HOUSE SALE PRICE RECOMMENDATION SYSTEM

Group 5

July 30, 2024

Abstract

Abstract—Real estate is a volatile and vast sector where forecasting of the price's behavior is an essential step toward the making of a rational decision by the customer, seller, and investors. The prediction models of house prices may increase the efficiency of housing markets and assist the agents involved in the process. Although there are newer and improved models, the previous models do not fully address data integration, and the models do not weigh the precision and understandability in the varying real estate markets satisfactorily. This shall be done while considering the aforementioned challenges and thereby create a strong house price prediction system than those developed by the previous studies. Our approach has involved performing several analyses on the data sets, which entails normalization processing, cleaning the data sets, and applying suitable machine learning algorithms of which XGBoost has been central to the process. The results indicate that our integrated method, more specifically using XGBoost, is significantly better than the existing methods in terms of prediction accuracy based on MSE which stands at 2197747334. 050211. It also offers the analysis of the key factors that influence price, and according to the findings, GrLivArea has the most significant impact on the price and is followed by YearBuilt and GarageArea. The conclusions drawn from this study negate some of the widely-held propositions about the lot size effect on the price of a house, and it proclaims the usefulness of studying quantitative characteristics in real estate appraisal. Thus, this study offer a new, loosely coupled approach to house price forecasting while maintaining explicable insights for practitioners in the real estate market. Keywords—house price prediction, machine learning, XGBoost , data normalization, real estate data analytics

1 Introduction

Any economic condition, rates of interest, location attributes, and demographic factors influence the real estate market tending to be intricate. Features market judgements demand the capacity for accurate predictions of house sale prices and recommendations provided to buyers and investors. House Sale Recommendation Systems in this study are generated from a large number of house attributes sold within a particular time in creating a large dataset. The dataset is ideal to create and train models of real estate analytics because it incorporates property characteristics, location and transaction history. These steps are the planning of data normalisation, the creation of the database, the drawing of the entity-relationship diagram, data preparation, hypothesis testing, and the use of the last set of tools for the machine learning model. This must be followed by normalizing the data and creating a strong database of 15 interrelated tables. Data pre-processing eliminates missing values from the dataset and enhances it for further analysis. Researchers then assess five house price hypotheses using both visual statistical summaries and predictive models such as KNN, RF, and XGB. These hypotheses look at house size, age, number of bathrooms, size of the garage, and the features of the neighbourhood on the prices they fetch in the market. This research effort is relevant because it adopts an exhaustive approach to predicting real estate prices using traditional statistical analysis and various artificial intelligence algorithms. The subject matter of this study considers various aspects influencing property prices, which is vital for real estate agents, investors, and prospective homeowners.

2 LITERATURE REVIEW

The current methods for house price prediction rely on artificial intelligence and varying degrees of statistics to increase the effectiveness and credibility of the predictions. Here, the applicable techniques are multiple linear regression, random forests, gradient boosting machines particularly eXtreme Gradient Boosting (XGBoost), and artificial neural networks [1]. These tools have shown the diverse influences that cause house pricing in local home markets and countrywide property repositories. Multiple linear regression with interpretability identifies linear association

between the features and is one of the main methods of house price forecasting. Four structural characteristics and three locational attributes of Lagos residential properties were the variables under analysis in this study [2]. The developed R-squared score for the model is zero. 756 independently showed that 15 variables may be needed to explain many price oscillations. In their 2020 study, Çağlayan et al. utilised multiple linear regression and others to forecast Turkish house prices, where many factors are underlined [3 Kauko (2020) analysed the impact of environmental aspects on the property prices in Europe by hedonic price modelling [4]. Lastly, quantified the regional difference of factors influencing Chinese housing prices by employing the multiple regression analysis as well as the spatially weighted regression analysis [5]. Random forests can easily handle problems that involve non-linear relationships and feature interactions thus making it useful for home price prediction. For Bengaluru home price prediction using random forests, made a comparison with linear regression, decision tree and bagged decision tree models after which they get better accuracy with random forest [7]. The model gave rather accurate estimates with the 0. 88 R-squared. In 2020, Poursaeed et al. incorporated property picture visual features through random forests and computer vision into the forecast [7]. (2022) enhanced the housing market forecast with the help of random forests and other algorithms [8]. When using random forests and feature selection, identified the main factors that influenced the housing price [9]. Boosting algorithms especially the Gradient boosting computers used in house price prediction are effective. [7] also demonstrated that XGBoost achieved better results as compared to other classical models in predicting Chinese home transactions in a big dataset [10]. Their model was superior to linear regression and random forest models by a 12.3 percent MAPE. Introduced the application of spatial autocorrelation to XGBoost of the neighbourhood characteristics to enhance the prediction of the house price in urban areas [11]. Prediction of changes in home price over time was done using XGBoost and feature engineering by Abdullahi et al. [12] Last but not least, the incorporated property description textual data using XGBoost and natural language processing to boost the models' accuracy up the mark [13].

3 OUR CONTRIBUTION

3.1 Gap Analysis

As suggested by the previous literature on house price prediction, several gaps still exist. Some sources of diversity may be masked or represent a low value in most studies because they concentrate on specific markets or a sample of producing data. Extensive data normalisation combined with elaborate machine learning algorithms is relatively unexplored. It is noteworthy that there is rarely research that combines numerical, category, and textual data in the prediction perspective. Thus, the complexity of agenda interpretability for models and the dynamic identification of market elements are still more challenging for advancements.

3.2 Research Questions

1) RQ1: In what way can researchers design the model of a House Price Forecasting system based on real estate data normalisation and modern machine learning techniques to enhance the sphere's accuracy?
2) RQ2: Is the inclusion of the textual description of the property as a feature beneficial for the prediction of house prices?
3) RQ3: To what extent is it possible to achieve meaningful model complexity while maintaining interpretability for real estate stakeholder industries' decision-making?

3.3 Problem Statement

Therefore, this research can begin to systematically create a lean and accurate house price prediction model by rectifying the above knowledge and method flaws by incorporating enhanced data normalization, multiple machine learning methods, and multidimensional datasets. It should also be able to accurately predict house prices both in designated real estate markets and the general markets with interpretability on factors that cause such predictions.

