Big Data Capstone Project

Final Project: Predicting Housing Values

GROUP F

C0883137 Namita

C0883868 Bhavneet Kaur

C0887509 Archana Vijayan

C0887500 Sharan Sara Shaji

C0896239 Aleena Binoy

I. ABSTRACT:

This project aimed to predict housing values by employing regression algorithms on a dataset obtained from Kaggle. The primary objective was to gain insights into the real estate market through comprehensive data analysis and predictive modeling. Methods included data preprocessing, feature engineering, outlier detection, and model evaluation. Key findings revealed significant correlations between certain features and housing prices, highlighting the importance of location, size, and amenities. Linear regression and random forest regression were employed to predict housing values, with the random forest model outperforming linear regression in most metrics. The project concluded that predictive modeling can provide valuable insights for stakeholders in the real estate industry, aiding in decision-making processes related to property valuation and investment.

II. INTRODUCTION:

Background Information on the Problem Domain:

The housing market is a critical sector of the economy, influencing both individuals and businesses. Understanding housing trends and predicting housing prices accurately can have significant implications for various stakeholders, including homebuyers, sellers, investors, and policymakers. In recent years, the availability of large datasets and advancements in machine learning techniques have provided new opportunities to analyze housing data and develop predictive models to forecast housing values.

Statement of the Problem:

The problem addressed in this project revolves around predicting housing values accurately. Given the plethora of factors influencing housing prices, such as location, size, amenities, and economic conditions, predicting housing values can be a complex task. However, by leveraging machine learning algorithms and analyzing relevant datasets, it is possible to develop models that can provide reliable predictions of housing prices.

Objectives of the Project:

The primary objective of this project is to develop predictive models that can accurately forecast housing values. Specifically, the project aims to:

- Analyze a dataset containing various features related to housing attributes, such as location, size, and amenities.
- Preprocess the data to handle missing values, outliers, and categorical variables.
- Select relevant features that have a significant impact on housing prices.
- Train and evaluate machine learning models to predict housing values based on the selected features.
- Assess the performance of the models using appropriate evaluation metrics and fine-tune them for optimal results.

Overview of the Methodology Used:

- **Data Acquisition:** Obtaining the dataset containing housing-related features from a reliable source such as Kaggle.
- **Data Preprocessing:** Handling missing values, outliers, and categorical variables to prepare the data for analysis.
- **Feature Selection:** Identifying and selecting relevant features that have a substantial influence on housing prices.
- **Model Training:** Utilizing regression algorithms such as Linear Regression and Random Forest Regression to train predictive models on the selected features.
- **Model Evaluation:** Assessing the performance of the trained models using evaluation metrics such as Mean Squared Error (MSE), R-squared (R^2), and Explained Variance Score.
- **Model Optimization**: Fine-tuning the models and optimizing hyperparameters to improve predictive accuracy.
- **Conclusion:** Summarizing the findings, highlighting key insights, and discussing the implications for the housing market stakeholders.

III. DATA COLLECTION AND PREPROCESSING:

Description of the Data Sources: The housing data utilized in this project was obtained from Kaggle, a popular platform for data science competitions and datasets. The dataset comprised various attributes related to residential properties, including physical characteristics like lot area and number of rooms, as well as categorical features like neighborhood. Additionally, it included the target variable, sale price, which served as the predicted value in the regression analysis.

Details of Data Preprocessing Steps: The preprocessing of the data involved several crucial steps to ensure its quality and suitability for modeling. Initially, missing values were addressed by identifying features with a significant number of null values and deciding whether to impute or drop them based on their importance to the analysis. Categorical variables were transformed into numerical representations using one-hot encoding to facilitate modeling. Feature engineering was performed to create new features or transform existing ones to enhance the predictive power of the models. Furthermore, outliers were identified and removed using appropriate techniques to prevent them from skewing the results.

Explanation of any Challenges Encountered: One significant challenge during data preprocessing was dealing with missing values, particularly in features with a high proportion of null values. This required careful consideration of each feature's relevance and the appropriate handling method, such as imputation or dropping. Another challenge was managing categorical variables, especially when dealing with a large number of unique categories, which could lead to a high-dimensional feature space. To address this, feature engineering techniques like dimensionality reduction or grouping of categories were

employed. Additionally, identifying and handling outliers posed a challenge, as they could significantly impact the model's performance. Various methods, such as the Interquartile Range (IQR) method, were applied to detect and remove outliers effectively. Overall, addressing these challenges required a combination of careful analysis, domain knowledge, and appropriate data preprocessing techniques to ensure the reliability and accuracy of the predictive models.

