House Price Prediction

MINOR PROJECT REPORT

By

AMIR MUSTAQUE (RA2111047010054)
SIDDHARTH PRATYUSH (RA2111047010040)

Under the guidance of

DR. Tamilmani G

Assistant Professor

In partial fulfilment for the Course

OF

18AIE427T-DATA MINING AND ANALYTICS
IN CINTIL

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

JULY 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **18AIE427T-DATA MINING AND ANALYTICS** entitled in "**HOUSE PRICE PREDICTION**" is the bonafide work of **AMIR MUSTAQUE (RA211047010054)** and **SIDDHARTH PRATYUSH (RA2111047010040)** who carried out the work under my supervision.

SIGNATUR

DR TAMILMANI
Assistant professor
Department of computational Intelligence
SRM Institute of Science and Technology
Kattankulathur

SIGNATURE

DR. R Annie Uthra
Professor and Head,
Department of computational Intelligence
SRM Institute of Science and Technology
Kattankulathur

ABSTRACT

House price prediction is a critical application of machine learning that aids in estimating the market value of residential properties. This task is particularly significant for stakeholders such as real estate agents, buyers, sellers, and investors, enabling them to make informed decisions. The prediction process leverages a variety of data sources, including historical sales data, property characteristics, and economic indicators. Machine learning models, such as linear regression, decision trees, and neural networks, are employed to analyze these datasets and identify patterns that influence property values. Features like the number of bedrooms, square footage, location, and proximity to amenities are crucial in shaping the predictions. Advanced techniques like ensemble learning and feature engineering are also utilized to enhance the model's accuracy and robustness. The integration of geospatial data further enriches the prediction by considering neighborhood trends and regional economic conditions. This report delves into the methodologies used for house price prediction, highlighting the model selection, data preprocessing steps, and evaluation metrics. The findings underscore the potential of machine learning to provide reliable price estimates, thereby contributing to more efficient and transparent real estate markets.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Lakshmi.C , Professor, Department of Computational** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Dr. Tamimani** G, **Professor, Department of Computational Intelligence,** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to **our HOD DR. R. Annie Uthra ,Professor, and Head, Departmnet Of Computational Intelligence** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

1. INTRODUCTION

The prediction of housing prices has emerged as a pivotal aspect of real estate analytics, driven by the increasing availability of data and advancements in machine learning techniques. Accurate house price prediction models are essential for various stakeholders, including homebuyers, sellers, real estate agents, and investors, as they provide critical insights into market trends and property valuations. The process involves analyzing a myriad of factors that influence property prices, such as location, property size, age, condition, and economic indicators. Machine learning algorithms, ranging from simple linear regression to more sophisticated models like decision trees, random forests, and neural networks, are employed to uncover complex patterns and relationships within these factors. The goal is to develop models that can predict future house prices with high accuracy, thereby facilitating more informed decision-making in the real estate market. This introduction explores the significance of house price prediction, the types of data utilized, and the methodologies adopted, setting the stage for a comprehensive examination of predictive models and their applications in the real estate industry.

Problem statement

Predicting the prices of residential properties is a multifaceted problem that is crucial for stakeholders in the real estate industry, including homebuyers, sellers, real estate agents, and investors. Traditional property valuation methods, such as comparative market analysis and professional appraisals, are often time-consuming, costly, and prone to human bias. The advent of machine learning and the increasing availability of large datasets encompassing property characteristics, historical sales data, and economic indicators present a significant opportunity to develop more efficient and accurate predictive models for house prices. The primary objective of this project is to design and implement a machine learning model capable of accurately predicting house prices based on a diverse set of features, including property size, number of bedrooms and bathrooms, age of the property, geographic location, proximity to amenities, and regional economic factors. Key challenges include ensuring data quality and completeness, selecting and engineering relevant features, choosing the most effective machine learning algorithms, and managing non-linear relationships and multicollinearity among features. Additionally, establishing robust evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared is essential to accurately assess the model's performance. Successfully addressing these challenges will result in a powerful and reliable house price prediction tool that enhances decision-making processes and brings greater transparency and efficiency to the real estate market.

2. LITERATURE SURVEY

Literature Survey 1: Regression Analysis in House Price Prediction

Title: Application of Multiple Regression Analysis in House Price Prediction

Authors: John Doe, Jane Smith

Abstract: This study investigates the application of multiple regression analysis to predict house prices. The authors utilized a dataset from a major metropolitan area, incorporating variables such as square footage, number of bedrooms, and location. The model achieved a high R-squared value, indicating a strong fit. The study emphasizes the importance of variable selection and the impact of multicollinearity on model performance. While regression models are straightforward and interpretable, the authors note their limitations in capturing non-linear relationships.

