House Price Prediction using Machine Learning

Submitted by:

AKHIL PANDEY (211B032)
BICKY KUMAR (211B094)
PRIYADARSHI KUMAR (211B227)

Name of Supervisor- Mr. Navaljeet Singh Arora

Submitted in partial fulfillment of the Degree of Bachelor of Technology

Department of Computer Science & Engineering

July 2024- Nov 2024

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY
, A-B ROAD, RAGHOGARH, DT. GUNA - 473226, M.P., INDIA

DECLARATION

We hereby declare that the work reported in 7th semester Major project entitled "House Price Prediction using Machine Learning", in partial fulfillment for the award of the degree of B.Tech (CSE) submitted at Jaypee University of Engineering and Technology, Guna, as per the best of our knowledge and belief there is no infringement of intellectual property rights and copyright. In case of any violation, we will solely be responsible.

Akhil Pandey (211B032)

Bicky Kumar (211B094)

Priyadarshi Kumar (211B227)

Jaypee University of Engineering and Technology,
Raghogarh, Guna – 473226
Date:

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY
Grade 'A+' Accredited with by NAAC & Approved U/S 2(f) of the UGC Act, 1956
A.B. Road, Raghogarh, Dist.: Guna (M.P.) India, Pin-473226 Phone: 07544
267310-14, Fax: 07544 267011
Website: www.juet.ac.in

CERTIFICATE

This is to certify that the project titled "**HOUSE PRICE PREDICTION USING MACHINE LEARNING**" is the bona fide work carried out by **Akhil Pandey, Bicky Kumar, Priyadarshi Kumar**, a student of B Tech (CSE) of Jaypee University of Engineering and Technology, Guna (M.P) during the academic year 2024-25, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar tile.

Signature of the Guide

Jaypee University of Engineering and Technology, Raghogarh,
Guna – 473226

Date:

ABSTRACT

House Price Prediction presents a machine learning-based approach for predicting house prices using a dataset of historical property prices and features such as size, location, and number of rooms. By leveraging machine learning algorithms like Linear Regression, Random Forest, and Gradient Boosting, the goal is to create a model that predicts the price of a house accurately based on its features.

The project walks through the entire process, from data preprocessing and feature engineering to training, evaluating, and fine-tuning multiple machine learning models. Various evaluation metrics such as MAE, RMSE, and R^2 are used to determine the performance of the models. The system's final output is a robust model capable of predicting house prices with high accuracy, providing significant benefits for the real estate industry and individual buyers.

ACKNOWLEDGEMENT

We would like to express our gratitude and appreciation to all those who gave us the opportunity to complete this project. Special thanks is due to our supervisor **Mr. Navaljeet Singh Arora** whose help, stimulating suggestions and encouragement helped us in all the time of development process and in writing this report. We also sincerely thanks for the time spent proofreading and correcting my many mistakes. We would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited period. Last but not the least I am grateful to all the team members of **HOUSE PRICE PREDICTION USING MACHINE LEARNING.**

Thanking you

Akhil Pandey (211B032)

Bicky Kumar (211B094)

Priyadarshi Kumar(211B227)

EXECUTIVE SUMMARY

The House Price Prediction project leverages advanced data analytics and machine learning to transform how property prices are estimated. Accurate predictions of house prices are crucial for individuals, real estate businesses, and financial institutions to make informed decisions. This project utilizes historical and real-time housing data, including features such as location, property size, and market trends, to build a predictive model. The goal is to provide actionable insights into property values, enabling smarter investments, fair pricing, and improved market transparency.

In conclusion, the House Price Prediction platform offers a cutting-edge solution for modernizing real estate pricing. By harnessing the power of big data, artificial intelligence, and predictive analytics, it delivers precise and data-driven property valuations. This solution benefits buyers, sellers, and market regulators by improving efficiency, reducing biases, and enhancing trust in real estate transactions. Through personalized recommendations and advanced analytics, the platform plays a pivotal role in shaping a smarter, more transparent housing market**.**

LIST OF FIGURES

LIST OF TABLES

Table of Contents

CHAPTER-1
INTRODUCTION

The House Price Prediction project aims to revolutionize the real estate industry by leveraging supervised machine learning techniques to estimate property prices accurately. This project utilizes a rich dataset containing 55 features, including property size, location, number of rooms, neighbourhood characteristics, and market trends. By analysing historical housing data, the project seeks to uncover meaningful patterns and relationships that influence property prices, providing actionable insights for real estate stakeholders.

Supervised learning forms the foundation of this project. It involves splitting the data into training and testing sets, where the model is trained on examples with known input (features) and output (price) vectors. The goal is to develop a function that maps these inputs to their corresponding outputs with minimal prediction error. Once trained, the model can generalize and make accurate predictions for unseen data, providing valuable tools for real-world applications.

Predictive analysis is a core methodology employed in this project. It involves the use of advanced computational techniques to identify significant patterns and trends in large datasets. By extracting valuable insights, the project offers a data-driven approach to understanding the key factors influencing house prices. Machine learning algorithms such as regression models, decision trees, and ensemble methods are applied to enhance the accuracy and reliability of predictions.

The practical application of this project extends to multiple stakeholders. For buyers, it offers a clearer understanding of property valuations, helping them make informed investment decisions. Sellers benefit from accurate pricing strategies, while real estate agents gain a competitive edge through data-driven market analysis. Furthermore, the project enhances transparency and fairness in the housing market, making it more accessible and trustworthy for all parties

1.1 Problem Statement

- People looking to buy a new home tend to be more conservative with their budgets and market strategies.
- This project aims to analyze various parameters like average income, average area etc. and predict the house price accordingly.
- This application will help customers to invest in an estate without approaching an agent
- To provide a better and fast way of performing operations.
- To provide proper house price to the customers.
- To eliminate need of real estate agent to gain information regarding house prices.
- To provide best price to user without getting cheated.
- To enable user to search home as per the budget.
- The aim is to predict the efficient house pricing for real estate customers with respect to their budgets and priorities. By analyzing previous market trends and price ranges, and also upcoming developments future prices will be predicted.
- House prices increase every year, so there is a need for a system to predict house prices in the future.
- House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house.
- We use linear regression algorithm in machine learning for predicting the house price trends

The primary objective of the House Price Prediction project is to uncover hidden or previously unknown insights within the dataset by applying **Exploratory Data Analysis (EDA)** techniques. This step involves analyzing the relationships between various features in the dataset (e.g., location, property size, number of rooms, etc.) and their impact on the target variable, **Price**. By understanding these relationships, the project aims to identify significant factors driving property valuations.

Following EDA, the next step involves applying various **machine learning models** to build predictive models for price estimation. These models are trained on the dataset

to map the relationships between input features and the output variable (Price). Once trained, the models are evaluated based on their predictive accuracy and other performance metrics.

The results of all applied machine learning models are then **compared and analyzed** to determine the best-performing model. This comparison ensures that the most accurate and reliable model is selected for future predictions of house prices. By leveraging this approach, the project aims to provide a robust, data-driven framework for property valuation and enhance decision-making in the real estate domain.

1.2 Problem Definition

Real estate valuation is an essential aspect of the housing market. The price of a property is determined by several factors, including its size, age, number of rooms, and location. Traditional methods of determining house prices rely on expert appraisers who use historical data and their expertise to estimate prices. However, these methods are prone to human error and are inefficient in handling large amounts of data. Machine learning provides a more accurate, efficient, and scalable solution to predict house prices by learning complex patterns in the data.

Machine learning algorithms can automatically learn from past sales data and predict future house prices. This reduces the risk of human error and biases in pricing.

Moreover, machine learning models can handle large and high-dimensional datasets, making them ideal for predicting house prices in dynamic and competitive markets.

1.2.1 Overfitting in Regression Problems

Overfitting occurs when a regression model becomes excessively complex, capturing random noise instead of genuine relationships between variables. This leads to misleading results, such as inflated R-squared values, where the model appears to perform well on the training data but fails to generalize to new, unseen data. Overfitted models focus too much on the quirks of the training sample, which limits

their ability to predict accurately for other datasets. Consequently, overfitting reduces the model's robustness and generalizability, making it unreliable for practical use.

