HOUSE PRICE PREDICTION USING MACHINE LEARNING AND DATA MINING TECHNIQUES

GROUP-7:

NAME	STUDENT ID
Ritika Kalyani	811289983
Akhil Seliveri	811290035
Navath Vamshi Krishna	811303145
Rajesh Katherasala	811314095

1. INTROUCTION:

Since the beginning of time, housing has been one of the most fundamental human needs, right up there with food and drink. The demand for homes has increased in tandem with the improvement in living standards. This demand is driven by both individual need and investment interests. The primary residence, income properties, or long-term investments are the three primary reasons why people purchase homes. Because of this ever-increasing demand, housing prices continue to rise year after year. There are several elements that contribute to this trend, including location, neighborhood amenities, and the conditions of the regional market. House price forecasting that is accurate has become increasingly important for a variety of stakeholders. Buyers are provided with information regarding potential investments, which assists them in making decisions that are in line with their financial constraints and the characteristics they want in a property. The utilization of predictive models is beneficial to both homebuilders and developers since it allows them to establish competitive rates, manage inventories, and make educated decisions regarding development. It is possible for real estate brokers to make use of such models to provide more accurate appraisals, hence increasing the level of client satisfaction and improving sales techniques. It is vital for lenders and banks to have these projections to evaluate the risks associated with mortgages and to change loan terms based on the conditions of the market.

In the field of housing price modeling, machine learning techniques such as support vector regression, neural networks, K-Nearest Neighbors, and ensemble approaches like Random Forest and AdaBoost have emerged as indispensable tools. The models in question can show patterns and connections through the examination of past data, which enables detailed market appraisals. These kinds of insights are extremely helpful because they help consumers select houses that are suitable for their lifestyle and financial circumstances, while also assisting builders and investors in making decisions that are informed by data.

Real-world uses of models that anticipate property prices include the following Real estate websites such as Zillow and Redfin offer home appraisals, which assist buyers and sellers in gaining a better understanding of market trends and values. To determine whether to approve a mortgage, lenders utilize price prediction to evaluate the worth of the property. They then change interest rates according on the level of risk involved. Cities study home price trends to promote affordable housing efforts and infrastructure planning. Predictive models are utilized by builders to ascertain economically viable locations, pricing, and features to incorporate based on the demand from buyers. These models are utilized by real estate investors to identify high-growth locations, with the goal of increasing return on investment (ROI) in properties that are either rented out or resold.

2. PROJECT DESCRIPTION:

Building a machine learning model that is capable of predicting residential property values based on important characteristics such as location, size, age, and amenities is the objective of this research. The purpose of the model is to provide purchasers, sellers, and real estate agents with realistic estimates of house values so that they can make informed financial decisions. During the course of the project, a number of different machine learning strategies, such as regression or treebased models, will be investigated in order to address the complicated nature of house pricing data, which can change greatly due to a considerable number of economic and local factors.

The management of data quality difficulties, the management of large variance in property prices, and the guaranteeing of model interpretability to make the results easily understandable by users are among the most important challenges. It is possible that one of the most important contributions will be the development of more advanced feature engineering methodologies, particularly for variables such as location, which can have a significant impact on costs. Additionally, novel modeling approaches could be utilized to improve the accuracy of predictions and the relevance of those predictions to end users.

Within the context of a collaborative team, the distribution of work is a crucial component. In most cases, the project is broken down into several important sections, including data preprocessing, model selection, training and optimization, evaluation, and documentation. Each member of the team is given a specific job within each of these stages to streamline the workflow and make the most of the skills that each individual possesses in order to finish a comprehensive and reliable prediction model.

3. BACKGROUND:

Attempting to forecast housing prices in metropolitan subregions is fraught with significant difficulties because of the complex combination of factors that have an effect on these values. Not only are social and economic elements a significant influence on housing trends, but also broader macroeconomic patterns and expectations for future price increase, which can cause a shift in the values of homes that are now on the market. The techniques of data mining make it possible to conduct an examination of historical pricing patterns and other variables that are pertinent, which assists in the development of models that can forecast future prices. Due to the fact that they offer

crucial insights for decision-making, these predictive models are extremely useful tools for a wide variety of stakeholders, including buyers, sellers, banks, and real estate experts.

Python libraries (such as Scikit-Learn and Pandas) are utilized frequently in this field for the purpose of data processing and model construction. Additionally, integrated development environments (IDEs) like Jupyter Notebook and databases like MySQL are utilized for the purpose of data management.

Standard computational resources are often sufficient for hardware, with the exception of situations in which the dataset is exceedingly huge or the model poses a particularly complicated challenge.

Programming abilities like as data analysis, machine learning, and experience with visualization and spatial data management tools are essential for this project. These skills enhance the model's capacity to give predictions that are relevant and accurate.

4. PROBLEM DEFINITION:

The creation of a machine learning model that can reliably forecast house prices based on a number of different parameters is the major objective of this study. The purpose of this problem is to estimate the selling price of residential homes by using input parameters like as location, property size, age, and amenities. The problem is written as a supervised regression because the objective is to predict the selling price. The management of the considerable variability in housing data, the guarantee of the interpretability of the model, and the incorporation of socio-economic and market-driven elements that influence housing prices are among the most significant issues. For the solution to be effective, it must be able to provide accurate forecasts that assist various stakeholders, including homebuyers, sellers, investors, and real estate brokers, in making decisions that are knowledgeable.

Challenges in addressing the issues with the project include the following: The Variability and Quality of the Data, which is Information on real estate can be inconsistent, with entries that are either missing or wrong, which might have an impact on the accuracy of predictions.

There is a substantial amount of variation in housing prices due to the fact that prices are highly dependent on local characteristics such as the quality of the neighborhood and the closeness to facilities. This variance makes it difficult to precisely forecast prices. Managing Diverse information Because housing data contains a combination of numerical and categorical information, it must undergo specialized preprocessing to be acceptable for machine learning. Interpretability of the Model It is of the utmost importance for stakeholders to see to it that the predictions made by the model are comprehensible, particularly when more complex models such as Random Forest or CatBoost are utilized.

Problem Solutions:

1. Data Preprocessing:

It is critically important to preprocess data to guarantee the quality and dependability of the model. The process of finding and correcting missing data by filling, imputing, or removing them is included in this step. Additionally, outliers that could potentially skew the learning process of

the model are considered to be eliminated. In addition, proper transformations (such scaling or normalizing) could be required for the necessary features in order to guarantee that all variables contribute in the appropriate manner during the training of the model. These preparation methods contribute to the dataset being cleaner, more consistent, and simpler for models to interpret.

2.Exploratory Data Analysis:

The process of examining and displaying data to discover patterns, distributions, and correlations is referred to as exploratory data analysis (EDA). It is essential to conduct this analysis to gain an understanding of the data structure, which can then lead to efficient feature engineering. Histograms, scatter plots, and correlation matrices are examples of visuals that are utilized during EDA. These visualizations assist in revealing crucial elements that have an effect on housing prices. These visualizations also provide insights into data patterns and correlations that may not be immediately obvious. To improve the prediction capacity of the model, these insights provide information about which traits should be retained, modified, or combined.

3. Encoding Categorical Variables:

There are a lot of categorical variables that are included in housing data. Some examples of these variables are the neighborhood or the type of home. Machine learning algorithms are unable to directly use these variables. The application of one-hot encoding, which transforms each category into a binary variable, is what provides them with the appropriateness. Using this strategy, the model is able to analyze categorical information without imposing any ordinal assumptions, which could potentially lead to inaccurate predictions. As an illustration, neighborhood categories are converted into binary features to efficiently capture location-based price implications.

4.Splitting the Data:

In order to conduct an accurate evaluation of the performance of the model, it is necessary to partition the dataset into training and testing sets. For the purpose of preventing the model from overfitting to the training data, this approach involves setting aside a subset of the data (often between 20 and 30 percent) as the test set. During testing, the model is evaluated based on data that it has not before encountered, which provides an objective assessment of its capacity for generalization. A divide like this is required to ascertain how well the model will function in situations that are based in the actual world.

