MSDS 411: Unsupervised Learning

Assignment 2

July 27, 2025

Comparative Study of Clustering Analyses

Yingyu Mao

Abstract

In this paper, I presented a comparative study of various unsupervised clustering techniques applied to a real estate dataset to identify distinct housing clusters. Utilizing nine property attributes as instructed in Assignment 2, the study aims to assess whether these attributes can effectively predict house types or locations. The dataset underwent thorough cleaning, including handling missing values by dropping complete entries, which was validated as representative through Kullback-Leibler divergence and correlation matrix comparisons. Outliers were identified using both Local Outlier Factor (LOF) and Isolation Forest (IF) algorithms, and common outliers were removed. I then compared K-means clustering, hierarchical clustering (with various linkages), Gaussian Mixture Models (GMM), and Mean Shift clustering. The effectiveness of these algorithms was evaluated using silhouette scores for hyperparameter tuning and visualization in Principal Component (PC) and UMAP spaces. Results indicate that K-means tends to divide data evenly to minimize within-cluster variance, while hierarchical clustering with average or single linkage highlighted anomalies. GMM produced a "striping pattern" potentially due to discrete features, and Mean Shift successfully identified minority groups of high-priced homes and large land sizes. Ultimately, the chosen attributes did not reliably predict house types or locations, suggesting the data in the 9-dimensional feature space likely originates from a single Gaussian distribution.

Keywords: clustering analysis, K-means, hierarchical clustering, Gaussian Mixture Models, mean shift clustering

1.0 Introduction

Clustering analysis is a powerful unsupervised machine learning technique used to group similar data points into distinct categories based on their features. In the real estate industry, clustering can help identify different types of houses by analyzing relevant property attributes such as square footage, number of bedrooms and bathrooms, location, price, and amenities.

For a real estate firm, understanding housing clusters can enhance marketing strategies by tailoring promotions to specific property segments. It can also assist in efficiently assigning territories to realtors based on neighborhood housing trends and property valuation by comparing houses within the same group, ensuring accurate appraisals and identifying undervalued investment opportunities.

In this paper, I will compare multiple clustering methods for identifying clusters of a housing dataset using 9 property attributes as instructed in Assignment 2. Practically, the house type and location information should be easily gathered instead of predicted. Logic suggests that it is reasonable to predict the price or size of a house based on its type and neighborhood but not the other way around, since similar prices could be seen in different neighborhoods or for different house types. But for exploratory analysis purposes, it is interesting to find out if the attributes used can identify house clusters that match their types or locations.

2.0 Methods and Literature Review

2.1 Data cleaning and exploratory analysis

The Melbourne housing market dataset was obtained. The following features were selected as instructed for further analysis: `"Rooms", "Price", "Distance", "Bedroom2", "Bathroom", "Car", "Landsize", "BuildingArea", "YearBuilt"`.
To evaluate the clusters by comparing with the `“Type”` and `“Latitude”/”Longitude”` features, I separately pulled these columns out of the original datasets. They were not included in model training but only used for evaluation purposes.
A look at the data statistics shows some anomaly with the maximum value of `YearBuilt` is 2106, which is in the future. This is likely a typo, so I changed the record to 2016. The minimum of `YearBuilt` is 1196. This could be a true observation, so I decided to not alter this record but let the downstream processes take care of it.

2.1.1 Deal with missing values

The percentage of missing values in this dataset is high. Only 8895 entries out of 34857 entries will remain after dropping all NAs. This is a big chunk of data loss (~75%). The assignment instruction asked to work only with complete entries. I want to show the reason for dropping all data points with NAs. If I am performing supervised learning, I would choose to set a threshold to keep entries with less than 2 NAs and try different imputation methods to predict these values. If the learning process is supervised, I can then use a test set to evaluate different imputers with model performance. But since we are performing cluster analysis with no labeled data, I need to mask some values in the complete 8895 entries and use mean squared error to evaluate different imputation methods by comparing the predicted values and the masked values. Before performing all this, I chose to calculate whether the 8895 set is representative of the entire sample set. The metric I used to compare the distribution of each feature before and after dropping all NAs is simply Kullback-Leibler divergence. Since these ranges in (0.012, 0.0306), the 25% complete data is representative of the original distribution. I also compared the differences between the correlation matrix before and after dropping all NAs, the mean squared error is only 0.0217, which is ignorable. This means that dropping all data points with any NA won't change the correlations between the features, thus won't change the principal component space of the dataset. Therefore, instead of going through selecting an imputer with a test dataset, I think it's reasonable to drop all entries with NAs. The remaining 25% complete entries are representative of the bigger sample set.