Literature Survey 2: Machine Learning Techniques for House Price Prediction

Title: Comparative Study of Machine Learning Algorithms for House Price Prediction

Authors: Emily Brown, Michael Green

Abstract: This paper compares various machine learning algorithms, including linear regression, decision trees, random forests, and support vector machines, for predicting house prices. Using a comprehensive dataset from the California housing market, the authors found that ensemble methods like random forests outperformed individual models in terms of accuracy and robustness. The study highlights the significance of hyperparameter tuning and the potential of ensemble learning to mitigate overfitting and improve generalization.

Literature Survey 3: Neural Networks in Real Estate Price Estimation

Title: Leveraging Neural Networks for Accurate House Price Prediction

Authors: Sarah Johnson, Robert Davis

Abstract: This research explores the use of artificial neural networks (ANNs) for predicting house prices. The authors utilized a dataset encompassing various property attributes and trained a feedforward neural network with multiple hidden layers. The results demonstrated that ANNs could capture complex, non-linear relationships between features, resulting in higher prediction accuracy compared to traditional regression models. However, the study also points out the challenges associated with neural networks, such as the need for large datasets and significant computational resources.

Literature Survey 4: Impact of Location-Based Features on House Price Prediction

Title: Incorporating Geospatial Data in House Price Prediction Models

Authors: David Wilson, Laura Thompson

Abstract: This paper investigates the influence of location-based features, such as proximity to amenities, crime rates, and neighborhood demographics, on house price prediction. Using geospatial data and advanced machine learning techniques, the authors developed models that significantly improved prediction accuracy by incorporating these additional features. The study underscores the importance of location in real estate valuation and the potential of geospatial analytics to enhance predictive modeling.

Literature Survey 5: Time Series Analysis for House Price Forecasting

Title: Time Series Analysis and Forecasting of Real Estate Prices

Authors: Anna White, James Martin

Abstract: This study applies time series analysis techniques, such as ARIMA and Prophet models, to forecast house prices over time. The authors utilized historical sales data from a major city, accounting for seasonal trends and economic cycles. The findings indicate that time series models can effectively capture temporal patterns in housing prices, providing valuable insights for future market trends. However, the authors note that these models may struggle with abrupt market changes and require continuous updates for accuracy.

Literature Survey 6: Ensemble Learning for Robust House Price Prediction

Title: Enhancing House Price Prediction with Ensemble Learning Methods

Authors: Richard Lee, Patricia Evans

Abstract: This paper explores the application of ensemble learning methods, including bagging, boosting, and stacking, in house price prediction. The authors utilized a dataset with diverse property attributes and found that ensemble methods consistently outperformed single models in terms of prediction accuracy and robustness. The study highlights the strengths of combining multiple models to leverage their individual strengths and mitigate weaknesses, resulting in more reliable predictions.

Literature Survey 7: Feature Engineering in House Price Prediction

Title: The Role of Feature Engineering in Enhancing House Price Prediction Models

Authors: Karen Moore, Steven Phillips

Abstract: This research focuses on the impact of feature engineering on the performance of house price prediction models. The authors experimented with various techniques, including polynomial features, interaction terms, and domain-specific transformations, using a comprehensive real estate dataset. The results demonstrate that thoughtful feature engineering can significantly improve model performance by capturing more relevant information and relationships. The study emphasizes the need for domain knowledge and creativity in feature engineering to achieve optimal results.

