Real Estate Price Prediction System

Group Members

210316016 – Berke Alpaslan

210316076 – Mert Doğan Aygün

230316002 – Çağrı Ertunç

210315090 – Elay Yusifli

Project Purpose

This project develops an automated real estate price prediction system that combines web scraping with machine learning. The system collects property listings from emlakjet.com, processes the data, and uses a Random Forest model to predict property prices based on area (m²). The main goal is to provide accurate price estimations for properties in the Turkish real estate market, helping users make informed decisions.

System Architecture

1. Data Collection Module

- **Technology**: Selenium WebDriver with Chrome
- **Process**: Automatically navigates through emlakjet.com pages, extracting property location, area, and price
- **Storage**: CSV format with automatic updates every 20 minutes
- **Validation**: Only accepts emlakjet.com URLs for data integrity

2. Data Processing Pipeline

- **Cleaning**: Implements IQR (Interquartile Range) method to remove outliers
- **Filtering**: Removes properties outside logical ranges (20-1000 m², 1K-10M TL)
- **Standardization**: Uses StandardScaler for feature normalization

3. Machine Learning Model

- **Algorithm**: Random Forest Regressor (100 trees)
- **Features**: Property area (m²) as the primary predictor
- **Training**: Automatic retraining when new data is collected (minimum 10 samples)
- **Performance**: Achieves R^2 scores between 0.70-0.85

4. Web Interface

- **Framework**: Flask with Bootstrap 5
- **Real-time Statistics**: Displays average price, area, and price per m²
- **Interactive Prediction**: Users input area to get instant price estimates
- **Visualization**: Scatter plot showing price-area relationship with trend line

Key Features

1. **Automated Data Collection**: One-click scraping from any emlakjet.com listing page
2. **Smart Model Training**: Automatically trains when sufficient new data is available
3. **Confidence Intervals**: Provides price ranges based on model uncertainty
4. **Model Transparency**: "Model Info" button shows algorithm details and training statistics
5. **Responsive Design**: Modern UI that works on all devices

Technical Stack

- **Backend**: Python 3.x, Flask
- **ML Libraries**: scikit-learn, pandas, numpy
- **Visualization**: matplotlib
- **Web Scraping**: Selenium
- **Frontend**: HTML5, Bootstrap 5, JavaScript
- **Data Storage**: CSV files, joblib for model persistence

API Endpoints

- GET /: Main interface
- POST /: Data collection trigger
- GET /api/stats: Current statistics
- GET /api/chart: Price-area scatter plot
- POST /api/predict: Price prediction
- GET /api/model-info: Model information