Time - Series Forecasting on Dom.kz

prepared by: Abdukerim Adina 220103396

Link to the project presentation:
Final Project Presentation

Introduction

This project focuses on the collection, cleaning, and analysis of data from the Dom.kz website, which specializes in selling apartments.

The general idea of the project is to collect data, then prepare and conduct Machine Learning Time Series techniques, specifically LSTM to build price prediction models based on date.

Project Structure

The project is divided into the following stages:

1. **Data Collection**: Automated web scraping of house prices using Selenium.
2. **Data Analysis**: Just having an overall view of our dataset
3. **Implementing 3 types of LSTM (Long-short Term Model):** Implement Vanilla, Bidirectional, and Stacked LSTMs
4. **Comparing models' performances**

This project implemented an automated data collection system using the Selenium library to scrape information about books from the website Dom.kz. The goal was to extract detailed information about apartments on sale, including their prices, number of rooms, residential complex, apartment area, and other characteristics.

Process Steps

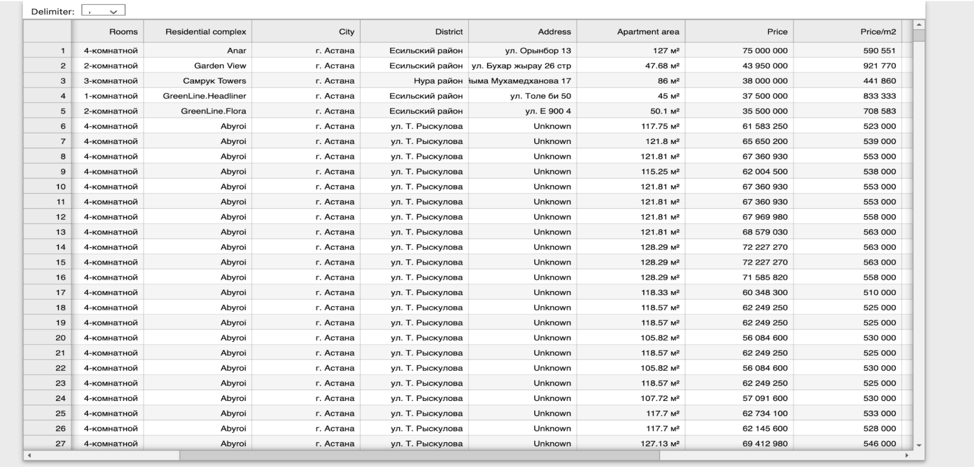
1. **Web Driver Initialization:**
    - **ChromeDriver** was used to control the Google Chrome browser.
    - Browser settings included incognito mode, disabling automation controls (to bypass anti-bot detection), and adding a custom User-Agent header.
2. **Navigating to the Target Website:**
    - The program accessed the specified apartment catalog page on the website.
3. **Page Scrolling:**
    - Automatic scrolling was implemented to load all elements on the page.
    - The current scroll position was updated to determine when the page had fully loaded.
4. **Data Collection from the Current Page:**
    - ○ For each product on the page, the following details were extracted:
        - ■ Book title, author, genre, price, description, number of pages, language, age restrictions, publisher, rating, year of publication, book series, binding type, and country of origin.
    - ○ XPath and CSS selectors were used to locate the required elements.
    - ○ Data for each product was saved in a CSV file, with preprocessing to remove unnecessary characters (e.g., line breaks).
5. **Working with Product Features:**
    - ○ To extract detailed data, the program automatically scrolled to the element and clicked the "Expand Characteristics" button (if available).
6. **Closing Pop-Ups:**
    - ○ A pop-up window for selecting a city was detected and closed automatically to ensure smooth data collection.
7. **Navigating Between Pages:**
    - ○ After collecting data from the current page, the program automatically located and clicked the "Next" button to navigate to the next page of the catalog.
8. **Error Handling:**
    - ○ Exception handling was implemented to improve the process's stability. This allowed the program to skip errors caused by missing elements and continue execution.
9. **Completion of the Process:**
    - ○ After processing all pages, the web driver automatically terminated, closing the browser.

Outcome

The automation successfully collected structured book data and saved it to a apartments.csv file. The file includes the following information:

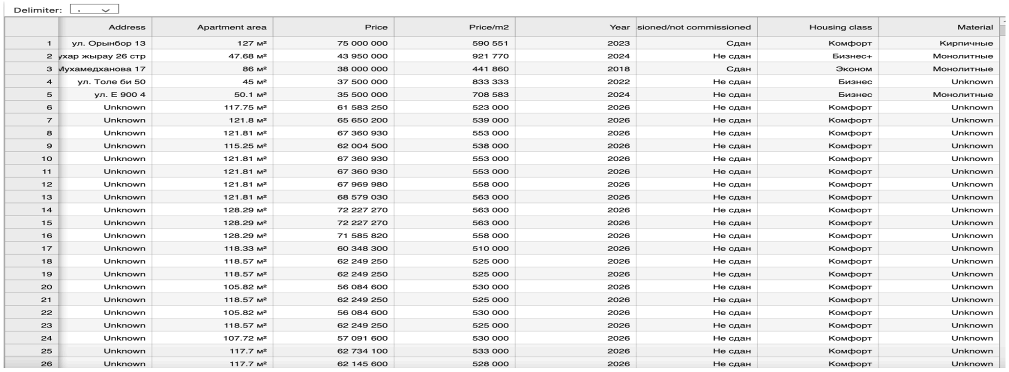
- Rooms
- Residential complex
- City
- Address
- Apartment area
- Price / m2
- Year
- Commissioned / Not Commissioned
- Housing class
- Material

Result dataset:



Delimiter: ,

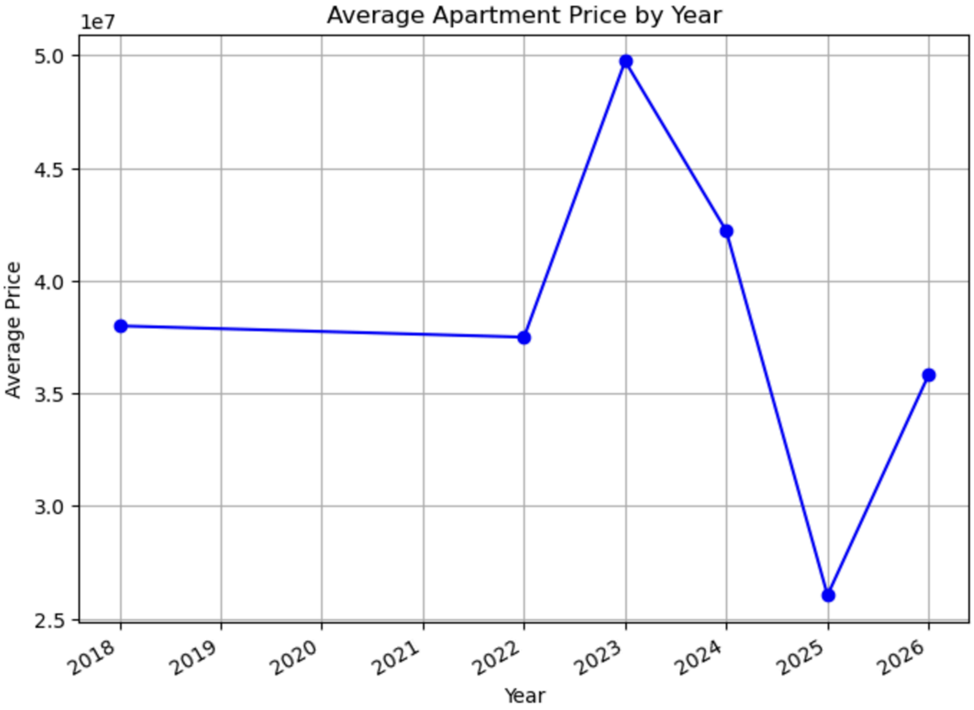
	Rooms	Residential complex	City	District	Address	Apartment area	Price	Price/m2
1	4-комнатной	Anar	г. Астана	Есильский район	ул. Орынбор 13	127 м²	75 000 000	590 551
2	2-комнатной	Garden View	г. Астана	Есильский район	ул. Бухар жырау 26 стр	47.68 м²	43 950 000	921 770
3	3-комнатной	Самрук Towers	г. Астана	Нура район	ыма Мухамедханова 17	86 м²	38 000 000	441 860
4	1-комнатной	GreenLine.Headliner	г. Астана	Есильский район	ул. Толе би 50	45 м²	37 500 000	833 333
5	2-комнатной	GreenLine.Flora	г. Астана	Есильский район	ул. Е 900 4	50.1 м²	35 500 000	708 583
6	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	117.75 м²	61 583 250	523 000
7	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	121.8 м²	65 650 200	539 000
8	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	121.81 м²	67 360 930	553 000
9	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	115.25 м²	62 004 500	538 000
10	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	121.81 м²	67 360 930	553 000
11	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	121.81 м²	67 360 930	553 000
12	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	121.81 м²	67 969 980	558 000
13	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	121.81 м²	68 579 030	563 000
14	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	128.29 м²	72 227 270	563 000
15	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	128.29 м²	72 227 270	563 000
16	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	128.29 м²	71 585 820	558 000
17	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	118.33 м²	60 348 300	510 000
18	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	118.57 м²	62 249 250	525 000
19	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	118.57 м²	62 249 250	525 000
20	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	105.82 м²	56 084 600	530 000
21	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	118.57 м²	62 249 250	525 000
22	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	105.82 м²	56 084 600	530 000
23	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	118.57 м²	62 249 250	525 000
24	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	107.72 м²	57 091 600	530 000
25	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	117.7 м²	62 734 100	533 000
26	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	117.7 м²	62 145 600	528 000
27	4-комнатной	Abyroi	г. Астана	ул. Т. Рыскулова	Unknown	127.13 м²	69 412 980	546 000