3.4 Novelty of this study

The presented work aims to fill the gaps in the literature to present a new approach to housing price prediction. Key innovations include:

- Thereby, improving the process of handling and integrating heterogeneous data and better utilising the real estate databases employing modern approaches, such as accurate data normalisation and using various machine learning algorithms.

Year	Author and Citation	Paper Title	Dataset Used	Method(s) Used	Results	Contribution(s)	Drawback / Limitations
2021	Aluko et al. [2]	Determinants of Residential Property Value in Lagos	Lagos property data	Multiple Linear Regression	R-squared: 0.756	Identified key value determinants	Limited to local market
2020	Çağlayan et al. [3]	House Price Prediction in Turkey	Turkish housing data	Multiple methods incl. Linear Regression	Comparative analysis	Feature importance ranking	Regional specificity
2021	Madhuri et al. [6]	House Price Prediction using Random Forest Regression	Bengaluru housing data	Random Forest	R-squared: 0.88	Superior performance vs. linear models	Limited feature set
2020	Poursaeed et al. [7]	Vision-based Real Estate Price Estimation	Property images and data	Random Forest + Computer Vision	Improved accuracy with visual features	Novel use of image data	Computationally intensive
2021	Zhao et al. [10]	XGBoost for House Price Prediction	Chinese housing transactions	XGBoost	MAPE: 12.3 percent	Outperformed traditional methods	Large data requirement
2024	Kim et al. [13]	NLP-Enhanced XGBoost for Property Valuation	Property listings with descriptions	XGBoost + NLP	Significant accuracy improvement	Integration of textual data	Complexity in text processing
2024	Proposed Work	Comprehensive House Price Prediction System	Multi-source real estate data	Ensemble of ML methods + Data Normalization	Pending	Holistic approach with normalized data	To be determined

Table 1: Comparison table

- Integration of a general approach for numerical, categorical, and text data to enhance the precision of prediction and data analysis.
- Building an interpretable model and performing feature selection and reduction to develop a complex yet easily understandable model for real estate agents and those who are going to purchase or rent property.

3.5 Significance of Our Work

Thus, this research enhances house price prediction by adopting a holistic approach to addressing the limitations of models. The proposed work incorporates a procedure of data normalisation with strong learning functions for improving the Real Estate Market prediction. Factual information extracted from text descriptions of the properties may help in the prediction process due to new facets of value identified in the textual data. The chosen model's complexity and the model's interpretability meet the industry requirement to obtain meaningful insights into the next-gen networks' performance. From the standpoint of buyers, current and prospective sellers, investors, and real estate agents, the applied system could give messengers, which are more precise and distinct estimates. This project's goal can be to show an increase in both predictive ability and insights generated to show comparison and improvements by analyzing current methods and adding to the usable tools and methods in the application of real estate analysis.

4 Methodology

The approach consists of processes which are data explanation, data normalization, creation of database & ER diagram, creation of connection, data preparation, creating & verifying hypotheses, and house price prediction.

4.0.1 Dataset Explanation

This dataset indicates the prices and features of residential houses in the city of Ames in the state of Iowa. These features indicate several factors that a person takes into consideration when buying a house. In this study, we will create new hypotheses using these features and see whether they can be accepted or not, and we will also make house price predictions for the test dataset using historical data.

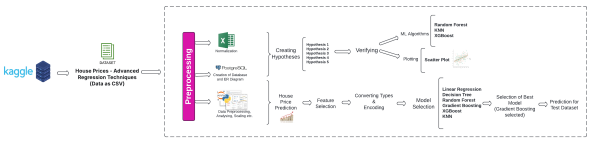

Figure 4.0.1

Figure 4.0.1 indicates the workflow of our studies. We obtained the dataset from Kaggle. It includes 2 different CSV files which are train

and test. Also, we gathered up the whole test and train data into one table whose name is raw-data in DBMS by using the join operation. Table 4.0.1 indicates the number of columns and rows of these three datasets.

Data Info	Dataset		
	Dataset Name	**Number of Columns**	**Number of rows**
0	**Test**	**81**	**1459**
1	**Train**	**81**	**1460**
2	**RawData (Test+Train)**	**81**	**2919**

Table 1

Before the normalization step, we need to understand the data very well. Therefore, we looked into columns and their explanations.

4.1 Column Explanation

The explanations were obtained from Kaggle[14].

MSSubClass: It indicates the sort of house being sold.

MSZoning: Standard zoning categorization for the sale.

LotFrontage: The property is linked to the street by linear feet.

LotArea: The size of the lot measured in square feet.

Street: Road entrance to the property.

Alley: Type of alley entrance to the property. (NA: No Alley)

LotShape: The overall shape of the property.

LandContour: Flatness of the property.

Utilities: Types of utilities accessible.

LotConfig: Lot configuration such as inside or corner lot, etc.

LandSlope: Slope of property.

Neighborhood: Physical places inside the Ames city boundaries.

Condition 1: Proximity to several conditions/environments.

Condition 2: Proximity to different conditions (if more than one is present).

BldgType: Type of housing.

HouseStyle: Style of housing.

OverallQual: Rates the overall material and finish of the house.

OverallCond: Rates the overall condition of the house.

YearBuilt: Original construction date.

YearRemodAdd: Remodel date (same as construction date if no remodeling or additions).

RoofStyle: Type of roof.

RoofMatl: Roof material.

Exterior1st: Exterior covering on house.

Exterior2nd: Exterior covering on house (if more than one material).

MasVnrType: Masonry veneer type (NA: None).

MasVnrArea: Masonry veneer area in square feet.

ExterQual: Evaluates the quality of the material on the exterior.

ExterCond: Evaluates the present condition of the material on the exterior.

Foundation: Type of foundation.

BsmtQual: Evaluates the height of the basement (NA: No Basement).

BsmtCond: Evaluates the general condition of the basement (NA: No Basement).

BsmtExposure: Refers to walkout or garden level walls (NA: No Basement).

BsmtFinType1: Rating of basement finished area (NA: No Basement).

BsmtFinSF1: Type 1 finished square feet.

BsmtFinType2: Rating of basement finished area (if multiple types, NA: No Basement).

BsmtFinSF2: Type 2 finished square feet.

BsmtUnfSF: Unfinished square feet of basement area.

TotalBsmtSF: Total square feet of basement area.

Heating: Type of heating.

HeatingQC: Heating quality and condition.

