Dynamic Property Analysis and Search

Made By :
Akaspreet

Submitted To :
Danish Bansal

Project Overview

Problem Statement

The task is to implement a Gen AI-based filtering system to efficiently handle and analyze property data provided in a JSON format. The dataset includes columns such as address, latitude, longitude, bedrooms, bathrooms, property type, area, postcode, sector, etc. The system should support advanced NLP capabilities for analysis and property recommendations. For instance, it should be able to analyze queries like "Tell me the change in property type over the years" and recommend properties based on specific criteria like "I want a 3BHK house in a posh area with good online facilities."

Application Structure

1. `gemini_handler.py`

- **Purpose:** Handles interactions with the Gemini API to interpret user queries and generate Python code for data analysis.
- **Key Functions:**
  - `interpret_query(self, query)`: Sends a prompt to the Gemini API to interpret a user query and extract parameters.
  - `generate_and_execute_analysis(self, query, parameters, df)`: Generates Python code for data analysis based on the interpreted query and executes it on the provided DataFrame.

2. `llm_handler.py`

- **Purpose:** Manages encoding and searching functionalities using SentenceTransformers.
- **Key Functions:**
  - `encode_query(self, query)`: Encodes a user query into embeddings.
  - `encode_properties(self, df)`: Encodes property data into embeddings.
  - `search_properties(self, query_embedding, property_embeddings, top_k=5)`: Searches for the top-k most similar properties based on query embeddings.

3. `llm_model.py`

- **Purpose:** Initializes and handles operations related to the language model from Hugging Face's transformers library.
- **Key Functions:**
  - `initialize_llm()`: Initializes the tokenizer and model.
  - `encode_query(tokenizer, model, query)`: Encodes a query using the model.
  - `search_properties(df, query_embedding, top_k=5)`: Placeholder for implementing similarity search logic.

4. `search_engine.py`

- **Purpose:** Orchestrates the property search process using embeddings.
- **Key Functions:**
  - `__init__(self, df, llm_handler)`: Initializes the search engine with property data and the LLM handler.
  - `search(self, query, parameters, top_k=5)`: Combines the user query and parameters to perform a property search.

5. `utils.py`

- **Purpose:** Provides utility functions for executing generated analysis code.
- **Key Functions:**
  - `execute_analysis_code(code, df)`: Executes the generated analysis code and saves the result.

6. `app.py`

- **Purpose:** Main Streamlit application for user interaction, data loading, and handling property analysis and search.
- **Key Components:**
  - Data loading and preprocessing using `load_and_preprocess_data()`.
  - Initializing handlers and search engine.
  - Input field for user queries.
  - Displaying query interpretation, search results, analysis results, and generated code.

7. `data_loader.py`

- **Purpose:** Loads and preprocesses the property data from a JSON file.
- **Key Functions:**
  - `load_data(file_path)`: Loads data from a JSON file.
  - `preprocess_data(df)`: Preprocesses the data by converting date columns, handling missing values, and parsing the lease term.

8. `gemini_api.py`

- **Purpose:** Initializes and interacts with the Gemini API.
- **Key Functions:**
  - `initialize_gemini(api_key)`: Configures the Gemini API with the provided API key.
  - `generate_code(model, prompt)`: Generates code using the Gemini model based on the provided prompt.

How It Solves the Problem Statement

Approach Overview

1. Data Handling and Preprocessing

- **File:** `data_loader.py`
- **Functionality:**
  - Load data from a JSON file.
  - Preprocess the data by converting date columns, handling missing values, and parsing the lease term.
- **Models/Tools Used:** Pandas for data manipulation

2. Interpreting User Queries with Gemini API

- **File:** `gemini_handler.py`
- **Functionality:**
  - Send user queries to the Gemini API to interpret the intent and extract parameters.
  - Generate Python code for data analysis based on interpreted queries.
- **Models/Tools Used:** Gemini API for natural language understanding and code generation

3. Encoding and Searching Properties with SentenceTransformers

- **File:** `llm_handler.py`
- **Functionality:**
  - Encode user queries and property data into embeddings using SentenceTransformers.
  - Search for similar properties based on query embeddings.
- **Models/Tools Used:** SentenceTransformers for embedding generation and similarity search.

