**Dataset Description**

The housing dataset contains property information with the following features:

* **price**: Target variable - the price of the property
* **sqft**: Square footage of the property
* **bedrooms**: Number of bedrooms
* **bathrooms**: Number of bathrooms
* **lot\_size**: Size of the property lot
* **year\_built**: Year the property was constructed
* **stories**: Number of stories in the building
* **neighborhood**: Location category (Downtown, Suburbs, Lakefront)
* **garage**: Number of garage spaces
* **condition**: Property condition (Excellent, Good, Fair)

**Ml Model Training Steps**

1. **Data Preparation:**

* Loading data from CSV files
* Handling missing values in numerical columns with mean imputation
* Encoding categorical features (neighborhood, condition) using one-hot encoding

1. **Model Development:**

* Feature scaling with StandardScaler
* Training a Gradient Boosting Regressor with hyperparameters:
* 100 estimators
* 0.1 learning rate
* Maximum depth of 3
* Minimum samples split of 5
* Minimum samples leaf of 2

1. **Model Evaluation:**

* Performance measured using MSE, RMSE, MAE, and R² Score
* Validation on a held-out test set
* Model saved as a pickle file for deployment

**How authentication was added?**

1. **User Registration**:

* UserCreationForm for new account creation
* Form validation and error handling
* Automatic login after successful registration

1. **Login/Logout**:

* AuthenticationForm for user login
* Username/password validation
* Session management with Django's login/logout functions
* Success/error messages using Django's messaging framework

1. **Access Control**:

* @login\_required decorator on protected views (predict, dashboard)
* Unauthenticated users redirected to login page
* Navigation links change based on authentication status

**Steps for Integration**

1. **Model Integration:**

* Loading the trained model and scaler from pickle files
* Error handling with fallback paths
* Model prediction function that:
* Processes form input
* Encodes categorical features
* Scales numerical features
* Makes price predictions

1. **Web Application Development:**

* Django models to store prediction history
* URL routing for different application functions
* View functions for handling web requests
* Templates for UI presentation

1. **Data Visualization:**

* Dashboard with statistics from stored predictions
* Charts (bar, pie) showing prediction distributions
* Performance metrics display

1. **User Experience:**

* Form validation
* Responsive results display
* Historical prediction tracking
* Clean UI with visualization components

**Challenges encountered**

* Training the Dataset
* Cleaning the Dataset
* Creating Django setup (needing to put -m for the pip to work) it was realize late so I encountered many errors making the Django setup
* Integrating Dataset into the program