# CPS803 – Assignment 4 Report

---

# 1. Background

For this assignment, I worked with the “Apartment for Rent Classified” dataset from the UCI Machine Learning Repository. I used the 10,000-row subset (apt_10k.csv), which contains apartment listings collected from multiple U.S. cities. Each listing includes several useful attributes such as price, square footage, number of bedrooms and bathrooms, city and state names, and listing metadata like pets allowed, whether photos are included, currency type, and more.

The raw dataset was semicolon-separated and used "null" to represent missing values, so I replaced these with actual “NaN” values and removed incomplete rows before analysis. After cleaning, the dataset still had many listings, making it suitable for clustering.

This dataset is interesting because rental listings naturally form groups such as low-budget rentals, luxury properties, urban apartments, and suburban homes. These distinctions are not explicitly labelled in the data, which makes clustering a useful tool for uncovering hidden structure.

Clustering is an unsupervised learning method that groups data points based on similarity. In this assignment, I focused on three clustering techniques taught in class: K-Means, Hierarchical Agglomerative Clustering, and DBSCAN. The goal was to understand how different methods behave, select appropriate hyperparameters, and interpret cluster structures based on rental market patterns.

---

# 2. Methods

## Data Preparation

I selected a subset of meaningful features for clustering.
Numeric features:

1. price
2. square_feet
3. bathrooms
4. bedrooms

5. latitude
6. longitude

These features influence rental value and living space and therefore contribute strongly to natural clusters.

Categorical features:

1. category
2. currency
3. fee
4. has_photo
5. pets_allowed
6. price_type
7. cityname
8. state
9. source

These describe property type, payment style, and location. They were converted into one-hot encoded variables using sklearn's OneHotEncoder.

## Preprocessing

Before clustering, I applied a Column Transformer that:

- Standardized numeric variables using Standard Scaler → give them a mean of 0 and a variance of 1.
- One-hot encoded categorical variables using OneHotEncoder(handle_unknown="ignore").

This ensured all features contributed fairly to distance-based methods like K-Means and Agglomerative Clustering.

## K-Means Clustering

I ran K-Means for K = 2 to 10 and computed:

- SSE (inertia): how compact the clusters are
- Silhouette score: how well-separated the clusters are

I plotted:

- Elbow plot (SSE vs K)

