

GeoProp Analyzer – Real Estate Analysis Tool

Overview

An interactive dashboard for analyzing real estate properties with geospatial visualization and investment scoring.

Features

- Interactive map with property locations
- Price per square foot visualization
- XGBoost-based investment scoring
- 12-month price projections
- Filters for:
 - Zipcode
 - Bedrooms/Bathrooms/Floors
 - Square footage range
 - Price range

Installation

Prerequisites

- Python 3.7 or higher
- pip package manager

Required Libraries

```
pip install dash plotly pandas scikit-learn xgboost joblib
```

Detailed Dependency List

Package	Version	Purpose
dash	>=2.0.0	Web framework
plotly	>=5.0.0	Interactive visualizations
pandas	>=1.0.0	Data processing
scikit-learn	>=1.0.0	Model support
xgboost	>=1.5.0	ML model
joblib	>=1.0.0	Model serialization

Installation Steps

1. Clone the repository:

```
git clone https://github.com/yourusername/GeoProp-Analyzer.git
```

```
cd GeoProp-Analyzer
```
2. Create and activate virtual environment:

```
python -m venv .venv
```

```
# On Windows:
```

```
.venv\Scripts\activate
```

```
# On macOS/Linux:
```

```
source .venv/bin/activate
```
3. Install dependencies:

```
pip install -r requirements.txt
```

File Structure

GeoProp-Analyzer/

└─ data/	
└─ filtered_real_estate_data.csv	# Sample dataset
└─ gui/	
└─ main_window.py	# Dashboard
implementation	
└─ xgboost_investment_model.pkl	# Trained model
└─ models/	
└─ data_processor.py	# Data handling
└─ real_estate.py	# Property class
└─ analysis/	
└─ adrs_ext.py	# Address resolution
using geocoding API	
└─ logistic_model_test.py	# Logistic regression
testing	
└─ model_comparison.py	# Model comparison via
scoring	
└─ model_training.py	# Random forest training
└─ xgboost_model_test.py	# XGBoost training and
evaluation	
└─ main.py	# Entry point
└─ README.md	# Documentation

Supplementary Files Overview

adrs_ext.py

Purpose: Uses the Google Maps Geocoding API to retrieve formatted addresses from latitude and longitude coordinates.

Key Components:

- get_address(lat, lng): Makes an HTTP request to the API and returns the formatted address.
- Requires an API key (API_KEY) and uses requests.

logistic_model_test.py

Purpose: Builds and evaluates a logistic regression model using real estate data.

Key Components:

- Loads data from filtered_real_estate_data.csv
- Creates features like price_per_sqft and affordability_score
- Splits data, trains a logistic regression model, and evaluates it with metrics and confusion matrix

model_comparison.py

Purpose: Compares multiple models on real estate data using investment scoring.

Key Components:

- Feature engineering for affordability_score, quality_score, and a combined investment_score
- Evaluation likely includes visual comparison (e.g., matplotlib, seaborn)

model_training.py

Purpose: Trains a Random Forest Regressor on real estate data.

Key Components:

- Loads and preprocesses the dataset

- Feature engineering for investment scoring
- Trains a RandomForestRegressor and calculates performance metrics like R^2 and RMSE
- Saves the model using joblib

xgboost_model_test.py

Purpose: Trains and evaluates an XGBoost model on the dataset.

Key Components:

- Uses xgboost.XGBRegressor
- Feature engineering and model evaluation (R^2 , RMSE)
- Includes feature importance plotting with plot_importance

Running the Application

1. Start the server:

python main.py

2. Access the dashboard:

Open your web browser

Navigate to: <http://localhost:8050/>

Troubleshooting

If the page doesn't load:

1. Verify the server is running (you should see output in your terminal)
2. Check your firewall settings

Notes

- The application requires the model file (xgboost_investment_model.pkl) to be present in the gui folder
- Sample data should be placed in the data folder
- First run may take longer as it loads the ML model