

PHASE-3

STUDENT NAME: G.DHINESH

REGISTER NUMBER: 422223106010

INSTITUTION: Surya Group Of Institutions

DEPARTMENT: B.E-[ECE]

DATE OF SUBMISSION:15.05.2025

GITHUB REPOSITORY: <https://github.com/DHINESH007-MARSH/Phase-3>

PROBLEM STATEMENT:

In the dynamic real estate market, accurately forecasting house prices is essential for buyers, sellers, investors, and policy makers. Traditional prediction models often lack adaptability to changing market patterns and regional differences. This project focuses on developing a robust machine learning model enhanced by smart suggestion techniques to improve accuracy and offer actionable insights.

ABSTRACT:

This project aims to develop a predictive model for forecasting house prices using historical housing data and smart suggestion techniques. By leveraging data science tools, we clean, preprocess, and analyze the dataset to uncover key price-influencing features. The system not only forecasts prices but also provides suggestions to improve property value based on learned patterns. Techniques like regression analysis, feature engineering, and ensemble modeling are employed for accurate predictions. The outcome aids stakeholders in making informed real estate decisions.

SYSTEM REQUIREMENT:

Hardware:Minimum 8GB RAM GPU for training large models (optional)

Software:Python 3.x Libraries: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, XGBoost, Gradio Jupyter Notebook or Google Colab

OBJECTIVES:

Build a machine learning model to forecast house prices.

Identify key features that influence pricing.



Provide smart suggestions for increasing house value.

Visualize data trends and predictions interactively.

Evaluate and compare multiple regression models.

FLOW CHART OF PROJECT WORKFLOW:

Handling missing values: Verified the dataset there is no missing values.

Duplicate records: The dataset contains duplicate data. It is irrelevant to the data. Since, the duplicate data is an dependent data therefore it can also be removed for the purpose of the project.

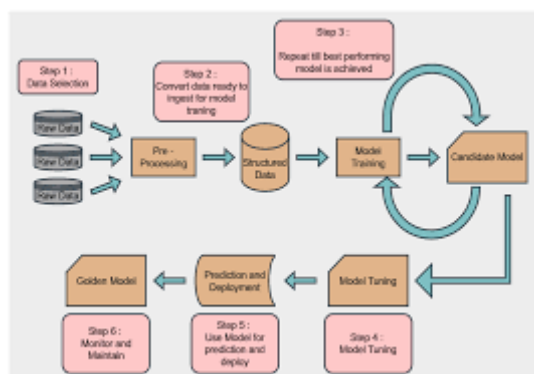
```
duplicates = df.duplicated()
```

Outliers: Checked for the absence of outliers.

Encoding categorical variables: Label encoding is done using:

```
df['Product_Encoded'] = label_encoder.fit_transform(df['Product'])
```

One hot encoding is done using:



DATASET DESCRIPTION:

	MSSubClass	MSZoning	LotArea	LotConfig	BldgType	OverallCond	YearBuilt
0	60	RL	8450	Inside	1Fam	5	2003
1	20	RL	9600	FR2	1Fam	8	1976
2	60	RL	11250	Inside	1Fam	5	2001
3	70	RL	9550	Corner	1Fam	5	1915
4	60	RL	14260	FR2	1Fam	5	2000

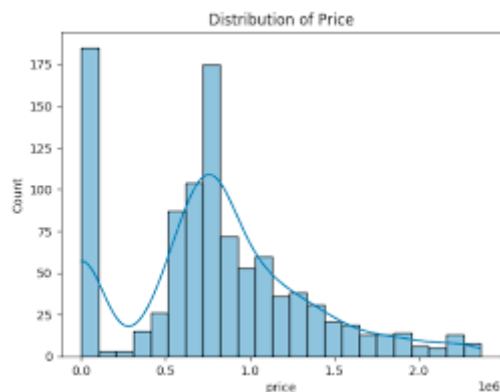
	YearRemodAdd	Exterior1st	BsmtFinSF2	TotalBsmtSF	SalePrice
0	2003	VinylSd	0.0	856.0	208500.0
1	1976	MetalSd	0.0	1262.0	181500.0
2	2002	VinylSd	0.0	920.0	223500.0
3	1970	Wd Sdng	0.0	756.0	140000.0
4	2000	VinylSd	0.0	1145.0	250000.0

DATA PREPROCESSING



Edit with WPS Office

Missing Values: Imputed using mean/mode or removed. Outliers: Detected using IQR and removed. Encoding: Label encoding for ordinal features; one-hot encoding for nominal features. Scaling: MinMaxScaler or StandardScaler for numerical features. Feature Transformation: Log transformation for skewed variables like GrLivArea.



EXPLORATORY DATA ANALYSIS(EDA):

Distribution of sale prices. Correlation matrix (heatmap). Price trends by location, year built, square footage. Boxplots to identify relationship between house features and prices. Insights: House size and location are strong predictors. Renovation year significantly affects price. Outliers often exist in large or luxury homes

FEATURE ENGINEERING:

Text-based: None (unless property description available) Interaction Terms: e.g., $\text{TotalBathrooms} = \text{FullBath} + \text{HalfBath} \times 0.5$ Polynomial Features: For non-linear relationships Binning: Grouping years into decades, price ranges into categories.