5. Model Building:

In order to determine which model is the most effective, a variety of machine learning techniques are utilized. These techniques include Support Vector Machine (SVM), Linear Regression, Random Forest Regression, and CatBoost approaches. Each model has its own set of benefits; for example, Support Vector Machines (SVM) are useful for smaller to medium-sized datasets, but Random Forest and CatBoost are able to handle more complicated links and interactions across larger datasets. When numerous models are tested, comparisons may be made, which enables the selection of the strategy that is both the most accurate and the most resilient.

6. Evaluation and Tuning:

The evaluation of the model is carried out with the help of metrics such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE). These metrics estimate the accuracy of the prediction by quantifying the discrepancies between the values that were predicted and the values that were actually observed. Model performance can be further improved through fine-tuning, which is commonly accomplished through hyperparameter optimization. Not only is this phase essential for enhancing accuracy and resilience, but it also guarantees that the model is not only predictive but also well-calibrated for the purpose of producing accurate pricing estimates in situations that are based on real-world circumstances.

These phases, when taken together, offer a method that is both organized and comprehensive for the purpose of establishing an efficient model for predicting house prices, overcoming the difficulties that are associated with housing data analysis, and providing stakeholders with insights that can be put into action.

5. THE PROPOSED TECHNIQUES:

1.Framework (problem settings):

The foundation of the house price prediction framework is the employment of a strong supervised machine learning pipeline. Developing a precise regression model that forecasts residential property values based on a variety of characteristics, such as location, property attributes, and other socioeconomic factors, is the goal. In addition to achieving excellent prediction performance, this system guarantees interpretability and usefulness for a range of stakeholders, including developers, buyers, and sellers.

Problem Definition:

Predicting house values based on a set of relevant features, including location, property size, building type, and year of construction, is the main goal of the problem, which is framed as a supervised regression functions. The objective is to develop a predictive model that reduces the variation between the anticipated and actual values using the goal variable, **SalePrice,** which represents the actual selling price of homes. In addition to attaining numerical precision, the project aims to offer practical insights into the elements that have the most effects on home prices. Stakeholders, including buyers, sellers, and developers, can make better judgments if they are aware of the significance of features, such as how neighborhood quality or basement size affects home values.

For the model to be reliable and applicable in the actual world, it must also be generalizable, which means it must function effectively on data that hasn't been seen before. In order to accomplish these objectives, a number of intrinsic difficulties must be resolved. The dataset

frequently contains missing data, which might impair the model's performance if not managed appropriately. The model's perception of feature importance may also be distorted by multicollinearity, which occurs when features have a high degree of correlation. Furthermore, a variety of factors, including local market conditions, economic trends, and particular property characteristics, influence house pricing data, which often shows considerable variance. As a result, it is crucial that the model captures these subtleties. This all-encompassing strategy guarantees that the model is not only predictive but also robust, interpretable, and useful for real-world applications.

Dataset and Its Characteristics:

There are 2,919 rows and 13 columns in the dataset utilized for the house price prediction project, which includes both category and numerical variables. A housing record containing characteristics that could affect the selling price is represented by each row. The dataset is varied and includes important categorical variables that give information about the property's classification, structure, and location, such as **MSZoning** (zoning classification), **BldgType** (building type), and **LotConfig** (lot configuration). The model's capacity to quantify elements of a home's physical attributes is further enhanced by numerical features such as **YearBuilt** (the year of building), **LotArea** (the entire lot size in square feet), and **TotalBsmtSF** (the total basement square footage). The focus of the predictive analysis is the goal variable, **SalePrice**, which is the actual selling price of each home.

Dataset features:

There are 13 features in the dataset:

1. ID	to count the records.
2. MSSubClass	Indicates the kind of residence being sold.
3. MSZoning	Determines the sale's general zoning classification.
4. LotArea	The size of the lot in square feet.
5. LotConfig	Lot configuration
6. BldgType	Type of building
7. OverallCond	Rates the house's overall condition
8. YearBuilt	Year of original construction
9. YearRemodAdd	Remodel Year(If there are no additions or remodeling, the RemodAdd Remodel date is the same as the construction date.)
10. Exterior1st	The house's first exterior covering
11. BsmtFinSF2	Type 2 completed square feet .
12. Total BsmtSF.	The basement's total area in square feet.
13. SalePrice	PurchasePrice to be estimated

Despite this, there are significant issues with the dataset that needs to be resolved for efficient modeling. The model's capacity to identify patterns and relationships can be severely hampered by missing data on important characteristics such as **MSZoning**, **Exterior1st** (the type of exterior covering), and **SalePrice**. To maintain the integrity of the dataset, rigorous imputation techniques are required due to this missingness. Additionally, there are luxury or high-value residences present, which is probably why the objective variable, **SalePrice**, has a skewed distribution with a rightward tail. Model predictions may be distorted by this skewness, therefore normalization or modification is necessary to guarantee the model's accuracy and fairness across all price points. The intricacy of the dataset and the necessity of careful preprocessing and feature engineering to optimize the model's predictive performance are highlighted by these traits and difficulties.

Management of Data Quality:

Problems with missing, inconsistent, or noisy data are common in housing datasets, and they can have a big influence on how well predictive models work and how reliable they are. The framework must first detect and handle missing values using domain-specific imputation techniques in order to guarantee data quality. For example, the mean or median can be used to fill in missing numerical values, while the mode or grouped means based on relevant variables can be used to impute categorical features. It's critical to identify anomalies like outliers and inconsistencies since they can distort the model's learning process. In order to identify anomalies and correlations in the dataset and provide insights into possible problems that need to be fixed, visual tools such as heatmaps and distribution plots are employed. To cut down on noise and improve computational performance, features with low variation or no association to the target variable, SalePrice, are eliminated. The dataset is kept clean, consistent, and prepared for efficient modeling thanks to this methodical approach to data quality control.

Exploratory Data Analysis:

To find patterns, correlations, and distributions in the data, a thorough Exploratory Data Analysis (EDA) is a necessary first step. The approach determines which features are most pertinent to house prices by employing heatmaps to analyze correlations between numerical features and the goal variable SalePrice. For instance, there are frequently high relationships between SalePrice and variables like YearBuilt and TotalBsmtSF. Histograms and box plots are examples of visualization techniques that show the distribution of information and aid in spotting outliers or skewness. For instance, skewness in SalePrice can suggest that log transformation is required to normalize the data. The distribution of categorical characteristics, such as MSZoning, and their effect on home prices are examined using bar graphs and violin plots. These kinds of visualizations show intricate relationships between features in addition to offering insights into the behavior of individual elements. By providing a more thorough grasp of the structure and

relationships of the dataset, this EDA phase establishes the foundation for well-informed feature engineering and model selection.

Below are some snapshots of EDA visualisations we have done to understand the correlation between variables and to understand their distribution:

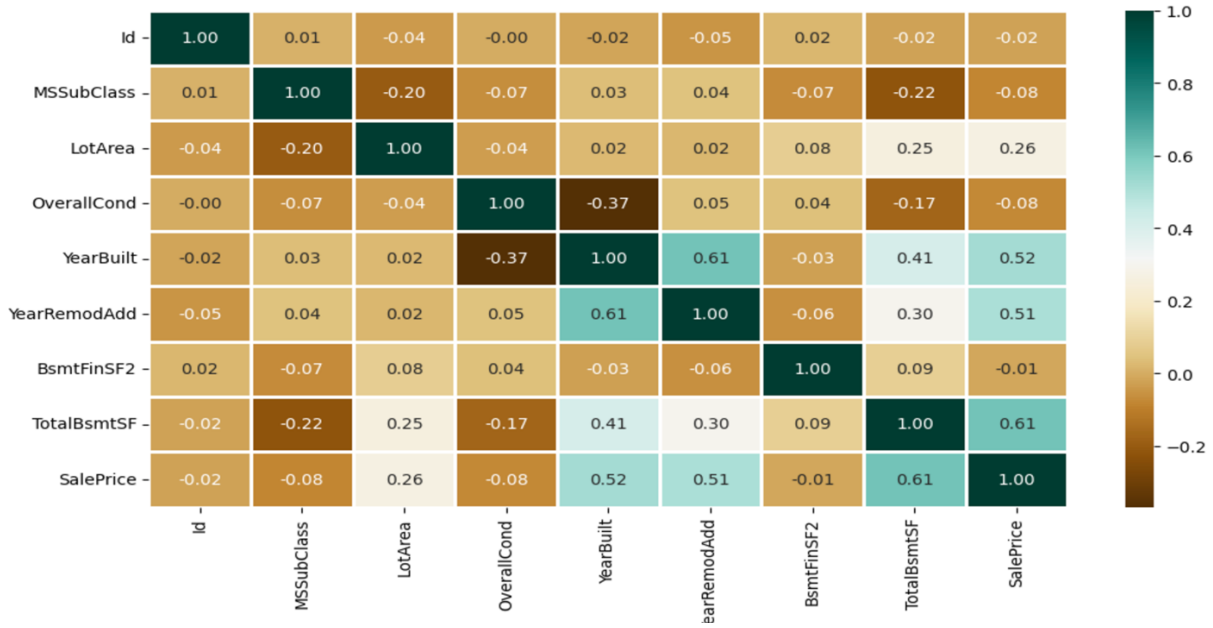


Figure: Correlation between numerical features

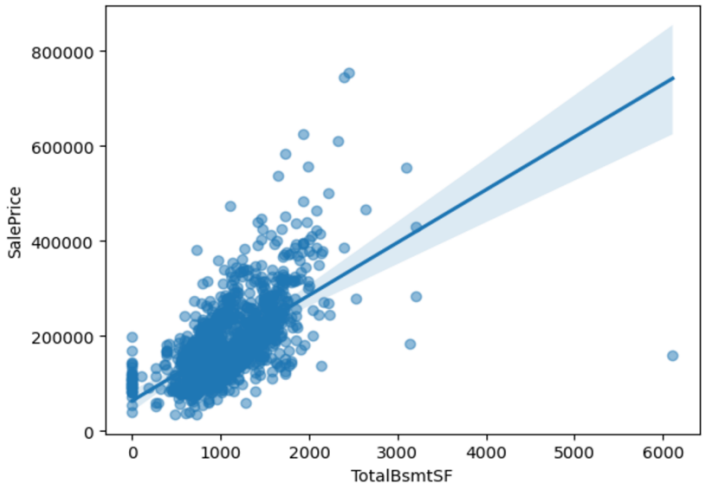


Figure: Linear relationship between salePrice and TotalBsmntSF

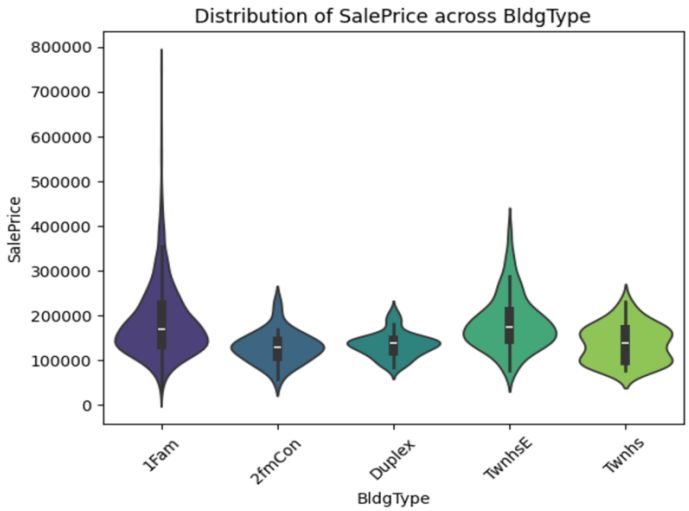


Figure: Distribution Of SalePrice across BldgType(Building Type)

Data Cleaning and Preprocessing:

In order to ensure that the dataset is dependable and consistent for efficient modeling, data cleaning and preprocessing are essential processes. Because features like **MSZoning**, **Exterior1st**, and **SalePrice** include gaps that could affect the forecast accuracy, the approach starts with finding missing values in the dataset. Since the mean maintains the general distribution without adding bias, it is used to replace missing values in the target variable, **SalePrice**. Given their modest percentage of the total dataset size, rows with missing entries are eliminated for other characteristics with few missing values, including **MSZoning**.

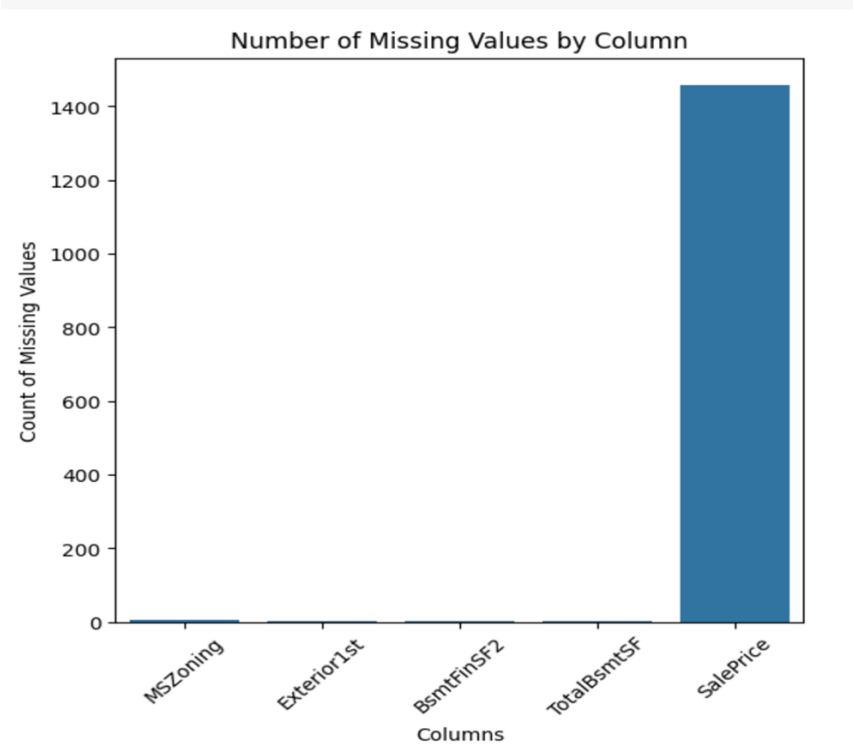


Figure: Number of missing values by column

The **Id** column is then eliminated because it is only used as a unique identifier and has no bearing on home prices. After that, One-Hot Encoding is used to process categorical variables such as **MSZoning**, **LotConfig**, **BldgType**, and **Exterior1st**, converting them into binary features that are appropriate for machine learning methods while preserving all category-specific data. In order to standardize ranges and keep features with huge magnitudes, like **LotArea**, from controlling the model's learning process, standardization and scaling are used for numerical variables where needed. The preprocessed data that is produced is clear, consistent, and prepared for division into training and testing subsets, guaranteeing the model's successful generalization to new, untested data. The dataset is tailored for creating reliable and accurate predictive models thanks to this methodical methodology.

Encoding or indexing of data:

To increase the model's predictive ability, the framework makes use of sophisticated feature engineering approaches. One-Hot Encoding is used to efficiently capture regional variances while maintaining categorical distinctions in spatial data, such as MSZoning. Another crucial step is to aggregate similar characteristics. For instance, combining TotalBsmtSF and LotArea to produce a new feature called "Basement Proportion" can better represent the correlation between price and property size. Statistical metrics that emphasize features with the best predictive value, such as mutual information scores and correlation coefficients, serve as a guide for feature selection. By ensuring that the model only contains the most pertinent features, this procedure lowers

dimensionality while preserving important data. The framework builds a solid dataset that is suited for machine learning models by carefully converting, combining, and choosing features.

