Advanced Machine Learning Approaches for Forecasting House Prices

Hari Chandana Kotnani

Introduction:

I explored the dynamic and complex world of real estate pricing, focusing specifically on forecasting house prices using historical data obtained from Zillow. The study encompasses a comprehensive dataset spanning from January 2000 to October 2023. I aim to leverage advanced statistical and machine learning models to predict future house prices for 12 to 24 months. Given the ever-changing nature of the real estate market, influenced by economic factors, population growth, and urban development, such predictions are invaluable for investors, homebuyers, and policymakers. This analysis aims not only to forecast prices but also to understand the underlying patterns and trends in the housing market, thereby contributing valuable insights into real estate economics.

I also utilized **Tableau** for data visualization to provide clear, insightful representations of complex datasets. Additionally, as part of this course, I have completed the **AWS Cloud Practitioner Certification**, further enriching my approach to data analysis and interpretation in this project.

Dataset:

The dataset for this study comes from Zillow, covering monthly house prices from January 2000 to October 2023 across the U.S. It includes the Home Value Index, representing average prices in various regions, along with details like region name, type (Metropolitan Statistical Area), and size rank. The data is curated to ensure accuracy, focusing on outliers and missing value management, crucial for reliable forecasting. Each record contains
Date: The date of the record.
Region ID: A unique identifier for the region.
Region Name: The name of the region, is a city or area name.
State Name: The name of the state where the region is located.
Home Value Index: This is an index value representing home prices.
Size Rank: A rank based on the size of the region, possibly in terms of population or area.
all these provide a detailed view of the housing market trends.

Preprocessing:

The dataset was first filtered to focus on **Salt Lake City**, Utah.

- **Data Filtering**: Select data specifically for "Salt Lake City, UT".
- **Date Formatting**: Convert the 'Date' column to datetime format for time-series analysis.
- **Feature Selection**: Retain only the 'Date' and 'Home Value Index' columns.
- **Data Normalization**: Apply MinMaxScaler to normalize the 'Home Value Index' between 0 and 1.
- **Train-Test Split**: Divide the dataset into training (90%) and testing sets (10%).

Models:

Bidirectional Long Short-Term Memory (Bi-LSTM):

I implemented a Bidirectional Long Short-Term Memory (Bi-LSTM) network, a sophisticated version of traditional LSTM models. The Bi-LSTM is particularly effective in handling sequential data due to its ability to process information in both forward and backward directions. This dual-direction capability enables the model to capture a more comprehensive context, significantly benefiting complex sequence prediction tasks.

Model Architecture and Training

The architecture of our Bi-LSTM model comprises key elements including an input layer and a Bidirectional wrapper around LSTM units, facilitating learning in both directions. Following this, dense layers enhance pattern recognition and output generation. We chose a mean squared error loss function and an Adam optimizer for efficient training. To prevent overfitting, we employed an EarlyStopping callback, ceasing training when no further improvement in validation loss was detected. The Sequential model features a Bi-LSTM layer and a dense layer, trained with both training and test data, integrating the EarlyStopping callback to optimize the training process.

Model Visualizations

The project included several visualizations: training data results, testing data performance, and future predictions. These visualizations delineate actual values, predicted values, and future forecasts, providing a comprehensive view of the model's effectiveness. In terms of performance, the Bi-LSTM model demonstrated remarkable accuracy in predicting future values in the dataset.

Convolutional Neural Network (CNN):

I implemented a Convolutional Neural Network (CNN) for my project, adapting this traditionally image-focused model for time-series data analysis. CNNs are effective in detecting patterns over intervals, making them suitable for tasks like forecasting house prices.

Model Architecture and Training:

Architecture: Includes convolutional layers with filters to capture local dependencies, pooling layers for reducing dimensionality, and dense layers for prediction.

Training: Employed mean squared error loss and Adam optimizer. An EarlyStopping callback was used to prevent overfitting.

