# **Project Report**

## **Project Title: House Price Prediction Using Machine Learning**

**1. Objective**

The primary goal of this project is to predict house prices based on a set of features using a machine learning model. The project utilizes a linear regression algorithm to establish a relationship between the independent variables (features) and the target variable (price).

**2. Workflow and Methodology**

**Step 1: Load the Dataset**

* The dataset is loaded from a CSV file (house(1).csv) using pandas.
* The dataset likely contains various features related to house characteristics, such as square footage, number of bedrooms, date, and price.

**Step 2: Handle Missing Values**

* Missing values are checked using data.isnull().sum(). However, no explicit strategy is implemented to handle them (e.g., dropping or imputing missing values).

**Step 3: Feature Selection and Preprocessing**

* The column 'sqft\_above' is dropped, likely because it was deemed unnecessary or redundant.
* The date column is transformed into a binary column (1 for the year 2014, 0 otherwise). This transformation simplifies the temporal data but may discard other useful time-based information.
* The dataset is inspected using data.head() and data.info() to ensure proper cleaning and preprocessing.

**Step 4: Define Features and Target Variable**

* The target variable is price (dependent variable).
* Features include all columns except id and price (independent variables). These are stored in the x array, while y stores the target.

**Step 5: Split Data into Training and Testing Sets**

* The data is split into training (80%) and testing (20%) subsets using train\_test\_split.
* This ensures the model is evaluated on unseen data, reducing the risk of overfitting.

**Step 6: Train the Model**

* A linear regression model is created and trained using the LinearRegression class from scikit-learn.
* The model learns the relationship between features and the target variable based on the training data.