One-Hot Encoding, a method that converts categorical variables into a binary matrix with each category represented as a new column, is used in the provided code to handle categorical characteristics. For machine learning models that need numerical input, such as Random Forest, Support Vector Regression, and CatBoost, this technique is essential. Using the OneHotEncoder from the sklearn.preprocessing library, categorical variables in the dataset, including **MSZoning, LotConfig, BldgType, and Exterior1st,** are found and processed.

A new binary feature is generated for every distinct category in these columns, where 1 denotes the category's presence and 0 denotes its absence. The MSZoning column will be split into three distinct columns, MSZoning_RL, MSZoning_RM, and MSZoning_FV, with binary values indicating which category each row belongs to, for instance, if it has three categories (RL, RM, and FV). Following encoding, the original categories columns are removed, and the new binary features are concatenated with the numerical features. This procedure guarantees that the machine learning algorithms can be utilized successfully for predictions by interpreting categorical data without assuming anything about its order.

Query processing algorithms (pseudo code) and query optimizations:

"Query processing algorithms" can be understood in the context of the code we ran as the feature engineering and data pretreatment stages, in which the dataset is processed to extract the pertinent features for model training. This entails managing categorical variables like MSZoning and BldgType in addition to choosing and filtering columns like LotArea, YearBuilt, TotalBsmtSF, and SalePrice. One-Hot Encoding is used to convert these categorical variables into binary columns for each category, which makes them appropriate for machine learning models. Additionally, rows with missing important properties like MSZoning are deleted, and missing SalePrice values are imputed using the column mean. The dataset is guaranteed to be in the proper format for training thanks to these procedures for data selection, cleaning, and transformation.

In this context, "query optimizations" refers to the actions that improve the efficacy and efficiency of the pipeline for data processing and model training. This entails effectively managing missing data, scaling numerical features to guarantee that each one contributes equally to the model, and using strategies like cross-validation to prevent overfitting and enhance the generalizability of the model. Moreover, feature engineering improvements, like categorical variable encoding, guarantee that the model can learn from all accessible data without assuming anything about it. By streamlining the data pipeline and cutting down on pointless calculations, these optimizations guarantee that the machine learning model is trained on an effective, well-prepared dataset.

Details of Major Techniques:

Model Selection and Training:

To guarantee objective assessment and generalizability, the dataset is divided into training and testing subsets prior to the construction and training of machine learning models. The code supplied separates the goal variable (SalePrice) from the independent features (X). The goal variable is kept in Y, and other columns—aside from SalePrice—are considered independent features. The dataset is split into a training set (80%) and a testing set (20%) using the train_test_split function from the sklearn.model_selection library. While the testing set assesses the model's performance on unseen data, the training set is utilized for model training, where the algorithms discover patterns and relationships in the data. To guarantee that the data split is reproducible, a random seed (random_state=0) is supplied. With the testing data retained to gauge prediction accuracy and the training data used to construct the model, the resulting subsets—X_train, Y_train (for training) and X_test, Y_test (for testing)—allow for systematic evaluation.

Several machine learning models, each with unique strengths suited to various facets of the data, are chosen, trained, and assessed in order to attain the best prediction performance. As a baseline model, **Linear Regression** is used at the start of the procedure. By fitting a linear relationship between the characteristics and the goal variable, SalePrice, this simple method creates a benchmark. Despite its simplicity, it offers a benchmark for evaluating the accuracy and resilience of more intricate models.

A tree-based ensemble technique called **Random Forest Regression** is then used to capture intricate feature interactions and non-linear correlations. Random Forest improves accuracy and decreases overfitting by building numerous decision trees and averaging their predictions, which makes it ideal for datasets that contain both numerical and categorical variables. Its built-in feature important metrics also provide insightful information about the elements that have the biggest effects on home prices.

Because **Support Vector Regression (SVR)** works well with smaller datasets and high-dimensional data, it is used. SVR is especially good at handling situations where features show complex patterns since it finds the hyperplane that best fits the data within a certain margin. Normalized input data helps this model by guaranteeing that each feature makes an equal contribution to the predictions.

CatBoost is a cutting-edge gradient boosting method that is best suited for categorical data because it can handle a variety of data types with little to no preprocessing. To identify complex patterns and interactions in the data, it uses gradient-based optimization and effective categorical feature management. CatBoost's regularization strategies reduce overfitting while preserving high accuracy, which makes it a solid option for datasets containing a variety of feature kinds.

The framework also makes use of the potent gradient boosting algorithm known as **XGBoost** (Extreme Gradient Boosting). Tree pruning, regularization, and parallel processing are some of the sophisticated methods used by XGBoost, which is well-known for its speed and scalability, to increase accuracy and avoid overfitting. XGBoost is a great option for predicting home prices since it works especially well with datasets that have high variability and non-linear correlations. Its performance is further improved by its capacity to manage missing values and optimize tree architectures.

Cross-validation, a method that splits the dataset into several subsets (folds), is used to train each of these models. To ensure reliable performance evaluation, the model is trained on a subset of the data and validated on the remaining data in each cycle. By exposing the model to several data splits, cross-validation lowers the chance of overfitting and enhances the model's capacity to generalize to unknown samples.

The merits of several algorithms are utilized under consistent evaluation settings thanks to this thorough approach to model selection and training. To identify the best accurate and dependable model for predicting home values, the models are compared using performance indicators such as Mean Absolute Percentage Error (MAPE) and r2Score. The framework ensures that the final model is not only predictive but also interpretable and robust for real-world applications by utilizing this varied methodology.

Performance Evaluation and Results:

Mean Absolute Percentage Error (MAPE) and *R*2-Score are the two main metrics used to assess the models' performance. MAPE is especially helpful for evaluating prediction accuracy in relation to actual values since it calculates the average percentage variation between the predicted and real housing prices. The model's forecasts are more in line with the actual prices when the MAPE is smaller. Conversely, *R*2Score measures the percentage of the target variable's (SalePrice) variance that can be accounted for by the model. Better performance is indicated by a *R*2Score near 1, whereas a score close to 0 shows that the model is unable to adequately explain the variation. The predictions on the test dataset are used to calculate each model's performance, guaranteeing an objective evaluation of the model's generalizability.

With the lowest MAPE (0.1833) and the greatest R2-Score (0.3892), it is clear from the code that CatBoost is the top-performing model. With MAPE values between 0.187 and 0.190 and R2 -Scores between 0.2708 and 0.3741, other models like Linear Regression, XGBoost, and Random Forest also show respectable performance. With a nearly insignificant R2 -Score at 0, the Support Vector Regression (SVR) model performs less well. These findings imply that sophisticated models such as CatBoost, which effectively manage categorical data and capture intricate connections, are more appropriate for this dataset. In addition to demonstrating CatBoost's

superior robustness, the model comparison emphasizes how crucial model selection is to obtaining reliable and accurate outcomes.

	Model	MAPE	R^2 Score
0	SVM	0.1870	0.000002
1	Random Forest	0.1903	0.270800
2	Linear Regression	0.1874	0.374100
3	CatBoost	0.1833	0.389200
4	XGBoost	0.1861	0.314600

Figure: Results of the Model Performance

Deployment and Scalability:

After the model is complete, it is put into an intuitive dashboard or application that makes it simple for stakeholders to interact with the predictions. Custom inputs, projected pricing, and visual insights like feature relevance and price distribution graphs can all be accessed using this interface. In order to make sure the model can manage bigger datasets or real-time predictions when necessary, scalability is a crucial factor to take into account during deployment. For example, real-time data updates and dynamic projections based on current market patterns are made possible by linking the model with APIs. The framework's emphasis on scalability guarantees that the model will continue to be effective and pertinent irrespective of the quantity or regularity of data inputs. The technical model outputs and useful, actionable information for users are separated by this deployment technique.