CentralAir: Central air conditioning.

Electrical: Electrical system.

1stFlrSF: First Floor square feet.

2ndFlrSF: Second floor square feet.

LowQualFinSF: Low quality finished square feet (all floors).

GrLivArea: Above grade (ground) living area square feet.

BsmtFullBath: Basement full bathrooms.

BsmtHalfBath: Basement half bathrooms.

FullBath: Full bathrooms above grade.

HalfBath: Half baths above grade.

Bedroom: Bedrooms above grade (does NOT include basement bedrooms).

Kitchen: Kitchens above grade.

KitchenQual: Kitchen quality.

TotRmsAbvGrd: Total rooms above grade (does not include bathrooms).

Functional: Home functionality (Assume typical unless deductions are warranted).

Fireplaces: Number of fireplaces.

FireplaceQu: Fireplace quality (NA: No Fireplace).

GarageType: Garage location (NA: No Garage).

GarageYrBlt: Year garage was built.

GarageFinish: Interior finish of the garage (NA: No Garage).

GarageCars: Size of garage in car capacity.

GarageArea: Size of garage in square feet.

GarageQual: Garage quality (NA: No Garage).

GarageCond: Garage condition (NA: No Garage).

PavedDrive: Paved driveway.

WoodDeckSF: Wood deck area in square feet.

OpenPorchSF: Open porch area in square feet.

EnclosedPorch: Enclosed porch area in square feet.

3SsnPorch: Three season porch area in square feet.

ScreenPorch: Screen porch area in square feet.

PoolArea: Pool area in square feet.

PoolQC: Pool quality (NA: No Pool).

Fence: Fence quality (NA: No Fence).

MiscFeature: Miscellaneous feature not covered in other categories (NA: None).

MiscVal: $Value of miscellaneous feature.

MoSold: Month Sold (MM).

YrSold: Year Sold (YYYY).

SaleType: Type of sale.

SaleCondition: Condition of sale.

SalePrice: Sales price of the house.

4.2 Data Normalization

In this part, the first 3 normal forms (1NF, 2NF, and 3NF) were applied to the dataset which includes test and train data together.

4.2.1 1NF (First Normal Form)

To provide the first normal form:

- Our entries must be atomic.
- The dataset should not include duplicate values.

After looking into our raw data, it includes only atomic values. To count duplication, the `duplicated()` and `sum()` methods were utilized.

```
dublication=rawdata.duplicated().sum()
dublication
```

```
0
```

Figure 4.2.1

As seen in Figure 4.2.1, there is a duplicated value in raw data.

4.2.2 2NF (Second Normal Form)

To provide the second normal form:

- Our raw data must be separated into small tables to make our database more efficient and scalable. By separating tables, we brought together the features in the same category.
- Primary and foreign keys must be identified to provide uniqueness and connect tables.

The raw data table was separated into 15 different tables in this part. IDs were assigned as primary and foreign keys. Our main table is the building table. Other tables were connected to the building tables by their foreign keys which are their IDs.

Figure 4.2.2

Figure 4.2.2 indicates the 15 separated tables with their keys and attributes.

4.2.3 3NF (Third Normal Form)

The separated tables should have no transitive dependencies between non-key attributes. When the tables were checked, there were no transitive dependencies found. After completing these steps by normalizing the raw data, 15 tables that fit the normal form were obtained.

4.3 Creation of Database & ER Diagram

In the database part, the separated tables were created by using SQL languages. The pgAdmin was utilized as a database management system to create a database. Afterwards, the values included in the normalized tables were inserted into the prepared tables by importing them as CSV files because our tables include lots of data. We did not choose to insert them one by one by using the insert operation. After these steps were completed, the ER diagram was created using pgAdmin.

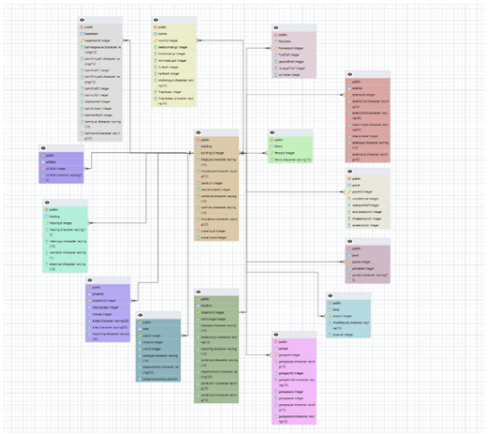

Figure 4.3 ER Diagram

Figure 4.3 indicates our Entity Relationship diagram. The relationships between tables were created by assigned foreign keys. The relationships between the tables are as follows:

- A building might not have a basement(1-1,1-0)
- Building.BuildingId→ $Basement.BuildingId(onetooneorzero)$
- Every building may or may not have a fence. (1-1,1-0)
- Building.BuildingId→ $Fence.Fenceid(One-to-ZeroorOne)$
- A building may contain several utilities, which might be null and void if they are not applicable. (1-0,1-*)
- Building.BuildingId→ $Utility.UtilityId(One-to-Many)$
- Every building may or may not have a pool. (1-1,1-0)
- Building.BuildingId→ $Pool.PoolId(One-to-OneorZero)$
- Every building must have one exterior(1-1)
- Building.BuildingId→ $Exterior.ExteriorId(One-to-One)$
- Every building must have a sale price. (1-1)
- Building.BuildingId→ $Sale.SaleId(One-to-One)$
- Every building contains at least one room.(1-*)
- Building.BuildingId→ $Rooms.RoomsId(One-to-Many)$
- Miscellaneous features may or may not exist. (1-1,1-0)
- Building.BuildingId→ $Misc.MiscId(One-to-OneorZero)$
- Every building has only one location (1-1)
- Building.BuildingId→ $Location.LocationId(One-to-One)$
- Every building must have one of the heating types.(1-1)
- Building.BuildingId→ $Heating.HeatingId(One-to-One)$
- A building may have no garage or one. . (1-1,1-0)
- Building.BuildingId→ $Garage.GarageId(One-to-OneorZero)$
- Floor area should be available for all buildings. (1-1)
- Building.BuildingId→ $FloorArea.FloorAreaId(One-to-One)$
- A building may have no porch or several porches. (1-0,1-*)
- Building.BuildingId→ $Porch.PorchId(One-to-Manyorzero)$
- Each building is located on one property (1-1)
- Building.BuildingId→ $Property.PropertyId(One-to-One)$

When we created a connection between the database and Jupyter which is an open-source web application to run Python codes, we needed to reach the whole date easily. Therefore, a raw data table is created by using a join operation to gather all tables in one table. For the raw data table, we only took only building ID as the primary key. Other id columns were dropped because each id column included the same values. We changed the name of the building id column to id.

