Predecting House Price Using Machine Learning

Kingston Engineering College code:5113

K.Santhosh Kumar(511321106021)(Team Head)

V.Dilli Babu(511321106009)

V.Dilli Babu(511321106008)

K.Dinesh(511321106011)

1. **Deep Learning**: Deep learning techniques, particularly neural networks, have been applied to house price prediction. Convolutional Neural Networks (CNNs) can be used to process images of properties, while Recurrent Neural Networks (RNNs) can handle time series data related to real estate market trends.
2. **Feature Engineering**: Advanced feature engineering can extract more meaningful information from raw data. For example, you can use natural language processing (NLP) to analyze property descriptions and sentiment analysis to understand the emotions associated with property listings.
3. **Ensemble Learning**: Techniques like Random Forests and Gradient Boosting can be used in ensemble models to combine the predictions of multiple algorithms, often leading to more accurate results.
4. **Geospatial Analysis**: Location is a critical factor in house pricing. Geospatial analysis and geographic information system (GIS) data can be integrated into the model to capture the impact of location on property values.
5. **Time Series Analysis**: Incorporating time series analysis can help capture seasonality and trends in the real estate market, allowing for more accurate predictions.
6. **Image Analysis**: If you have access to images of properties, you can use computer vision techniques to extract features from these images. This could include identifying architectural styles, amenities, or even the quality of materials used in construction.
7. **Natural Language Processing (NLP)**: Analyzing property descriptions, customer reviews, or real estate market reports using NLP can provide valuable insights that affect property prices.
8. **Transfer Learning**: Pre-trained models like BERT or GPT-3 can be fine-tuned for NLP tasks related to real estate, such as sentiment analysis or property description generation.
9. **Fairness and Bias Mitigation**: Addressing fairness and bias concerns in housing predictions is an essential innovation. Algorithms should be designed and tested to ensure they do not discriminate against protected groups or perpetuate existing biases in the housing market.
10. **Explainable AI (XAI)**: Developing models that provide transparent explanations for their predictions is crucial, especially in real estate, where trust and accountability are vital. Techniques like LIME (Local Interpretable Model-agnostic Explanations) can be used to explain complex model predictions.
11. **Blockchain and Smart Contracts**: Innovations in blockchain technology can be used to create transparent and tamper-proof property records and smart contracts for real estate transactions.
12. **Real-time Data Integration**: Leveraging real-time data sources, such as social media trends, news sentiment, and economic indicators, can enhance the accuracy of predictions by capturing the most up-to-date market conditions.
13. **Hybrid Models**: Combining machine learning with domain-specific knowledge and expert systems can lead to hybrid models that are more

accurate and explainable.

SOFTWARE OPTIONS

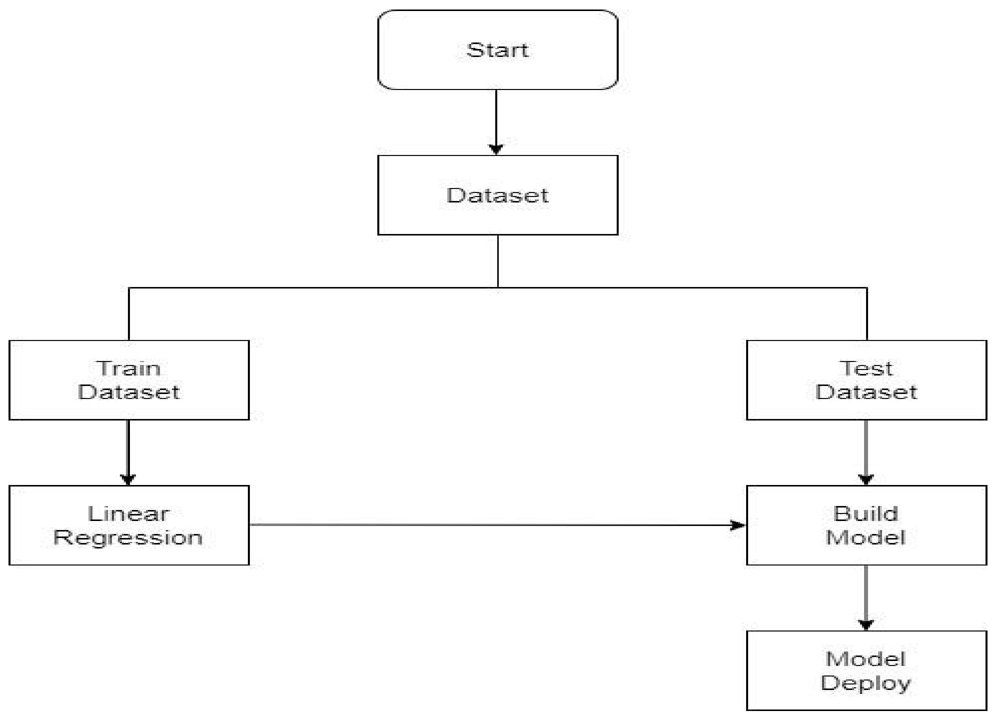
1. **KNIME**: KNIME is an open-source data analytics, reporting, and integration n. It provides a graphical user interface for designing data pipelines and building machine learning models, including regression models for house price prediction.
2. **IBM Watson Studio**: IBM Watson Studio is a cloud-based data science platform that offers tools for data preparation, model building, and deployment. It supports Python and R, making it suitable for building machine learning models for various tasks, including predicting house prices.
3. **DataRobot**: DataRobot is an automated machine learning platform that automates many of the tasks involved in building predictive models. It can be used for regression tasks like predicting house prices and offers a userfriendly interface.
4. **Microsoft Azure Machine Learning**: Azure Machine Learning is a cloudbased service by Microsoft that simplifies the process of building, training, and deploying machine learning models. It supports various regression algorithms for house price prediction.
5. **Google AutoML**: Google's AutoML platform offers automated machine learning capabilities. While it may not have as extensive a user interface as some of the other tools, it provides a user-friendly way to build regression models, and it's well-integrated with Google Cloud services.