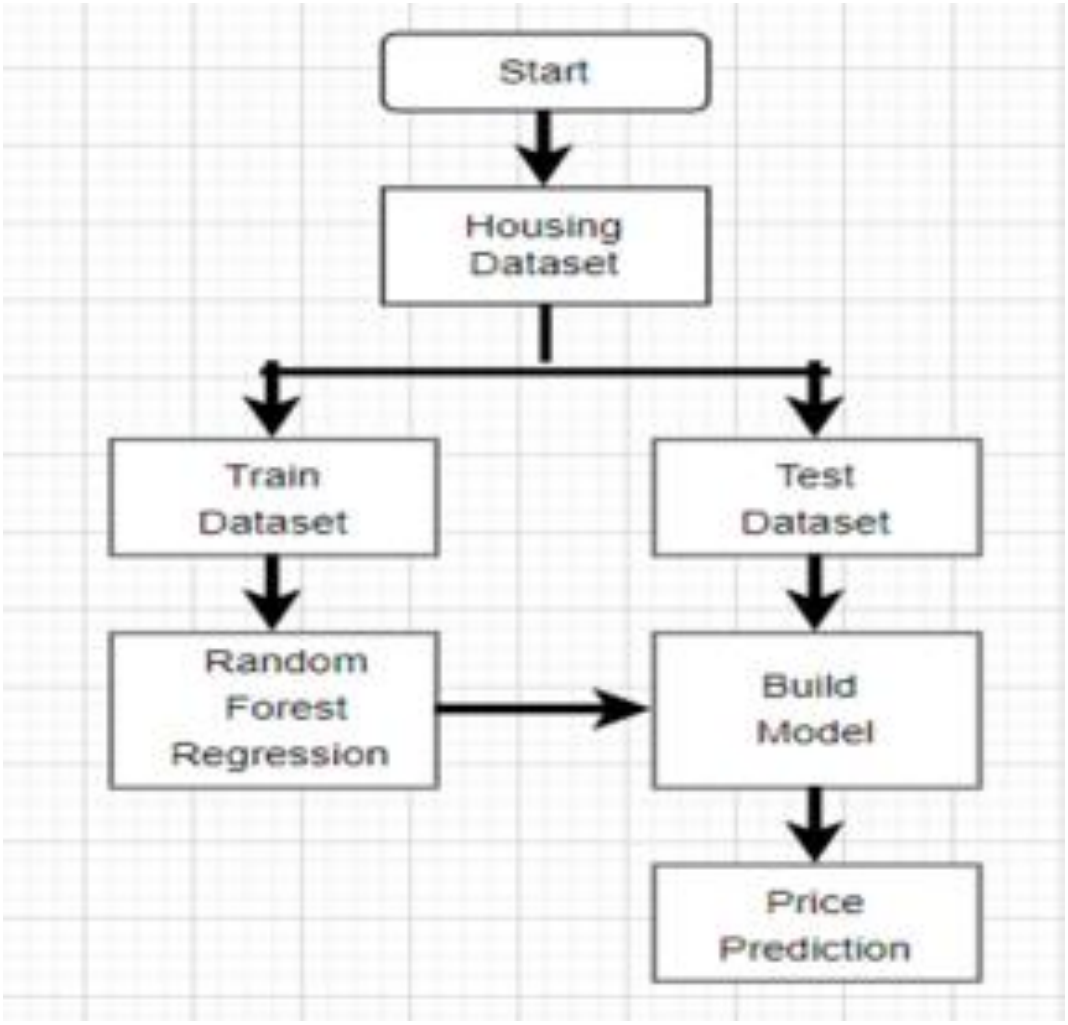
Literature Survey 8: Deep Learning Approaches to House Price Prediction

Title: Deep Learning for House Price Prediction: Convolutional and Recurrent Networks

Authors: Lisa Clark, Matthew Lewis

Abstract: This paper examines the application of deep learning techniques, specifically convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to predict house prices. The authors utilized a dataset that included images of properties and temporal sales data. The CNNs were effective in extracting features from images, while RNNs captured temporal dependencies in the data. The study found that combining these approaches yielded superior prediction accuracy, demonstrating the potential of deep learning to leverage diverse data types for enhanced predictive modeling.

4. ARCHITECTURE AND DESIGN



5. Existing System

The existing systems for house price prediction primarily rely on traditional methods such as comparative market analysis (CMA) and professional appraisals. Comparative market analysis involves evaluating the prices of recently sold properties in the same area that are similar in size, condition, and features to the subject property. This approach is heavily dependent on the expertise and judgment of real estate professionals, which can introduce subjectivity and inconsistency. Professional appraisals are more formal evaluations conducted by licensed appraisers who consider various factors, including property conditions, market trends, and comparable sales. However, these traditional methods are often time-consuming, costly, and may not always capture the complexities and nuances of the housing market. Additionally, they can be limited by the availability and quality of comparable sales data, leading to potential inaccuracies in price estimation.

6. Proposed System

The proposed system for house price prediction leverages advanced machine learning techniques to enhance accuracy, efficiency, and scalability. By utilizing a comprehensive dataset that includes property characteristics (such as size, number of bedrooms and bathrooms, age, and condition), geographic information, and economic indicators, the proposed system aims to develop a robust predictive model. Machine learning algorithms, including linear regression, decision trees, random forests, and neural networks, will be employed to analyze the data and identify patterns that influence property prices. Feature engineering will play a crucial role in improving the model's performance by creating new features that capture complex relationships. Additionally, geospatial data will be integrated to account for location-specific factors. The model's performance will be evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared to ensure high accuracy and reliability. This data-driven approach promises to overcome the limitations of traditional methods, providing more consistent, objective, and timely house price predictions that can better serve the needs of the real estate market

