Housing Price Prediction Project Report

1. Problem Description

This project focused on predicting house **prices** using various property features. The main goal was to create a machine learning model that could estimate a house's **price** based on its characteristics. I selected this problem because predicting housing prices is a common and very useful application of machine learning. It offers a straightforward result that is simple to comprehend and assess.

2. Dataset

The information for this project came from a file called "*Housing.csv*" on **Kaggle**. This file has many details about houses. For the first model, I only used two things: the house's "**area**" and its "**stories**". I chose these because they are basic parts of a house and usually affect its price. The file had other details like bedrooms, bathrooms, and parking, but I did not use them in this first try.

3. Modeling Approach

For this project, I chose to use a simple method called Linear Regression. This method is good because it's easy to understand and works well for predicting numbers. The data received the following preprocessing steps:

1. **Data Splitting**

   The dataset was divided into training and testing sets. **70%** of the data was allocated for training the model, and the remaining **30%** was reserved for evaluating its performance on unseen data. A **random_state** of 42 was set to ensure reproducibility of the split.

2. **Scaling the Features**

   The **"area"** and **"stories**" of a house are very different in size (e.g., area could be thousands, stories are usually under ten). To make sure both features were equally important to the model, I used something called StandardScaler. This made the numbers for both "area" and "stories" have an average of 0 and a standard difference of 1. This step is important for Linear Regression so that features with large numbers don't unfairly affect the model, and it helps the model learn more quickly. I applied this scaling to the training data first, and then used the same scaling rules for the test data.

After these steps, I trained the Linear Regression model using the prepared training data (the scaled "area" and "stories" values) and the matching house prices.The **LinearRegression** model was then trained using the scaled training features **(X_scaled)** and the corresponding training prices **(y_train)**.

4. Evaluation

The model's performance was evaluated using three common regression metrics and the evaluation results were as follows:

```
Training Data:
Mean Absolute Error: 918408.8496619147
Mean Squared Error: 1673224013890.9565
R2 Score: 0.4114569105882494

Testing Data:
Mean Absolute Error: 1162055.035550395
Mean Squared Error: 2632570392001.295
R2 Score: 0.3459551624767614
```

The R2 scores of approximately 0.30 for the training set and 0.41 for the testing set indicate that the model, using only **area** and **stories**, explains only about **34-41%** of the variability in house prices. The large MAE and MSE values further confirm that the model's predictions have substantial errors compared to the actual house prices.

5. Reflection

The biggest problem was that my model wasn't very good at guessing prices. The area and stories alone didn't give enough information. It seems many other things affect house prices. The large error metrics also underscored this limitation.

This project strongly reinforced the critical role of comprehensive feature selection. Even a simple linear model requires relevant and sufficient input features to perform well.The necessity of **StandardScaler** was evident, as unscaled features could lead to biased coefficient estimates and slower convergence.Next time, my immediate focus would be on **enriching the feature set**. I would incorporate more of the available features from the Housing.csv dataset, such as: **bedrooms, bathrooms** and **parking**