Agent Quick Hack

MAX.IO +

Max Irwin
February, 2025

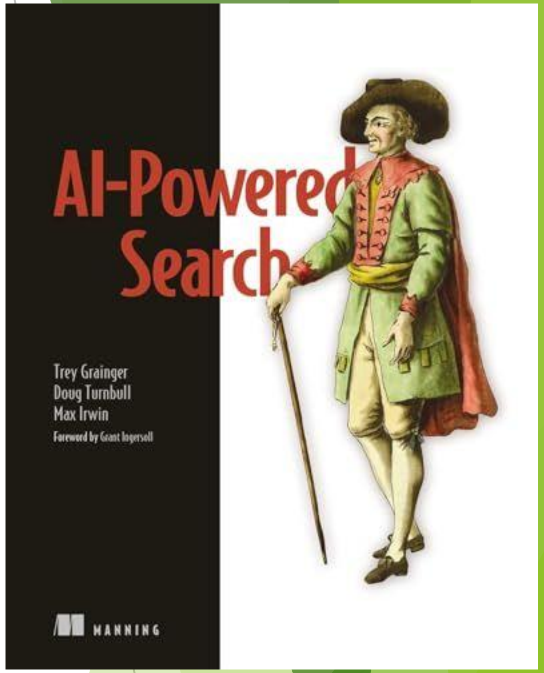
Hi! I'm Max Irwin.

- 25 years of experience in software delivery
- 14 years of experience in information retrieval
- 10 years of experience in AI

Founder/CEO of MAX.IO LLC, an AI company

Founder of the Flower City AI Conference

Contributing author of "AI-Powered Search"



Contributing Author

I have a deep understanding of problems and solutions for knowledge discovery and information retrieval

Email: max@max.io | LinkedIn: https://www.linkedin.com/in/maxirwin/

MAX.IO

Team Questions

- Does everyone have a github account?
- Who is good with FastAPI or Flask?
- Who is good with Pandas/Parquet?
- Who is good with SQL/SQLite?
- Who is good with Prompts?
- Who is good with OpenAI?

MAX.IO

Our Dev Process

https://github.com/maxdotio/agent-quick-hack

- Fork the Repository
- Clone the Fork Locally
- Make Changes and Commit
- Push Changes to GitHub
- Create a Pull Request*
- Get the Pull Request Merged
- Fetch any changes from Upstream

When you open a new PR, shout!

MAX.IO

Let’s Build an Agent! Quick!

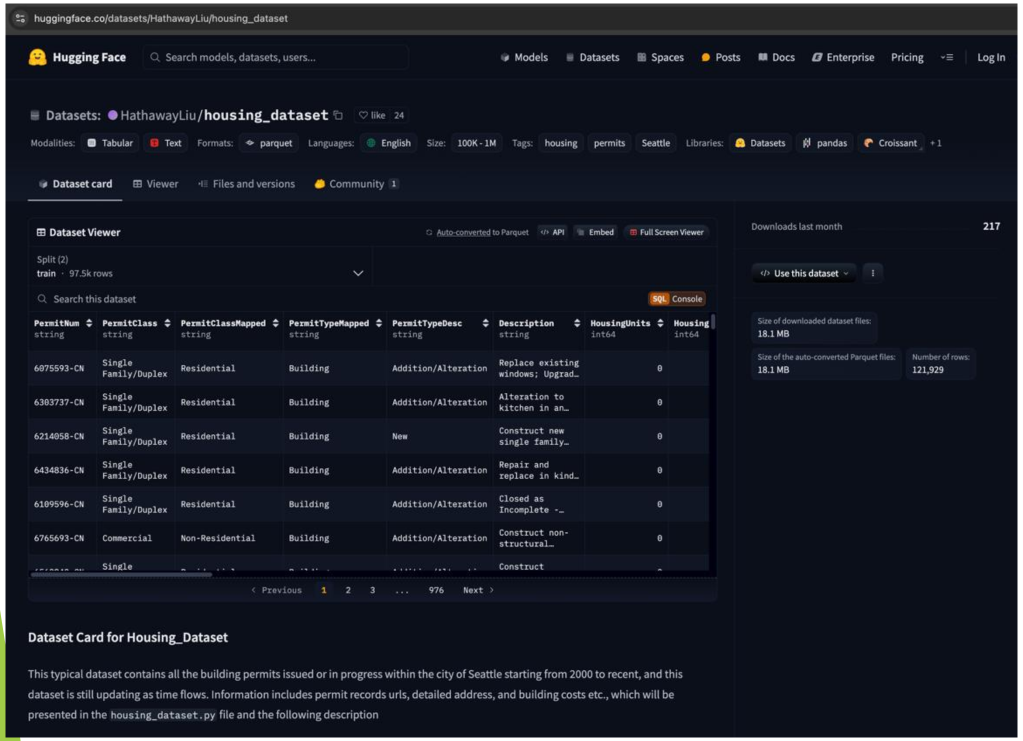
- Download a dataset
- Export the dataset to a sqlite table
- Make a FastAPI or Flask app
- Accept a user request from a webpage
- Send the user request with a prompt and some context to an LLM
- Have the LLM generate some SQL and a post-processor
- Execute the generated SQL on our table
- Execute the post-processor on the resultset
- Return the results to the webpage
- Iterate (if we have time)

Can we do this in an hour???

