**Applied Machine Learning**

Contents

[K-Means and DBSCAN Clustering 2](#_Toc191919993)

[Discussion 2](#_Toc191919994)

[Load and Preprocess the Dataset 2](#_Toc191919995)

[Determine the Optimal Number of Clusters (K) 2](#_Toc191919996)

[Apply K-Means Clustering 3](#_Toc191919997)

[Cluster Interpretation 4](#_Toc191919998)

[Handling Outliers 5](#_Toc191919999)

[Comparison with Other Clustering Algorithms (DBSCAN) 6](#_Toc191920000)

[Comparison with K-Means 7](#_Toc191920001)

[Conclusion 8](#_Toc191920002)

[Linear Regression Discussion 8](#_Toc191920003)

[Implementation and Analysis 8](#_Toc191920004)

[Data Preprocessing and Exploration 9](#_Toc191920005)

[Exploratory Data Analysis (EDA) 9](#_Toc191920006)

[Feature Engineering and Selection 12](#_Toc191920007)

[Train a Linear Regression Model 13](#_Toc191920008)

[Implementing Linear Regression Using Gradient Descent 13](#_Toc191920009)

[Predicting House Prices for New Data 15](#_Toc191920010)

[Conclusion 15](#_Toc191920011)

# 

# K-Means and DBSCAN Clustering

### Discussion

This section details the implementation and analysis of K-Means and DBSCAN clustering algorithms on the customer dataset to segment customers based on their purchasing behavior and engagement metrics. The dataset, after preprocessing, included features such as:

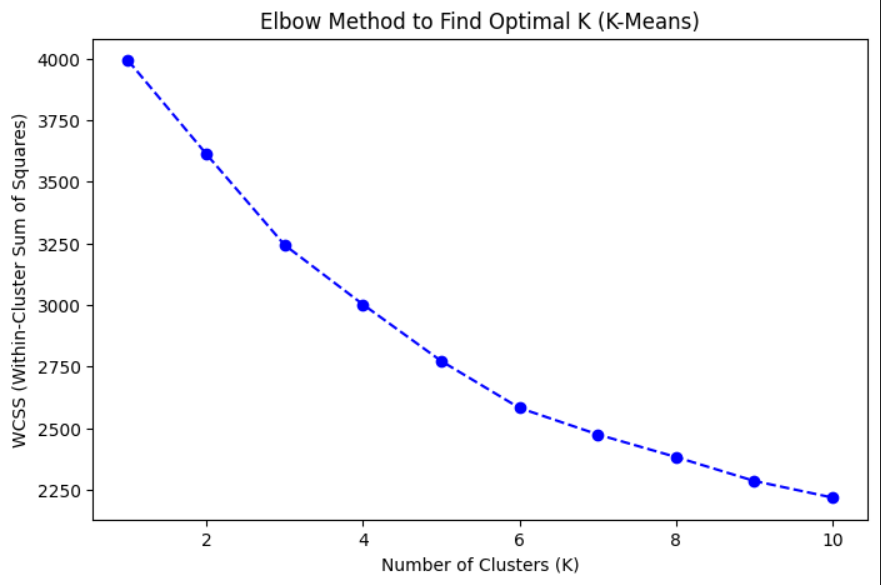
* **Annual\_Income**
* **Spending\_Score**
* **Website\_Visits**
* **Product\_Categories\_Purchased**
* **Total\_Purchase\_Amount**
* **Average\_Session\_Duration**
* **Return\_Rate**
* **Discount\_Usage**

### Load and Preprocess the Dataset

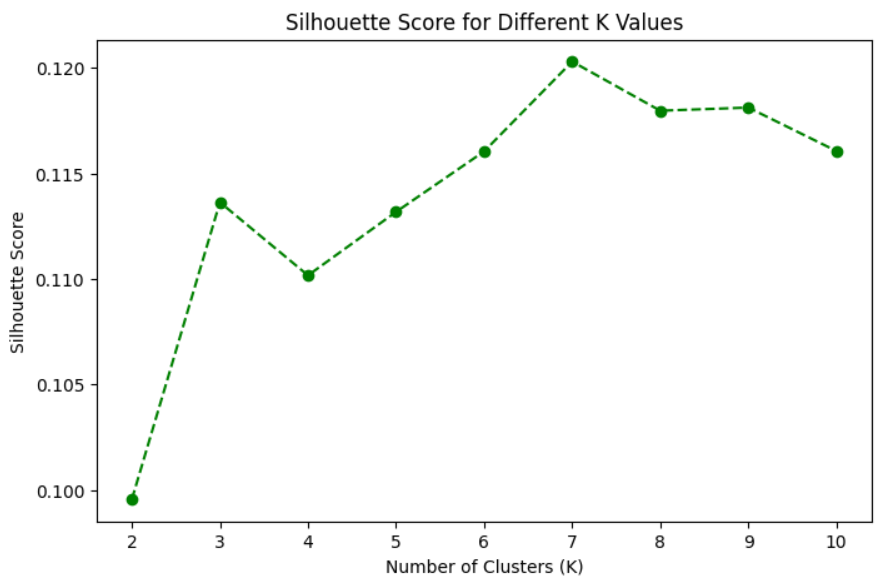
* The dataset was loaded from *kmeans\_dataset.csv*, and irrelevant columns (**Customer\_ID** and **Age**) were dropped to focus on numerical features relevant to clustering.
* Missing values were handled by replacing them with the mean of each column, ensuring no data points were lost.
* Numerical features were normalized using **Z-score standardization** (subtracting the mean and dividing by the standard deviation) to ensure equal contribution to distance-based clustering algorithms. This step is crucial for **K-Means and DBSCAN**, as they rely on Euclidean distance.

