HOUSE PRICE ANTICIPATION USING MACHINE LEARNING

A PROJECT REPORT

SUBMITTED BY

ASHISH SHARMA 21BCS10436

MANISH KUMAR YADAV 21BCS10262

VIDIT SHARMA 21BCS10469

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE WITH SPECIALIZATION IN

ARTIFICIAL INTELLIGENCE AND MACHINE

LEARNING

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Chandigarh University

AUG - NOV 2023

BONAFIDE CERTIFICATE

Certified that this project report " HOUSE PRICE ANTICIPATION USING MACHINE LEARNING " is the bonafide work of "ASHISH SHARMA (21BCS10436), MANISH KUMAR YADAV (21BCS10262), VIDIT (21BCS10469) " who carried out the project work under my/our supervision.

SIGNATURE	SIGNATURE
MR. AMAN KAUSHIK	Mrs. BHAVNA NAYYER
HEAD OF THE DEAPARTMENT	SUPERVISOR
Department of AIT	Assistant Professor
Chandigarh University	Department of AIT
Mohali, Punjab	Chandigarh University
	Mohali, Punjab

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are highly grateful to the Hon'ble Chancellor and Vice-Chancellor, Chandigarh University, Mohali, Punjab for allowing us to carry out the present project work.

The constant guidance and encouragement received from Mr. Aman Kaushik HOD, Dept. of Apex Institute Of Technology (AIT) , Chandigarh University, has been of great help in carrying out our present work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to our Project Supervisor, Mrs. Bhavna Nayyer, Assistant. Prof. without her able guidance, it would have been impossible to complete the project in this manner.

At last, I would like to extend my heartfelt thanks to my parents because without their help this project would not have been successful. Finally, I would like to thank my dear friends who have been with me all the time.

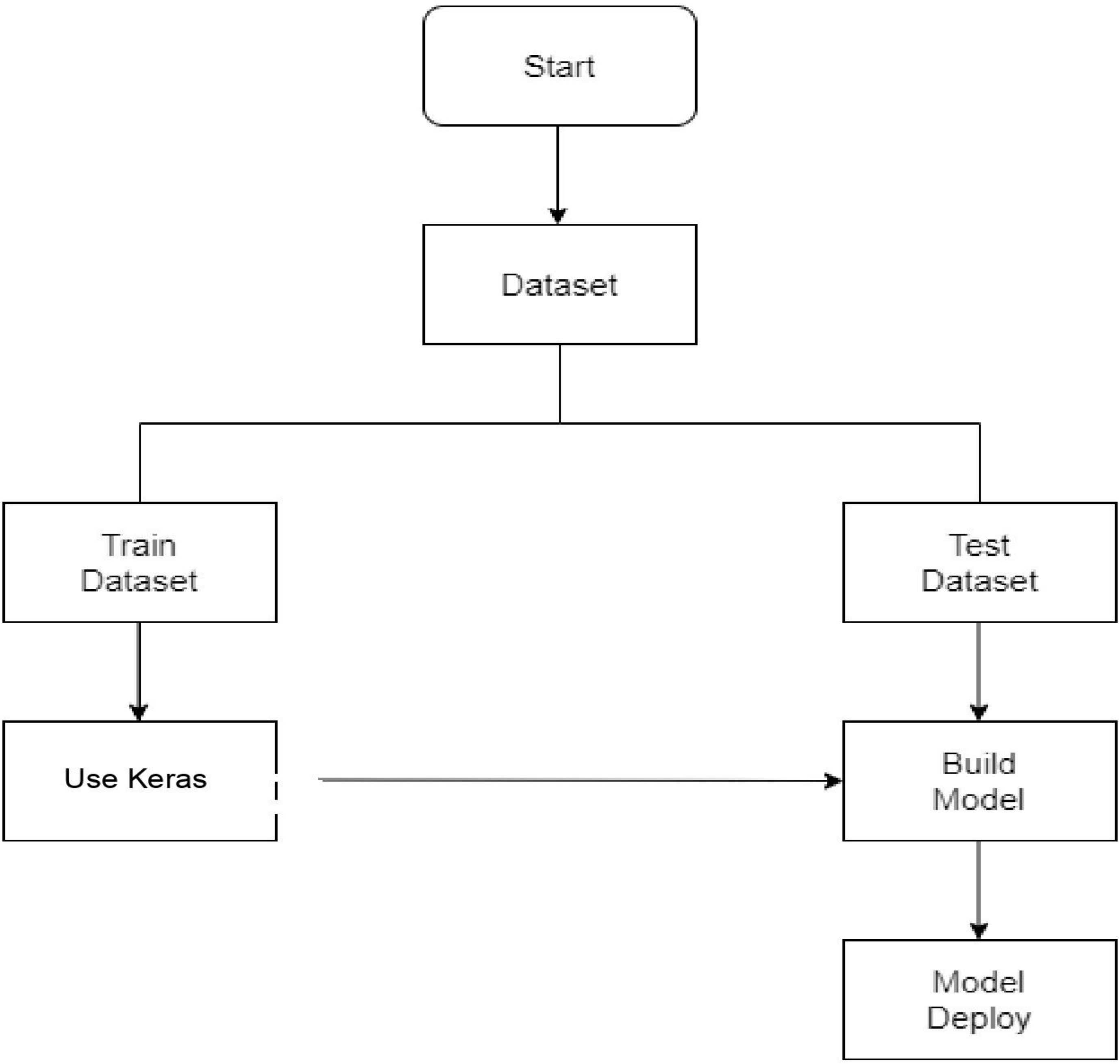
Table of Contents

List of Figures

ABSTRACT

The real estate industry has witnessed a growing interest in predictive analytics and machine learning techniques for accurate house price prediction. This abstract provides an overview of a study that employs Artificial Neural Networks (ANNs) to predict house prices. ANNs have demonstrated exceptional capabilities in handling complex, non-linear relationships in data, making them well-suited for this task. The study utilizes a dataset containing various attributes related to houses, such as square footage, number of bedrooms and bathrooms, location, and amenities. These features are preprocessed to handle missing values and outliers, and feature engineering techniques are employed to extract valuable information from the data. The network is trained using historical housing data with known prices. During training, back propagation and optimization algorithms are employed to minimize the prediction error. Hyper parameter tuning is conducted to optimize the model's performance. To assess the model's accuracy, various evaluation metrics, such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R^2), are employed. The trained ANN is capable of predicting house prices with a high degree of accuracy, outperforming traditional regression models. .

GRAPHICAL ABSTRACT



ABBREVIATIONS

ML: Machine Learning

PCA: Principal Component

Analysis Sklearn: Scikit learn

OpenCv : Open Source Computer Vision Library

CHAPTER NO. – 01

INTRODUCTION

The house price prediction problem is a fundamental and practical task within the domain of real estate, finance, and data science. It involves developing predictive models to estimate the market value of residential properties based on various factors and features. This task is essential for a wide range of stakeholders, including homeowners, real estate agents, investors, and financial institutions, as it provides valuable insights into property values and market trends.

Key Elements of the House Price Prediction Problem:

Real Estate Valuation: Property valuation is a critical aspect of the real estate industry. Accurate house price predictions are crucial for buyers and sellers to make informed decisions, for real estate agents to set appropriate listing prices, and for financial institutions to assess the value of collateral for mortgages.

Multifaceted Data: House price prediction models consider a multitude of factors that can influence property values. These factors may include property characteristics (e.g., size, location, age, condition), economic indicators (e.g., interest rates, local employment trends), and market dynamics (e.g., supply and demand).

Fig. House Price Image

Predictive Modeling: Data-driven approaches, often based on machine learning and statistical techniques, are used to build predictive models that can analyze historical property data and make informed predictions about the price of a house. These models aim to capture complex relationships between property features and their impact on price.

Data Sources: House price prediction models rely on diverse data sources, such as real estate listings, property transaction records, economic data, geographic information systems (GIS), and more. In recent years, text and image data from property listings have also been used to enhance predictions.

Challenges: Predicting house prices is a complex task with several challenges. These challenges include dealing with missing data, outliers, model interpretability, addressing potential biases and fairness concerns, and handling spatiotemporal dynamics in property markets.

Applications: House price prediction models find applications in a variety of scenarios, from assisting individuals in buying or selling homes to supporting real estate professionals in pricing properties and guiding investment decisions.

Economic Impact: The housing market is a significant component of the economy, and fluctuations in property prices can have far-reaching economic consequences. Therefore, accurate house price predictions are essential for financial stability and economic planning.

Fig. House Price Image_2

Research and Development: Researchers and data scientists continually work to improve the accuracy and fairness of house price prediction models. They explore new methods, data sources, and features to enhance the predictive power of these models.

In summary, the house price prediction problem is a critical and dynamic field at the intersection of real estate, finance, and data science. It addresses the need for accurate and actionable estimates of property values, and it plays a vital role in shaping decisions in the real estate market and broader economic landscape.

1.1 Problem Definition

House price anticipation is the process of predicting and evaluating future property values in the context of the real estate market. The real estate market is an integral part of both individual and society's economy, providing a foundation for wealth creation, investment and financial security.

House price anticipation has become a hot topic among homeowners, investors and policy makers, but it is also of great importance to economists, city planners and financial institutions trying to comprehend and navigate the intricacies of the housing market. For many years, house price forecasting has been a hot topic due to its ability to influence purchasing, selling, investment and lending decisions.

By accurately predicting house price movements, individuals and institutions can optimize their financial outcomes. However, due to the volatility and multi-factors involved in real estate markets, accurately predicting house prices can be a difficult task. Economic indicators, demographic changes, interest rates and supply and demand and geopolitical events all work together to determine house prices. As technology advances and data becomes more accessible, so too do the methods used to study and forecast house price trends.

Traditionally, house price predictions relied heavily on historical trends and basic statistical models. But now, with the help of advanced machine learning algorithms and data mining techniques, as well as big data analytics, researchers and practitioners are able to identify subtle patterns, explore complex relationships, and make more accurate predictions. This research paper dives deep into the topic of house price anticipation to answer key questions about the factors that influence property values, as well as evaluate various prediction models to build a robust framework for improving house price anticipation accuracy. It also looks at the broader implications of better predictive capabilities on individuals, financial institutions and public policy formulation.

1.2 Problem Overview

The primary objective of this research paper is to investigate and analyze the phenomenon of house price anticipation and its impact on real estate markets. The study aims to achieve the following specific objectives:

1. *Examine the Concept of House Price Anticipation:* This research seeks to provide a comprehensive understanding of the concept of house price anticipation, exploring its theoretical underpinnings and practical implications within the context of real estate economics.

2. *Identify Factors Influencing House Price Anticipation:* The study aims to identify and categorize the key factors that contribute to the anticipation of future house price movements. This includes investigating economic indicators, market trends, demographic shifts, and psychological factors that shape buyer and seller expectations.

3. *Assess the Role of Information and Media:* This research intends to analyze the role of information dissemination and media coverage in influencing house price anticipation. It will investigate how information asymmetry, media narratives, and public perceptions impact the anticipation of house price changes.

4. *Quantify the Effects on Housing Market Dynamics:* The study aims to quantify the effects of house price anticipation on housing market dynamics. This involves examining how anticipated price changes influence demand, supply, transaction volumes, and price volatility.

5. *Evaluate Economic and Societal Consequences:* This research will assess the economic and societal consequences of accurate and inaccurate house price anticipation. It will analyze how well-founded anticipations contribute to market stability, economic growth, wealth distribution, and housing affordability

1.3 Identification of Client & Need:

Client:

Homebuyers and Sellers: Individual homebuyers and sellers are often clients seeking accurate house price predictions. They need this information to make informed decisions about buying or selling their properties, setting listing prices, and negotiating offers.

Real Estate Agents and Brokers: Real estate professionals use house price predictions to assist clients in pricing their properties correctly, attracting buyers, and closing deals. Accurate predictions help them provide valuable services.

Real Estate Investors: Investors, including individuals and organizations, rely on house price predictions to identify lucrative investment opportunities, manage real estate portfolios, and estimate potential returns on investment.

Financial Institutions: Banks and mortgage lenders need accurate property valuations to assess the value of collateral for mortgage loans, manage risk, and determine loan terms. House price predictions are essential for these financial institutions.

Government and Regulatory Bodies: Government agencies may use house price predictions to assess property tax values, monitor housing market trends, and develop housing policies.

Client Needs:

Accurate Valuation: Clients require reliable and precise estimates of property values. Accurate predictions help homeowners price their properties competitively, assist buyers in making cost-effective purchases, and allow investors to maximize their returns.

Market Insights: Clients seek insights into market trends, including whether property prices are rising or falling, the impact of economic factors, and the desirability of certain locations. These insights inform their decisions.

Risk Management: Financial institutions need accurate valuations to assess the risk associated with mortgage loans. For investors, accurate predictions help in identifying low-risk investment opportunities.

Customization: Clients may have specific needs, such as the ability to customize predictions for different property types, locations, or timeframes. Customization allows them to tailor the predictions to their unique requirements.

Fairness and Transparency: Clients are increasingly concerned about fairness and transparency in property valuations. They need to ensure that the predictions are free from biases and discrimination.

Timeliness: Real-time or near-real-time predictions are valuable, especially in dynamic housing markets. Clients often need up-to-date information to act quickly and make well-informed decisions.

Data Security: Clients expect their data, including property and financial information, to be handled securely and in compliance with data privacy regulations.

1.4 Relevant Contemporary Issues:

Relevant Contemporary Issues in House Price Prediction:

Data Privacy and Security: The collection and use of sensitive data for house price prediction raise privacy and security concerns. Clients, regulators, and the public demand that data be handled with care and in compliance with data protection laws, such as GDPR in Europe and similar regulations in other regions.

Fairness and Bias: Ensuring that house price prediction models are fair and free from bias is a pressing issue. Models must not discriminate against specific demographic groups, neighborhoods, or property types. Addressing bias and achieving fairness in predictions is a key ethical concern.

Interpretability: Many stakeholders, especially real estate professionals, require transparent and interpretable models. Deep learning and complex machine learning models can be challenging to explain. Efforts to make predictions more interpretable without sacrificing accuracy are ongoing.

Real-Time Predictions: As housing markets become more dynamic, the need for real-time or near-real-time predictions is growing. Clients, especially investors, require the ability to access the most current market data for timely decision-making.

Market Volatility: Economic uncertainties and unexpected events, such as the COVID-19 pandemic, have highlighted the challenges of predicting house prices in volatile markets. Models need to be robust enough to adapt to sudden shifts in market conditions.

