Executive Summary - NASA Interview Project

Senior Data Engineer interview
Steven DeLaurentis
2024-07-01

What

The goal of this project was to derive insight into the impact of a measure known as Social Vulnerability Index (from the CDC) on housing data (from Zillow)—both publicly available datasets—while at the same time showcasing the implementor's knowledge of common data engineering practices, exercising software development skills, making technical trade-offs given time and resource constraints, and communicating those decisions clearly.

Fig 1. Interactive Panel/Holoviz Dashboard

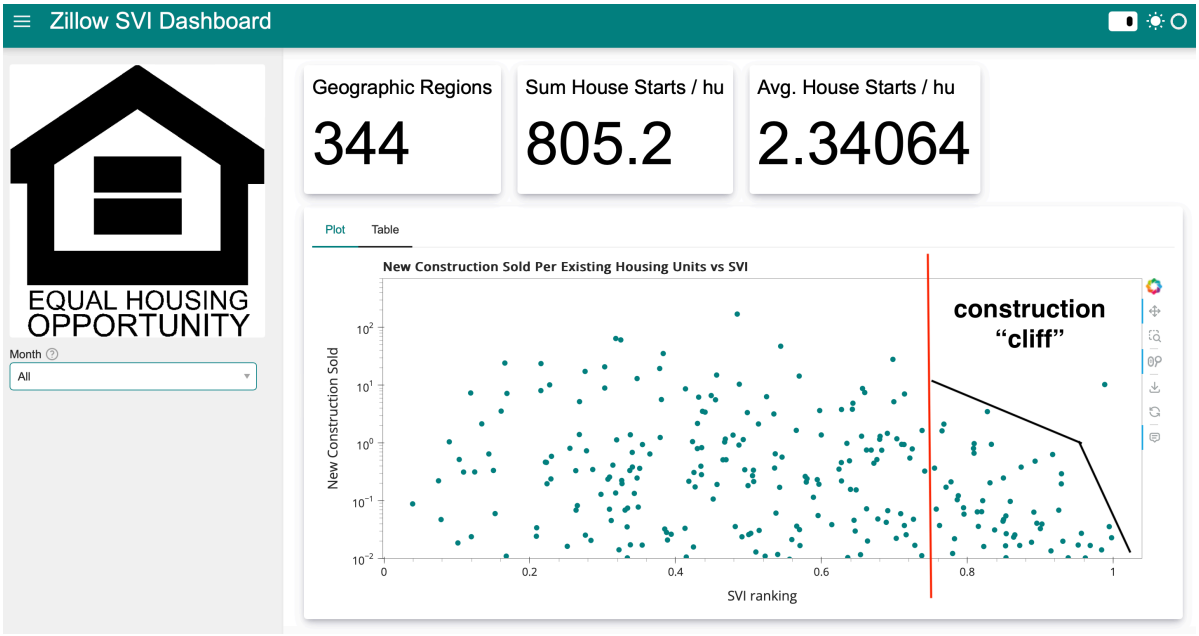
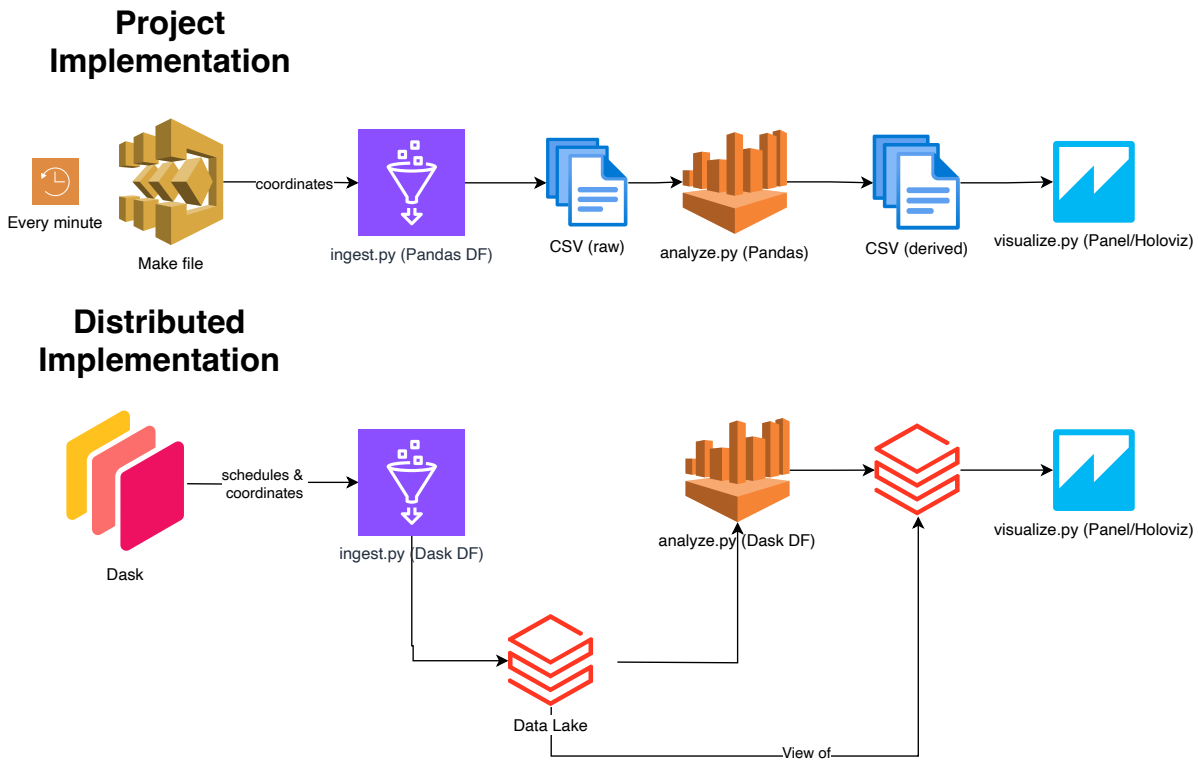