### Determine the Optimal Number of Clusters (K)

* The **Elbow Method** and **Silhouette Score** were used to determine the optimal number of clusters for K-Means.
* The **Elbow Method plot** showed a WCSS (Within-Cluster Sum of Squares) decrease with increasing K like L shape, with an elbow point around **K = 7** indicating these as potential optimal values.



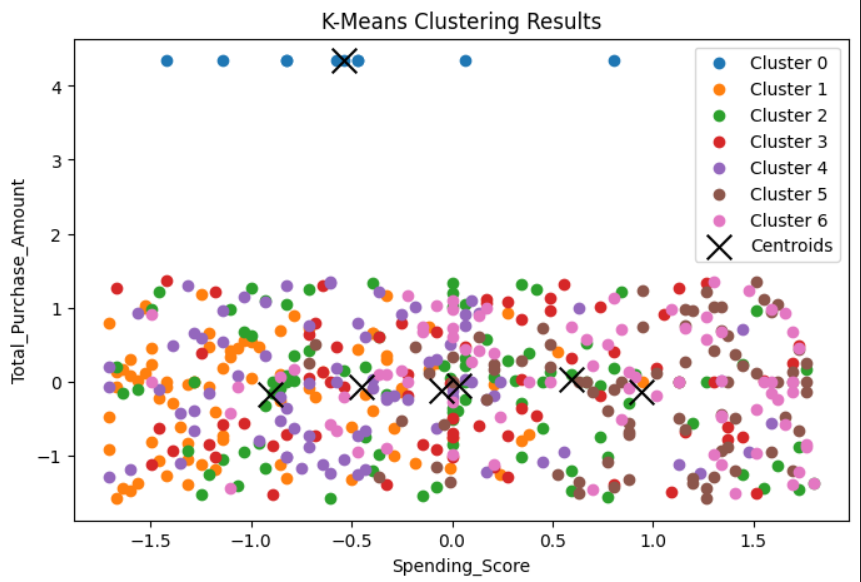
* The **Silhouette Score plot** indicated the highest score (~0.12) at **K = 7**, suggesting **K = 7** provides the best separation between clusters.



* Based on these results, **K = 7** was chosen for K-Means clustering.

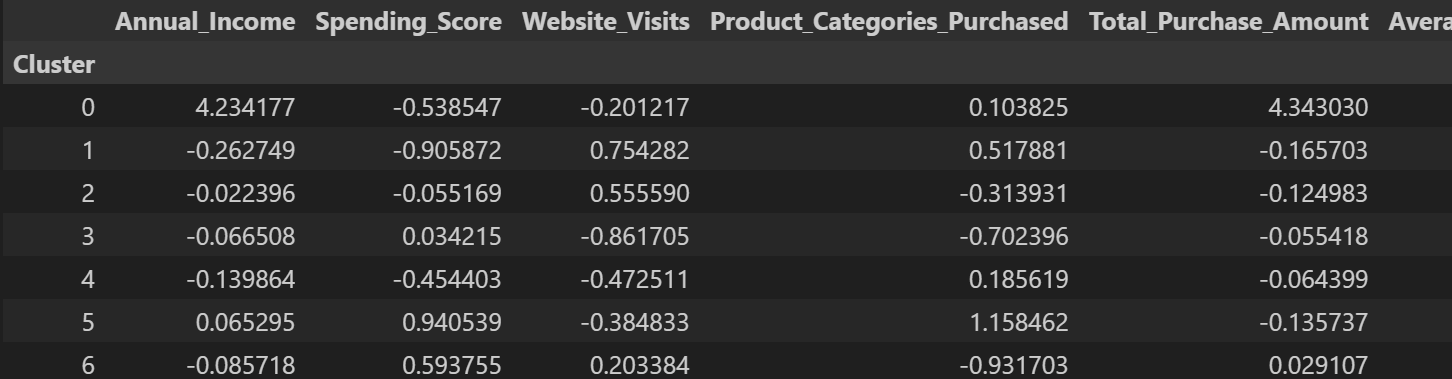
### Apply K-Means Clustering

* A manual **K-Means implementation** was applied with **K = 7** on the preprocessed dataset.
* The clustering results were visualized using a **scatter plot of Spending\_Score vs. Total\_Purchase\_Amount**, showing seven distinct clusters with centroids marked.