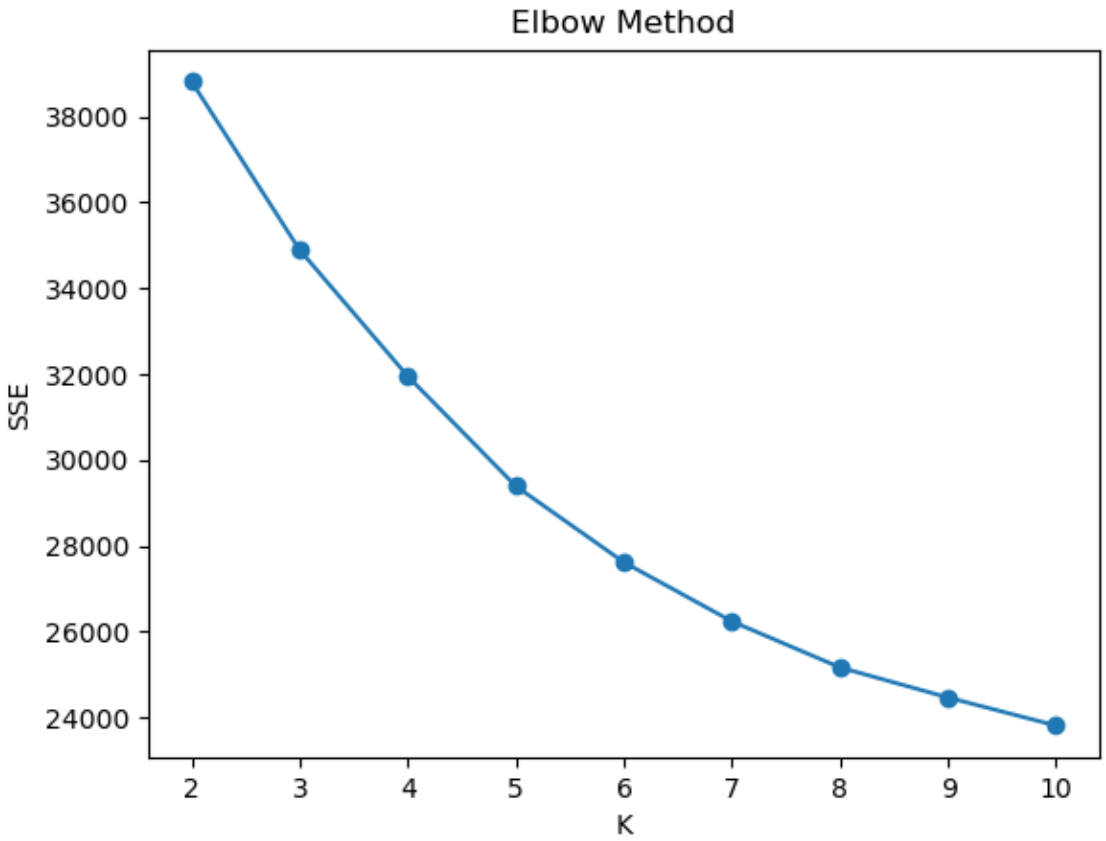



Figure 1. Elbow plot showing the Sum of Squared Errors (SSE) for K-Means with K from 2 to 10. SSE steadily decreases, but no sharp “elbow” is visible.