Fig. 1 demonstrates that the new construction sold divided by the estimated housing units (here, a derived measure called "housing starts rate") in all Zillow regions since 2018 drops off as SVI for the region increases. Above the threshold of 0.75, which the CDC defines as severely distressed, there is a clear indication that new construction sold over the lifetime of the dataset is much lower and does not match the distribution of those regions below the 0.75 threshold. Therefore, it can be inferred that in distressed regions new housing has not been constructed at a similar pace to more affluent and less distressed areas. This could lead to housing shortages and increased prices where such effects have worse impact due to lower income levels and denser household populations.

How

Fig 2. Current implementation and possible future distributed architecture



The pipeline is constructed as follows:

1) A Makefile serves as the coordinator of the pipeline. Because each make target downstream of the initial “ingest” step are linked, each subsequent step is executed linearly and atomically as a node in a directed acyclic graph. The docker-compose executor runs a command that triggers the pipeline every 60 seconds. This is to satisfy the automated scheduling requirement.

2) The ingest.py file connects to both the CDC and Zillow data sources and downloads them to raw data CSV files on disk. It also downloads the Zillow-provided cross-walk necessary to join the Zillow geographic metro regions with county-level CDC data.

3) The analyze.py file reads the raw csv files in Pandas DataFrames and joins the Zillow and CDC data across the Zillow metro region areas and the CDC data by county.

Then, it computes derived and aggregate fields used later in analysis. RPL_THEMES is renamed to SVI_INDEX: this column is the overall CDC measure of percentile ranking for each county across all vulnerability themes (like ethnicity, socio-economic, etc.).

Additionally, there is a function defined here which is calculated in real-time during the visualization step: the aggregation of the “housing starts rate”. Because the E_HU field in the CDC dataset contains estimated housing units per county, and Zillow provides estimated new housing sold per month in each metro region, the aggregation function “groups by” county into the Zillow region and calculates the time window’s housing starts/housing units measure

dynamically. This field is calculated so that the measure is a rate that can be compared across regions. Otherwise, larger regions would have a naturally higher value.

The joined table with its derived data columns is then saved as an intermediate CSV file.

4) The visualize.py file then reads the joined CSV file data and displays it in an interactive and reactive dashboard served by the Panel framework rendered by the Holoviz library. The "Month" dropdown is interactive and selects time windows including the derived sum "All" along with the original time windows provided from 2018 onward. There are plot and table tabs to explore the data as the user wishes. There are also some summary statistics displayed at the top: the total number of geographic regions, the sum of housing starts per total housing units in all the regions, and the average housing starts per housing units for each region for the selected time period.

Why

The above architecture was selected and implemented primarily driven by time constraints.

Each step in the pipeline was constructed incrementally and tested individually to verify completeness before tackling the next stage. While this ultimately limited the scope of the tools implemented here in the project, Figure 2's "Distributed Architecture" shows how this project could be extended to a more robust distributed execution engine if the size of the data exceeded the memory of one compute node or saving the CSV files became slow and took up too much disk space.

The Makefile which defines the structure of the DAG executed here in the project would be replaced by some external scheduler and executor engine. One choice would be Airflow, a common tool, but Dask is ultimately a more natural fit given that the Pandas DataFrames used here could be trivially extended into a Dask cluster backed by a PyArrow, for example.

The CSV files themselves would be replaced in the raw step by dumping the data into an appropriate data lake solution, whether that be SaaS like Databricks, a self-hosted or cloud column-oriented RDBMS like MariaDB or a document store like AWS S3/Backblaze B2. The analyze step could save its results into a Delta Lake view in Databricks or even keep the results stored in the Dask cluster. The Panel dashboard visualize.py then would only need to swap out its query code with an appropriate request to an API or database using SQLAlchemy or `requests`.

The biggest lacking feature here in the project might be the data analysis step. The preliminary insight derived above demands further analysis, if given more time. A scatter plot is a minimal start to exploring the data and deriving some insight visually. Ideally more advanced plots would be rendered—a 2-D Gaussian of housing starts vs SVI index, for example. Even better, a machine learning model like Logistic Regression or Random Forest could be trained to see if SVI index could predict Zillow housing outcomes or vice-versa. However, this deliverable chose to focus on coding a working end-to-end pipeline, wrapping it in a deployable docker-compose solution, and communicating the above.

Acknowledgements

As a personal note, I want to thank the whole panel for the time and consideration put into interviewing candidates. I appreciate that while the weekend project was personally a significant effort, there was reciprocal effort put into its preparation and the document which communicated what was expected. Having done similar interview projects before, I think

having a common and time-limited window for all candidates seems the fairest way to assign it. The thoughtful interview process overall gives me a positive impression of the NASA organization, and I would be remiss if I did not reiterate my personal excitement for the chance to join it.