### Cluster Interpretation

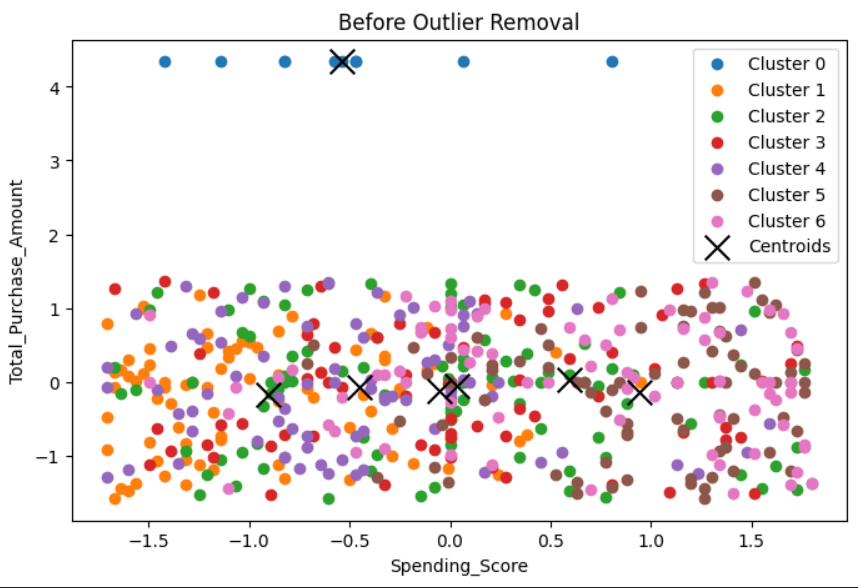
* Cluster labels were assigned to the dataset, and **average Z-scores** of each feature were computed for the **7 clusters**



* + **Cluster 0**: High **Annual\_Income** and **Total\_Purchase\_Amount**, but low **Spending\_Score** – high-income, low-spending customers.
  + **Cluster 1**: Low **Spending\_Score**, high **Website\_Visits**, and **Product\_Categories\_Purchased** – frequent visitors spending less.
  + **Cluster 2**: Moderate features, low **Discount\_Usage** – value-driven, price-sensitive customers.
  + **Cluster 3**: Low **Website\_Visits** and **Product\_Categories\_Purchased**, moderate **Spending\_Score** – less engaged customers.
  + **Cluster 4**: Low **Average\_Session\_Duration**, high **Return\_Rate** – dissatisfied customers or frequent returners.
  + **Cluster 5**: High **Spending\_Score** and **Product\_Categories\_Purchased** – high-spending, engaged customers.
  + **Cluster 6**: High **Spending\_Score**, moderate **Website\_Visits**, but missing data for some features – unique customer behavior or data issues.
* These insights help in **targeted marketing strategies**, such as promotions for low-spending customers or loyalty programs for high-spending segments.

### Handling Outliers

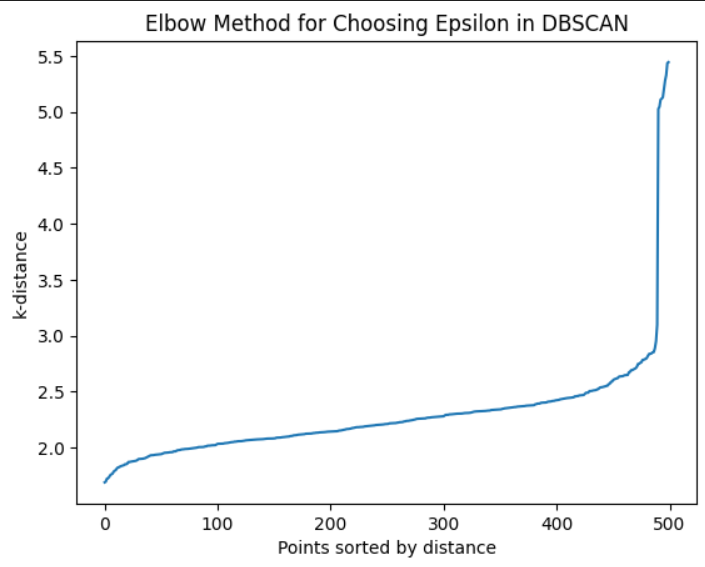
* **Outliers were detected and removed using the Interquartile Range (IQR) method**, filtering data points outside **1.5 \* IQR** from the quartiles.
* **K-Means clustering was reapplied**, and results were visualized:
  + **Before Outlier Removal** : Clusters had extreme points, skewing centroids.