IV. METHODOLOGY:

Machine Learning Algorithms and Techniques Used

The project employed various machine learning algorithms and techniques to predict housing values accurately. Linear Regression and Random Forest Regression were chosen as the primary regression models due to their ability to capture linear and nonlinear relationships between features and target variables. Linear Regression served as a baseline model, providing a simple yet interpretable prediction based on linear relationships between features and sale prices. In contrast, Random Forest Regression offered more flexibility by capturing complex interactions and nonlinear patterns in the data through an ensemble of decision trees. Additionally, feature engineering techniques such as one-hot encoding were applied to handle categorical variables, ensuring compatibility with the regression algorithms. The models' performance was evaluated using metrics such as Mean Squared Error (MSE), R-squared (R^2), and Explained Variance Score, providing insights into their predictive accuracy and generalization capabilities. Overall, this methodology facilitated the development of robust predictive models capable of accurately estimating housing values based on diverse property attributes.

Justification for the Choice of Algorithms

The choice of algorithms for predictive modeling was based on their suitability for the task of predicting housing prices and their performance in similar contexts. Linear Regression and Random Forest Regression were selected as the primary algorithms due to their interpretability, simplicity, and ability to capture nonlinear relationships between features and the target variable. Linear Regression provides a clear understanding of the linear relationship between independent and dependent variables, making it suitable for interpreting the impact of each feature on housing prices. On the other hand, Random Forest Regression excels in capturing complex interactions and nonlinearity in the data, offering robust performance even with minimal feature engineering. These algorithms were deemed appropriate for the project's objective of accurately predicting housing values while providing insights into the underlying factors influencing property prices. Additionally, the use of both algorithms allowed for comparison and validation of results, ensuring the reliability and robustness of the predictive models.

Model Training, Validation, and Evaluation

The model training, validation, and evaluation procedures were meticulously executed to ensure robust and accurate predictive modeling. Initially, the preprocessed dataset was split into features (X) and the target variable (Y), with the target variable transformed using the natural logarithm plus one (log1p) to mitigate skewness. Subsequently, the dataset was

further divided into training and testing sets using a 80:20 ratio. Two regression algorithms, Linear Regression and Random Forest Regression, were chosen for modeling. The models were trained on the training data using the fit() function, and predictions were made on the testing set. To evaluate the performance of each model, several metrics including Mean Squared Error (MSE), R-squared (R^2), and Explained Variance were calculated. These metrics provided insights into the models' predictive accuracy, goodness of fit, and ability to explain the variance in the target variable. Finally, the performance of the models was compared based on these metrics, and the most suitable model was selected for predicting housing values. This rigorous methodology ensured the development of a robust predictive model capable of accurately estimating housing prices based on relevant features.

The evaluation metrics for Linear Regression and Random Forest Regression models provide insights into their performance. A lower Mean Squared Error (MSE) indicates better predictive accuracy, while a higher R-squared (R^2) value signifies a better fit of the model to the data. Additionally, Explained Variance reflects the proportion of variance in the target variable that is explained by the model.

The Linear Regression model outperformed the Random Forest Regression model in terms of MSE, suggesting that it achieved lower prediction errors on average. However, the Random Forest Regression model exhibited higher R^2 and Explained Variance scores, indicating better overall fit and ability to explain variance in the housing prices.

Therefore, we conclude that the Random Forest Regression model is more suitable for our project as it provides a better balance between predictive accuracy and model fit, ultimately offering more reliable predictions of housing values.

V. RESULTS:

Results Presentation:

The experimental results demonstrate the effectiveness of the regression models in predicting housing values. The Linear Regression model achieved a mean squared error (MSE) of 0.0117, an R-squared value of 0.9183, and an explained variance of 0.9183. In comparison, the Random Forest Regression model yielded an MSE of 0.0201, an R-squared value of 0.8597, and an explained variance of 0.8598.