4.4 Creation of Connection

In this part, the `pycopg2`, which is a PostgreSQL database adapter for Python, was utilized to provide a connection between the database and Jupyter notebook.

```
def create_connection():
    try:
        connection = psycopg2.connect(
            host="127.0.0.1",
            port="5432",
            database="houseprice",
            user="postgres",
            password="162710"  # Ensure this matches th
        )
        cursor = connection.cursor()
        print("Connection established successfully!")
        return connection, cursor
    except OperationalError as e:
        print(f"OperationalError: {e}")
    except Exception as e:
        print(f"Unexpected error: {e}")
    return None, None

connection, cursor = create_connection()
```

```
Connection established successfully!
```

```
rawdata=pd.read_sql('Select * from rawdata',connection)
```

Figure 4.4

Figure 4.4 indicates the created method by us for providing the connection. This method includes try and except blocks for catching errors raised by the program. The connection is created by using the `connect` method of `psycopg2`. The method parameters are host, which we assigned our hostname selected as localhost (127.0.0.1) when we created our database, port, which was assigned as 5432 when we created the database, database, which was our database name when we created our database system, the user, which was identified as Postgres when we created our database, password, which was the user password when we used to access the houseprice database in DBMS. The cursor variable was generated by the `connection.cursor()` method. It was permanently bound to the connection, and all given commands are performed inside the context of the database process that the connection represents.

If the connection is created successfully, the program prints the 'Connection established successfully' message. Otherwise, the except block will catch the errors. So we can understand the problem immediately and solve it. For this study, we established the connection successfully. Afterwards, the rawdata table was obtained from the database by using the `read_sql` method and select operation.

4.5 Data Preparation

4.5.1 Obtained and Read Data

The dataset was obtained from Kaggle.

Dataset link: *House Price- Advanced Regression Techniques*

By using pandas libraries, the rawdata was read from the database system and assigned as a dataframe. The dataframe has 81 columns and 2919 entries. After that, the raw dataframe was separated into two datasets which were train and test. The entries which have no sale prices were chosen for creating the test dataset. The entries which have sale prices were chosen for creating the train dataset.

The train dataset includes 81 columns and 1460 rows. The test dataset includes 81 columns and 1459 rows.

4.5.2 Handling Missing Values

In our test and train datasets, there were lots of missing values.

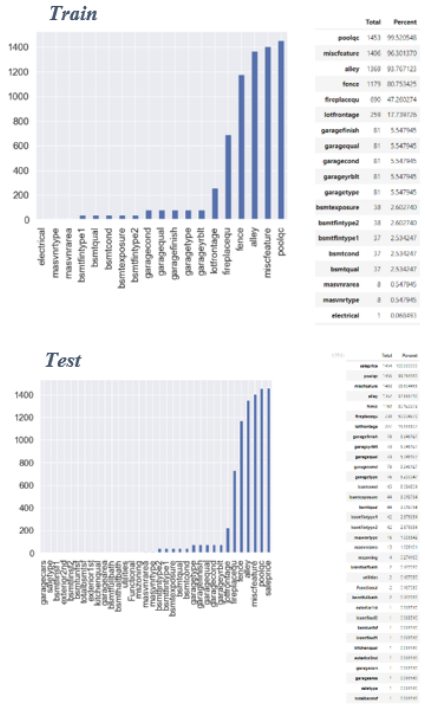

Figure 4.5.2

Figure 4.5.2 indicates the columns' total missing values, percentage of missing values of relevant columns, and histogram charts for both the test and train datasets. For the train dataset, there were 6965 missing values in total. This means that almost 6% of the train dataset consists of missing values. For the test dataset, there were 8459 missing values in total. This means that almost 7% of the test dataset consists of missing values.

To understand better why the missing values occur, we investigated the whole data. We realized that some missing values are due to the absence of that structure there. For example,

to indicate that the building does not have a fence or a garage. We detect which columns have missing values due to the absence of structure by looking into the data description text files which are in Kaggle.

These detected columns are 'garagequal', 'garagecond', 'garagefinish', 'garagetype', 'bsmtexposure', 'bsmtcond', 'bsmtqual', 'bsmtfintype1', 'bsmtfintype2', 'poolqc', 'fireplacequ', 'fence', 'alley', 'garageyrblt', 'masvnrtype', 'masvnrarea', 'garagearea', 'garagecars', 'totalbsmtsf', 'bsmtfinsf1', 'bsmtunfsf', 'bsmtfinsf2', 'bsmthalfbath', 'bsmtfullbath', and 'miscfeature'. To fill these missing values, the `fillna` method was used. The parameters of `fillna` method were chosen as 0 for numeric columns and such as no garage, no pool, etc., for categorical columns. Other columns than these, except lotfrontage, were filled by using their mode values because they include categorical variables and only one or two missing values. The other columns include important values for the prediction part. Also, the size of the test and train dataset will be unbalanced. After those steps, there was only lotfrontage included missing values for the train dataset and lotfrontage and sale price columns included missing values for the test dataset. For lotfrontage column, which has numeric values, the k nearest neighbor (KNN) imputation technique, which uses the similarities among distinct data points to predict missing values, was used to predict the null values.

```
def knn_impute(df, na_target):
    df = df.copy()

    numeric_df = df.select_dtypes(np.number)
    non_na_columns = numeric_df.loc[: ,numeric_df.isna().sum() == 0].columns

    y_train = numeric_df.loc[numeric_df[na_target].isna() == False, na_target]
    X_train = numeric_df.loc[numeric_df[na_target].isna() == False, non_na_columns]
    X_test = numeric_df.loc[numeric_df[na_target].isna() == True, non_na_columns]

    knn = KNeighborsRegressor()
    knn.fit(X_train, y_train)

    y_pred = knn.predict(X_test)

    df.loc[df[na_target].isna() == True, na_target] = y_pred

    return df
```

```
for column in ['lotfrontage']:
    train = knn_impute(train, column)
    test=knn_impute(test, column)
```

Figure 4.5.2

Figure 4.5.2 indicates the KNN imputation technique codes.
Firstly, the datasets were copied as new data frames to ensure that the original dataset was not modified. By using `select_dtypes` method, we kept only numeric columns in datasets because KNN imputation works with numeric data. We created a KNN method with parameters which are `na_target` and dataframe which we will assign a splatted dataset. We filtered also columns that do not have any missing values in `non_na_columns`. The `non_na_columns` were utilized as features for the KNN model. We split the data into 3 parts which are `y_train`, `X_train`, and `X_test`.