Source code :

```
# Import necessary libraries import
pandas as pd import numpy as np
from sklearn.model_selection import train_test_split from
sklearn.linear_model import LinearRegression from sklearn.metrics
import mean_squared_error, r2_score

# Load your dataset (replace 'data.csv' with your dataset file) data =
pd.read_csv('data.csv')

# Define features (X) and target variable (y)
X = data[['feature1', 'feature2', 'feature3']]  # Add your relevant features here
y = data['price']  # Replace 'price' with the actual target column name
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Linear Regression model
model = LinearRegression()

# Train the model on the training data model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred) rmse =
np.sqrt(mse) r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}') print(f'Root Mean
Squared Error: {rmse}')
print(f'R-squared: {r2}')

# Now you can use the trained model to predict house prices for new data.
# For example:
# new_data = pd.DataFrame({'feature1': [value1], 'feature2': [value2],
'feature3': [value3]})
# predicted_price = model.predict(new_data)
```


BLOCK DIAGRAM


Start
Dataset
Train Dataset
Test Dataset
Linear Regression
Build Model
Model Deploy

PROPOSED SYSTEM PHASES

Phase 1: Collection of data

Data processing techniques and processes are numerous. We collected data for USA/Mumbai real estate properties from various real estate websites. The data would be having attributes such as Location, carpet area, built-up area, age of the property, zip code, price, no of bedrooms etc. We must collect the quantitative data which is structured and categorized. Data collection is needed before any kind of machine learning research is carried out. Dataset validity is a must otherwise there is no point in analyzing the data.

Phase 2: Data preprocessing

Data preprocessing is the process of cleaning our data set. There might be missing values or outliers in the dataset. These can be handled by data cleaning. If there are many missing values in a variable we will drop those values or substitute it with the average value.

Phase 3: Training the model

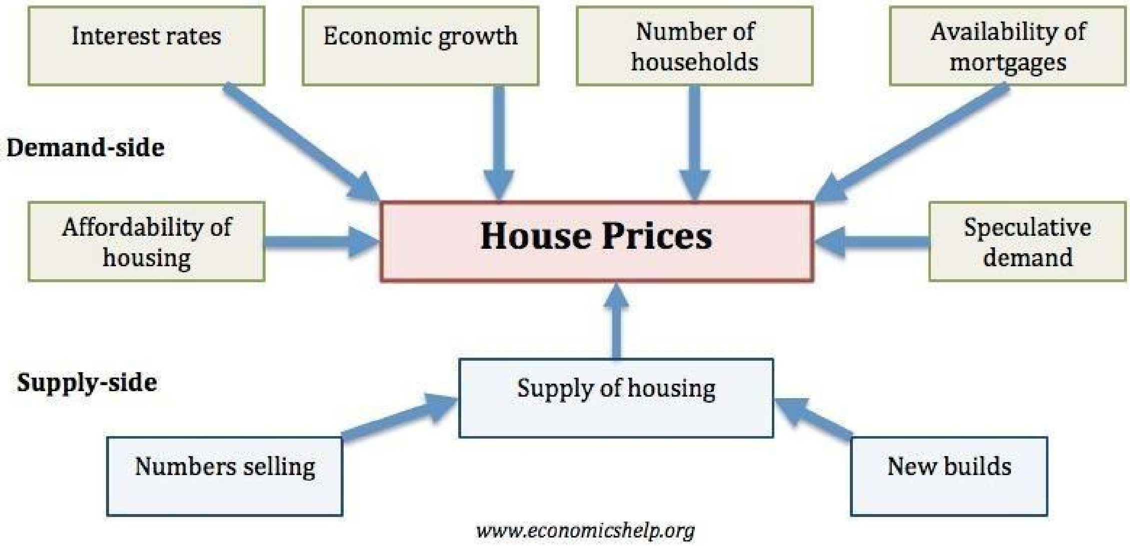
Since the data is broken down into two modules: a Training set and Test set, we must initially train the model. The training set includes the target variable. The decision tree regressor algorithm is applied to the training data set. The Decision tree builds a regression model in the form of a tree structure.