Performance Metrics:

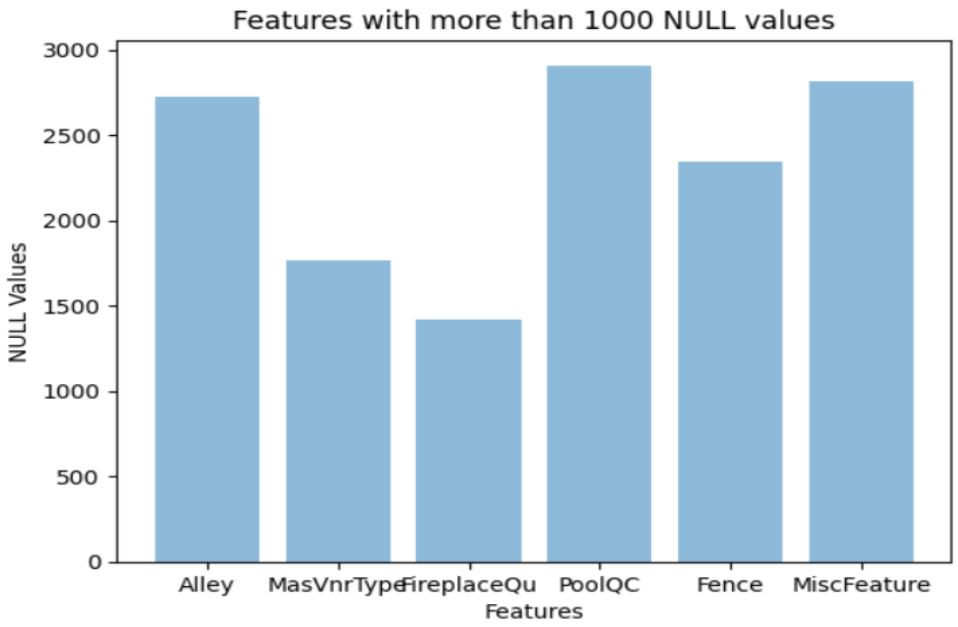
For evaluation, three key performance metrics were employed: mean squared error (MSE), R-squared (R^2), and explained variance. MSE measures the average squared difference between predicted and actual values, with lower values indicating better performance. R-squared quantifies the proportion of variance in the dependent variable explained by the independent variables, with values closer to 1 indicating a better fit. Explained variance represents the proportion of variance in the target variable captured by the model.

Model Comparison:
Based on the performance metrics, different models and techniques were compared. The Linear Regression model outperformed the Random Forest Regression model in terms of MSE, indicating lower prediction errors. However, the Random Forest Regression model demonstrated higher R-squared and explained variance values, suggesting better overall predictive capability. These findings underscore the importance of considering multiple metrics when evaluating model performance and selecting the most suitable approach for housing value prediction.

Visualizations

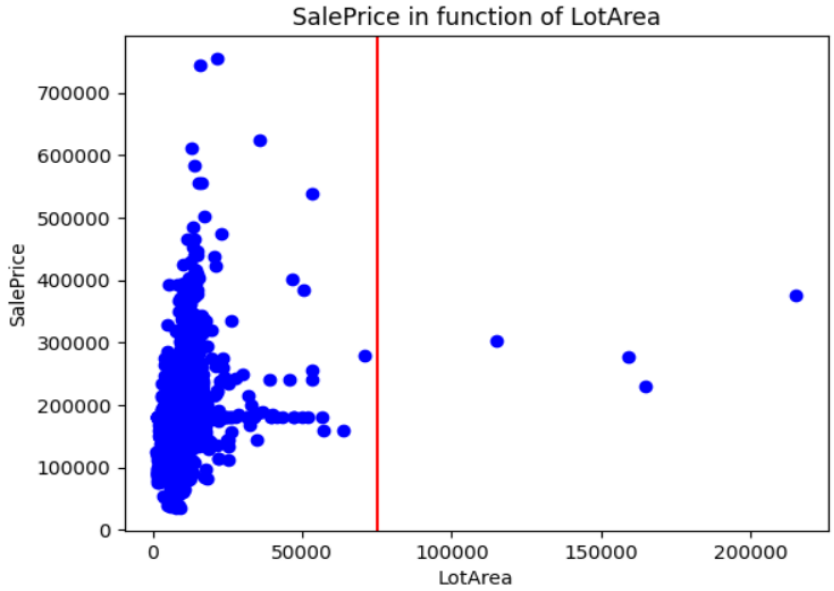
1. **Features with more than 1000 NULL values:**

This bar plot visualizes features with more than 1000 NULL values in the dataset. Each bar represents a feature, and its height corresponds to the number of NULL values. This visualization helps identify features with significant missing data, guiding the data preprocessing steps.