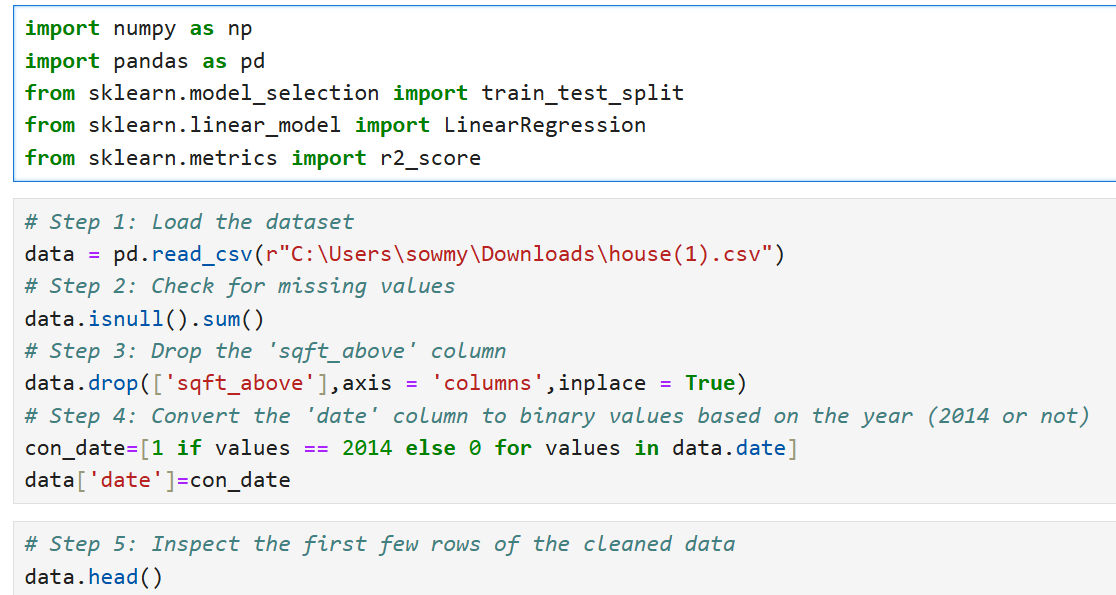
**Step 7: Make Predictions**

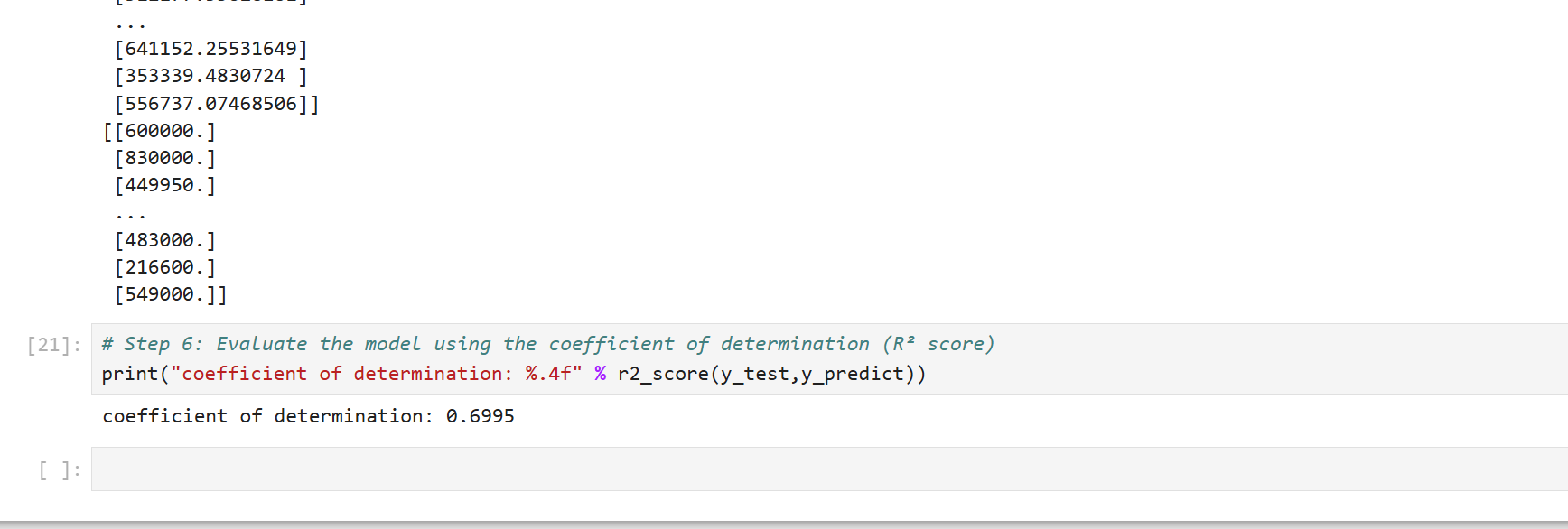
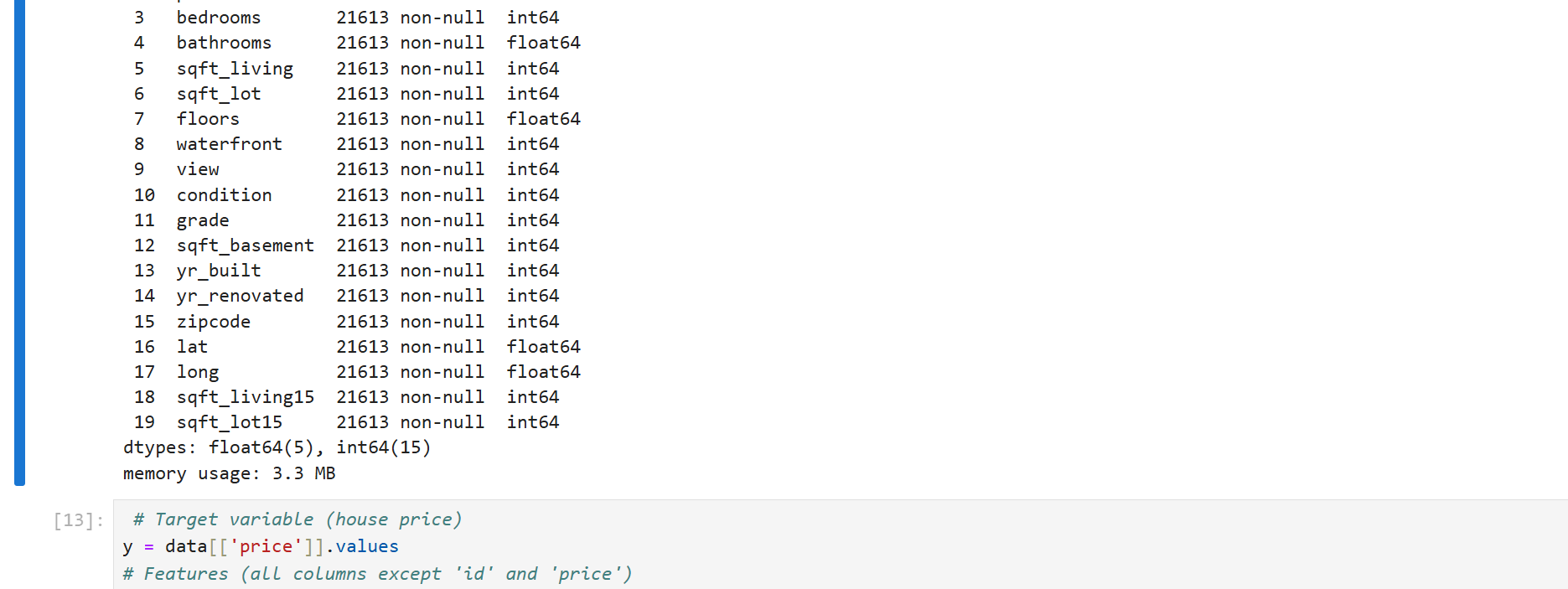
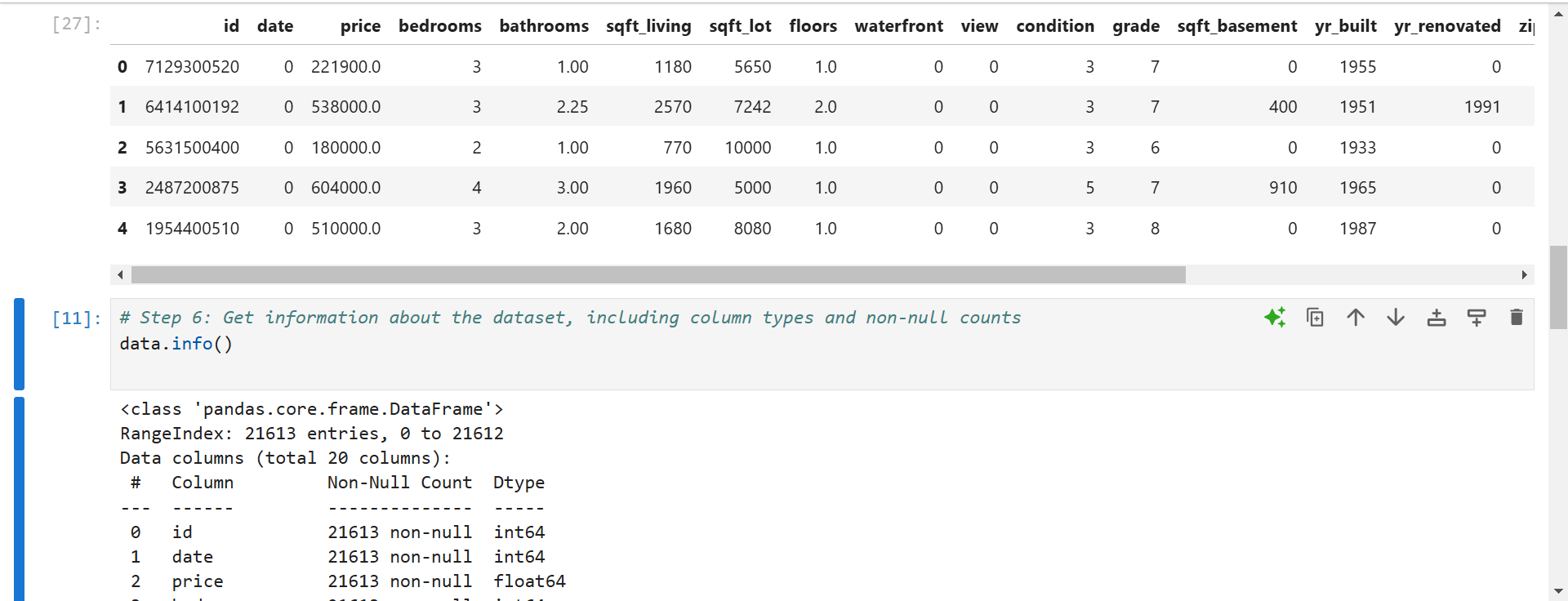
* The trained model is used to predict house prices for the test dataset (x\_test).