Based on common sense, the variables `Rooms, Bedroom2, Bathroom, Car` are not continuous data but integers. However, since they are all ordinal datatypes, I can treat them as continuous.

2.1.2 Deal with outliers

Exploring the data distribution across features with boxplot, I see quite a lot of outliers for each feature. Instead of imputing values based on univariate outlier detection approach (1.5 or 3 IQR below or above first or third quartile), I chose two multivariate methods for identifying outliers.

`LocalOutlierFactor` calculates an anomaly score for each sample, which measures the local deviation of the density of a given sample with respect to its k-nearest neighbors. Samples that have a substantially lower density than their neighbors are outliers (Breunig 2000).

`IsolationForest` detects anomalies by recursively partitioning the data randomly across the feature space. The number of splits required to isolate a sample, or path length, can be averaged among a forest of random trees. Outliers have much shorter path lengths than the normal range of path lengths (Liu 2012).

The former used Euclidean distance as the distance metric for determining the neighborhood and the reachability between objects, which means it is sensitive to scaling. I first scaled the data with MinMaxScaler, since it preserves the shape of the original data distribution and does not change the sign of the data values, which are all positive for all features. I then estimated the percentage of outliers roughly using univariate distributions. With `n_neighbors=20, contamination=0.05`, LOF found 445 outliers.

IF is a tree-based algorithm and is therefore scale-invariant. IF can be used to fit the raw values of the dataset. With the same contamination estimate, IF found 445 outliers as well.

The common outliers (72 identified) from the two algorithms were omitted from the dataset for further analysis.

2.2 Clustering algorithms

Grouping similar data points into clusters is performed on their proximity or similarity, measured using distance metrics such as Euclidean, Manhattan, cosine, or Jaccard distance. The goal is to ensure that points within the same cluster are closer to each other than points in other clusters. Popular distance-based clustering algorithms include K-means and hierarchical clustering. K-means partitions data into K clusters by minimizing within-cluster variance, while hierarchical clustering builds a tree-like structure of clusters using agglomerative or divisive approaches. K-means clustering is faster than hierarchical clustering when working with large datasets, although it is sensitive to initiation and needs to specify the number of clusters beforehand. K-means++

implemented in scikit learn selects initial cluster centroids using sampling based on an empirical probability distribution of the points' contribution to the overall inertia and improves convergence (Arthur and Vassilvitskii 2007).

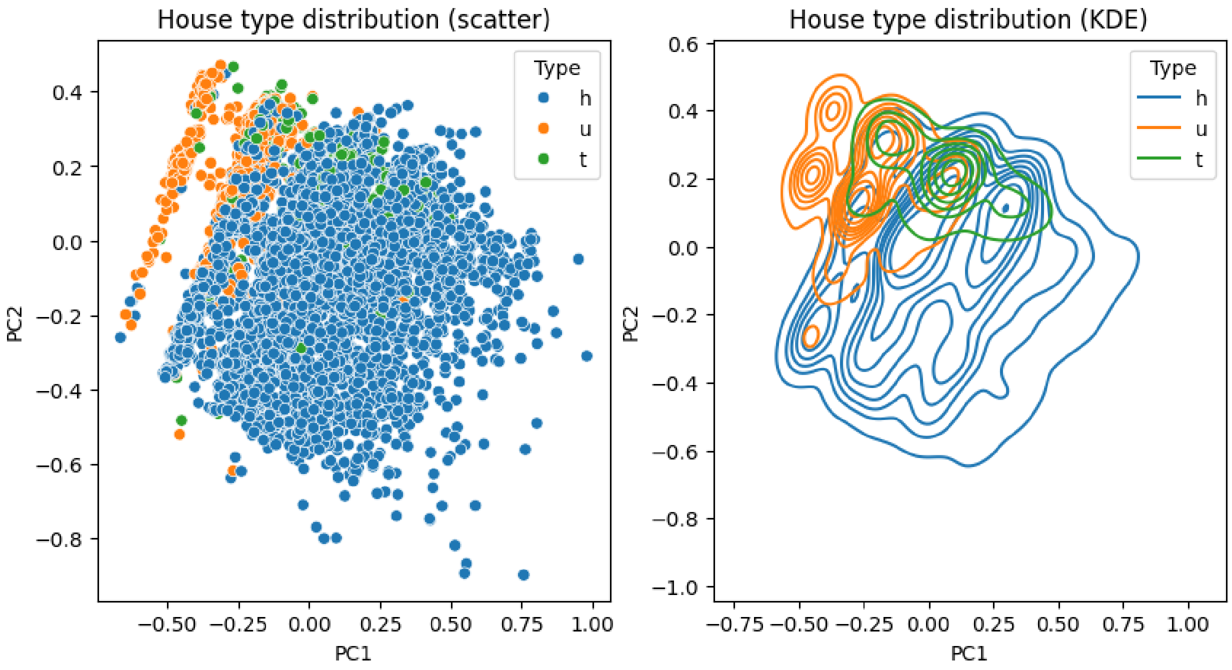
Other limitations of K-means are difficulty handling non-spherical clusters and varying cluster sizes and densities. Gaussian mixture models (GMM) take these factors into account. It assumes that data points are generated from a mixture of several Gaussian distributions, and the shape, size, and density of a cluster are represented by the covariance matrix. Expectation-Maximization (EM) optimization algorithm is used to fit GMMs, iteratively estimating cluster parameters (means, covariances, weights). The E-step computes the probability of each point belonging to each cluster, while the M-step updates the model parameters to maximize likelihood, repeating until convergence (Dempster, Laird and Rubin 1977).