Phase 4: Testing and Integrating with UI

The trained model is applied to test dataset and house prices are predicted. The trained model is then integrated with the front end using Flask in python

FACTORS THAT AFFECT HOUSE PRICING

In order to predict house prices, first we have to understand the factors that affect house pricing.



- **Economic growth.** Demand for housing is dependent upon income. With higher economic growth and rising incomes, people will be able to spend more on houses; this will increase demand and push up prices. In fact, demand for housing is often noted to be income elastic (luxury good); rising incomes leading to a bigger % of income being spent on houses. Similarly, in a recession, falling incomes will mean people can't afford to buy and those who lose their job may fall behind on their mortgage payments and end up with their home repossessed.

- **Unemployment.** Related to economic growth is unemployment. When unemployment is rising, fewer people will be able to afford a house. But, even the fear of unemployment may discourage people from entering the property market. • **Interest rates.** Interest rates affect the cost of monthly mortgage payments. A period of high- interest rates will increase cost of mortgage payments and will cause lower demand for buying a house. High-interest rates make renting relatively

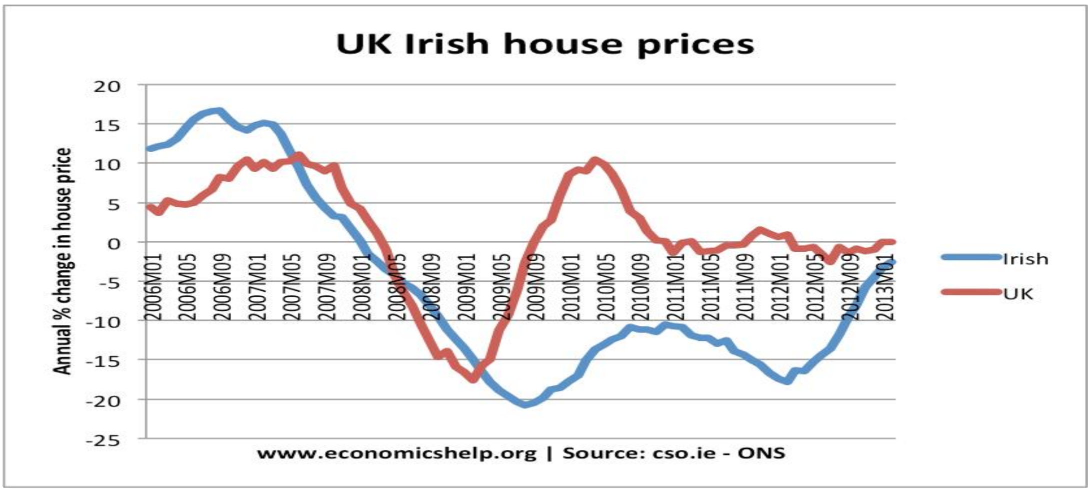
1

0 more attractive compared to buying. Interest rates have a bigger effect if homeowners have large variable mortgages. For example, in 1990-92, the sharp rise in interest rates caused a very steep fall in UK house prices because many homeowners couldn't afford the rise in interest rates.

- **Consumer confidence**. Confidence is important for determining whether people want to take the risk of taking out a mortgage. In particular expectations towards the housing market is important; if people fear house prices could fall, people will defer buying.