**Step 8: Evaluate the Model**

* The performance of the model is evaluated using the R² (coefficient of determination) metric.
  + R² measures how well the model explains the variance in the target variable. A value closer to 1 indicates a better fit.
* Predicted (y\_predict) and actual (y\_test) values are compared to understand the model's accuracy.

**Code:**

****

****

**`3. Results**

* The predicted house prices (y\_predict) and the actual house prices (y\_test) are printed for comparison.
* The R² score is calculated and printed. This score provides a quantitative evaluation of the model's performance.

**4. Observations**

1. **Data Preprocessing**:
   * The dataset is preprocessed effectively but could benefit from additional cleaning steps (e.g., handling missing values and scaling features).
   * The binary conversion of the date column simplifies the data but may result in a loss of temporal trends.
2. **Model Performance**:
   * The R² score evaluates the model's predictive power. If the score is significantly less than 1, it indicates the need for better features or model selection.
3. **Predicted vs. Actual Values**:
   * Printing both predicted and actual values is a good practice for visually analyzing model accuracy, though it does not quantify performance.

**5. Limitations**

1. **Feature Engineering**:
   * The binary transformation of the date column may oversimplify the data. Extracting features like year, month, or day could improve performance.
2. **Missing Value Handling**:
   * The code identifies missing values but does not handle them, which might lead to errors or reduced performance.
3. **Scaling**:
   * Features are not scaled. Linear regression assumes features are on a similar scale, and large variations could affect the model.
4. **Evaluation**:
   * The project uses only R² for evaluation. Including additional metrics like Mean Squared Error (MSE) or Mean Absolute Error (MAE) would provide more insights.
5. **Regularization**:
   * No regularization techniques like Ridge or Lasso regression are used. These could prevent overfitting and improve generalization.

**6. Recommendations**

* **Feature Engineering**: Extract additional features from the date column and explore interactions between features.
* **Feature Scaling**: Normalize or standardize features using StandardScaler from sklearn.
* **Handle Missing Values**: Drop or impute missing values to avoid errors during training.
* **Regularization**: Experiment with Ridge or Lasso regression for better generalization.
* **Evaluation Metrics**: Use MSE or MAE alongside R² to get a more comprehensive understanding of model performance.
* **Cross-Validation**: Implement cross-validation to ensure robust evaluation.

**7. Conclusion**

This project demonstrates a basic implementation of linear regression for house price prediction. While the model provides a starting point, incorporating additional preprocessing steps, feature engineering, and evaluation metrics could significantly enhance its performance and applicability.