HOUSE PRICING PREDICTION

Project Report

By:

Sudeep Raj
Section: K21ML
Roll.No: 23

Prabhat
Section: K21ML
Roll. No: 24

Department of Intelligent Systems
School of Computer Science Engineering
Lovely Professional University, Jalandhar
November-2023

Student Declaration

This is to declare that this report has been written by me. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I aware that if any part of the report is found to be copied, I will take full responsibility for it.

Signature:
Name: Sudeep Raj
Roll.No: 23

Place: Lovely Professional University
Date: 31 October-2023

TABLE OF CONTENTS

BONAFIDE CERTIFICATE

Certified that this project report "**House Pricing Estimation**" is the bonafide work of "**Sudeep Raj and Prabhat**" who carried out the project work under my supervision.

Signature of the Supervisor:
Name: Dr. Dhanpratap Singh
ID:25706
Department:
Computer Science Engineering

Objective:

The objective of house pricing prediction is to accurately estimate the value of a residential property (house) based on various factors and features associated with it. This is typically done in the context of real estate, and it serves several purposes:

1. **Buying and Selling:** Buyers and sellers use house pricing predictions to make informed decisions. For buyers, it helps them understand if a property is reasonably priced, while sellers can set a competitive asking price.
2. **Investment:** Real estate investors use pricing predictions to identify properties with good potential for appreciation. They aim to buy properties at a lower price and sell them at a higher price in the future.
3. **Mortgage Lending:** Lenders use house pricing predictions to assess the value of a property when considering mortgage applications. This helps determine the amount of the loan and the terms.
4. **Insurance:** Insurance companies use property value estimates to calculate homeowners' insurance premiums. The property's value influences the cost of coverage.
5. **Property Taxation:** Local governments use property values to assess property taxes. Accurate pricing predictions can help ensure fair taxation.

To achieve the objective of house pricing prediction, data analysis and machine learning techniques are often employed. Key features considered in these predictions may include the property's location, size, age, condition, the number of bedrooms and bathrooms, neighborhood characteristics, and recent sales of similar properties (comps). Advanced methods like regression analysis, artificial neural networks, and other machine learning algorithms can be used to build predictive models.

The accuracy of these predictions is critical, as both overestimating and underestimating property values can have financial implications for buyers, sellers, and investors. Therefore, the objective is to develop models that provide reliable estimates of house prices based on the available data.

History:

The history of house pricing prediction is closely linked to the development of real estate markets, data analysis, and statistical modeling techniques. Here is a brief overview of the history of house pricing prediction:

1. **Manual Appraisals (Pre-20th Century):** Before the advent of modern data analysis and computing, property valuations were largely done manually. Appraisers would consider factors such as location, size, condition, and comparable sales in the area to estimate a property's value.
2. **Emergence of Real Estate Markets (Late 19th Century):** As real estate markets began to develop in the late 19th century, there was a growing need for more systematic methods of property valuation. This led to the formalization of appraisal methods and the establishment of appraisal organizations.
3. **Use of Simple Regression Models (20th Century):** In the 20th century, statistical methods like simple linear regression were introduced to predict house prices. These early models considered a limited set of variables, and calculations were typically performed manually or with basic calculators.
4. **Computerization and Data Availability (Late 20th Century):** The advent of computers and databases revolutionized the field of house pricing prediction. Appraisers and real estate professionals gained access to vast amounts of data, which allowed for more sophisticated modeling. Automated valuation models (AVMs) were developed to estimate property values based on statistical analyses of historical sales data.
5. **Machine Learning and Advanced Models (21st Century):** In the 21st century, the field of house pricing prediction has been transformed by machine learning and advanced data analysis techniques. Machine learning models, such as random forests, gradient boosting, and neural networks, have become popular for predicting house prices. These models can analyze a wide range of features, including property characteristics, neighborhood data, and economic indicators, to make more accurate predictions.
6. **Online Real Estate Platforms (2000s - Present):** Online real estate platforms like Zillow, Redfin, and Realtor.com have played a significant role in house pricing prediction. These platforms offer automated estimates of property values, often using machine learning models and providing users with a quick estimate of a property's worth.
7. **Challenges and Controversies:** House pricing prediction has not been without challenges and controversies. Issues related to bias, data quality, and the influence of real estate market dynamics on predictions have been subjects of concern. Additionally, some experts argue that overreliance on automated models can lead to inaccuracies and disputes over property values.
8. **Ongoing Advancements:** House pricing prediction continues to evolve, with ongoing advancements in data collection, model development, and the

incorporation of additional data sources, such as satellite imagery and social media sentiment analysis, to improve accuracy.