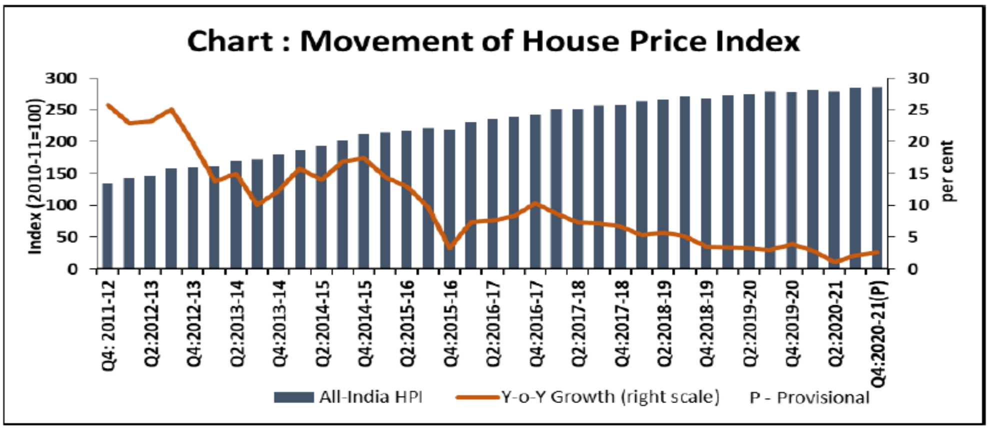


Fig. House price Issues

Blockchain and Smart Contracts: Some projects explore the integration of blockchain technology for property transactions and smart contracts. Blockchain can enhance transparency and security in real estate transactions and potentially impact price prediction methods.

Regulatory Compliance: House price prediction models must adhere to evolving regulations in the real estate and financial sectors. Compliance with regulations such as Truth in Lending Act (TILA) and Real Estate Settlement Procedures Act (RESPA) in the United States is essential.

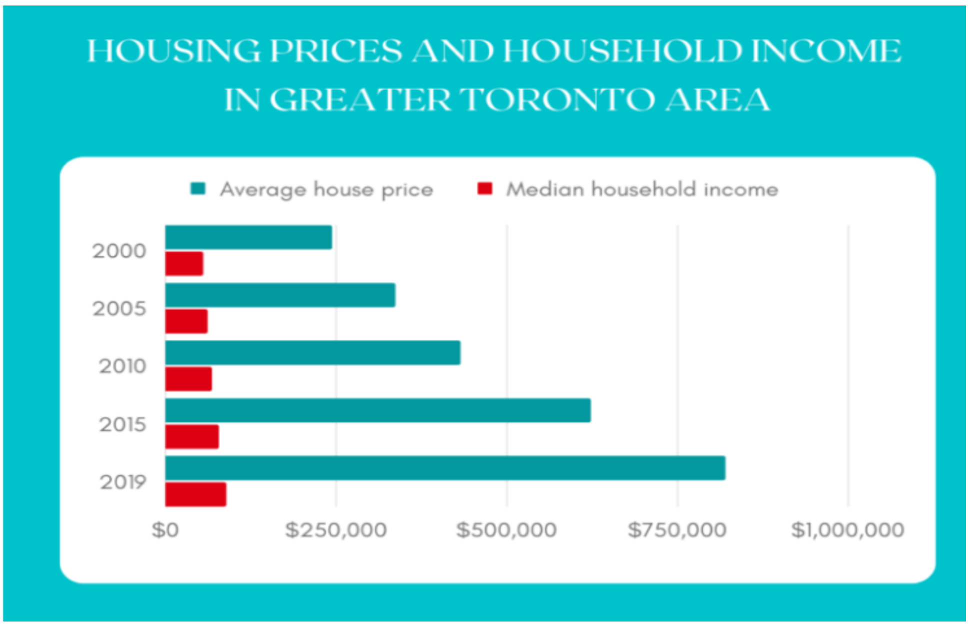
1.5 Task Identification:

House price prediction is a multifaceted task within the real estate industry, aiming to provide accurate and reliable estimates of property values. This task serves the diverse needs of stakeholders, including homebuyers, sellers, real estate professionals, investors, and financial institutions. Key tasks in house price prediction encompass data collection, preprocessing, and feature engineering to assemble a comprehensive dataset. Machine learning models are selected and trained on this data, often employing techniques like linear regression, decision trees, and deep learning. Evaluation metrics such as Mean Absolute Error and R-squared gauge model accuracy.

Interpretability of predictions is important for real estate professionals, allowing them to explain property valuations to clients. Addressing bias and ensuring fairness in predictions are vital ethical considerations. The deployment of models involves making predictions accessible through user-friendly interfaces or applications. Maintenance, updates, and the handling of contemporary issues such as data privacy, security, fairness, and the impact of market volatility remain ongoing challenges. As the real estate landscape evolves, so does the house price prediction field, necessitating continuous adaptation and innovation to meet the changing demands and complexities of the industry.

1.6 Organization of the Report:

This report will be organized into several sections, including an overview of the project, a review of the existing literature on face mask detection, a description of the methodology used to develop the system, a discussion of the results, and a conclusion summarizing the key findings and recommendations for future research. The report will also include appendices with technical details, such as code snippets and diagrams, to support the methodology and results presented.



1.7 Software Specification

Numpy

The numpy library is a popular numerical computing library for Python, which provides fast and efficient array operations. It is often used in machine learning applications, including face mask detection.Overall, numpy is used in this process to perform efficient array operations on the image data, such as resizing, normalization, and data conversion. It is also used to preprocess the output of the face mask detection model, such as converting probability scores to labels and visualizing the results.

- Pandas

The pandas library is a popular data manipulation library for Python. It provides data structures for efficiently storing and manipulating large and complex datasets, as well as tools for data analysis, cleaning, and transformation. Here are some key features and functionalities of the pandas library:

Data Structures:
pandas provides two primary data structures - Series and DataFrame - for storing and manipulating data. Series is a one-dimensional labeled array that can store any data type, while DataFrame is a two-dimensional labeled data structure that can store heterogeneous data types.

Data Cleaning and Transformation:
pandas provides a range of tools for data cleaning and transformation, including filtering, sorting, aggregating, merging, and reshaping datasets. These tools enable data analysts to efficiently clean and transform datasets for further analysis.

- Neural network

A neural network is a computational model inspired by the structure and function of the human brain. It's a type of machine learning algorithm used for tasks such as pattern recognition, classification, regression, and more. Neural networks are composed of interconnected nodes or artificial neurons organized into layers. There are three main types of layers in a neural network:

Input Layer: This layer receives the initial data and passes it to the network. Each node in this layer represents a feature or input variable.

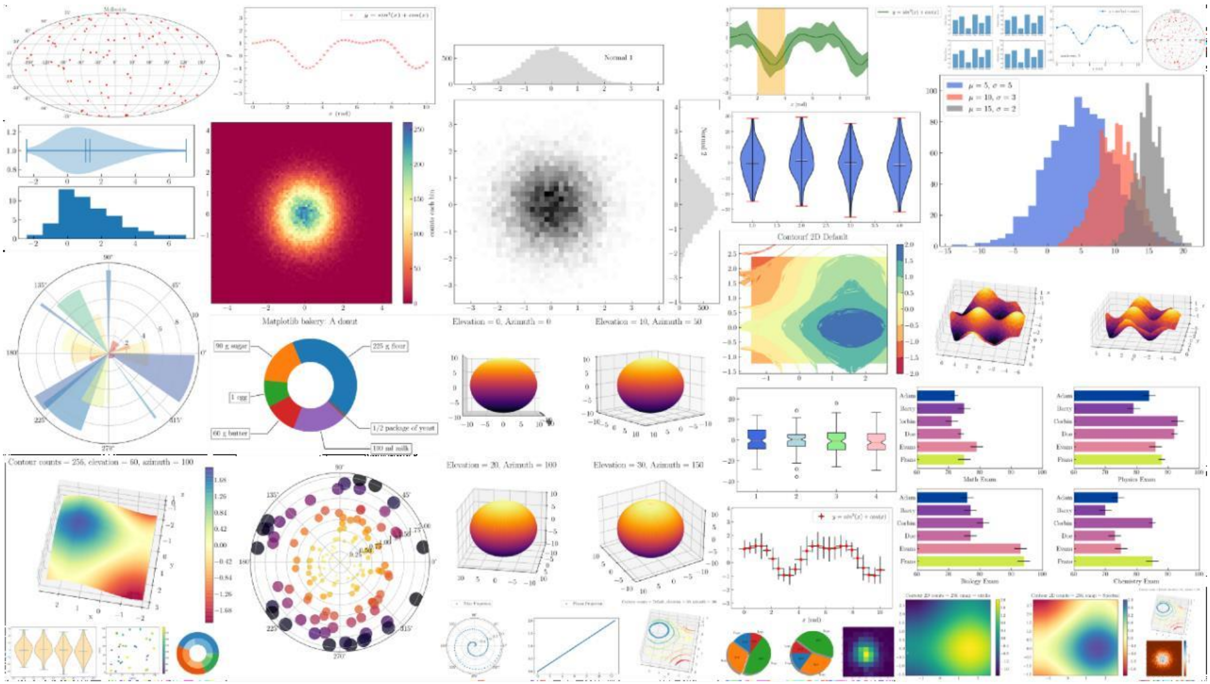
Hidden Layers: These intermediate layers, which can vary in number and size, process the data through a series of weighted connections and non-linear activation functions. The hidden layers extract patterns and relationships from the input data.

Output Layer: The final layer produces the network's predictions or classifications. The number of nodes in the output layer depends on the specific task (e.g., one node for binary classification, multiple nodes for multiclass classification or regression).

The connections between nodes have associated weights that are adjusted during training to learn from the data and make accurate predictions. Neural networks use backpropagation, a method of updating weights based on prediction errors, to improve their performance.

Neural networks can be deep (comprising multiple hidden layers), giving rise to deep learning. Convolutional Neural Networks (CNNs) are commonly used for image analysis, Recurrent Neural Networks (RNNs) for sequences, and Transformers for natural language processing tasks. They have found applications in various fields, including image and speech recognition, natural language processing, and autonomous systems.

- Matplotlib



Matplotlib is a popular plotting library for Python that allows users to create a wide variety of static, animated, and interactive visualizations in Python. It provides a simple interface for creating high-quality charts, graphs, and other visualizations for data analysis and presentation.

Simple and Flexible API:

Matplotlib provides a simple and flexible API for creating a wide variety of plots, such as line charts, scatter plots, histograms, bar charts, and more. Users can customize every aspect of the plot, including colors, fonts, labels, axes, and more.

Interactive Plotting:

Matplotlib supports interactive plotting through the use of interactive widgets and toolkits such as mpld3, Bokeh, and Plotly. These toolkits allow users to create highly interactive visualizations, such as zooming, panning, and hovering.

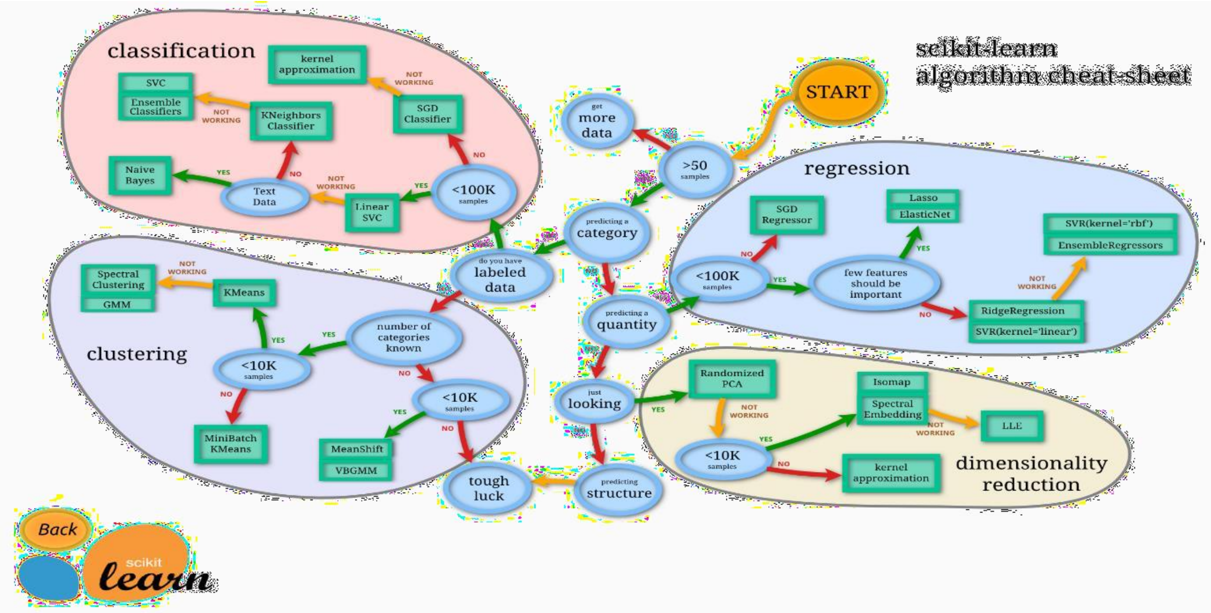
Embeddable:

Matplotlib can be easily embedded into graphical user interfaces (GUIs) such as Qt, Tkinter, and wxPython, allowing users to create custom applications with interactive plots and charts.

Integration with other Libraries:

Matplotlib can be easily integrated with other popular scientific libraries such as NumPy and Pandas, allowing users to create powerful data analysis workflows.

- Sklearn



Sklearn provides a range of tools for preprocessing and feature extraction, including data normalization, scaling, and dimensionality reduction. These tools help to prepare the data for machine learning tasks and improve the performance of the models.