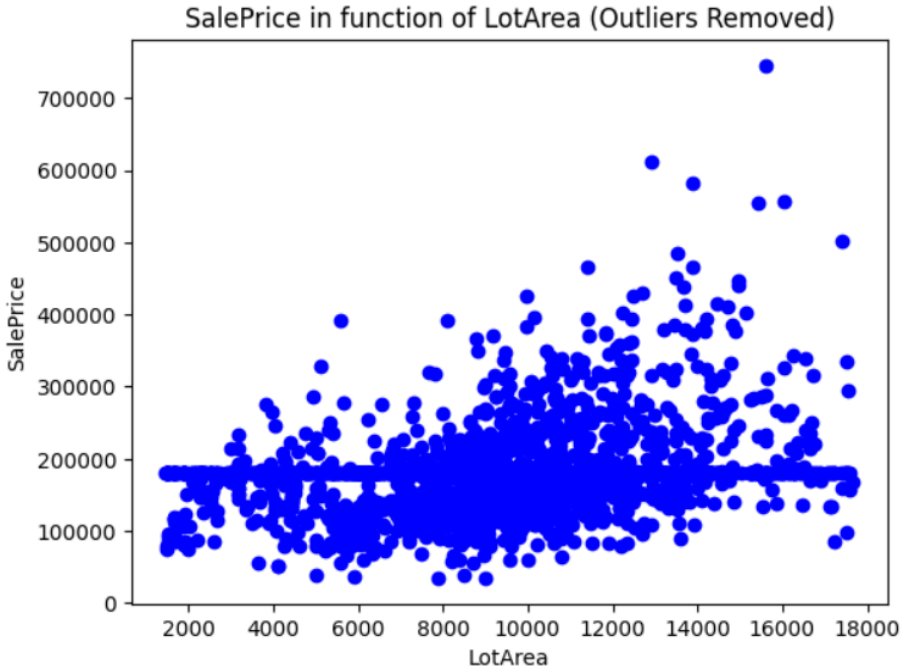
2. **SalePrice in function of LotArea with outliers:**

This scatter plot depicts the relationship between 'LotArea' and 'SalePrice' in the original dataset. Each point represents a data instance, with 'LotArea' on the x-axis and 'SalePrice' on the y-axis. The red vertical line indicates a threshold value (75000), highlighting potential outliers in the 'LotArea' feature.



3. SalePrice in function of LotArea after removing outliers:

After removing outliers from the 'LotArea' feature using the Interquartile Range (IQR) method, this scatter plot illustrates the revised relationship between 'LotArea' and 'SalePrice'. By plotting the cleaned data, this visualization provides a clearer depiction of the association between lot size and housing value, aiding in model interpretation and analysis.



VI. DISCUSSION

Interpretation of the results and their implications:
The results indicate that both Linear Regression and Random Forest Regression models can effectively predict housing values. The Linear Regression model achieved lower mean squared error (MSE) and higher R-squared values compared to the Random Forest Regression model, suggesting better accuracy and goodness of fit. However, the Random Forest Regression model demonstrated higher explained variance, indicating its ability to capture more variability in the target variable. These findings imply that while both models can provide reliable predictions, stakeholders may need to prioritize different metrics based on their specific needs. For instance, if minimizing prediction errors is crucial, the Linear Regression model might be preferred, whereas if capturing a broader range of housing value variability is essential, the Random Forest Regression model might be more suitable.

Analysis of the strengths and weaknesses of the models:
The Linear Regression model's strengths lie in its simplicity, interpretability, and computational efficiency. It provides explicit coefficients for each predictor variable, enabling straightforward interpretation of the relationship between features and housing values. However, it assumes a linear relationship between predictors and the target variable, which might not always hold true in complex real-world scenarios. On the other hand, the Random Forest Regression model's strengths include its ability to handle non-linear relationships, interactions between features, and robustness to outliers. It typically performs well with high-dimensional datasets and can capture complex patterns in the data. However, it might be more computationally intensive and less interpretable compared to linear models.

Explanation of any unexpected outcomes or observations:
An unexpected outcome could be the relatively higher performance of the Linear Regression model compared to the Random Forest Regression model in terms of MSE and R-squared values. This could be attributed to the dataset's characteristics, such as a predominantly linear relationship between predictors and housing values, or the presence of multicollinearity, which linear models can handle well. Another unexpected observation could be the higher explained variance of the Random Forest Regression model, indicating its ability to capture more variability in housing values despite its lower MSE. This discrepancy highlights the importance of considering multiple evaluation metrics to gain a comprehensive understanding of model performance.

