Project 2 Report

Machine Learning for Data Science

Arman Durmus, Javad Kasravi, Gandlur Phani shankar Bharadwaj

January 2022

Exploratory data analysis

Adult dataset has 14 features and one target which is *class_label*. These features are *age*, *work_class*, *fnlwgt*, *education*, *education_num*, *marital_status*, *occupation*, *relationship* *race*, *sex*, *capital_gain*, *capital_loss*, *hours_per_week*, and *native_country*. Our goal is clustering the given data.

Feature selection - Dataset preparation

(a) **Dropping the data**

(i) There are a total of 32561 rows of data , The dataset contains 2399 missing values that are marked with a question mark character (?). Missing values that are marked with a ? character , these rows are deleted from the dataset.

(ii) Categories like " Native country" "work class" "capital loss" "capital gain" parameters have also been dropped , as the bar graph shows one particular Parameter in these categories has been completely dominating , which is not healthy for training the algorithm.

(b) **Feature selection**
Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.
In this study, we will use k-means, k-medoid and DBSCAN algorithms. These algorithms aren't directly applicable to categorical data, for various reasons. The sample space for categorical data is discrete, and doesn't have a natural origin. A Euclidean distance function on such a space isn't really meaningful. Therefore, decision has been made to use numerical data for clustering. Initially, we used *age*,*fnlwgt*, and *hours_per_week*.
using *elbow_value* and *k_mean* algorithm, we found that the optimum k for the mentioned numerical data is 6.

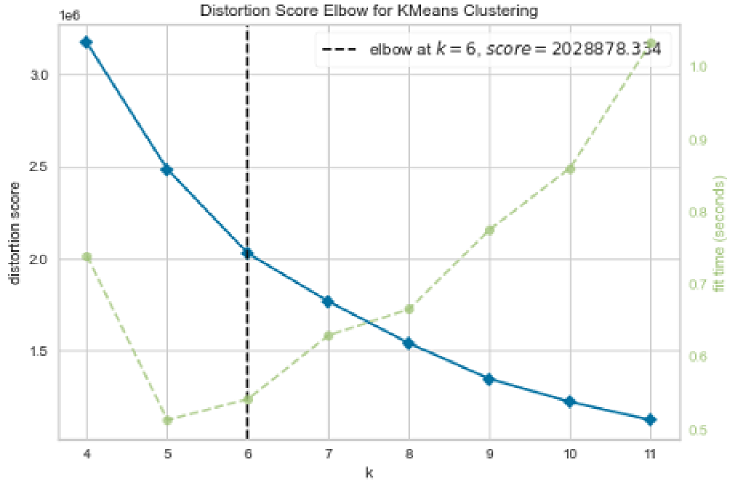

Figure 1: optimum k for tree features

However, the silhouette score (shown below) for the given k is 0.4, resulting in poor clustering.

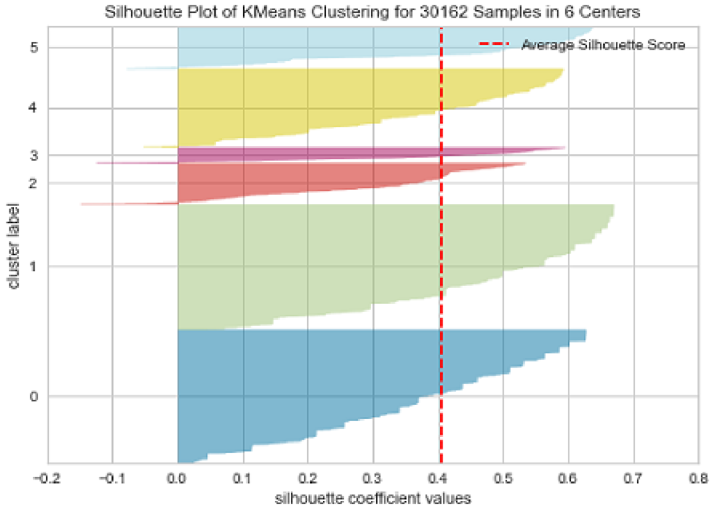

Figure 2: silhouette score for tree features with optimum k (K=6)