4. **Executing Generated Analysis Code**

- **File:** `utils.py`
- **Functionality:**
    - Execute the Python code generated by the Gemini API for data analysis.
- **Models/Tools Used:** Built-in Python functions for code execution.

5. **Orchestrating the Search Process**

- **File:** `search_engine.py`
- **Functionality:**
    - Combine user queries and parameters to perform a property search.
- **Models/Tools Used:** Custom logic using encoded embeddings for similarity search.

6. **Streamlit Application for User Interaction**

- **File:** `app.py`
- **Functionality:**
    - Provide an interactive user interface for data loading, query input, and displaying results.
- **Models/Tools Used:** Streamlit for building the web application.

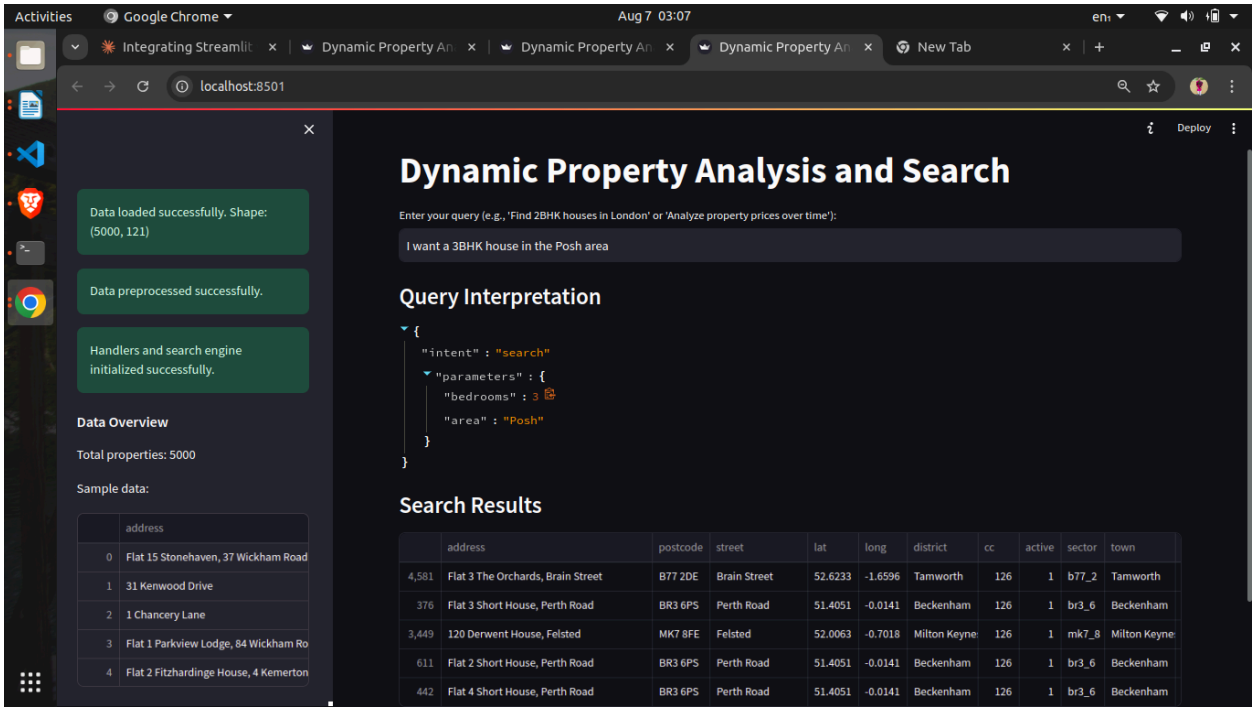
Models Used and Their Purpose

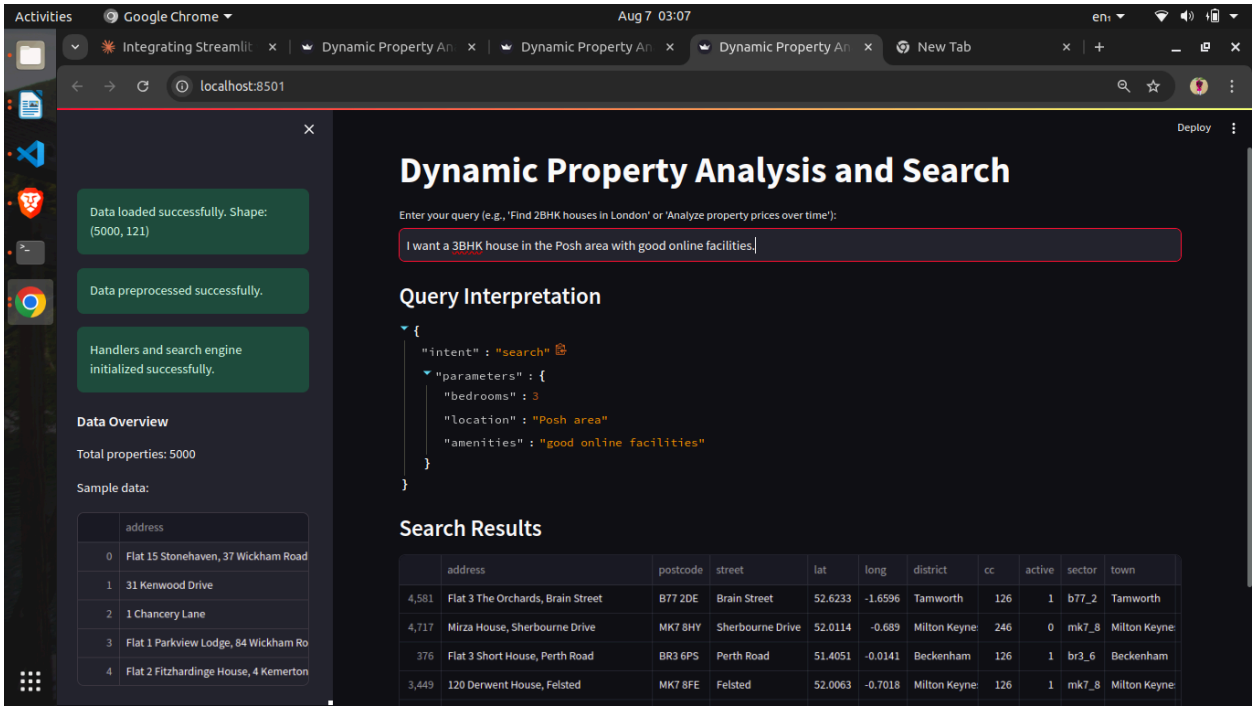
1. **Gemini API**
    - **Purpose:** Interprets natural language queries and generates Python code for data analysis.
    - **Why:** Provides a powerful way to understand complex user queries and automate code generation for analysis tasks.
2. **SentenceTransformers (all-MiniLM-L6-v2)**
    - **Purpose:** Encodes user queries and property data into embeddings for similarity search.
    - **Why:** Efficient for generating high-quality embeddings suitable for similarity search, ensuring relevant property recommendations.