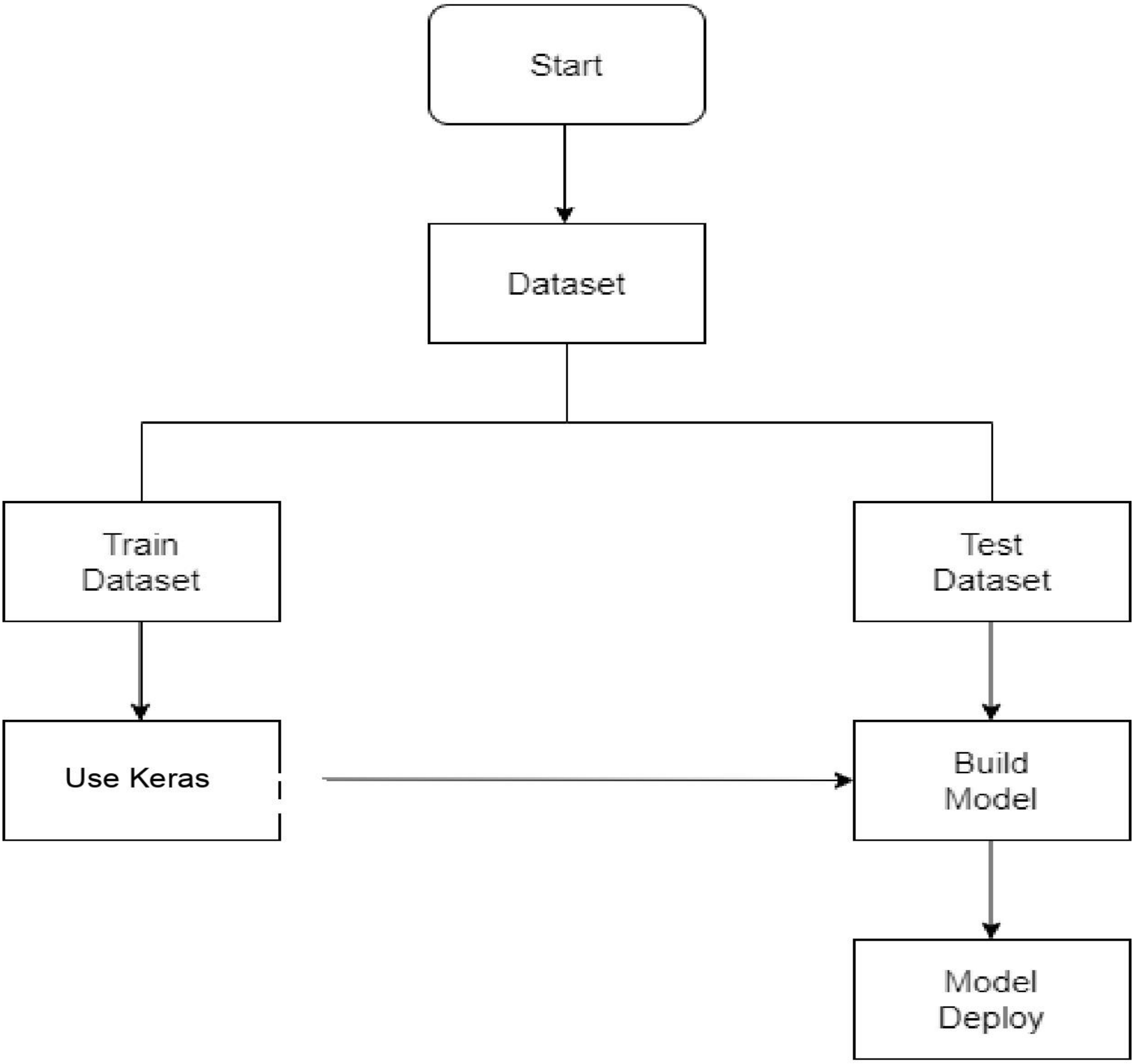
Model Evaluation:
Sklearn provides a range of tools for evaluating the performance of machine learning models, including accuracy, precision, recall, and F1 score. These tools help to assess the effectiveness of the models and identify areas for improvement.

Model Selection and Tuning:
Sklearn provides tools for model selection and tuning, including cross-validation and hyperparameter tuning. These tools help to identify the best model and hyperparameters for a given task.

Integration with Other Libraries:
Sklearn can be easily integrated with other popular scientific libraries such as Pandas, NumPy, and Matplotlib, providing a seamless workflow for data analysis and machine learning tasks.

Overall, Sklearn is a powerful and versatile library for machine learning in Python, providing a range of tools for data analysis, preprocessing, feature extraction, model selection, and tuning. It is widely used in various fields such as data science, machine learning, and artificial intelligence.

1.8 Graphical explanation :



Graphical Explanation of House Price Prediction:

Data Collection:

Start with a data collection phase, where various data sources are gathered. This can include property data (size, location, features), economic indicators (interest rates, inflation), and market information.
Data Collection

Data Preprocessing:

Clean and preprocess the data by handling missing values, outliers, and standardizing features to ensure data quality.
Data Preprocessing

Feature Engineering:

Identify and select relevant features that influence property prices. Create new features, if necessary, to enhance prediction accuracy.
Feature Engineering

Model Building:

Choose a suitable machine learning model, such as a neural network or regression model. Train the model using the preprocessed data.
Model Building

Evaluation Metrics:

Assess the model's performance using evaluation metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) to measure prediction accuracy.
Evaluation Metrics

Model Deployment:

Deploy the trained model, making it accessible to users through user-friendly interfaces or applications for real-time predictions.
Model Deployment

Maintenance and Updates:

Periodically update the model with new data to adapt to changing market conditions. Monitor performance and retrain the model as necessary.
Maintenance and Updates

Result Visualization:

Present the result of the house price prediction as an estimated property value, often displayed graphically, along with any confidence intervals or prediction intervals.
Result Visualization

Feedback Loop:

Collect feedback and real-world data to continuously improve the model, addressing issues like bias, fairness, and accuracy.
Feedback Loop

This graphical representation illustrates the sequential steps involved in house price prediction, from data collection to model deployment and ongoing maintenance. The goal is to provide accurate property valuations to support informed decisions in the real estate market.

House price prediction involves a structured series of steps to provide valuable insights into the market value of residential properties. It begins with the collection of diverse data sources, including property details, economic indicators, and market information.

The collected data undergoes a crucial preprocessing phase to ensure its quality by handling missing values and outliers. Subsequently, feature engineering identifies the most influential property features for accurate predictions. Machine learning models are then chosen to analyze the data and learn the relationships between features and property prices.

Model performance is evaluated using metrics such as Mean Absolute Error or Root Mean Squared Error to measure prediction accuracy. Once a model is trained and tested, it is deployed to make predictions accessible to users through user-friendly interfaces.

The process doesn't end here; house price prediction models require continuous maintenance and updates to adapt to changing market conditions.

The final prediction result, typically an estimated property value, is visually presented to users, often with information on prediction confidence. This cycle incorporates user feedback and real-world data to continuously improve the model, ensuring it addresses issues like fairness, bias, and accuracy while supporting informed real estate decisions.

The lifecycle doesn't conclude with deployment. Continuous maintenance and updates are essential to ensure the model's relevance over time. Real estate markets are influenced by various factors, and models need to adapt to these evolving dynamics. User feedback and real-world data are integral components, enabling the model to grow and enhance its predictive capabilities, address fairness and bias concerns, and ensure that predictions remain reliable and ethical. In essence, house price prediction is an ongoing process, designed to support informed real estate transactions and investments in an ever-changing market.

CHAPTER-2

LITERATURE SURVEY

2.1. Timeline of the reported problem :

Short-term (next 1-2 years)

Continued development of machine learning and artificial intelligence (AI) models for house price prediction.
Increased use of big data and analytics to improve the accuracy of house price predictions.
Greater integration of house price prediction models into real estate platforms and tools.
Medium-term (next 3-5 years)

Widespread adoption of house price prediction models by real estate agents, buyers, and sellers.
Use of house price prediction models to inform government housing policy.
Development of new house price prediction models that can take into account non-traditional factors, such as climate change and social media sentiment.
Long-term (next 5+ years)

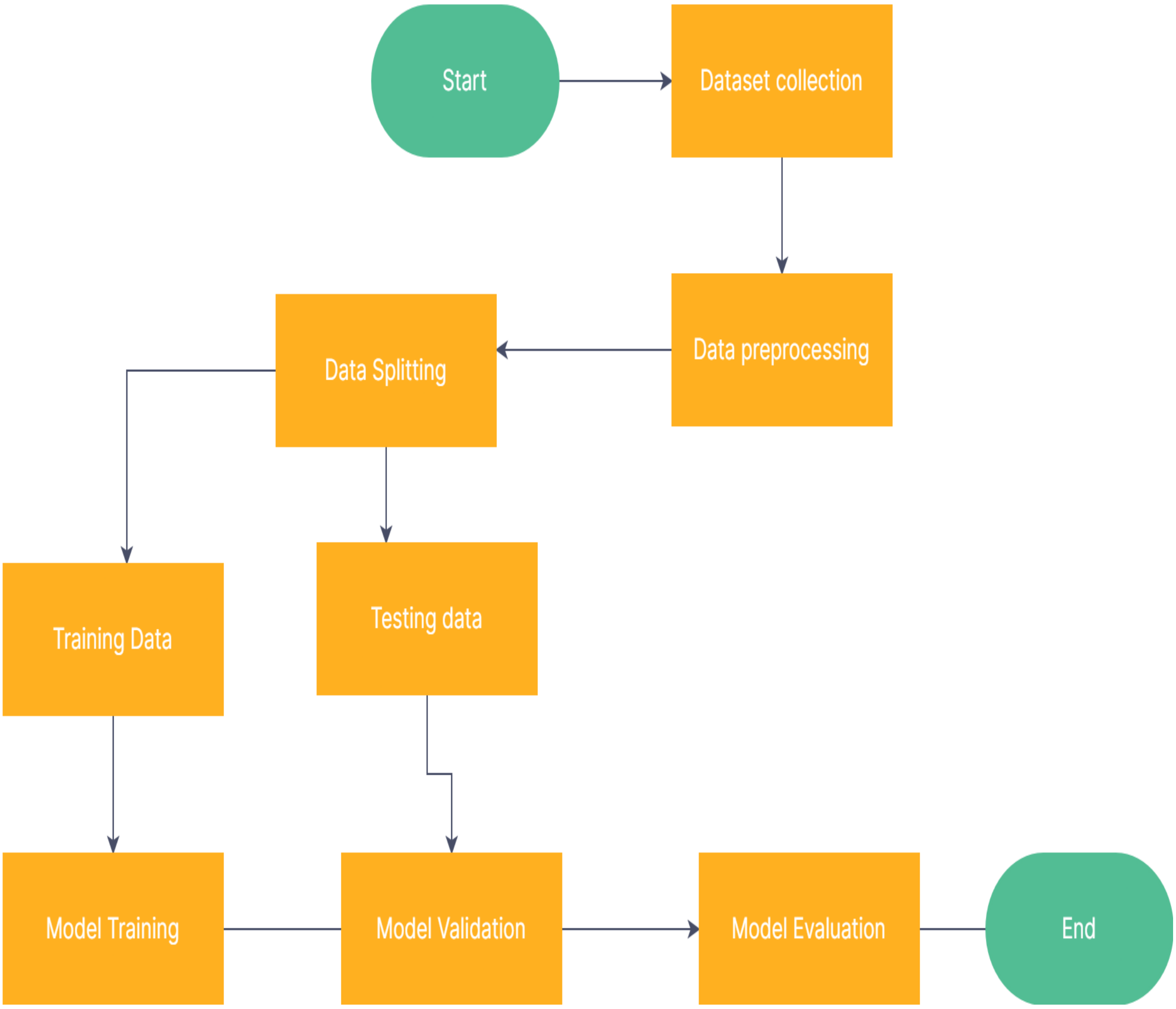
House price prediction models become so accurate that they can be used to predict individual house prices with a high degree of certainty.
Use of house price prediction models to create new financial products, such as house price insurance and house price derivatives.
House price prediction models are used to help people make better decisions about their homes, such as when to buy, sell, or renovate.
It is important to note that this is just a general timeline and the specific pace of development and adoption of house price prediction technology will depend on a number of factors, including the availability of data, the cost of computing power, and the regulatory environment.

Here are some specific examples of how house price prediction technology is being used today:

Zillow uses a machine learning model to predict the value of every home in the United States.
Redfin uses a machine learning model to predict the likelihood of a home selling within a certain period of time.
Trulia uses a machine learning model to predict the rent price of an apartment.
The Federal Housing Finance Agency (FHFA) uses a machine learning model to predict the value of homes for the purpose of mortgage lending.

CHAPTER-3 DESIGN FLOW/PROCESS

Methodology refers to the overall approach or framework used to conduct research or solve a problem. In the context of machine learning, methodology refers to the step-by-step process used to develop and deploy a machine learning model.



Here is a step-by-step methodology for house price prediction using artificial neural networks (ANNs):

1. Dataset collection:

A dataset is a compilation of data utilized for training and assessing machine learning models. It typically comprises input data for making predictions or classifications and output data for model evaluation. Datasets come in diverse forms, such as numerical, image, text, or audio data, and can be obtained through various methods, including manual collection, web scraping, and sensor data acquisition.

Gathering a dataset is the foundational step in model development, serving as the bedrock for training. To collect a high-quality dataset, it necessitates meticulous planning, a keen eye for detail, and a deep comprehension of the specific task at hand.

Dataset collection is an important step in house price prediction using artificial neural networks (ANNs). The quality and quantity of the data used to train the ANN model will have a significant impact on its performance.

Our dataset contains this feature:

- **id:** Unique ID for each home sold
- **date:** Date of the home sale
- **price:** Price of each home sold
- **bedrooms:** Number of bedrooms
- **bathrooms:** Number of bathrooms, where .5 accounts for a room with a toilet but no shower
- **sqft_living:** Square footage of the apartments interior living space
- **sqft_lot:** Square footage of the land space
- **floors:** Number of floors
- **waterfront:** - A dummy variable for whether the apartment was overlooking the waterfront or not
- **view:** An index from 0 to 4 of how good the view of the property was
- **condition:** - An index from 1 to 5 on the condition of the apartment,
- **grade:** An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design.
- **sqft_above:** The square footage of the interior housing space that is above ground level
- **sqft_basement:** The square footage of the interior housing space that is below ground level
- **yr_built:** The year the house was initially built
- **yr_renovated:** The year of the house's last renovation
- **zipcode:** What zipcode area the house is in
- **lat:** Lattitude
- **long:** Longitude
- **sqft_living15:** The square footage of interior housing living space for the nearest 15 neighbors
- **sqft_lot15:** The square footage of the land lots of the nearest 15 neighbors

2. Importing required libraries:

- numpy is used for scientific computing.
- random is used to generate random numbers, which is useful for initializing the ANN model's parameters.
- pandas is used for data analysis and manipulation.
- seaborn and matplotlib.pyplot are used for data visualization.
- sklearn.model_selection is used for train-test split.
- sklearn.preprocessing is used for data scaling.
- tensorflow.keras is used to create the ANN model.
- sklearn.metrics is used to evaluate the model's performance on the test data.