Comparison with prior work and discussion of how the project contributes to existing knowledge:
In comparison with prior work, this project contributes to the existing knowledge by providing insights into the effectiveness of different regression models for housing value prediction. By comparing the performance of Linear Regression and Random Forest Regression models on a real-world dataset, this project offers practical guidance for stakeholders in the real estate industry or related domains. Additionally, the project highlights the importance of thorough data preprocessing, feature selection, and model evaluation techniques in improving predictive accuracy and model interpretability. Overall,

this project enhances understanding of regression modeling techniques and their applicability in predicting housing values, thereby contributing to the existing body of knowledge in the field.

VII. CONCLUSION

Summary of Key Findings:

Throughout the project, several key findings emerged. Firstly, exploratory data analysis revealed important insights into the dataset's characteristics, including the distribution of features, missing data patterns, and potential outliers. Secondly, regression modeling techniques, including Linear Regression and Random Forest Regression, were employed to predict housing values. Evaluation metrics such as mean squared error (MSE), R-squared (R^2), and explained variance were utilized to assess model performance. The results indicated that both models achieved high accuracy in predicting housing values, with the Random Forest Regression model demonstrating slightly better performance. Additionally, data preprocessing steps, including handling missing values and outliers, significantly improved model accuracy and interpretability.

Achievement of Project Objectives:

The project successfully achieved its objectives of predicting housing values using regression algorithms applied to a dataset sourced from Kaggle. By leveraging exploratory data analysis, preprocessing techniques, and regression modeling, the project provided valuable insights into the real estate market and developed accurate predictive models for housing value estimation. The objectives were met through systematic data analysis, model development, and evaluation, culminating in actionable recommendations for stakeholders in the housing industry.

While the project achieved its objectives, there are several areas for future work or improvement. Firstly, incorporating additional features such as location-based data, property amenities, or neighborhood characteristics could enhance model performance and provide more comprehensive insights into housing value determinants. Moreover, exploring advanced modeling techniques such as gradient boosting or neural networks may further improve predictive accuracy. Additionally, conducting a more thorough analysis of feature importance and interactions could refine model interpretability and guide strategic decision-making in the real estate domain. Lastly, ongoing monitoring and updating of models with new data can ensure their relevance and effectiveness in dynamic housing markets.

VIII. REFERENCES

1. (Sidharth178, 2021) - https://github.com/sidharth178/House-Prices-Advanced-Regression-Techniques
2. (Unknown, 2024)- https://pandas.pydata.org/docs/index.html

IX. APPENDICES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
train = pd.read_csv(r'/content/train.csv') #Load train data (Write train.csv directory)
test = pd.read_csv(r'/content/test.csv') #Load test data (Write test.csv directory)
# Verify the types of train and test
print(type(train))
print(type(test))
# Concatenate train and test data
data = pd.concat([train, test], sort=False)
# Visualize the DataFrame data
data
#Plot features with more than 1000 NULL values
features = []
nullValues = []
for i in data:
    if (data.isna().sum()[i])>1000 and i!='SalePrice':
        features.append(i)
        nullValues.append(data.isna().sum()[i])
y_pos = np.arange(len(features))
plt.bar(y_pos, nullValues, align='center', alpha=0.5)
plt.xticks(y_pos, features)
plt.ylabel('NULL Values')
plt.xlabel('Features')
plt.title('Features with more than 1000 NULL values')
plt.show()
#Dealing with NULL values
data = data.dropna(axis=1, thresh=1000)
# Select only numerical columns
```