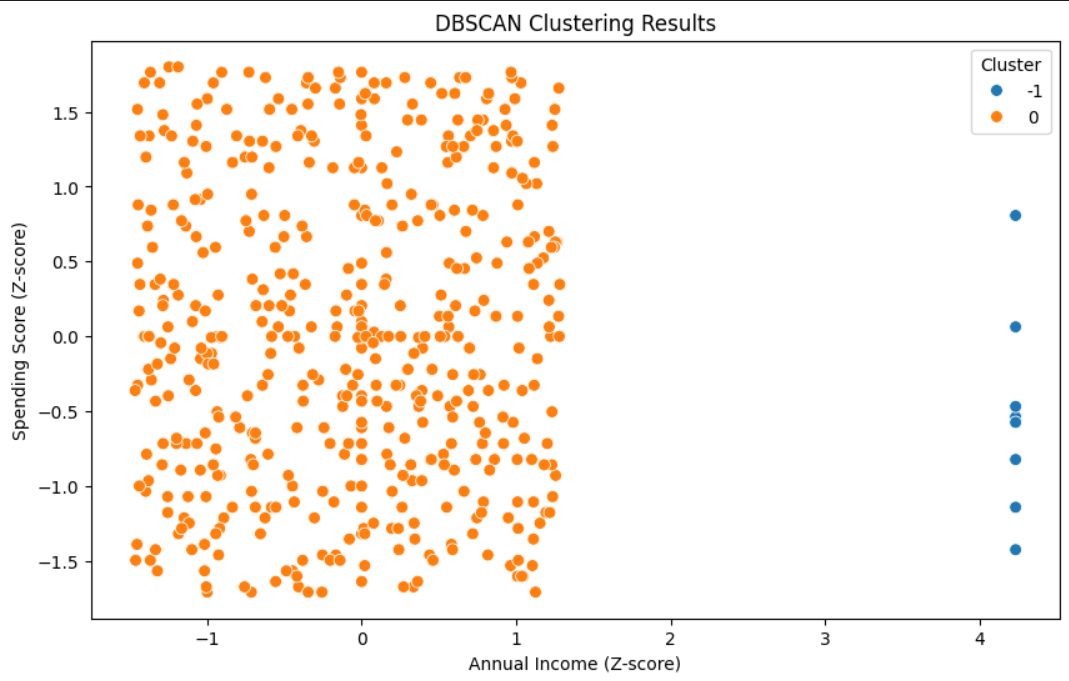
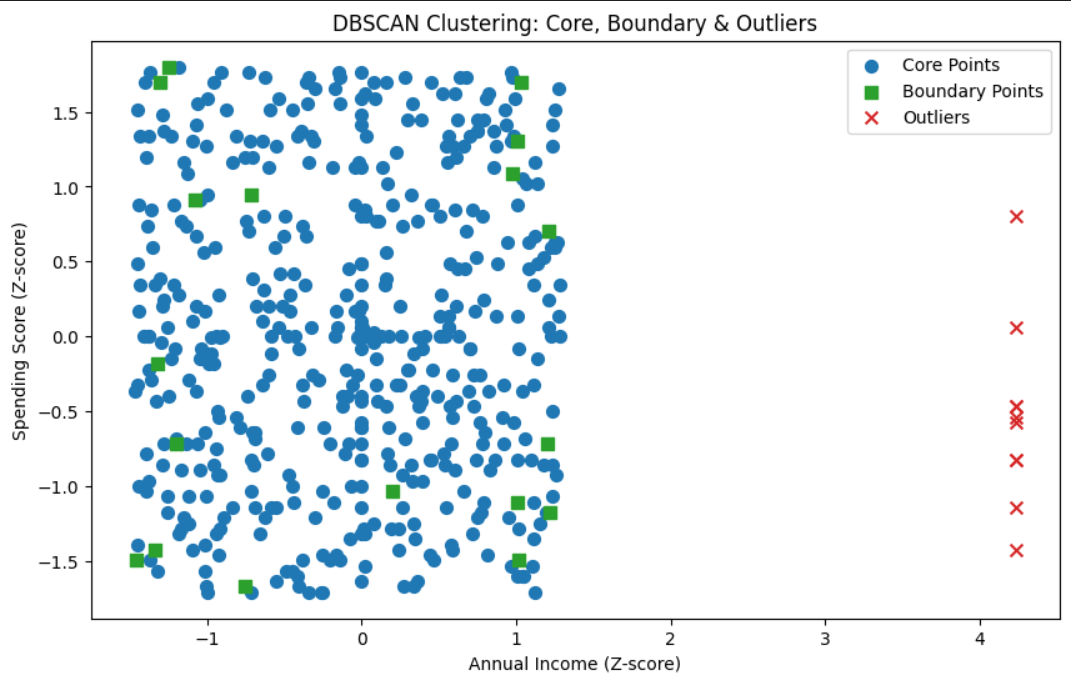


* + **After Outlier Removal**: More compact clusters with clearer separation, improving interpretability.  
    

### Comparison with Other Clustering Algorithms (DBSCAN)

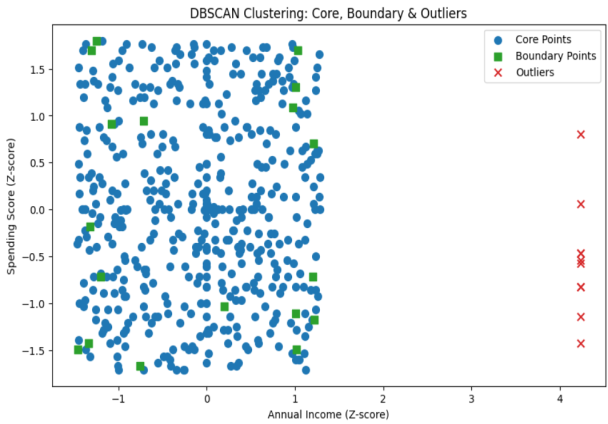
* **DBSCAN** was implemented using the cleaned dataset.
* **Optimal eps** (distance threshold) was determined using the **Elbow Method on k-distance values** with **min\_samples = 16**.
* The **k-distance plot** suggested an **elbow at eps = 3.0**, which was chosen.

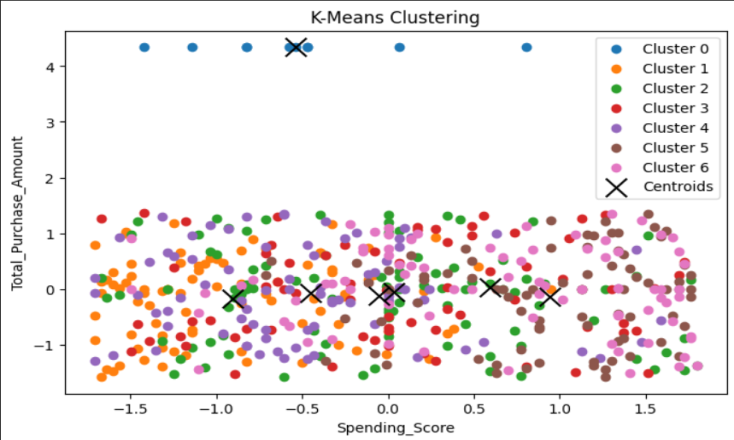


* DBSCAN identified **two clusters** and **noise points** (labeled -1), visualizedin (**Annual\_Income vs. Spending\_Score**):
  + **Cluster 0 (orange)** contained 490 data points.
  + **Cluster -1 (blue)** contained 10 noise points.  
    