GMM assumes gaussian distribution and requires specifying the number of clusters beforehand, or this needs to be estimated somehow. Mean shift clustering is also distribution-based but does not require specification of the cluster numbers, or that the clusters are generated from a Gaussian distribution. Mean shift estimates the multivariate distribution of the data by kernel density estimation with a flat or Gaussian kernel. It then identifies the modes of the distribution as the centers of clusters through a gradient ascent algorithm.

In this paper, I will train and compare the outcomes of these clustering algorithms on the housing dataset.

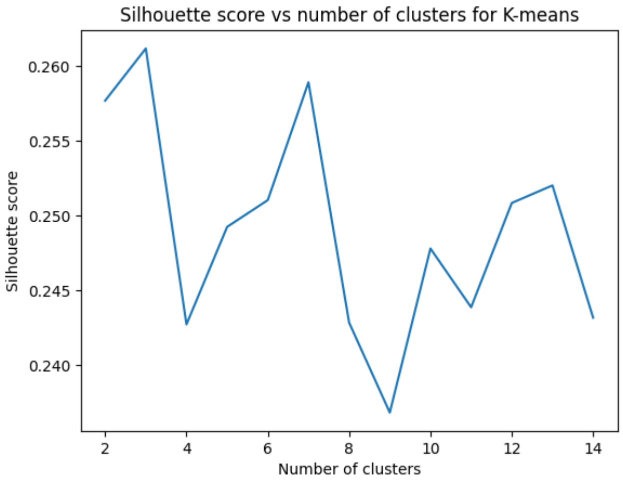
3.0 Model Training and Results

The dataset without the outliers was first scaled using a MinMaxScaler. To best illustrate the different clustering in a coordinate system, I then did a full principal component projection, preserving all the variances in the dataset. This is equivalent to rotating the data matrix from its original feature coordinates to its principal component coordinates. This operation will not change the outcome of clustering. The first two principal components explain about 70% of the variance. Visualizing the distribution of the data points in the principal component space and labeling the housing type in a standard scatter plot and a kernel density estimate (KDE) plot, it shows that there is a significant overlap between townhouses and houses and most units. Only some 'unit' type properties might be distinguishable from the rest of the data.

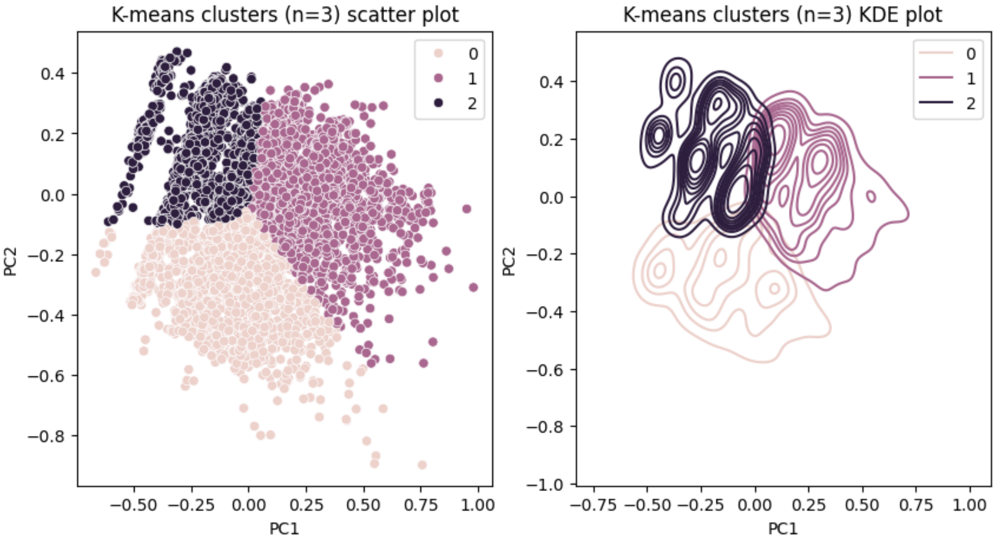


3.1 K-means Clustering

Silhouette scores were used to determine `n_clusters` hyperparameter of K-means clustering. Three clusters result in the highest average silhouette score, indicating the best fit. The Silhouette score is plotted against the number of clusters as shown below.