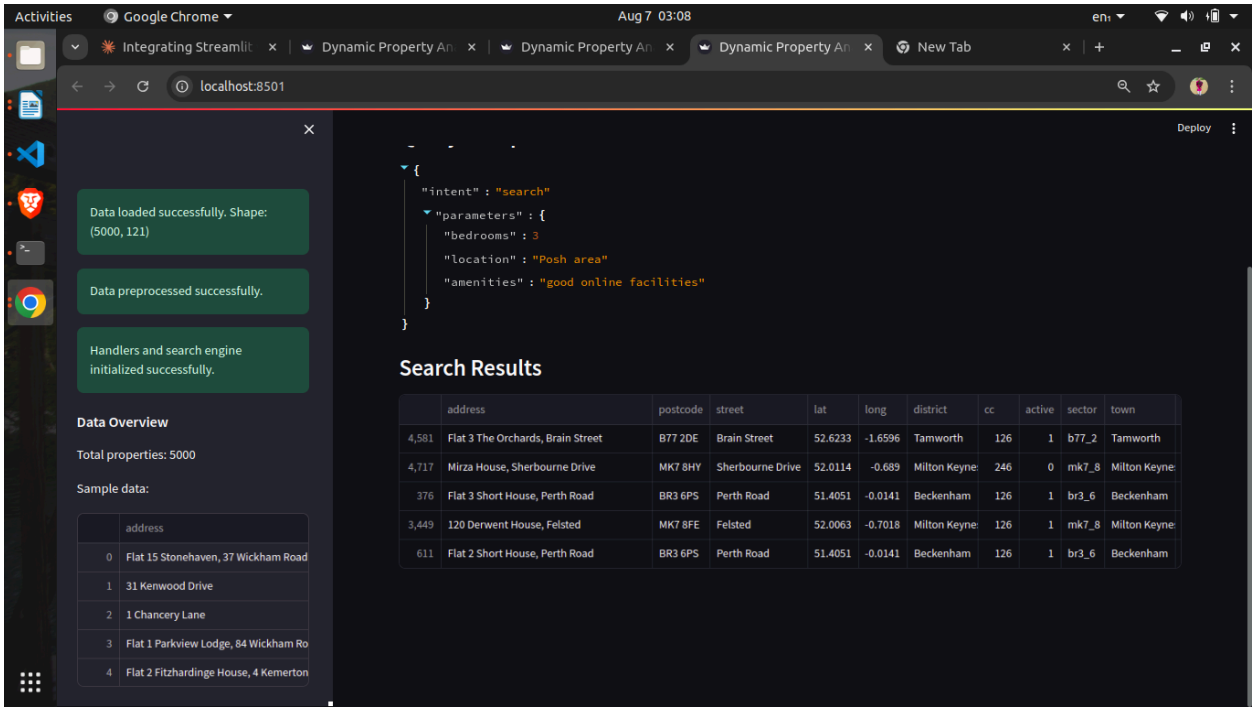
Overall Workflow

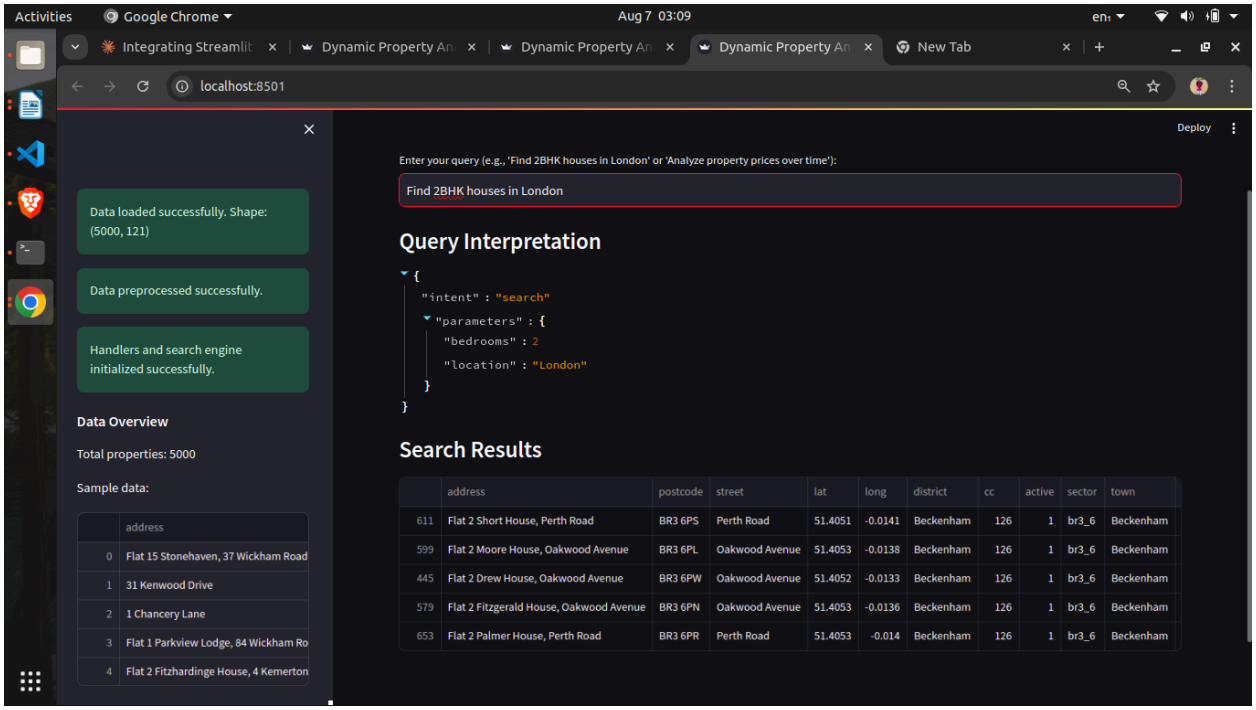
1. **Data Loading and Preprocessing:**
    - Load and preprocess the property data from a JSON file.
2. **Query Interpretation and Code Generation:**
    - Use Gemini API to interpret user queries and generate code for analysis.
3. **Embedding and Similarity Search:**
    - Encode queries and property data using SentenceTransformers.
    - Perform similarity search to find relevant properties based on user queries.
4. **Executing Analysis and Displaying Results:**
    - Execute generated analysis code and display results using Streamlit.
    - Provide an interactive UI for users to input queries and view recommendations or analysis results.