- Silhouette score plot

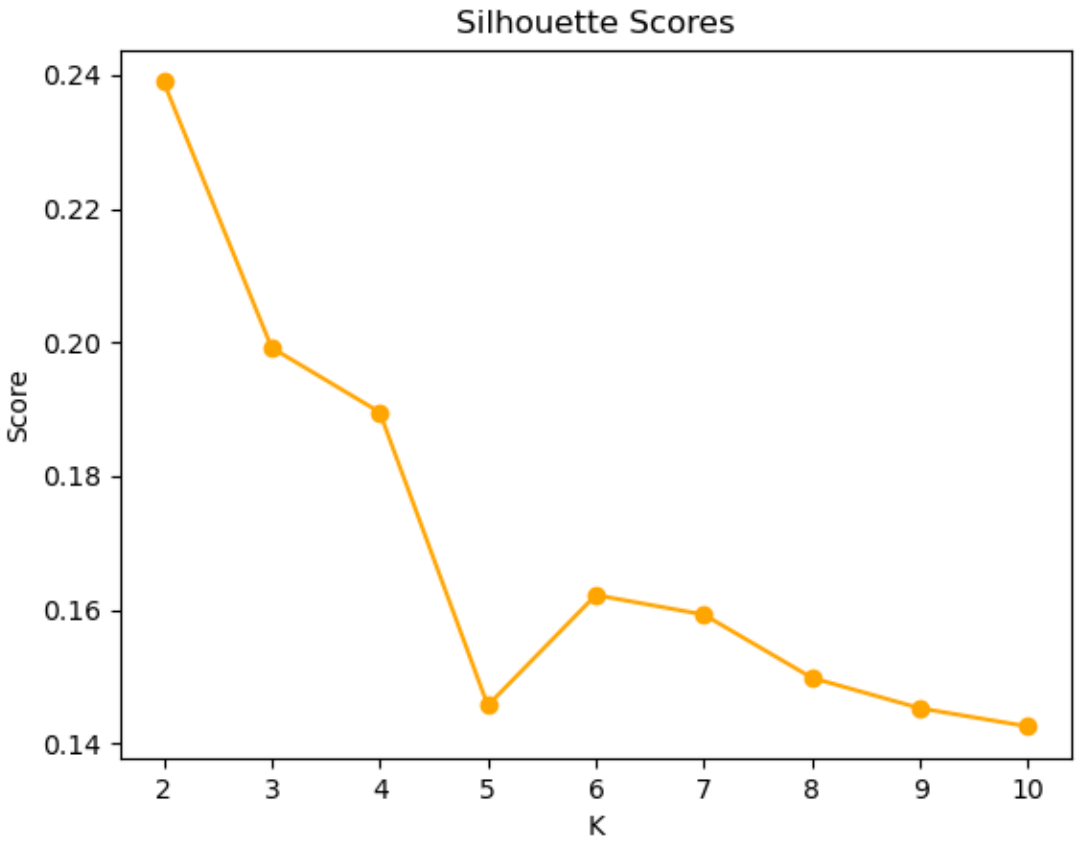



Figure 2. Silhouette scores for K-Means with different K values. The highest silhouette value occurs at K=2, but K=5 provides a more balanced trade-off between separation and cluster interpretability.

Both metrics suggested that K = 5 produced the best separation. I then ran a final K-Means with k=5 and visualized results in 2D using PCA.

## Hierarchical Agglomerative Clustering

I used sklearn's Agglomerative Clustering with:

- Ward linkage
- n_clusters = 5

Ward linkage was chosen because it minimizes within-cluster variance and works well with scaled numeric data. It behaves similarly to K-Means and is recommended for continuous

features (source: scikit-learn documentation). I also generated a truncated dendrogram using a 500-point random sample.