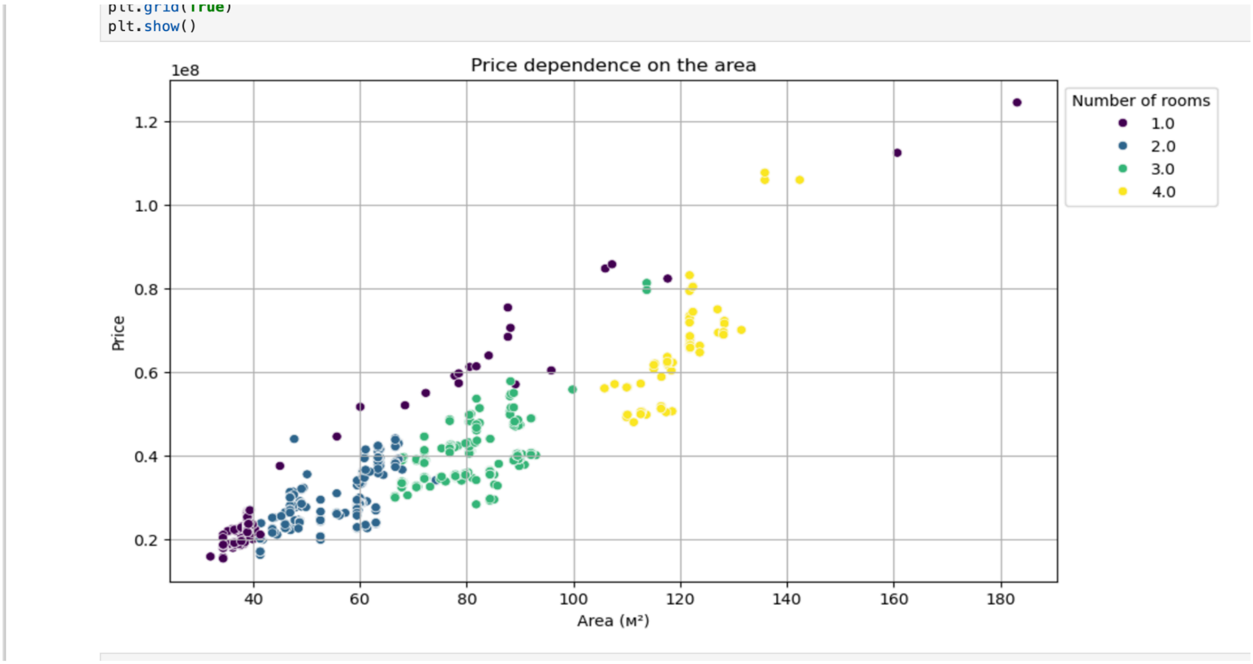
Delimiter: ,

	Address	Apartment area	Price	Price/m2	Year	sioned/not commissioned	Housing class	Material
1	ул. Орынбор 13	127 м²	75 000 000	590 551	2023	Сдан	Комфорт	Кирпичные
2	ухар жырау 26 стр	47.68 м²	43 950 000	921 770	2024	Не сдан	Бизнес+	Монолитные
3	Мухамедханова 17	86 м²	38 000 000	441 860	2018	Сдан	Эконом	Монолитные
4	ул. Толе би 50	45 м²	37 500 000	833 333	2022	Не сдан	Бизнес	Unknown
5	ул. Е 900 4	50.1 м²	35 500 000	708 583	2024	Не сдан	Бизнес	Монолитные
6	Unknown	117.75 м²	61 583 250	523 000	2026	Не сдан	Комфорт	Unknown
7	Unknown	121.8 м²	65 650 200	539 000	2026	Не сдан	Комфорт	Unknown
8	Unknown	121.81 м²	67 360 930	553 000	2026	Не сдан	Комфорт	Unknown
9	Unknown	115.25 м²	62 004 500	538 000	2026	Не сдан	Комфорт	Unknown
10	Unknown	121.81 м²	67 360 930	553 000	2026	Не сдан	Комфорт	Unknown
11	Unknown	121.81 м²	67 360 930	553 000	2026	Не сдан	Комфорт	Unknown
12	Unknown	121.81 м²	67 969 980	558 000	2026	Не сдан	Комфорт	Unknown
13	Unknown	121.81 м²	68 579 030	563 000	2026	Не сдан	Комфорт	Unknown
14	Unknown	128.29 м²	72 227 270	563 000	2026	Не сдан	Комфорт	Unknown
15	Unknown	128.29 м²	72 227 270	563 000	2026	Не сдан	Комфорт	Unknown
16	Unknown	128.29 м²	71 585 820	558 000	2026	Не сдан	Комфорт	Unknown
17	Unknown	118.33 м²	60 348 300	510 000	2026	Не сдан	Комфорт	Unknown
18	Unknown	118.57 м²	62 249 250	525 000	2026	Не сдан	Комфорт	Unknown
19	Unknown	118.57 м²	62 249 250	525 000	2026	Не сдан	Комфорт	Unknown
20	Unknown	105.82 м²	56 084 600	530 000	2026	Не сдан	Комфорт	Unknown
21	Unknown	118.57 м²	62 249 250	525 000	2026	Не сдан	Комфорт	Unknown
22	Unknown	105.82 м²	56 084 600	530 000	2026	Не сдан	Комфорт	Unknown
23	Unknown	118.57 м²	62 249 250	525 000	2026	Не сдан	Комфорт	Unknown
24	Unknown	107.72 м²	57 091 600	530 000	2026	Не сдан	Комфорт	Unknown
25	Unknown	117.7 м²	62 734 100	533 000	2026	Не сдан	Комфорт	Unknown
26	Unknown	117.7 м²	62 145 600	528 000	2026	Не сдан	Комфорт	Unknown

Dataset Overview

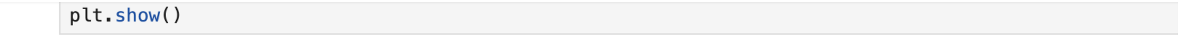


The graph below shows how the apartment price has changed over years. Overall, the price has been stable from 2018 till 2022, following with significant growth in 2022 and unexpected drop in 2025.