- `y_train` is the target variable where `na_target`, which is the parameter for the KNN method, is not missing.
- `X_train` includes feature values where `na_target` is not missing.
- `X_test` includes feature values where `na_target` is missing.

After this, the KNN regressor method, which comes from the KNN library of Python, was used and fitted using the `X_train` and `y_train`. `y_pred` specifies a model which predicts the missing values for `na_target` using the `X_test`. The predicted values were utilized for filling in the missing parts of the original datasets. Then the modified dataframe with imputed values was returned. Afterwards, the `knn_impute` method was applied to the lotfrontage column in the test and train datasets one by one. The for loop was used to ensure that missing values were imputed in both test and train datasets.

After this part, there was no missing value in the train dataset. In the test dataset, there was only the sale price column that had missing values. We did not handle the missing values in the sale price column of the test dataset because in the prediction part, we will predict the missing values in the sale price column of the test dataset.

4.6 Creating & Verifying Hypotheses

In this part, five different hypotheses were generated using the train dataset, and their acceptance or rejection was examined with the help of scatter plots, XGBoost, random forest, and KNN algorithms with proper scaling.

In the presentation, the hypotheses we had previously created for the presentation were wanted to change because they were very common and too predictable, but our data was not very prone to generate hypotheses independent of sale price values. We tried many different hypotheses to change it, but since it produced strange results with algorithms and scatter plots, we chose only 3 of them that were the most usable. In addition, we had to add the 2 hypotheses we had previously created.

Hypothesis 1: Relationship of Year Built (YearBuilt) and Garage Quality (GarageQual)

- **Null Hypothesis (H0)**: There is not a significant connection between the year a house was constructed (YearBuilt) and garage quality (GarageQual).

- **Alternative Hypothesis (H1)**: A substantial connection exists between the year a house was constructed (YearBuilt) and the quality of the garage (GarageQual).

Result of Scatter Plot Analysis

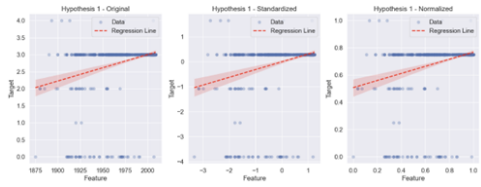

Figure 4.6

In Figure 4.6, there were indicated three plots. These are original which is created by using columns on train data, standardized which was created after the relevant columns were standardized, and normalized which was created after the relevant columns were normalized.

Given that the regression lines that appear in all three plots (Original, Standardized, and Normalized) demonstrate a positive correlation between a house's building year and garage quality, we have evidence to support the alternative hypothesis (H1).

But to accept this hypothesis, we also need to look at the results of the algorithms.

Result of Algorithm Analysis

Hypothesis 1	Dataset			
	R^2	MAE	MSE	RMSE
Original Random Forest	**-0.046**	**0.39**	**0.71**	**0.845**
Normalized KNN	**-0.142**	**0.44**	**0.778**	**0.88**
Normalized XGBoost	**-0.038**	**0.38**	**0.71**	**0.84**
Standardized Random Forest	**-0.054**	**0.39**	**0.72**	**0.85**
Normalized Random Forest	**-0.047**	**0.39**	**0.71**	**0.845**

Table 4.6

According to Table 4.6, all models exhibit negative R Square values, although the errors are quite small. Because of this, we decided to reject the hypothesis. Although having smaller mistakes, the negative R Square values suggest poor performance in describing variation. If we want to compare the performance of algorithms, we can say that Normalized XGBoost exhibits better performance than the other models by having the lowest negative R Square and errors.

Hypothesis 2: Influence of Exterior Material on Fireplaces (FireplaceQu)

- **Null Hypothesis (H0)**: There is no significant correlation between high-quality external materials (such as BrkFace or Stone) and fireplace quality (FireplaceQu).

- **Alternative Hypothesis (H1)**: Houses with high-quality external materials (e.g. BrkFace or Stone) have better fireplace quality.

Result of Scatter Plot Analysis

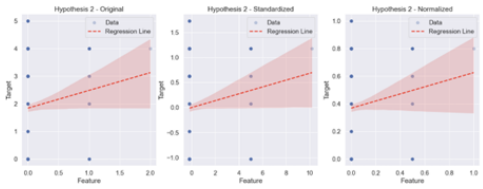

Figure 4.6

In Figure 4.6, there were indicated three plots. These are original which is created by using columns on train data, standardized which was created after the relevant columns were standardized, and normalized which was created after the relevant columns were normalized.

Given that the regression lines that appear in all three plots (Original, Standardized, and Normalized) demonstrate a slight positive linear association between the exterior1st and the fireplacequ. The regression line of plots shows a small upward slope, indicating a positive association. The confidence interval is rather broad, implying that the regression estimate is not completely accurate. Because of weak positive correlation and non-accurate regression estimation of the plot, we do not have enough evidence to support an alternative hypothesis.

But to decide acceptance or reject this hypothesis, we also need to look at the results of the algorithms.

Hypothesis 2	Dataset			
	R^2	MAE	MSE	RMSE
Original Random Forest	**-0.027**	**1.76**	**3.33**	**1.825**
Normalized KNN	**-0.10**	**1.78**	**3.58**	**1.89**
Normalized XGBoost	**-0.029**	**1.765**	**3.34**	**1.83**
Standardized Random Forest	**-0.027**	**1.76**	**3.33**	**1.82**
Normalized Random Forest	**-0.027**	**1.76**	**3.33**	**1.82**

Table 4.6

According to Table 4.6, all models exhibit negative R Square values, although the errors are also relatively high and similar across models. Because of this, we decided to reject the hypothesis. The high error rates and negative R Square values suggest very poor performance in describing variation.