MAX.IO

Dataset

https://huggingface.co/datasets/HathawayLiu/housing_dataset



huggingface.co/datasets/HathawayLiu/housing_dataset

Hugging Face | Search models, datasets, users... | Models | Datasets | Spaces | Posts | Docs | Enterprise | Pricing | Log In

Datasets: HathawayLiu/housing_dataset | like 24

Modalities: Tabular, Text | Formats: parquet | Languages: English | Size: 100K - 1M | Tags: housing, permits, Seattle | Libraries: Datasets, pandas, Croissant, +1

Dataset card | Viewer | Files and versions | Community 1

Dataset Viewer | Auto-converted to Parquet | API | Embed | Full Screen Viewer

Split (2)
train · 97.5k rows

Search this dataset | SQL Console

PermitNum string	PermitClass string	PermitClassMapped string	PermitTypeMapped string	PermitTypeDesc string	Description string	HousingUnits int64	Housing int64
6075593-CN	Single Family/Duplex	Residential	Building	Addition/Alteration	Replace existing windows; Upgrad…	0	
6303737-CN	Single Family/Duplex	Residential	Building	Addition/Alteration	Alteration to kitchen in an…	0	
6214058-CN	Single Family/Duplex	Residential	Building	New	Construct new single family…	0	
6434836-CN	Single Family/Duplex	Residential	Building	Addition/Alteration	Repair and replace in kind…	0	
6109596-CN	Single Family/Duplex	Residential	Building	Addition/Alteration	Closed as Incomplete -…	0	
6765693-CN	Commercial	Non-Residential	Building	Addition/Alteration	Construct non-structural…	0	
[illegible]	Single	[illegible]	[illegible]	[illegible]	Construct	[illegible]	

‹ Previous 1 2 3 ... 976 Next ›

Downloads last month 217

Use this dataset

Size of downloaded dataset files: 18.1 MB

Size of the auto-converted Parquet files: 18.1 MB

Number of rows: 121,929

Dataset Card for Housing_Dataset

This typical dataset contains all the building permits issued or in progress within the city of Seattle starting from 2000 to recent, and this dataset is still updating as time flows. Information includes permit records urls, detailed address, and building costs etc., which will be presented in the `housing_dataset.py` file and the following description

MAX.IO

Web App

- We need a basic Python app that we can view in the browser
- Jupyter notebooks are tired. We want something real!
- Use FastAPI or Flask
- Two routes:
  - GET `` `/` `` → returns an HTML form with a textbox
  - POST `` `/report` `` → takes the form input and displays the output
- This will also be where we put our Python modules

SQL

- We need to create a SQL table to hold our dataset.
- We’re using sqlite3 because it’s a builtin and it’s easy to use.
- The SQL table should be documented with a data dictionary.
- Write a command-line method to use the *datasets* module to download the dataset and populate the SQL table
- Write a module that will be used by the Web App to execute SELECTs

MAX.IO

Prompts

- We need to make two prompts that will generate SQL and a Post-Processor
- The SQL should be a valid SQLite SELECT statement.
  - WARNING: We will run the LLM output on our SQLite database!
- The Post-processor should be valid Python to generate HTML.
  - WARNING: We will run the Python code with exec() using the resultset!
- Consider writing more prompts to check the safety of the output

MAX.IO

LLM Integration

- Using our prompts, we need a way to call OpenAI's API.
- Make one or more methods that accepts a prompt and returns the output
  - I'll email you a temporary OPENAI_API_KEY
  - Use python-dotenv to manage the key (don't save it in github!)
- Consider building in some checks using the safety prompt!
- Consider using structured output with chain-of-thought!

MAX.IO

Controller

- We need a module that runs everything!
  - accepts a natural language query from user input
  - calls OpenAI with the SQL prompt
  - gets the SQL output
  - executes the SQL command against the database
  - calls OpenAI with the Post-processor prompt
  - executes the post-processor code with the resultset
  - returns the HTML output

MAX.IO

Let’s Goooooooo!