Model Visualizations:

Visualizations displayed the CNN's performance on training and test data, and its future predictions. Comparative plots highlighted the accuracy of the model by juxtaposing actual and predicted values.

Long Short-Term Memory (LSTM) :

I developed a Long Short-Term Memory (LSTM) model, a type of recurrent neural network (RNN) particularly adept at processing and making predictions based on time series data. Unlike Bi-LSTM, LSTM processes data in a single forward direction, which can be more suitable for certain types of sequential data where future context is not essential.

Model Architecture and Training:

The architecture of the LSTM model includes an input layer, LSTM units, and dense layers. The LSTM units are responsible for learning temporal dependencies in the data. Following the LSTM layers, dense layers are used for pattern recognition and generating outputs. I opted for a mean squared error loss function and an Adam optimizer for training efficiency. To mitigate overfitting, an EarlyStopping callback was implemented, halting training when the validation loss ceased to improve. The model, comprising an

LSTM layer followed by a dense layer, was trained using both training and test data. The EarlyStopping callback was integrated to optimize training.

Model Visualizations:

The project encompassed several visualizations including training data outcomes, test data performance, and future predictions. These visualizations distinctively illustrate actual values, predicted values, and future forecasts, offering a comprehensive perspective on the model's efficacy.

Random Forest:

In this project, I implemented a Random Forest model, known for its robustness and accuracy in handling regression tasks. This ensemble learning method, comprising numerous decision trees, excels in reducing overfitting by averaging the results of individual trees, thereby enhancing prediction reliability.

Model Architecture and Training:

I employed the RandomForestRegressor from sklearn, focusing on key parameters like the number of trees. The model was trained on preprocessed historical house price data, involving normalization and feature engineering to improve accuracy.

Model Evaluation:

The Random Forest model's performance was assessed using metrics like Root Mean Squared Error (RMSE). Hyperparameter tuning was applied for optimization.

Visualization and Predictions:

Visualizations were created to compare actual values with predictions, and to forecast future house prices, providing insights into market trends.

Recurrent Neural Network (RNN):

I developed a Recurrent Neural Network (RNN) model to manage sequential data. Unlike more advanced models like the Bi-LSTM, a basic RNN processes information in a straightforward, unidirectional manner.

Model Architecture and Training:

The RNN's architecture is simpler. It includes an input layer followed by RNN units, and then dense layers for pattern recognition and output. The model is trained using a mean squared error loss function and an Adam optimizer. To mitigate overfitting, an EarlyStopping callback is used, halting training when there's no improvement in validation loss. The model is structured with a SimpleRNN layer followed by a dense layer, trained with both training and test data sets, integrating EarlyStopping for efficient training.

Model Visualizations:

The project incorporates various visualizations to demonstrate the model's performance on training and testing data, and to make future predictions. These visualizations compare actual values with the model's predictions and project future values, offering insights into the RNN model's effectiveness.

XGBoost (Extreme Gradient Boosting):

I implemented XGBoost, a powerful and efficient machine-learning technique known for its performance in structured data. XGBoost stands out for its speed and accuracy, especially in regression tasks.

Model Architecture and Training:

The XGBoost model was configured with parameters including a learning rate of 0.1, max depth of 5, and 100 trees (n_estimators). Regularization (alpha = 10) was used to prevent overfitting. The model, aimed at regression, was trained on features derived from our dataset, correcting errors sequentially with each new tree.

Model Evaluation:

Performance was evaluated using Root Mean Squared Error (RMSE), indicating the model's accuracy. The low RMSE scores on both training and testing datasets highlighted the model's predictive strength.

Visualizations:

We created visualizations to illustrate the model's predictions compared to actual values. These helped in assessing the model's accuracy and understanding its forecasting ability.

Results:

By comparing the performance of various machine learning models in predicting future house prices. The models included Bi-LSTM, CNN, LSTM, Random Forest, RNN, and XGBoost. The effectiveness of each model was assessed based on their Root Mean Squared Error (RMSE) values for both training and testing datasets, and their ability to forecast future values.

