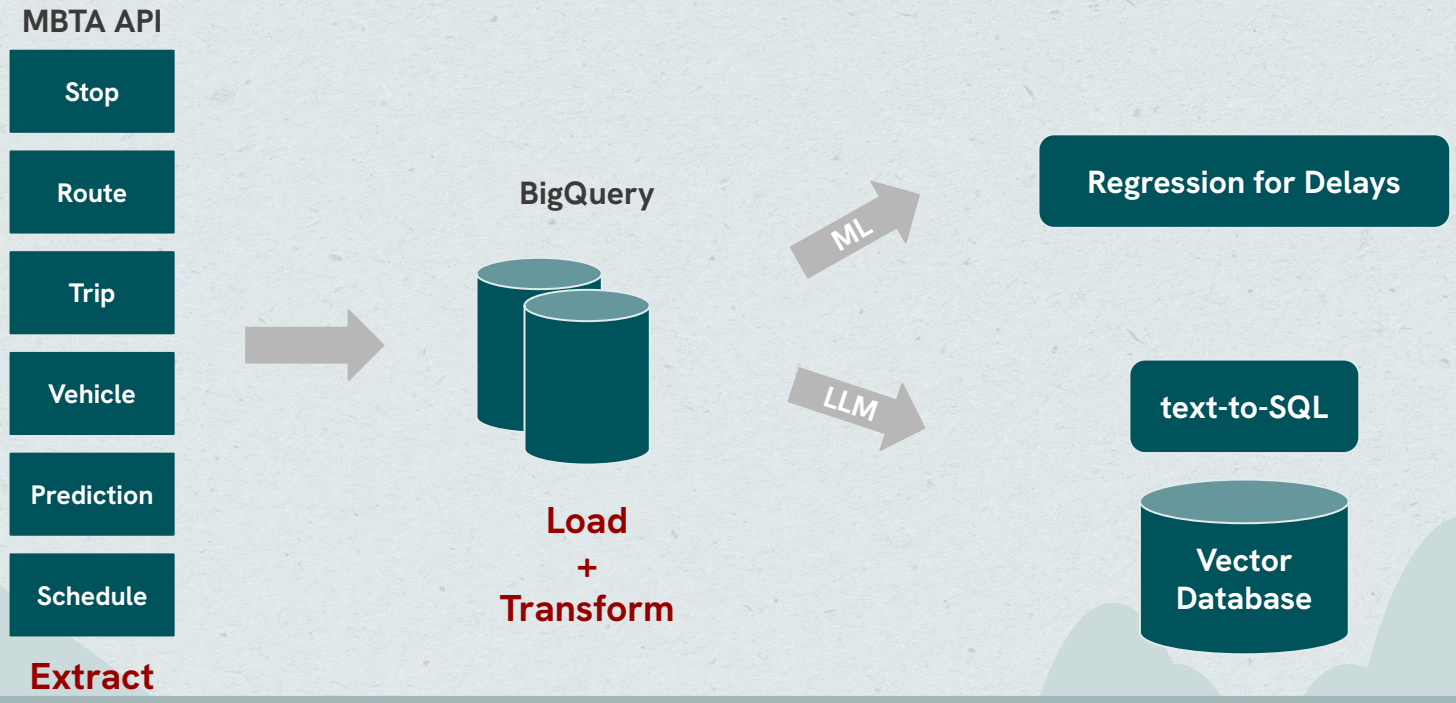


Recap

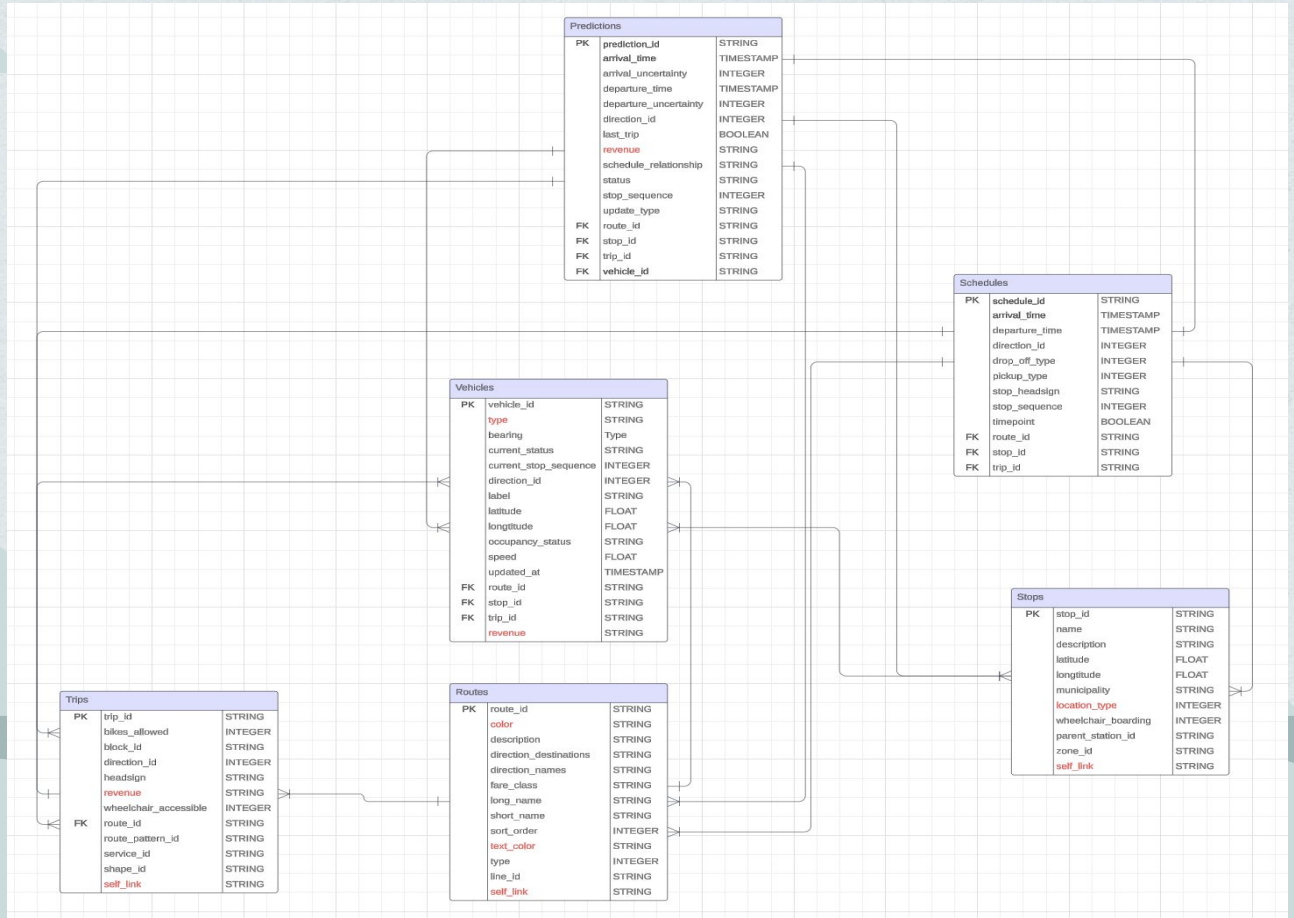
In the initial phase of the project, we extracted data from the MBTA API, which is updated weekly, and used this dataset to build visualizations on an Apache Superset dashboard. Following that, we implemented AutoML models to predict delays within the transit system.

For the current phase, we have transitioned to using new tools to further