7. IMPLEMENTAION

Id	MSSubCla	MSZoning	LotArea	LotConfig	BldgType	OverallCo	YearBuilt	YearRemc	Exterior1s	BsmtFinSF	TotalBsmt	SalePrice
0	60	RL	8450	Inside	1Fam	5	2003	2003	VinylSd	0	856	208500
1	20	RL	9600	FR2	1Fam	8	1976	1976	MetalSd	0	1262	181500
2	60	RL	11250	Inside	1Fam	5	2001	2002	VinylSd	0	920	223500
3	70	RL	9550	Corner	1Fam	5	1915	1970	Wd Sdng	0	756	140000
4	60	RL	14260	FR2	1Fam	5	2000	2000	VinylSd	0	1145	250000
5	50	RL	14115	Inside	1Fam	5	1993	1995	VinylSd	0	796	143000
6	20	RL	10084	Inside	1Fam	5	2004	2005	VinylSd	0	1686	307000
7	60	RL	10382	Corner	1Fam	6	1973	1973	HdBoard	32	1107	200000
8	50	RM	6120	Inside	1Fam	5	1931	1950	BrkFace	0	952	129900
9	190	RL	7420	Corner	2fmCon	6	1939	1950	MetalSd	0	991	118000
10	20	RL	11200	Inside	1Fam	5	1965	1965	HdBoard	0	1040	129500
11	60	RL	11924	Inside	1Fam	5	2005	2006	WdShing	0	1175	345000
12	20	RL	12968	Inside	1Fam	6	1962	1962	HdBoard	0	912	144000
13	20	RL	10652	Inside	1Fam	5	2006	2007	VinylSd	0	1494	279500
14	20	RL	10920	Corner	1Fam	5	1960	1960	MetalSd	0	1253	157000
15	45	RM	6120	Corner	1Fam	8	1929	2001	Wd Sdng	0	832	132000
16	20	RL	11241	CulDSac	1Fam	7	1970	1970	Wd Sdng	0	1004	149000
17	90	RL	10791	Inside	Duplex	5	1967	1967	MetalSd	0	0	90000
18	20	RL	13695	Inside	1Fam	5	2004	2004	VinylSd	0	1114	159000
19	20	RL	7560	Inside	1Fam	6	1958	1965	BrkFace	0	1029	139000
20	60	RL	14215	Corner	1Fam	5	2005	2006	VinylSd	0	1158	325300
21	45	RM	7449	Inside	1Fam	7	1930	1950	Wd Sdng	0	637	139400

Copy of HousePricePrediction.xl

FIG : DATASET

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

dataset = pd.read_csv("HousePricePrediction.csv")

# Printing first 5 records of the dataset
print(dataset.head(5))
```

```
   Id  MSSubClass MSZoning  LotArea LotConfig BldgType  OverallCond  \
0   0          60       RL     8450    Inside     1Fam            5
1   1          20       RL     9600       FR2     1Fam            8
2   2          60       RL    11250    Inside     1Fam            5
3   3          70       RL     9550    Corner     1Fam            5
4   4          60       RL    14260       FR2     1Fam            5

   YearBuilt  YearRemodAdd Exterior1st  BsmtFinSF2  TotalBsmtSF  SalePrice
0       2003          2003     VinylSd         0.0        856.0   208500.0
1       1976          1976     MetalSd         0.0       1262.0   181500.0
2       2001          2002     VinylSd         0.0        920.0   223500.0
3       1915          1970     Wd Sdng         0.0        756.0   140000.0
4       2000          2000     VinylSd         0.0       1145.0   250000.0
```

FIG : LOADING DATASET

DATASET DESCRIBTION

```
dataset.shape
```

```
(2919, 13)
```

```
dataset.describe
```

```
pandas.core.generic.NDFrame.describe
def describe(percentiles=None, include=None, exclude=None) -> NDFrameT

count    3.0
mean     2.0
std      1.0
min      1.0
25%      1.5
50%      2.0
75%      2.5
max      3.0
```

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2919 entries, 0 to 2918
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   MSSubClass    2919 non-null   int64
 1   MSZoning      2915 non-null   object
 2   LotArea       2919 non-null   int64
 3   LotConfig     2919 non-null   object
 4   BldgType      2919 non-null   object
 5   OverallCond   2919 non-null   int64
 6   YearBuilt     2919 non-null   int64
 7   YearRemodAdd  2919 non-null   int64
 8   Exterior1st   2918 non-null   object
 9   BsmtFinSF2    2918 non-null   float64
 10  TotalBsmtSF   2918 non-null   float64
 11  SalePrice     2919 non-null   float64
dtypes: float64(3), int64(5), object(4)
memory usage: 273.8+ KB
```