- **Bidirectional Long Short-Term Memory (Bi-LSTM):**
  - **Train RMSE:** 2248.33
  - **Test RMSE:** 11364.27
  - The Bi-LSTM model showed relatively low RMSE scores, indicating good predictive accuracy. It predicted a steady increase in house prices over the next two years, reaching over 1.2 million by October 2025.
- **Convolutional Neural Network (CNN):**
  - **Train RMSE:** 3684.99
  - **Test RMSE:** 26093.70
  - CNN had higher RMSE values, suggesting less accuracy. Its predictions were less dynamic, with values fluctuating slightly around the 500k mark before dropping and then plateauing around 440k in mid-2024.
- **Long Short-Term Memory (LSTM) :**
  - **Train RMSE:** 2822.85
  - **Test RMSE:** 18807.96
  - LSTM performed moderately well. Its predictions showed a gradual increase in house prices, reaching just over 1 million by October 2025.
- **Random Forest:**
  - **Train RMSE:** 773.20
  - **Test RMSE:** 92753.70
  - Despite a low training RMSE, Random Forest had a very high testing RMSE, indicating overfitting. Its future value predictions were constant, not reflecting real-world expectations.
- **Recurrent Neural Network (RNN):**
  - **Train RMSE:** 2143.39
  - **Test RMSE:** 7300.05
  - RNN demonstrated good predictive accuracy, with RMSE values better than most other models. It forecasted a steady increase in house prices, reaching around 568k by October 2025.

- **Extreme Gradient Boosting (XGBoost):**
    - **Train RMSE:** 44813.24
    - **Test RMSE:** 254029.01
    - XGBoost had the highest RMSE scores, indicating poor performance in this scenario. It predicted a flat line for future values, staying around 275k, which is unrealistic.

Conclusion:

The comparative analysis of various machine learning models for predicting future house prices reveals significant insights into the strengths and weaknesses of each approach. Key Findings:

- Bi-LSTM and RNN emerged as the most effective models in this scenario. They demonstrated a good balance between training and testing accuracy, suggesting their robustness and reliability in handling sequential data for price prediction.
- LSTM showed moderate accuracy, making it a potentially viable option, although it was outperformed by Bi-LSTM and RNN.
- CNN and Random Forest displayed certain limitations. CNN struggled with higher error rates, and Random Forest exhibited signs of overfitting, reflected in its high test RMSE despite a low train RMSE.
- XGBoost, known for its effectiveness in various scenarios, surprisingly performed poorly in this specific task. Its predictions were unrealistic, indicating that it might not be the best choice for time-series data like house price forecasting for my dataset.