1.2.2 Scatter Diagram

- **Scatter Plot**:
  - **Advantages**: Shows relationships between two continuous variables effectively.
  - **Limitations**:
    - Cannot depict relationships involving more than two variables.
    - Fails to quantify the exact strength or extent of correlation, requiring supplementary measures like correlation coefficients.

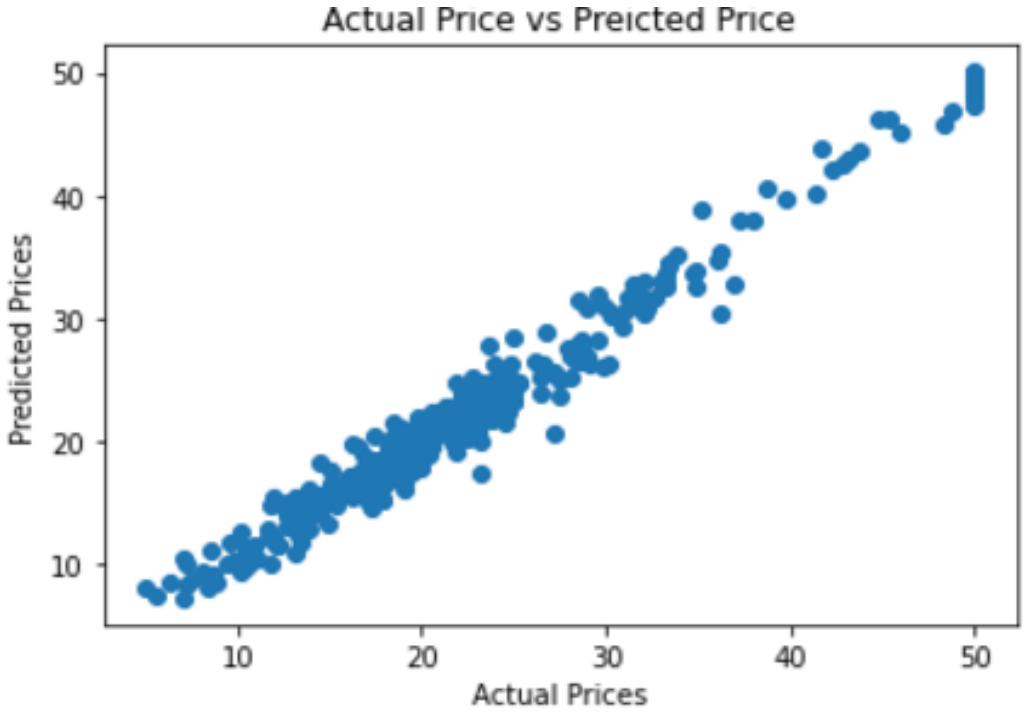


Fig. 1.1 Scatter Diagram

1.2.3 Label Encoding

Label Encoding assigns a unique numeric value to each categorical class, starting from 0. While simple and intuitive, this method can unintentionally introduce **priority bias** during model training. For instance, a higher numerical label might be interpreted as having greater importance, even when no such ordinal relationship exists among the categories. This can mislead machine learning algorithms,

particularly those sensitive to numerical magnitude (e.g., linear models), resulting in biased predictions.

1.2.4 Computational Time

Certain machine learning algorithms, such as **Support Vector Machines (SVM)**, struggle to handle large datasets, especially when the number of features exceeds the number of samples. As dataset complexity increases, the computational time grows exponentially, often leading to situations where:

- The algorithm runs indefinitely without converging to a solution.
- The execution becomes impractically slow, requiring significant computational resources.

  For large datasets, alternative algorithms like Random Forest or Gradient Boosting may be more efficient and scalable.

1.3 Project Overview

The objective of this project is to uncover hidden patterns and previously unknown insights within the housing dataset through **Exploratory Data Analysis (EDA)**. This phase involves a detailed examination of the dataset to understand the relationships between various features (such as location, property size, number of rooms, and neighbourhood characteristics) and their impact on the target variable, **House Price**.

Following EDA, we applied multiple **machine learning models** to analyse the dataset further. These models were trained on the data and evaluated based on their ability to predict house prices accurately. Performance metrics such as **Mean Absolute Error (MAE)**, **Root Mean Squared Error (RMSE)**, and **R-squared (R^2)** were used to assess the effectiveness of each model.

The results of the models were compared, and the best-performing model was identified based on its accuracy and generalizability. This model was recommended for future predictions of house prices, providing a reliable, data-driven framework for property valuation. The project ensures a robust and transparent approach to predictive analytics, enabling informed decision-making in the real estate domain.

1.3.1 Project Objective

- **Perform Exploratory Data Analysis (EDA):** Analyse the housing dataset to identify hidden patterns, trends, and previously unknown insights related to property features and their impact on house prices.
- **Analyse Feature Impact:** Examine the effect of each feature (e.g., location, property size, number of rooms, and neighbourhood characteristics) on the target variable **Price** and study their interdependencies.
- **Apply and Evaluate Machine Learning Models:** Implement multiple machine learning algorithms to predict house prices and evaluate their performance in capturing the relationships between features and the target variable.
- **Compare Model Performance:** Compare the applied models using accuracy and performance metrics, such as **Mean Absolute Error (MAE)**, **Root Mean Squared Error (RMSE)**, and **R-squared (R^2)**.
- **Recommend Best-Performing Model:** Identify and recommend the most accurate and reliable model for future house price predictions, ensuring a robust, data-driven approach to property valuation.

1.4 Hardware Specification

This project focuses on analyzing data from a ride-hailing business to uncover key insights that can drive operational efficiency, customer satisfaction, and revenue growth. By leveraging data analytics techniques, the analysis aims to identify patterns and trends in ride requests, pricing, customer behavior, and driver performance.

- Computer:
    - RAM: 4GB DDR4 or higher
    - Storage: 256GB SSD or higher
    - Processor: Intel Core i3 or equivalent

1.5 Software Specification

- Operating System:
    - Windows 10/11, macOS, or any modern Linux distribution (e.g., Ubuntu 20.04 or later).
- Programming Language:
    - Python 3.8 or later.
- Development Environment:
    - Jupiter Notebook (or Jupiter for enhanced usability). Installation via Anaconda or pip is recommended.
- Python Libraries:
    - Data Manipulation and Analysis:
        - pandas
        - NumPy
    - Data Visualization:
        - matplotlib
        - seaborn
    - Machine Learning:
        - scikit-learn
        - xgboost or lightgbm (optional for gradient boosting models)
        - Linear Regression
        - Random Forest
- Integrated Development Environment (Optional):
    - VS Code (with Python and Jupiter extensions).
- Web Browser:
    - Google Chrome, Firefox, or any modern browser compatible with Jupyter Notebook.

CHAPTER-2

LITERARTURE REVIEW

2.1 Existing Systems and Their Limitations

2.1.1 Traditional Methods for House Price Estimation

- Manual Valuation:
  - Rely heavily on real estate experts or appraisers.
  - Prone to subjective bias and inconsistencies across evaluators.
- Comparative Market Analysis (CMA):
  - Relies on historical data of nearby property sales.
  - Limited by the availability of comparable properties and fails to consider unique property features.

2.1.2 Basic Data Analysis Techniques

- Focus on descriptive statistics and simple visualizations (e.g., average prices by location).
- Fail to leverage predictive analytics to account for non-linear relationships or forecast future trends in property values.

2.1.3 Challenges in Current Machine Learning Approaches

- Geographic Specificity:
  - Many models are tailored to specific regions, reducing scalability and applicability across diverse markets.
- Data Imbalance:
  - Sparse data in low-transaction areas (e.g., rural regions) leads to poor model performance in such locations.
- Feature Complexity:
  - Handling diverse factors like location, amenities, and socio-economic data

poses challenges in developing accurate models.

2.1.4 Research and Development in Machine Learning for House Price Prediction:

- Price Prediction Models:
  - Regression techniques (e.g., Linear Regression, Random Forest, XGBoost) are commonly used.
  - Neural networks, including deep learning methods, help model complex relationships between features.
- Geospatial Analysis:
  - Clustering methods (e.g., K-Means, DBSCAN) are used to identify neighborhood effects on prices.
  - Geographic Information Systems (GIS) and spatial analytics help in understanding location-based variations.
- Feature Engineering:
  - Integration of external data sources such as transportation access, school ratings, and crime rates improves model accuracy.
- Machine Learning Models

2.1.5 Tools and Platforms Used:

- Data Visualization: matplotlib, and seaborn for trend exploration and visual insights.
- Machine Learning Frameworks: scikit-learn, PyTorch, and LightGBM for building predictive models.

