Forecasting Future Home Prices

Zayd Ghaffar

Problem Statement

- Use the real estate market to close the wealth gap

- Develop a Machine Learning model that can project potential wealth both short and long term on any given property

- Use AI models and machine learning to explore housing data

Blueprint

1. Define the Problem:
    - Use the real estate market to close the wealth gap. Develop a Machine Learning model that can project potential wealth both short and long term on any given property
2. Acquire Data:
    - Gather historical data on house prices that is compatible with a time series model. The ideal data would give the historical price of a property throughput the years. The more data the better.
3. Clean and Prepare the Data:
    - Clean the data by handling missing values, removing duplicates, and correcting errors.
4. Choose a Model:
    - Based on the data and the nature of your prediction, choose an appropriate machine learning model. Train the Model:
    - Split your data into training and testing sets to evaluate the performance of your model.
    - Train your model using the training set.
5. Refine the Model:
    - Adjust parameters, consider adding or removing features, or try different algorithms based on the performance.

Acquiring Data

The first step in my project involved acquiring the necessary data. After extensive research, it became clear that predicting future property prices requires substantial historical data. While this type of data was hard to find, I managed to access it through a paid API on RapidAPI. This API provided monthly Zestimate prices for various properties, some dating back as far as 10 years, offering a robust dataset for our analysis. The next step in my project is to clean and prepare this data for further analysis.

Cleaning the Data

When I initially downloaded the CSV file containing our data, I intended to use the MLS number as a unique identifier for each property, along with columns for the price and date—essential components for our time series model. However, upon reviewing the file, I noticed several issues with the formatting. Specifically, the column names included unnecessary spaces, and there was a timestamp column filled with random numbers. Although this timestamp could serve as a unique identifier, I preferred to use the MLS ID as the primary key and decided to remove the timestamp column.

Additionally, I renamed the columns to eliminate the spaces, ensuring a cleaner dataset. During the data cleaning process, I also removed records for properties that only had one year of data. Given our goal of predicting long-term property prices, a single year's data seemed insufficient for reliable forecasting. This step involved filtering out those entries from our CSV file to focus on more robust, long-term datasets.

```python
import pandas as pd

df = pd.read_csv('zestimate_history_unclean.csv')

df.drop(columns=[' Timestamp170619808', '161772', '2014-03', '1396249200000'], axis=1, inplace=True)

df.columns = [col.strip().lower().replace(' ', '_') for col in df.columns]

filtered_df = df.groupby('mls#').filter(lambda x: len(x) > 12)

filtered_df.to_csv('Clean_Data.csv', index=False)
```

UNCLEAN DATA

code > zestimate_history_unclean.csv > data

```
1  MLS#, Price, Date, Timestamp170619808,161772,2014-03,1396249200000
2  170619808,156109,2014-04,1398841200000
3  170619808,157468,2014-05,1401519600000
4  170619808,154984,2014-06,1404111600000
5  170619808,154232,2014-07,1406790000000
6  170619808,157851,2014-08,1409468400000
7  170619808,158080,2014-09,1412060400000
8  170619808,155565,2014-10,1414738800000
9  170619808,155767,2014-11,1417Z334400000
10 170619808,153982,2014-12,1420012800000
11 170619808,154985,2015-01,1422691200000
```

CLEANED DATA

code > Clean_Data.csv > data

```
1  mls#,price,date
2  170619808,156109,2014-04
3  170619808,157468,2014-05
4  170619808,154984,2014-06
5  170619808,154232,2014-07
6  170619808,157851,2014-08
7  170619808,158080,2014-09
8  170619808,155565,2014-10
9  170619808,155767,2014-11
10 170619808,153982,2014-12
11 170619808,154985,2015-01
12 170619808,153697,2015-02
```

Neural Prophet

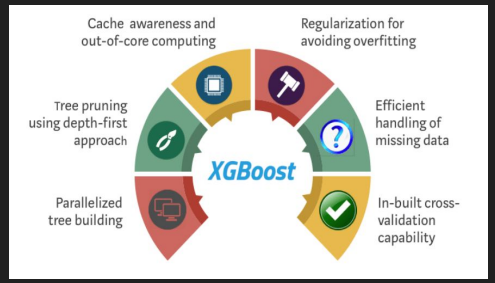
Initially, I chose Neural Prophet for our project due to its minimal setup requirements. With just about 30 lines of code, it allowed us to quickly generate good predictions, making it an efficient choice for obtaining rapid results without the need for complex data science expertise. However, as the project progressed, we encountered certain limitations with the framework. Specifically, Neural Prophet proved too rigid when we tried to add feature engineering, such as adding feature weights. Neural Prophet restricts us from adding weights so I tried alternative methods such as creating duplicating rows which was also restricted. These limitations highlighted the need for a more flexible modeling approach to accommodate our specific analytical requirements.