The history of house pricing prediction reflects the broader evolution of data-driven decision-making in the real estate industry, with a shift from manual appraisals to increasingly sophisticated and data-driven modeling techniques.

What is House Pricing

House pricing refers to the determination of the monetary value or price associated with a residential property, such as a house or a condominium. It involves assessing various factors and features related to the property to determine its market value. The price of a house is typically influenced by a combination of factors, including:

1. Location: The geographic location of a property significantly impacts its price. Proximity to amenities, schools, transportation, and desirable neighborhoods can lead to higher property values.

2. Size and Layout: The size of the house, the number of bedrooms, bathrooms, and the overall layout can affect the price. Larger properties with more rooms often have higher values.

3. Condition: The condition of the house, including any necessary repairs or renovations, can impact its price. Well-maintained properties are generally worth more.

4. Age: The age of the house can be a factor. Older homes may have lower prices unless they have been renovated or restored.

5. Amenities and Features: Special features like swimming pools, fireplaces, smart home technology, or energy-efficient appliances can add value to a property.

6. Market Conditions: The state of the real estate market at a given time can influence house prices. In a seller's market, prices may be higher due to high demand, while in a buyer's market, prices may be more competitive.

7. Neighborhood and Community: The quality of the neighborhood, crime rates, school district, and community amenities can affect property values.

8. Comparable Sales (Comps): Recent sales of similar properties in the same area are used to establish a baseline for property pricing. These sales, known as "comparables" or "comps," provide a benchmark for assessing a property's value.

House pricing is a critical aspect of the real estate industry, and accurately estimating a property's value is essential for both buyers and sellers. Various methods, including appraisals by professionals, automated valuation models (AVMs), and machine learning models, are used to determine house prices, taking into account these factors and others to provide a fair and accurate valuation of a property.

Importance of House Pricing Estimation

The importance of house pricing is significant in the real estate industry and extends to various aspects of the economy and individuals' financial well-being. Here are some key reasons highlighting the importance of house pricing:

1. Real Estate Market Stability: House pricing plays a central role in maintaining stability within the real estate market. Accurate pricing ensures that properties are fairly valued, reducing the likelihood of housing bubbles or crashes.

2. Investment Decision Making: For real estate investors, house pricing is crucial. It guides them in identifying properties with the potential for appreciation and making informed investment decisions.

3. Homeownership: Accurate house pricing affects the ability of individuals and families to purchase homes. An inflated price may limit access to homeownership, while an undervalued property may result in lost equity.

4. Mortgage Lending: Financial institutions rely on property valuations to determine the terms and amount of mortgages. Accurate pricing ensures that homebuyers receive appropriate loan amounts and interest rates.

5. Insurance Premiums: Insurance companies use property values to calculate homeowners' insurance premiums. Accurate pricing helps homeowners secure the right coverage at the right cost.

6. Property Taxation: Local governments use property values to assess property taxes. Fair and accurate pricing ensures that taxpayers contribute equitably to community services.

7. **Seller's Decisions: For individuals selling their homes, house pricing is vital. A well-priced property can lead to a faster sale and a better return on investment.**

8. Buyer's Decisions: Prospective homebuyers rely on property valuations to assess affordability and make offers that reflect the market value.

9. Economic Indicator: The real estate market, including house pricing trends, is often used as an economic indicator. Fluctuations in house prices can signal changes in the economy.

10. Wealth Accumulation: For many individuals, their home is a significant source of wealth. House pricing trends directly impact an individual's net worth and long-term financial stability.

11. Community Planning: Local governments and urban planners use house pricing data to make informed decisions about zoning, land use, and infrastructure development.

12. Market Efficiency: Accurate pricing promotes market efficiency by matching buyers and sellers at prices that reflect the true value of properties.

13. Housing Affordability: House pricing impacts housing affordability, which is a critical concern in many regions. Accurate pricing helps balance supply and demand, ultimately influencing affordability.

14. Real Estate Transactions: The entire process of buying or selling a property is dependent on accurate pricing. It helps facilitate smooth and transparent real estate transactions.

15. Investor Confidence:Accurate house pricing fosters investor confidence in the real estate market, which can lead to increased investments in property development and housing-related industries.

In summary, house pricing is a cornerstone of the real estate market and has wide-reaching implications for individuals, businesses, and the overall economy. Accurate and transparent pricing practices are essential to ensure fairness, stability, and efficiency in the housing market.