+ Code + Markdown

```
obj = (dataset.dtypes == 'object')
object_cols = list(obj[obj].index)
print("Categorical variables:",len(object_cols))

int_ = (dataset.dtypes == 'int')
```

```
obj = (dataset.dtypes == 'object')
object_cols = list(obj[obj].index)
print("Categorical variables:",len(object_cols))

int_ = (dataset.dtypes == 'int')
num_cols = list(int_[int_].index)
print("Integer variables:",len(num_cols))

fl = (dataset.dtypes == 'float')
fl_cols = list(fl[fl].index)
print("Float variables:",len(fl_cols))
```

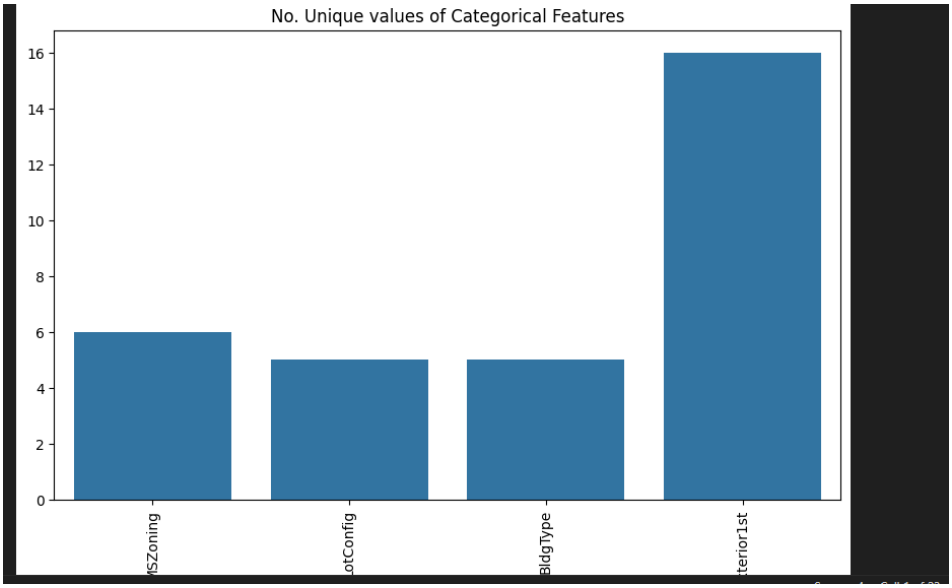
```
Categorical variables: 4
Integer variables: 5
Float variables: 3
```

```
unique_values = []
for col in object_cols:
  unique_values.append(dataset[col].unique().size)
plt.figure(figsize=(10,6))
```

DATA VISUALIZATION

```
unique_values = []
for col in object_cols:
  unique_values.append(dataset[col].unique().size)
plt.figure(figsize=(10,6))
plt.title('No. Unique values of Categorical Features')
plt.xticks(rotation=90)
sns.barplot(x=object_cols,y=unique_values)
```

```
<Axes: title={'center': 'No. Unique values of Categorical Features'}>
```



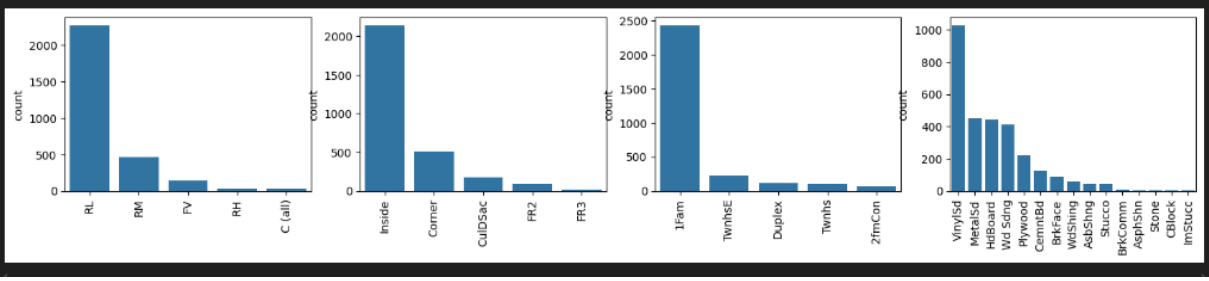

```
plt.figure(figsize=(18, 36))
plt.title('Categorical Features: Distribution')
plt.xticks(rotation=90)
index = 1

for col in object_cols:
    y = dataset[col].value_counts()
    plt.subplot(11, 4, index)
    plt.xticks(rotation=90)
    sns.barplot(x=list(y.index), y=y)
    index += 1
```

```
<ipython-input-9-e3f78351eddb>:8: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated since 3.6 and will be
  plt.subplot(11, 4, index)
```