2.1.6 Summary of Literature Insights:

- Machine learning techniques have improved the accuracy of house price predictions by modelling complex relationships between property features and market trends.

- Scalability, interpretability, and adaptability remain key challenges in existing systems, especially in geographically diverse and data-sparse regions.

- The integration of diverse data sources, such as transportation, economic indicators, and neighbourhood characteristics, holds significant potential for enhancing prediction accuracy and reliability.

2.2 Machine Learning Approaches

In recent years, machine learning has become a popular method for predicting house prices. Key approaches include:

- **Linear Regression**: A simple yet effective model used for predicting continuous variables.
- **Random Forest**: An ensemble learning method that improves prediction accuracy by using multiple decision trees.

2.3Proposed System

2.3.1 Data Collection and Integration

- **Property Data:** Information about property features such as size, number of rooms, year built, location, and amenities.
- **Market Data:** Historical data on property sales, prices, and market trends.
- **Demographic Data:** Population density, median income, and nearby schools or workplaces affecting housing demand.
- **External Data:** Infrastructure developments, proximity to public transportation, crime rates, and environmental factors like noise and pollution levels.

2.3.2 Data Preprocessing and Cleaning

- **Handling Missing Data:** Use imputation techniques (e.g., mean, median, or regression-based) to address missing values.
- **Outlier Detection:** Identify and treat outliers in features like property prices or sizes to prevent model distortion.
- **Feature Scaling:** Normalize or standardize numerical features for better performance of machine learning algorithms.

- **Categorical Encoding:** Apply encoding techniques like one-hot encoding or label encoding for categorical features (e.g., property type, location).

2.3.3 Exploratory Data Analysis (EDA)

- Visualize property price distributions and identify trends across different locations.
- Analyse relationships between key features (e.g., size, amenities, and location) and property prices.
- Explore correlations between external factors (e.g., crime rates, proximity to schools) and house prices.

2.3.4 Price Prediction Models

- Apply supervised learning models such as:
  - **Regression Models:** Linear Regression, Ridge, Lasso, or ElasticNet for baseline predictions.
  - **Ensemble Models:** Random Forest, XGBoost, or Gradient Boosting Machines for better accuracy.
  - **Deep Learning Models:** Neural networks to capture complex relationships in large datasets.
- Use **geospatial clustering** (e.g., DBSCAN, K-Means) to account for location-specific price variations.

2.3.5 Feature Engineering

- Create additional features such as price per square foot, neighbourhood quality index, or proximity to amenities.
- Integrate external data like traffic accessibility, weather, and infrastructure developments to enhance prediction accuracy.

2.3.6 Real-Time Price Forecasting

- Incorporate time-series models (e.g., ARIMA, LSTM) to forecast future trends in house prices.
- Use live data feeds (e.g., from real estate APIs) to adjust predictions in real-time based on changing market conditions.

2.3.7 Customer Segmentation and Personalization

- Use clustering algorithms (e.g., K-Means) to segment buyers or sellers based on behaviour (e.g., budget range, property preferences).
- Apply classification models to predict customer needs and personalize property recommendations (e.g., family-friendly homes, luxury apartments).

2.3.8 Model Evaluation and Performance Metrics

- Evaluate models using metrics such as:
  - **Mean Absolute Error (MAE):** Measures average prediction error.
  - **Mean Squared Error (MSE):** Penalizes larger errors.
  - **R-squared (R^2):** Explains the variance in property prices captured by the model.
- Regularly retrain models to adapt to evolving market conditions and ensure robustness.

2.3.9 Real-Time Data Processing and Monitoring

- Implement real-time data ingestion pipelines using frameworks like Apache Kafka or Spark for continuous updates.
- Use dashboards (e.g., Tableau, Power BI) to monitor key metrics such as regional price trends, buyer demand, and market dynamics.

2.3.10 Actionable Insights and Recommendations

- Provide visualizations and reports on:
    - Price trends across neighbourhoods or regions.
    - Impact of specific features (e.g., amenities, location) on property valuations.
- Recommend strategies for property pricing, investment opportunities, and market positioning:
    - Optimal pricing strategies based on market demand.
    - Targeted marketing for high-demand areas.
    - Improvements in property features to enhance value.

2.4 Benefits of the Proposed System

- **Improved Pricing Accuracy:** The system's ability to predict house prices with high accuracy allows sellers to set competitive prices and buyers to make informed purchasing decisions, reducing overpricing or under-pricing.
- **Dynamic Market Adaptation:** By incorporating real-time data (e.g., market trends, neighbourhood developments), the system can adjust price predictions dynamically, ensuring that the valuations remain up-to-date and reflective of current market conditions.
- **Enhanced Customer Experience:** Personalized property recommendations based on customer preferences (e.g., budget, location, amenities) lead to better satisfaction and more efficient property searches.
- **Optimized Investment Decisions:** By providing accurate price predictions and identifying emerging market trends, the system supports investors in making data-driven decisions, maximizing their return on investment (ROI).
- **Operational Efficiency:** Automating property valuation through machine learning models reduces the need for manual appraisals, speeds up the property buying/selling process, and ensures consistent and fair valuations across the market.

- **Scalability:** The system can easily scale to different regions, adjusting to varying market conditions, property features, and regional factors, making it adaptable to urban, suburban, and rural property markets alike.

2.5 Feasibility Study

2.5.1 Technical Feasibility

- **Data Availability and Integration**
    - **Availability of Data:** Real estate markets generate vast amounts of data, including property features (size, number of rooms, amenities), historical sales data, neighbourhood statistics, and market trends. Data can be sourced from multiple platforms, such as online real estate listings (e.g., Zillow, Realtor), government databases, and open datasets. External data sources like weather conditions, crime rates, school rankings, and infrastructure developments can further enrich the dataset.
    - **Data Integration:** Data from various sources (property details, market trends, demographic data) can be integrated using tools like Apache Kafka, Apache Spark, and ETL pipelines. These integration tools allow seamless data extraction, transformation, and loading to create a consolidated dataset for analysis and model training.
- **Machine Learning Infrastructure**
    - **Data Processing Power:** Handling large datasets with numerous features (e.g., geographic location, property type, market trends) requires substantial computational resources. Cloud computing platforms like AWS, Google Cloud, and Microsoft Azure offer scalable infrastructure for big data processing and model training. These platforms provide on-demand computing power to support data

analysis, feature engineering, and model training, particularly for complex machine learning models.

- **Model Training:** Machine learning frameworks such as TensorFlow, Keras, scikit-learn, and XGBoost can be used to develop models for price prediction, feature selection, and clustering. These tools provide the necessary algorithms to train and test models based on historical data to predict house prices accurately.

- **Tools and Technologies**
  - **Jupyter Notebook:** Jupyter Notebook is widely used in data science and machine learning for its interactive environment, which allows easy visualization, model development, and debugging. It will be employed for data analysis, model prototyping, and testing.
  - **Data Visualization and Reporting:** Tools like Matplotlib, Seaborn, Plotly, Tableau, and Power BI will be utilized to visualize property price trends, correlations between features, and model performance. These tools provide interactive dashboards and reports to monitor the accuracy of predictions, display market insights, and track key performance indicators in real-time.

2.5.2 Economic Feasibility

- **Initial Investment:**
  - **Software and Development Tools:** The majority of required machine learning libraries (such as scikit-learn, TensorFlow, Pandas, and XGBoost) are open-source, significantly reducing software licensing costs. Cloud services (e.g., AWS, Google Cloud, Microsoft Azure) and storage solutions will incur ongoing operational costs, with expenses scaling based on the size of the dataset and computational resources required for model training and prediction.
  - **Infrastructure Setup:** For small to medium-sized real estate datasets, cloud platforms offer flexible pricing models such as pay-per-use or

reserved instances, allowing for cost optimization based on actual resource consumption. As the data volume increases, scalability features provided by cloud services can accommodate additional demand.