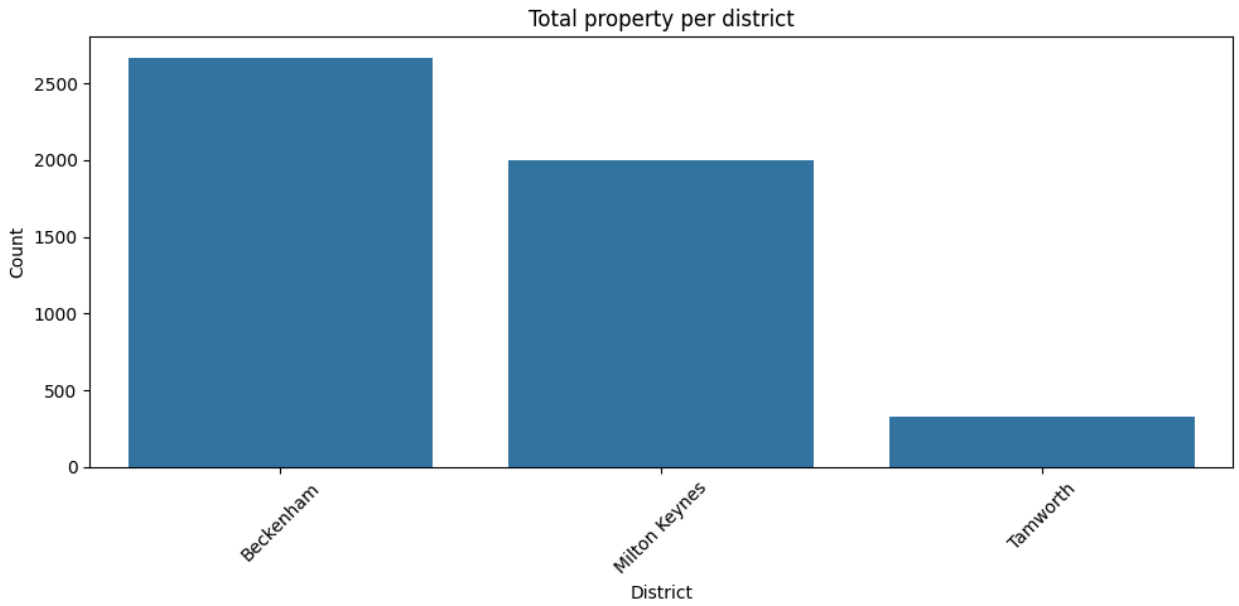
Results



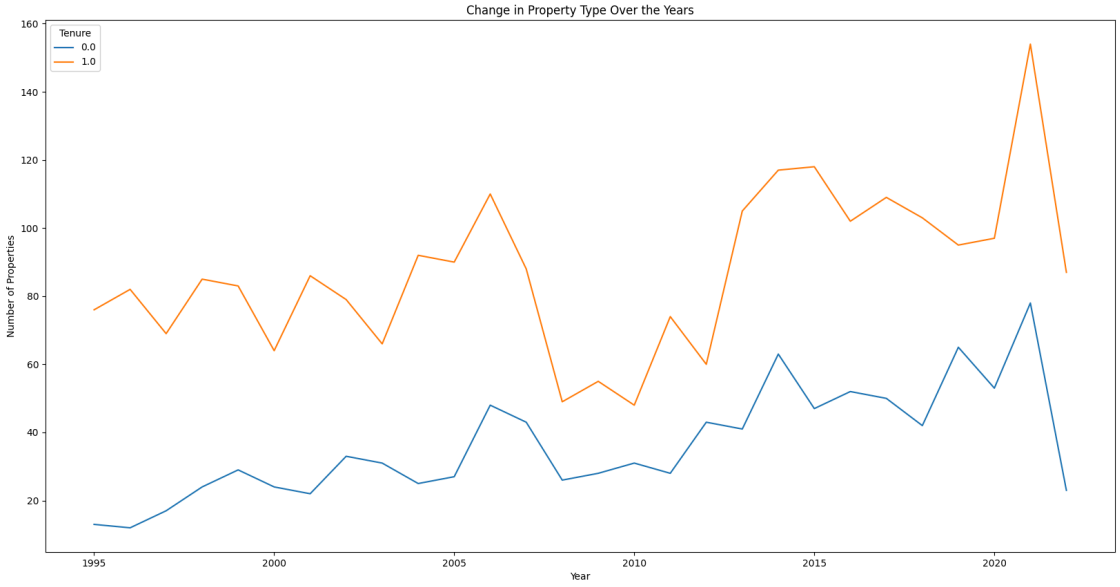

Activities
Google Chrome
Aug 7 03:07
Integrating Streamlit
Dynamic Property An
Dynamic Property An
Dynamic Property An
New Tab
localhost:8501
Deploy
Data loaded successfully. Shape: (5000, 121)
Data preprocessed successfully.
Handlers and search engine initialized successfully.
Data Overview
Total properties: 5000
Sample data:
address
0 Flat 15 Stonehaven, 37 Wickham Road
1 31 Kenwood Drive
2 1 Chancery Lane
3 Flat 1 Parkview Lodge, 84 Wickham Ro
4 Flat 2 Fitzhardinge House, 4 Kemerton
Dynamic Property Analysis and Search
Enter your query (e.g., 'Find 2BHK houses in London' or 'Analyze property prices over time'):
I want a 3BHK house in the Posh area with good online facilities.
Query Interpretation
{
"intent" : "search"
"parameters" : {
"bedrooms" : 3
"location" : "Posh area"
"amenities" : "good online facilities"
}
}
Search Results
address postcode street lat long district cc active sector town
4,581 Flat 3 The Orchards, Brain Street B77 2DE Brain Street 52.6233 -1.6596 Tamworth 126 1 b77_2 Tamworth
4,717 Mirza House, Sherbourne Drive MK7 8HY Sherbourne Drive 52.0114 -0.689 Milton Keyne 246 0 mk7_8 Milton Keyne
376 Flat 3 Short House, Perth Road BR3 6PS Perth Road 51.4051 -0.0141 Beckenham 126 1 br3_6 Beckenham
3,449 120 Derwent House, Felsted MK7 8FE Felsted 52.0063 -0.7018 Milton Keyne 126 1 mk7_8 Milton Keyne