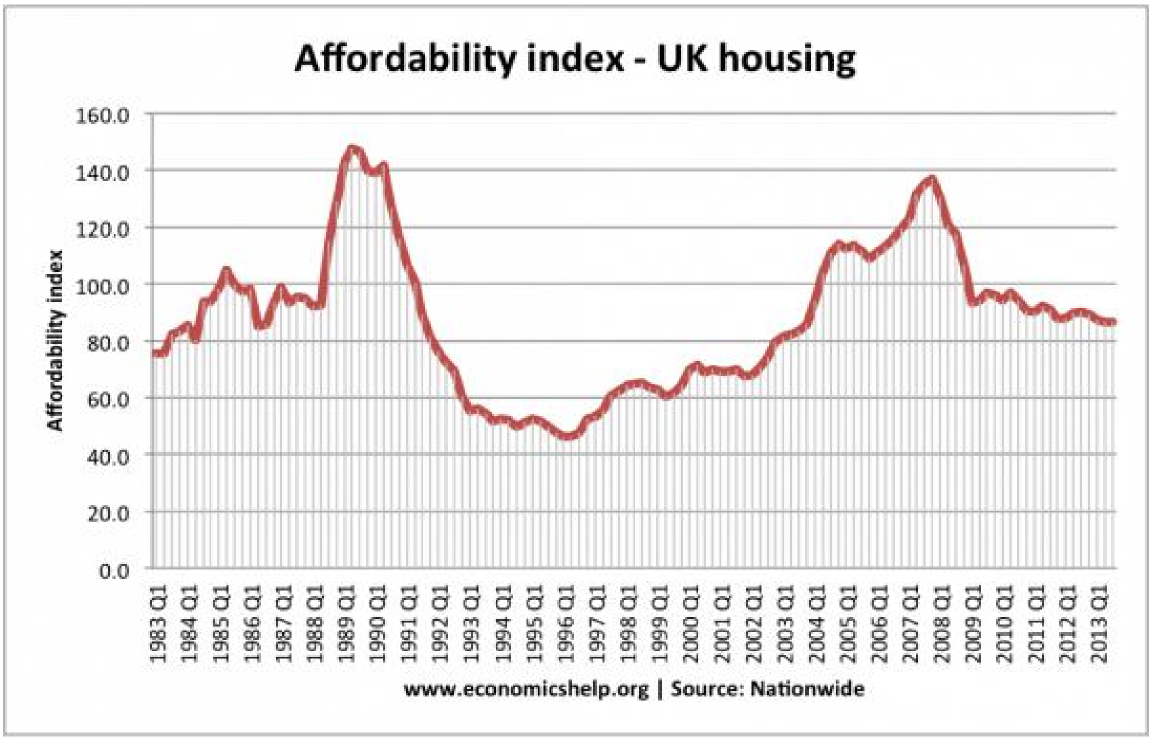
- **Mortgage availability**. In the boom years of 1996-2006, many banks were very keen to lend mortgages. They allowed people to borrow large income multiples (e.g. five times income). Also, banks required very low deposits (e.g. 100% mortgages). This ease of getting a mortgage meant that demand for housing increased as more people were now able to buy. However, since the credit crunch of 2007, banks and building societies struggled to raise funds for lending on the money markets. Therefore, they have tightened their lending criteria requiring a bigger deposit to buy a house. This has reduced the availability of mortgages and demand fell.

- **Supply**. A shortage of supply pushes up prices. Excess supply will cause prices to fall. For example, in the Irish property boom of 1996-2006, an estimated 700,000 new houses were built. When the property market collapsed, the market was left with a fundamental oversupply. Vacancy rates reached 15%, and with supply greater than demand, prices fell.



By contrast, in the UK, housing supply fell behind demand. With a shortage, UK house prices didn't fall as much as in Ireland and soon recovered – despite the ongoing credit crunch. The supply of housing depends on existing stock and new house builds. Supply of housing tends to be quite inelastic because to get planning permission and build houses is a time-consuming process. Periods of rising house prices may not cause an equivalent rise in supply, especially in countries like the UK, with limited land for home-building.

- **Affordability/house prices to earnings.** The ratio of house prices to earnings influences the demand. As house prices rise relative to income, you would expect fewer people to be able to afford. For example, in the 2007 boom, the ratio of house prices to income rose to 5. At this level, house prices were relatively expensive, and we saw a correction with house prices falling.



Another way of looking at the affordability of housing is to look at the percentage of take-home pay that is spent on mortgages. This takes into account both house prices, but mainly interest rates and the cost of monthly mortgage payments. In late 1989, we see housing become very unaffordable because of rising interest rates.

housing markets are highly geographical. For example, national house prices may be falling, but This caused a sharp fall in prices in 1990-92.

SAMPLE CODE

```
import pandas as pd
import numpy as np
import seaborn as sns
import
matplotlib.pyplot as plt

%matplotlib inline

HouseDF = pd.read_csv('USA_Housing.csv')

HouseDF.head()

HouseDF=HouseDF.reset_index()

HouseDF.head()

HouseDF.info()

HouseDF.describe()

HouseDF.columns

sns.pairplot(HouseDF)

sns.distplot(HouseDF['Price’])

sns.heatmap(HouseDF.corr(),

annot=True)

X = HouseDF[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area

Number of Bedrooms', 'Area Population']] y = HouseDF['Price’] from sklearn.model_selection import

train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)

from sklearn.linear_model import

minmaxscaler lm =
```

```
minmaxscaler(feature_range=(0,1))

lm.fit_transform(X_train,y_train)

print(lm.intercept_)

coeff_df =
pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient’])
coeff_df
```

```
from keras.layers import Dense,Dropout,LSTM  from keras.models
import Sequential model = Sequential() model.add(LSTM(units =
50,activation = 'relu',return_sequences = True,input_shape =
(x_train.shape[1], 1)))
model.add(Dropout(0.2))

model.add(LSTM(units = 60,activation = 'relu',return_sequences = True)) model.add(Dropout(0.3))

model.add(LSTM(units = 80,activation = 'relu',return_sequences = True))
model.add(Dropout(0.4))

model.add(LSTM(units = 120,activation = 'relu'))
model.add(Dropout(0.5))

model.add(Dense(units = 1))

model.compile(optimizer='adam', loss = 'mean_squared_error’)
model.fit(x_train, y_train,epochs=50)

print(lm.intercept_)

coeff_df =
pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient’])
coeff_df

predictions = lm.predict(X_test)

scale_factor = 1/0.02099517
y_predicted = y_predicted *
scale_factor y y_test = y_test *
scale_factor
```

```
plt.scatter(y_test,predictions)

sns.distplot((y_test-predictions),bins=50);
```