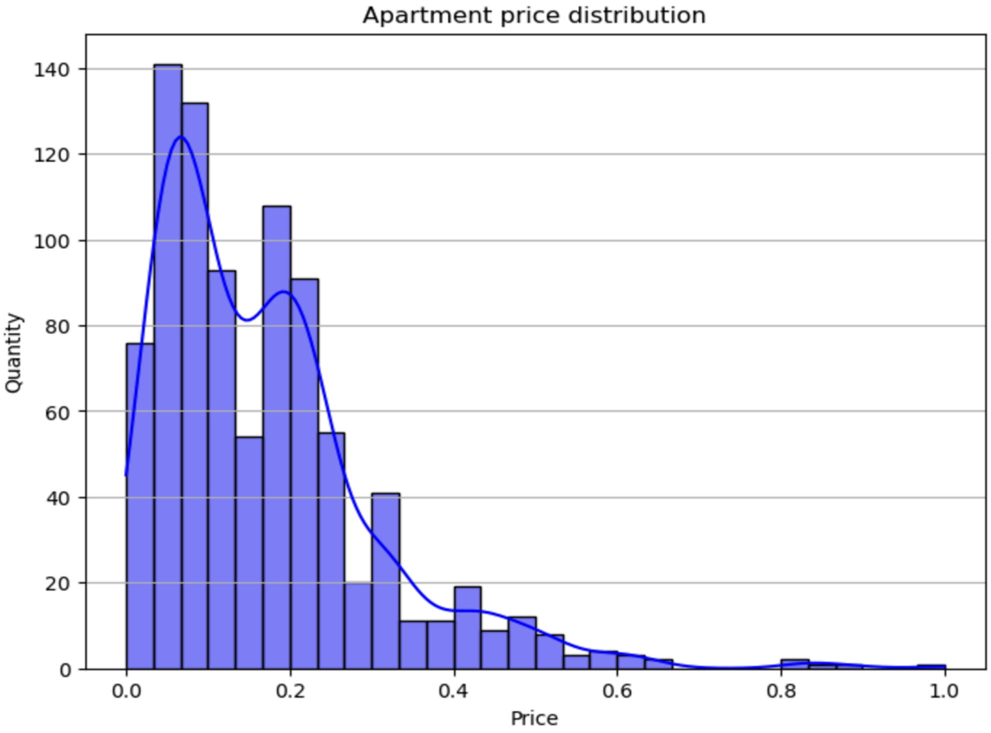


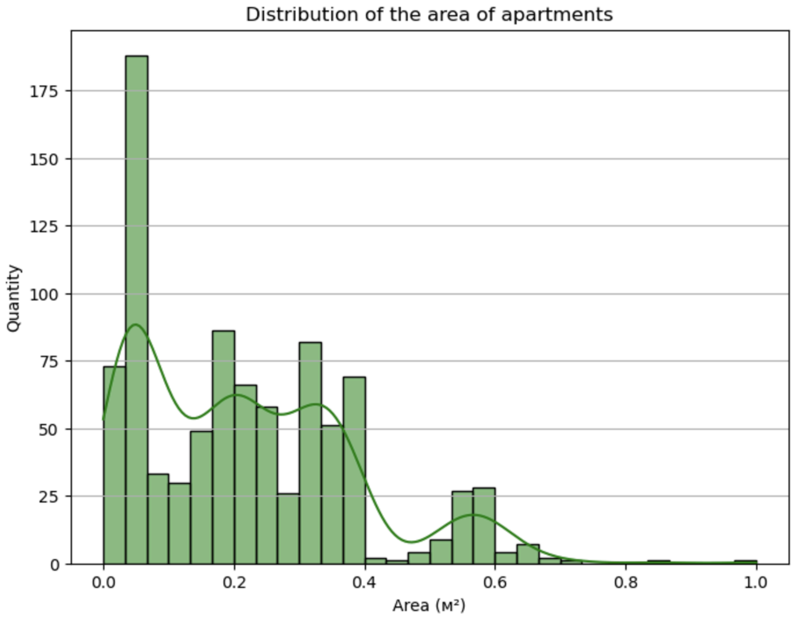
```
plt.grid(True)
plt.show()
```

We have a scatter plot showing the Price dependence on the area of apartments. It is seen that the area and price is gonna be smaller depending on the number of rooms. Some outliers might be the indication of having a small number of rooms but high area, thus a high price and vice-versa.



```
plt.show()
```





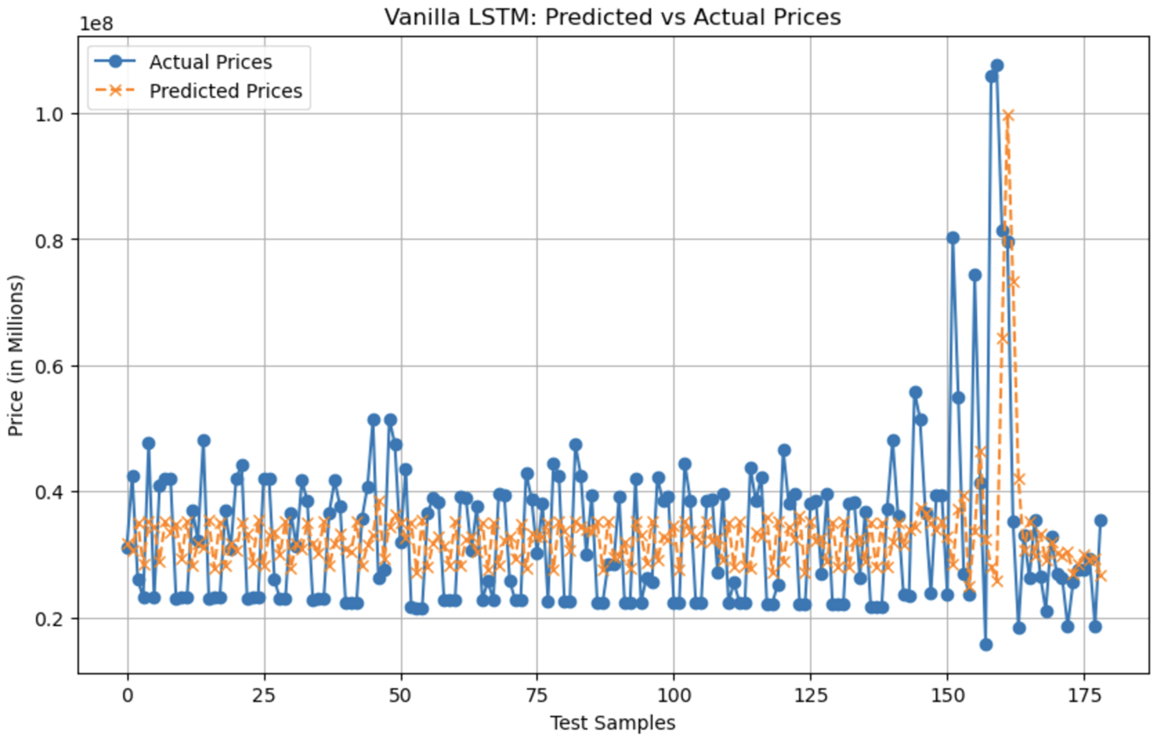
Here both the distribution of area of apartments and apartment price is observed. Indicating both rose and decrease in quantity.

Implementing 4 types of LSTM(Long-Short Term Memory)

1) Vanilla LSTM (Single featured)

It's simple and works well for basic time series forecasting tasks, like predicting apartment prices.

The dataset and task (predicting a single value for each time step) do not require stacked or bidirectional LSTMs.



Observations

General Pattern:

The predicted prices (orange dashed line with 'x' markers) generally follow the trend of the actual prices (blue line with 'o' markers).

This indicates that the model is learning the overall pattern of the data.