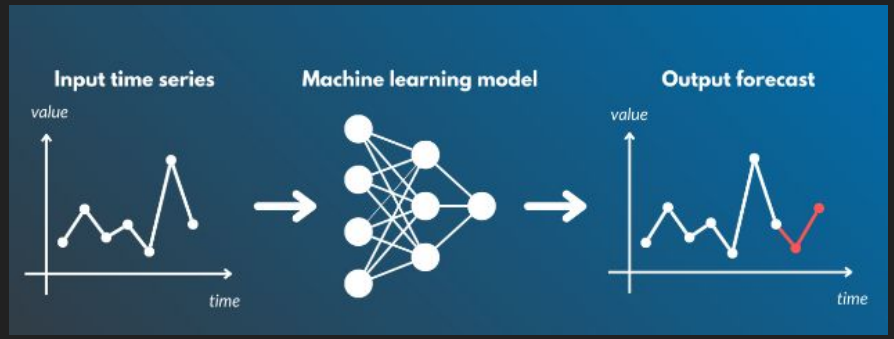
Results:

```
Current Price of MLS# 170619808 = 415,000
Predicted price in 1 year for MLS# 170619808: $391330.41
Predicted price in 5 years for MLS# 170619808: $580911.41
```

XGBoost

For my official model I opted for XGBoost which is a gradient boosting framework that uses a decision tree-based learning algorithm and is implemented in a way that is efficient for both computational speed and model performance. It is developed with both scalability and speed in mind, which allows it to handle large datasets with ease. The primary reason for choosing XGBoost is its efficiency and versatility. It can process complex data structures and uncover intricate patterns within the data, which is crucial for the real estate prediction model. XGBoost's ability to manage seasonal changes and trends and allow for the users to add feature engineering, makes it an excellent choice.





Refinements

The code features a sophisticated approach to feature engineering, prioritizing the creation of lag and cyclic features to enhance the predictive capabilities of the real estate pricing model. Lag features enable the model to capture historical price trends by incorporating past data as separate input features, which is essential for recognizing patterns over time. Additionally, cyclic features, derived from the sine transformation of months, help the model account for seasonal variations in property prices, ensuring that the forecasts reflect typical annual fluctuations in the market. These engineered features, combined with decay weights that emphasize recent data, significantly improve the model's accuracy and responsiveness to market dynamics, making it a powerful tool for predicting future real estate prices with a greater degree of precision.

XGBoost Results

This code is particularly effective for predicting real estate prices because it allows for custom feature engineering, unlike libraries like Neural Prophet which have limited options. By using lag and cyclic features, the model captures detailed patterns and seasonal trends, resulting in more accurate and reliable forecasts.

```
e/statxgboost.py
Enter MLS ID of the property: 170619808
c:\Users\Zaydg\Downloads\Projects\code\statxgboost.py:16: Futu
sion. Use obj.ffill() or obj.bfill() instead.
  house_data = house_data.asfreq('MS').fillna(method='ffill')
Optimal n_lags: 30, Optimal Decay Factor: 0.1
Predicted price in 1 year for MLS# 170619808: $278151.06
Predicted price in 3 years for MLS# 170619808: $302891.91
Predicted price in 5 years for MLS# 170619808: $419171.50
```

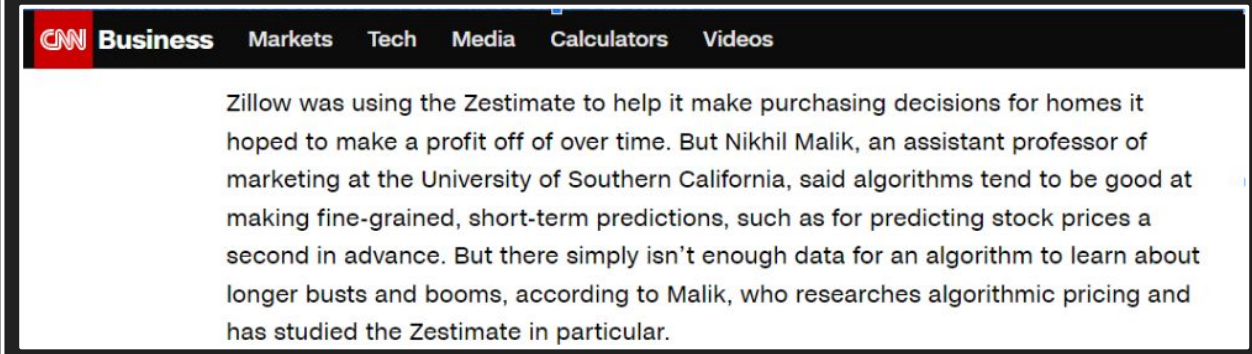
So how do we know our model is accurate?

