We want to first build out a total return time series. Then we are going to test a momentum implementation using various machine learning techniques.

To design a system for predicting home prices using historical time series data from the 100 largest metro areas in the US, we can break the system down into several steps. Here’s a detailed specification:

**Step 1: Data Preprocessing**

**Inputs:**

* **Historical Home Prices Data**: A time series of home prices for the 100 metro areas.

**Outputs:**

* **Processed Data**: Cleaned and possibly normalized time series data.
* **Features and Labels**: Split data into features (X) and labels (y).

**Dimensions:**

* **Raw Data**: (100, T), where 100 is the number of metro areas and T is the number of time periods.
* **Processed Data**: (100, T), after cleaning and normalizing.
* **Features (X)**: (100, T-1) if using a simple shift for features.
* **Labels (y)**: (100, T-1) if predicting the next time period.

**Step 2: Feature Engineering**

**Inputs:**

* **Processed Data**: Time series data for 100 metro areas.

**Outputs:**

* **Engineered Features**: Time series features that might include lagged values, moving averages, or other relevant features.

**Dimensions:**

* **Features (X)**: (100, T-n) where n depends on the number of lagged values or features included.
* **Labels (y)**: (100, T-n) same number of time periods as features.

**Step 3: Train-Test Split**

**Inputs:**

* **Engineered Features (X)**: Feature set.
* **Labels (y)**: Corresponding labels.

**Outputs:**

* **Training Set**: Subset of the data used for training.
* **Validation/Testing Set**: Subset of the data used for validation/testing.

**Dimensions:**

* **Training Features (X\_train)**: (100, T\_train)
* **Training Labels (y\_train)**: (100, T\_train)
* **Validation/Test Features (X\_val/test)**: (100, T\_val/test)
* **Validation/Test Labels (y\_val/test)**: (100, T\_val/test)

**Step 4: Model Design**

**Inputs:**

* **Training Set**: (X\_train, y\_train)

**Outputs:**

* **Trained Model**: A trained deep learning model (e.g., LSTM, GRU, Transformer).

**Step 5: Model Training**

**Inputs:**

* **Training Features (X\_train)**: (100, T\_train)
* **Training Labels (y\_train)**: (100, T\_train)
* **Model**: Chosen deep learning model.

**Outputs:**

* **Trained Model**: Model trained on the training data.

**Step 6: Model Evaluation**

**Inputs:**

* **Validation/Test Features (X\_val/test)**: (100, T\_val/test)
* **Validation/Test Labels (y\_val/test)**: (100, T\_val/test)
* **Trained Model**: Trained model from the previous step.

**Outputs:**

* **Evaluation Metrics**: Performance metrics like MAE, RMSE, etc.

**Step 7: Prediction**

**Inputs:**

* **New Data**: New time series data for prediction.
* **Trained Model**: Trained model from the previous step.

**Outputs:**

* **Predicted Prices**: Future home prices for the 100 metro areas.

**Dimensions:**

* **New Data**: (100, T\_new)
* **Predicted Prices**: (100, T\_pred)

**Detailed Example using LSTM:**

**Step 1: Data Preprocessing**

* Raw Data: (100, 120) assuming 10 years of monthly data.
* Normalize data (e.g., MinMax scaling).

**Step 2: Feature Engineering**

* Lagged features: Use previous 12 months for prediction.
* Features (X): (100, 108, 12) - 108 time steps remaining, 12 lagged values.
* Labels (y): (100, 108)

**Step 3: Train-Test Split**

* Split 80-20: Training on the first 80%, testing on the last 20%.
* Training Features (X\_train): (100, 86, 12)
* Training Labels (y\_train): (100, 86)
* Test Features (X\_test): (100, 22, 12)
* Test Labels (y\_test): (100, 22)

**Step 4: Model Design (LSTM)**

* Input shape for LSTM: (86, 12) per metro area.

**Step 5: Model Training**

* Train the LSTM model on (X\_train, y\_train).

**Step 6: Model Evaluation**

* Evaluate on (X\_test, y\_test).
* Calculate RMSE, MAE, etc.

**Step 7: Prediction**

* Use the model to predict future prices based on the latest available data.

**System Architecture Summary:**

1. **Data Collection**: Gather historical prices.
2. **Data Preprocessing**: Clean and normalize.
3. **Feature Engineering**: Create lagged features.
4. **Train-Test Split**: Divide data.
5. **Model Training**: Train chosen model.
6. **Model Evaluation**: Assess performance.
7. **Prediction**: Forecast future prices.