enhance our solution. We have also integrated Text-to-SQL using Streamlit, allowing users to query the database using natural language, and incorporated Generative AI to provide deeper insights and automation in analyzing MBTA data.

Recap

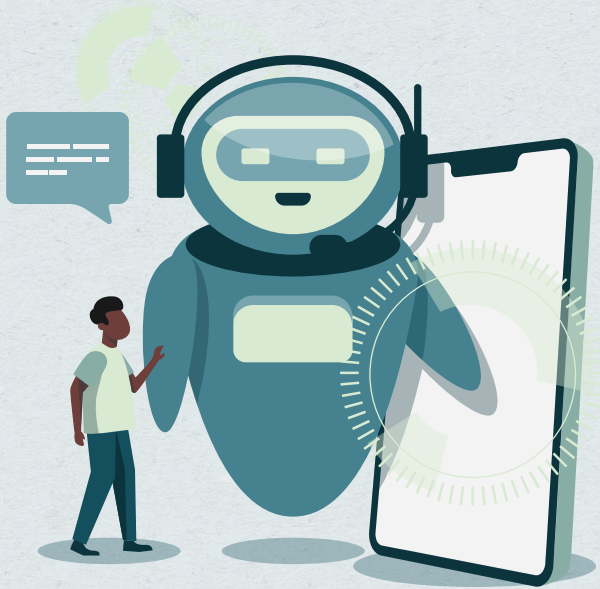

MBTA API
Stop
Route
Trip
Vehicle
Prediction
Schedule
Extract
BigQuery
Load
+
Transform
ML
LLM
Regression for Delays
text-to-SQL
Vector Database