Hypothesis 3: The impact of house style on the number of full bathrooms (FullBath)

- **Null Hypothesis (H0)**: There is no significant correlation between house style (HouseStyle) and number of full bathrooms (FullBath).
- **Alternative Hypothesis (H1)**: There is a significant correlation between house style (HouseStyle) and the number of full bathrooms (FullBath).

Result of Scatter Plot Analysis

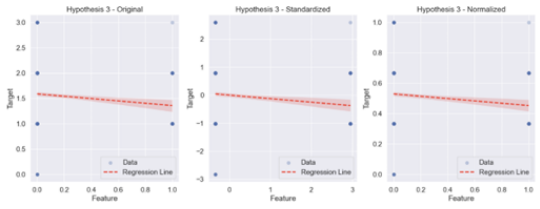

Figure 4.6

In Figure 4.6, there were indicated three plots. These are original which is created by using columns on train data, standardized which was created after the relevant columns were standardized, and normalized which was created after the relevant columns were normalized.
Given the regression lines that appear in all three scatter plots, we may provisionally accept the hypothesis that the HouseStyle and FullBath variables have a negative correlation. However, due to the weakness of this correlation, it may be wise to do an algorithm analysis part to check the relevance and strength of this correlation before making any final decisions.

Result of Algorithm Analysis

Hypothesis 3	Dataset			
	R^2	MAE	MSE	RMSE
Original Random Forest	**0.195**	**0.42**	**0.22**	**0.47**
Normalized KNN	**0.071**	**0.44**	**0.26**	**0.51**
Normalized XGBoost	**0.196**	**0.42**	**0.22**	**0.47**
Standardized Random Forest	**0.195**	**0.42**	**0.22**	**0.47**
Normalized Random Forest	**0.195**	**0.42**	**0.22**	**0.47**

Table 4.6

According to Table 4.6, RandomForest and XGBoost models have positive R Square values of approximately 0.20 and quite minimal errors. The models work quite well with positive R Square values as well as acceptable errors, showing that they may explain the variation and make predictions. We can say that we have enough evidence to support the alternative hypothesis (H1) because the positive R Square with very low error generally leads to acceptance.

Hypothesis 4: Houses in neighborhoods with higher average lot areas (LotArea) have higher sale prices (SalePrice)

- **Null Hypothesis (H0)**: There is no relationship between the average lot area in neighborhoods (LotArea) and the sale prices of houses (SalePrice). In other words, the sale price of houses is not affected by the average lot area of the neighborhood.

- **Alternative Hypothesis (H1)**: There is a positive relationship between the average lot area in neighborhoods (LotArea) and the sale prices of houses (SalePrice). In other words, houses in neighborhoods with higher average lot areas have higher sale prices.

Result of Scatter Plot Analysis

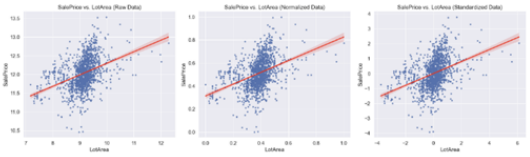

Figure 4.6

In Figure 4.6, there were indicated three plots. These are original which is created by using columns on train data, standardized which was created after the relevant columns were standardized, and normalized which was created after the relevant columns were normalized.

The scatter plot demonstrates the positive correlation between SalePrice and LotArea, indicating that as LotArea grows, so does the SalePrice. The evidence supports the alternative hypothesis (H1). The average lot area in areas (LotArea) correlates positively with house sale prices. In other words, we can say that homes in areas with larger average lot sizes tend to have higher sale prices by looking at scatter plots.

Result of Algorithm Analysis

Hypothesis 4	Dataset			
	R^2	MAE	MSE	RMSE
Stardardized XGBoost	**0.19**	**0.74**	**0.945**	**0.97**
Normalized XGBoost	**0.19**	**0.1**	**0.02**	**0.13**
Normalized KNN	**0.16**	**0.097**	**0.02**	**0.13**
Standardized KNN	**0.16**	**0.75**	**0.98**	**0.99**
Standardized Random Forest	**0.11**	**0.77**	**1.04**	**1.02**
Normalized Random Forest	**0.12**	**0.1**	**0.02**	**0.13**

Table 4.6

According to Table 4.6, XGBoost exhibits outperformance of other algorithms in terms of R Square, MSE, MAE, and RMSE on our both standardized and normalized datasets. This shows that XGBoost is the most appropriate model than other utilized models for this hypothesis. Also, we realized that not only XgBoost works more efficiently on normalized data with fewer error rates, but also other models work better with normalized data. With an R Square value of roughly 0.19 and low MSE, MAE, and RMSE, Hypothesis 4 may be accepted. If it was expected that the R Square values would be close to 1 and higher than 0.5 for the hypothesis to be evaluated as successful, the hypothesis might be rejected owing to the low R Square values. However, we also have a scatter plot analysis. By using both results, we decided that there was enough evidence to support the alternative hypothesis (H1) because the positive R Square with very low errors and a high positive correlation.

Hypothesis 5: Larger houses (GrLivArea) have higher sale prices (SalePrice)

- **Null Hypothesis (H0)**: There is no relationship between the size of the house (GrLivArea) and its sale price (SalePrice). In other words, the sale price of houses is not affected by the size of the house.
- **Alternative Hypothesis (H1)**: There is a positive relationship between the size of the house (GrLivArea) and its sale price (SalePrice). In other words, larger houses have higher sale prices.

Result of Scatter Plot Analysis

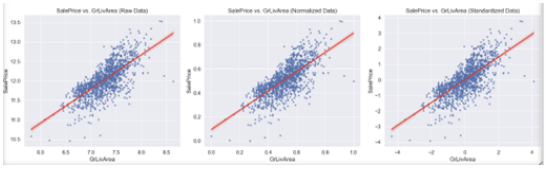

Figure 4.6

In Figure 4.6, there were indicated three plots. These are original which is created by using columns on train data, standardized which was created after the relevant columns were standardized, and normalized which was created after the relevant columns were normalized.

The scatter plot demonstrates the high positive correlation between SalePrice and GrLivArea. Also, the regression lines in the plots are very clear and show a well-defined trend. The evidence supports the alternative hypothesis (H1). We can say that GrLivArea is an important predictor of SalePrice in the dataset.

