T.C.
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ

DATA SCIENCE
FINAL PROJECT

CLUSTERING

Merve YAVUZ
1621221003

FINAL PROJECT: CLUSTERING

Dataset definition: Dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months. The file is at a customer level with 18 behavioral variables.
Dataset source (web address): https://www.kaggle.com/arjunbhasin2013/ccdata
Aim of the project: Preparation of special opportunities for users in these groups by grouping credit card users according to their characteristics such as expenses and limits.

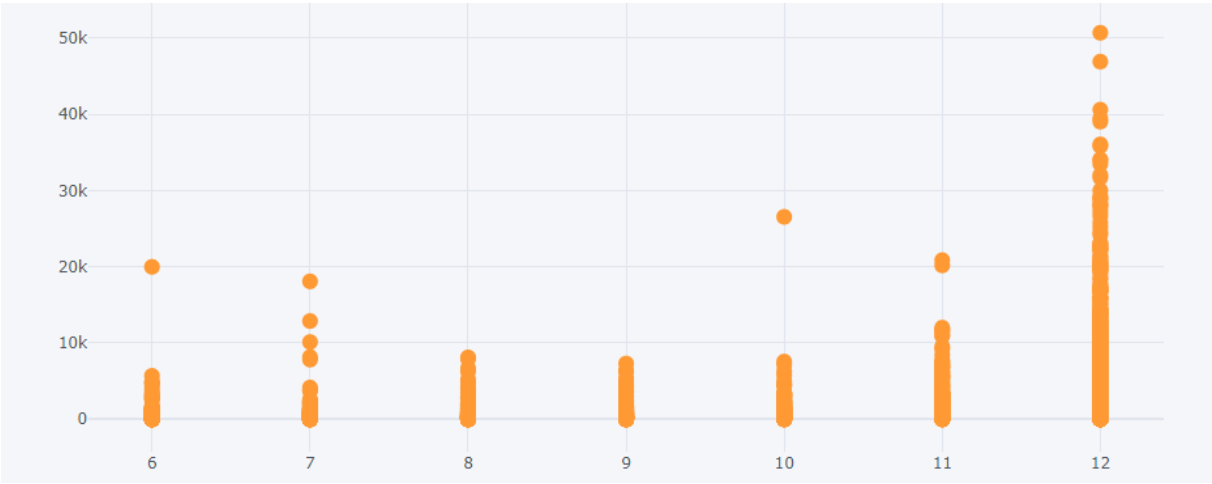
Exploratory Data Analysis

Dataset Shape
The dataset contains 17 columns containing 8950 data.

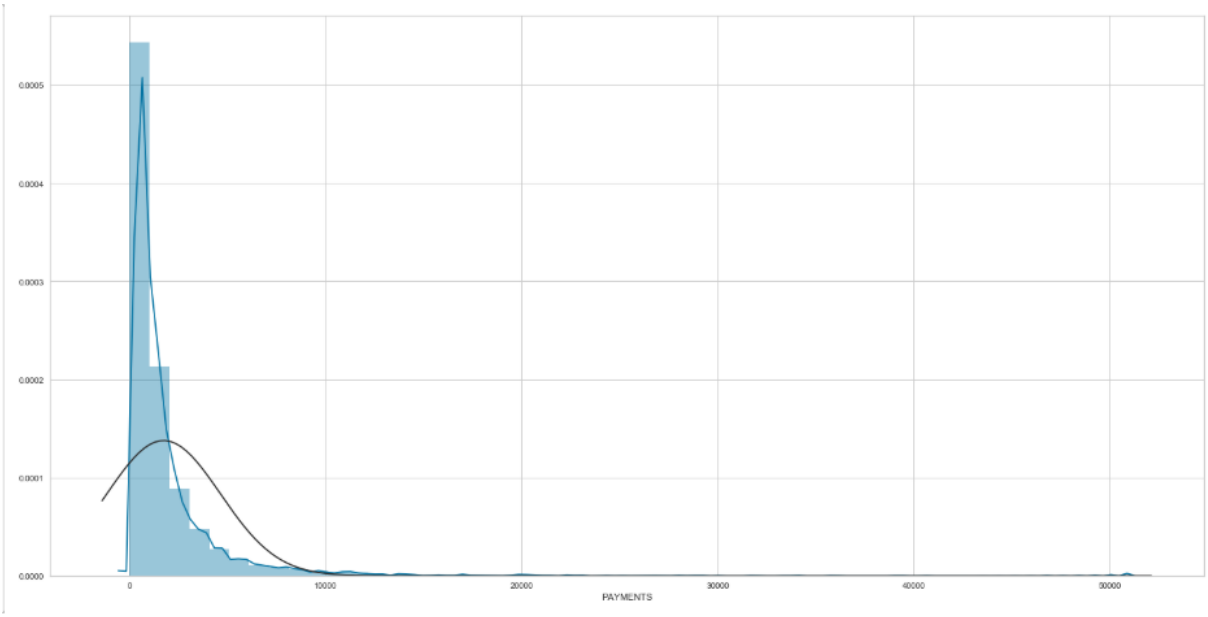
Column Types

```
BALANCE                             float64
BALANCE_FREQUENCY                   float64
PURCHASES                           float64
ONEOFF_PURCHASES                    float64
INSTALLMENTS_PURCHASES              float64
CASH_ADVANCE                        float64
PURCHASES_FREQUENCY                 float64
ONEOFF_PURCHASES_FREQUENCY          float64
PURCHASES_INSTALLMENTS_FREQUENCY    float64
CASH_ADVANCE_FREQUENCY              float64
CASH_ADVANCE_TRX                      int64
PURCHASES_TRX                         int64
CREDIT_LIMIT                        float64
PAYMENTS                            float64
MINIMUM_PAYMENTS                    float64
PRC_FULL_PAYMENT                    float64
TENURE                                int64
dtype: object
```