15

```
plt.figure(figsize=(12,6))

plt.plot(y_test,'b',label = 'Original

Price') plt.plot(y_predicted,'r',label =

'Predicted Price') plt.xlabel('Time')

plt.ylabel('Price') plt.legend()

plt.show()

from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test,
predictions)) print('MSE:',
metrics.mean_squared_error(y_test, predictions))
print('RMSE:',
np.sqrt(metrics.mean_squared_error(y_test,
predictions)))
```

EXPLANATION THE DATASET

	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

First we import a sample data from sklearn library , you can get different types of sample data from Kaggle. The data taken here is the data of various parameters and the house prices in a given city called boston in the year between 1970 to 2020.

Here the data parameters are explained as follows:

```
1. CRIM      per capita crime rate by town
2. ZN        proportion of residential land zoned for lots over
             25,000 sq.ft.
3. INDUS     proportion of non-retail business acres per town
4. CHAS      Charles River dummy variable (= 1 if tract bounds
             river; 0 otherwise)
5. NOX       nitric oxides concentration (parts per 10 million)
6. RM        average number of rooms per dwelling
7. AGE       proportion of owner-occupied units built prior to 1940
8. DIS       weighted distances to five Boston employment centres
9. RAD       index of accessibility to radial highways
10. TAX      full-value property-tax rate per $10,000
11. PTRATIO  pupil-teacher ratio by town
12. B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks
             by town
13. LSTAT    % lower status of the population
14. MEDV     Median value of owner-occupied homes in $1000's
```

Here for understanding purpose we have taken first 5 index/instance of data and printed them. In total there are 506 rows of data from the dataset , of which we have printed first 5 rows using head() function. There are 14 columns in total, i.e, 13 colums containing data of the place, and the 14^{th} column is the target column which contains the house prices.

18

Then we check if our data has some null values i.e missing values. Since if the data is incomplete , then there will be error during processing state which may lead to loss of accuracy in predicting model. Here in our given data , there is no missing value as we can see.

```
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
```

Since our data contains no missing value, the program will skip the dropping phase in data processing, where data is dropped to increase accuracy and fit missing values in a way so that it is suitable for modelling.

Next we try to describe the data in such a way so that both people and machine find it easy to understand the given data . In order to do thiswe use the describe() function.

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	price
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032	12.653063	22.532806
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	7.141062	9.197104
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000	5.000000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	6.950000	17.025000
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	11.360000	21.200000
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	16.955000	25.000000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	37.970000	50.000000

Counts refers to the number of instances of data in each column i.e 506 since there are 506 rows of data for each column Mean refers to mean value of data in given colum.

Std means the standard value i.e the most common value in given set of data for a particular column.

Min refers the least data value in each column.

Max refers to the maximum data value in each column.

25% refers that 25 percentile of the data in that column is equal to or below that value.

Next we try to understand the correlation between the different values, in order to do that, the best way is by using heat map. Heat map is a representation of data in the form of a map or diagram in which data values are represented as colours.

Correlation is a statistical measure that expresses the extent to which two variables are linearly related (meaning they change together at a constant rate)

There are two types of correlation, they are:

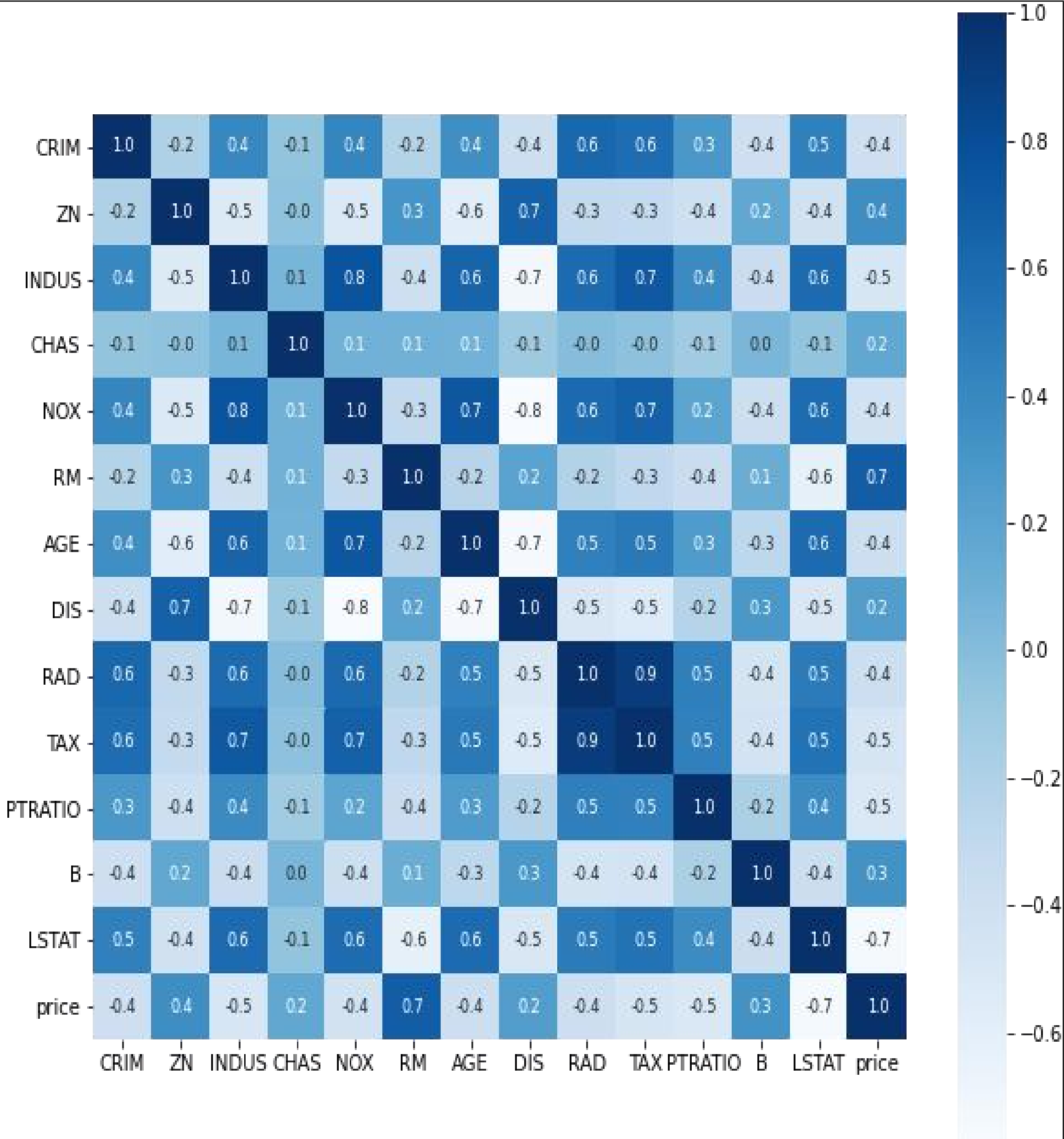
1. Positive correlation: A positive correlation is a relationship between two variables that move in tandem—that is, in the same direction. A positive correlation exists when one variable decreases as the other variable decreases, or one variable increases while the other increases.

2. Negative correlation: Negative correlation is a relationship between two variables in which one variable increases as the other decreases, and vice versa.

In statistics, a perfect negative correlation is represented by the value -1.0, while a 0 indicates no correlation, and +1.0 indicates a perfect positive correlation. A perfect negative correlation means the relationship that exists between two variables is exactly opposite all of the time. These are two types of correlation are represented numerically and as well as by shade of colour in the heat map.

20

HEATMAP – for better understanding of which place is best suited for individual personal preference based on given dataset. This uses correlation concept



Next we split our data into variables x and y , in order to train our model to predict data.

```
        CRIM    ZN   INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  \
0    0.00632  18.0    2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0
1    0.02731   0.0    7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0
2    0.02729   0.0    7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0
3    0.03237   0.0    2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0
4    0.06905   0.0    2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0
..       ...   ...     ...   ...    ...    ...   ...     ...  ...    ...
501  0.06263   0.0   11.93   0.0  0.573  6.593  69.1  2.4786  1.0  273.0
502  0.04527   0.0   11.93   0.0  0.573  6.120  76.7  2.2875  1.0  273.0
503  0.06076   0.0   11.93   0.0  0.573  6.976  91.0  2.1675  1.0  273.0
504  0.10959   0.0   11.93   0.0  0.573  6.794  89.3  2.3889  1.0  273.0
505  0.04741   0.0   11.93   0.0  0.573  6.030  80.8  2.5050  1.0  273.0

     PTRATIO       B  LSTAT
0       15.3  396.90   4.98
1       17.8  396.90   9.14
2       17.8  392.83   4.03
3       18.7  394.63   2.94
4       18.7  396.90   5.33
..       ...     ...    ...
501     21.0  391.99   9.67
502     21.0  396.90   9.08
503     21.0  396.90   5.64
504     21.0  393.45   6.48
505     21.0  396.90   7.88

[506 rows x 13 columns]
0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
       ...
501    22.4
502    20.6
503    23.9
504    22.0
505    11.9
Name: price, Length: 506, dtype: float64
```


Here the varible x contains the value of the first 13 columns i.e the parameters that are required for calculating and predicting the house prices. The varible y contains the 14th column values which are the house prices.