Project Description:

Project Description: Housing Price Prediction

Project Overview: The housing price prediction project aims to develop a machine learning model that accurately estimates the prices of residential properties based on various features and factors. This project can be valuable for real estate professionals, homebuyers, sellers, investors, and financial institutions involved in mortgage lending and insurance. The model will take into account data about individual properties, neighborhoods, economic indicators, and historical sales records to make predictions.

Key Components:

1. **Data Collection:** Gather comprehensive data related to residential properties, including features like size, location, number of bedrooms and bathrooms, condition, and recent sales data. Additionally, collect neighborhood information, such as crime rates, school quality, and proximity to amenities, as these can significantly impact property values.
2. **Data Preprocessing:** Clean and preprocess the collected data. This step may involve handling missing values, outliers, and encoding categorical variables. It's essential to ensure data quality and consistency for accurate predictions.
3. **Feature Engineering:** Create new features or transform existing ones to capture valuable information. For example, calculate the price per square foot, create distance features to important amenities, or engineer features that represent the age of the property.
4. **Exploratory Data Analysis (EDA):** Perform EDA to gain insights into the data. Visualize the relationships between different features and the target variable (house prices) to identify patterns and correlations.
5. **Model Selection:** Choose appropriate machine learning algorithms for the task. Common choices include linear regression, decision trees, random forests, gradient boosting, and neural networks. Experiment with different models to find the one that best fits the data.
6. **Model Training:** Split the dataset into training and testing sets. Train the selected model(s) on the training data and evaluate their performance on the testing data. Employ appropriate evaluation metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).
7. **Hyperparameter Tuning:** Optimize the hyperparameters of the chosen model(s) to improve predictive accuracy. Techniques like grid search or random search can be used for hyperparameter tuning.
8. **Model Evaluation:** Assess the model's performance using cross-validation techniques and by comparing it to baseline models or industry standards.
9. **Deployment:** Once the model is trained and validated, deploy it to make real-time predictions. This could be through a web application, API, or integration with existing real estate platforms.

10. **Documentation and Reporting:** Provide clear documentation of the project, including the data sources, preprocessing steps, feature engineering, model selection, and hyperparameter tuning. Create a report summarizing the findings and the model's performance.

Ethical Considerations:

- Ensure that the data used for training and prediction is free from biases that could result in unfair pricing or discrimination.
- Transparently communicate the limitations and potential uncertainties associated with the model's predictions.
- Comply with data privacy regulations when handling sensitive information related to properties and individuals.

Future Enhancements:

- Continuous model monitoring and retraining to adapt to changing real estate market conditions.
- Integration with external data sources, such as economic indicators or satellite imagery.
- Exploring alternative modeling techniques and ensembles for improved accuracy.

This housing price prediction project is a practical application of machine learning and data analysis in the real estate industry, providing valuable insights for various stakeholders in the housing market.

House price prediction involves data analysis and machine learning, and there are several libraries in various programming languages that can be used for this task. Some of the commonly used libraries for house price prediction include:

1. Python Libraries:

a. NumPy: NumPy is essential for numerical operations and array manipulations, making it a fundamental library for data preprocessing and feature engineering.

b. Pandas: Pandas is used for data manipulation and analysis. It allows you to load, clean, and preprocess data efficiently.

c. Scikit-Learn: Scikit-Learn is a popular machine learning library that provides tools for data preprocessing, model selection, training, and evaluation. It includes regression algorithms for house price prediction, such as linear regression, decision trees, and random forests.

d. XGBoost and LightGBM: These gradient boosting libraries are known for their high predictive accuracy and are often used in house price prediction tasks.

e. TensorFlow and Keras: If you're considering deep learning models, TensorFlow and Keras are powerful libraries for building and training neural networks.

f. StatsModels: This library is particularly useful for performing statistical analysis and generating detailed statistics about the regression models.

g. Seaborn and Matplotlib: These libraries are used for data visualization, helping you gain insights from your data and visualize the model's performance.

h. Flask or Django: If you plan to deploy your house price prediction model as a web application, Flask or Django can be used to create the backend of your application.

2. R Libraries:

a. Tidyr and Dplyr: These libraries are commonly used for data wrangling and manipulation in R.

b. Caret: Caret is an R package that provides a unified interface for training and evaluating various machine learning models, including regression models for house price prediction.

c. RandomForest and XGBoost:These packages in R are equivalent to their Python counterparts and are used for building ensemble models.