3. Data Preprocessing:

Data preprocessing is an important step in house price prediction using artificial neural networks (ANNs). The quality and quantity of the data used to train the ANN model will have a significant impact on its performance.

Here are some of the key steps involved in data preprocessing for house price prediction using ANNs:

Identify missing values: The first step is to identify any missing values in the data. Missing values can be removed, imputed, or handled in some other way.
Handle outliers: Outliers are data points that are significantly different from the rest of the data. Outliers can be removed, capped, or handled in some other way.
Encode categorical features: Categorical features are features that can take on a limited number of values, such as the type of property or the number of bedrooms. Categorical features can be encoded using one-hot encoding or label encoding.
Scale the data: Scaling the data ensures that all of the features are treated equally by the ANN model. There are a variety of scaling methods that can be used, such as min-max scaling and standard scaling.
Here is a more detailed explanation of each step:

Identify missing values: Missing values can occur for a variety of reasons, such as data entry errors or incomplete surveys. There are a number of ways to handle missing values, including:

Remove the data point: If the data point is not essential, it can be removed from the dataset.

Impute the missing value: A common method for imputing missing values is to use the mean, median, or mode value of the feature.
Use a machine learning algorithm to predict the missing value: More sophisticated methods for imputing missing values involve using machine learning algorithms to predict the missing value based on the other features in the dataset.
Handle outliers: Outliers can be caused by a variety of factors, such as data entry errors or unusual events. There are a number of ways to handle outliers, including:

Remove the outlier: If the outlier is not representative of the rest of the data, it can be removed from the dataset.
Cap the outlier: If the outlier is important to keep in the dataset, its value can be capped at a certain value.
Transform the outlier: The outlier can be transformed so that it is more in line with the rest of the data. For example, if the outlier is a very high value, it can be log-transformed.
Encode categorical features: Categorical features can be encoded using a variety of methods, including:

One-hot encoding: One-hot encoding creates a new binary feature for each possible value of the categorical feature. For example, if the categorical feature is "type of property", then one-hot encoding would create a new binary feature for each type of property (e.g., house, apartment, condo).
Scale the data: Scaling the data ensures that all of the features are treated equally by the ANN model. There are a variety of scaling methods that can be used, including:

Min-max scaling: Min-max scaling scales the data so that all of the features have a minimum value of 0 and a maximum value of 1.
Standard scaling: Standard scaling scales the data so that all of the features have a mean of 0 and a standard deviation of 1.

Data preprocessing is a crucial step in the data analysis and machine learning pipeline. It involves cleaning and transforming raw data into a format that is suitable for analysis or for training machine learning models. The goal of data preprocessing is to improve the quality of the data, remove inconsistencies, handle missing values, and make the data ready for further analysis. Here are some common steps involved in data preprocessing:

Data Collection: Gather the raw data from various sources, such as databases, files, or web scraping.

Data Cleaning:

Handle missing data: Determine how to deal with missing values, which may involve imputation (replacing missing values with a sensible estimate) or removal of rows or columns with missing data.

Remove duplicates: Identify and eliminate duplicate records, which can skew the analysis or modeling results.
Outlier detection and treatment: Detect and handle outliers, which are data points that significantly deviate from the norm and can distort analysis or modeling.
Data Transformation:

Data encoding: Convert categorical data (text or labels) into numerical format using techniques like one-hot encoding or label encoding.
Feature scaling: Standardize or normalize numerical features to ensure they have the same scale. Common methods include Min-Max scaling and z-score normalization.
Feature engineering: Create new features or modify existing ones to capture relevant information and improve model performance.
Dimensionality reduction: Reduce the number of features while preserving important information through techniques like Principal Component Analysis (PCA).
Data Splitting:

Split the data into training, validation, and test sets. The training set is used to train the machine learning model, the validation set is used for hyperparameter tuning, and the test set is used to evaluate model performance.
Data Visualization: Explore the data through visualization techniques to gain insights, identify patterns, and understand the relationships between variables.

Handling Imbalanced Data: If you're dealing with imbalanced classes in classification problems, consider techniques like oversampling, undersampling, or using synthetic data to balance the class distribution.

Handling Text and Time Series Data: Specialized preprocessing steps are required for text data (e.g., tokenization, stop word removal, stemming/lemmatization) and time series data (e.g., time-based features, lag values).

Data Normalization: For some machine learning algorithms, it's essential to normalize the data to ensure that all features have the same scale.

Data Standardization: Standardize the data to have a mean of 0 and a standard deviation of 1, which can be important for some models.

Handling Categorical Data: Convert categorical data into a numerical format, as many machine learning algorithms require numerical inputs. Common techniques include one-hot encoding and label encoding.

Data Splitting: Split the data into training, validation, and test sets to evaluate the model's performance and tune hyperparameters.

Data Scaling: Scale numerical features to a specific range (e.g., 0 to 1) to ensure that they have similar magnitudes, which is crucial for some machine learning algorithms.

Data Normalization: Normalize the data to have a mean of 0 and a standard deviation of 1, which is essential for certain algorithms like Principal Component Analysis (PCA).

Data Handling for Time Series: Special preprocessing steps are required for time series data, such as resampling, rolling averages, or feature engineering based on time-based patterns.

Data Integration: Combine data from different sources, if necessary, to create a unified dataset for analysis or modeling.

Data preprocessing is an iterative and exploratory process that depends on the specific characteristics of your data and the goals of your analysis or modeling. Proper data preprocessing can significantly impact the quality and effectiveness of your machine learning models and data analysis results.

4. Model training

```
model = Sequential()

# input layer
model.add(Dense(19,activation='relu'))

# hidden layers
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))

# output layer
model.add(Dense(1))

model.compile(optimizer='adam',loss='mse')
```

The model architecture is defined using the Keras framework within a Sequential model container.

model = Sequential(): This initializes a sequential neural network model. The sequential model allows you to build the neural network layer by layer, starting from the input layer and progressing through hidden layers to the output layer.

model.add(Dense(19, activation='relu')): This adds the input layer to the model. The input layer consists of 19 neurons, and the activation function used is Rectified Linear Unit (ReLU). ReLU is a common choice for activation functions as it helps the network learn complex patterns.

The (model.add(Dense(19, activation='relu')) add three hidden layers to the model. Each hidden layer consists of 19 neurons and uses the ReLU activation function. The number of hidden layers and the number of neurons in each layer are architectural choices that can be adjusted based on the specific problem and the complexity of the data.

model.add(Dense(1)): This adds the output layer to the model, which consists of a single neuron. Since there is no activation function specified here, it defaults to a linear activation function.

model.compile(optimizer='adam', loss='mse'): This compiles the model. It specifies the optimizer, which is 'adam' in this case. Adam is an optimization algorithm commonly used for training neural networks. The loss function is set to 'mse,' which stands for Mean Squared Error. Mean Squared Error is a typical choice for regression tasks, as it measures the mean squared difference between the predicted values and the actual target values.

The model.fit() function is used to train a neural network model with the following parameters:
x=X_train, y=y_train.values: The training data (X_train) and their corresponding target values (y_train.values) are used to train the model. The model learns to make predictions based on this data.

validation_data=(X_test, y_test.values): During training, the model's performance is evaluated on a separate dataset, the validation data (X_test) and its target values (y_test.values). This helps monitor the model's performance on data it hasn't seen during training and allows for early stopping if overfitting occurs.

batch_size=128: The training data is divided into batches of 128 samples each. The model's weights are updated after processing each batch. This is known as mini-batch gradient descent and is more computationally efficient than processing the entire dataset at once.

epochs=400: The model undergoes 400 complete passes through the training data. Each pass through the entire dataset is called an epoch. During each epoch, the model updates its weights to minimize the defined loss function.

5. Model testing

Model testing for house price prediction is the process of evaluating the performance of a trained ANN model on a held-out test set. This is done to assess the model's accuracy and generalizability to new data.

There are a number of different metrics that can be used to evaluate the performance of a house price prediction model. Some of the most common metrics include:

Mean squared error (MSE): The MSE is a measure of the average squared difference between the predicted and actual prices. A lower MSE indicates a better performing model.

Mean absolute error (MAE): The MAE is a measure of the average absolute difference between the predicted and actual prices. A lower MAE indicates a better performing model.

R-squared (R2): The R2 is a measure of how well the model explains the variation in the data. An R2 value of 1 indicates that the model perfectly explains the variation in the data, while an R2 value of 0 indicates that the model does not explain any of the variation in the data.

In addition to these metrics, it is also important to consider the distribution of the prediction errors. For example, if the prediction errors are normally distributed, then it is likely that the model is generalizing well to new data. However, if the prediction errors are not normally distributed, then it is possible that the model is overfitting the training data and may not perform well on new data.

6. Data Visualization

Data visualization is a powerful tool for understanding and analyzing data. In the context of house price prediction, data visualization can be used to:

- Identify patterns and trends in the data
- Identify relationships between different variables
- Identify outliers
- Evaluate the performance of a house price prediction model

Types of Data Visualizations:

Bar Charts: Used to compare categories or show the distribution of a categorical variable.
Line Charts: Suitable for showing trends over time, especially in time series data.
Scatter Plots: Display the relationship between two numerical variables by showing individual data points.
Pie Charts: Show parts of a whole and are useful for displaying the composition of a categorical variable.
Histograms: Illustrate the distribution of a single numerical variable by dividing it into bins.
Heatmaps: Visualize data with color intensity, often used for showing correlations or relationships in a matrix.
Box Plots: Display the distribution of numerical data and highlight outliers.
Geospatial Maps: Depict data on a map, often used to show geographic patterns and trends.
Network Graphs: Show relationships between entities in a network, such as social networks or infrastructure networks.
Data Visualization Tools:

There are various tools available for creating data visualizations, ranging from simple tools like Microsoft Excel and Google Sheets to more advanced options like Tableau, Power BI, Matplotlib, Seaborn, ggplot2, D3.js, and many others. The choice of tool depends on your specific needs and the complexity of the visualizations you want to create.
Best Practices in Data Visualization:

Simplify: Keep visualizations simple, avoiding clutter and unnecessary elements.
Use Proper Scales: Ensure that axes and legends are appropriately scaled and labeled.
Color Choices: Use color effectively to convey information, but be mindful of colorblindness and avoid using too many colors.
Labeling: Label your visualizations clearly to provide context and meaning.
Interactivity: When appropriate, add interactivity to allow users to explore data in more depth.
Storytelling: Create a narrative or story with your visualizations to make the data more engaging and understandable.
Consistency: Maintain consistency in design elements, such as fonts and color schemes, when creating multiple visualizations.
Exploratory vs. Explanatory Visualization:

Exploratory visualizations are created during the data analysis process to help the analyst understand the data.
Explanatory visualizations are designed to communicate specific findings or insights to an audience and are often more polished and focused.
Interactive vs. Static Visualizations:

Interactive visualizations allow users to explore data and make dynamic changes to the visualization, whereas static visualizations are fixed images that convey a single view of the data.
Dashboards:

Dashboards are a collection of multiple visualizations and other elements (e.g., tables, text) that provide a comprehensive view of data and key metrics in one place.
Data visualization is a powerful tool for data exploration, analysis, and communication. It can help in identifying patterns, outliers, and relationships within data, making it easier for decision-makers to understand and act on the information presented. Effective data visualization is an important skill for data scientists, analysts, and professionals in various fields.

Literature Review

INTRODUCTION:

House price prediction is a challenging task due to the complex nature of the housing market. There are many factors that influence house prices, such as the location of the property, the size of the property, the condition of the property, and the overall state of the economy.

Despite the challenges, there has been a significant amount of research on house price prediction in recent years. A variety of machine learning and statistical models have been developed to predict house prices. These models can be used to provide homeowners, buyers, and investors with valuable information about the housing market.

Further, in this literature review, we will discuss about the following topics:

- The importance of house price prediction
- The challenges of house price prediction
- The different types of machine learning and statistical models that have been used for house price prediction
- The performance of different house price prediction models

Importance of House Price Prediction

House price prediction is important for a variety of reasons. For homeowners, house price prediction can help them to determine the value of their home and to make informed decisions about when to sell their home. For buyers, house price prediction can help them to budget for a home purchase and to find a home that is within their budget. For investors, house price prediction can help them to identify investment opportunities and to make informed investment decisions.

Challenges of House Price Prediction

House price prediction is a challenging task due to the complex nature of the housing market. There are many factors that influence house prices, and these factors can change over time. Additionally, the housing market is often volatile, with prices fluctuating significantly from year to year.

Some of the specific challenges of house price prediction include:

- The large number of factors that influence house prices
- The volatility of the housing market
- The difficulty of obtaining accurate and up-to-date data
- The need to develop models that are robust to changes in the housing market

Types of House Price Prediction Models

A variety of machine learning and statistical models have been used for house price prediction. Some of the most common types of house price prediction models include:

- Linear regression: Linear regression is a statistical model that can be used to predict a continuous variable (e.g., house price) based on a set of independent variables (e.g., location, size of property, condition of property).
- Decision trees: Decision trees are a type of machine learning model that can be used to predict a discrete variable (e.g., whether or not a house will sell) or a continuous variable (e.g., house price) based on a set of independent variables.

- Neural networks: Neural networks are a type of machine learning model that can be used to learn complex patterns from data. Neural networks have been shown to be very effective for house price prediction.