Explanation of Features
Tenure:
The duration of the credit card service for the user is 6,7,8,9,10,11,12 months. The majority of users enjoy 12 months of service.



Payments:
The normal distribution of the amount of payment made by the user is between 0 and 5000. Majority of credit card users do not make high payments.



Preprocessing

Missing Values
CREDIT_LIMIT has 1, MINIMUM_PAYMENTS has 313 missing values.

Handling Missing Values
Null data is filled by taking the average of the data in the numerical columns.

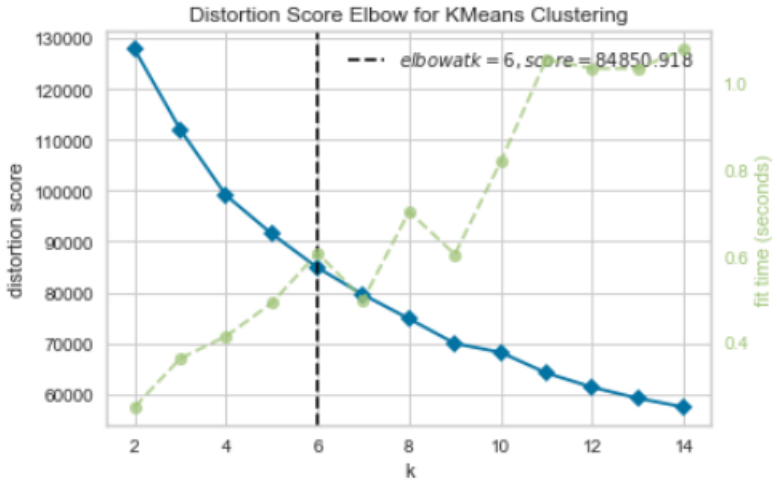
Transformations
StandardScaler was used to standardize. Standardize features by removing the mean and scaling to unit variance.

Clustering Evaluation

Clustering Evaluation Methods and Visualizations

K-Elbow:
The number of clusters is a previously reported clustering method. It aims to group K sets. It allows assigning the points to the most suitable cluster center.
The number K is determined. According to the K number, random cluster centers are determined among the points. All points have distances from cluster centers. The points are assigned to the set where it has the minimum distance. The cluster center is recalculated and distances are recovered. The process is repeated until the cluster center does not change. In this method, determining the number of K is an important problem. There are several methods to choose the appropriate K number.

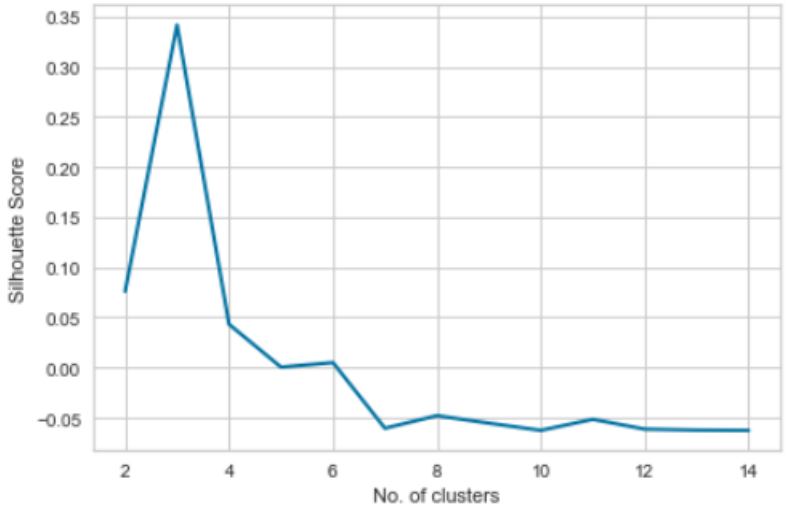
Elbow method; The sum of squares of the distance of the points to the cluster center is calculated according to each K value. A graph is drawn for each K value according to these values. The point of elbow where the difference between the totals starts to decrease on the graph is determined as the most appropriate K value.