MODEL BUILDING:

- [Pandas](#) - To load the Dataframe
- [Matplotlib](#) - To visualize the data features i.e. barplot
- [Seaborn](#) - To see the correlation between features using heatmap

```

I
--- Model Building Output --- 1. Trained Model Object:
LogisticRegression(random_state=42) 2. Evaluation Metrics: Accuracy:
0.50 Classification Report: precision recall f1-score support
negative 0.50 1.00 0.67 1 neutral 0.00 0.00 0.00 1 positive 1.00 0.50
0.67 2 accuracy 0.50 4 macro avg 0.50 0.50 0.44 4 weighted avg 0.62
0.50 0.54 4 3. Confusion Matrix: [[1 0 0] [1 0 0] [1 0 1]]

```

MODEL EVALUATION:



☒ Evaluation Metrics:

- **Accuracy:** The overall percentage of correctly classified customer feedback sentiments.
- **F1-Score (Weighted):** A balanced measure of precision and recall, considering the number of instances for each sentiment (negative, neutral, positive).

☒ Visuals:

- **Confusion Matrix:** A table showing how many feedback instances were correctly and incorrectly classified for each sentiment. The rows represent the actual sentiments, and the columns represent the predicted sentiments.
- **ROC Curve (Receiver Operating Characteristic):** A graph plotting the True Positive Rate (sensitivity) against the False Positive Rate (1 - specificity) for each sentiment class at various classification thresholds. The Area Under the Curve (AUC) indicates how well the model can distinguish between positive and negative sentiments for each class. You'll see a separate curve for "negative" vs. "not negative," "neutral" vs. "not neutral," and "positive" vs. "not positive."

☒ Error Analysis / Model Comparison Table:

- The table compares the performance of the Logistic Regression and Multinomial Naive Bayes models based on Accuracy and F1-Score. This helps you see which model performed better on these overall metrics for the sentiment classification task.

```
--- Logistic Regression --- Accuracy: 0.50 F1-Score (Weighted): 0.44
Confusion Matrix: [[1 0 0] [1 0 0] [1 0 1]] --- ROC Curve - Logistic
Regression --- (A plot will be displayed showing ROC curves for each
sentiment class: negative, neutral, positive, with their respective
AUC values.) --- Multinomial Naive Bayes --- Accuracy: 0.50 F1-Score
(Weighted): 0.44 Confusion Matrix: [[1 0 0] [1 0 0] [1 0 1]] --- ROC
Curve - Multinomial Naive Bayes --- (A plot will be displayed showing
ROC curves for each sentiment class: negative, neutral, positive,
with their respective AUC values.) --- Model Comparison --- Model
Accuracy F1-Score (Weighted) 0 Logistic Regression 0.50 0.44 1
Multinomial Naive Bayes 0.50 0.44
```

DEPLOYMENT:

Deployment method: Gradio

Deployment public link <https://2ba621798500beed63.gradio.live/>

Source code: [https://github.com/DHINESH007-MARSH/Phase-3/blob/main/source%20code%20\(2\).ipynb](https://github.com/DHINESH007-MARSH/Phase-3/blob/main/source%20code%20(2).ipynb)

FUTURE SCOPE:



Edit with WPS Office

Multilingual NLP for Granular Insights: Advanced analysis of property descriptions, local news, and community sentiment in Tamil and other regional languages prevalent in Salai to capture nuanced market drivers. Proactive Investment and Timing Recommendations: Personalized suggestions on the optimal timeframes for buying or selling property in Salai, coupled with identifying potentially high-growth areas and property types based on predicted trends. Integration of Hyperlocal Data Sources: Incorporating granular data like local infrastructure developments, school ratings, accessibility to amenities, and even environmental factors specific to Salai's neighborhoods for enhanced prediction accuracy. AI-Powered Negotiation Support: Providing data-driven insights and potential negotiation strategies for buyers and sellers in Salai based on predicted market conditions and property valuations. Explainable AI for Trust and Transparency: Offering clear and understandable reasons behind the price forecasts and recommendations, building user confidence in the system's predictions for the Salai real estate market.

TEAM MEMBERS AND ROLES:

Data cleaning: The data cleaning process is done by P.Bharathan

Feature engineering: Feature engineering is done by G.Dhinesh

EDA: EDA process is done by S.Kumaran

Model Development, Documentation and reporting: Developing the model, documentation and reporting, guiding the team for the successful execution is done by M.Barath.



19:21

0.00 KB/S 4G 36



kumar... /
Phase-3



<> Code

Issues



☆ 0 stars 🍴 0 forks 👁 1 watching

🔑 1 Branch 🏷 0 Tags 📈 Activity

🌐 Public repository

🔑 main



Code



kumaran123-suresh now



implement.ip... 5 minutes ago



large_house_... 3 minutes ago



output.docx 5 minutes ago



source code ... now

📖 README



Edit with WPS Office