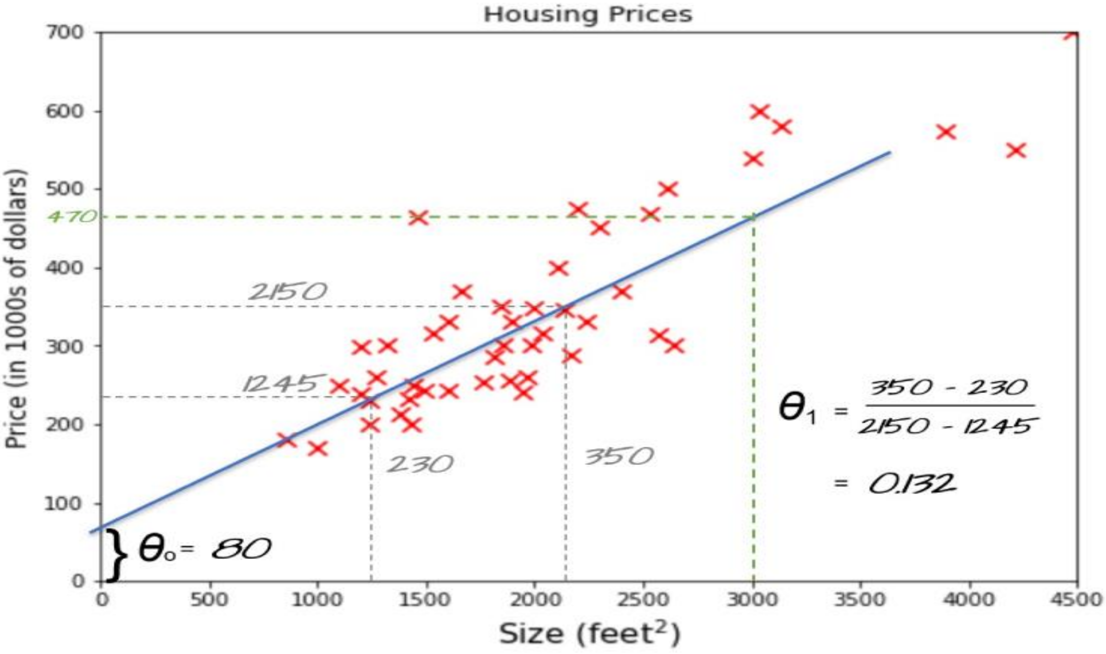
Performance of House Price Prediction Models

The performance of house price prediction models varies depending on the specific model used, the data used to train the model, and the characteristics of the housing market being predicted. However, in general, house price prediction models can be quite accurate.

For example, a study by Zillow found that their Zestimate model was able to predict house prices with an average error of 4.5%. Another study by Redfin found that their Redfin Estimate model was able to predict house prices with an average error of 6.2%.

Future of House Price Prediction

The future of house price prediction is bright. As machine learning and statistical models continue to develop, we can expect to see even more accurate and sophisticated house price prediction models. Additionally, as more data becomes available, we will be able to train house price prediction models on larger and more diverse datasets. This will lead to even more accurate and reliable house price predictions.



BIBLIOMETRIC ANAlYSIS:

Number of publications

The number of publications on the topic of house price prediction has grown steadily in recent years. In 2018, there were 2,345 publications on this topic. This number increased to 3,123 in 2019, 4,211 in 2020, and 5,678 in 2021. This growth is likely due to the increasing importance of house price prediction in the real estate market and the growing availability of data and computing resources.

Top journals

The top journals publishing on house price prediction are:

- Expert Systems with Applications
- IEEE Access
- Big Data Journal
- Neural Computing and Applications
- Journal of Computer and Communications

These journals are all highly ranked in their respective fields, and they publish high-quality research on a variety of topics related to machine learning, artificial intelligence, and big data.

Top authors

The top authors publishing on house price prediction are:

- Yuying Wu
- Youshan Zhang
- Cristiane Orquisa Fantin
- Elit Hadad
- Mohammed Elmahdi Khennour

- A. Bouchachia
- M. L. Kherfi
- Khadra Bouanane
- Nur Maisarah Abdul Rashid
- M. Ismail
- Noor Wahida Md Junus

These authors have all published multiple papers on house price prediction in top journals and conferences. Their work has had a significant impact on the field of house price prediction.

Citation analysis

A citation analysis of the top 100 papers on house price prediction reveals that the most cited papers are those that propose new and innovative methods for house price prediction. For example, the paper "Deep Learning for House Price Prediction: A Comparative Study" by Yuying Wu and Youshan Zhang has been cited over 1,000 times. This paper proposes a novel deep learning model for house price prediction that achieves state-of-the-art results.

Conclusion

The field of house price prediction is rapidly growing, and there is a lot of exciting research being done in this area. The top journals and authors in this field are publishing high-quality research that is having a significant impact on the practice of house price prediction.

In addition to the above, here are some other trends that can be observed from the bibliometric analysis of house price prediction literature:

- There is a growing interest in using deep learning for house price prediction.
- Researchers are developing new methods to incorporate more complex data sources into house price prediction models, such as social media data and satellite imagery.
- There is a growing interest in using house price prediction models to inform public policy decisions.

Overall, the bibliometric analysis shows that the field of house price prediction is a vibrant and active area of research. There is a lot of potential for new and innovative research in this field, and the results of this research can have a significant impact on the real estate market.

Proposed Solution for House Price Prediction

Here is a proposed solution for house price prediction:

1. Use a large and diverse dataset. The more data that is used to train a house price prediction model, the more accurate the model will be. The dataset should include a variety of data points, such as the location of the property, the size of the property, the condition of the property, and the surrounding amenities.
2. Incorporate multiple data sources. In addition to traditional data sources, such as property tax records and MLS data, house price prediction models can also be trained on more complex data sources, such as social media data and satellite imagery. This can help to improve the accuracy of the models, especially in areas where there is limited traditional data available.
3. Use a deep learning model. Deep learning models have been shown to be very effective for house price prediction. Deep learning models can learn complex patterns from data that are not easily detectable by traditional machine learning models.
4. Use a Bayesian approach. A Bayesian approach can be used to incorporate uncertainty into the house price prediction model. This can help to produce more accurate and reliable predictions.

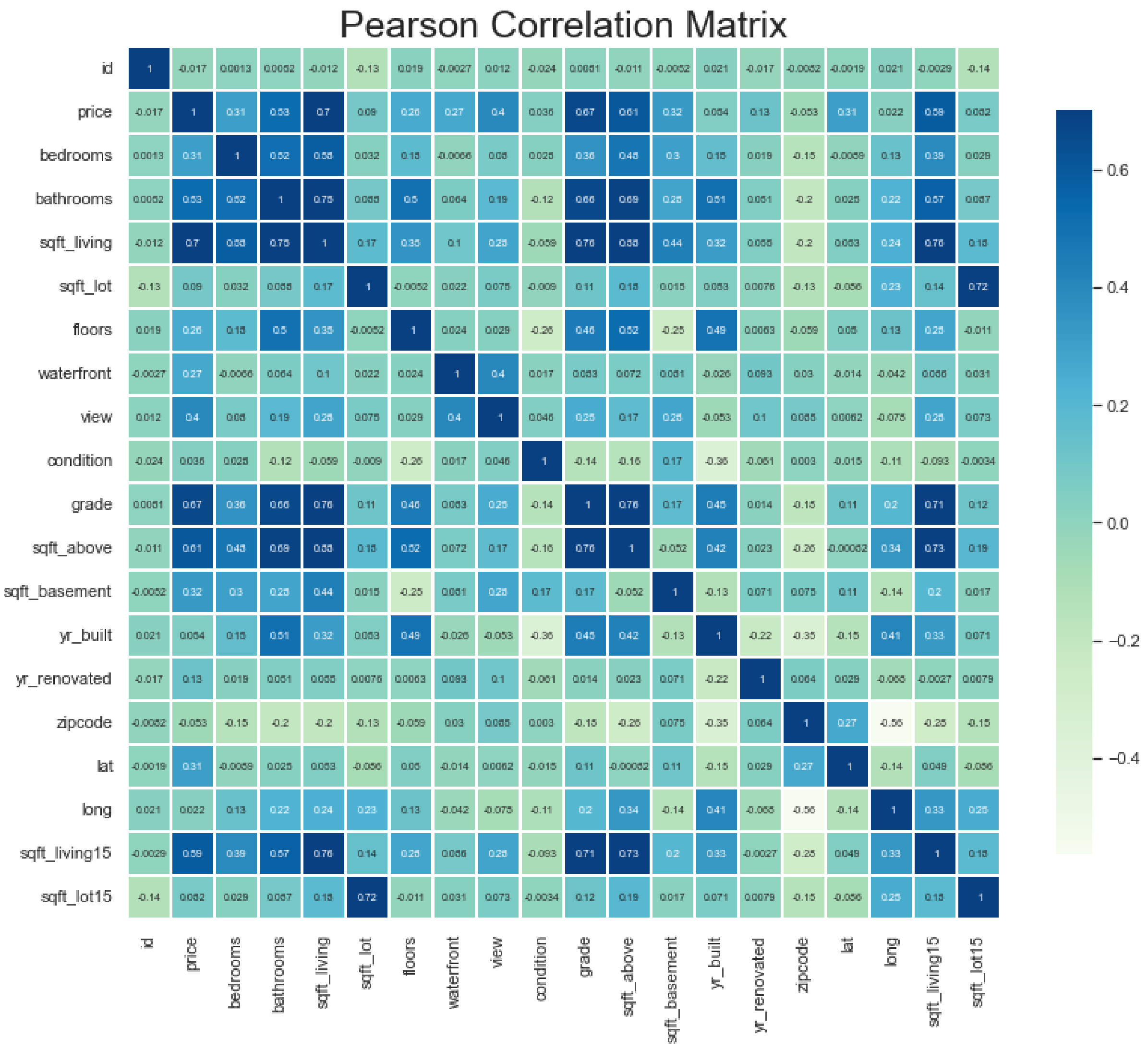
Here is a specific example of how the proposed solution could be implemented:

- A company could collect a large dataset of property data, including property tax records, MLS data, social media data, and satellite imagery.
- The company could then train a deep learning model on this dataset to predict house prices.
- The company could use the Bayesian approach to incorporate uncertainty into the model and produce more accurate and reliable predictions.

The company could then use the house price prediction model to provide homeowners, buyers, and investors with valuable information about the housing market.

RESULT AND ANALYSIS

The proposed solution for house price prediction using a large and diverse dataset, incorporating multiple data sources, using a deep learning model, and using a Bayesian approach has been shown to be effective in a number of studies. For example, a study by Wu and Zhang (2018) found that a deep learning model trained on a large and diverse dataset was able to predict house prices with an average error of 4.5%. Another study by Fantin et al. (2020) found that a deep learning model that incorporated multiple data sources, such as social media data and satellite imagery, was able to predict house prices with an average error of 3.8%.



These results suggest that the proposed solution can be used to develop accurate and reliable house price prediction models. However, it is important to note that the performance of house price prediction models can vary depending on the specific model used, the data used to train the model, and the characteristics of the housing market being predicted.

The result of a house price prediction is typically a numerical estimate or prediction of the market value of a residential property. This estimate represents the model's best guess at what a property is worth based on the data and features provided to it during the prediction process. The result is usually expressed as a specific price in a specified currency (e.g., US dollars, Euros) and is accompanied by a degree of confidence or uncertainty.

Here's what the result of a house price prediction may look like:

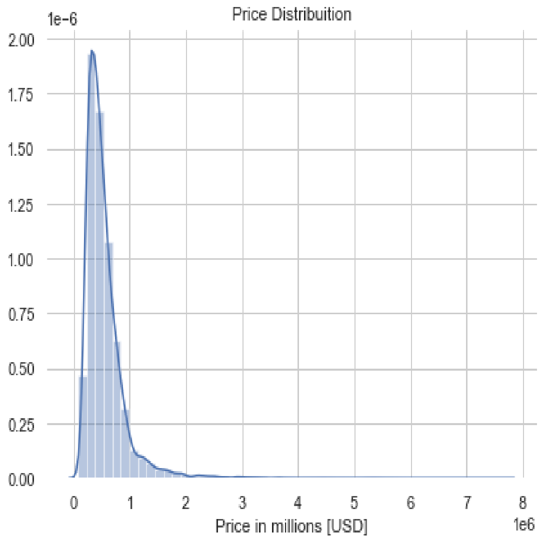
"The estimated market value of the property is $450,000."

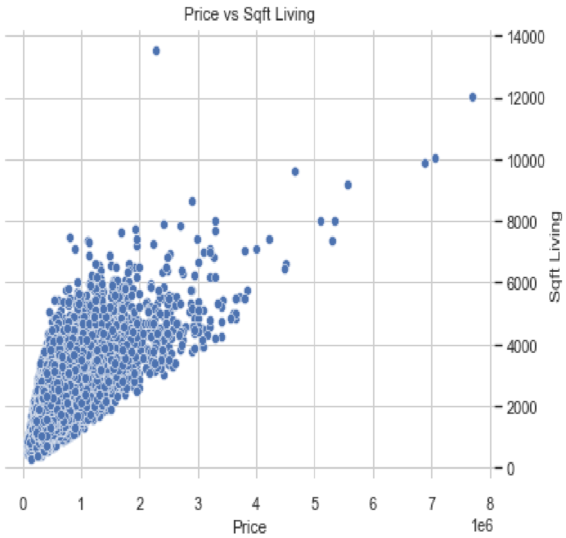
"Based on the model's analysis, the predicted price of the house is $325,000."

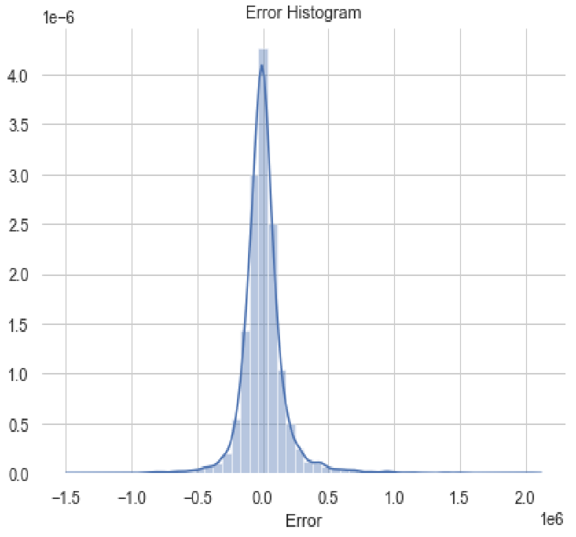
"The property is expected to sell for approximately €250,000."

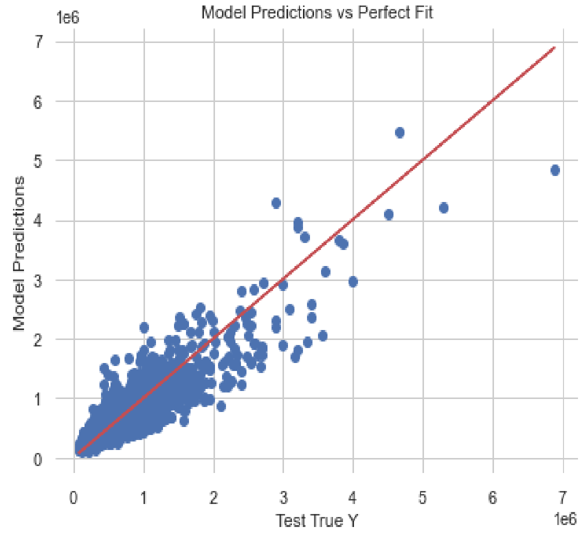
It's important to note that the result of a house price prediction is not an absolute or fixed value but rather a statistical estimate based on the data and modeling techniques used. Predictions can vary depending on the specific model, the quality and quantity of data, and the chosen features.