3. Java Libraries:

a. Weka: Weka is a popular data mining and machine learning library in Java. It provides a graphical user interface (GUI) for model building and evaluation.

b. Apache Spark MLlib:If you are dealing with big data, Apache Spark MLlib is a distributed machine learning library that can be used for house price prediction on large datasets.

4. MATLAB:

a. MATLAB's Statistics and Machine Learning Toolbox: MATLAB provides a rich set of tools for data analysis and machine learning, including regression models that can be used for house price prediction.

These libraries can be chosen based on your preferred programming language, the

specific machine learning algorithms you want to use, and your familiarity with the tools. Python is one of the most popular choices due to its extensive ecosystem of libraries and its widespread adoption in the data science and machine learning community.

Working:

Step-1:

```
In [2]: import pandas as pd
        fed_files = ["MORTGAGE30US.csv", "RRVRUSQ156N.csv", "CPIAUCSL.csv"]
        dfs = [pd.read_csv(f, parse_dates=True, index_col=0) for f in fed_files]

In [3]: dfs[0]
```

Out[3]:

DATE	MORTGAGE30US
1971-04-02	7.33
1971-04-09	7.31
1971-04-16	7.31
1971-04-23	7.31
1971-04-30	7.29
...	...
2022-07-14	5.51
2022-07-21	5.54
2022-07-28	5.30
2022-08-04	4.99
2022-08-11	5.22

2681 rows × 1 columns

Step-2:

```
In [4]: fed_data = pd.concat(dfs, axis=1)

In [5]: fed_data
```

Out[5]:

DATE	MORTGAGE30US	RRVRUSQ156N	CPIAUCSL
1947-01-01	NaN	NaN	21.48
1947-02-01	NaN	NaN	21.62
1947-03-01	NaN	NaN	22.00
1947-04-01	NaN	NaN	22.00
1947-05-01	NaN	NaN	21.95
...	...	...	...
2022-07-14	5.51	NaN	NaN
2022-07-21	5.54	NaN	NaN
2022-07-28	5.30	NaN	NaN
2022-08-04	4.99	NaN	NaN
2022-08-11	5.22	NaN	NaN

3507 rows × 3 columns

Step-3:

```
In [6]: fed_data = fed_data.ffill().dropna()
```

```
In [7]: fed_data
```

Out[7]:

DATE	MORTGAGE30US	RRVRUSQ156N	CPIAUCSL
1971-04-02	7.33	5.3	40.100
1971-04-09	7.31	5.3	40.100
1971-04-16	7.31	5.3	40.100
1971-04-23	7.31	5.3	40.100
1971-04-30	7.29	5.3	40.100
...	...	...	...
2022-07-14	5.51	5.6	295.271
2022-07-21	5.54	5.6	295.271
2022-07-28	5.30	5.6	295.271
2022-08-04	4.99	5.6	295.271
2022-08-11	5.22	5.6	295.271

3215 rows × 3 columns

Step-4:

```
In [8]: zillow_files = ["Metro_median_sale_price_uc_sfrcondo_week.csv", "Metro_zhvi_uc_sfrcondo_tier_0.33_0.67_month.csv"]

        dfs = [pd.read_csv(f) for f in zillow_files]
```

```
In [9]: dfs[0]
```

Out[9]:

	RegionID	SizeRank	RegionName	RegionType	StateName	2008-02-02	2008-02-09	2008-02-16	2008-02-23	2008-03-01	...	2022-05-07	2022-05-14	2022-
0	102001	0	United States	Country	NaN	190000.0	190000.0	193000.0	189900.0	194900.0	...	369900.0	370000.0	370
1	394913	1	New York, NY	Msa	NY	400000.0	418250.0	420000.0	420000.0	400000.0	...	550000.0	555000.0	550
2	753899	2	Los Angeles-Long Beach-Anaheim, CA	Msa	CA	497500.0	515000.0	520000.0	525000.0	498250.0	...	914000.0	925000.0	925
3	394463	3	Chicago, IL	Msa	IL	245000.0	245000.0	251000.0	255000.0	255000.0	...	315000.0	310000.0	315
4	394514	4	Dallas-Fort Worth, TX	Msa	TX	144250.0	148900.0	139000.0	143700.0	145900.0	...	422000.0	430000.0	430
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
79	394528	90	Daytona Beach, FL	Msa	FL	NaN	170000.0	182400.0	170000.0	170000.0	...	340500.0	345000.0	327
80	394531	91	Des Moines, IA	Msa	IA	138000.0	160000.0	150000.0	151750.0	154500.0	...	270000.0	285250.0	295
81	395006	100	Provo, UT	Msa	UT	NaN	206000.0	215500.0	210000.0	210000.0	...	527000.0	540000.0	539
82	394549	104	Durham, NC	Msa	NC	210000.0	170000.0	170500.0	197500.0	180000.0	...	456000.0	450000.0	428
83	394602	159	Fort Collins, CO	Msa	CO	NaN	255000.0	205000.0	229900.0	235000.0	...	575000.0	585250.0	575