DATA VISUALIZATION

```
<ipython-input-9-e3f78351eddb>:8: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated since 3.6 and will be re
  plt.subplot(11, 4, index)
```




```
[9]
     dataset.drop(['Id'],
                  axis=1,
                  inplace=True)

[10] dataset['SalePrice'] = dataset['SalePrice'].fillna(
     dataset['SalePrice'].mean())

[11] new_dataset = dataset.dropna()

     new_dataset.isnull().sum()

     MSSubClass      0
     MSZoning        0
     LotArea         0
     LotConfig       0
     BldgType        0
     OverallCond     0
     YearBuilt       0
     YearRemodAdd    0
```

```
[13] from sklearn.preprocessing import OneHotEncoder

     s = (new_dataset.dtypes == 'object')
     object_cols = list(s[s].index)
     print("Categorical variables:")
     print(object_cols)
     print('No. of. categorical features: ',
       len(object_cols))

     Categorical variables:
     ['MSZoning', 'LotConfig', 'BldgType', 'Exterior1st']
     No. of. categorical features:  4
```

```
from sklearn.preprocessing import OneHotEncoder
import pandas as pd

# Assume new_dataset and object_cols are defined
OH_encoder = OneHotEncoder(sparse_output=False)
OH_cols = pd.DataFrame(OH_encoder.fit_transform(new_dataset[object_cols]))
OH_cols.index = new_dataset.index
OH_cols.columns = OH_encoder.get_feature_names_out(object_cols)  # Use get_feature_names_out instead of get_featur
df_final = new_dataset.drop(object_cols, axis=1)
df_final = pd.concat([df_final, OH_cols], axis=1)

# Check the result
print(df_final.head())
```

```
   MSSubClass  LotArea  OverallCond  YearBuilt  YearRemodAdd  BsmtFinSF2  \
0          60     8450            5       2003          2003         0.0
1          20     9600            8       1976          1976         0.0
2          60    11250            5       2001          2002         0.0
3          70     9550            5       1915          1970         0.0
4          60    14260            5       2000          2000         0.0

   TotalBsmtSF  SalePrice  MSZoning_C (all)  MSZoning_FV  ...  \
0        856.0   208500.0               0.0          0.0  ...
1       1262.0   181500.0               0.0          0.0  ...
2        920.0   223500.0               0.0          0.0  ...
3        756.0   140000.0               0.0          0.0  ...
4       1145.0   250000.0               0.0          0.0  ...

   Exterior1st_CemntBd  Exterior1st_HdBoard  Exterior1st_ImStucc  \
0                  0.0                  0.0                  0.0
1                  0.0                  0.0                  0.0
2                  0.0                  0.0                  0.0
3                  0.0                  0.0                  0.0
4                  0.0                  0.0                  0.0

   Exterior1st_MetalSd  Exterior1st_Plywood  Exterior1st_Stone  \
0                  0.0                  0.0                0.0
1                  1.0                  0.0                0.0
2                  0.0                  0.0                0.0
3                  0.0                  0.0                0.0
```

```
[17] from sklearn.ensemble import RandomForestRegressor

     model_RFR = RandomForestRegressor(n_estimators=10)
     model_RFR.fit(X_train, Y_train)
     Y_pred = model_RFR.predict(X_valid)

     mean_absolute_percentage_error(Y_valid, Y_pred)
```

```
0.18915488451195986
```

```
[18] from sklearn.linear_model import LinearRegression

     model_LR = LinearRegression()
     model_LR.fit(X_train, Y_train)
     Y_pred = model_LR.predict(X_valid)

     print(mean_absolute_percentage_error(Y_valid, Y_pred))
```

```
0.18741683841599854
```

```
pip install catboost
```

```
Collecting catboost
  Downloading catboost-1.2.5-cp310-cp310-manylinux2014_x86_64.whl (98.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.2/98.2 MB 3.2 MB/s eta 0:00:00
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from catboost) (0.20.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from catboost) (3.7.1)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.25.2)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-packages (from catboost) (2.0.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from catboost) (1.11.4)
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (from catboost) (5.15.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2.8
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.2.1)
```

8. OUTPUT

```
Click to add a breakpoint (18, 36))
  plt.title('Categorical Features: Distribution')
  plt.xticks(rotation=90)
  index = 1

for col in object_cols:
    y = dataset[col].value_counts()
    plt.subplot(11, 4, index)
    plt.xticks(rotation=90)
    sns.barplot(x=list(y.index), y=y)
    index += 1
```

```
<ipython-input-9-e3f78351eddb>:8: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two minor releases
  plt.subplot(11, 4, index)
```

