PROJECT A

Using Heuristics to Resolve Conflicting Attributes

OVERTURE MAPS FOUNDATION

By: Githika Annapureddy and Ryan Okimoto

Problem Statement

Project A: Discovering New Places

- Where are all the world’s open places?
- What Open signals can help us grow the data?
- What tools do we have to ensure data quality?
- How can we develop heuristics to ensure quality over time?

Sponsored by Overture Maps Foundation

OKRs

Objective: Develop a prototype framework that improves the consistency and quality of place data by verifying validity across the names of entities for the Overture dataset

Key Result 1: Identify and document 3 open datasets relevant to place data and extract a subset of entities for analysis.

Key Result 2: Define and implement rule based matching initially, then utilize LLMs and Vector Spaces, to analyze differences in names between overture and open datasets

Key Result 3: Create a web-based app, that uses a pipeline to work with many different datasets, to show name mismatches across Overture dataset and external datasets

Collaboration

- SparkGeo (Lauren, Greg, Gordon)
    - Meet once at the beginning of quarter
    - Email correspondence
- Limited Direct Knowledge
    - How the data is collected and sourced?
    - How confidence scores are calculated?
    - What infrastructure with our project already exists?
- Valuable Insights and Guidance

Approach and Methodology

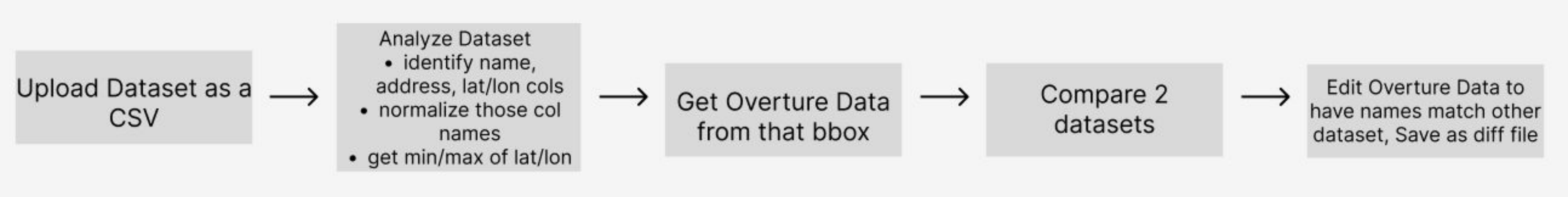
Choosing Datasets

NYC Restaurant Inspection
NYC Minority Owned Business Data
NYC Firefighter Response Data

Heuristics

Address Matching
Long/Lat Matching
Calculated similarity score on names using vectors

Our Pipeline



Upload Dataset as a CSV

→

Analyze Dataset

- identify name, address, lat/lon cols
- normalize those col names
- get min/max of lat/lon

→

Get Overture Data from that bbox

→

Compare 2 datasets

→

Edit Overture Data to have names match other dataset, Save as diff file

Analyze non-Overture Dataset

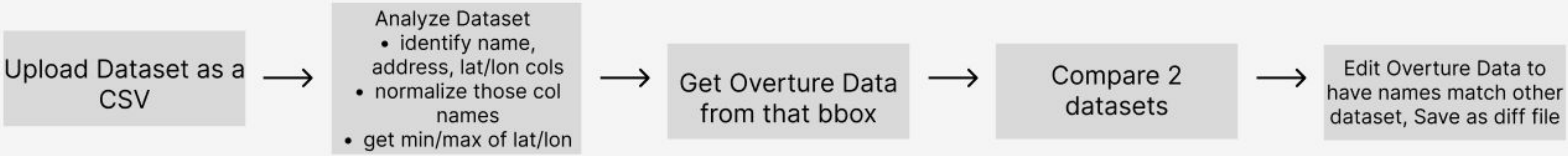
Use DSPY prompting and GPT Turbo 3.5 Model to

- Extract column names that contain name, address, latitude/longitude
- Generate a summary of the dataset
- Generate a summary for each column

Normalized name, address, lat/lon columns across datasets by creating new columns for each