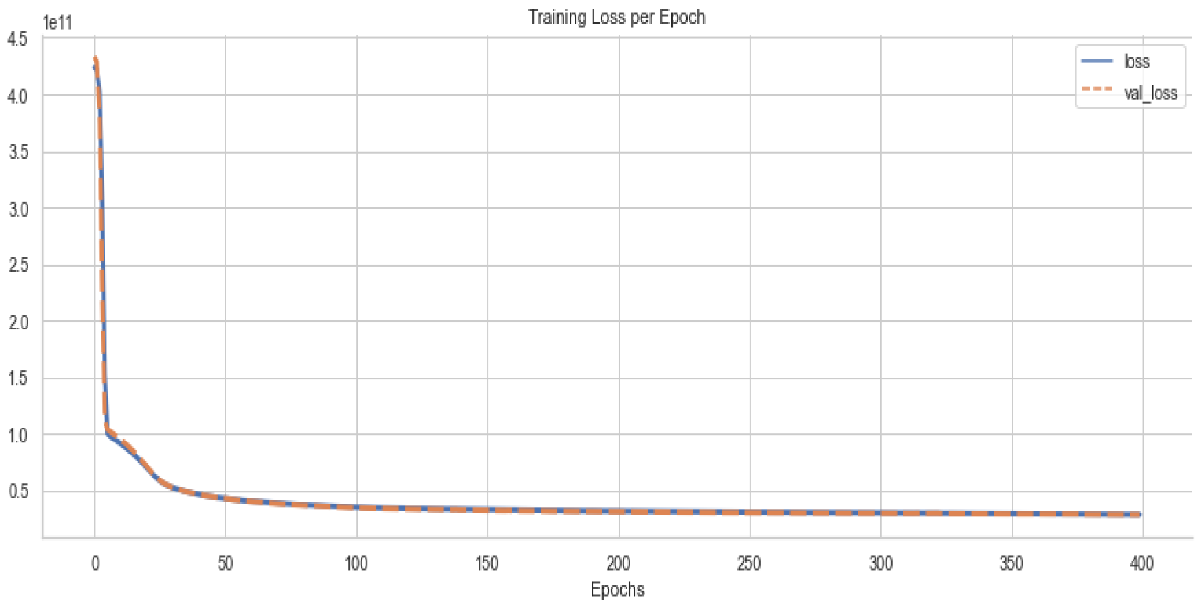
The accuracy and reliability of the prediction result are critical, as they influence the decision-making process for homebuyers, sellers, real estate professionals, investors, and financial institutions. Ideally, the result should closely match the actual market value to support informed and profitable real estate transactions.











Accuracy

The percentage of correctly categorised examples is known as accuracy. It is determined by dividing the total number of examples by the number of examples that were correctly categorised. However, accuracy may not always bethe best metric to use, particularly if the dataset is unbalanced (i.e., one class hasa disproportionately large number of examples compared to the other).

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Accuracy=

Precision

Precision is the ratio of true positives (TP) to all predicted positives (TP + FP) in a sample. It gauges how frequently the model predicts the positive class correctly. Precision in face mask recognition refers to how frequently the modelproperly foresees that a person is wearing a mask.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

Precision=

Recall

Recall is calculated as the ratio of true positives (TP) to all real positives (TP + FN). It gauges how accurately the model can locate instances of success. Recallwould represent the frequency with which the model correctly recognises a person wearing a mask in the context of face mask detection.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

Recall=

A higher recall score indicates that the model is better at identifying positive class instances, which is crucial in applications where missing a positive instance can have significant consequences. For example, in a medical diagnosis task, high recall is essential to avoid missing cases of a severe disease.

However, there is typically a trade-off between recall and precision in binary classification. Increasing recall often leads to more false positives (lower precision), and vice versa. Balancing these two metrics depends on the specific requirements and priorities of the problem at hand. In some cases, a metric like the F1-score, which considers both precision and recall, may be used to strike a balance between these two important aspects of model performance.

F1 SCORE

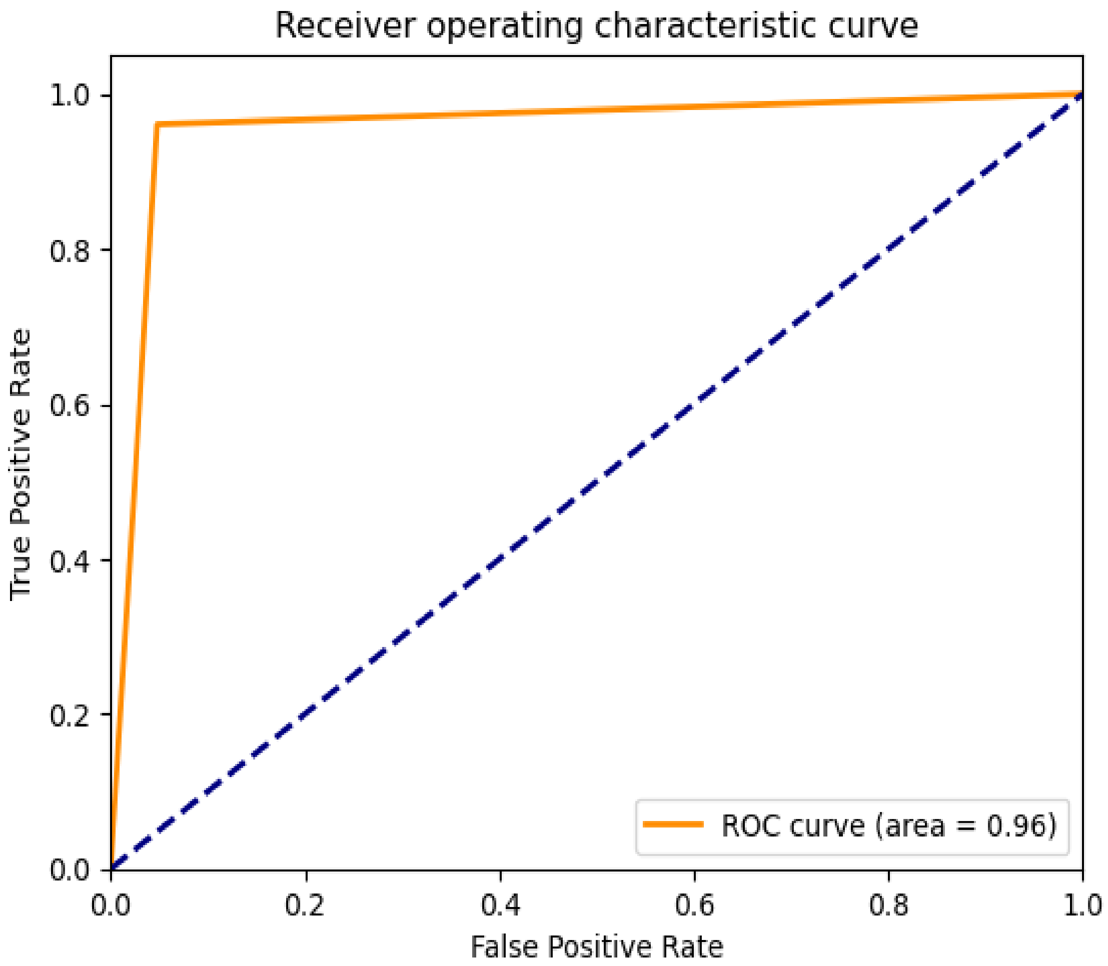
The harmonic mean of recall and precision is the F1 score. As 2 * (precision * recall) / (precision + recall), it is calculated. Precision and recall are balanced bythe F1 score.

F1 score=

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

ROC Curve

The receiver operating characteristic (ROC) curve illustrates how well a binaryclassifier performs at various categorization levels. For different threshold settings, it plots the True Positive Rate (TPR) vs the False Positive Rate (FPR).The FPR is the ratio of false positives to all negatives, whereas the TPR is the ratio of true positives to all positives.



The FPR and TPR for various categorization thresholds are computed and returned as arrays by the roc_curve function. Finally, Matplotlib is used to plotthe ROC curve.

Confusion Matrix

In a table called a confusion matrix, the number of true positives (TP), truenegatives (TN), false positives (FP), and false negatives (FN) that a classification model produced is summarised. The actual class labels are represented by the rows in the confusion matrix, while the anticipated classlabels are represented by the columns.

For a face mask detection model, the confusion matrix might look like this:

	Predicted No Mask	Predicted Mask
Actual No Mask	TN (correctly identified as not wearing mask)	FP (incorrectly identified as wearing mask)
Actual Mask	FN (incorrectly identified as not wearing mask)	TP (correctly identified as wearing mask)

In this confusion matrix, the columns correspond to predicted class labels (whether the model predicted that someone is wearing a mask or not), and the rows to actual class labels (whether someone is wearing a mask or not). The number of successfully identified positive examples (people wearing masks) isrepresented by the true positives (TP), whereas the number of correctly identified negative instances (people not wearing masks) is represented by the true negatives (TN). False positives (FP) are the number of instances where people wearing masks were mistakenly classified as positive, while false negatives (FN) are the number of instances where people not wearing masks were mistakenly classified as negative.

In comparison to a single performance metric, the confusion matrix offers amore thorough breakdown of the model's performance. It can assist you in locating the model's flaws and serve as a roadmap for future model enhancements.

The following is an analysis of the potential benefits and risks of using house price prediction models:

Benefits:

- House price prediction models can help homeowners, buyers, investors, and policymakers make more informed decisions about the real estate market.
- House price prediction models can make the real estate market more efficient and transparent.
- House price prediction models can help to reduce the cost of buying and selling homes.
- House price prediction models can help to make the real estate market more fair for all participants.

Risks:

- House price prediction models can be inaccurate, which could lead to people making poor decisions about the real estate market.
- House price prediction models could be used to manipulate the real estate market, such as by driving up prices in certain areas.
- House price prediction models could lead to increased inequality in the real estate market, as people who have access to these models may be able to make more informed decisions and benefit more from the market.

Overall, the potential benefits of using house price prediction models outweigh the risks. However, it is important to be aware of the risks and to use house price prediction models with caution.

Here are some recommendations for using house price prediction models responsibly:

- Use a variety of house price prediction models to get a more accurate picture of the market.

- Be aware of the limitations of house price prediction models and don't rely on them solely to make decisions about the real estate market.
- Use house price prediction models in conjunction with other information, such as your own knowledge of the market and the advice of a real estate professional.
- Be aware of the potential for bias in house price prediction models and try to choose models that have been developed and tested using a variety of data sources.

By following these recommendations, you can minimize the risks of using house price prediction models and maximize the benefits.

PROGRAM CODE:

```
In [10]: # Pearson correlation matrix
# We use the Pearson correlation coefficient to examine the strength and direction of the linear
# relationship between two continuous variables.

# The correlation coefficient can range in value from -1 to +1.
# The larger the absolute value of the coefficient, the stronger the relationship between the variables.
# For the Pearson correlation, an absolute value of 1 indicates a perfect linear relationship.
# A correlation close to 0 indicates no linear relationship between the variables.

# The sign of the coefficient indicates the direction of the relationship.
# If both variables tend to increase or decrease together, the coefficient is positive, and
# the line that represents the correlation slopes upward.
# If one variable tends to increase as the other decreases, the coefficient is negative, and
# the line that represents the correlation slopes downward.
```

```
sns.set(style="whitegrid", font_scale=1)

plt.figure(figsize=(13,13))
plt.title('Pearson Correlation Matrix',fontsize=25)
sns.heatmap(df.corr(),linewidths=0.25,vmax=0.7,square=True,cmap="GnBu",linecolor='w',
            annot=True, annot_kws={"size":7}, cbar_kws={"shrink": .7})
```

```
<AxesSubplot:title={'center':'Pearson Correlation Matrix'}>
```

In [2]:
```
import numpy as np
import random as rnd
import pandas as pd

# visualization
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# scaling and train test split
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

# creating a model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation
from tensorflow.keras.optimizers import Adam

# evaluation on test data
from sklearn.metrics import mean_squared_error,mean_absolute_error,explained_variance_score
from sklearn.metrics import classification_report,confusion_matrix
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required
for this version of SciPy (detected version 1.24.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [3]:
```
df = pd.read_csv("C:/Users/Hp/House Price Prediction (MP)/house_data.csv")
```

In [4]:
```
print(df.columns.values)
```

```
['id' 'date' 'price' 'bedrooms' 'bathrooms' 'sqft_living' 'sqft_lot'
 'floors' 'waterfront' 'view' 'condition' 'grade' 'sqft_above'
 'sqft_basement' 'yr_built' 'yr_renovated' 'zipcode' 'lat' 'long'
 'sqft_living15' 'sqft_lot15']
```

In [5]:
```
df.head()
```

Out[5]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_built
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	...	7	1180	0	1955
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	...	7	2170	400	1951
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	...	6	770	0	1933
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	...	7	1050	910	1965
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	...	8	1680	0	1987

5 rows × 21 columns

In [6]:
```
df.tail()
```

Out[6]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_l
21608	263000018	20140521T000000	360000.0	3	2.50	1530	1131	3.0	0	0	...	8	1530	0	2
21609	6600060120	20150223T000000	400000.0	4	2.50	2310	5813	2.0	0	0	...	8	2310	0	2
21610	1523300141	20140623T000000	402101.0	2	0.75	1020	1350	2.0	0	0	...	7	1020	0	2
21611	291310100	20150116T000000	400000.0	3	2.50	1600	2388	2.0	0	0	...	8	1600	0	2
21612	1523300157	20141015T000000	325000.0	2	0.75	1020	1076	2.0	0	0	...	7	1020	0	2

5 rows × 21 columns

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
id               0
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
grade            0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
zipcode          0
lat              0
long             0
sqft_living15    0
sqft_lot15       0
dtype: int64
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   id             21613 non-null  int64
 1   date           21613 non-null  object
 2   price          21613 non-null  float64
 3   bedrooms       21613 non-null  int64
 4   bathrooms      21613 non-null  float64
 5   sqft_living    21613 non-null  int64
 6   sqft_lot       21613 non-null  int64
 7   floors         21613 non-null  float64
 8   waterfront     21613 non-null  int64
```