The distributions of the clusters are plotted below with both scatter and KDE plots.

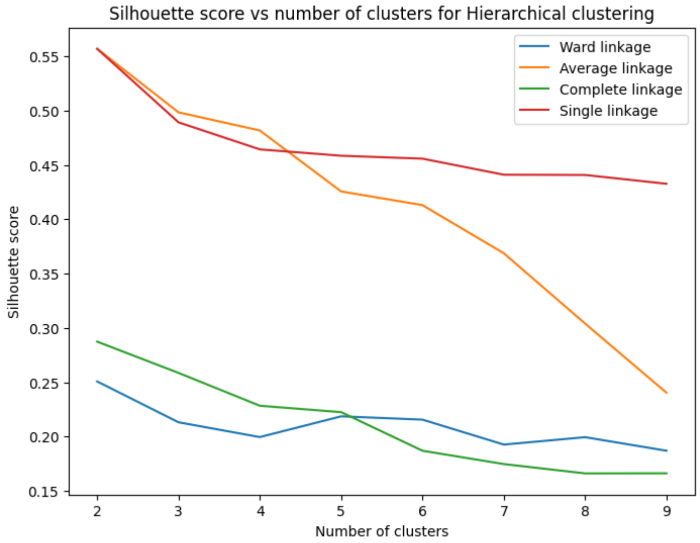


From these, it is noticed that the algorithm tried to divide the dataset in the middle into 3 almost even clusters. As the first two components explain roughly the same percentage of variance, dividing the data this way into three clusters minimizes the within-cluster variance. We can also see that this distribution pattern deviates quite a bit from that of the house type. This can also be indicated by checking the house type value counts within each cluster.

```
cluster 0
h    2081
u      51
t      18
cluster 1
h    2912
t     426
u     102
cluster 2
h    1594
u    1357
t     281
```

3.2 Hierarchical Clustering

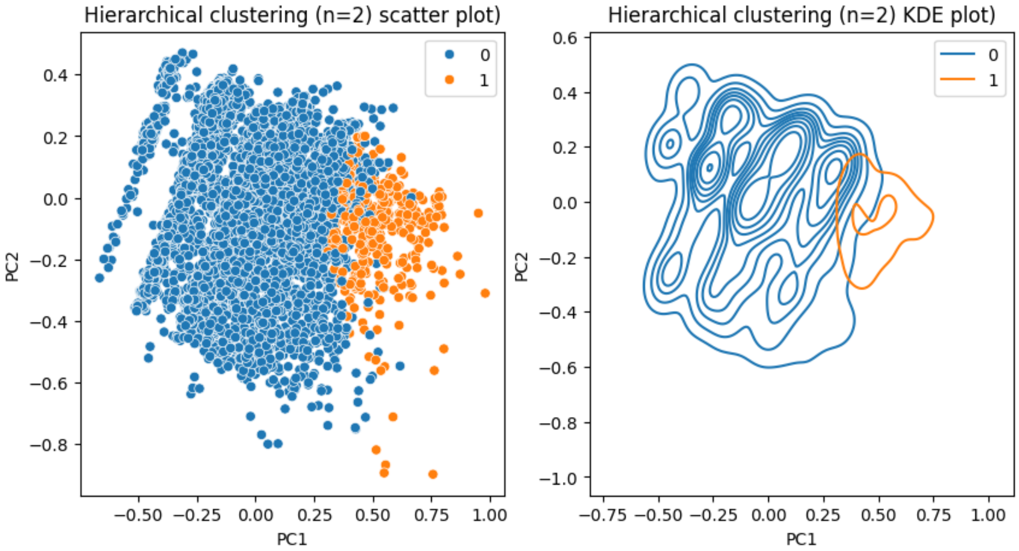
As k-means, Euclidean distance metric was used for similarity measures for this numerical dataset. The Ward linkage method minimizes the within-cluster variance and is usually suitable for spherical clusters as well as compact clustering outcomes, which is similar to that of k-means. To learn a different clustering structure, I compared all linkage methods for this dataset. Silhouette scores were then used to determine `n_clusters` of agglomerative hierarchical clustering. This parameter is not required for hierarchical clustering itself, as the grouping of data points starts from bottom up and iterates until all data points are clustered. But the implementation in Scikit learn needs this parameter for trimming the dendrogram. The average silhouette scores for different numbers of clusters are shown below. While Ward linkage resulted in similar silhouette scores as those of k-means (since both reduce within-cluster variance), average and single linkage significantly improved this metric, indicating better fit.