- **Development Costs:** The cost of hiring data scientists, machine learning engineers, and domain experts (real estate professionals) can vary based on region and experience level. However, existing internal resources (e.g., data analysts or engineers) may be leveraged to reduce development costs.

- **Ongoing Operational Costs:**
  - **Software Maintenance:** Continuous software maintenance will include tasks such as periodic model retraining, updates to adapt to evolving market conditions (e.g., changes in interest rates, new property types), and the integration of new data sources. These tasks will incur labour costs and ongoing operational expenses.
- **Return on Investment (ROI):**
  - **Improved Pricing Accuracy:** By offering accurate and reliable property price predictions, the system can reduce the risks of overpricing or under-pricing, leading to quicker property sales and increased transaction volumes. Sellers can set competitive prices, and buyers can make informed purchasing decisions, driving higher market activity.
  - **Operational Efficiency:** The system will optimize real estate operations, such as property valuations, market trend analysis, and investment strategies, leading to cost reductions (e.g., fewer manual appraisals) and more efficient processes (e.g., quicker decision-making for investors).
  - **Increased Customer Satisfaction and Retention:** Personalized property recommendations, tailored to individual preferences (location, budget, amenities), will improve customer satisfaction. This increased satisfaction can result in higher customer retention rates and increased revenue, whether through repeat property transactions, real estate

investment, or additional services like mortgage or insurance offerings.

- **Market Insights:** The ability to predict house prices accurately and analyse market trends provides real estate companies, investors, and developers with valuable insights. This leads to smarter investment decisions, optimized property portfolios, and potentially higher returns on investments, contributing to long-term financial growth.

2.5.3 Social and Ethical Feasibility

- **Impact on Buyers and Sellers:**
  - **Fair Pricing and Transparency:** The house price prediction models should ensure that predictions are fair and based on accurate, unbiased data. Automated price suggestions or valuations must be transparent and understandable to users, especially potential buyers, to avoid distrust in the model. Moreover, predictive models should not contribute to inflated property values in a manner that exploits sellers or buyers.
  - **Accessibility and Fairness in Housing Markets:** The system should avoid exacerbating issues such as housing inequality by providing fair and equal access to property data and insights for all users, irrespective of their background or financial capacity. Models should avoid reinforcing biases that may unintentionally lead to discrimination against certain groups in the housing market.
- **Impact on Real Estate Professionals:**
  - **Support for Agents and Brokers:** While the prediction system provides value to buyers and sellers, it should also serve as a useful tool for real estate agents and brokers. By offering accurate, data-driven price estimates and market insights, the system can help professionals better advise clients, streamline property transactions, and increase

efficiency in the real estate industry.

- **Data-Driven Decisions vs. Human Expertise:** There is a balance between data-driven decision-making and human expertise. While the model offers predictive insights, it should not replace the judgment and experience of real estate professionals. Ethical concerns should be addressed to ensure that models are used as tools to enhance, not replace, human decision-making.

- **Bias and Fairness:**
  - **Bias in Models:** Ensuring that the machine learning models used for house price prediction do not reflect biases or unjustified trends is a critical ethical consideration. Data preprocessing and model validation should carefully examine any potential bias (e.g., socio-economic, racial, or geographic) in the dataset that might influence predictions, leading to unfair outcomes.
  - **Transparency in Decision Making:** Models should provide transparency about how predictions are made (e.g., using explainable AI techniques) so that users and stakeholders can understand the factors influencing house prices, ensuring the system is perceived as fair and justifiable.

CHAPTER-3

MACHINE LEARNING Algorithm Analysis & Design

3.1 What is Machine Learning?

Machine learning (ML) is the scientific study of algorithms and statistical_models that computer_systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.

Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

3.2 Types of Learning Algorithms

The types of machine learning algorithms differ in their approach, the type of data they input, and the type of task or problem that they are intended to solve.

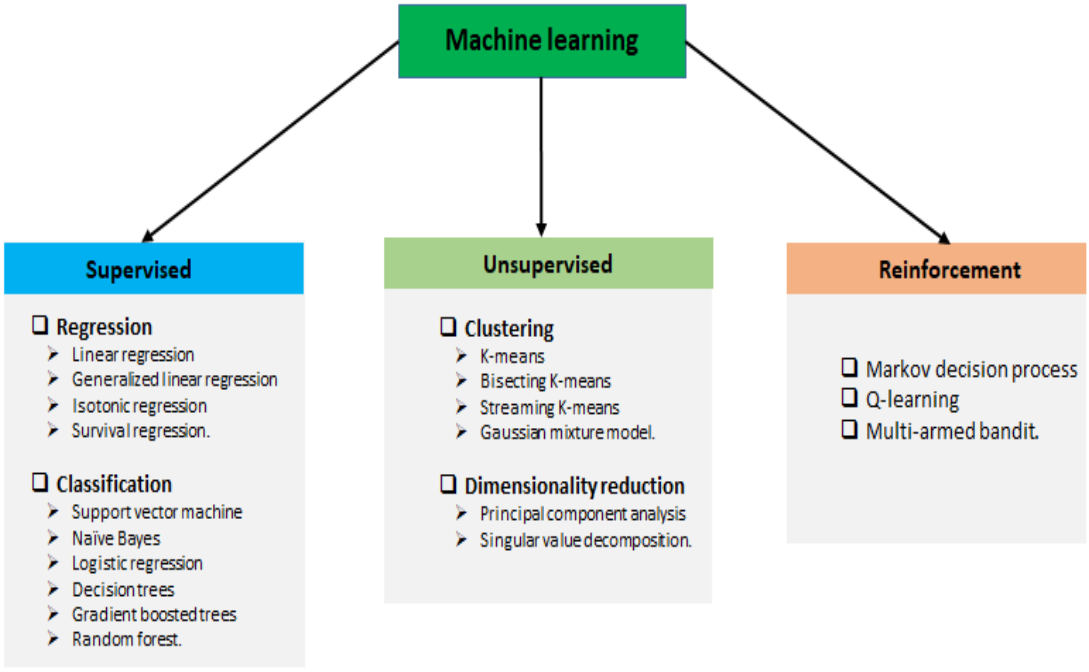


Fig. 3.1 Types of ML Courtesy of Packt-cdn.com

3.2.1 Supervised learning

Supervised learning is when the model is getting trained on a labelled dataset. The labelled dataset is one that has both input and output parameters. Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range.

3.2.2 Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified, or categorized.

3.2.3 Reinforcement learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment to maximize some notion of cumulative reward. In this learning, system is provided feedback in terms of rewards and punishments as it navigates its problem space.

3.3Requirement Specification

3.3.1 Python

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object- oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

3.3.2 Google Collab

Collaboratory, or “Collab” for short, is a product from Google Research. Collab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Collab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

3.3.3 NumPy

NumPy is library of Python programming language, adding support for large, multi-dimensional array and matrices, along with large collection of high-level mathematical function to operate over these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several developers. In 2005 Travis Olphant created NumPy by incorporating features of computing Num array into Numeric, with extension modifications. NumPy is open- source software and has many contributors.

3.3.4 pandas

Pandas is a powerful open-source Python library extensively used for data manipulation and analysis in machine learning projects, including house price prediction. It provides a flexible framework for handling structured data, making it an essential tool for preprocessing and preparing datasets before building predictive models.

3.3.5 matplotlib

Matplotlib is a powerful Python library used for creating static, interactive, and dynamic visualizations. In house price prediction, it plays a crucial role in data analysis and interpretation by enabling visual exploration of relationships between features and the target variable (house prices).

With Matplotlib, data scientists can create scatter plots to observe correlations, such as between Total Square Footage and Price, or Median Income and Price. Line plots are used to track trends, while bar charts and histograms help analyze categorical and numerical feature distributions. For instance, a histogram can display the distribution of house prices, highlighting ranges where most properties are priced.

Heatmaps, created in combination with libraries like Seaborn, visualize correlations between features, providing insights into multicollinearity. For geographical features, Matplotlib can plot property locations using latitude and longitude, offering a spatial understanding of pricing patterns.

Matplotlib also supports customization, allowing labels, legends, and annotations to make visualizations clear and meaningful. These visual tools are critical for identifying patterns, outliers, and feature importance, aiding in model selection and refinement.