First we predict the values in y using the values in x . Then we compare the actual prices and predicted prices by using scatter plot. Then we find the r square error and mean square error between them . If the errors is less enough then we proceed for testing of the model since the training phase is over. If the error is large , then we use optimizers like adam, and repeat drop and fitting process for a set number of epochs to reduce the error.

The r square error or mean square error for good accuracy of the model in predicting the data is indicated numerically also.

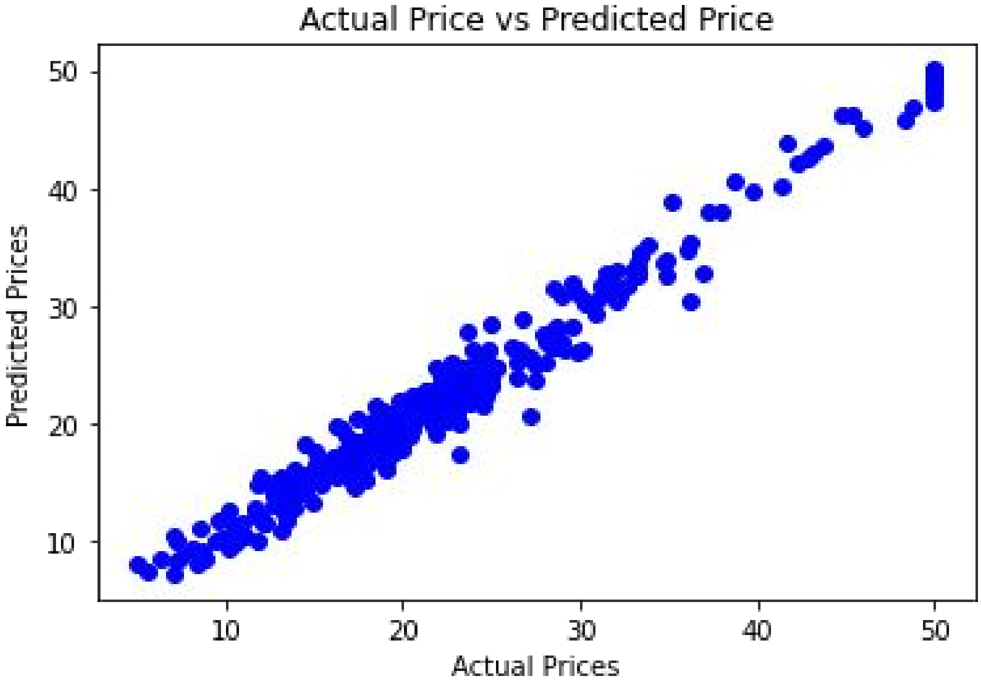
A model is good if these error values are less then 5.

Then during testing process we predict the future house prices using present and past data parameters of houses in an location. Then we plot this graphically as a house price over time graph.

For training the model , the error needs to be minimum for greater accuracy of model. The error between the actual and predicted price is plotted graphically using scatter plot. Here we can see that error is

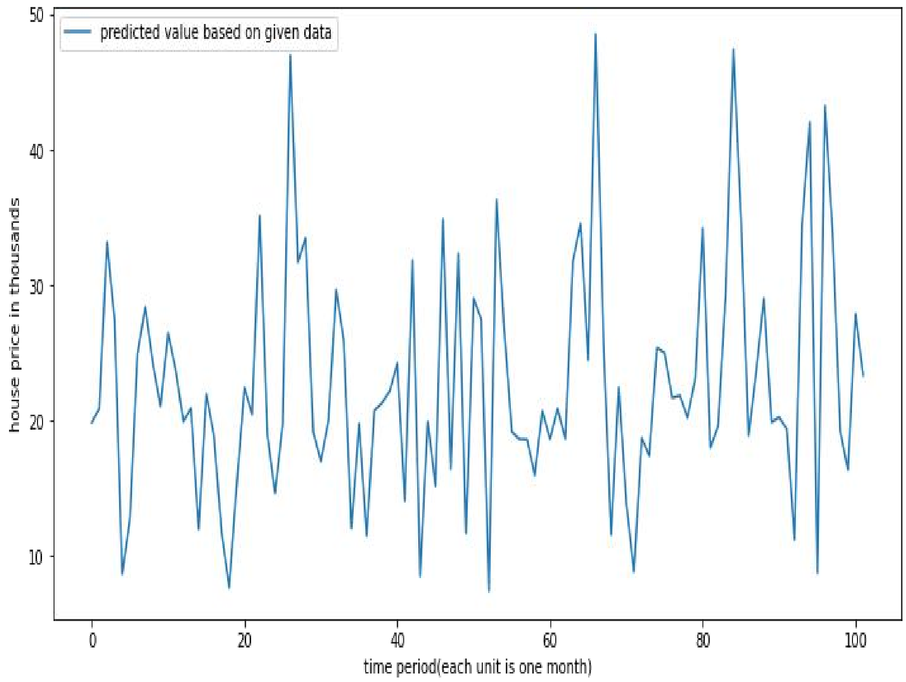
21

minimum since the data points of actual and predicted value are close to each other



22

PREDICTED VALUE OF HOUSE PRICE BASED ON TEST SAMPLE DATA



Advantages:

1. Informed Decision-Making: Buyers, sellers, and real estate professionals can make more informed decisions about purchasing or selling properties. Machine learning models can provide accurate estimates of property values based on historical data and current market conditions.