The models were retrained with `n_clusterst=2` and the Ward or average or single linkage. This reveals that both average and single linkage classified house 16424 in a single cluster. This incident is centered on PC1 and PC2 but is off the center on PC3, 4, 5 and 6, which when combined, explains almost 30% of the total variance. After checking its original data, I found that its anomaly stemmed from the fact that it's built in 1196, which is the oddly low minimum value of `YearBuilt` as mentioned in section 2.1. It was not the common outlier identified by the outlier detection methods. However, hierarchical clustering detects it again. The high silhouette score, when this incident is in a cluster by itself, from another angle shows that a model that is close to a one-cluster model fits the data much better than the two-cluster model (Silhouette score cannot be calculated with one cluster, so a direct comparison would be unattainable.).

After excluding house 16424, I plotted the dendrograms using average, single, complete and Ward linkage (data not shown but can be found in the Jupyter notebook file). There are still several low-count clusters that are at the top of the hierarchy in the dendrograms of average and single linkage, which are potentially outliers depending on the judging criteria. The dendrogram of Ward linkage is more balanced overall. Comparing different outlier detection algorithms is another topic that does not match the goal of this study and will not be further discussed.

Since complete linkage resulted in better silhouette score, it was used to build the final model. The two clusters separate a majority of the incidents from a group located on the right along the PC1 axis. The distribution of data points of each cluster is shown below with a scatter and KDE plot.



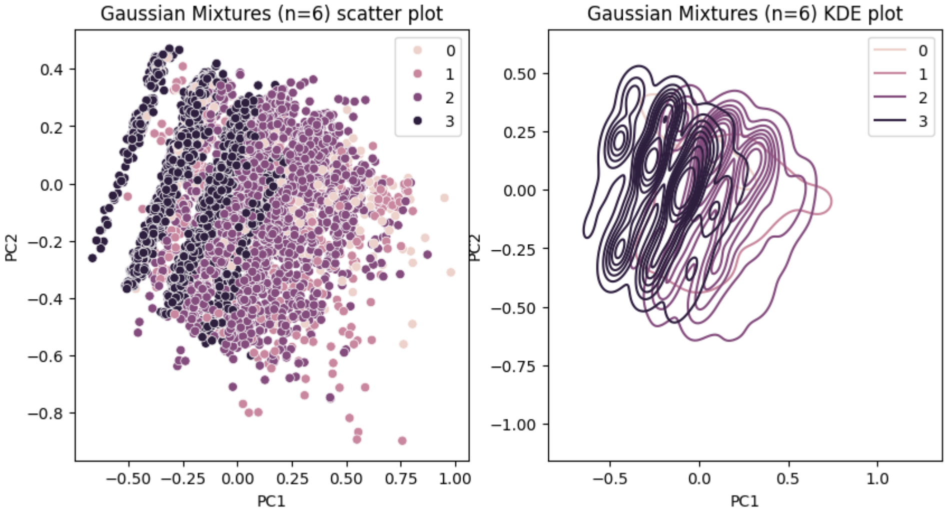
The house type value counts reveals that cluster 1 is mostly houses, likely sharing certain similar characteristics.

```
cluster 0
h    6145
u    1509
t     716
cluster 1
h    442
t      9
u      1
```

3.3 Gaussian Mixtures

Based on the analyses done so far, it appears that the data is likely generated from a single Gaussian process or if several Gaussian processes, their distributions are very similar. This could cause challenges for the EM algorithm to unmix the data. The parameters for Gaussian mixture algorithm implemented in Scikit learn, `n_components`, `covariance_type`, were tuned with a parameter grid search for lowest Bayesian information criterion (BIC) score. The best parameters are `covariance_type='full'`(each component has its own general covariance), and `n_components=4`.