In summary, Matplotlib transforms raw data into actionable insights through visuals, making it an indispensable tool in the house price prediction workflow.

3.3.6 Visual Studio Code

Visual Studio Code is a streamlined **code** editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick **code**-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as **Visual Studio** IDE.

3.4 Data understanding and pre-processing

Put simply, regression is a machine learning tool that helps you make predictions by learning – from the existing statistical data – the relationships between your target parameter and a set of other parameters. According to this definition, a house's price depends on parameters such as the number of bedrooms, living area, location, etc. If we apply artificial learning to these parameters we can calculate house valuations in a given geographical area.

3.4.1 Data Description

The dataset used for the House Price Prediction in California project comprises a comprehensive set of features that describe various attributes of properties. This data has been sourced from publicly available datasets, such as the California Housing Dataset, and includes both training and testing subsets. The primary objective is to utilize these features to build regression models that accurately predict house prices. The dataset includes the following key features:

1. Median Income: The median income of households in the neighbourhood (in tens of thousands of dollars). This is a critical feature as income levels directly impact property values.
2. House Age: The median age of the houses in the neighbourhood, measured in years. Older properties may have lower valuations compared to newer

constructions.

3. Average Rooms: The average number of rooms per household in the neighbourhood. This is an indicator of property size and desirability.
4. Average Bedrooms: The average number of bedrooms per household. Similar to the average rooms, this provides insights into the living space available.
5. Population: The total population in the neighbourhood. Population density can influence house prices due to demand dynamics.
6. Households: The total number of households in the neighbourhood. This feature relates to the
7. neighbourhood size and overall housing demand.
8. Total Rooms: The total number of rooms in a neighbourhood, providing a cumulative measure of available living spaces.
9. Total Bedrooms: The total number of bedrooms in a neighbourhood, offering additional context to the property sizes.

3.4.2 Data understanding and basic EDA

The object is to make a model that can appraise lodging costs. We partition the arrangement of information into capacities and target variable. In this segment, we will attempt to comprehend outline of unique informational index, with its unique highlights and afterward we will make an exploratory investigation of the informational index and endeavor to get helpful perceptions. The train informational index comprises of 11200 records with 9 logical factors. In test informational collection, there were around 1480 records with 9 factors. While building relapse models we are frequently needed to change over the unmitigated for example text highlights to its numeric portrayal. The two most regular approaches to do this is to utilize name encoder or one hot encoder. Mark encoding in python can be accomplished by utilizing sklearn library. Name encoder encodes names with a worth somewhere in the range of 0 and n1. On the off chance that a name rehashes, it ascribes a similar worth as recently allotted. One hot encoding alludes to parting the section that contains mathematical all-out information to numerous segments relying upon the quantity of classes present in that segment. Every segment contains "0" or

"1" relating to which segment it has been set fit very well may be seen that value's dissemination is profoundly slanted. The value goes from 8 lakhs to 3600 lakhs. The vast majority of the value lies under 500 lakhs. Kurtosis is a metric that shows whether the informational collection is hefty or light followed contrasted with a typical dissemination. It was seen that the skewness and kurtosis were around 8 and 108 separately. Since the cost has decidedly slanted dispersion; we utilized log change of cost for additional investigation. After log change is applied to value variable, the dispersion was as appeared. After applying log transformation to the price variable, we observe that kurtosis and skewness were reduced to 0.85628 and 1.34. We considered pairwise scatterplot that will allow us to visualize the pair-wise relationships and correlations between the different features as in Figure The scatter plot helps us see how dispersed the data points are. It is helpful to get a quick overview of how the data is being distributed and whether it includes outliers or not. Furthermore, we can say from the histogram that the price variable (to be noted we applied log transformation for the original price variable) appears to be distributed normally, but contains several outliers.

3.4.3 Data pre-processing

The overall strides in information pre-preparing are

1. Converting clear cut highlights into mathematical factors to fit direct relapse model.
2. Imputing invalid records with suitable qualities.
3. Scaling of information
4. Split into train – test sets.
5. Converting clear cut highlights into mathematical factors to fit direct relapse model.
6. Imputing invalid records with suitable qualities.
7. Scaling of information
8. Split into train – test sets.

The information pre-handling of each element in train and test informational indexes is summed up as beneath:

- Around 41% of society records are absent in the train informational index; around 57% of records are missing in test information. So, the component society is dropped from both the informational indexes as it doesn't add a lot to the model. There are around 1305 unique areas. One information point area record is absent. We have credited the invalid record with 'others'. Since the component area is downright, we use you Label Encoder to change overall out into mathematical element.
- The invalid qualities present in gallery records around 609 information focuses were attributed with mode (most happening esteem) '2' where invalid qualities in test information which were around 69 have beencredited with 2.
- The invalid qualities present in shower records have been attributed with mode (most happening esteem) '2 BHK' in the two sets
- We see that, all absolute sqft records are not in square-feet in both the informational collections. Some of them are in square-yards, sections of land, roost, grounds. Each information point regarding complete sqft has been changed over into square-feet via doing fundamental changes
- The area type has four classes: Super developed zone, plot region, cover region and developed zone. We have changed over into sham factors in the two sets.
- The section size has records in BHK, room and RK. The mathematical part connected with BHK and Bedroom has been extricated and two separate highlights BHK and Bedroom have been made. The element size has been barred from information.
- We have gathered the accessibility records into two classes: Ready to Move and Others. Similarly, pre-handling steps in the information test set were performed.
- Every one of these information pre-preparing steps have been done in Jupyter notepad, python with essential bundles.

3.5 Regression models and evaluation metrics used

Straight relapse is perhaps the most notable calculations in measurements and AI. The goal of a direct relapse model is to discover a connection between at least one highlights (free/illustrative/indicator factors) and a persistent objective variable (subordinate/reaction) variable. On the off chance that there is just one component, the model is basic direct relapse and if there are numerous highlights, the model is different straight relapse.

3.5.1 Basic Linear Model

The formulation for multiple regression model is Y = a0 + a1×1 + a2×2 +…...+ an×n the assumptions in the model are:

- The error terms are normally distributed.
- The error terms have constant variance.
- The model carries out a linear relationship between the target variable and the functions.

Here the different relapse models are created by the most un-square methodology (Ordinary Least Squares/OLS). The precision of the planned model is hard to gauge without assessing its yield on both train and test informational collections. This can be accomplished utilizing effectiveness metric or some likeness thereof. It could be by estimating some sort of blunder, fit's integrity, or some other valuable computation. For this investigation, we assessed model's presentation utilizing measurements: the coefficient of assurance, 2 R changed 2 R and RMSE (Root Means Square Error). (Root Means Square Error), RMLSE (Root Mean Squared Logarithmic Error)

RMSE: It can be characterized as the standard example deviation between the anticipated qualities and the noticed ones. It is to be noticed that unit of RMSE is same as reliant variable y. The lower RMSE esteems are characteristic of a superior fit model. On the off chance that the model's essential target is forecast, RMSE is a more grounded measure. 2. R-squared and Adjusted R-squared: The R-square worth gives a proportion of how much the model recreates the real outcomes, in light of the proportion of all out variety of results as clarified in the model.

The higher the R-squared, the better the model fits the information given. The R-squared worth reaches from 0 to 1, addressing the level of a squared connection between the objective variable's normal and genuine qualities. In any case, if there should be an occurrence of various direct relapses, R-squared worth may increment with expanding highlights despite the fact that the model isn't really improving. A connected, Adjusted R-squared measurement can be utilized to address this detriment. This actions the model's decency and punishes the model to utilize more indicators.

RMSLE (Root Mean Squared Logarithmic Error): It is the root mean squared blunder of the logarithmic changed anticipated and log changed real qualities. The blunder term can be detonated to an exceptionally high worth if anomalies are available in the event of RMSE, though in the event of RMLSE the exceptions are radically downsized in this manner so that effect is nullified We have part the preparation information into subsets of train and test information. We endeavor to assemble straight relapse model utilizing OLS in python utilizing fundamental bundles in particular sk-learn and scikit bundles after information preprocessing and train-test split of informational index. We realize that multi-collinearity is the impact of the various relapse models having related indicators. In straightforward, when informational collection has an enormous number of indicators, it is conceivable that couple of these indicators might be profoundly connected. Presence of high relationship between free factors is called multi-collinearity. Presence of multi-collinearity can destabilize the model. Aside from connection framework, to check such relations between factors we use, variety expansion factor (VIF). It estimates the extent of multi-collinearity. It is characterized as follows:

$$\text{VIF} = 1/(1-R2)$$

There are many approaches to handle multicollinearity. One simple technique is to dropping the variables from the model building that is highly correlated with others with the help of statistic and Variation Inflation factor. The threshold value for VIF is 5. VIF greater than 5 requires further investigation to assess the impact of multi-collinearity. Taking into account all the above metrics, we attempt to build basic regression model.