By removing *hours_per_week* feature the silhouette score has been improved which will be shown in the next sections.
Additionally, adding further features, such as (for example) education number, which we had assumed to be correlated with the income using our domain knowledge, gave us much worse clustering results with regards to evaluation, like a silhouette coefficient of around 0.25. So we further reduced the dimensionality to get better results.
To summarize, we used the following features for our clustering:

- *fnlwgt*
- *age*

Clustering

(a) **K-means Algorithm**
we applied K-means Algorithm to cluster the adult data set.

(i) Parameter Tuning

To perform efficient clustering, optimum k should be selected. The next graph shows that the optimum k for the given data is 7.

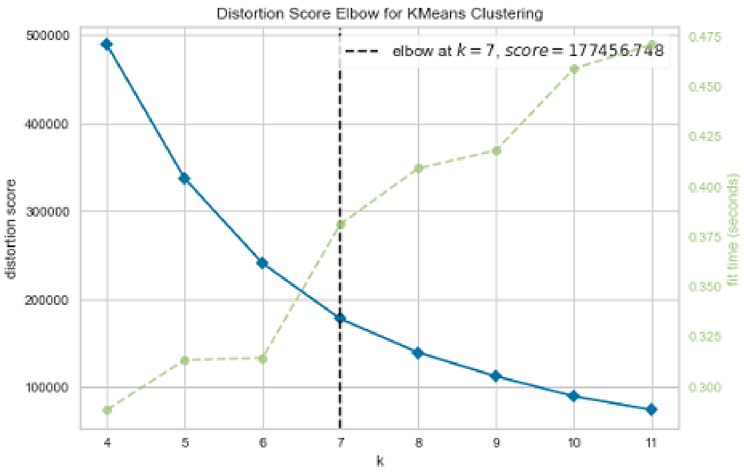

Figure 3: Optimum k for K_means algorithm

(ii) Internal evaluation measures

The Silhouette Coefficient is calculated using the mean intra-cluster distance, and the mean nearest-cluster distance for each sample. Silhouette score of all clusters are more than 0.7 and the average score is 0.54.

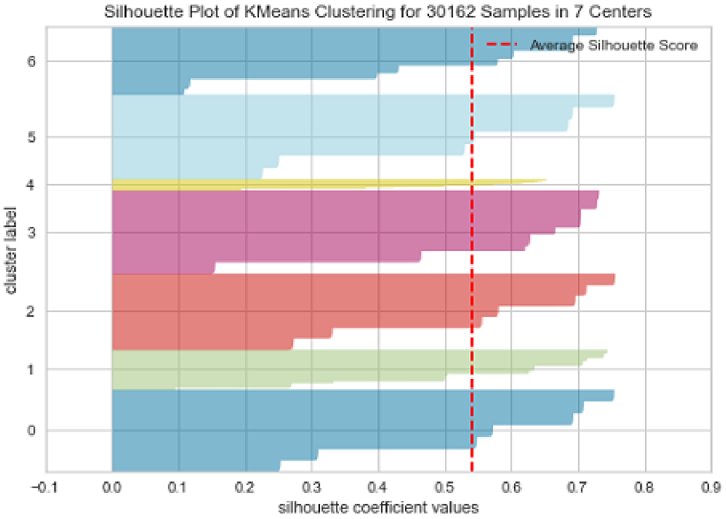

Figure 4: Silhouette score for K_means algorithm with optimum k (K=7)

(iii) External evaluation measures

In this study, we implemented a function that can calculate *Purity* and *Entropy* to evaluate the clustering externally according to the formulas introduced in the lecture. In this data set, *Purity* and *Entropy* are 0.09 and 0.72 respectively.

