**CS4104- APPLIED MACHINE LEARNING**

**ASSIGNMENT NO 1 \_\_ REPORT**

**Name: Maryam Tariq.**

**Roll no: f21-9337, BCS-8B**

**Date: 3/4/2025**

**Instructor: Mr. Ali Raza**

**1. Introduction**

Machine learning techniques play a crucial role in data analysis and predictive modeling. In this assignment, we explore two key concepts:

1. **Clustering (Unsupervised Learning)**: Implementing **K-Means** and **DBSCAN** to identify patterns in a dataset.
2. **Regression (Supervised Learning)**: Applying **Linear Regression** and **Gradient Descent** to predict house prices.

### ****Objectives:****

* Implement **K-Means and DBSCAN** clustering algorithms.
* Identify the optimal number of clusters using **Elbow Method** and **Silhouette Score**.
* Compare the effectiveness of **K-Means vs DBSCAN**.
* Perform **Linear Regression using Scikit-learn** and **implement Gradient Descent manually**.
* Evaluate model performance using **MAE, MSE, and R-squared Score**.

# ****2. Methodology****

## ****2.1 Clustering (K-Means & DBSCAN)****

#### ****Step 1: Load & Preprocess Dataset****

* Loaded the dataset and handled missing values by filling them with the median.
* Standardized the numerical features using **StandardScaler**.

#### ****Step 2: Finding the Optimal Number of Clusters (K)****

* Applied the **Elbow Method** by plotting **inertia vs K**.
* Used **Silhouette Score** to validate the optimal cluster count.

#### ****Step 3: Apply K-Means Clustering****

* Implemented **K-Means** with the optimal K value.
* Visualized the clustering results using a scatter plot.

#### ****Step 4: Cluster Interpretation****

* Assigned cluster labels and computed the **mean feature values per cluster**.

#### ****Step 5: Handling Outliers****

* Used **Z-score method** to detect outliers.
* Removed outliers and re-applied **K-Means**, comparing results.

#### ****Step 6: Comparison with DBSCAN****

* Implemented **DBSCAN clustering** and compared it with **K-Means**.
* Analyzed scenarios where **DBSCAN is more effective than K-Means**.

## ****2.2 Linear Regression (Prediction Task)****

#### ****Step 1: Data Preprocessing & EDA****

* Handled missing values.
* Performed **Exploratory Data Analysis (EDA)** by visualizing feature distributions and relationships with **House\_Price**.
* Used **correlation heatmaps** to identify important features.

#### ****Step 2: Feature Engineering & Selection****

* Scaled the features using **StandardScaler**.
* Selected relevant features based on correlation analysis.
* Created **polynomial features** to capture non-linearity.

#### ****Step 3: Train a Linear Regression Model****

* Split data into **train (80%) and test (20%)**.
* Trained a **Linear Regression model using Scikit-learn**.
* Evaluated performance using **MAE, MSE, and R-squared Score**.

#### ****Step 4: Implement Linear Regression using Gradient Descent****

* Implemented **Gradient Descent from scratch**.
* Compared performance with **Scikit-learn’s Linear Regression model**.

#### ****Step 5: Predict House Prices****

* Created a function to **predict house prices for new inputs**.
* Applied appropriate **feature scaling & transformation** before predictions.

# ****3. Results & Discussion****

## ****3.1 Clustering Results****

#### ****Optimal K Selection****

* **Elbow Method** suggested an optimal K of **3**.
* **Silhouette Score** also indicated that **3 clusters** were best.

#### ****K-Means Clustering Interpretation****

* Clusters were well-separated in the visualization.
* The **mean feature values per cluster** provided insights into different groups.

#### ****DBSCAN vs. K-Means****

* **K-Means performed well** for compact clusters.
* **DBSCAN handled outliers better** and detected arbitrary-shaped clusters.
* **K-Means requires a predefined K**, while **DBSCAN automatically finds clusters**.

## ****3.2 Regression Results****

#### ****Linear Regression Performance (Scikit-learn)****

* **MAE:** 47,552
* **MSE:** 7,035,494,815
* **R² Score:** 0.972

#### ****Gradient Descent Performance****

* **MAE:** 47,602
* **MSE:** 7,030,830,459
* **R² Score:** 0.972

#### ****Comparison of Models****

* Both **Scikit-learn Linear Regression** and **Gradient Descent** performed similarly.
* Gradient Descent was **slightly less efficient** but still produced comparable results.
* **Polynomial features improved prediction accuracy**.

# ****4. Conclusion****

* **K-Means is best for well-separated clusters**, while **DBSCAN is better for complex data**.
* **Feature selection and polynomial features** improved regression accuracy.
* **Gradient Descent is useful**, but Scikit-learn's built-in model is more efficient.
* **Final house price predictions were reasonable** based on feature inputs.

# ****5. References****

* Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow – Aurélien Géron
* Python Machine Learning – Sebastian Raschka & Vahid Mirjalili
* Scikit-learn Documentation (https://scikit-learn.org/)