2. Pricing Accuracy: Machine learning models can provide more accurate and data-driven property valuations compared to traditional methods like comparative market analysis (CMA). This helps sellers avoid underpricing or overpricing their properties.

3. Reduced Subjectivity: Machine learning models are less susceptible to human bias and emotions, which can affect traditional pricing methods. They provide a more objective assessment of property values.

4. Faster Valuations: Automated house price prediction can provide quick valuations, which is especially useful for real estate professionals and individuals who need to make decisions promptly.

5. Market Trends: Machine learning can analyze large datasets to identify market trends and patterns, helping investors and real estate professionals make strategic decisions.

6. Risk Assessment: Machine learning can assist in evaluating the risk associated with buying or selling properties, such as identifying neighborhoods with potential for growth or decline.

7. Improved Investment Strategies: Investors can use machine learning to identify properties that are likely to appreciate in value over time, helping them develop more profitable investment strategies.

8. Customization: Machine learning models can be tailored to specific regions or property types, allowing for more accurate predictions in local real estate markets.

9. Cost Reduction: By automating the valuation process, real estate professionals can save time and resources, making their operations more efficient.

10. Accessibility: Property valuation through machine learning is accessible to a wider audience, not just real estate experts, enabling homeowners and potential buyers to estimate property values on their own.

11. Enhanced Marketing: Real estate agents can use accurate price predictions to create more effective marketing strategies for their listings.

12. Data-Driven Negotiations: Buyers and sellers can use machine learning predictions as a basis for negotiations, making the process more transparent and fair.

13. Improved Customer Experience: Buyers and sellers benefit from a smoother and more data-driven experience when they can access reliable pricing information.

14. Scalability: Machine learning models can handle a large volume of property data, making them suitable for analyzing real estate markets at scale.

Disadvantages:

1. Data Quality and Quantity: Machine learning models rely heavily on the quality and quantity of data. If the training data is incomplete, inaccurate, or biased, the model's predictions may be unreliable.

2. Data Availability: In some regions or for certain property types, there may be limited data available, making it challenging to train accurate machine learning models.

3. Market Volatility: Real estate markets can be subject to sudden and significant changes due to economic, political, or environmental factors. Machine learning models may struggle to adapt quickly to such fluctuations.

4. Overfitting: Machine learning models can overfit the training data, meaning they perform well on historical data but struggle to generalize to new data. This can result in inaccurate predictions for properties that differ significantly from the training dataset.

5. Model Complexity: Some machine learning models, especially deep learning models, can be complex and computationally intensive, making them difficult to implement and maintain for smaller businesses or individuals.

6. Interpretability: Many machine learning models, such as deep neural networks, are considered "black boxes," making it challenging to understand the factors influencing their predictions. This lack of interpretability can be a drawback, especially when trying to explain pricing decisions to clients or stakeholders.

7. Limited Feature Set: Machine learning models rely on the features (input variables) provided to them. If important features are missing from the dataset, the model may not capture all the factors influencing property prices.

8. Legal and Ethical Concerns: The use of machine learning for house price prediction can raise legal and ethical concerns, especially when it comes to fair housing and discrimination. Biased data or models can perpetuate inequalities and result in unfair pricing.

9. Model Maintenance: Machine learning models require ongoing maintenance and updates to stay relevant and accurate. The real estate market is dynamic, and the models need to adapt to changing conditions.

10. Human Expertise: While machine learning can provide valuable insights, it cannot replace the expertise of real estate professionals who have a deep understanding of local markets and can consider intangible factors that impact property values.

11. Security and Privacy: Handling large datasets of property and personal information poses security and privacy risks. Protecting sensitive data from breaches and misuse is crucial.

12. Costs: Developing, training, and maintaining machine learning models can be costly, and smaller real estate agencies or individuals may find it challenging to invest in these technologies.

13. Limited Data History: Historical data may not accurately capture the long-term trends or cyclical patterns in the real estate market, which can limit the model's ability to predict future prices.

CONCLUSION:

In conclusion, machine learning models for house price prediction offer significant advantages for both individuals and the real estate industry. They enable more informed decision-making, enhance pricing accuracy, reduce subjectivity, and provide faster valuations. These models can also identify market trends, assess risk, improve investment strategies, and reduce costs. Additionally, they make property valuation accessible to a broader audience and contribute to improved customer experiences.

Ultimately, machine learning models can be powerful assets for real estate decision-makers, but they are most effective when integrated into a broader decision-making framework that combines data-driven insights with domain knowledge and human judgment.