R-Squared incentive for model created is 0.418 and changed R-squared worth is

0.418. The p esteems and VIF regarding highlights were in allowable reach. We have negative incentive for test set which needs further examination. A prescient model must be straightforward as basic conceivable, however no less complex. Regularization is a strategy used to build an ideally unpredictable model, in other words a model that is just about as basic as conceivable when performing great on preparing information. Through this interaction, we can find some kind of harmony between keeping model less difficult, yet not making it too guileless to be of any utilization. The relapse just attempts to limit the blunder and it doesn't represent model intricacy. A portion of the successful regularization methods bring down the intricacy of the model and forestall overfitting. Lately, numerous specialists have utilized these high-level models to deal with multi-collinearity. We at that point arranged a progression of fits utilizing two regularized direct relapse models-Ridge and Lasso relapse models. the model and forestall overfitting. As of late, numerous specialists have utilized these high-level models to deal with multi-collinearity. We at that point arranged a progression of fits utilizing two regularized direct relapse models-Ridge and Lasso relapse models

3.5.2 Linear regression

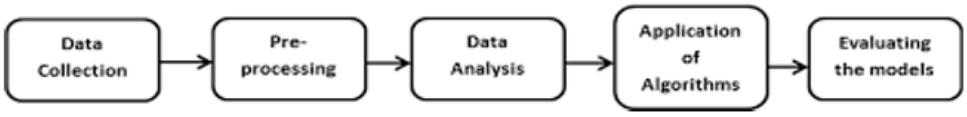
Linear Regression is an AI calculation dependent on directed learning. It plays out a relapse task. Relapse models an objective forecast esteem dependent on free factors. It is for the most part utilized for discovering the connection among factors and gauging. Diverse relapse models contrast dependent on – the sort of connection among needy and free factors, they are thinking about and the quantity of autonomous factors being utilized.

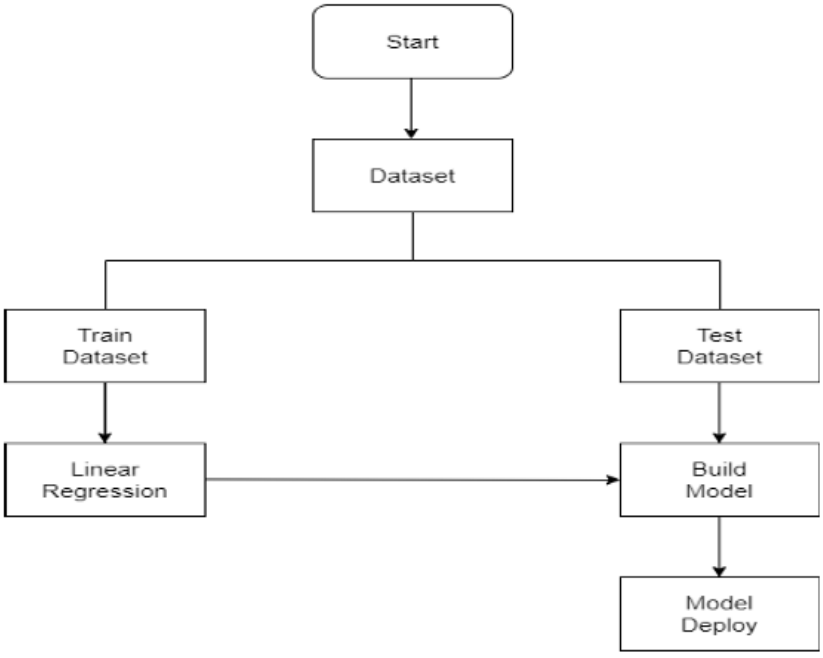
This article will exhibit how to utilize the different Python libraries to execute direct relapse on a given dataset. We will exhibit a twofold direct model as this will be simpler to picture. There are two kinds of directed AI calculations: Regression and order. The previous predicts constant worth yields while the last predicts discrete yields. For example, foreseeing the cost of a house in dollars is a relapse issue while anticipating whether a tumor is threatening or generous is an order issue.we will momentarily examine what direct relapse is and how it tends to be executed utilizing the Python Scikit-Learn library, which is quite possibly the most well-known AI

libraries for Python. The expression "linearity" in polynomial math alludes to a straight connection between at least two factors. On the off chance that we attract this relationship a two-dimensional space (between two factors, forthis situation), we get a straight line.

We should consider a situation where we need to decide the straight connection between the quantities of hours an understudy examines and the level of imprints that understudy scores in a test. We need to discover that given the quantity of hours an understudy gets ready for a test, about how high of a score can the understudy accomplish? On the off chance that we plot the free factor (hours) on the x-pivot and ward variable (percentage) on the y-axis, linear regression gives us a straight line that best fitsthe data points, as shown in the figure below.

3.6 Block Diagram





Chapter 4

PROPOSED WORK & OUTPUT

4.1 Data Preparation

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	price
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2

4.2 Modeling

The process of modeling means training a machine-learning algorithm to predict the labels from the features, tuning it for the business needs, and validating it on holdout data. When you train an algorithm with data it will become a model. One important aspect of all machine learning models is to determine their accuracy. Now to determine their accuracy, one can train the model using the given dataset and then predict the response values for the same dataset using that model and hence, find the accuracy of the model.

In this project, we use Scikit-Learn to rapidly implement a few models such as Linear Regression, Decision Tree, Random Forest, and Gradient Boosting.

4.2.1. Linear Regression

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous in the range such as salary, age, price, etc. It is a statistical approach that models the relationship between input features and output. The input features are called the independent variables, and the output is called a dependent variable. Our goal here is to predict the value of the output based on the input features

by multiplying it with its optimal coefficients. The name linear regression was come due to its graphical representation.

There are two types of Linear Regression: -

- **Simple Linear Regression**- In a simple linear regression algorithm the model shows the linear relationship between a dependent and a single independent variable. In this, the dependent variable must be a continuous value while the independent variable can be any continuous or categorical value.
- **Multiple Linear Regression**- In a multiple linear regression algorithm the model shows the linear relationship between a single dependent and more than one independent variable.

4.2.2. Random Forest

Random forest is a supervised learning algorithm which can be used for both classification and regression problem. It is a collection of Decision Trees. In general, Random Forest can be fast to train, but quite slow to create predictions once they are trained. This is due because it has to run predictions on each tree and then average their predictions to create the final prediction. A more accurate prediction requires more trees, which results in a slower model. In most real-world applications the random forest algorithm is fast enough, but there can certainly be situations where run-time performance is important and other approaches would be preferred. A random forest is a meta-estimator (i.e. it combines the result of multiple predictions) which aggregates many decision trees, with some helpful modifications. Random forest first splits the dataset into n number of samples and then apply decision tree on each sample individually. After that, the final result is that predicted accuracy whose majority is higher among all.

Random Forest depends on the concept of ensemble learning. An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. A model comprised of many models is called an Ensemble model.

Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between those trees while building random forest model.

4.2.3. Gradient Boosting

```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
price      0
dtype: int64
```

Gradient boosting is a technique which can be used for both classification and regression problem. This model combines the predictions from multiple decision trees to generate the final predictions. Also, each node in every other decision tree takes a different subset of features for selecting the best split. But there is a slight difference in gradient boosting in comparison to random forest that is gradient boosting builds one tree at a time and combines the results along the way. Also, it gives better performance than random forest. The idea of gradient boosting originated in the observation by Leo Bierman that boosting can be interpreted as an optimization algorithm on a suitable cost function. Gradient Boosting trains many models in a gradual, additive, and sequential manner.