* **Core, Boundary, and Outliers Analysis**:
  + **Core points (blue)** dominated the dataset.
  + **Boundary points (green)** were minimal.
  + **Outliers (red 'x')** were effectively identified.  
    

### Comparison with K-Means

|  |  |  |
| --- | --- | --- |
| Metric | K-Means | DBSCAN |
| Cluster Shape | Spherical | Arbitrary |
| Handles Noise | No | Yes |
| Requires K | Yes | No |
| Performance | Good for structured data | Best for noisy data |

* **K-Means**: Best when clusters are **spherical and evenly sized**, producing **7 overlapping but clear clusters**.
* **DBSCAN**: Best for **irregularly shaped clusters** and **detecting noise**, but it **struggled with dense, overlapping structures**, labeling most data as a **single cluster**.



### Conclusion

* **K-Means (K = 7)** successfully identified **7 customer segments**, including **high-spending engaged customers (Cluster 5)**, **low-spending frequent visitors (Cluster 1)**, and **dissatisfied customers (Cluster 4)**.
* **Outlier removal using IQR improved cluster compactness and interpretability**.
* **DBSCAN effectively detected noise but struggled with clustering this dataset**, grouping most points into a **single cluster**.
* **K-Means is preferable when clusters are spherical and estimable**, while **DBSCAN is better for irregular shapes and noise detection**.
* **Future Work**: Further refinement (e.g., **tuning K, adjusting DBSCAN parameters, or adding features**) could enhance clustering quality and interpretability.

# ****Linear Regression Discussion****

## Implementation and Analysis

This section presents the implementation and analysis of a linear regression model to predict **House\_Price** using the features **House\_Age, Num\_Bedrooms, Area\_Sqft, and Distance\_to\_City\_Center** from the housing dataset, as specified in the assignment. The analysis covers data preprocessing, exploratory data analysis (EDA), feature engineering, model training, gradient descent implementation, and prediction for new data.

### Data Preprocessing and Exploration

The dataset was loaded from linear\_regression\_dataset.csv using Pandas, and initial checks revealed missing values summarized as follows:

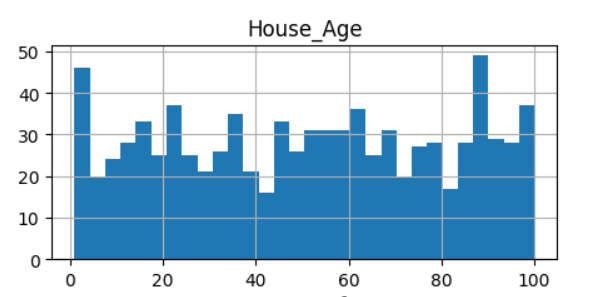
* **House\_Age**: 14 missing values
* **Num\_Bedrooms**: 14 missing values
* **Area\_Sqft**: 14 missing values
* **Distance\_to\_City\_Center**: 14 missing values
* **House\_Price**: 0 missing values

Rows with missing values were removed using dropna(), reducing the dataset from 864 to 852 rows. Post-cleaning, no missing values remained, confirming successful preprocessing.

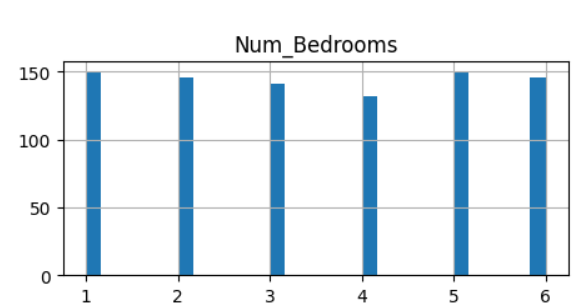
### Exploratory Data Analysis (EDA)

Feature distributions were visualized using histograms:

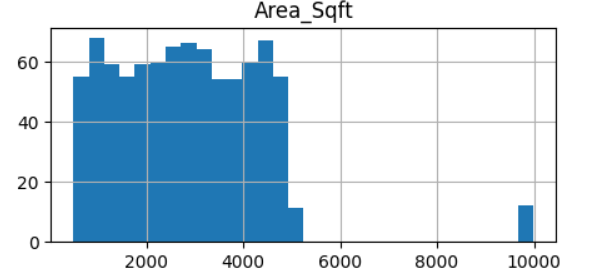
* House\_Age: Right-skewed distribution, with most houses aged between 20 and 80 years



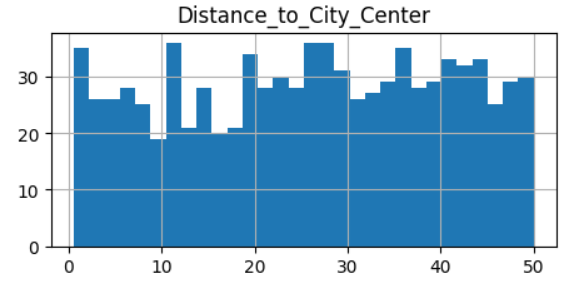
* Num\_Bedrooms: Discrete distribution, predominantly 3–6 bedrooms.