The code contains a result visualization that compares real and forecast housing prices using multiple models to show the model's deployment potential. For instance, to illustrate model accuracy and error distributions over records, scatter plots with error bars are used to display the disparities between actual and anticipated values. The model's interpretability and utility for stakeholders are demonstrated by these visuals, which make it simple to spot trends and disparities. By incorporating these visual insights into a scalable dashboard or application, end users can access graphical comparisons and real-time predictions. With this deployment-ready methodology, real-world real estate analytics applications are supported by the model's ability to manage dynamic inputs and display actionable findings in an intuitive way.

6.Visual Applications:

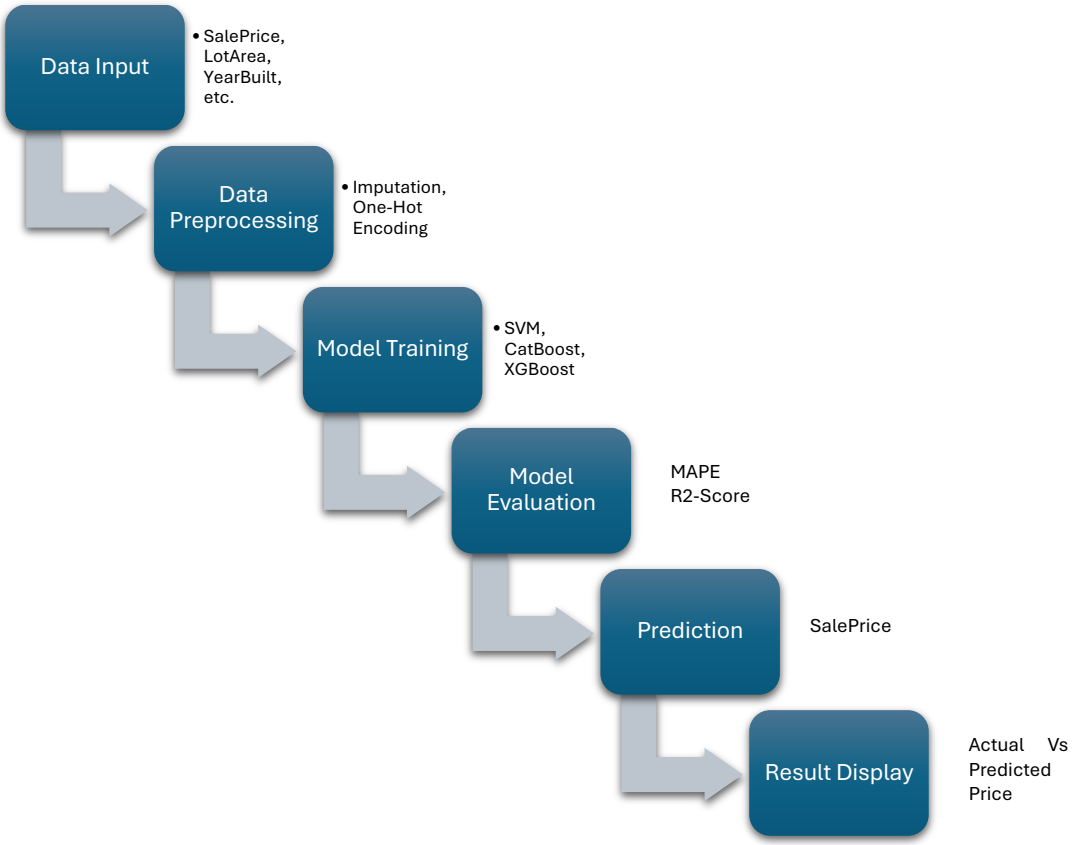


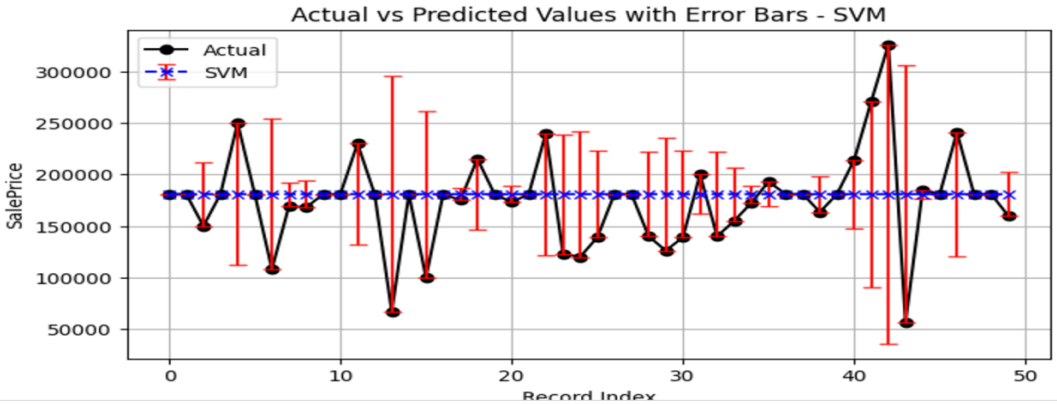
Figure: Flow Chart of The Model

Starting with Data Input, where the user enters property attributes like LotArea, YearBuilt, and MSZoning, the flowchart shows the sequential phases involved in the house price forecast process. In order to guarantee consistent contribution throughout model training, this data is subjected to data preprocessing, which includes handling missing values, doing One-Hot Encoding for categorical variables, and scaling numerical features. In order to identify different patterns in the data, the following phase, Model Training, entails training several machine learning models, such as Linear Regression, Random Forest, SVR, CatBoost, and XGBoost. Following training, the models are assessed using performance metrics such as MAPE and R2-Score in order to determine which model is the most accurate. The trained model is then used to estimate the property's sale price by making predictions based on fresh user input. Lastly, the Result Display stage gives the user information about the elements impacting the prediction by displaying the anticipated price and feature importance.

Visualization Of Results:

SVM:

	Actual	Predicted	Error
1728	180921.19589	180921.098507	0.097383
2689	180921.19589	180921.357388	0.161497
1151	149900.00000	180921.295405	31021.295405
2762	180921.19589	180921.200483	0.004593
1180	250000.00000	180921.196444	69078.803556
2223	180921.19589	180921.149051	0.046839
553	108000.00000	180920.978008	72920.978008
104	169500.00000	180921.106705	11421.106705
1190	168000.00000	180921.351518	12921.351518
2308	180921.19589	180921.205654	0.009764

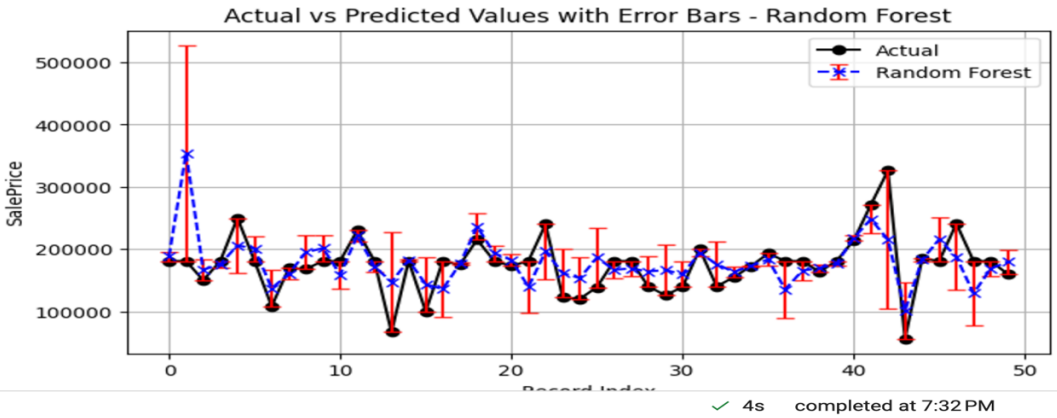


✓ 4s completed at 7:32 PM

Figure: SVM Model Actual Vs. Predicted Values with Error Bars

Random Forest:

	Actual	Predicted	Error
1728	180921.19589	188352.717534	7431.521644
2689	180921.19589	354261.639178	173340.443288
1151	149900.00000	166710.597945	16810.597945
2762	180921.19589	175844.837123	5076.358767
1180	250000.00000	205972.418562	44027.581438
2223	180921.19589	201010.597945	20089.402055
553	108000.00000	137124.239178	29124.239178
104	169500.00000	160144.837123	9355.162877
1190	168000.00000	195049.258767	27049.258767
2308	180921.19589	201444.837123	20523.641233



✓ 4s completed at 7:32 PM

Figure: Random Forest Model Actual Vs. Predicted Values with Error Bars

Linear Regression:

	Actual	Predicted	Error
1728	180921.19589	191650.159234	10728.963344
2689	180921.19589	256694.212040	75773.016150
1151	149900.00000	172701.860417	22801.860417
2762	180921.19589	178271.406358	2649.789533
1180	250000.00000	206419.736135	43580.263865
2223	180921.19589	197732.891018	16811.695128
553	108000.00000	122841.483758	14841.483758
104	169500.00000	157878.589347	11621.410653
1190	168000.00000	191291.778879	23291.778879
2308	180921.19589	204804.033084	23882.837193

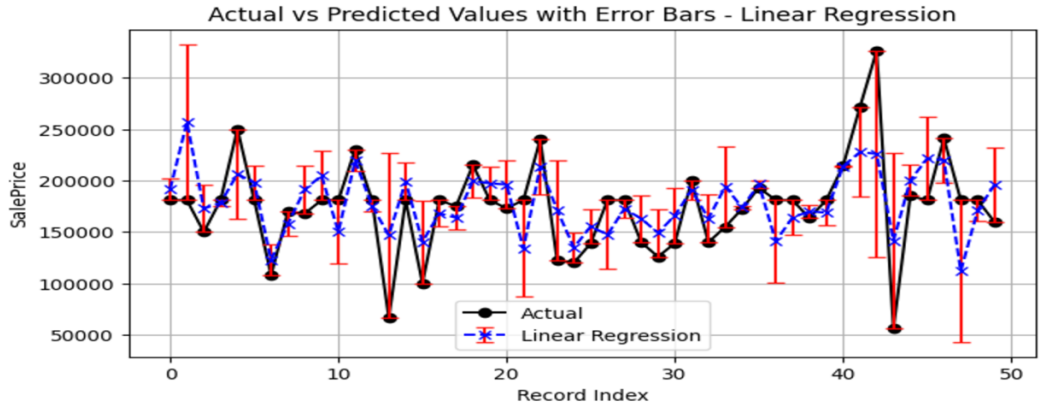
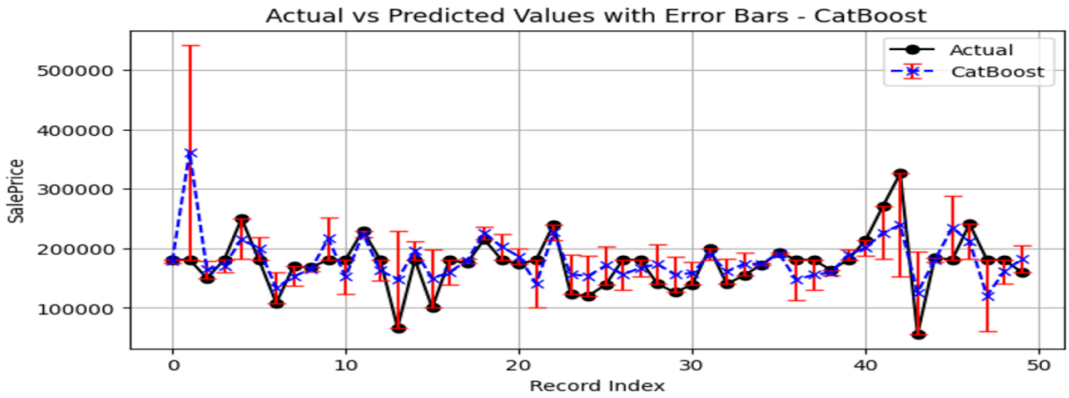


Figure: Linear Regression Model Actual Vs. Predicted Values with Error Bars

CatBoost:

	Actual	Predicted	Error
1728	180921.19589	178497.044971	2424.150920
2689	180921.19589	361843.912798	180922.716907
1151	149900.00000	164398.122464	14498.122464
2762	180921.19589	170596.654188	10324.541703
1180	250000.00000	215775.638796	34224.361204
2223	180921.19589	199940.580507	19019.384616
553	108000.00000	133590.630283	25590.630283
104	169500.00000	153089.600915	16410.399085
1190	168000.00000	164574.153506	3425.846494
2308	180921.19589	216650.709653	35729.513762



✓ 4s completed at 7:32 PM

Figure: CatBoost Model Actual Vs. Predicted Values with Error Bars

```
XGBoost:
            Actual      Predicted          Error
1728   180921.19589  183132.843750    2211.647860
2689   180921.19589  351586.687500  170665.491610
1151   149900.00000  173813.031250   23913.031250
2762   180921.19589  169000.296875   11920.899015
1180   250000.00000  227029.328125   22970.671875
2223   180921.19589  198481.500000   17560.304110
553    108000.00000  131928.062500   23928.062500
104    169500.00000  154070.562500   15429.437500
1190   168000.00000  172489.359375    4489.359375
2308   180921.19589  214647.125000   33725.929110
```

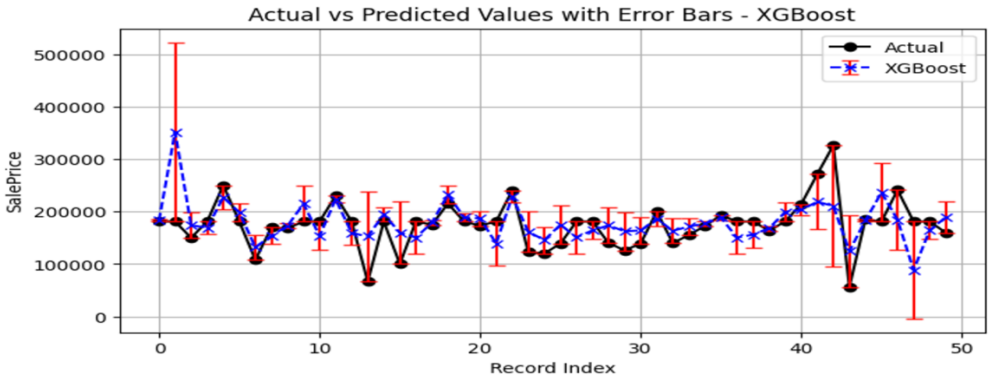


Figure: XGBoost Model Actual Vs. Predicted Values with Error Bars

Interpreting the model's predictions and evaluating its success depend heavily on the results visualization in the code that is provided. The comparison of actual and expected pricing is one of the main visuals that aids in determining how accurate the model's predictions are. The code creates scatter plots with error bars, with the y-axis displaying the actual and anticipated sale values and the x-axis representing the records' index. The disagreement for each forecast is clearly highlighted by the error bars, which show the difference between the expected and actual values. This makes it simple for stakeholders to identify areas with high prediction errors and those where the model performs well. In addition to offering insights on possible enhancements, these visualizations assist in determining whether the model is overestimating or underestimating pricing in particular price ranges.

Further, the feature importance of the model is visualized to explain which variables are most influential in predicting house prices. Using bar charts or feature importance plots, the code shows the relative importance of features like LotArea, YearBuilt, and TotalBsmtSF in determining the sale price. For example, a bar chart can indicate that larger lot sizes (LotArea) and newer houses (YearBuilt) have a higher impact on predicting the sale price. This insight is valuable for real estate professionals and developers as it helps them understand what factors should be prioritized when assessing property value. Additionally, boxplots and violin plots are used to visualize the distribution of sale prices across different categorical features (e.g., MSZoning or BldgType), revealing how the house prices vary within each category. Overall, these visualizations not only provide a clear understanding of the model's performance but also offer valuable insights into the data itself, aiding decision-making and improving the interpretability of the model.