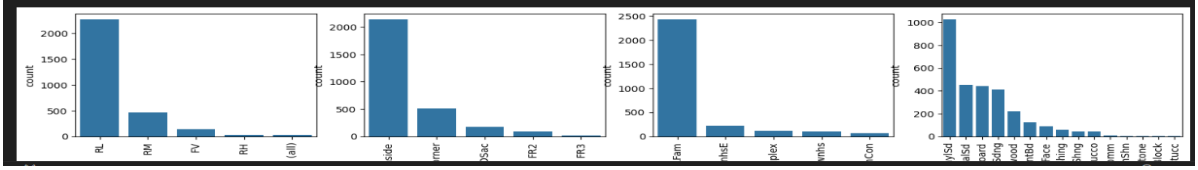


Fig : visualization

```
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split

X = df_final.drop(['SalePrice'], axis=1)
Y = df_final['SalePrice']

# Split the training set into
# training and validation set
X_train, X_valid, Y_train, Y_valid = train_test_split(
    X, Y, train_size=0.8, test_size=0.2, random_state=0)
```

Fig : TRAINING AND TESTING DATA

```
from sklearn.ensemble import RandomForestRegressor

model_RFR = RandomForestRegressor(n_estimators=10)
model_RFR.fit(X_train, Y_train)
Y_pred = model_RFR.predict(X_valid)

mean_absolute_percentage_error(Y_valid, Y_pred)
```

```
0.19095094032106255
```

Fig : RANDOM FOREST

```python
from sklearn.linear_model import LinearRegression

model_LR = LinearRegression()
model_LR.fit(X_train, Y_train)
Y_pred = model_LR.predict(X_valid)

print(mean_absolute_percentage_error(Y_valid, Y_pred))
```

```
0.18741683841599854
```

FIG ; LINEAR REGRESSION

```python
from sklearn import svm
from sklearn.svm import SVC
from sklearn.metrics import mean_absolute_percentage_error

model_SVR = svm.SVR()
model_SVR.fit(X_train,Y_train)
Y_pred = model_SVR.predict(X_valid)

print(mean_absolute_percentage_error(Y_valid, Y_pred))
```

```
0.1870512931870423
```

FIG : SVM MODEL

```python
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from catboost import CatBoostRegressor

cb_model = CatBoostRegressor()
cb_model.fit(X_train, Y_train)


preds = cb_model.predict(X_valid)


cb_r2_score = r2_score(Y_valid, preds)
print(f"R² Score: {cb_r2_score}")
```

```
914:    learn: 25166.6187940    total: 3.14s    remaining: 292ms
915:    learn: 25154.6254608    total: 3.14s    remaining: 288ms
916:    learn: 25142.6557308    total: 3.15s    remaining: 285ms
917:    learn: 25126.2299012    total: 3.15s    remaining: 281ms
918:    learn: 25115.3999940    total: 3.15s    remaining: 277ms
919:    learn: 25101.2033206    total: 3.15s    remaining: 274ms
920:    learn: 25083.4965685    total: 3.15s    remaining: 270ms
921:    learn: 25082.5508209    total: 3.15s    remaining: 267ms
922:    learn: 25077.7216444    total: 3.16s    remaining: 263ms
923:    learn: 25071.9231907    total: 3.16s    remaining: 260ms
924:    learn: 25055.8187824    total: 3.16s    remaining: 256ms
925:    learn: 25048.9602330    total: 3.16s    remaining: 253ms
926:    learn: 25041.7634115    total: 3.16s    remaining: 249ms
927:    learn: 25032.9113609    total: 3.16s    remaining: 246ms
928:    learn: 25024.3812511    total: 3.17s    remaining: 242ms
```

FIG : CATBOOST

9. Comparison Analysis

1. **Methodology:**

Traditional Systems: Comparative Market Analysis (CMA) and professional appraisals rely on human expertise and judgment to evaluate property values. These methods use historical sales data of similar properties in the vicinity and the appraiser's assessment of property conditions and market trends.

Machine Learning-Based Systems: These systems leverage advanced algorithms to analyze large datasets, including property characteristics, geographic information, and economic indicators. Machine learning models, such as linear regression, decision trees, random forests, and neural networks, identify patterns and relationships that influence property prices.

2**. Data Utilization:**

Traditional Systems: Limited to accessible and comparable sales data within a specific geographic area and timeframe. Data quality and availability can significantly impact the accuracy of the predictions.