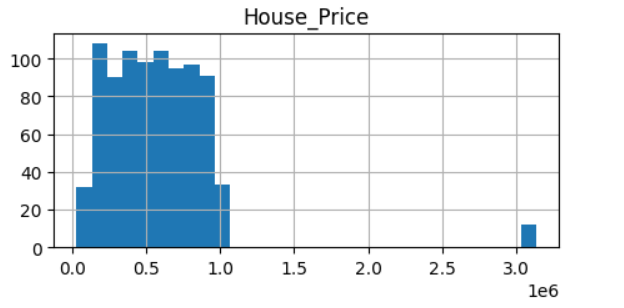
* Area\_Sqft: Right-skewed, with most houses between 1,000 and 5,000 sq. ft., with some outliers up to 10,000 sq. ft.



* Distance\_to\_City\_Center: Right-skewed, mostly between 10–40 units from the city center.

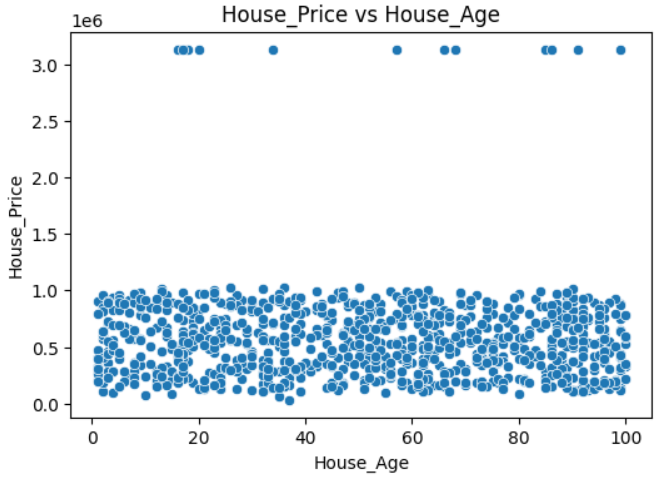


* House\_Price: Right-skewed, ranging from ~$100,000 to over $3,000,000, concentrated below $1,000,000.

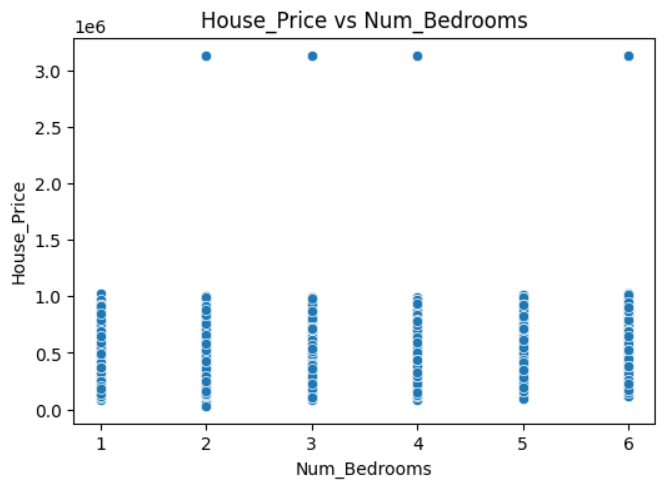


Relationships between features and House\_Price were explored using scatter plots:

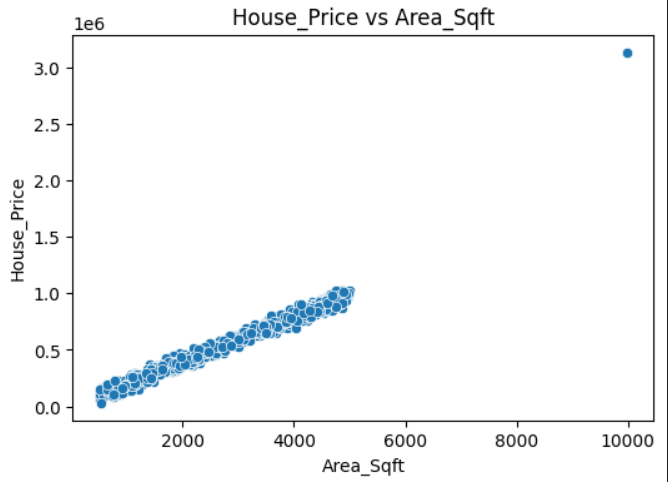
* House\_Price vs. House\_Age: No clear trend, suggesting weak correlation.



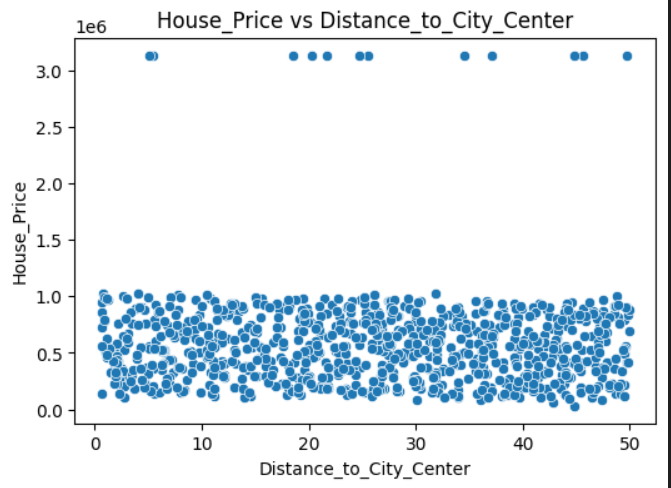
* House\_Price vs. Num\_Bedrooms: Slight positive correlation, but with high variability.