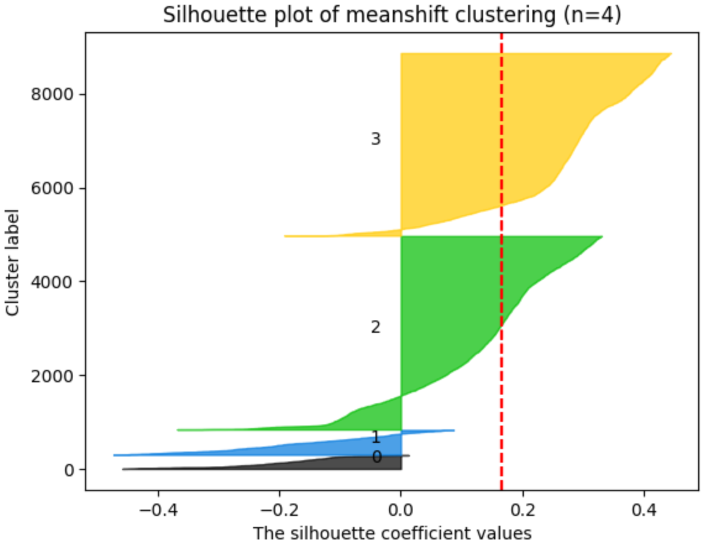
The distribution of the 4 clusters is shown below in a scatter plot and a KDE plot. It interestingly generates a striping pattern of clusters, which might be a result of the discrete nature of some features.



The house type value counts within each cluster also does not correlate well with house type:

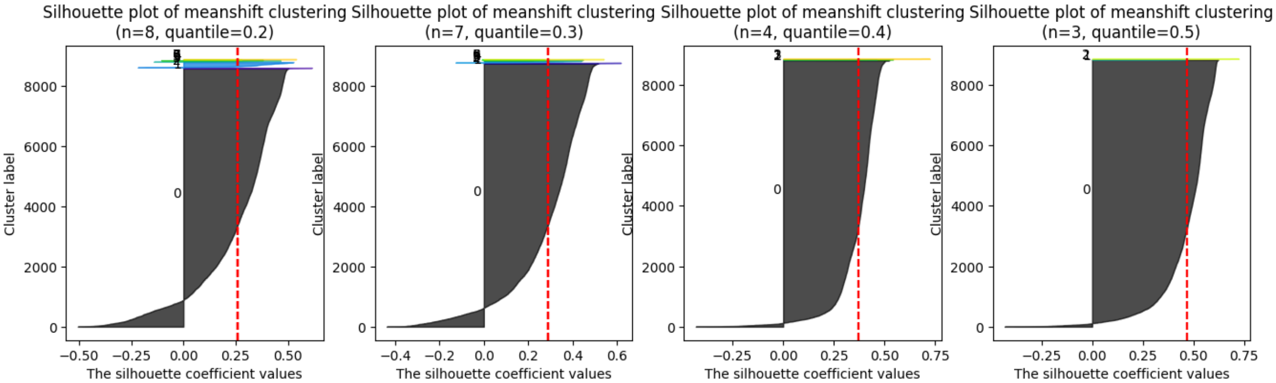
```
cluster 0
h    178
u     97
t     16
cluster 1
h    459
u     40
t     29
cluster 2
h    3397
t     490
u     231
cluster 3
h    2553
u    1142
t     190
```

The Silhouette plot was used to check each cluster's quality. It shows most of the data points in cluster 0 and 1 are misclassifications (negative Silhouette scores).



3.4 Mean shift Clustering

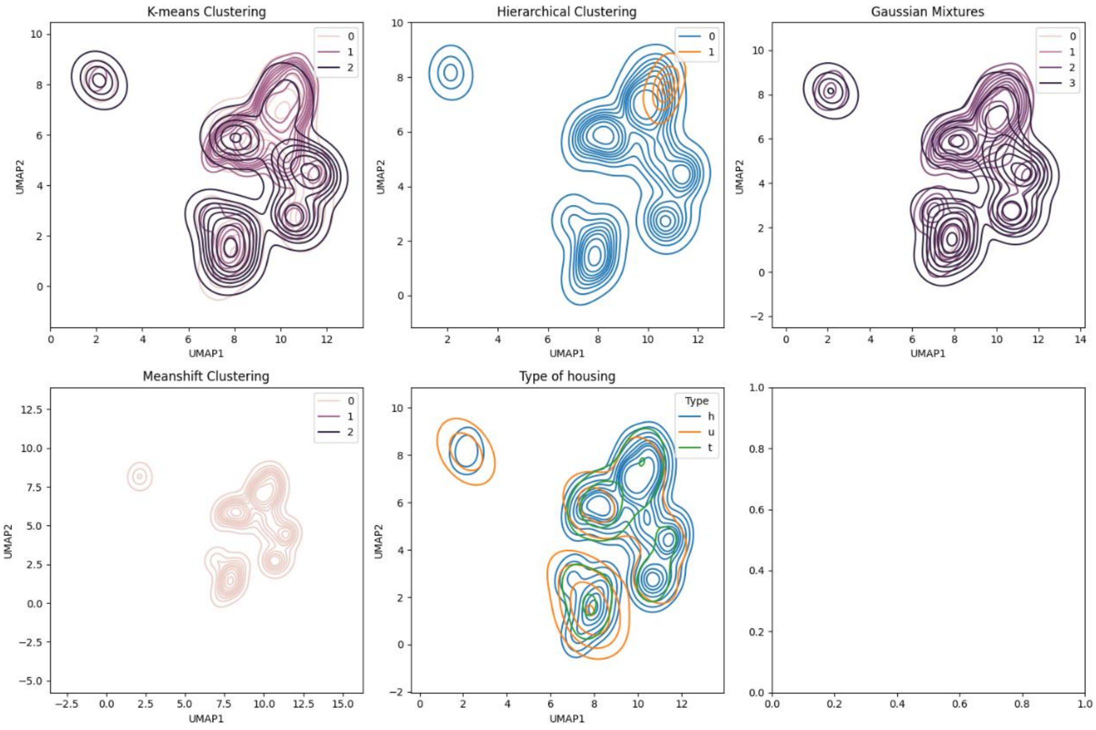
Since the mean shift algorithm is based on kernel density estimate, the parameter that affects model training is the bandwidth of the smoothing window. It defines the size of the neighborhood around each data point. Scikit learn uses `NearestNeighbors.kneighbors` function to estimate bandwidth as the average of the maximum distance in each batch of data with a preset quantile expected as neighbors from different batches. This means that a lower quantile models the data as sparser, while a higher quantile models the data as denser. Out of the `500` sampling size, I picked `0.2,0.3,0.4,0.5` as the expected percentage of neighbors to train the algorithm. As expected, lower quantile results in a higher number of small clusters due to smaller neighborhoods as shown below.