Deviations:

At some points, there are significant deviations between predicted and actual prices, particularly where the actual prices spike (e.g., around test samples 150–175). This suggests the model struggles with sudden changes or outliers in the dataset.

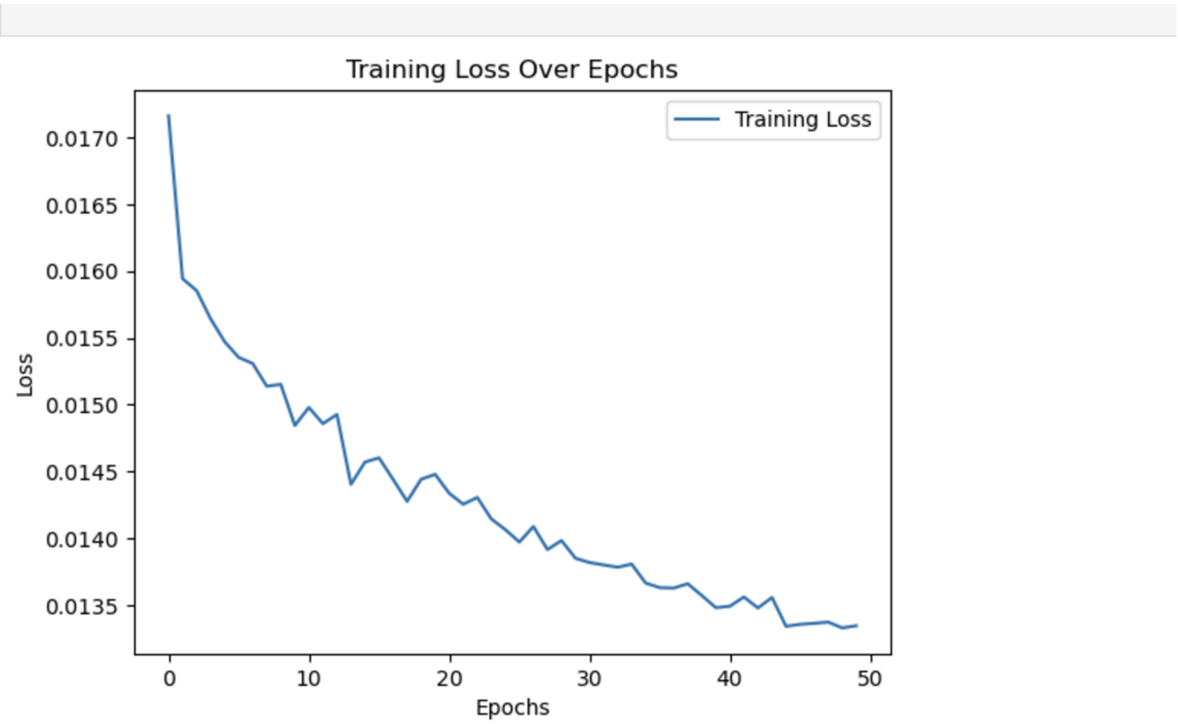
Overlapping Predictions:

For many test samples, the predicted prices align closely with the actual prices (e.g., from test samples 0–125). This indicates the model performs well for stable or consistent price ranges.

Sharp Peaks:

The spikes in actual prices are not accurately captured by the predictions. This could indicate: A lack of sufficient data for extreme price points (outliers). Model underfitting for these specific patterns.

It would be a good practice to see how our model is performing at each epoch



What Happens During Training?

Forward Propagation:

The input sequence (X_train) is passed through the LSTM network to predict the target (y_train).

Loss Calculation:

The model calculates the difference between the predicted and actual values using the loss function (mse).

Backward Propagation:

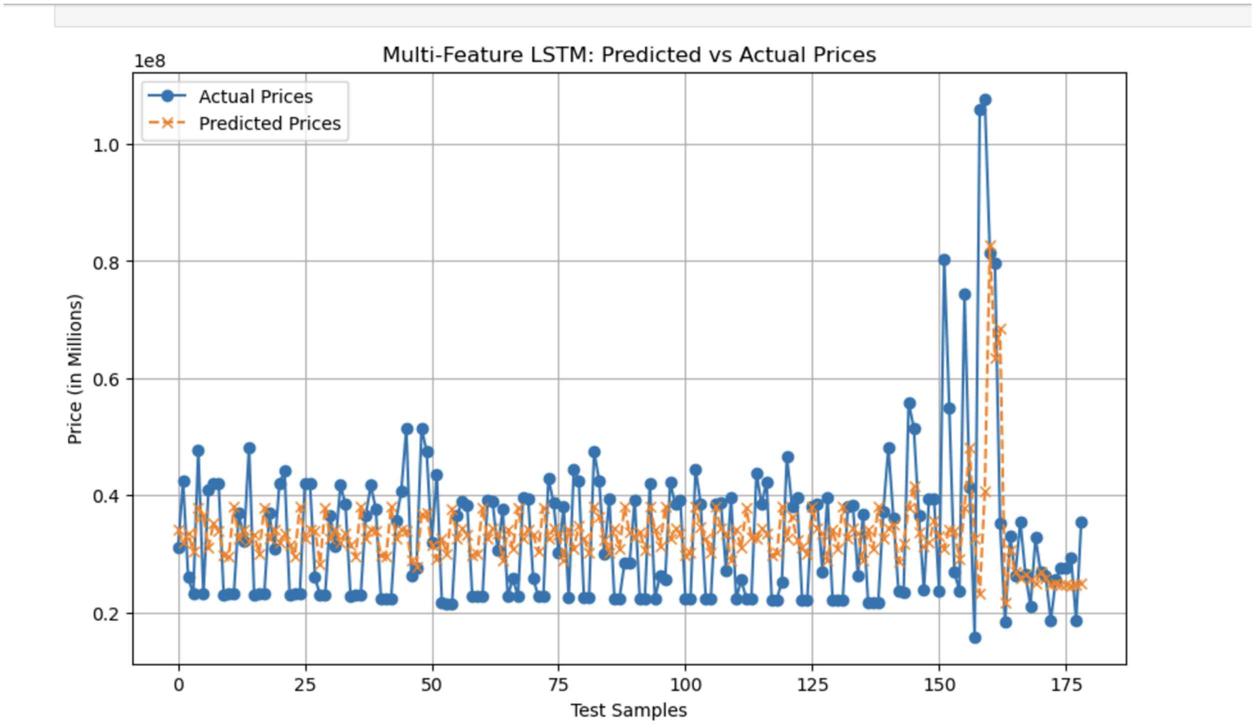
The model adjusts its weights based on the loss to improve predictions in subsequent iterations.