* House\_Price vs. Area\_Sqft: Strong positive correlation, indicating high predictive power.



* House\_Price vs. Distance\_to\_City\_Center: Weak negative correlation.

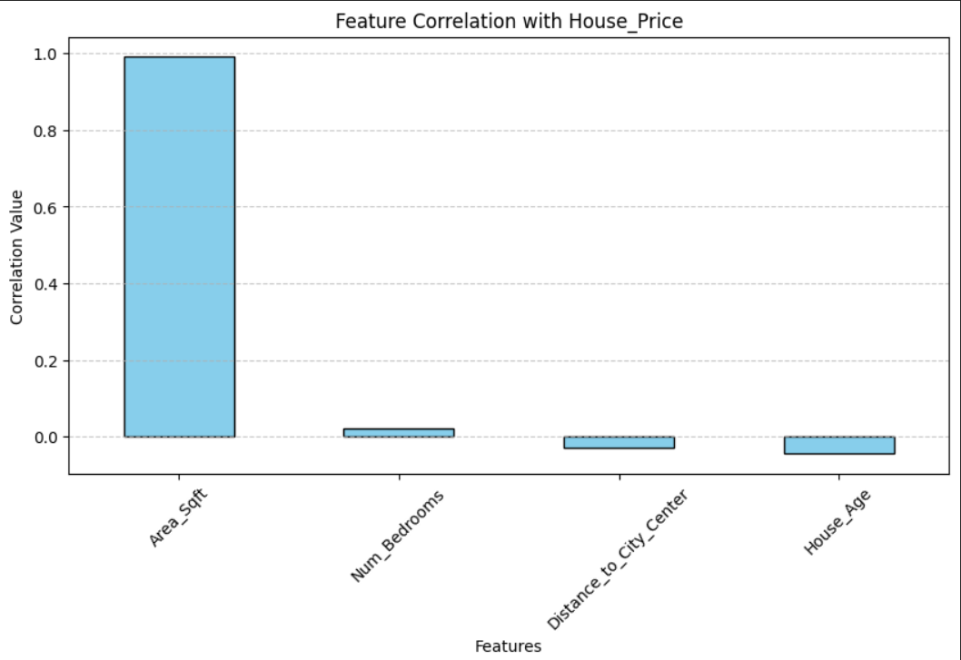


Outliers were identified and removed using the Interquartile Range (IQR) method, reducing the dataset from 864 to 852 rows for improved data quality.

### Feature Engineering and Selection

Numerical features were standardized using Z-score normalization to ensure consistent scaling, stored in df\_cleaned. The computed correlations with House\_Price were:

* **Area\_Sqft**: 0.9916 (strong positive correlation)
* **Num\_Bedrooms**: 0.0208 (very weak positive correlation)
* **Distance\_to\_City\_Center**: -0.0294 (very weak negative correlation)
* **House\_Age**: -0.0441 (very weak negative correlation)



Polynomial features (squared and cubed terms) were created to capture potential non-linearity, improving model flexibility.

### Train a Linear Regression Model

The dataset was split into training (80%) and testing (20%) sets, with 681 training rows and 171 testing rows. A linear regression model was trained LinearRegression, incorporating polynomial features. Model evaluation results were:

* **Mean Absolute Error (MAE)**: 0.6838
* **Mean Squared Error (MSE)**: 0.7009
* **R² Score**: 0.8497

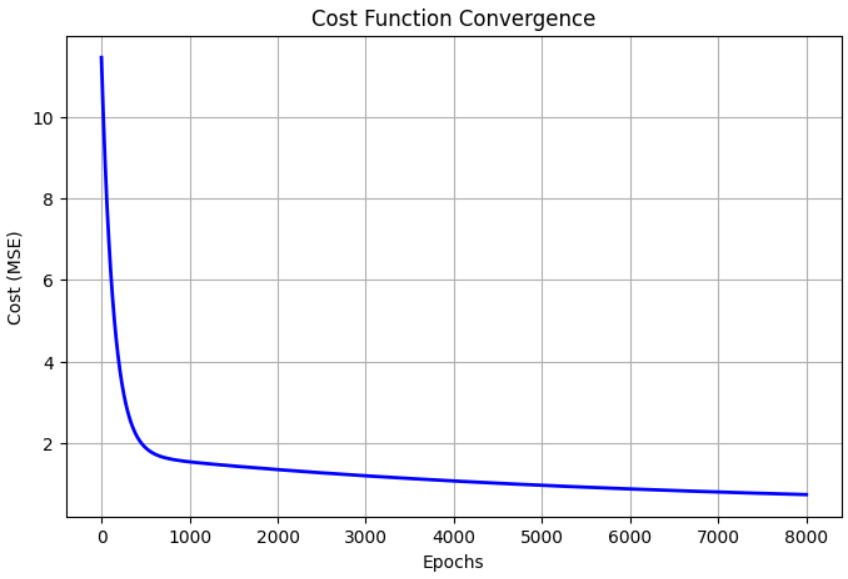
These metrics confirm the model's effectiveness, with Area\_Sqft as the dominant predictor.