(iv) Discriminative behaviour

In this model, we assume that *sex* and *race* are protected features. Models supposed to avoid cluster instances based on these features and the number of instances in these clusters should be distributed equally in order to avoid Discriminative behaviour. In the following graph, it shown that this model cannot completely avoid discriminative behaviour with respect to *sex* and *race* features. However, the distribution of these features is somewhat fair with respect to the cluster sizes andgeneral representation of different groups within the dataset. We can see that males and White people are over-represented. Considering this, the distribution of ”female” among the clusters is similar to the distribution of ”male”.

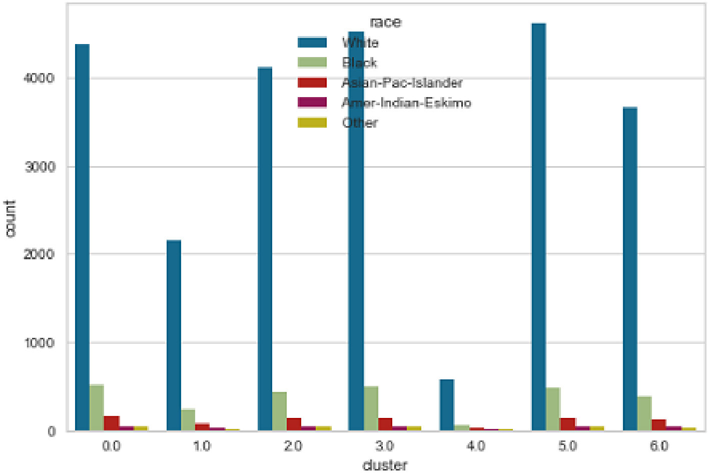

Figure 5: discriminative behaviour of K_means with respect to *sex*

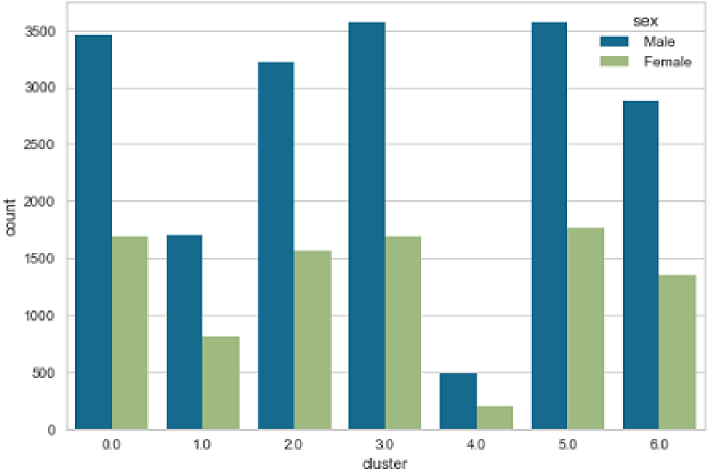

Figure 6: discriminative behaviour of K_means with respect to *race*

(b) K_medoids Algorithm

We applied K_medoids Algorithm to cluster the adult data set.

(i) Parameter Tuning

To perform efficient clustering, optimum k should be selected. The next graph shows that the optimum k for the given data is 5.