Silhouette Score:
The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b). To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of. Note that Silhouette Coefficient is only defined if number of labels is 2 <= n_labels <= n_samples - 1.

$$s = \frac{b - a}{max(a, b)}$$

The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering. Scores around zero indicate overlapping clusters. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster. The Silhouette Coefficient is generally higher for convex clusters than other concepts of clusters, such as density based clusters like those obtained through DBSCAN.



Calinski and Harabasz Score:
The Calinski-Harabasz index also known as the Variance Ratio Criterion, is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters, the higher the score , the better the performances.
For a set of data E size of n_e which has been clustered into k clusters, the Calinski-Harbazs score s is defined as the ratio of the between clusters dispersion mean and within cluster dispersion.

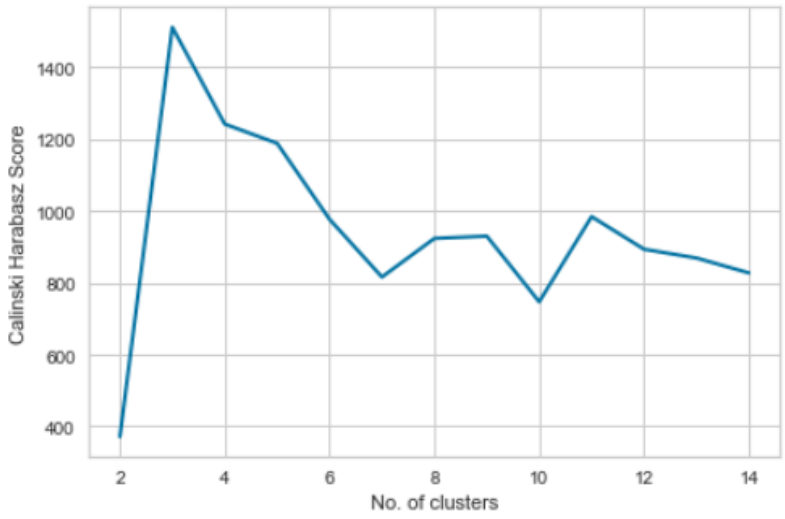
$$s = \frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)} \times \frac{n_E - k}{k - 1}$$

Where $\mathrm{tr}(B_k)$ is trace of between group dispersion matrix and $\mathrm{tr}(W_x)$ is the trace of the within cluster dispersion matrix defined by:

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

With C_q the set of points in cluster q, c_q the center of cluster q, c_e the center of E and n_q the number of points in cluser q.
The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster. The score is fast to compute.
The Calinski-Harabasz index is generally higher for convex clusters than other concepts of clusters, such as density based clusters like those obtained through DBSCAN.



Davies Bouldin Score:
This index signifies the average ‘similarity’ between clusters, where the similarity is a measure that compares the distance between clusters with the size of the clusters themselves. A lower Davies-Bouldin index relates to a model with better separation between the clusters.
S_i: Average distance between each point of cluster i and the centroid of that cluster.
D_{ij}: the distance between cluster centroids i and j.

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

The usage of centroid distance limits the distance metric to Euclidean space. The Davies-Boulding index is generally higher for convex clusters than other concepts of clusters, such as density based clusters like those obtained from DBSCAN.

Clustering Algorithms, Implementation and Performance Comparision

Clustering Algorithms

Kmeans:
The letter "k" in the name of the algorithm actually indicates the number of clusters: The algorithm also searches for the number of clusters that will minimize the Terrestrial Error Function commonly used in error calculation. The given “n” number data set is placed in “k” cluster to minimize this error function. For this reason, cluster similarity is measured by the approximation of the values in the cluster. This is the cluster's center of gravity. The value at the center of the cluster is the representative value of the cluster and is called the medoid.
Parameters need to be tuned:

- n_clustersint, default=8. The number of clusters to form as well as the number of centroids to generate.
- init : Method for initialization ‘k-means++’ : selects initial cluster centers for k-mean clustering in a smart way to speed up convergence. See section Notes in k_init for more details. ‘random’: choose n_clusters observations (rows) at random from data for the initial centroids.
- Algorithm: K-means algorithm to use.

Agglomerative: All the data is converted into a cluster first. If there are N elements, N clusters are formed. Later, clusters that are close to each other merge to form a new cluster. This continues until the system is stable. Divisive is the opposite of Agglomerative. At first all data is created in a single cluster. Then, clustering is done by breaking this cluster.
Parameters need to be tuned:

- n_clusters: The number of clusters to find. It must be None if distance_threshold is not None.
- Affinity: Metric used to compute the linkage. Can be “euclidean”, “l1”, “l2”, “manhattan”, “cosine”, or “precomputed”. If linkage is “ward”, only “euclidean” is accepted. If “precomputed”, a distance matrix (instead of a similarity matrix) is needed as input for the fit method.
- Linkage: Which linkage criterion to use.
  Single Linkage: Calculates the closest distance between two sets.
  Complete Linkage: Calculates the longest distance between two sets.
  Average Linkage: Calculates the average distance between two sets

DBSCAN: DBSCAN is a density-based algorithm. If a point has more points than the minimum number of points (MinPts) given in the Eps radius, that point is called the center point. These points are points located in the inner parts of the cluster. If a point has fewer points than the minimum number of points (MinPts) given in the Eps radius and that point is adjacent to a center point, the point is called a boundary point.
Parameters need to be tuned:

- Eps: The maximum distance between two samples for one to be considered as in the neighborhood of the other.
- min_samples: The number of samples in a neighborhood for core point.