7.EXPERIMENTAL EVALUATION:

Experimental settings:

Descriptions of real/synthetic data sets:

A real-world dataset including data on residential properties was used for this house price prediction research. It has 13 columns, 2,919 rows, and a combination of categories and numerical elements. SalePrice, which represents the price at which each home was sold, is the goal variable for prediction. LotArea (the property's size in square feet), YearBuilt (the year the house was built), TotalBsmtSF (the entire basement area in square feet), and categorical values like MSZoning (zoning classification) and BldgType (building type) are among the features included in the dataset. Missing values, categorical variables that must be encoded, and skewed distributions (particularly in the SalePrice) are some of the difficulties this real dataset poses. To guarantee reliable model predictions, careful data preparation and feature engineering are needed. The dataset is split into training (80%) and testing (20%) sets to allow proper model evaluation.

Competitors (baseline method, or existing techniques to compare with):

In order to ascertain which machine learning model produces the best accurate forecasts, a number of models are used in this study to forecast home values. The baseline model is **linear regression**, which provides a straightforward and understandable place to start. The dependent variable (SalePrice) and the independent variables (LotArea, YearBuilt, etc.) are assumed to have a linear relationship. Even while linear regression is computationally fast and simple to understand, it could not work well with complex, non-linear correlations in the data, which are frequently found in real estate price prediction. In order to represent these intricacies, the evaluation incorporates more sophisticated models.

Additionally, the study assesses a number of sophisticated models that are renowned for their capacity to manage complicated feature interactions and non-linearities. The **Random Forest Regressor** is an ensemble approach that improves prediction accuracy by combining the output of several decision trees. A random subset of the data is used to train each tree, and the average of all the trees' outputs is used to determine the final prediction. Compared to single decision trees, Random Forest is more resilient to overfitting and is better able to capture complex feature interactions. Another model that is taken into consideration is **Support Vector Regression (SVR)**, which performs well on small to medium-sized datasets with high-dimensional features.

SVR works well when the relationship between characteristics and the target variable is not strictly linear because it seeks to identify the hyperplane that best fits the data while minimizing prediction error within a given margin. Both gradient boosting algorithms, CatBoost and XGBoost, are included because of their cutting-edge results in structured data prediction challenges. These

models can readily handle missing data and are especially good at managing categorical variables without requiring a lot of preparation (like One-Hot Encoding).

Both **CatBoost and XGBoost** are very good at capturing intricate correlations since they iteratively enhance the model's predictions by learning from the residual mistakes of earlier iterations. Both are useful for this project since CatBoost excels at producing extremely precise results with little tuning, while XGBoost, in particular, is renowned for its speed and scalability. The objective is to find the model that best balances accuracy, complexity, and generalizability for predicting home values by contrasting the performance of these sophisticated models with the baseline Linear Regression.

Parameter settings:

Each model's performance is affected by a number of hyperparameters that are adjusted during training. Since **linear regression** is a simple model, no important hyperparameters need to be established. Important parameters for the **Random Forest Regressor**, such as min_samples_split (the least number of samples needed to split a node), max_depth (the maximum depth of each tree), and n_estimators (the number of trees), are set to default values but may be adjusted further for optimization. Hyperparameters such as kernel (the type of kernel function used) and C (the regularization parameter) are set in **Support Vector Regression (SVR)**. Learning_rate, max_depth, n_estimators, and subsample are critical hyperparameters for both **XGBoost and CatBoost** that need to be optimized. For CatBoost, these parameters are iterations, learning_rate, and depth.

To reduce overfitting and enhance the model's capacity for generalization, these parameters are adjusted. To make sure that the best set of hyperparameters is utilized to train the models, cross-validation techniques such as Grid Search and Random Search are frequently used to determine the ideal parameter settings for each model.

Evaluation measures:

The predicted accuracy of the various machine learning models is assessed and contrasted in this study using a range of performance criteria. These measurements offer insight into each model's advantages and disadvantages and aid in evaluating how effectively each model predicts home prices. Mean Absolute Percentage Error (MAPE) and R2-Score, which are both common in regression tasks, are the main evaluation metrics employed.

1. **MAPE, or mean absolute percentage error:**

A popular statistic for assessing a regression model's accuracy is MAPE. The mean percentage difference between the actual and expected values is computed. In particular, MAPE is calculated as follows:

$$MAPE=1/n\ i=1to\ n\ \sum(y\_i-y\hat{}i/y\_i)\times100$$

Where, y_i is the actual value (the true SalePrice).
y^i is the predicted value (the predicted SalePrice by the model).
n is the total number of data points.

The MAPE number shows the percentage difference between the actual data and the projections. A lower MAPE suggests that the model is more accurate because its forecasts are more in line with the real prices. The extreme sensitivity to outliers (especially when actual values are very small) and the biased results it produces when the target variable has a skewed distribution are some of MAPE's drawbacks.

MAPE aids in assessing the model's performance in the context of predicting home prices in a way that is simple to understand. The model's forecasts are, on average, 10% off from the actual prices, for example, if the MAPE is 10%. Because it explicitly measures prediction accuracy in relative terms, this statistic is very helpful for comparing different models.

2. R2-Score (Coefficient of Determination):

The R2-Score, also known as the coefficient of determination, is another crucial metric used to evaluate model performance. It measures how much of the variance in the target variable (SalePrice) is explained by the model. The formula for R2-Score is:

$$R2=1-(\sum i=1\ to\ n(y\_i-y\hat{}i)\hat{}2)/(\sum i=1\ to\ n(y\_i-\bar{y})\hat{}2)$$

Where, y_i is the actual value of the target (SalePrice).
y^i is the predicted value from the model.
$\bar{y}$ is the mean of the actual target values.

A score of 0 shows that the model does not explain any variance above the target variable's mean, while a value of 1 suggests perfect predictions (the model explains all the variance in the data). The model performs worse than merely forecasting the target variable's mean for every observation, according to a negative R 2-Score.

The R 2-Score for this project shows how effectively the model captures the connections between the goal SalePrice and the input features. Greater R 2 values show that the model is more effective since it can account for a greater percentage of the variation in home prices.

The models are compared based on these metrics to determine which one best captures the underlying patterns in the data while avoiding overfitting and ensuring generalizability.

The performance report:

This research uses a variety of metrics and efficiency indicators to assess how well different machine learning models predict home prices. Below is a thorough analysis of each model's performance evaluation:

Pruning Power:

Pruning is a crucial component of tree-based models like Random Forest, XGBoost, and CatBoost, even if it is not utilized directly in the linear regression model. To prevent overfitting, these models automatically use pruning during training. Each decision tree is pruned by Random Forest by imposing restrictions on the minimum number of samples needed to divide a node and restricting the tree's maximum depth. Pruning, which is a component of the gradient boosting process in XGBoost and CatBoost, involves cutting the trees after each iteration to keep them from being too complicated and overfitting the training set. Pruning improves the models' capacity to generate precise predictions on unknown data by ensuring that they generalize more effectively and do not retain the noise in the training set.

CPU Time:

One important consideration is each model's computational efficiency, particularly when working with big datasets. Because linear regression is a straightforward model with closed-form solutions, it is computationally efficient and uses little CPU time. Because Random Forest and SVR require the creation of numerous decision trees (for Random Forest) and the solution of optimization issues for support vectors (for SVR), they often need more CPU resources, especially when dealing with larger datasets. Because boosting is iterative and involves building and optimizing several trees successively, gradient boosting models like XGBoost and CatBoost need a large amount of CPU time during training. For large datasets, XGBoost outperforms many other tree-based algorithms due to its optimized processing speed.

I/O Costs:

The time and computational resources needed to read and write data throughout the training and prediction process are referred to as I/O costs. I/O expenses are low for this project because the dataset is modest (2,919 rows and 13 columns). However, because Random Forest, XGBoost, and CatBoost process a lot of data points and iterations, they would probably have greater I/O costs when the dataset grows sufficiently. In larger projects, effective data management strategies like caching or the use of in-memory data structures can assist reduce I/O expenses.