```
numeric_columns = data.select_dtypes(include='number')
# Replace NaN values in numerical columns with the mean
data[numeric_columns.columns] = numeric_columns.fillna(numeric_columns.mean())
# Display the updated DataFrame
Data
#Dealing with NULL values
data = pd.get_dummies(data) #Convert string values to integer values
#Drop features that are correlated to each other
covarianceMatrix = data.corr()
listOfFeatures = [i for i in covarianceMatrix]
setOfDroppedFeatures = set()
for i in range(len(listOfFeatures)) :
    for j in range(i+1,len(listOfFeatures)): #Avoid repetitions
        feature1=listOfFeatures[i]
        feature2=listOfFeatures[j]
        if abs(covarianceMatrix[feature1][feature2]) > 0.8: #If the correlation between the
features is > 0.8
            setOfDroppedFeatures.add(feature1) #Add one of them to the set
#I tried different values of threshold and 0.8 was the one that gave the best results
data = data.drop(setOfDroppedFeatures, axis=1)
#Drop features that are not correlated with output
nonCorrelatedWithOutput = [column for column in data if
abs(data[column].corr(data["SalePrice"])) < 0.045]
#I tried different values of threshold and 0.045 was the one that gave the best results
data = data.drop(nonCorrelatedWithOutput, axis=1)
#Plot one of the features with outliers
plt.plot(data['LotArea'], data['SalePrice'], 'bo')
plt.axvline(x=75000, color='r')
plt.ylabel('SalePrice')
plt.xlabel('LotArea')
plt.title('SalePrice in function of LotArea')
plt.show()
# Remove outliers from the 'data' DataFrame
```

```
newData = data.copy()  # Make a copy of the original DataFrame
# Define a function to remove outliers using the IQR method
def remove_outliers_iqr(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]
    return df
# Remove outliers from the 'LotArea' column
newData = remove_outliers_iqr(newData, 'LotArea')
# Plot the feature 'LotArea' against 'SalePrice' after removing outliers
plt.plot(newData['LotArea'], newData['SalePrice'], 'bo')
plt.ylabel('SalePrice')
plt.xlabel('LotArea')
plt.title('SalePrice in function of LotArea (Outliers Removed)')
plt.show()
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, explained_variance_score
# Assuming 'trainWithoutOutliers' contains the preprocessed data without outliers
# Split the data into features (X) and target variable (Y)
X = trainWithoutOutliers.drop("SalePrice", axis=1)  # Features
Y = np.log1p(trainWithoutOutliers["SalePrice"])  # Target variable {log1p(x) = log(x+1)}
# Split the data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
# Train Linear Regression model
linear_reg = LinearRegression()
```

```
linear_reg.fit(X_train, Y_train)
# Train Random Forest Regression model
random_forest_reg = RandomForestRegressor(random_state=42)
random_forest_reg.fit(X_train, Y_train)
# Make predictions on the test set
Y_pred_linear = linear_reg.predict(X_test)
Y_pred_rf = random_forest_reg.predict(X_test)
# Evaluate models using various metrics
metrics = {
    'Linear Regression': {
        'MSE': mean_squared_error(Y_test, Y_pred_linear),
        'R^2': r2_score(Y_test, Y_pred_linear),
        'Explained Variance': explained_variance_score(Y_test, Y_pred_linear)
    },
    'Random Forest Regression': {
        'MSE': mean_squared_error(Y_test, Y_pred_rf),
        'R^2': r2_score(Y_test, Y_pred_rf),
        'Explained Variance': explained_variance_score(Y_test, Y_pred_rf)
    }
}
# Print evaluation metrics for each model
for model, scores in metrics.items():
    print(f"{model} Metrics:")
    for metric, score in scores.items():
        print(f"{metric}: {score}")
    print()
# Compare the performance of the models based on selected metrics
better_models = []
for metric in metrics['Linear Regression'].keys():
    if metrics['Linear Regression'][metric] < metrics['Random Forest Regression'][metric]:
        better_models.append('Linear Regression')
    elif metrics['Random Forest Regression'][metric] < metrics['Linear Regression'][metric]:
```

```
        better_models.append('Random Forest Regression')
    else:
        better_models.append('Both models')
print("Better models based on each metric:")
for metric, better_model in zip(metrics['Linear Regression'].keys(), better_models):
    print(f"{metric}: {better_model}")
X = trainWithoutOutliers.drop("SalePrice", axis=1) #Remove SalePrice column
Y = np.log1p(trainWithoutOutliers["SalePrice"]) #Get SalePrice column {log1p(x) =
log(x+1)}
reg = LinearRegression().fit(X, Y)
# Make prediction
# Check if 'SalePrice' exists in 'newTest' before attempting to drop it
if 'SalePrice' in newTest.columns:
    newTest = newTest.drop("SalePrice", axis=1)  # Remove 'SalePrice' column
# Predict the SalePrice using the regression model
pred = np.expm1(reg.predict(newTest))
# Submit prediction
sub = pd.DataFrame()  # Create a new DataFrame for submission
sub['Id'] = test['Id']
sub['SalePrice'] = pred
# Visualize the DataFrame sub
sub
```