Choosing Evaluation Technique

The silhouette score gives the distance between sets between -1 and 1. Thus, if the score is close to one, it can be understood that the clusters are far away, if it is close to zero, the boundaries of the clusters are close, and if it is close to minus the clusters are intertwined. Therefore, it gives better results in convex clusters than DBSCAN.

Clustering Algorithms based on Evaluation Technique

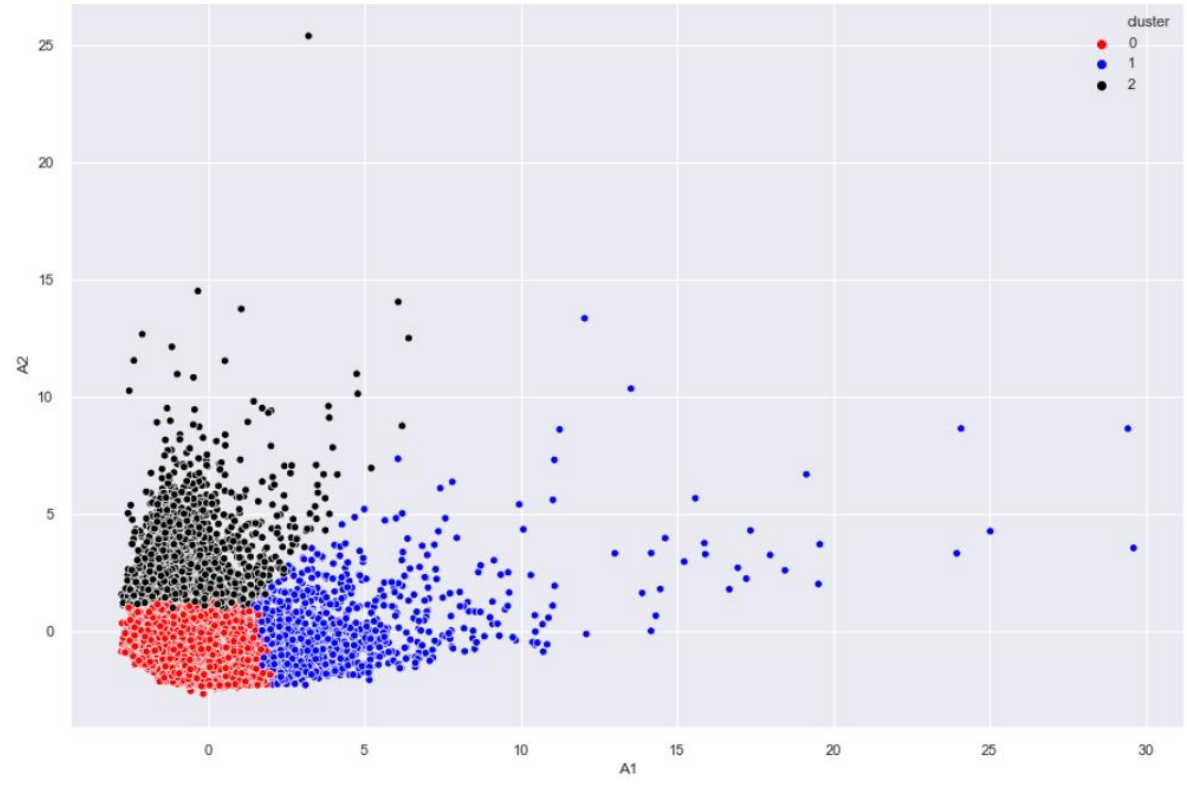
The first two columns have scores before using techniques, and the last two columns have scores after using techniques. With the parameters applied for each clustering method, parameters matching the data set were selected.

	Clustering Method	Silhouette Score	Clustering Method	Silhouette Score
0	KMeans	0.218552	KMeans	0.251787
1	Agglomerative	0.177545	Agglomerative	0.840813
2	DBSCAN	-0.443701	DBSCAN	0.835587