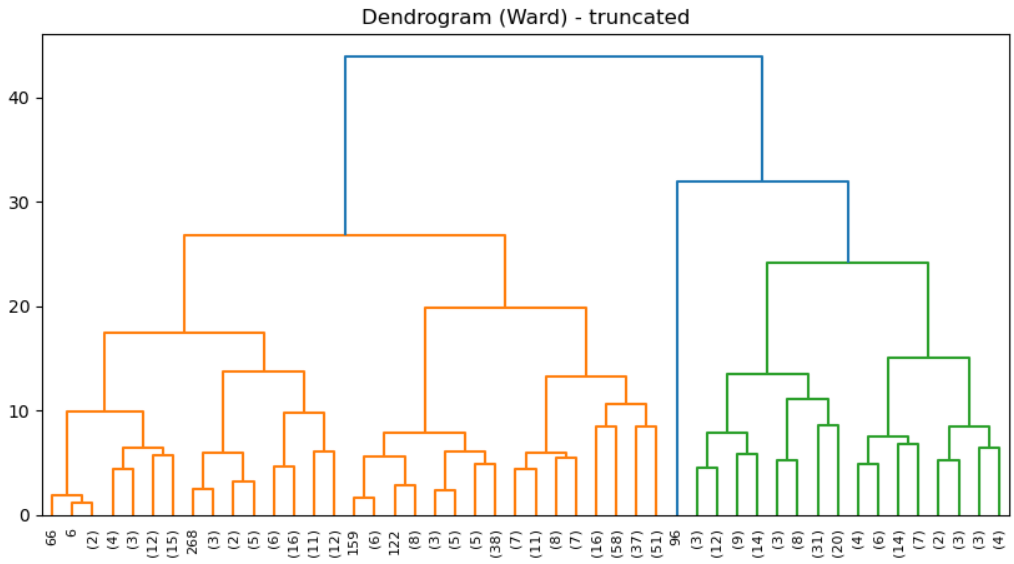



Figure 3. Truncated Ward dendrogram using a 500-point sample. Several large merges occur near the top, suggesting 4–6 meaningful clusters.

## DBSCAN

I tested density-based clustering using:

- eps = 1.3
- min_samples = 20

These were tuned experimentally. DBSCAN is useful for detecting outliers and clusters of arbitrary shape. I plotted DBSCAN results with PCA and labeled noise points in black.

## Evaluation Strategy

- Used silhouette scores to select K for K-Means.
- Compared cluster patterns visually through PCA.
- Used cluster profiles to interpret the meaning of each cluster (average price, size, etc.).
- Compared differences between the three methods.