84 rows × 759 columns

```
In [10]: dfs[1]
```

Out[10]:

	RegionID	SizeRank	RegionName	RegionType	StateName	1996-01-31	1996-02-29	1996-03-31	1996-04-30	1996-05-31	...	2021-10
0	102001	0	United States	Country	NaN	108641.264685	108472.728880	108532.280074	108739.077466	108986.621607	...	318648.557
1	394913	1	New York, NY	Msa	NY	188550.306900	186833.460516	186448.089063	186132.003064	186023.979536	...	573099.183
2	753899	2	Los Angeles-Long Beach-Anaheim, CA	Msa	CA	186683.041088	186015.165187	185723.797105	185700.155256	185569.298179	...	854076.999
3	394463	3	Chicago, IL	Msa	IL	147341.931571	147341.152880	146420.379418	147841.918776	148371.389735	...	288899.103
4	394514	4	Dallas-Fort Worth, TX	Msa	TX	113283.512989	113199.113561	113519.423633	114048.064719	114287.444295	...	338187.525
...	...	...	...	...	...	...	...	...	...	...	...	
908	394767	929	Lamesa, TX	Msa	TX	NaN	NaN	NaN	NaN	NaN	...	92899.038
909	753874	930	Craig, CO	Msa	CO	66532.401041	66795.083062	67178.691523	67680.532389	67149.040671	...	244845.354
910	394968	931	Pecos, TX	Msa	TX	NaN	NaN	NaN	NaN	NaN	...	169404.973
911	395188	932	Vernon, TX	Msa	TX	NaN	NaN	NaN	NaN	NaN	...	80043.974
912	394743	933	Ketchikan, AK	Msa	AK	NaN	NaN	NaN	NaN	NaN	...	341065.613

913 rows × 324 columns

```
In [11]: dfs = [pd.DataFrame(df.iloc[0,5:]) for df in dfs]
         for df in dfs:
             df.index = pd.to_datetime(df.index)
             df["month"] = df.index.to_period("M")
```

```
In [12]: dfs[0]
```

Out[12]:

	0	month
2008-02-02	190000.0	2008-02
2008-02-09	190000.0	2008-02
2008-02-16	193000.0	2008-02
2008-02-23	189900.0	2008-02
2008-03-01	194900.0	2008-03
...	...	...
2022-06-11	370000.0	2022-06
2022-06-18	375000.0	2022-06
2022-06-25	370000.0	2022-06
2022-07-02	370000.0	2022-07
2022-07-09	362500.0	2022-07

754 rows × 2 columns

In [13]:
```
price_data = dfs[0].merge(dfs[1], on="month")
```

In [14]:
```
price_data.index = dfs[0].index
```

In [15]:
```
price_data
```

Out[15]:

	0_x	month	0_y
2008-02-02	190000.0	2008-02	206885.853266
2008-02-09	190000.0	2008-02	206885.853266
2008-02-16	193000.0	2008-02	206885.853266
2008-02-23	189900.0	2008-02	206885.853266
2008-03-01	194900.0	2008-03	205459.521952
...	...	...	...
2022-06-11	370000.0	2022-06	357473.327397
2022-06-18	375000.0	2022-06	357473.327397
2022-06-25	370000.0	2022-06	357473.327397
2022-07-02	370000.0	2022-07	357107.271636
2022-07-09	362500.0	2022-07	357107.271636

754 rows × 3 columns

In [16]:
```
del price_data["month"]
price_data.columns = ["price", "value"]
```

In [17]:
```
price_data
```

Out[17]:

	price	value
2008-02-02	190000.0	206885.853266
2008-02-09	190000.0	206885.853266
2008-02-16	193000.0	206885.853266
2008-02-23	189900.0	206885.853266
2008-03-01	194900.0	205459.521952
...	...	...
2022-06-11	370000.0	357473.327397
2022-06-18	375000.0	357473.327397
2022-06-25	370000.0	357473.327397
2022-07-02	370000.0	357107.271636
2022-07-09	362500.0	357107.271636