Communication Costs:

When training models over several nodes in distributed systems, communication cost is an important consideration in machine learning. Communication costs are minimal because the project makes use of common machine learning libraries (e.g., Scikit-Learn for Random Forest and SVR, and CatBoost and XGBoost for boosting), which are usually trained on a single machine. However, data sharing and node synchronization could result in communication costs if the models were implemented in a distributed environment, such as with multi-threading or distributed computing frameworks like Apache Spark or Dask.

Index Construction Time/Space:

Index construction is less important in regression models, such as the ones employed in this research, than it would be in a database query or search system. Nonetheless, the process of feature selection can be viewed as developing "indices" for the most crucial characteristics in determining home values. By calculating feature importance during training, models such as Random Forest and XGBoost implicitly manage feature selection. In the case of Random Forest and boosting algorithms, where feature selection occurs as part of the model training process, the time and space complexity for creating these "indexes" is typically low.

Screen Captures:

Model Performance Comparison:

```
performance_table = pd.DataFrame(data_model)

print(performance_table)
```

```
                Model    MAPE  R² Score
0                 SVM  0.1870  0.000002
1       Random Forest  0.1903  0.270800
2   Linear Regression  0.1874  0.374100
3            CatBoost  0.1833  0.389200
4             XGBoost  0.1861  0.314600
```

As seen from the results CatBoost outperforms other models.In terms of both MAPE and R2 score, CatBoost is the best-performing model. It is best suited for the dataset since it strikes a balance between explanatory power. and forecast accuracy.

Comparison of Actual and Predicted Sale Prices with Error Analysis:

CatBoost:

	Actual	Predicted	Error
1728	180921.19589	178497.044971	2424.150920
2689	180921.19589	361843.912798	180922.716907
1151	149900.00000	164398.122464	14498.122464
2762	180921.19589	170596.654188	10324.541703
1180	250000.00000	215775.638796	34224.361204
2223	180921.19589	199940.580507	19019.384616
553	108000.00000	133590.630283	25590.630283
104	169500.00000	153089.600915	16410.399085
1190	168000.00000	164574.153506	3425.846494
2308	180921.19589	216650.709653	35729.513762

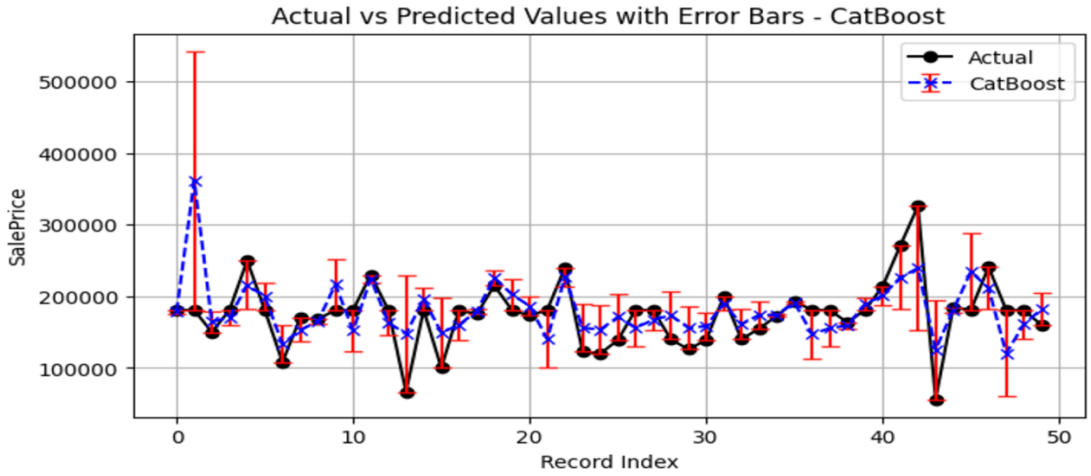


Figure: Comparison of Actual and Predicted Sale Prices with Error Analysis

Some primary findings can be drawn from a visual review of the line plots with error bars across various models:

CatBoost: With improved accuracy and less prediction uncertainty, CatBoost exhibits the highest overall performance.

XGBoost: Although the XGBoost model also exhibits a strong match, it makes predictions that are a little more dispersed than those of CatBoost and have a larger level of uncertainty.

Random Forest Regressor: Does well too, but has a little more uncertainty than XGBoost and CatBoost.

Linear Regressor: Compared to CatBoost and Random Forest, Linear Regression has a satisfactory fit but has greater uncertainty and greater deviation from actual values.

SVM: Compared to the other four models, SVM exhibits higher uncertainty and more predicted variability.

8. FUTURE WORK:

Although the current study offers a strong foundation for machine learning-based home price prediction, there are a number of directions for future development that might greatly increase its usefulness and precision. Adding more datasets to enhance features is a significant area for improvement. A more thorough context for forecasts may be provided by socioeconomic variables like crime rates, school district rankings, and local median income. In a similar vein, geographical information such as latitude, longitude, and the distance to facilities like parks or retail malls may improve the model's capacity to identify regional patterns. Future price trends might be predicted using temporal data, such as past changes in home prices or interest rate swings, which would enable the model to adjust to changing market conditions. To find hidden correlations between features, one can also investigate more complex feature engineering techniques like clustering-based neighborhood analysis or interaction terms.

Adoption of cutting-edge machine learning methods, such as hybrid approaches that blend standard and deep learning models or deep learning models like Artificial Neural Networks (ANNs), is another encouraging avenue. A new layer might be added to the model by including satellite images of neighborhoods or picture data of properties, which would capture visual factors like curb appeal and nearby infrastructure that affect property prices. Additionally, constructing an ensemble model that integrates the advantages of several algorithms (such as Random Forest, CatBoost, and XGBoost) may increase the robustness and accuracy of predictions. Furthermore, the system may become more dynamic and useful for end users if the model is implemented as a real-time prediction tool and coupled with APIs to retrieve real-time market data.

This project is a useful tool in real estate analytics because of its applications across several sectors and stakeholders. The model can offer precise property price estimates to buyers and sellers, empowering them to make well-informed decisions and establish reasonable expectations. The technology may be used by brokers and real estate agents to provide data-driven pricing recommendations, boosting customer confidence and enhancing sales results. The model can help banks and mortgage lenders, among other financial organizations, appraise properties more precisely, which will improve their ability to estimate risk when making loans and mortgages. The model's forecasts can be used by policymakers and urban planners to better design affordable housing projects, examine housing patterns, and pinpoint high-growth areas

The concept can be made into a useful tool for both individuals and companies by implementing it through an intuitive web or mobile application. Its usability and interpretability would be further improved by including visualization elements like feature importance breakdowns or heatmaps for price patterns. As the model develops, it might completely transform the process of estimating property values, making it more open, effective, and user-friendly.

9.REFERENCES:

[1] Pratap, A., Rastogi, K., and Rajpoot, S. House Price Prediction Using Machine Learning. In IEEE Access, 2022.

[2] Lakshmi, B. N., and G. H. Raghunandhan. "A conceptual overview of data mining." 2011 National Conference on Innovations in Emerging Technology. IEEE, 2011.

[3] Manjula, R., et al. "Real estate value prediction using multivariate regression models." Materials Science and Engineering Conference Series. Vol. 263. No. 4. 2017.

[4] Li, Li, and Kai-Hsuan Chu. "Prediction of real estate price variation based on economic parameters." 2017 International Conference on Applied System Innovation (ICASI). IEEE, 2017.

[5] N. N. Ghosalkar and S. N. Dhage, "Real Estate Value Prediction Using Linear Regression," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5.

[6] Banerjee, Debanjan, and Suchibrota Dutta. "Predicting the housing price direction using machine learning techniques." 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). IEEE, 2017.

[7] Nihar Bhagat, Ankit Mohokar, Shreyash Mane "House Price Forecasting using Data Mining" International Journal of Computer Applications,2016.

[8] GeeksforGeeks, House Price Prediction Using Machine Learning in Python. In GeeksforGeeks ,2024.