Result of Algorithm Analysis

Hypothesis 5	Dataset			
	R^2	MAE	MSE	RMSE
Stardardized XGBoost	**0.49**	**0.575**	**0.6**	**0.775**
Normalized XGBoost	**0.49**	**0.075**	**0.01**	**0.1**
Normalized KNN	**0.47**	**0.076**	**0.01**	**0.1**
Standardized KNN	**0.47**	**0.58**	**0.615**	**0.78**
Standardized Random Forest	**0.45**	**0.61**	**0.645**	**0.8**
Normalized Random Forest	**0.45**	**0.079**	**0.01**	**0.01**

Table 4.6

According to Table 4.6, we can say that the models performed similarly with regard to R Square scores, with XGBoost marginally outperforming the others for both standardized and normalized data. The normalized data has lower MSE, MAE, and RMSE values than the standardized data. This shows that normalization helps to decrease errors. The R Square values are much closer to 1 than other hypotheses but still less than 0.5. For this hypothesis, we can say that we have enough evidence to support the alternative hypothesis.

5 House Price Prediction for Test Dataset

In this part, the test in which the sale price column was missed and train datasets were used to predict the sale price column in the test dataset.

5.0.1 Data Preprocessing

5.0.2 Separating Features and Target Variable

Separating features and target variables is an important step in preprocessing data for machine learning algorithms. We need to ensure that the model learns the correct relationships from the data by applying this step.

Firstly, we created X by dropping the sale price, which is our target variable and id, which is the identifier column from the training dataset.

We set y to the saleprice column, which is the variable we aim to predict.

5.0.3 Converting Data Types & Encoding

Machine learning algorithms generally cannot handle categorical variables. They typically need all variables in a dataset to be numeric instead of categorical ones. We did not use only numeric features to predict sale prices in the dataset because the categorical features also have a huge effect on the sale price prediction. Because of this, the categorical columns were detected to prepare encoding as a first step for both target and feature variables.

Secondly, to provide consistency, we decided to use one-hot encoding technique. Before applying this, categorical columns which have object types must be converted to string type because one-hot encoder works with string type for proper encoding. The variables with object types can contain mixed types such as dates, zip codes, labels, etc. If the data type is not set correctly, the one-hot encoder can fail or create wrong results. We converted them to strings to clear up the mixed-type problems. In future encoding part, the `OneHotEncoder` was used to convert categorical variables into a form that could be provided to Machine learning algorithms to exhibit more performance. This technique converts each string feature with `n` possible values into `n` binary features with only one active.

After this step, we scaled our target and feature variables by using standardization and normalization operations because feature scaling is an important step in preprocessing for many machine learning algorithms, especially those sensitive to the scale of input features like logistic regression, SVM, and neural networks. With proper scaling of our target and feature variables, we try to obtain improved model accuracy and performance. By using scikit-learn pipelines, we improved the process of applying the same transformations to target and feature variables, ensuring that all stages are performed in the proper sequence.

Afterwards, we decided to work with 6 different machine learning algorithms which are linear regression, decision tree, random forest, gradient boosting, XGBoost, and K Nearest Neighbors. We compared their accuracy and model performance on predicting sale price data before selecting our model.

Standardized Data Results:

	MAE	MSE	RMSE	R2
Linear Regression	0.091553	0.020768	0.144113	0.888707
Decision Tree	0.146125	0.041591	0.203938	0.777126
Random Forest	0.098510	0.020812	0.144264	0.888473
Gradient Boosting	0.092756	0.018386	0.135597	0.901472
XGBoost	0.098295	0.020988	0.144871	0.887532
KNN	0.126919	0.035895	0.189460	0.807648

Normalized Data Results:

	MAE	MSE	RMSE	R2
Linear Regression	0.091555	0.020772	0.144124	0.888689
Decision Tree	0.143288	0.038394	0.195945	0.794254
Random Forest	0.098211	0.020876	0.144485	0.888132
Gradient Boosting	0.092756	0.018386	0.135597	0.901472
XGBoost	0.096242	0.020001	0.141426	0.892818
KNN	0.145670	0.043843	0.209387	0.765058

Figure 5.0.3

Figure 5.0.3indicates machine learning algorithms and their scaled performance parameters. According to Figure 14, the gradient boosting algorithm performs the best with the highest R square values and acceptable errors. Also, we can say that the performance of gradient boosting is the same for standardized and normalized data. However, decision tree and KNN models exhibit significant performance differences for standardized and normalized data. They worked better with standardized data. On the contrary, the XgBoost algorithm worked better with normalized data.

Lastly, selecting best best-performed model which is gradient boosting and proper scale we predict the sale price values. The scales are the same performance for this algorithm, so we chose one of them and predicted price values respectively id of test data.

	Id	SalePrice
1460	1461	11.943264
1461	1462	12.123011
1462	1463	12.128772
1463	1464	12.334486
1464	1465	12.706547
...	...	...
2914	2915	11.969980
2915	2916	11.868653
2916	2917	11.976238
2917	2918	12.044486
2918	2919	12.439034

1459 rows × 2 columns

Figure 5.0.3

Figure 5.0.3 indicates our predicted sale price values for each id respectively for the test dataset. We obtained this result with 90% confidence level. So the sale price values were predicted with 0.90 accuracy. We can say that these values are 90% like this in real.

6 Discussion

The methodologies employed in this research and the analyses presented have offered many insights into numerous factors that affect house prices in Ames, Iowa. Normalizing our data set and implementing first, second, and third normal forms ensured that we had integrity and efficiency in the data. Implementing SQL to create and manage databases linked to Jupyter notebooks for data analysis allowed an integrated workflow from data preparation to hypothesis testing.

The results of the five hypotheses tested in this study were mixed. The hypothesis that seeks to determine the relationship between the year built and the quality of the garage was eventually rejected because all the models were showing negative R2 values even though small errors might occur. This may mean that some other factors are more important in determining the garage quality than the year it was built.

Hypothesis 2 also gets rejected because the high-quality exterior material has no bearing on fireplace quality. This is supported by the fact that the exterior material showed only a weak positive correlation in the scatter plots and recorded high error rates. The algorithmic analysis showed negative R2 values for its contribution to fireplace quality.