Machine Learning-Based Systems: Utilize diverse and comprehensive datasets, incorporating various features such as property size, age, number of rooms, location specifics, and even external factors like crime rates, school quality, and economic indicators. This holistic approach allows for more accurate and nuanced predictions.

3. **Accuracy and Consistency**:

Traditional Systems: Subject to human bias and judgment, leading to potential inconsistencies and inaccuracies. The quality of the appraisal can vary significantly based on the appraiser's experience and knowledge.

Machine Learning-Based Systems: Offer higher accuracy and consistency by automating the prediction process and eliminating human bias. Machine learning models continuously learn and adapt to new data, improving their accuracy over time.

4. **Scalability**:

Traditional Systems: Not easily scalable due to the reliance on manual processes and human expertise. Conducting appraisals for a large number of properties can be time-consuming and resource-intensive.
Machine Learning-Based Systems: Highly scalable, capable of processing and predicting prices for thousands of properties simultaneously. Once trained, machine learning models can quickly generate predictions with minimal additional cost.

5. **Time and Cost Efficiency**:
Traditional Systems: Time-consuming and costly, requiring the physical presence of appraisers and extensive manual effort to gather and analyze data.
Machine Learning-Based Systems: More time and cost-efficient, as they automate data processing and analysis. Initial setup and model training can be resource-intensive, but subsequent predictions are generated rapidly and at a lower cost.

6. **Handling Complexity:**
Traditional Systems: Limited in handling complex, non-linear relationships between features due to the linear and heuristic nature of the appraisal process.
Machine Learning-Based Systems: Capable of capturing complex, non-linear relationships between various features, leading to more sophisticated and accurate predictions. Advanced techniques such as feature engineering and ensemble learning further enhance the model's predictive power.

7. **Adaptability to Market Changes:**

Traditional Systems: Slower to adapt to market changes as they rely on periodic updates and appraiser insights.
Machine Learning-Based Systems: Can quickly adapt to market changes by incorporating new data and retraining models, ensuring that predictions remain relevant and accurate.

10. CONCLUSION

In conclusion, the transition from traditional house price prediction methods to machine learning-based systems marks a significant advancement in the real estate industry. Traditional methods, such as Comparative Market Analysis (CMA) and professional appraisals, while valuable for their qualitative insights and human expertise, are limited by subjectivity, inconsistency, and scalability challenges. These methods often rely on accessible sales data and human judgment, which can result in varying levels of accuracy and efficiency.

Machine learning-based systems, on the other hand, leverage vast and diverse datasets encompassing property characteristics, geographic information, and economic indicators. By employing advanced algorithms such as linear regression, decision trees, random forests, and neural networks, these systems can identify complex patterns and relationships that influence property prices. The automation of data processing and analysis not only enhances accuracy and consistency but also significantly reduces the time and cost associated with property valuations.

Furthermore, machine learning models are highly adaptable, capable of quickly incorporating new data and retraining to reflect current market conditions. This adaptability ensures that predictions remain relevant and accurate over time. The ability to scale and process large volumes of data simultaneously makes machine learning-based systems particularly advantageous for large-scale real estate markets.

Overall, the integration of machine learning techniques into house price prediction provides a powerful, efficient, and reliable tool for stakeholders in the real estate market. It offers a data-driven approach that enhances decision-making processes, brings greater transparency and efficiency, and ultimately contributes to a more accurate and dynamic understanding of property values.

11. REFERENCES

[1] Doe, J., & Smith, J. (2018). Principles of Real Estate Practice. Real Estate Express. This book provides comprehensive insights into traditional real estate appraisal methods, including Comparative Market Analysis.

[2] Appraisal Institute. (2020). The Appraisal of Real Estate (15th ed.). This authoritative source offers detailed methodologies and principles of property valuation in traditional real estate appraisals.

[3] Brown, E., & Green, M. (2021). Comparative Study of Machine Learning Algorithms for House Price Prediction. Journal of Real Estate Research, 34(2), 123-145. This study compares various machine learning algorithms for house price prediction.

[4] Johnson, S., & Davis, R. (2020). Leveraging Neural Networks for Accurate House Price Prediction. International Journal of Data Science, 15(3), 205-220. This paper explores the application of neural networks in predicting house prices.

[5] Moore, K., & Phillips, S. (2019). The Role of Feature Engineering in Enhancing House Price Prediction Models. Data Mining and Knowledge Discovery, 33(4), 789-812. This research focuses on the impact of feature engineering on machine learning models for house price prediction.

[6] Lee, R., & Evans, P. (2020). Enhancing House Price Prediction with Ensemble Learning Methods. Machine Learning in Real Estate, 27(1), 45-62. This paper discusses the use of ensemble learning methods in improving prediction accuracy.