---

# 3. Results

## K-Means

The elbow plot showed a smooth decline in SSE, but the silhouette scores showed a clear local maximum around K = 5, which is why I selected best_k = 5.

The PCA visualization showed clear, well-separated clusters. From the K-Means cluster profiles:

- One cluster grouped low-price, small apartments.
- Another contained mid-size, mid-price units.
- A high-end cluster contained large, high-price homes.
- Two mixed clusters combined mid-range apartments in different cities.

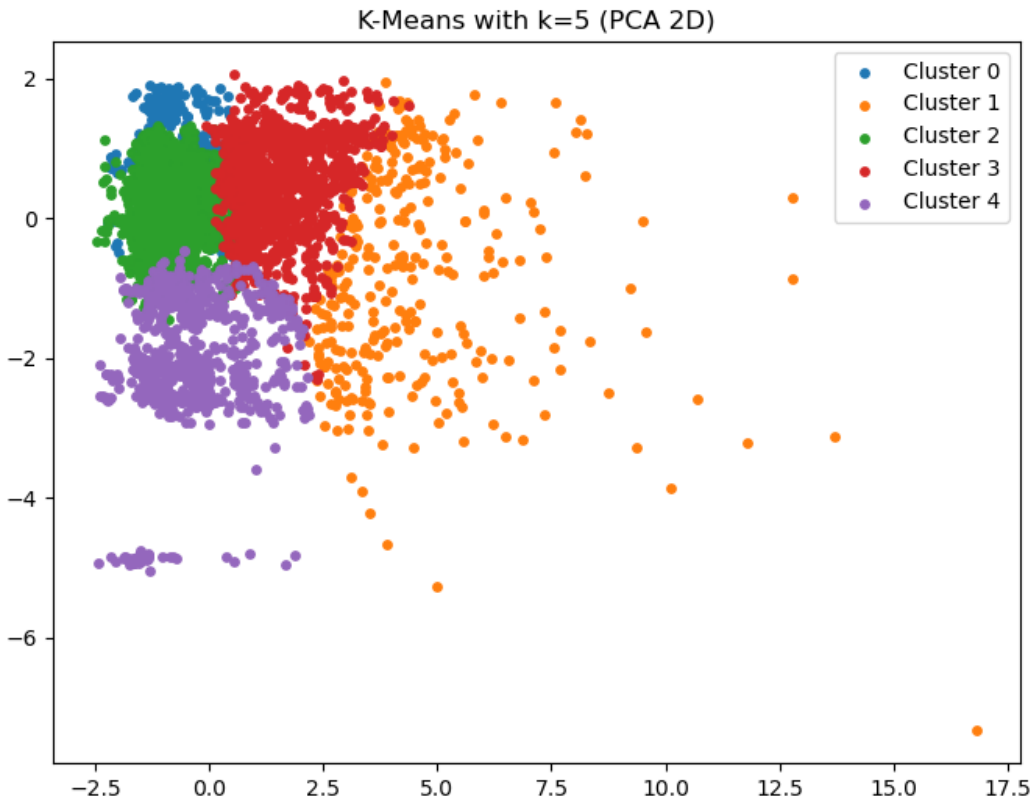
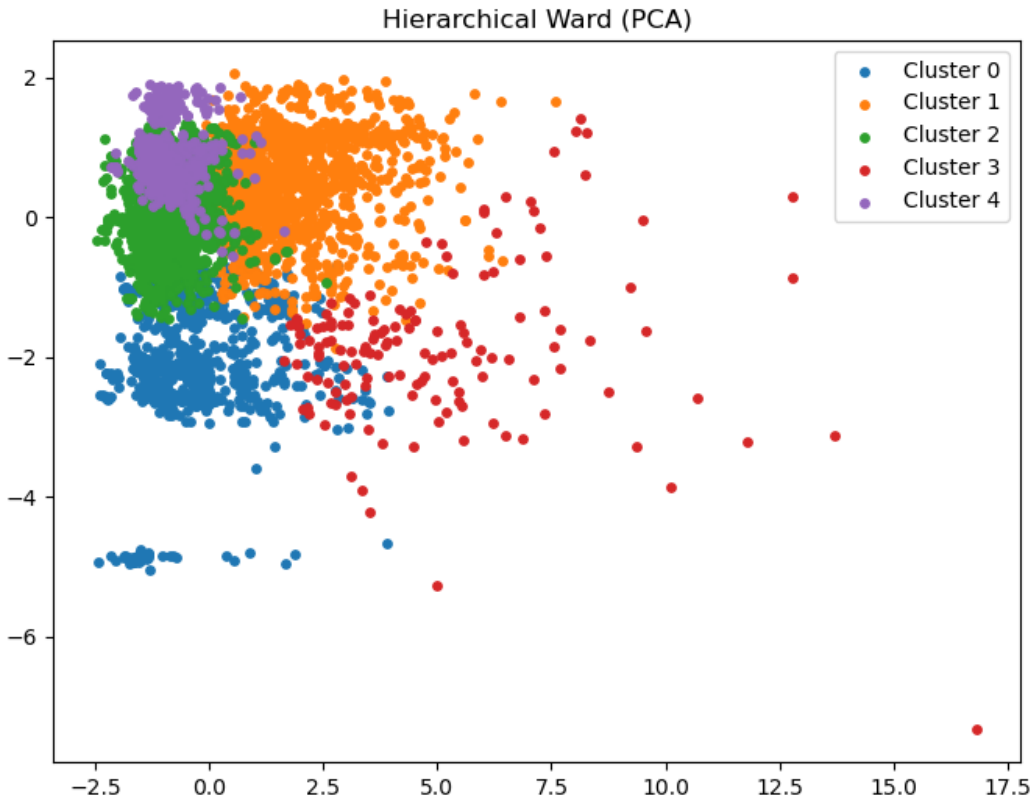