### Implementing Linear Regression Using Gradient Descent

A manual linear regression model was implemented using gradient descent on the normalized dataset. The process included:

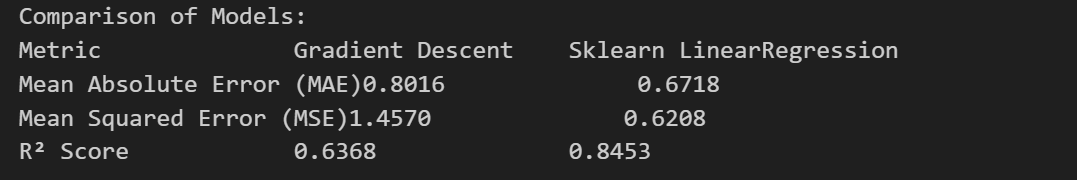
* Adding a bias term (column of ones).
* Initializing weights randomly.
* Setting hyperparameters: learning rate = 0.001, epochs = 8,000.

The cost function showed rapid initial convergence, stabilizing at 0.9288 after 8,000 epochs.

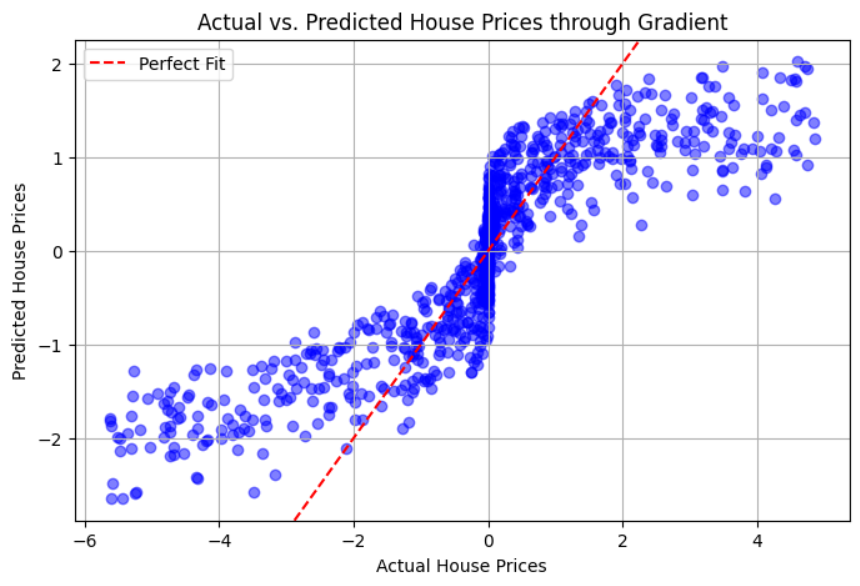


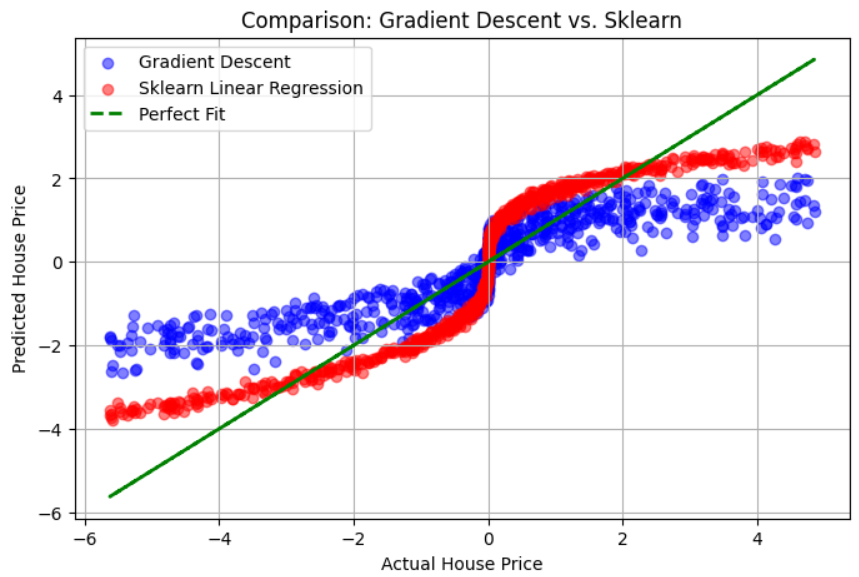
Model performance on normalized data:

* **MAE**: 0.8016
* **MSE**: 1.4570
* **R² Score**: 0.6368



Actual vs. predicted plots highlighted the gradient descent model's dispersion, indicating underperformance compared to Scikit-learn’s implementation.





### Predicting House Prices for New Data

A function predict\_house\_price was created to predict house prices using the trained Scikit-learn model. An example prediction for a house with:

* House\_Age = 93
* Num\_Bedrooms = 1
* Area\_Sqft = 3,885
* Distance\_to\_City\_Center = 36.65

yielded a predicted price of **804,874.37**.



### Conclusion

The linear regression model successfully predicted House\_Price with high accuracy using Scikit-learn’s implementation, achieving an **R² score of 0.8497** and **MAE of 0.6718**. Data preprocessing (handling missing values and outliers) and feature engineering (Z-score normalization and polynomial features) ensured robust modeling, with Area\_Sqft identified as the most predictive feature (correlation: 0.9916). The gradient descent model, while functional, underperformed (R² = 0.6368) due to hyperparameter limitations, though it remains a viable alternative with further tuning. The predict\_house\_price function demonstrated practical real-world applicability, making the **Scikit-learn model the recommended approach for reliable house price predictions.**