The modeling is done in the following steps: -

- First, we split the dataset into a training set and a testing set.
- Then we train the model on the training set.
- And at last, we test the model on the testing set and evaluate how well our model performs.

So, after applying these models we get the following accuracy:

Table 4.1: Model Accuracy Table

S.No.	Models	Accuracy
1	Linear Regression	0.747545073
2	Random Forest	0.962269474
3	Gradient Boosting Regressor	0.963187213

4.3 Testing

In Machine Learning the main task is to model the data and predict the output using various algorithms. But since there are so many algorithms, it was really difficult to choose the one for predicting the final data. So, we need to compare our models and choose the one with the highest accuracy.

Machine learning applications are not 100% accurate, and approx. never will be. There are some of the reasons why testers cannot ignore learning about machine learning. The fundamental reason is that these applications learning limited by data they have used to build algorithms. For example, if 99% of emails aren't spammed, then classifying all emails as not spam gets 99% accuracy through chance. Therefore, you need to check your model for algorithmic correctness. Hence testing is required. Testing is a subset or part of the training dataset that is built to test all the possible combinations and also estimates how well the model trains. Based on the test data set results, the model was fine-tuned.

4.3.1 Mean Absolute Error (MAE)

It is the mean of all absolute error. MAE (ranges from 0 to infinity, lower is better) is much like RMSE, but instead of squaring the difference of the residuals and taking the square root of the result, it just averages the absolute difference of the residuals. This produces positive numbers only and is less reactive to large errors. MAE takes the average of the error from every sample in a dataset and gives the output.

Hence, MAE = True values – Predicted values

4.3.2 Mean Squared Error (MSE)

It is the mean of square of all errors. It is the sum, overall, the data points, of the square of the difference between the predicted and actual target variables, divided by the number of data points. MSE is calculated by taking the average of the square of the difference between the original and predicted values of the data.

4.3.3 Root Mean Squared Error (RMSE)

RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model. RMSE (ranges from 0 to infinity, lower is better), also called Root Mean Square Deviation (RMSD), is a quadratic-based rule to measure the absolute average magnitude of the error.

In our project, we perform testing on two models: Linear Regression and Random Forest.

Linear Regression Model Testing:

Fig. 4.1 Scatter Plot for Linear Regression

We draw a scatter plot between predicted and tested values and then find errors like MSE, MAE, and RMSE. After that, we also draw a distribution plot of the difference between actual and predicted values using the seaborn library. A distplot or distribution plot represents the overall distribution of continuous data variables.

Table 4.2: Error table for Linear Regression

Serial No.	Models	Accuracy
1	Mean Absolute Error	1.992295685 93
2	Mean Squared Error	20.0334370
3	Root Mean Absolute Error	0.91159376

Table 4.3 Error table

S.no	Models	Accuracy
1	Mean Squared Error	1.99229568593
2	Root Mean Absolute Error	0.91159376

4.4 Price Prediction Function

After finding the errors for both linear regression and random forest algorithm, we build a function name “predict price” whose purpose is to predict the price by taking 4 parameters as input. These four parameters are cab name, source, surge multiplier, and icon (weather). As the dataset train on the continuous values and not on categorical values, these values are also passed in the same manner i.e. in integer type. We create a manual for users which gives instructions about the input like what do you need to type for a specific thing and in which sequence.

We use random forest model in our function to predict the price. First, we search for all the desired rows which have the input cab name and extract their row number. After then we create an array x which is of the length of the new dataset and it’s initially all values are zero. After creating the blank array, we assign the input values of source, surge multiplier, and icon to the respected indices. Following it we check the count of all desired rows if it was greater than zero or not. If the condition gets true, we assign the value 1 to the index of x array and return the price using the predict function with trained random forest algorithm.

It somehow works like a hypothesis space because it gives an output for any input from input the space.

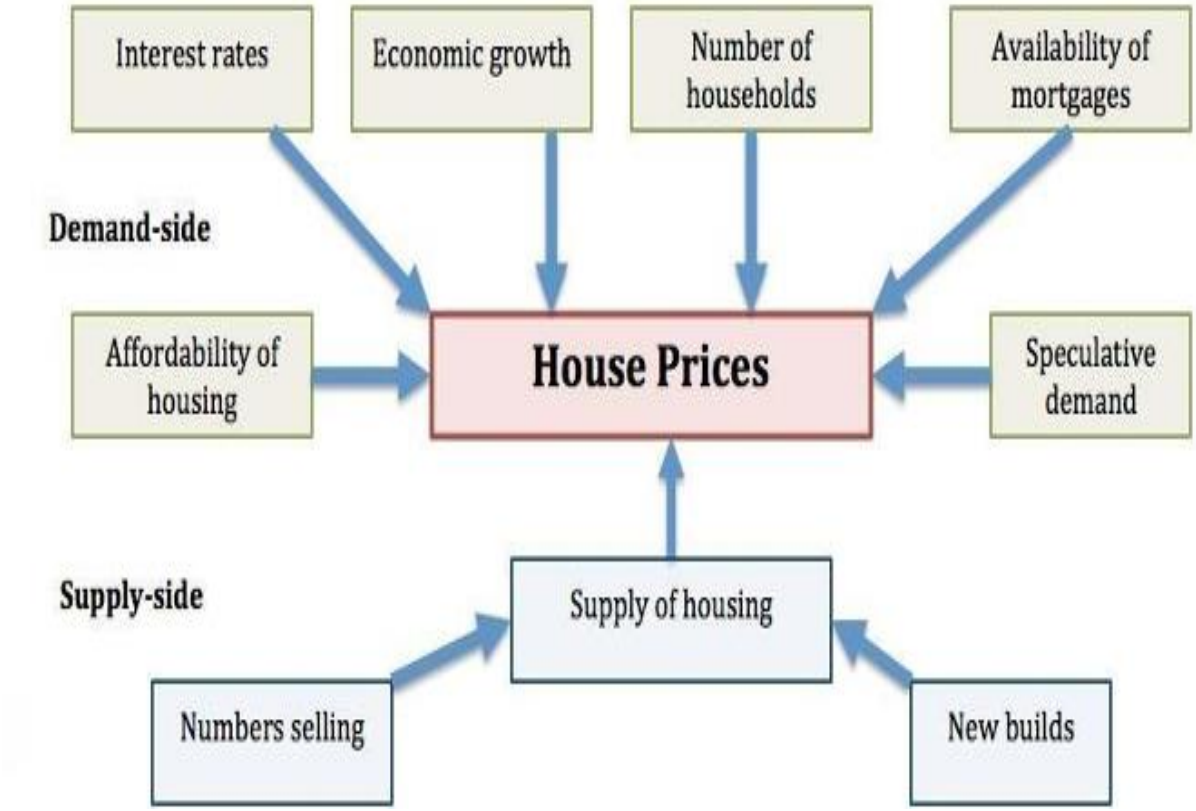


Fig. 4.2 Price Prediction Function

CHAPTER-5

SOFTWARE DEVELOPMENT METHOLOGY

5.1 Requirement Gathering

- **Goal**: The goal of this project is to predict house prices based on various features, such as geographic location, number of rooms, and median income of the area. The project will use machine learning algorithms to predict house prices based on the input data.
- **Input Features**: These could include location coordinates (latitude, longitude), number of rooms, population, household size, and median income.
- **Target Output**: The target variable is the **house price**, typically represented as **median house value**.
- ### 5.2 Feasibility Study
- **Technical Feasibility**:
    - Tools: The project will utilize tools and libraries such as **Python**, **scikit-learn**, **Pandas**, and **Matplotlib**.
    - Machine Learning Models: Algorithms such as **Linear Regression**, **Random Forest Regressor**, and **XGBoost** will be tested for predictive accuracy.
- **Economic Feasibility**:
    - Open-source tools such as **Python** and **scikit-learn** make the project cost-effective. If cloud resources are used, they are charged based on the computational power required.
    - **Ongoing Costs**: Cloud costs and software maintenance for retraining the models periodically.
- **Legal and Ethical Feasibility**: Ensure the dataset is compliant with data privacy regulations such as **GDPR** if it contains any personal information.
- ### 5.3 Design
- **Data Preprocessing**: This step includes data cleaning, handling missing values, encoding categorical data (e.g., ocean proximity), and scaling numerical features (e.g., total rooms, median income).