ER-Diagram


Predictions
PK prediction_id STRING
arrival_time TIMESTAMP
arrival_uncertainty INTEGER
departure_time TIMESTAMP
departure_uncertainty INTEGER
direction_id INTEGER
last_trip BOOLEAN
revenue STRING
schedule_relationship STRING
status STRING
stop_sequence INTEGER
update_type STRING
FK route_id STRING
FK stop_id STRING
FK trip_id STRING
FK vehicle_id STRING
Schedules
PK schedule_id STRING
arrival_time TIMESTAMP
departure_time TIMESTAMP
direction_id INTEGER
drop_off_type INTEGER
pickup_type INTEGER
stop_headsign STRING
stop_sequence INTEGER
timepoint BOOLEAN
FK route_id STRING
FK stop_id STRING
FK trip_id STRING
Vehicles
PK vehicle_id STRING
type STRING
bearing Type
current_status STRING
current_stop_sequence INTEGER
direction_id INTEGER
label STRING
latitude FLOAT
longtitude FLOAT
occupancy_status STRING
speed FLOAT
updated_at TIMESTAMP
FK route_id STRING
FK stop_id STRING
FK trip_id STRING
revenue STRING
Stops
PK stop_id STRING
name STRING
description STRING
latitude FLOAT
longtitude FLOAT
municipality STRING
location_type INTEGER
wheelchair_boarding INTEGER
parent_station_id STRING
zone_id STRING
self_link STRING
Trips
PK trip_id STRING
bikes_allowed INTEGER
block_id STRING
direction_id INTEGER
headsign STRING
revenue STRING
wheelchair_accessible INTEGER
FK route_id STRING
route_pattern_id STRING
service_id STRING
shape_id STRING
self_link STRING
Routes
PK route_id STRING
color STRING
description STRING
direction_destinations STRING
direction_names STRING
fare_class STRING
long_name STRING
short_name STRING
sort_order INTEGER
text_color STRING
type INTEGER
line_id STRING
self_link STRING

Generative AI Implications

Retrieval Augmented Generation



Pinecone

To create a vector database for storing our data as embeddings by using sentence transformers

Sentence Transformers

Convert the data to embeddings for storing in Pinecone and converting user query to an embedding vector to query the Pinecone database

Gemini

Converted the user query and responses to a contextual prompt to be fed into Gemini for producing a conversational answer

Challenges

Inaccuracy

Data retrieved upon querying the user query embedding vector with the Pinecone database produced inaccurate answers occasionally.

Reasons

This inaccuracy could have stemmed from limitations on semantic understanding of the embeddings model or overlapping patterns in the data

Text-to-SQL

New Joined Table

A unified table by joining the prediction, schedule, route, and stop tables, including key features such as delay minutes, delay time, delay day, route, stop, and municipality for each delay record.

Generative Model

gemini-1.5-flash-002

Output

Users can input a prompt to generate SQL queries and view the resulting tables.

Interactive SQL Query Generator with GenAI

Enter your query prompt, and the system will generate and execute the corresponding SQL on BigQuery.

Enter your query prompt:

Return the municipality name and the average of delays associated for each route. Show the top 5 municipality with the largest average of municipality only.

Generate and Execute SQL Query

Generated SQL Query

```
SELECT municipality, AVG(delay) AS avg_delay FROM `ba882-team8-fall24.mbta_LLM.LLM
```

Query Results

	municipality	avg_delay
0	Ipswich	9.3
1	Rowley	9
2	Hamilton	8.3636
3	Lincoln	4.6286
4	Concord	4.1846

https://streamlit-rag-app-1029824820040.us-central1.run.app

Demo

- Return the route name and the average of delays associated for each route. Show the top 5 route with the largest average of route only.
- Return the stop name and the average of delays associated for each route. Show the top 10 stop with the largest average of stop on Monday only.
- Return the municipality name and the average of delays associated for each route. Show the bottom 5 municipality with the smallest average of municipality only.

Conclusion

- Utilized vector databases and generative AI for analyzing MBTA transit delays.
- Built a LLM-powered text-to-SQL tool, using Streamlit, making data accessible to all users and simplify complex interactions.
- Built a pipeline and applied predictive modeling to uncover critical delay patterns.
- Insights support MBTA in optimizing resources, improving reliability, and reducing commuter frustration.
- Recommendations align transit services with urban mobility demands for better customer satisfaction.

Future Steps & Challenges

Collaborate with other data

Enhance data granularity and integrate weather and event impacts into predictive models.

Feature expansion

Improve the LLM-powered text-to-SQL tool by adding capabilities such as delay and schedule visualizations for richer insights.

Thank You!