Save this info in tmp/dataset_name/

- Dataset_name_edited.csv (dataset with normalized columns)
- Descriptions.json (GPT generated column descriptions)
- Summary.txt (GPT generated summary)



Upload Dataset as a CSV → Analyze Dataset

- identify name, address, lat/lon cols
- normalize those col names
- get min/max of lat/lon

→ Get Overture Data from that bbox → Compare 2 datasets → Edit Overture Data to have names match other dataset, Save as diff file

Fetch (from Overture)

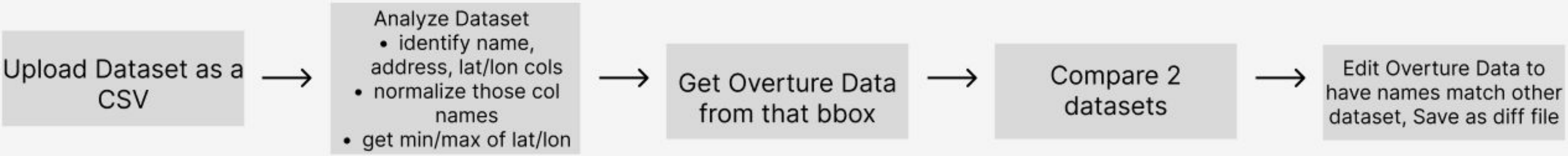
Find min/max of lat and lon of dataset.

Get Overture Data from that bbox as a Pandas DF

- Got rid of geometry feature of Geopandas GDF (exact coordinates) since we had the bbox of the specific location
- Check to make sure bbox is not larger than . Throw error message if so.

Save this info in tmp/dataset_name/

- Dataset_name_edited.csv (dataset with normalized columns)
- Descriptions.json (GPT generated column descriptions)
- Summary.txt (GPT generated summary)
- Overture_data.csv



Upload Dataset as a CSV

Analyze Dataset
- identify name, address, lat/lon cols
- normalize those col names
- get min/max of lat/lon

Get Overture Data from that bbox

Compare 2 datasets

Edit Overture Data to have names match other dataset, Save as diff file

Compare - Match

- Loads data into Pandas Dataframes

- Extracts key features (street name, street number, name, etc)
  - Using features generated during Fetch

- Create mapping between open dataset and overture dataset from parsed addresses

Compare - Differentiate

- Compute semantic similarity using sentence embeddings (SentenceTransformer/all-MiniLM-L6-v2)

- Calculate confidence score (0-1) from multiple heuristics (semantic similarity, coordinate difference, addressing)

Categorize

- 0.8 - 1.0 -> Safe
- 0.5 - 0.79 -> Unsure
- 0.0 - 0.49 -> Wrong

Compare - Verify

- Data in the Unsure Category is passed through second LLM verification
- Using context on the area presented determines if overture has valid or invalid data
- gpt-3.5-turbo

Open Source Contribution

Github Repo

Link to Demo

Results & Impact

Enables quick visualization of how Overture Data differs from user-provided datasets.

Uses lightweight models to keep latency and cost low for large datasets.

Designed to be compatible with a wide variety of dataset formats and schemas.

Empowers human supervisors and analysts to process large datasets with minimal manual intervention.

Next Steps/Reflection

Next Steps:

- Create more robust dataset analyzing process that handles a larger variety of datasets.
- Perform further checks to see if the changes should be made (web-scraping).

Reflection:

- Streamlit was a great tool to easily create UI from our code!
- Building the project step-by-step—from hardcoding to full automation—helped us stay organized and scale effectively.
- Testing different comparison strategies led us to favor vector-based comparison due to its flexibility and accuracy.
- This project deepened our appreciation for structured data workflows and thoughtful UI design.
- We're grateful to Overture Maps for providing open and accessible geospatial data—this project wouldn't have been possible without their contribution to the data science community.

THANKS
THANKS
QUESTIONS?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon** and infographics & images by **Freepik**

Initial Attempts

Selected Datasets

- Hardcoded relevant features (longitude, latitude, name, address, etc)
- Manually create data pipeline

Differentiating

- Manual data cleaning with regex and string formatting tools
- Rapidfuzz to generate similarity score