754 rows × 2 columns

In [18]:
```
from datetime import timedelta

fed_data.index = fed_data.index + timedelta(days=2)
```

In [19]:
```
fed_data.tail(10)
```

Out[19]:

DATE	MORTGAGE30US	RRVRUSQ156N	CPIAUCSL
2022-06-18	5.78	5.6	295.328
2022-06-25	5.81	5.6	295.328
2022-07-02	5.70	5.6	295.328
2022-07-03	5.70	5.6	295.271
2022-07-09	5.30	5.6	295.271
2022-07-16	5.51	5.6	295.271
2022-07-23	5.54	5.6	295.271
2022-07-30	5.30	5.6	295.271
2022-08-06	4.99	5.6	295.271
2022-08-13	5.22	5.6	295.271

In [20]:
```
price_data = fed_data.merge(price_data, left_index=True, right_index=True)
```

In [21]:
```
price_data
```

Out[21]:

	MORTGAGE30US	RRVRUSQ156N	CPIAUCSL	price	value
2008-02-02	5.68	10.1	212.174	190000.0	206885.853266
2008-02-09	5.67	10.1	212.687	190000.0	206885.853266
2008-02-16	5.72	10.1	212.687	193000.0	206885.853266
2008-02-23	6.04	10.1	212.687	189900.0	206885.853266
2008-03-01	6.24	10.1	212.687	194900.0	205459.521952
...	...	...	...	...	...
2022-06-11	5.23	5.6	295.328	370000.0	357473.327397
2022-06-18	5.78	5.6	295.328	375000.0	357473.327397
2022-06-25	5.81	5.6	295.328	370000.0	357473.327397
2022-07-02	5.70	5.6	295.328	370000.0	357107.271636
2022-07-09	5.30	5.6	295.271	362500.0	357107.271636

735 rows × 5 columns

In [22]:
```
price_data.columns = ["interest", "vacancy", "cpi", "price", "value"]
```

In [23]:
```
price_data
```

Out[23]:

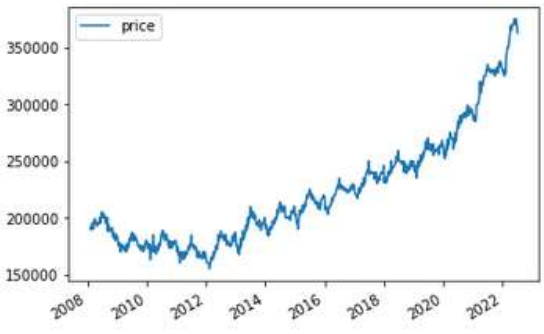
	interest	vacancy	cpi	price	value
2008-02-02	5.68	10.1	212.174	190000.0	206885.853266
2008-02-09	5.67	10.1	212.687	190000.0	206885.853266
2008-02-16	5.72	10.1	212.687	193000.0	206885.853266
2008-02-23	6.04	10.1	212.687	189900.0	206885.853266
2008-03-01	6.24	10.1	212.687	194900.0	205459.521952
...	...	...	...	...	...
2022-06-11	5.23	5.6	295.328	370000.0	357473.327397
2022-06-18	5.78	5.6	295.328	375000.0	357473.327397
2022-06-25	5.81	5.6	295.328	370000.0	357473.327397
2022-07-02	5.70	5.6	295.328	370000.0	357107.271636
2022-07-09	5.30	5.6	295.271	362500.0	357107.271636

735 rows × 5 columns

In [24]:
```
price_data["adj_price"] = price_data["price"] / price_data["cpi"] * 100
price_data["adj_value"] = price_data["value"] / price_data["cpi"] * 100
```

In [25]:
```
price_data.plot.line(y="price", use_index=True)
```

Out[25]: <AxesSubplot:>



In [26]:
```
price_data.plot.line(y="adj_price", use_index=True)
```

Out[26]: <AxesSubplot:>

In [27]:
```
price_data["next_quarter"] = price_data["adj_price"].shift(-13)
```

In [28]:
```
price_data.dropna(inplace=True)
```

In [29]:
```
price_data
```

Out[29]:

	interest	vacancy	cpi	price	value	adj_price	adj_value	next_quarter
2008-02-02	5.68	10.1	212.174	190000.0	206885.853266	89549.143627	97507.636782	90610.014498
2008-02-09	5.67	10.1	212.687	190000.0	206885.853266	89333.151533	97272.448841	90563.547824
2008-02-16	5.72	10.1	212.687	193000.0	206885.853266	90743.674978	97272.448841	91014.739229
2008-02-23	6.04	10.1	212.687	189900.0	206885.853266	89286.134084	97272.448841	90610.014498
2008-03-01	6.24	10.1	212.687	194900.0	205459.521952	91637.006493	96601.824254	92933.348203
...	...	...	...	...	...	...	...	...
2022-03-12	3.85	5.8	287.708	350000.0	344042.433111	121651.118495	119580.419422	125284.429516
2022-03-19	4.16	5.8	287.708	350000.0	344042.433111	121651.118495	119580.419422	126977.462347
2022-03-26	4.42	5.8	287.708	355000.0	344042.433111	123388.991617	119580.419422	125284.429516
2022-04-02	4.67	5.8	287.708	360000.0	350515.841789	125126.864738	121830.412011	125284.429516
2022-04-09	4.72	5.6	288.663	365000.0	350515.841789	126445.024128	121427.353623	122768.575309

In [30]:
```
price_data["change"] = (price_data["next_quarter"] > price_data["adj_price"]).astype(int)
```

In [31]:
```
price_data
```

Out[31]:

	interest	vacancy	cpi	price	value	adj_price	adj_value	next_quarter	change
2008-02-02	5.68	10.1	212.174	190000.0	206885.853266	89549.143627	97507.636782	90610.014498	1
2008-02-09	5.67	10.1	212.687	190000.0	206885.853266	89333.151533	97272.448841	90563.547824	1
2008-02-16	5.72	10.1	212.687	193000.0	206885.853266	90743.674978	97272.448841	91014.739229	1
2008-02-23	6.04	10.1	212.687	189900.0	206885.853266	89286.134084	97272.448841	90610.014498	1
2008-03-01	6.24	10.1	212.687	194900.0	205459.521952	91637.006493	96601.824254	92933.348203	1
...	...	...	...	...	...	...	...	...	...
2022-03-12	3.85	5.8	287.708	350000.0	344042.433111	121651.118495	119580.419422	125284.429516	1
2022-03-19	4.16	5.8	287.708	350000.0	344042.433111	121651.118495	119580.419422	126977.462347	1
2022-03-26	4.42	5.8	287.708	355000.0	344042.433111	123388.991617	119580.419422	125284.429516	1
2022-04-02	4.67	5.8	287.708	360000.0	350515.841789	125126.864738	121830.412011	125284.429516	1
2022-04-09	4.72	5.6	288.663	365000.0	350515.841789	126445.024128	121427.353623	122768.575309	0

722 rows × 9 columns

In [32]:
```
price_data["change"].value_counts()
```

Out[32]:
```
1    379
0    343
Name: change, dtype: int64
```

In [33]:
```
predictors = ["interest", "vacancy", "adj_price", "adj_value"]
target = "change"
```

In [34]:
```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import numpy as np

START = 260
STEP = 52

def predict(train, test, predictors, target):
    rf = RandomForestClassifier(min_samples_split=10, random_state=1)
    rf.fit(train[predictors], train[target])
    preds = rf.predict(test[predictors])
    return preds

def backtest(data, predictors, target):
    all_preds = []
    for i in range(START, data.shape[0], STEP):
        train = price_data.iloc[:i]
        test = price_data.iloc[i:(i+STEP)]
        all_preds.append(predict(train, test, predictors, target))

    preds = np.concatenate(all_preds)
    return preds, accuracy_score(data.iloc[START:][target], preds)
```

In [35]:
```python
preds, accuracy = backtest(price_data, predictors, target)
```