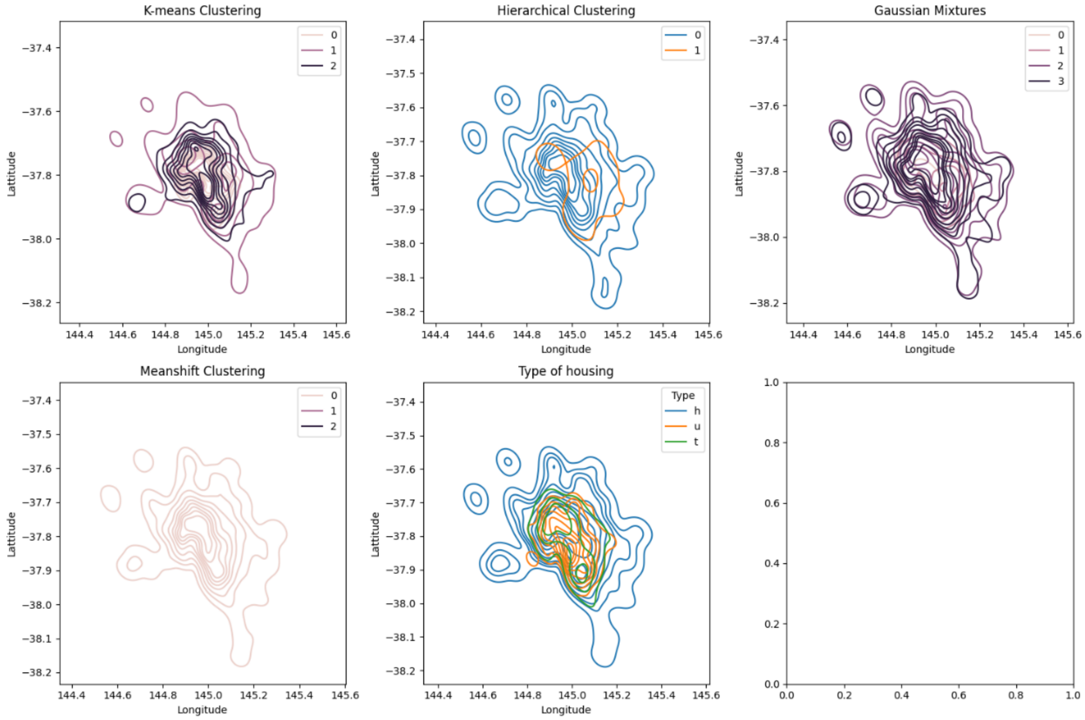
Silhouette plot of meanshift clustering (n=8, quantile=0.2)
Silhouette plot of meanshift clustering (n=7, quantile=0.3)
Silhouette plot of meanshift clustering (n=4, quantile=0.4)
Silhouette plot of meanshift clustering (n=3, quantile=0.5)

More interestingly, the clustering quality is much better when most of the dataset is called to be in one cluster, a similar finding as the hierarchical clustering with average or single linkage. After inspecting the incidences of the minority clusters, it appears that cluster 1 all seems to be high-priced homes ranging from 6M-9M, while cluster 2 all have relatively large `Landsize`. Since the mean shift algorithm detests modes of a multivariate density distribution, the algorithm seems to have successfully identified the two minority groups that distribute closely together.