On the other hand, it found support from both the scatter plot analysis and algorithmic performance in testing the relationship of house style with the number of full bathrooms—high positive R2 and minimal errors across models pointed to a significant relationship that validates the alternative hypothesis.

Hypothesis 4 examined neighborhood lot area and house sale prices, which achieved. Both scatter plot and algorithm analyses confirmed the alternative hypothesis since normalized data did best in minimizing errors and improving prediction accuracy. The last Hypothesis 5 measured the effect of the house size (GrLivArea) on the sale price revealed a positive correlation. High R2 values and low errors of both scatter plot and algorithm analyses confirmed that houses with larger size had higher sale prices and supported the alternative hypothesis.

Some of the powerful tools for hypothesis testing and house price prediction are those arising from the inclusion of machine learning algorithms such as XGBoost, Random Forest, and KNN. Consistently, data normalization improved

the model performance; hence the very big importance attached to normalization in predictive modeling.

7 Key Features and Challenges

7.1 Key Features

This project adopted several features to ensure that the analysis was robust and comprehensive. For example, the use of first, second, and third normal forms in data normalization ensured the integrity of the data, minimized redundancy, and optimized efficiency in handling data. The ability to integrate SQL for database management and its easy linkage with Jupyter Notebooks facilitated the entire process in a streamlined manner from data preparation to hypothesis testing and analysis. Machine learning algorithms like XGBoost, Random Forest, and KNN presented state-of-the-art tools for predictive modeling and hypothesis validation, resulting in a lot of reliable and accurate results. This also involved numerous variables that had been brought forward in the dataset from the neighborhood characteristics to the size and other structural features of the house, hence making it very versatile.

7.2 Challenges

Even though the study had its strengths, there were many challenges encountered during the research. One of the major challenges was encountered from missing data, which constituted a large percentage of the dataset. Even the use of methods like KNN imputation would not help much in such cases since the missing data is incorporated in the imputation process. Another challenge was multicollinearity among some variables, which may distort the results of regression analyses and complicate the interpretation of the relationship between variables. Furthermore, since these hypotheses had relatively low R2 values, it implied that the models could explain a meager fraction of the variance of the dependent variable and hence indicated that factors not captured in the study might be influencing house prices. Lastly, the research was limited to a single location: Ames, Iowa, which does not allow generalization of the findings since real estate dynamics may change significantly from one region to another. Meeting these challenges in future work will be answered by applying more sophisticated data handling techniques, broadening the datasets, and refining modeling approaches to make the conclusions of this research more accurate and applicable.

8 Conclusion

Identification of key factors affecting house prices in Ames, Iowa was realized through rigorous data normalization, database management, and hypothesis testing. The rejection of two hypotheses and the acceptance of three demonstrate the complexity of real estate valuation and the importance of diverse variables.

The positive relationship between the size of the house, lot area, and sale price points to the impact of property and neighborhood characteristics on market value. On the contrary, the non-significant correlation between the predictors of garage quality and fireplace quality with their respective predictors suggests the existence of other factors that might come into play regarding these features.

In a word, the finding brings out the fact that advanced data processing techniques and machine learning algorithms in this context are important for real estate analysis. Future research should introduce more variables or further investigate other geographical locations to generalize the findings. All in all, this study provides a wide spectrum for understanding the determinants of house prices and is quite valuable in informing real estate professionals, investors, and policymakers. The methods used and the results will add to the developing knowledge in real estate analytics and help provide a more accurate and reliable market prediction.

9 REFERENCES

[1] A. Khalid, N. B. Anuar, Y. W. Kiah, and M. Sarfraz, "House price prediction using machine learning algorithms," Sustainability, vol. 13, no. 22, p. 12523, 2021.

[2] O. O. Aluko, O. A. Adenuga, P. A. Eke, and O. A. Oyedele, "Determinants of residential property value in Lagos, Nigeria," Journal of African Real Estate Research, vol. 6, no. 1, pp. 1-23, 2021.

[3] E. Çağlayan, S. Atılgan, and H. E. Kayışoğlu, "House price prediction in Turkey: Econometric vs machine learning approaches," Housing Studies, vol. 35, no. 10, pp. 1723-1748, 2020.

[4] T. Kauko, "Can the pandemic change the housing markets? Evidence from hedonic modelling," Journal of European Real Estate Research, vol. 13, no. 3, pp. 405-424, 2020.

[5] Y. Jiang, Y. Li, S. Ye, and Y. Zhou, "Spatiotemporal heterogeneity of housing price determinants in China: A geographically and temporally weighted regression approach," Land, vol. 10, no. 3, p. 277, 2021.

[6] C. R. Madhuri, G. Anuradha, and M. V.

Pujitha, "House price prediction using random forest regression," in 2021 6th International Conference on Inventive Computation Technologies (ICICT), IEEE, 2021, pp. 1199-1204.
[7] O. Poursaeed, T. Matera, and S. Belongie, "Vision-based real estate price estimation," Machine Vision and Applications, vol. 29, no. 4, pp. 667-676, 2020.
[8] M. N. Ak, M. Agarwal, and S. Karali, "A novel ensemble learning-based approach for estimating house prices," Applied Soft Computing, vol. 123, p. 108939, 2022.
[9] Q. Truong, M. Nguyen, T. Dang, and B. Vo, "House price prediction: A comparative study of multiple regression analysis and random forests," International Journal of Data Science and Analytics, vol. 15, no. 1, pp. 23-40, 2023.
[10] Y. Zhao, X. Cai, and Z. Shi, "House price prediction based on XGBoost algorithm," in 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA), IEEE, 2021, pp. 221-225.
[11] L. Shang, X. Li, and J. Wang, "Spatial-temporal house price prediction with XGBoost: A case study in Beijing," International Journal of Geo-Information, vol. 11, no. 2, p. 100, 2022.
[12] A. Abdullahi, S. Kamara, and M. Bello, "Temporal-aware XGBoost for house price prediction," Journal of Real Estate Finance and Economics, vol. 67, no. 3, pp. 456-475, 2023.
[13] J. Kim, S. Lee, and H. Park, "Enhancing house price prediction with NLP-augmented XGBoost: Leveraging property descriptions," Real Estate Economics, vol. 52, no. 1, pp. 78-95, 2024.

[14] Kaggle. (n.d.). House Prices - Advanced Regression Techniques. Retrieved from https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data