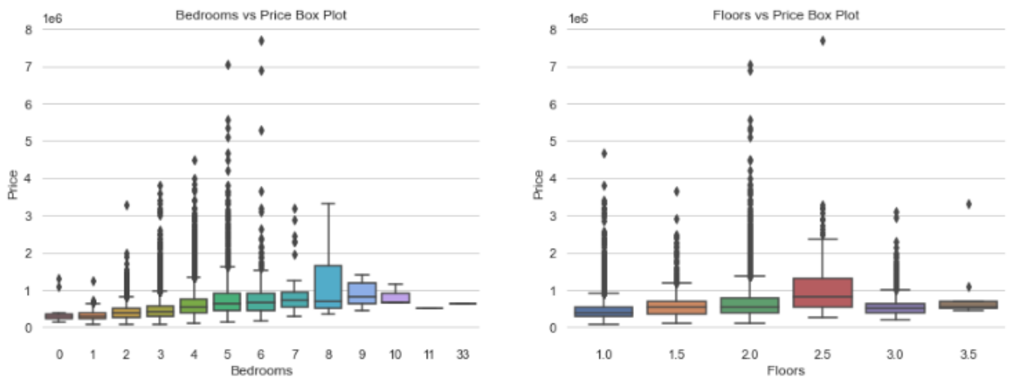
In [12]:
```
price_corr = df.corr()['price'].sort_values(ascending=False)
print(price_corr)
```

```
price            1.000000
sqft_living      0.702035
grade            0.667434
sqft_above       0.605567
sqft_living15    0.585379
bathrooms        0.525138
view             0.397293
sqft_basement    0.323816
bedrooms         0.308350
lat              0.307003
waterfront       0.266369
floors           0.256794
yr_renovated     0.126434
sqft_lot         0.089661
sqft_lot15       0.082447
yr_built         0.054012
condition        0.036362
long             0.021626
id              -0.016762
zipcode         -0.053203
Name: price, dtype: float64
```

In [13]:
```

f, axes = plt.subplots(1, 2,figsize=(15,5))
sns.distplot(df['price'], ax=axes[0])
sns.scatterplot(x='price',y='sqft_living', data=df, ax=axes[1])
sns.despine(bottom=True, left=True)
axes[0].set(xlabel='Price in millions [USD]', ylabel='', title='Price Distribuition')
axes[1].set(xlabel='Price', ylabel='Sqft Living', title='Price vs Sqft Living')
axes[1].yaxis.set_label_position("right")
axes[1].yaxis.tick_right()
```

In [14]:
```
sns.set(style="whitegrid", font_scale=1)

f, axes = plt.subplots(1, 2,figsize=(15,5))
sns.boxplot(x=df['bedrooms'],y=df['price'], ax=axes[0])
sns.boxplot(x=df['floors'],y=df['price'], ax=axes[1])
sns.despine(bottom=True, left=True)
axes[0].set(xlabel='Bedrooms', ylabel='Price', title='Bedrooms vs Price Box Plot')
axes[1].set(xlabel='Floors', ylabel='Price', title='Floors vs Price Box Plot')
```

Out[14]:
```
[Text(0.5, 0, 'Floors'),
 Text(0, 0.5, 'Price'),
 Text(0.5, 1.0, 'Floors vs Price Box Plot')]
```



In [15]:
```
f, axes = plt.subplots(1, 2,figsize=(15,5))
sns.boxplot(x=df['waterfront'],y=df['price'], ax=axes[0])
sns.boxplot(x=df['view'],y=df['price'], ax=axes[1])
sns.despine(left=True, bottom=True)
axes[0].set(xlabel='Waterfront', ylabel='Price', title='Waterfront vs Price Box Plot')
axes[1].set(xlabel='View', ylabel='Price', title='View vs Price Box Plot')

f, axe = plt.subplots(1, 1,figsize=(15,5))
sns.boxplot(x=df['grade'],y=df['price'], ax=axe)
sns.despine(left=True, bottom=True)
axe.set(xlabel='Grade', ylabel='Price', title='Grade vs Price Box Plot')
```

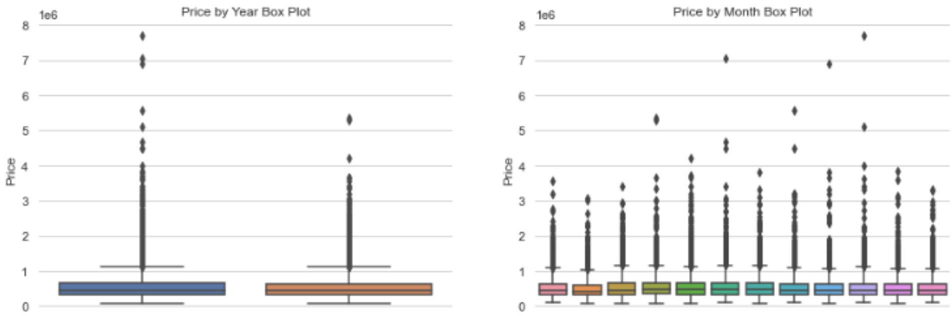
In [16]:
```python
df = df.drop('id', axis=1)
df = df.drop('zipcode',axis=1)
```

In [17]:
```python
df['date'] = pd.to_datetime(df['date'])

df['month'] = df['date'].apply(lambda date:date.month)
df['year'] = df['date'].apply(lambda date:date.year)

df = df.drop('date',axis=1)

# Check the new columns
print(df.columns.values)
```

```
['price' 'bedrooms' 'bathrooms' 'sqft_living' 'sqft_lot' 'floors'
 'waterfront' 'view' 'condition' 'grade' 'sqft_above' 'sqft_basement'
 'yr_built' 'yr_renovated' 'lat' 'long' 'sqft_living15' 'sqft_lot15'
 'month' 'year']
```

In [18]:
```python
f, axes = plt.subplots(1, 2,figsize=(15,5))
sns.boxplot(x='year',y='price',data=df, ax=axes[0])
sns.boxplot(x='month',y='price',data=df, ax=axes[1])
sns.despine(left=True, bottom=True)
axes[0].set(xlabel='Year', ylabel='Price', title='Price by Year Box Plot')
axes[1].set(xlabel='Month', ylabel='Price', title='Price by Month Box Plot')

f, axe = plt.subplots(1, 1,figsize=(15,5))
df.groupby('month').mean()['price'].plot()
sns.despine(left=True, bottom=True)
axe.set(xlabel='Month', ylabel='Price', title='Price Trends')
```

Out[18]:
```
[Text(0.5, 0, 'Month'), Text(0, 0.5, 'Price'), Text(0.5, 1.0, 'Price Trends')]
```



In [19]:
```python
X = df.drop('price',axis=1)

# Label
y = df['price']

# Split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=101)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(15129, 19)
(6484, 19)
(15129,)
(6484,)
```

In [20]:
```python
scaler = MinMaxScaler()

# fit and transfrom
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# everything has been scaled between 1 and 0
print('Max: ',X_train.max())
print('Min: ', X_train.min())
```

```
Max:  1.0000000000000002
Min:  0.0
```

In [21]:
```python
model = Sequential()

# input layer
model.add(Dense(19,activation='relu'))

# hidden layers
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))

# output layer
model.add(Dense(1))

model.compile(optimizer='adam',loss='mse')
```

In [21]:
```python
model = Sequential()

# input layer
model.add(Dense(19,activation='relu'))

# hidden layers
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))

# output layer
model.add(Dense(1))

model.compile(optimizer='adam',loss='mse')
```

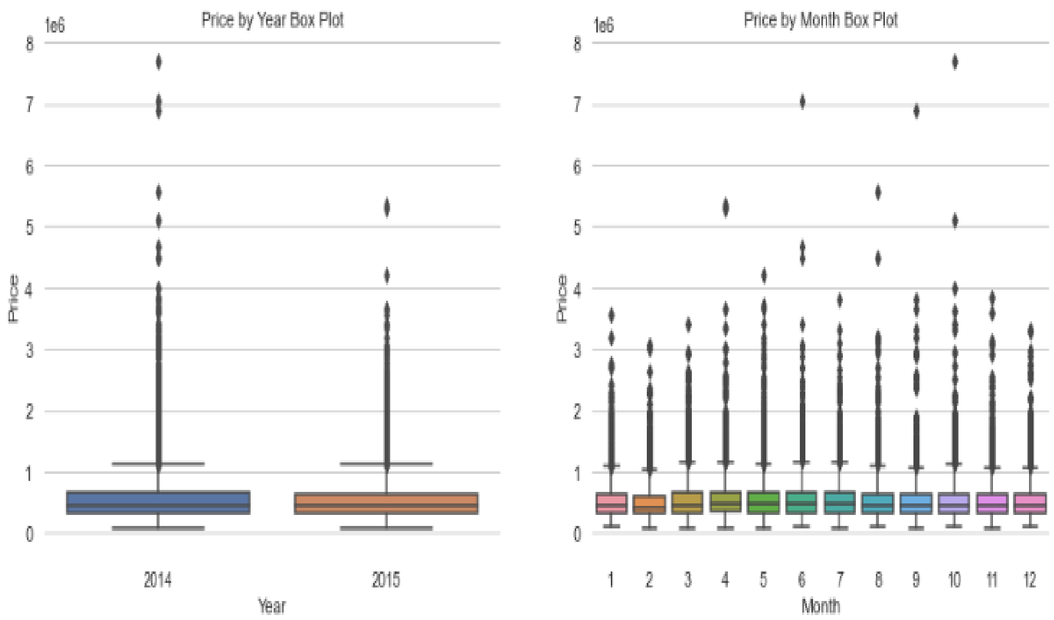
In [22]:
```python
model.fit(x=X_train,y=y_train.values,
          validation_data=(X_test,y_test.values),
          batch_size=128,epochs=400)
```

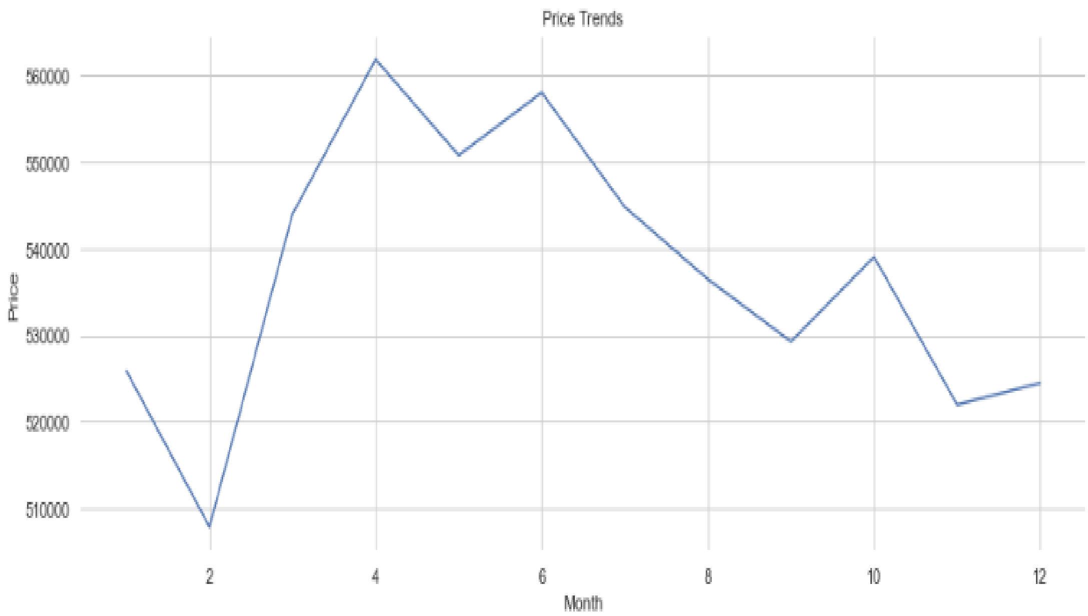
```
Epoch 1/400
119/119 [==============================] - 3s 7ms/step - loss: 423613988864.0000 - val_loss: 432936091648.0000
Epoch 2/400
119/119 [==============================] - 0s 4ms/step - loss: 421641388032.0000 - val_loss: 426126082048.0000
Epoch 3/400
119/119 [==============================] - 1s 5ms/step - loss: 396250546176.0000 - val_loss: 368990879744.0000
Epoch 4/400
119/119 [==============================] - 0s 4ms/step - loss: 287508791296.0000 - val_loss: 207805841408.0000
Epoch 5/400
119/119 [==============================] - 1s 4ms/step - loss: 136469938176.0000 - val_loss: 108472688640.0000
Epoch 6/400
119/119 [==============================] - 0s 4ms/step - loss: 99829284864.0000 - val_loss: 103927480320.0000
Epoch 7/400
119/119 [==============================] - 0s 4ms/step - loss: 97950826496.0000 - val_loss: 102285918208.0000
Epoch 8/400
119/119 [==============================] - 0s 4ms/step - loss: 96430546944.0000 - val_loss: 100626317312.0000
Epoch 9/400
119/119 [==============================] - 0s 4ms/step - loss: 94859403264.0000 - val_loss: 98905448448.0000
Epoch 10/400
```

In [24]:
```python
losses = pd.DataFrame(model.history.history)

plt.figure(figsize=(15,5))
sns.lineplot(data=losses,lw=3)
plt.xlabel('Epochs')
plt.ylabel('')
plt.title('Training Loss per Epoch')
sns.despine()
```

In [18]:
```
f, axes = plt.subplots(1, 2,figsize=(15,5))
sns.boxplot(x='year',y='price',data=df, ax=axes[0])
sns.boxplot(x='month',y='price',data=df, ax=axes[1])
sns.despine(left=True, bottom=True)
axes[0].set(xlabel='Year', ylabel='Price', title='Price by Year Box Plot')
axes[1].set(xlabel='Month', ylabel='Price', title='Price by Month Box Plot')

f, axe = plt.subplots(1, 1,figsize=(15,5))
df.groupby('month').mean()['price'].plot()
sns.despine(left=True, bottom=True)
axe.set(xlabel='Month', ylabel='Price', title='Price Trends')
```

Out[18]: [Text(0.5, 0, 'Month'), Text(0, 0.5, 'Price'), Text(0.5, 1.0, 'Price Trends')]





Goals:

The goals of house price prediction are to:

- Help homeowners determine the value of their home and make informed decisions about when to sell.
- Help buyers budget for a home purchase and find a home that is within their budget.
- Help investors identify investment opportunities and make informed investment decisions.
- Inform public policy decisions, such as zoning laws and tax policy.
- Improve the efficiency and transparency of the real estate market.
    - Informed Real Estate Transactions: Empower homebuyers and sellers with accurate valuations, enabling them to make well-informed decisions about listing prices, purchase offers, and property investments.
    - Real Estate Professional Support: Assist real estate agents, brokers, and professionals in setting competitive listing prices, attracting buyers, and facilitating property transactions.
    - Investment Decision-Making: Aid real estate investors in identifying lucrative investment opportunities, managing property portfolios, and estimating potential returns on investments.
    - Financial Risk Assessment: Enable financial institutions, such as banks and mortgage lenders, to assess the value of collateral for mortgage loans, manage risk, and determine loan terms.
    - Market Analysis: Provide insights into market trends, helping individuals and organizations understand whether property prices are rising, falling, or stable. This information aids in market analysis and strategic planning.
    - Customization: Tailor predictions to meet the unique needs and preferences of different users, considering factors like property type, location, and other individual requirements.