Additionally, UMAP projection of the dataset into 2D space reveals different groupings of the data from the clustering methods used, as well as the type of houses, since there are significant overlaps of different house types and clusters within any UMAP cluster. The least overlap is seen in hierarchical clustering.



Comparison of the geometrical distribution of all clusters generated by different algorithms and house types within the longitude and latitude coordinates also shows significant overlaps of the clusters. In Hierarchical clustering, cluster 1 seems to have identified the ‘house’ type that are close to the business district of Melbourne. The 'townhouse' and 'unit' house types do concentrate on the business district.



In conclusion, the variables suggested by the assignment instruction cannot reliably predict the house types or the location of the house. The data is likely drawn from the same Gaussian distribution in the 9-dimension feature space of the features selected by the assignment instruction.

4.0 Discussion

The objective of this study was to evaluate the efficacy of various unsupervised clustering algorithms in identifying distinct house clusters within a real estate dataset, specifically examining if the chosen property attributes could reliably predict house types or locations. The analysis revealed that despite applying robust data cleaning and outlier detection methods, the selected nine property attributes ("Rooms", "Price", "Distance", "Bedroom2", "Bathroom", "Car", "Landsize", "BuildingArea", "YearBuilt") did not allow for a clear differentiation of house types or geographical locations.

K-means clustering, while effectively minimizing within-cluster variance, tended to divide the dataset into three almost even clusters in the principal component space, which did not align well with actual house types. This suggests that the features, when analyzed by K-means, form spherical clusters that do not naturally correspond to the inherent categories of 'house', 'unit', and 'townhouse'.

Hierarchical clustering, particularly with average and single linkages, demonstrated a propensity to isolate individual or very small clusters. While complete linkage yielded better silhouette scores and produced two clusters, one of which predominantly contained 'house' types, the overall separation based on house type or location remained limited. This indicates that while hierarchical clustering can reveal outliers or distinct small groups, its ability to generalize to meaningful, larger-scale distinctions based on house type or location using these features is constrained.

Gaussian Mixture Models, when optimized for the lowest Bayesian Information Criterion (BIC), resulted in four clusters with a "striping pattern". The silhouette analysis for GMM indicated significant misclassifications, particularly in clusters 0 and 1, with negative silhouette scores. This performance suggests that assuming a mixture of several Gaussian distributions for the given data, while accounting for varying shapes, sizes, and densities, did not effectively unmix the data into distinct house type or location-based clusters. This could be due to the possibility that the data is generated from a single Gaussian process or several very similar ones, making it challenging for the EM algorithm to differentiate.

Mean Shift clustering, which does not require a predefined number of clusters and is based on kernel density estimation, identified minority groups corresponding to high-priced homes and properties with large land sizes. This demonstrates its effectiveness in detecting modes in the data distribution, suggesting that these specific characteristics form denser regions in the feature space. However, similar to hierarchical clustering, the mean shift algorithm also indicated that a model with most of the data points in a single cluster provided better quality, reinforcing the idea of a dominant underlying distribution.

The visualizations using Principal Components and UMAP further supported the conclusion that there is significant overlap between different house types and the clusters generated by all methods. While hierarchical clustering showed the least overlap in the UMAP projection, and some geographical concentration of 'house' types was observed with hierarchical clustering, a comprehensive and reliable prediction of house types or locations was not achieved.

In conclusion, the chosen property attributes, as instructed by the assignment, do not provide sufficient information for clustering that reliably predicts house types or locations. This suggests that the data, in the 9-dimensional feature space, is likely drawn from a single, broadly distributed Gaussian process rather than distinct, separable Gaussian processes corresponding to different house types or geographical segments. Future work could explore incorporating additional features, such as more granular location data or neighborhood-specific characteristics.

References

Arthur, D., and S. Vassilvitskii. 2007. "k-means++: the advantages of careful seeding." *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* 1027–1035.

Breunig, Markus M. and Kriegel, Hans-Peter and Ng, Raymond T. and Sander, J\"{o}rg. 2000. "LOF: identifying density-based local outliers." *SIGMOD Rec.* 93–104.

Dempster, A.P., N.M. Laird, and D.B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society, Series B* 39 (1): 1–38.

Liu, Fei Tony and Ting, Kai Ming and Zhou, Zhi-Hua. 2012. "Isolation-Based Anomaly Detection." *ACM Trans. Knowl. Discov. Data* 1-39.