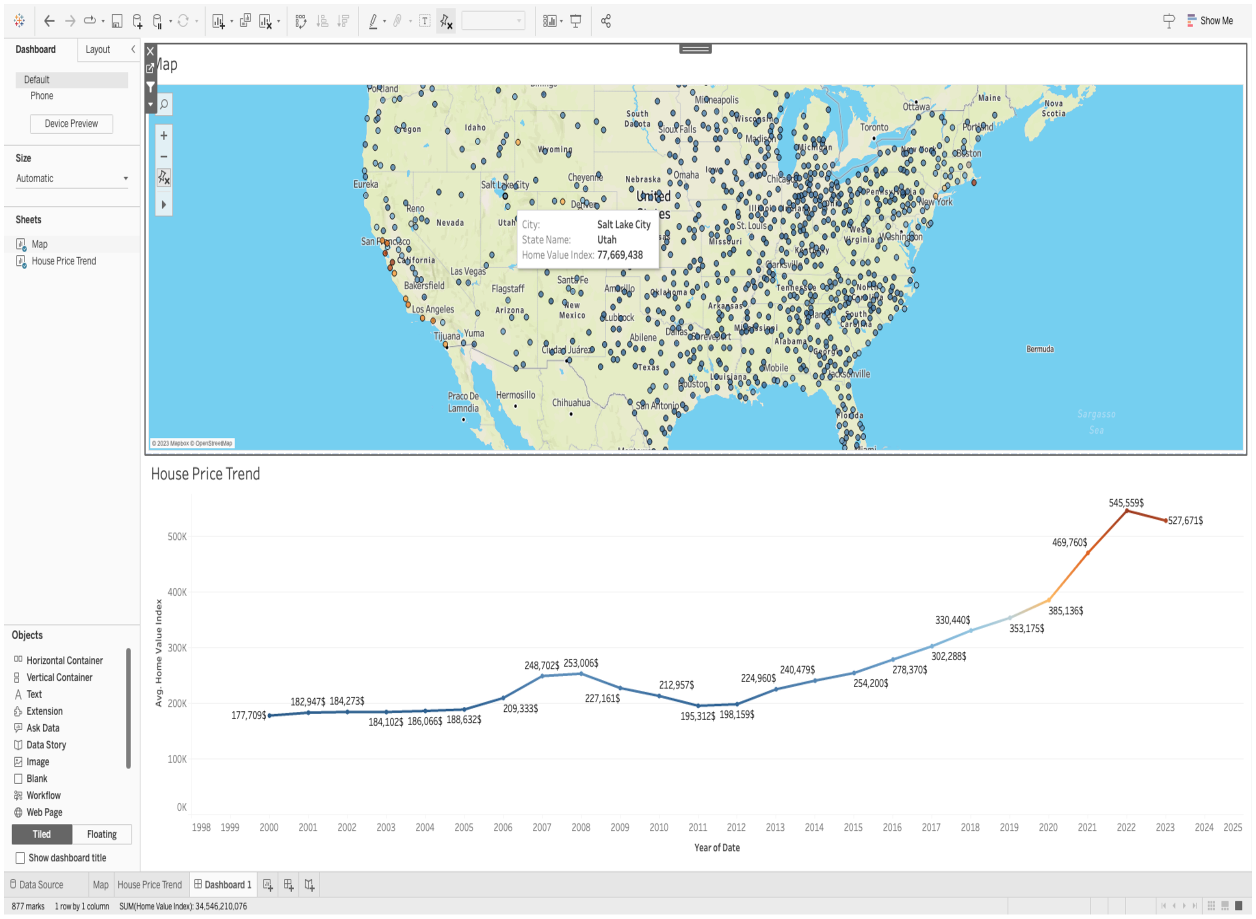
Implications:
These findings underscore the importance of model selection in machine learning tasks. They highlight that no single model is universally superior; rather, the effectiveness of a model is contingent upon the nature of the data and the specific requirements of the task. For time-series forecasting, like house price prediction, models adept at handling sequential data, such as Bi-LSTM and RNN, tend to offer more accurate and reliable predictions.

Data Visualization(Tableau):

The Tableau dashboard presents a clear picture of house prices across the United States using a cleaned-up Zillow dataset, the same dataset used for the predictions. The dataset has undergone meticulous cleaning and preparation to get better visualizations. The map shows that cities in California have some of the highest house prices, which can be seen from the many markers in that area. This tells us that California's real estate market is indeed one of the priciest.

To make the dashboard even better, there's an idea to add a line chart that changes when you click on a place on the map. For example, if someone clicks on a city, they can see how house prices there have gone up or down over the years. This chart would make it easy for people to understand the history of house prices in different places.

The dashboard is not just about what house prices have been; it also has the potential to show where they might go in the future using predictions from machine learning models. This could be very helpful for people looking to buy a house, invest in property, or make policies related to housing. The goal of the dashboard is to make it easy for anyone to get insights into the housing market, whether they are just curious or need the information for important decisions.



City: Salt Lake City
State Name: Utah
Home Value Index: 77,669,438

877 marks 1 row by 1 column SUM(Home Value Index): 34,546,210,076

.