Repeat:

This process is repeated for each epoch, and the weights are updated to minimize the loss.¶

2) Implementing Multi-feature LSTM

This model would be helpful to identify how adding multiple features will affect model performance(to see how Price will be predicted based on Area, Housing class and Year)



Observations

The predicted prices align with the general trend of the actual prices, indicating that the model has successfully learned the relationship between features (e.g., year, apartment area, and housing class) and prices.

Spikes and Variations:

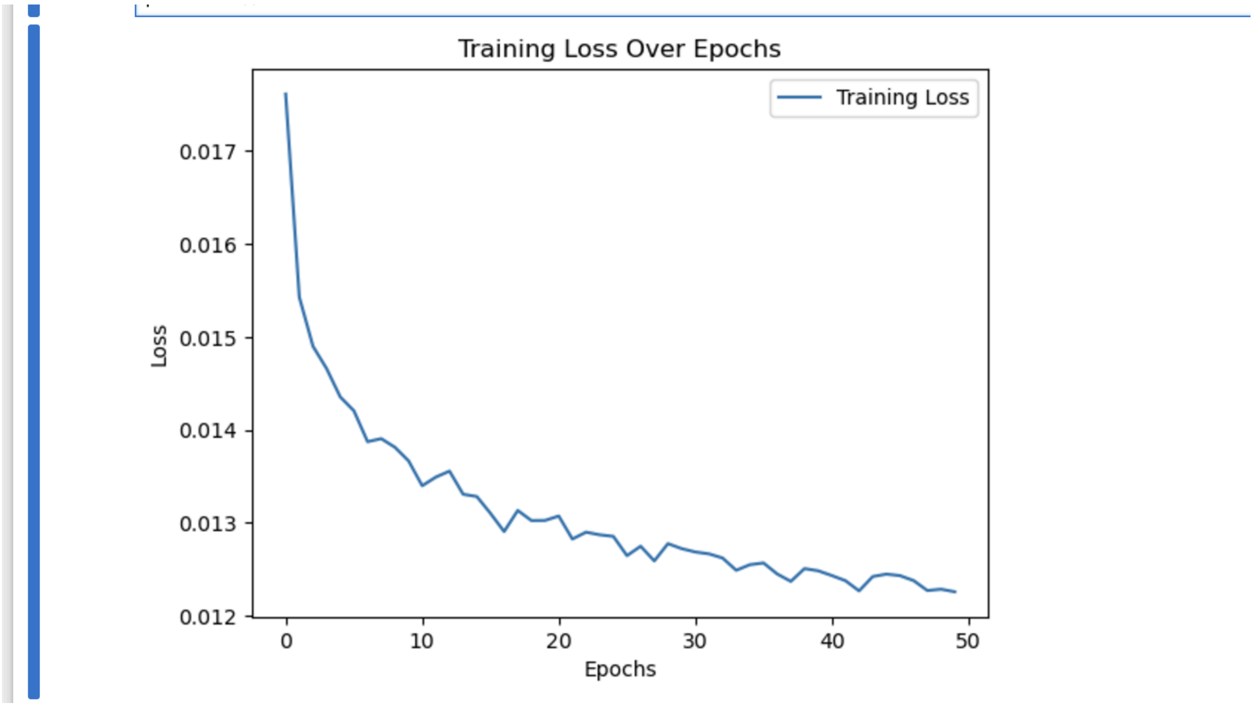
The model struggles to capture the sharp price spikes observed in the actual prices (e.g., around sample 150). This suggests that the model might underperform in predicting extreme or outlier values, which could be due to:

Performance on Stable Regions:

In regions where the actual prices exhibit smaller variations, the model's predictions are closer to the actual values. This indicates that the model performs well in stable or less volatile segments of the data.

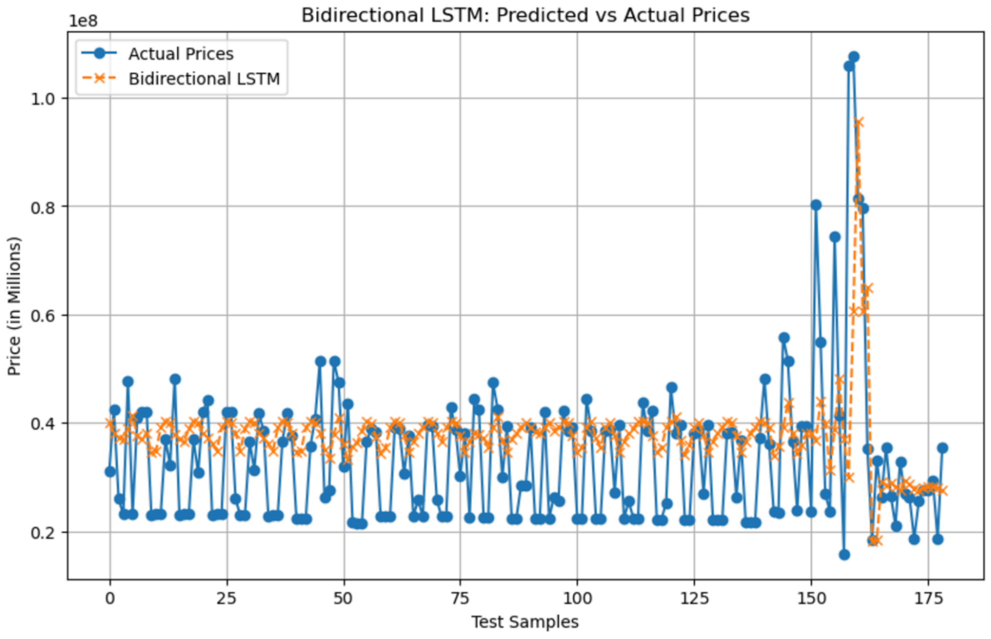
See how our model is performing at each epoch

And again we are convinced that the loss is decreasing at each epoch



3) Implementing Bidirectional LSTM

Bidirectional LSTMs process the sequence data in both forward and backward directions. This helps capture both past and future temporal dependencies.