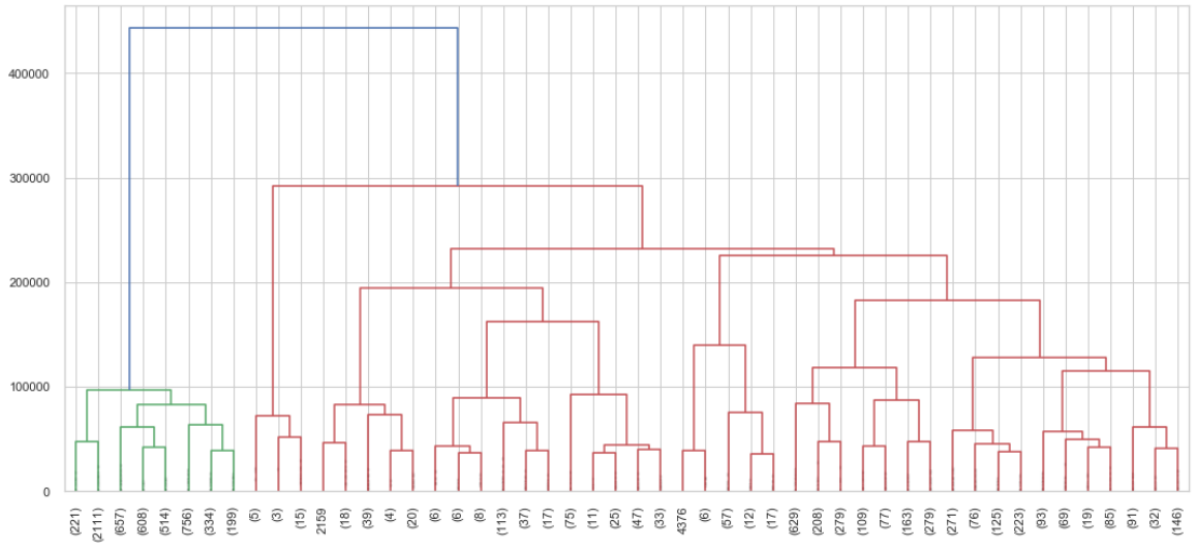
It is seen in the images and score that my cluster is better when the parameters matching the data set are used in Kmeans.
In agglomerative and DBSCAN clustering algorithms, the desired cluster was not achieved before and after the matching parameters were used. The reason for this is that these algorithms determine outlier values.
When DBSCAN images are examined, it can be observed that correct outlier values are found by improving parameters.
In the dentogram visualization used for agglomerative, clustering is observed to occur in a way that separates ourlier value.
In this case, it has been observed that KMeans algorithm clumps better without considering too much outlier values. Therefore, KMeans performed better clustering, although the scores of other clusters were higher.
DBSCAN and Agglomerative clustering according to density and intimacy did not perform well clustering, because the data were very close to each other and gave good scores.

KMEANS: Before parameter tuning:

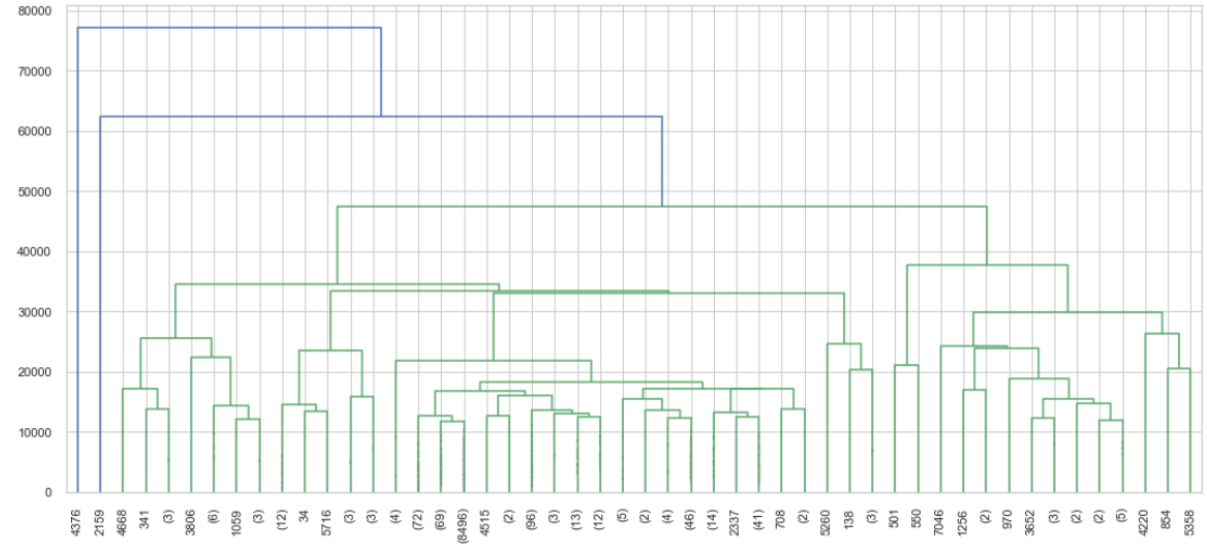
KMEANS: After parameter tuning:



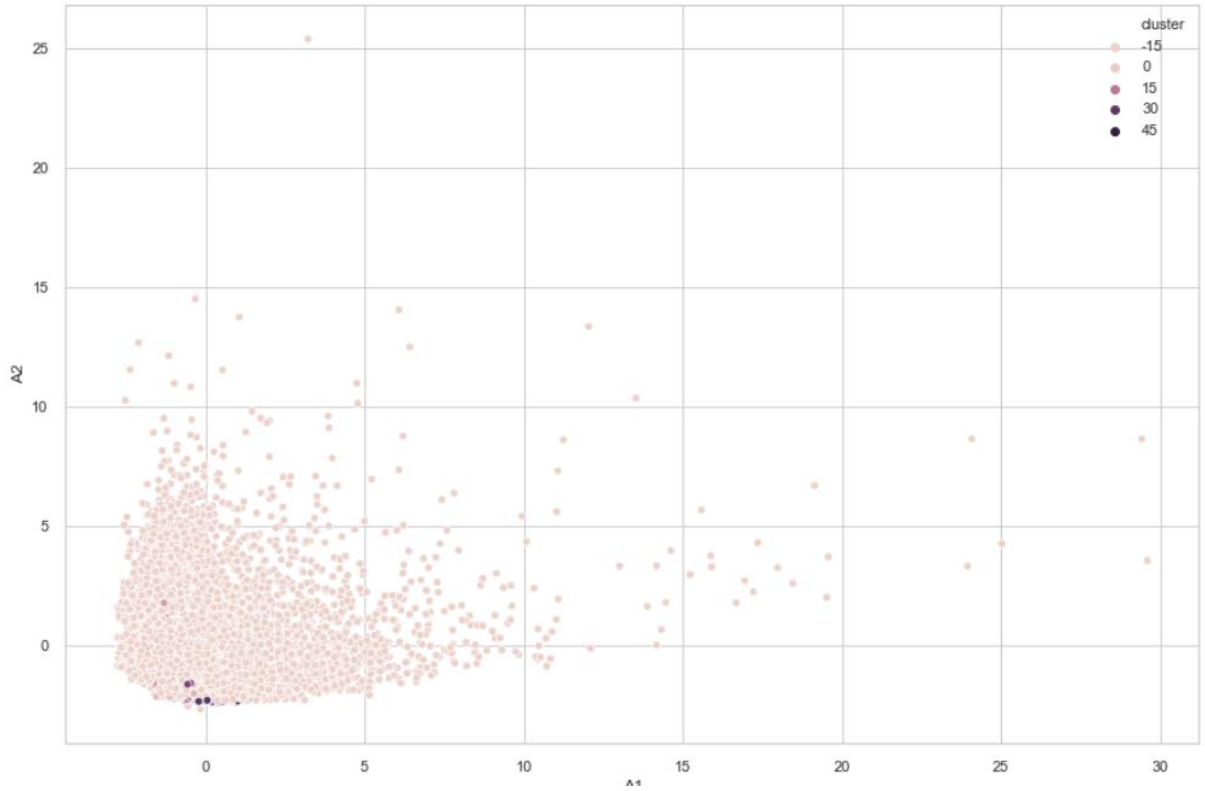
AGGLOMERATIVE: Before Parameter Tuning



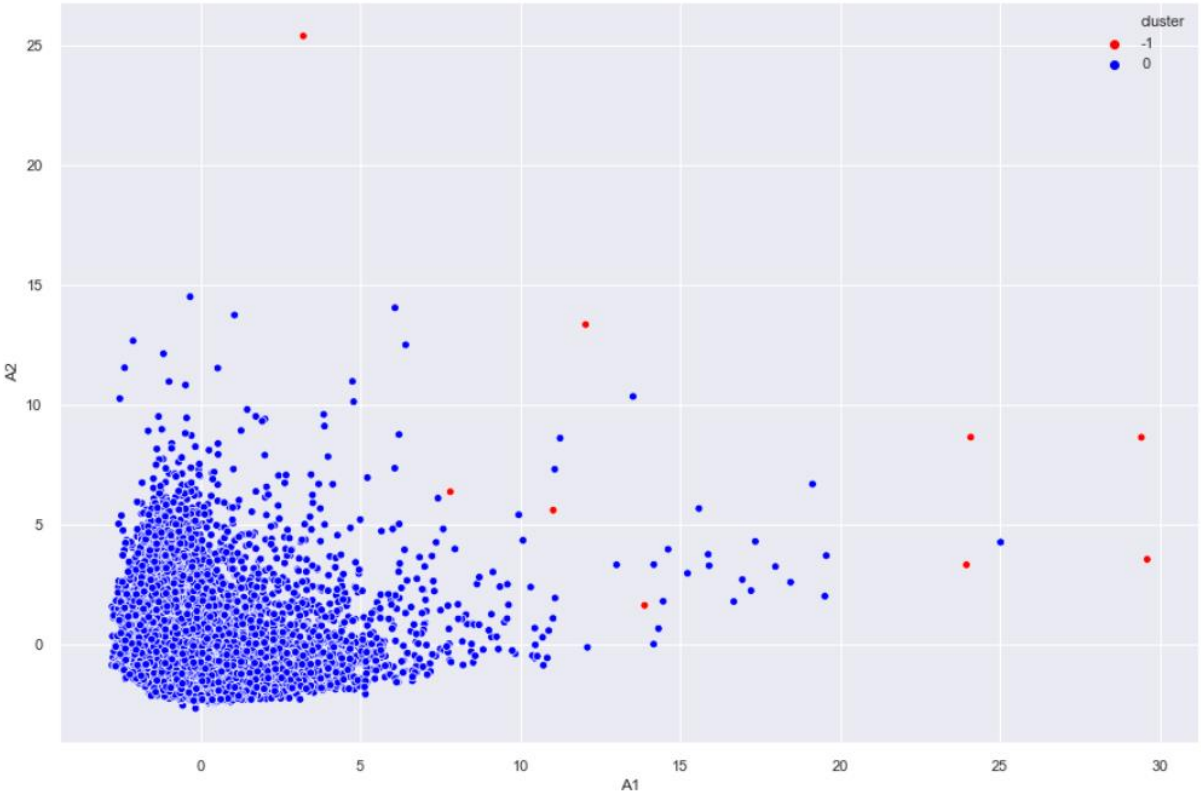
AGGLOMERATIVE: After Parameter Tuning



DBSCAN: Before Parameter Tuning



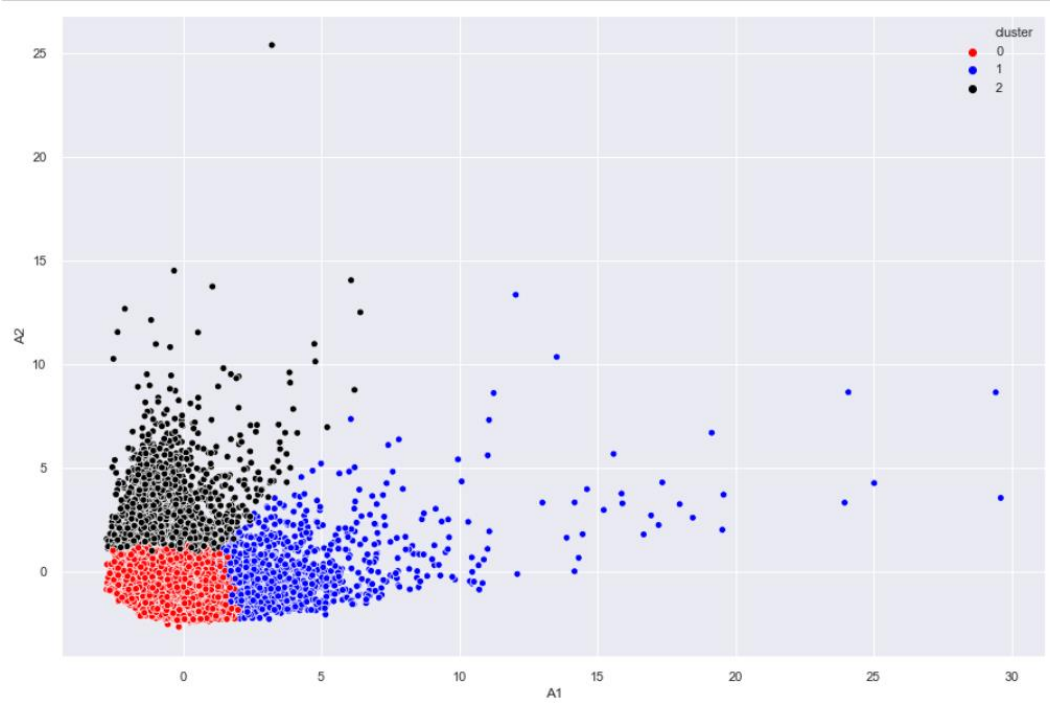
DBSCAN: After Parameter Tuning



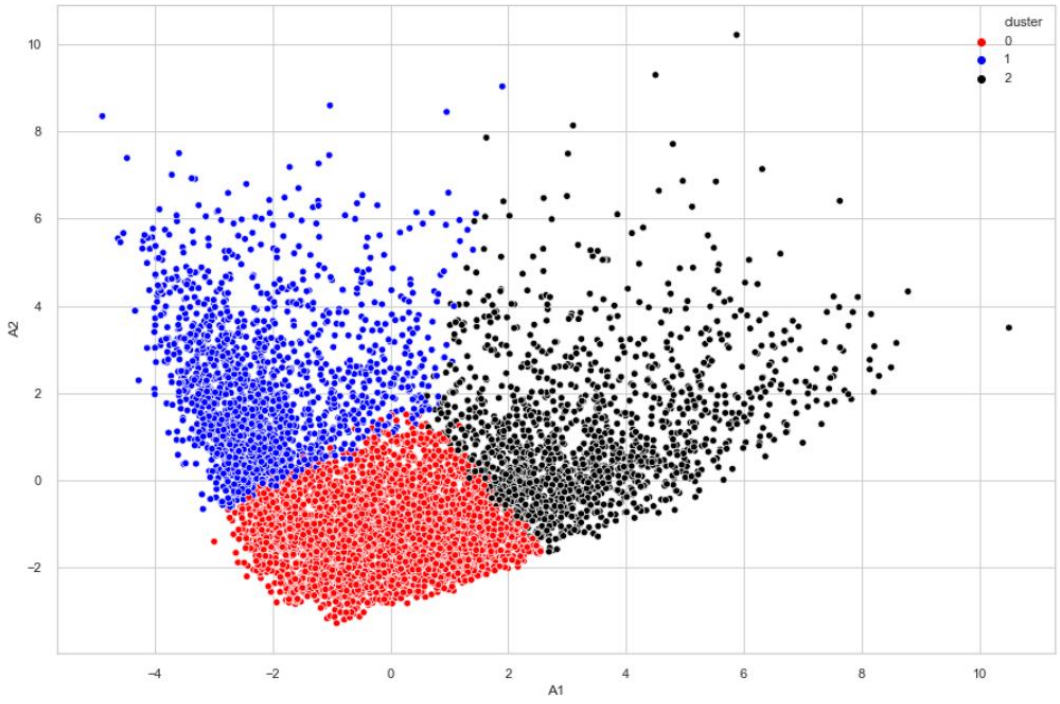
Further Performance Improvement

The reason for the score to decrease when the outlier subtracts the values is that the clusters approach each other because the data is very close.
