# Data Mining Homework 3

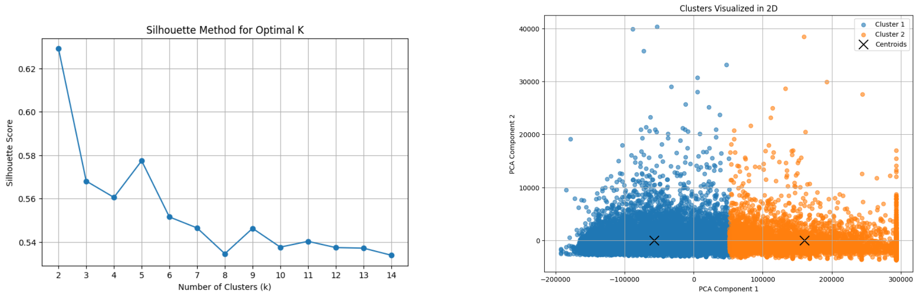
Alessio Maiola

## 1 Clustering on raw data

I implemented the Silhouette method to find the optimal number of clusters and chose K-Means++ as the clustering algorithm. The only transformation i applied to raw data was indexing the categorical ocean_proximity feature.

I evaluated the results of clustering, using the optimal number of clusters (2), getting a clear separation based on the feature with the highest variance, which is median_house_value.

The clusters were visualized thanks to the application of PCA to the raw data, by reducing them to two dimensions. The scikit-learn implementation I used did not rescale the features, maintaining the original variance and therefore the original separation of the raw features, which is evident in Figure 1.



(a) Silhouette scores of clustering on raw data.

(b) Clustering on raw data visualized using PCA.

Figure 1: Clustering on raw data.

## 2 Exploratory Data Analysis

At this point, I conducted some exploratory data analysis. Strong correlations were found between any pair containing the following fields: population, households, total_rooms, total_bedrooms. Moreover, these fields have a weak negative correlation with housing_median_age, indicating that the most densely populated areas usually have more modern households. Furthermore, median_income and median_house_value had a strong correlation as well.

After one-hot encoding of the categorical ocean_proximity feature, I noticed that the correlation with the median_house_value grew together with the proximity to the ocean, indicating that those households are the most expensive.

I also checked out the distributions of the data. The four strongly correlated fields above mentioned have a normal distribution, as well as median_income and median_house_value. The field housing_median_age resembled instead more of a uniform distribution, rather than a normal one.

## 3 Experimenting Feature Engineering techniques

I conducted five separate experiments on the data, always comparing the performance on clustering on the raw data and comparing the clusters in the normalized feature space, but also doing the opposite, clustering the data in the normalized feature space and checking whether it generalized also in the unscaled features space.

First of all, in all the experiments, despite the dimensionality increase or reduction, the optimal number of clusters estimated using silhouette was consistently 2. Moreover, all my experiments did not significantly impact the running time of K-Means++ algorithm, always around $C * 10^{-4}s$ with $0 < C < 1$.

My experiments were the following:

**Experiment 1: Increasing Feature Space** Expanded the dimensionality of the dataset by creating or extracting new meaningful features to enhance the feature space since the initial dimensionality was low.

**Experiment 2: Employing Dimensionality Reduction Techniques** Applied PCA and Random Projections to reduce noise by projecting data onto a lower-dimensional space.

**Experiment 3: Feature Selection** Removed some of the initial features to reduce noise and potentially irrelevant information in the dataset.

**Experiment 4: Re-expanding Feature Space** Increased the feature space again by generating additional features, aiming to capture more complex patterns or relationships.

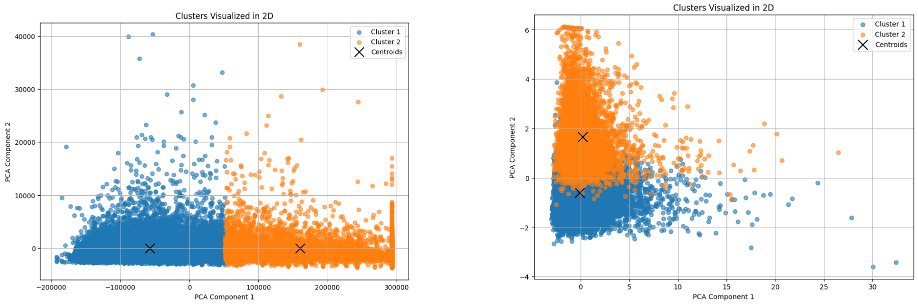
**Experiment 5: Dimensionality Reduction (again)** Re-applied dimensionality reduction techniques to compress the feature space while retaining the most significant information.

In my opinion, what mostly improved the clustering performances was removing unnecessary features in Experiment 3.

In particular, I transformed latitude and longitude fields into a more significant feature, representing their distance from the average values of both fields (the center of the points in those dimensions).

After this transformation, the most separated clusters in both spaces came up, with the silhouette methods always suggesting to cluster data into two groups. The following results refer to applying clustering in the raw unscaled dimensional space of the features, checking also how the clustering generalizes in the normalized feature space.

The separation in the normalized space was significantly improved, compared to the initial benchmark, which can be found in the notebook.

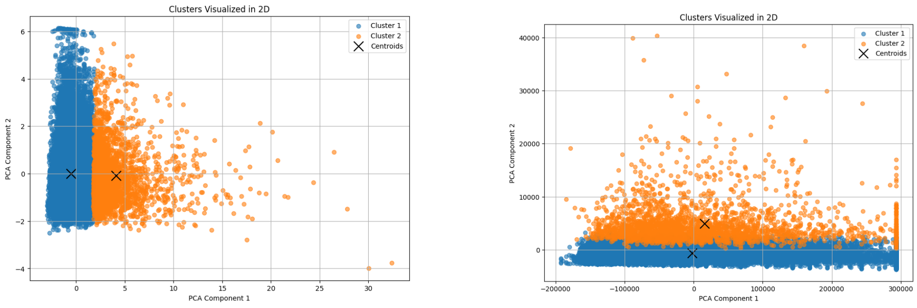


(a) Clustering visualized on raw data.

(b) Clustering visualized on normalized data.

Figure 2: Clustering on raw data in Experiment 3.

The clustering on normalized data, despite separating the data across a different dimensions, managed to clearly divide the points into clusters also in the unscaled raw features space. This was a surprising and satisfying result.



(a) Clustering visualized on raw data.

(b) Clustering visualized on normalized data.

Figure 3: Clustering on normalized data in Experiment 3.

The feature engineering techniques I applied aimed to make this separation sharper and more evident. The new features were introduced based on a number of experiments and performance checking, which required a lot of time. Unfortunately, most new features one could come up with introduced additional noise, despite trying to separate the data in a more evident way.

Finally even dimensionality reduction using PCA did not prove to be so useful in removing the noise in the augmented features (even getting rid of more than half of them). Conversely, Random Projections seemed to introduce distortion on the data, introducing even more noise and lowering K-Means++ clusters' quality.

To conclude, while these techniques may have improved clustering performance on one of the spaces, their clusters generalized worse on the rescaled normalized or not normalized feature space.