To gauge the accuracy of predictions for future predictions you typically use historical data to test the model. This involves dividing your data into a training set and a test set. You train your model on the training set and then use the test set to make predictions. By comparing these predictions with the actual values in the test set, you can measure how well your model might perform on similar future data. Common metrics for this include mean squared error, mean absolute error, or R-squared, which provide quantifiable measures of the model's predictive accuracy.

HOWEVER

The market is extremely volatile, influenced by a multitude of unpredictable factors including economic shifts, policy changes, natural disasters, or global events like the COVID-19 pandemic. Even sophisticated models like Zillow's Zestimate, which is backed by extensive data and advanced algorithms, struggle with accuracy due to these unforeseen circumstances. While these models can leverage historical data to generate useful estimates and identify trends, they often fail to account for sudden market dynamics or unique changes to individual properties, such as renovations. Therefore, predictions from such models should be viewed as informative tools that aid in decision-making, rather than definitive forecasts, acknowledging their limitations in predicting the future with absolute certainty.

Nikhil Malik, an assistant professor of marketing at the University of Southern California, said algorithms tend to be good at making fine-grained, short-term predictions, such as for predicting stock prices a second in advance. But there simply isn't enough data for an algorithm to learn about longer busts and booms, according to Malik, who researches algorithmic pricing and has studied the Zestimate in particular.



The Issues with AI predicting Real Estate

In the screenshot below, I applied my model to a property currently listed at $350k, yet the model predicted its value at $100k for both one and five years ahead, demonstrating a clear discrepancy. The reason for this significant variance lies in the historical data; from 2016 to early 2024, the property was valued between $60k and $100k. However, its assessed value jumped to $336k the following month, likely due to renovations or other significant improvements. Such changes are difficult to track systematically, as data on renovations or similar substantial updates isn't typically available in standard real estate datasets. This case illustrates the limitations of AI in real estate valuation; without specific, timely data on property enhancements, predictive models can miss sudden increases in value, which shows the inherent challenge in using AI to accurately predict home prices.

```
Enter MLS ID of the property: 170619797
c:\Users\Zaydg\Downloads\Projects\code\statxgboost.py:16: Fut
sion. Use obj.ffill() or obj.bfill() instead.
  house_data = house_data.asfreq('MS').fillna(method='ffill')
Optimal n_lags: 48, Optimal Decay Factor: 0.1
Predicted price in 1 year for MLS# 170619797: $101209.61
Predicted price in 3 years for MLS# 170619797: $90840.91
Predicted price in 5 years for MLS# 170619797: $95432.59
```

```
mls#,price,date
170619797,94516,2021-04
170619797,92400,2021-08
170619797,93000,2021-09
170619797,93000,2021-10
170619797,92300,2021-11
170619797,88600,2021-12
170619797,98900,2022-03
170619797,118300,2022-07
170619797,90200,2022-09
170619797,121800,2022-10
170619797,127900,2022-11
170619797,120100,2022-12
170619797,113600,2023-01
170619797,113000,2023-03
170619797,92500,2023-06
170619797,95300,2023-07
170619797,94800,2023-08
170619797,93700,2023-09
170619797,99300,2023-10
170619797,100000,2023-11
170619797,93600,2023-12
170619797,336000,2024-03
```

Lessons Learned

For my machine learning class project, I chose to develop a real estate price prediction model, fully aware of the inherent challenges in achieving perfect accuracy due to the volatility and unpredictability of the market. The primary motivation behind this choice was to enhance my understanding of complex market dynamics and to refine my skills in applying advanced machine learning techniques. This project provided an excellent opportunity to work with real-world data, tackling practical problems that professionals face in the field. Furthermore, by attempting to model such a challenging dataset, I aimed to improve my analytical skills and gain insights into handling uncertainties and variability in data, which are valuable competencies in any data-driven profession. The project also allowed me to explore innovative approaches in data preprocessing and feature engineering, contributing to my overall growth as a data scientist. In essence, this project was not just about achieving perfect predictions, but about learning and applying the principles of machine learning in a meaningful and challenging context.

THANK YOU