Observations

The predicted prices align with the general trend of the actual prices, indicating that the model has successfully learned the relationship between featuresю. And even performs better that Vanilla LSTM

Spikes and Variations:

The model works well to capture the sharp price spikes observed in the actual prices (e.g., around sample 150), which was hard for Vanilla LSTM. This suggests that the model might perform better and has learnt trends better than Vanilla LSTM.

Mean Absolute Error (MAE): 9,663,217.68

What it means: On average, the predicted apartment prices are off by approximately 9.66 million KZT from the actual prices.

Interpretation: This gives you a clear idea of the average magnitude of errors, regardless of direction (positive or negative).

Root Mean Squared Error (RMSE): 13,045,817.82

What it means: The standard deviation of the prediction errors is around 13.05 million KZT.

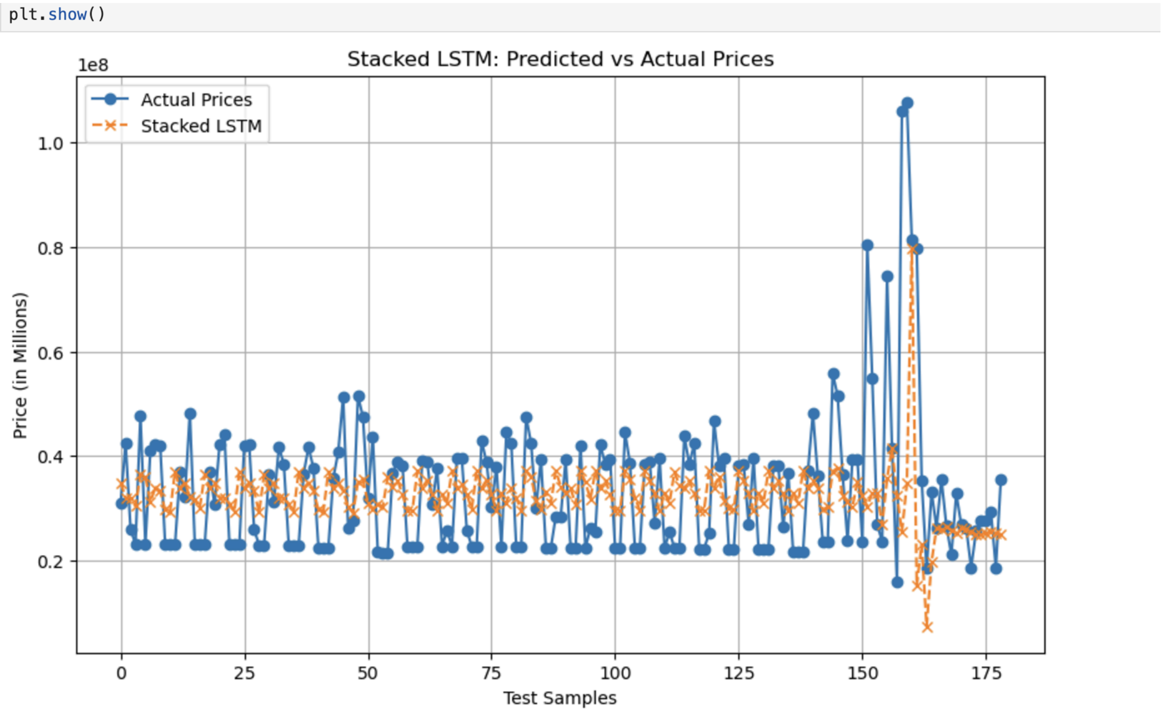
Interpretation:

RMSE penalizes larger errors more than MAE because it squares the errors.

A higher RMSE compared to MAE suggests that there may be some outliers (large prediction errors) in the dataset that are affecting the model's performance.

4) Implementing Stacked LSTM

Stacking LSTMs can capture hierarchical patterns and dependencies in sequential data more effectively, leading to improved model performance.



```
plt.show()
```

Observations:

1. **General Trend Alignment**:

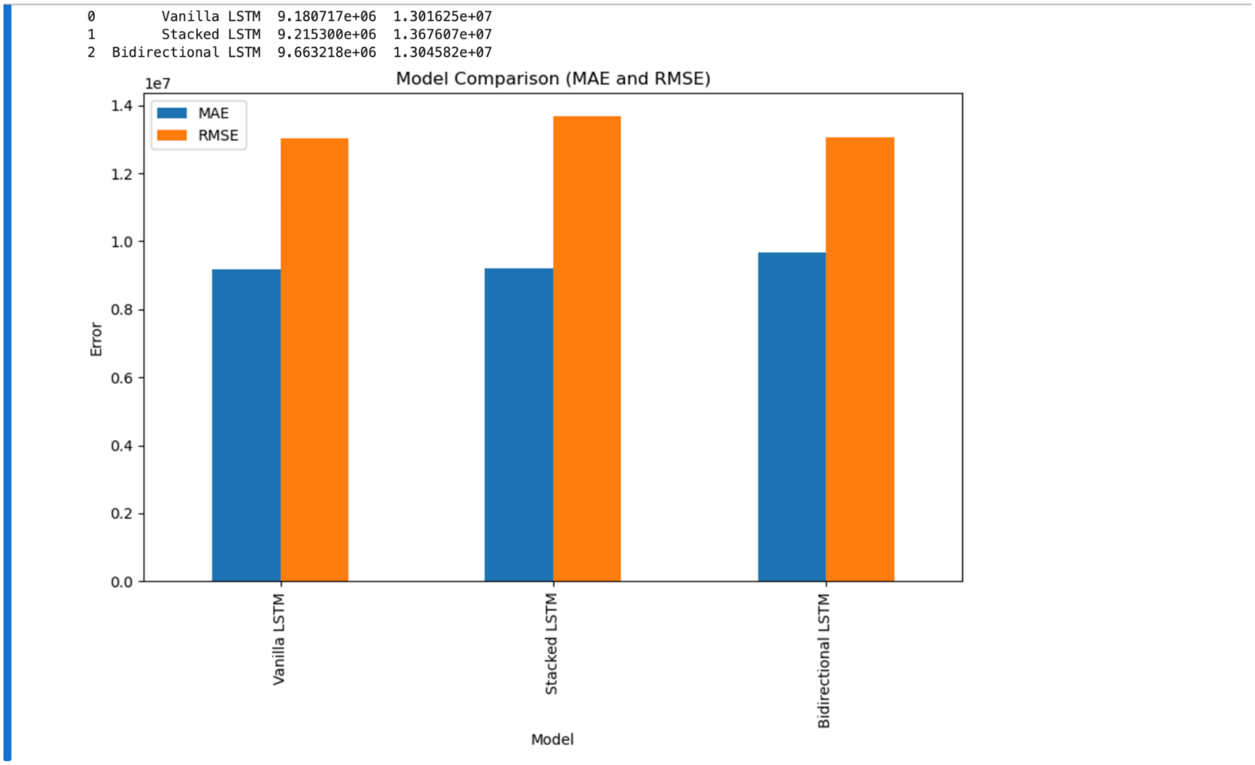
- The Stacked LSTM model captures the overall trend of actual prices, demonstrating its ability to learn from multiple features.

2. **Performance**

   The model performs well in regions with smaller fluctuations, where the predicted prices align closely with the actual prices.

3. **Challenges with Sharp Spikes**:
   - In areas with sharp peaks or sudden price changes (e.g., around sample 150), the model struggles to accurately predict the exact values, as seen by the deviations between the actual and predicted prices. This may indicate a need for more training data or additional features to better capture extreme variations.

5) Models Comparison



```
0        Vanilla LSTM  9.180717e+06  1.301625e+07
1        Stacked LSTM  9.215300e+06  1.367607e+07
2  Bidirectional LSTM  9.663218e+06  1.304582e+07
```

Conclusion

Vanilla LSTM performs best in terms of both MAE and RMSE, showing it can make accurate predictions without excessive complexity.

Stacked LSTM has the highest RMSE, which could indicate overfitting or insufficient regularization.

Bidirectional LSTM provides a balance, capturing both temporal dependencies in the data and maintaining reasonable error rates.

This visualization effectively highlights the trade-offs between model complexity and prediction accuracy

General Terms

- **Time-Series Data**: A sequence of data points recorded over time intervals. In this project, apartment prices are recorded over years and related to other features.
- **Forecasting**: The process of predicting future values based on historical data.
- **Temporal Order**: The chronological sequence of data points, critical in time-series analysis to avoid data leakage.

Dataset and Features

- **Apartment Price**: The target variable in the dataset, representing the price of an apartment in the given data.
- **Area**: A feature indicating the size of the apartment in square meters, used to predict prices.
- **Year**: A feature indicating the year associated with the recorded apartment prices.
- **Housing Class**: A categorical feature indicating the type of housing (e.g., economy, luxury).

Machine Learning Concepts

- **LSTM (Long Short-Term Memory)**: A type of Recurrent Neural Network (RNN) designed to handle sequential data and long-term dependencies. It’s widely used in time-series forecasting.
  - **Vanilla LSTM**: A simple LSTM with a single hidden layer.
  - **Stacked LSTM**: A deeper version of LSTM with multiple layers stacked to learn complex patterns.
  - **Bidirectional LSTM**: An LSTM that processes data in both forward and backward directions, capturing past and future dependencies.

Evaluation Metrics

- **Mean Absolute Error (MAE)**: The average absolute difference between predicted and actual values. It gives an easy-to-interpret measure of error.
- **Root Mean Squared Error (RMSE)**: The square root of the average squared differences between predicted and actual values. It penalizes large errors more than MAE.
- **Mean Squared Error (MSE)**: The average of the squared differences between predicted and actual values. Used to highlight larger errors.

Model Training and Validation

- **Training Set**: A subset of the data used to train the machine learning model.
- **Test Set**: A subset of the data used to evaluate the model's performance on unseen data.
- **Cross-Validation**: A method to evaluate the model's robustness by splitting the data into multiple training and testing sets.
  - **Rolling Window Cross-Validation**: A specific cross-validation technique for time-series data that ensures the temporal order of data is preserved.

Optimization and Activation

- **Adam Optimizer**: An adaptive learning rate optimization algorithm used to minimize the model's loss function.
- **ReLU (Rectified Linear Unit)**: An activation function that outputs the input directly if positive, otherwise outputs zero. It helps prevent vanishing gradients.

Preprocessing Techniques

- **Normalization**: The process of scaling features to a specific range (e.g., 0 to 1) to improve model training.
- **Feature Engineering**: The process of creating additional features, such as cyclical representations of time (`Year_Sin` and `Year_Cos`), to better capture patterns in the data.

Visualization

- **Predicted vs Actual Prices Plot**: A line graph comparing the model's predictions against actual prices to visually assess its performance.
- **Model Comparison Bar Chart**: A bar plot showing MAE and RMSE for different LSTM models to compare their effectiveness.

Errors and Challenges

- **Overfitting**: When a model performs well on the training data but poorly on unseen data. It can occur with overly complex models like Stacked LSTMs.
- **Underfitting**: When a model fails to capture the underlying patterns in the data, resulting in poor performance on both training and testing data.
- **Data Leakage**: When information from the test set unintentionally influences the training process, leading to overly optimistic evaluations.