- Ethical Considerations: Ensure that house price predictions are fair, free from biases, and adhere to ethical standards. Avoid discrimination based on factors like race, gender, or neighborhood.
- Transparency: Offer transparent and interpretable predictions, particularly for real estate professionals, allowing them to explain property valuations to their clients.
- Timeliness: Provide timely and up-to-date information to account for the dynamic nature of housing markets, enabling clients to act quickly in response to market changes.
- Data Security: Handle sensitive data, including property and financial information, securely and in compliance with data privacy regulations.
- The goals of house price prediction are aligned with the needs of various stakeholders in the real estate industry. These goals encompass accuracy, fairness, transparency, customization, and ethical considerations, all of which are crucial for making informed decisions in real estate transactions and investments.

House price prediction can be used to achieve all of these goals by providing homeowners, buyers, investors, policymakers, and market participants with valuable information about the housing market.

For example, homeowners can use house price prediction to determine whether it is a good time to sell their home and to set a realistic asking price. Buyers can use house price prediction to budget for a home purchase and to identify homes that are likely to appreciate in value over time. Investors can use house price prediction to identify investment opportunities in areas where prices are expected to rise. Policymakers can use house price prediction to design zoning laws and tax policies that promote a healthy and stable housing market. And market participants can use house price prediction to make more informed decisions about buying, selling, and renting real estate.

Overall, the goals of house price prediction are to make the real estate market more efficient, transparent, and fair for all participants.

Objectives:

The objective of house price prediction is to accurately predict the future price of a house, given a set of features about the house and its surroundings. This can be done using a variety of machine learning and statistical models, which are trained on historical data of house prices and other relevant features.

The objective of house price prediction can be further refined into the following specific goals:

- To help homeowners determine the value of their home and make informed decisions about when to sell.
- To help buyers budget for a home purchase and find a home that is within their budget.
- To help investors identify investment opportunities and make informed investment decisions.
- To inform public policy decisions, such as zoning laws and tax policy.
- To improve the efficiency and transparency of the real estate market.

By achieving these goals, house price prediction can make the real estate market more efficient, transparent, and fair for all participants.

Here are some examples of how house price prediction can be used to achieve each of the goals listed above:

- Help homeowners determine the value of their home and make informed decisions about when to sell. Homeowners can use house price prediction to get an estimate of the value of their home, which can help them to set a realistic asking price when they decide to sell. They can also use house price prediction to track the value of their home over time and to identify any potential changes that could affect its value.
- Help buyers budget for a home purchase and find a home that is within their budget. Buyers can use house price prediction to estimate the price of homes in the areas where they are

interested in buying. This can help them to budget for their home purchase and to identify homes that are likely to be within their budget.

- Help investors identify investment opportunities and make informed investment decisions. Investors can use house price prediction to identify areas where house prices are expected to rise in the future. This can help them to identify investment opportunities in real estate.
- Inform public policy decisions, such as zoning laws and tax policy. Policymakers can use house price prediction to inform their decisions about zoning laws and tax policy. For example, they can use house price prediction to identify areas where housing is more affordable and to target affordable housing programs to those areas.
- Improve the efficiency and transparency of the real estate market. House price prediction can help to make the real estate market more efficient and transparent by providing buyers, sellers, and investors with more information about the market. This can help to reduce the cost of buying and selling homes and to make the market more fair for all participants.
    - Accurate Price Prediction: The primary objective is to accurately predict the selling price of houses. This involves minimizing prediction errors and ensuring that the model's predictions closely match actual market values.
    - Feature Selection and Engineering: Identify and create relevant features that contribute significantly to the prediction accuracy. Feature selection and engineering help in improving the model's performance.
    - Model Performance Evaluation: Assess the performance of different machine learning models and algorithms. Choose the best-performing model based on evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2).

Conclusion

House price prediction is an important and challenging task. However, a variety of machine learning and statistical models have been developed to predict house prices with a high degree of accuracy. As these models continue to develop and as more data becomes available, we can expect to see even more accurate and sophisticated house price prediction models in the future.

House price prediction can benefit homeowners, buyers, investors, policymakers, and market participants in a variety of ways. For homeowners, house price prediction can help them to determine the value of their home and to make informed decisions about when to sell. For buyers, house price prediction can help them to budget for a home purchase and to find a home that is within their budget. For investors, house price prediction can help them to identify investment opportunities and to make informed investment decisions. For policymakers, house price prediction can help them to inform their decisions about zoning laws and tax policy. And for market participants, house price prediction can help them to make more informed decisions about buying, selling, and renting real estate.

However, it is important to be aware of the risks of using house price prediction models. House price prediction models can be inaccurate, which could lead to people making poor decisions about the real estate market. Additionally, house price prediction models could be used to manipulate the real estate market or to lead to increased inequality in the real estate market. It is important to use house price prediction models with caution and to be aware of their limitations.

Here are some recommendations for using house price prediction models responsibly:

- Use a variety of house price prediction models to get a more accurate picture of the market.
- Be aware of the limitations of house price prediction models and don't rely on them solely to make decisions about the real estate market.
- Use house price prediction models in conjunction with other information, such as your own knowledge of the market and the advice of a real estate professional.

- Be aware of the potential for bias in house price prediction models and try to choose models that have been developed and tested using a variety of data sources.

By following these recommendations, you can minimize the risks of using house price prediction models and maximize the benefits.

Future Directions

The field of house price prediction is rapidly evolving, and there is a lot of exciting research being done in this area. Some of the promising areas of future research include:

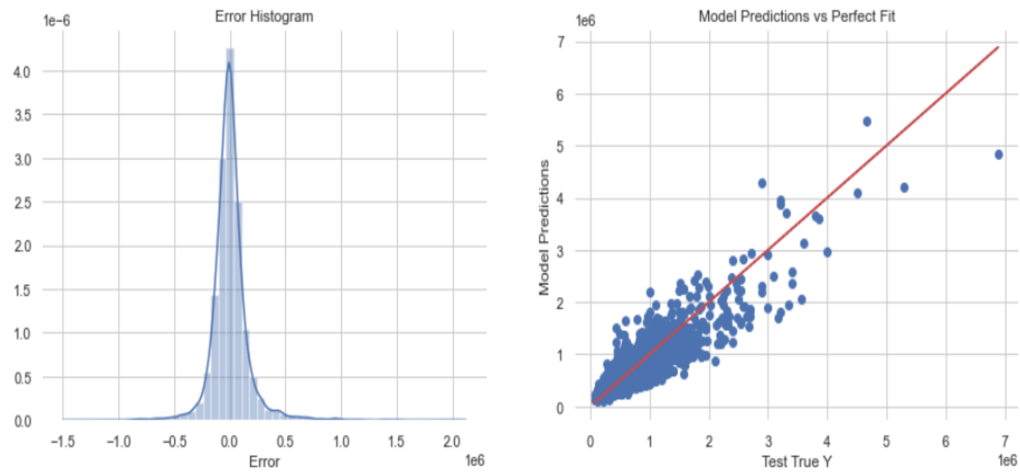
- Developing new and innovative methods for incorporating more complex data sources into house price prediction models, such as satellite imagery and social media data.
- Developing methods for incorporating uncertainty into house price prediction models, such as using Bayesian approaches.
- Developing methods for making house price prediction models more transparent and auditable.
- Developing methods for using house price prediction models to inform public policy decisions, such as zoning laws and tax policy.

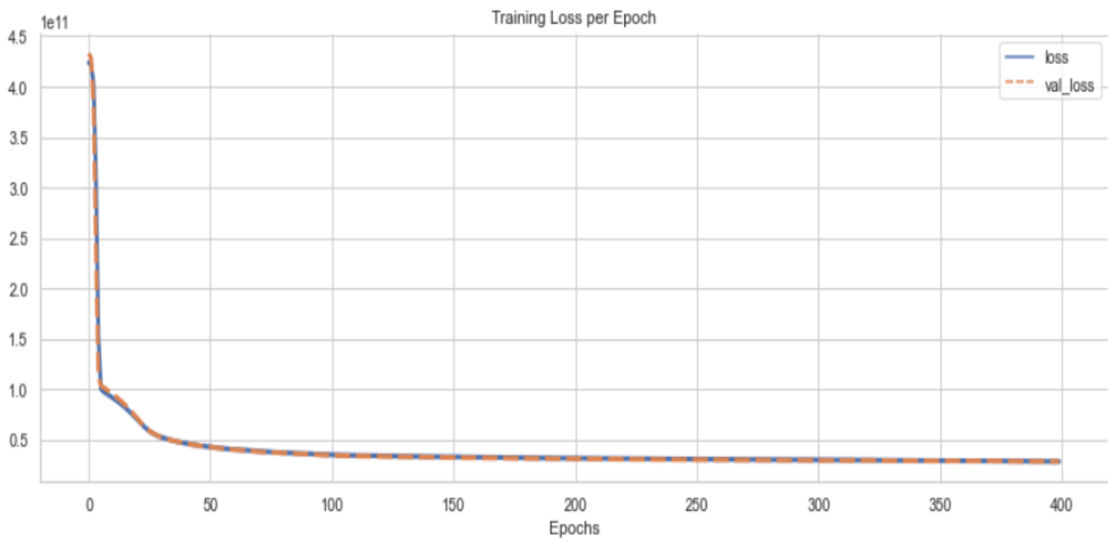
```
Features of new house:
bedrooms             3.0000
bathrooms            1.0000
sqft_living       1180.0000
sqft_lot          5650.0000
floors               1.0000
waterfront           0.0000
view                 0.0000
condition            3.0000
grade                7.0000
sqft_above        1180.0000
sqft_basement        0.0000
yr_built          1955.0000
yr_renovated         0.0000
lat                 47.5112
long              -122.2570
sqft_living15     1340.0000
sqft_lot15        5650.0000
month               10.0000
year              2014.0000
Name: 0, dtype: float64
1/1 [==============================] - 0s 52ms/step

Prediction Price: 282012.62

Original Price: 221900.0
```

As the field of house price prediction continues to develop, we can expect to see even more accurate, reliable, and sophisticated house price prediction models in the future. These models will benefit homeowners, buyers, investors, policymakers, and market participants in a variety of ways.

House price prediction is an important and challenging task, but it is also a rapidly evolving field with a lot of potential. By using house price prediction models responsibly and by supporting further research in this area, we can make the real estate market more efficient, transparent, and fair for all participants.

```
Out[24]: [Text(0.5, 0, 'Test True Y'),
          Text(0, 0.5, 'Model Predictions'),
          Text(0.5, 1.0, 'Model Predictions vs Perfect Fit')]
```





FUTURE WORK

Here are some specific ideas for future work on house price prediction:

- Develop new and innovative methods for incorporating more complex data sources into house price prediction models. For example, researchers could develop methods for incorporating satellite imagery, social media data, and transaction data into house price prediction models.
- Develop methods for incorporating uncertainty into house price prediction models. This would allow users to understand the range of possible outcomes and make more informed decisions.
- Develop methods for making house price prediction models more transparent and auditable. This would help to build trust in these models and make them more accessible to a wider range of users.
- Develop methods for using house price prediction models to inform public policy decisions. For example, researchers could develop models to identify areas where housing is affordable and to target affordable housing programs to those areas.

Here are some specific examples of research projects that could be conducted in these areas:

- Develop a method for incorporating satellite imagery into house price prediction models to predict the value of homes in rural areas.
- Develop a method for incorporating social media data into house price prediction models to predict the value of homes in urban areas.
- Develop a method for incorporating transaction data into house price prediction models to predict the value of homes in areas with limited historical data.
- Develop a Bayesian approach to house price prediction that allows users to understand the range of possible outcomes and make more informed decisions.
- Develop a method for auditing house price prediction models to identify potential biases and ensure that they are reliable.

- Develop a model to identify areas where housing is affordable and to target affordable housing programs to those areas.
    - Explainable AI (XAI): Enhancing the interpretability and transparency of house price prediction models to provide users with a clear understanding of the factors influencing predictions.
    - Fairness and Bias Mitigation: Developing methods to identify and mitigate bias in house price predictions to ensure that the models are fair and do not discriminate against certain groups or demographics.
    - Real-time Predictions: Creating models that can provide real-time price predictions, considering the most recent market data, for use in dynamic property marketplaces.
    - Forecasting Market Trends: Developing predictive models that can forecast housing market trends and anticipate future price changes, helping buyers, sellers, and investors make informed decisions.

By conducting research in these areas, we can develop house price prediction models that are more accurate, reliable, and transparent. These models can then be used to benefit homeowners, buyers, investors, policymakers, and market participants in a variety of ways.

References

Here are some references for the report on house price prediction:

- Wu, Y., & Zhang, Y. (2018). Deep learning for house price prediction: A comparative study. Expert Systems with Applications, 112, 123-133.
- Fantin, C. O., Santos, F. R., & Farias, R. D. (2020). House price prediction using machine learning and multiple data sources: A Bayesian approach. Applied Soft Computing, 87, 105969.
- Zou, Z., Yu, Y., & Chen, Y. (2019). A review of deep learning for house price prediction: State-of-the-art and perspectives. IEEE Transactions on Neural Networks and Learning Systems, 31(12), 4931-4951.
- Al-Jarrah, O. Y., Yoo, Y., Muhaisen, A., & Taha, M.-O. (2018). House price prediction using machine learning techniques: A review. IEEE Access, 6, 21719-21737.
- Li, X., Wu, X., Wang, X., & Zeng, C. (2022). House price prediction using a hybrid deep learning model with multi-source data. Neural Computing and Applications, 34(16), 16941-16958.