Activities
Google Chrome
Aug 7 03:08
Integrating Streamlit
Dynamic Property An
Dynamic Property An
Dynamic Property An
New Tab
localhost:8501
Deploy
Data loaded successfully. Shape: (5000, 121)
Data preprocessed successfully.
Handlers and search engine initialized successfully.
Data Overview
Total properties: 5000
Sample data:
address
0 Flat 15 Stonehaven, 37 Wickham Road
1 31 Kenwood Drive
2 1 Chancery Lane
3 Flat 1 Parkview Lodge, 84 Wickham Ro
4 Flat 2 Fitzhardinge House, 4 Kemerton
{
"intent" : "search"
"parameters" : {
"bedrooms" : 3
"location" : "Posh area"
"amenities" : "good online facilities"
}
}
Search Results
address postcode street lat long district cc active sector town
4,581 Flat 3 The Orchards, Brain Street B77 2DE Brain Street 52.6233 -1.6596 Tamworth 126 1 b77_2 Tamworth
4,717 Mirza House, Sherbourne Drive MK7 8HY Sherbourne Drive 52.0114 -0.689 Milton Keyne 246 0 mk7_8 Milton Keyne
376 Flat 3 Short House, Perth Road BR3 6PS Perth Road 51.4051 -0.0141 Beckenham 126 1 br3_6 Beckenham
3,449 120 Derwent House, Felsted MK7 8FE Felsted 52.0063 -0.7018 Milton Keyne 126 1 mk7_8 Milton Keyne
611 Flat 2 Short House, Perth Road BR3 6PS Perth Road 51.4051 -0.0141 Beckenham 126 1 br3_6 Beckenham


Activities
Google Chrome
Aug 7 03:09
Integrating Streamlit
Dynamic Property An
Dynamic Property An
Dynamic Property An
New Tab
localhost:8501
Deploy
Data loaded successfully. Shape: (5000, 121)
Data preprocessed successfully.
Handlers and search engine initialized successfully.
Data Overview
Total properties: 5000
Sample data:
address
0 Flat 15 Stonehaven, 37 Wickham Road
1 31 Kenwood Drive
2 1 Chancery Lane
3 Flat 1 Parkview Lodge, 84 Wickham Ro
4 Flat 2 Fitzhardinge House, 4 Kemerton
Enter your query (e.g., 'Find 2BHK houses in London' or 'Analyze property prices over time'):
Find 2BHK houses in London
Query Interpretation
{
"intent" : "search"
"parameters" : {
"bedrooms" : 2
"location" : "London"
}
}
Search Results
address postcode street lat long district cc active sector town
611 Flat 2 Short House, Perth Road BR3 6PS Perth Road 51.4051 -0.0141 Beckenham 126 1 br3_6 Beckenham
599 Flat 2 Moore House, Oakwood Avenue BR3 6PL Oakwood Avenue 51.4053 -0.0138 Beckenham 126 1 br3_6 Beckenham
445 Flat 2 Drew House, Oakwood Avenue BR3 6PW Oakwood Avenue 51.4052 -0.0133 Beckenham 126 1 br3_6 Beckenham
579 Flat 2 Fitzgerald House, Oakwood Avenue BR3 6PN Oakwood Avenue 51.4053 -0.0136 Beckenham 126 1 br3_6 Beckenham
653 Flat 2 Palmer House, Perth Road BR3 6PR Perth Road 51.4053 -0.014 Beckenham 126 1 br3_6 Beckenham


Total property per district
Count
2500
2000
1500
1000
500
0
Beckenham
Milton Keynes
Tamworth
District


Property Prices Over Time
Average Latest Sale Price
500000
450000
400000
350000
300000
250000
200000
150000
100000
1995
2000
2005
2010
2015
2020
Year


Change in Property Type Over the Years
Tenure
0.0
1.0
Number of Properties
160
140
120
100
80
60
40
20
1995
2000
2005
2010
2015
2020
Year


Property Prices Over Time
1e6
2.0
1.5
1.0
0.5
0.0
Latest Sale Price
1995
2000
2005
2010
2015
2020
Year