Kmeans score before removing outliers: 0.251787
Kmeans score after removing outliers: 0.20696304758722564

KMeans before removing outliers:

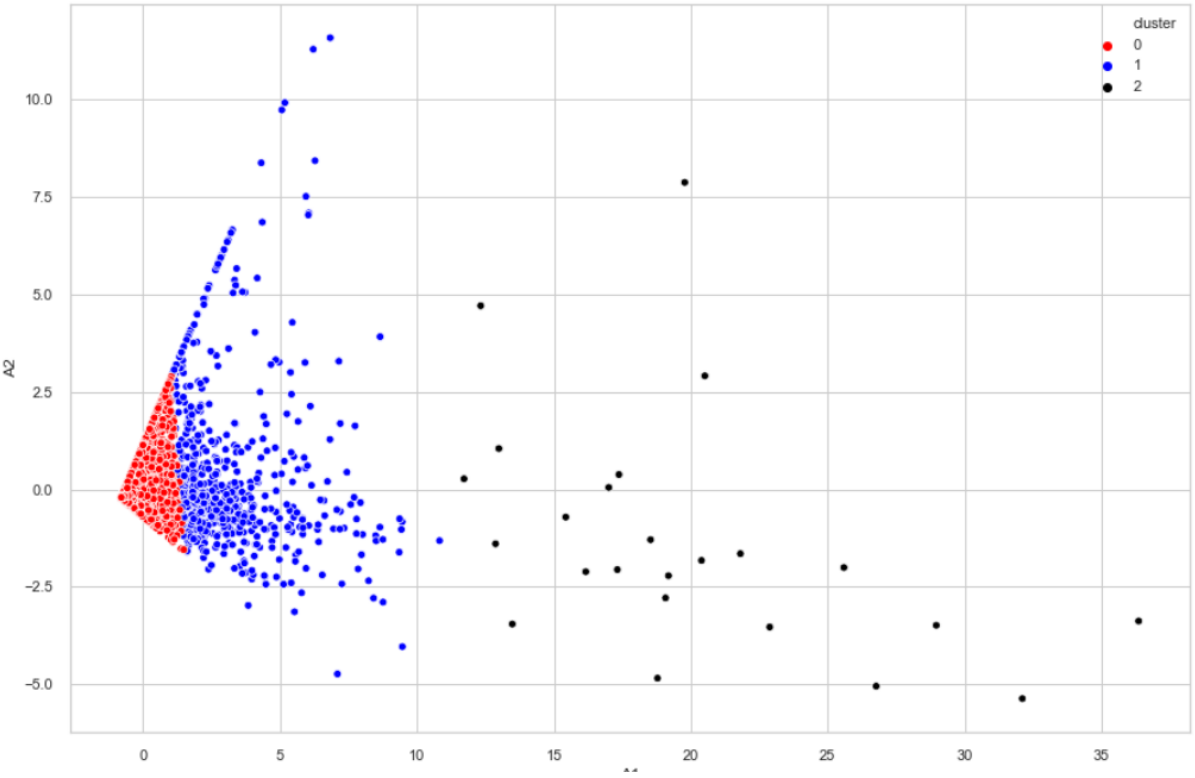


KMeans after removing outliers:



The reason for the better score after the feature selection is due to the fact that the structure of the clusters becomes easier to decide.
Before Feature Selection
0.251787

After Feature Selection
0.7442309443166978



CLUSTERS

Cluster 0: The cluster with the lowest payment amounts and credit card limits. Their minimum spending is high. Payment amounts are low. They may be lower income or new customers.

Cluster 1: Payments and credit card limits are less than the zero cluster, and the second cluster. They avoid spending too much.

Cluster 2: They buy a lot and pay a lot. Credit card limits are high. Their minimum spending is low and the remaining balances are high.