- **Feature Engineering**: Identifying the most important features (e.g., median income, total rooms, and longitude) for house price prediction.

- **Model Selection**: Select appropriate machine learning models like **Random Forest**, **Linear Regression**, or **Boost**.
- **Evaluation Metrics**: Evaluate the model using metrics like **Mean Absolute Error (MAE)**, **Root Mean Squared Error (RMSE)**, and **R-squared**.

5.4 Implementation

- **Data Collection**: Use publicly available datasets such as the **California housing dataset** (available on Kaggle or UCI repository).
- **Data Preprocessing**:
    - Clean the data by handling missing values .
    - Encode categorical features .
    - Normalize/standardize the numerical features .
- **Model Building**:
    - Implement and train models using **scikit-learn**.
    - Train multiple models and evaluate their performance using cross-validation.
- **Model Evaluation**: Use evaluation metrics to determine the best-performing model. Compare **linear regression**, **random forest**, and other models to assess predictive accuracy.

5.5 Testing

- **Unit Testing**: Each module (data preprocessing, model training, etc.) will be unit-tested to ensure they work as expected.
- **Integration Testing**: Ensure the entire system works as a whole, with the proper integration of data preprocessing, model training, and prediction.
- **Performance Testing**: Measure the model's performance, such as the time taken for training and prediction.

5.6 Deployment

- **Deployment Strategy**: The model can be deployed as a web application using frameworks like **Flask** or **streamlit** in Python and , where users can input their parameters (e.g., number of rooms, median income) and get a predicted house price.
- **Model Deployment**: The trained model can be exported and deployed to a cloud service or web server, where it can make predictions on new data.

5.7 Maintenance

- **Continuous Monitoring**: Monitor the model's performance over time. If there's any data drift or significant performance drop, retrain the model with updated data.
- **Model Retraining**: Periodically update the model with newer data to maintain accuracy and relevance. This could be automated using cloud services that allow for scheduled retraining.

5.8 Feedback and Iteration

- **User Feedback**: After deployment, collect feedback from users to improve the system. If necessary, additional features can be added to improve prediction accuracy.
- **Model Improvement**: Based on feedback and additional data, refine the model, tweak hyperparameters, or try different algorithms.

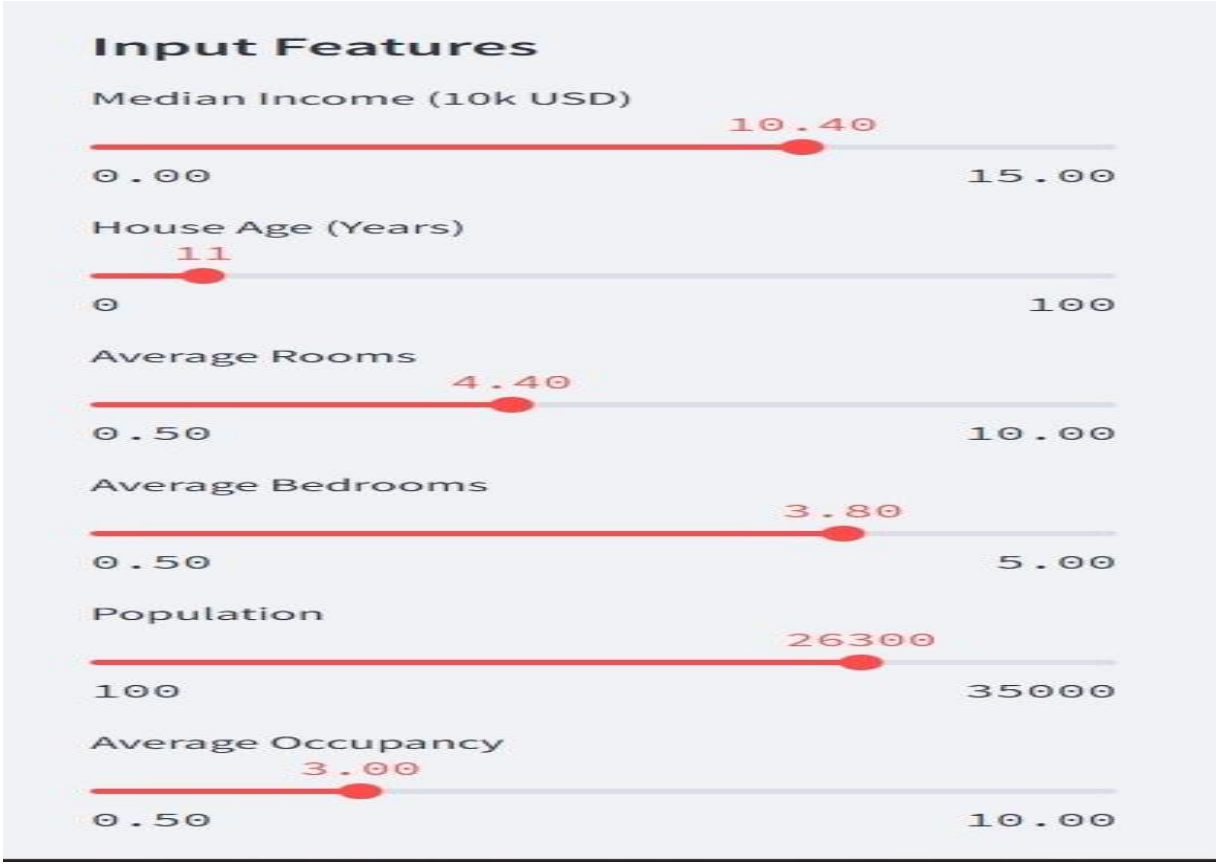
5.9 Software Development Phases Breakdown

- **Initiation Phase**:
    - Define the project goals, requirements, and constraints.
    - Gather the necessary datasets and define the scope of data features.
- **Planning Phase**:

- Develop a detailed plan, including the timeline for data preprocessing, model training, and deployment.
- Choose the right machine learning algorithms and evaluation metrics.

- **Execution Phase**:
  - Carry out the data cleaning and preparation tasks.
  - Implement machine learning models.
  - Conduct initial testing and validation.
- **Closure Phase**:
  - Deploy the model to provide house price predictions.
  - Monitor the model's performance and retrain when necessary.

SCREENSHOTS



California House Price Prediction

This app predicts the house price based on key features.

User Input Features

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	10.4	11	4.4	3.8	26,300	3	37	-120

Predicted House Price

$4.40

CHAPTER-6

CONCLUSIONS

Thus, the machine learning model to predict the house price based on given dataset is executed successfully using Linear regressor (an upgraded/ slighted boosted form of regular linear regression, this gives lesser error). This model further helps people understand whether this place is more suited for them based on heatmap correlation. It also helps people looking to sell a house at best time for greater profit. Any house price in any location can be predicted with minimum error by giving appropriate dataset.

CHAPTER-7

REFERENCES

Real Estate Price Prediction with Regression and Classification, CS 229 Autumn 2016 Project Final Report

• Gongzhu Hu, Jinping Wang, and Wenying Feng Multivariate Regression Modelling for Home Value Estimates with Evaluation using Maximum Information Coefficient

• Byeonghwa Park, Jae Kwon Bae (2015). Using machine learning algorithms for housing price prediction, Volume 42, Pages 2928-2934 [4] Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining, 2015. Introduction to Linear Regression Analysis.

• Iain Pardoe, 2008, Modelling Home Prices Using Realtor Data • Aaron Ng, 2015, Machine Learning for a London Housing Price Prediction Mobile Application

• Wang, X., Wen, J., Zhang, Y. Wang, Y. (2014). Real estate price forecasting based on SVM optimized by PSO. Optik-International Journal for Light and Electron Optics, 125(3), 14391443

STUDENTS PROFILE

Name: Akhil Pandey
Enrollment no.: 21B032
Email: 211b032@juetguna.in
Address: Lucknow, Uttar Pradesh
Contact: 9305828258

Name: Bicky Kumar
Enrollment no.: 21B094
Email: 211b094@juetguna.in
Address: Aurangabad, Bihar
Contact: 9801841167

Name: Priyadarshi Kumar
Enrollment no.: 21B227
Email: 211b227@juetguna.in
Address: Aurangabad, Bihar
Contact: 9650387593