Figure 4. PCA projection of K-Means results with K=5. Clusters appear compact and well-separated in reduced 2D space.

## Hierarchical Clustering

Using Ward linkage and 5 clusters, the resulting clusters were similar to K-Means but slightly less balanced. The dendrogram showed strong separations in the highest merges, supporting around 4–6 meaningful clusters.

Hierarchical clustering confirmed the structure seen in K-Means.




## DBSCAN

DBSCAN identified:

- 3–4 dense clusters
- A noticeable number of noise points

Most noise points were either extremely expensive listings, very large properties, or geographically isolated units.

DBSCAN was useful for highlighting outliers that K-Means and hierarchical methods forced into regular clusters.

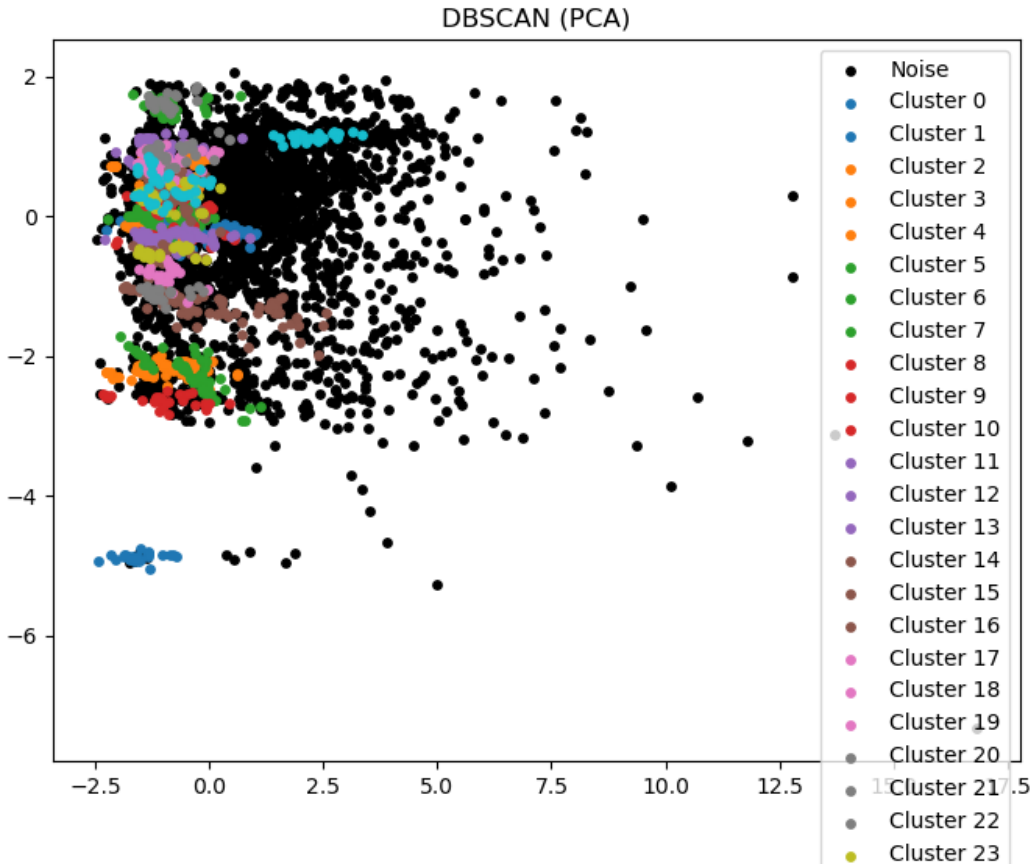



Figure 6. PCA visualization of DBSCAN clustering. Black points represent noise. DBSCAN finds several dense clusters and a significant amount of noise/outliers.

## Cluster Profiles

The K-Means cluster profile CSV showed:

- Cluster 0: low price, small square footage, mostly urban apartments.
- Cluster 1: medium price, moderate size, typical 1–2 bedroom units.
- Cluster 2: high square footage and high price (luxury units).
- Cluster 3: mixed properties with varied locations.
- Cluster 4: highest prices and large homes, likely suburban or premium listings.

These results confirm that clusters mainly differ by price, size, and geography.

kmeans_profiles

| kmeans_cluster | price | square_feet | bathrooms | bedrooms | latitude | longitude | count |
|---|---|---|---|---|---|---|---|
| 0 | 1064.7 | 698.24 | 1.02 | 1.18 | 31.6 | -94.86 | 1113 |
| 1 | 3295.98 | 2256.36 | 2.73 | 3.61 | 37.49 | -103.94 | 306 |
| 2 | 1162.78 | 729.27 | 1.01 | 1.4 | 40.78 | -86.07 | 2094 |
| 3 | 1516.01 | 1327.22 | 2.03 | 2.58 | 36.54 | -88.04 | 1401 |
| 4 | 1879.33 | 745.48 | 1.19 | 1.39 | 41.88 | -121.51 | 861 |

---

# 4. Conclusions

The dataset naturally forms meaningful groups that reflect real patterns in the U.S. rental housing market. K-Means with five clusters produced the most interpretable structure, dividing rentals by size, price range, and location. Hierarchical clustering supported these findings, while DBSCAN highlighted outliers and rare listings.

Overall, this project demonstrated how clustering can reveal structure in unlabeled real-world data. It also showed how preprocessing choices (scaling, encoding) and hyperparameter selection affect clustering behavior. The methods used here can be applied to other market datasets where categories and numerical values interact.

---

# 5. References


1. UCI Machine Learning Repository – Apartment for Rent Classified Dataset
   https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified
2. Scikit-learn documentation: Agglomerative Clustering (Ward linkage)
   https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering
3. Scikit-learn documentation: K-Means, DBSCAN, PCA, Preprocessing
   https://scikit-learn.org/stable/
4. CPS803 Lecture Slides – Units 6 & 7