In [36]:
```python
preds
```

Out[36]:
```
array([1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1,
       1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

In [37]:
```
accuracy
```

Out[37]:
```
0.5952380952380952
```

In [38]:
```
yearly = price_data.rolling(52, min_periods=1).mean()
```

In [39]:
```
yearly_ratios = [p + "_year" for p in predictors]
price_data[yearly_ratios] = price_data[predictors] / yearly[predictors]
```

In [40]:
```
price_data
```

Out[40]:

	interest	vacancy	cpi	price	value	adj_price	adj_value	next_quarter	change	interest_year	vacancy_year	adj_pri
2008-02-02	5.68	10.1	212.174	190000.0	206885.853266	89549.143627	97507.636782	90610.014498	1	1.000000	1.000000	
2008-02-09	5.67	10.1	212.687	190000.0	206885.853266	89333.151533	97272.448841	90563.547824	1	0.999119	1.000000	
2008-02-16	5.72	10.1	212.687	193000.0	206885.853266	90743.674978	97272.448841	91014.739229	1	1.005272	1.000000	
2008-02-23	6.04	10.1	212.687	189900.0	206885.853266	89286.134084	97272.448841	90610.014498	1	1.045435	1.000000	
2008-03-01	6.24	10.1	212.687	194900.0	205459.521952	91637.006493	96601.824254	92933.348203	1	1.063032	1.000000	
...	...	...	...	...	...	...	...	...	...	...	...	
2022-03-12	3.85	5.8	287.708	350000.0	344042.433111	121651.118495	119580.419422	125284.429516	1	1.235955	0.977317	
2022-03-19	4.16	5.8	287.708	350000.0	344042.433111	121651.118495	119580.419422	126977.462347	1	1.326140	0.980494	
2022-03-26	4.42	5.8	287.708	355000.0	344042.433111	123388.991617	119580.419422	125284.429516	1	1.397289	0.983692	
2022-04-02	4.67	5.8	287.708	360000.0	350515.841789	125126.864738	121830.412011	125284.429516	1	1.462275	0.986911	
2022-04-09	4.72	5.6	288.663	365000.0	350515.841789	126445.024128	121427.353623	122768.575309	0	1.464264	0.956636	

In [41]:
```
preds, accuracy = backtest(price_data, predictors + yearly_ratios, target)
```

In [42]:
```
accuracy
```

Out[42]:
```
0.6536796536796536
```

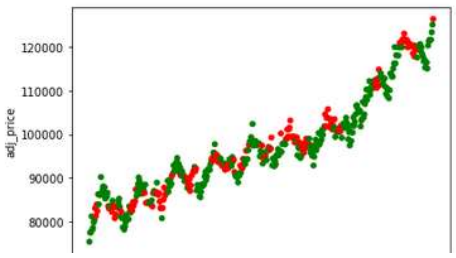
In [43]:
```
pred_match = (preds == price_data[target].iloc[START:])
pred_match[pred_match == True] = "green"
pred_match[pred_match == False] = "red"
```

In [44]:
```
import matplotlib.pyplot as plt

plot_data = price_data.iloc[START:].copy()

plot_data.reset_index().plot.scatter(x="index", y="adj_price", color=pred_match)
```

Out[44]:
```
<AxesSubplot:xlabel='index', ylabel='adj_price'>
```



In [45]:
```
from sklearn.inspection import permutation_importance

rf = RandomForestClassifier(min_samples_split=10, random_state=1)
rf.fit(price_data[predictors], price_data[target])

result = permutation_importance(rf, price_data[predictors], price_data[target], n_repeats=10, random_state=1)
```

In [46]:
```
result["importances_mean"]
```

Out[46]:
```
array([0.17451524, 0.15540166, 0.27576177, 0.34861496])
```

In [47]:
```
predictors
```

Out[47]:
```
['interest', 'vacancy', 'adj_price', 'adj_value']
```

Conclusion:

In conclusion, housing price prediction is a valuable and practical application of data analysis and machine learning in the real estate industry. It offers a wide range of benefits for various stakeholders, including homebuyers, sellers, investors, financial institutions, and local governments. Here are some key takeaways:

1. Data-Driven Decision Making: Housing price prediction leverages data to provide informed, data-driven decision-making in the real estate market. This empowers buyers and sellers to make more accurate pricing decisions.

2. Market Insights: The analysis involved in house price prediction projects offers valuable insights into the factors influencing property values, helping stakeholders understand the dynamics of the real estate market.

3. Investment Opportunities: Real estate investors can use housing price prediction to identify properties with good potential for appreciation, aiding in sound investment decisions.

4. Mortgage Lending and Insurance: Financial institutions use property value estimates to determine mortgage terms and insurance premiums, impacting the financial well-being of homeowners.

5. Fair Property Taxation: Accurate property valuations contribute to fair property taxation, ensuring that homeowners pay their fair share of taxes.

6. Challenges and Ethical Considerations: House price prediction also comes with challenges, including the potential for biases in data and predictions. Ethical considerations are important to ensure fair and unbiased pricing.

7. Continual Evolution: The field of housing price prediction continues to evolve, with advancements in data collection, modeling techniques, and integration with external data sources.

Overall, housing price prediction is a dynamic field that reflects the evolution of data-driven decision-making in the real estate industry. It offers opportunities for stakeholders to make more informed decisions and plays a crucial role in the broader real estate market.