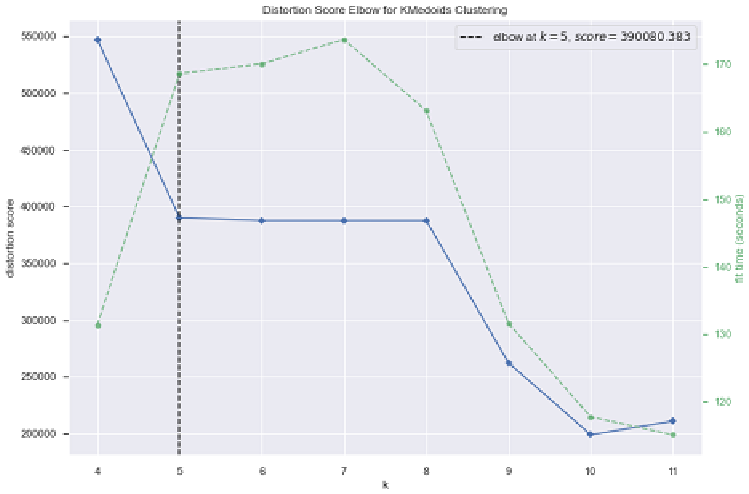

Figure 7: Optimum k for *K_medoids* algorithm

(ii) Internal evaluation measures
The Silhouette Coefficient is calculated using the mean intra-cluster distance, and the mean nearest-cluster distance for each sample. Silhouette score of all clusters are more than 0.7 and the average score is 0.52.

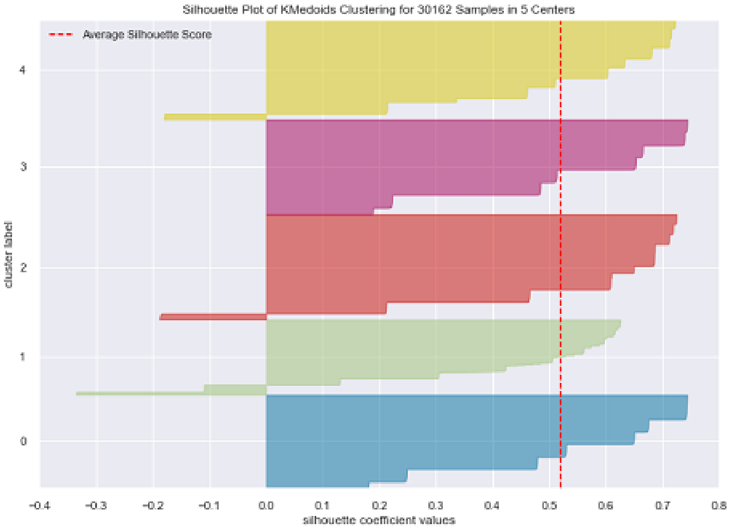

Figure 8: Silhouette score for *K_medoids* algorithm with optimum k (K=5)

(iii) External evaluation measures In this study, we implemented a function that can calculate *Purity* and *Entropy* to evaluate the clustering externally. In this data set, *Purity* and *Entropy* are 0.12 and 0.72 respectively.

(iv) Discriminative behaviour

In this model, we assume that *sex* and *race* are protected features. Models supposed to avoid cluster instances based on these features and the number of instances in these clusters should be distributed equally in order to avoid Discriminative behaviour. In the following graph, it shown that this model cannot avoid discriminative behaviour with respect to *sex* and *race* features.

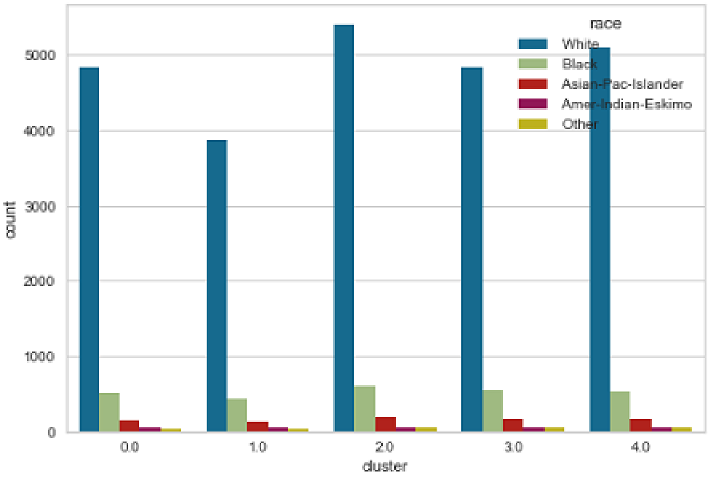

Figure 9: discriminative behaviour of K_mdoids with respect to *sex*

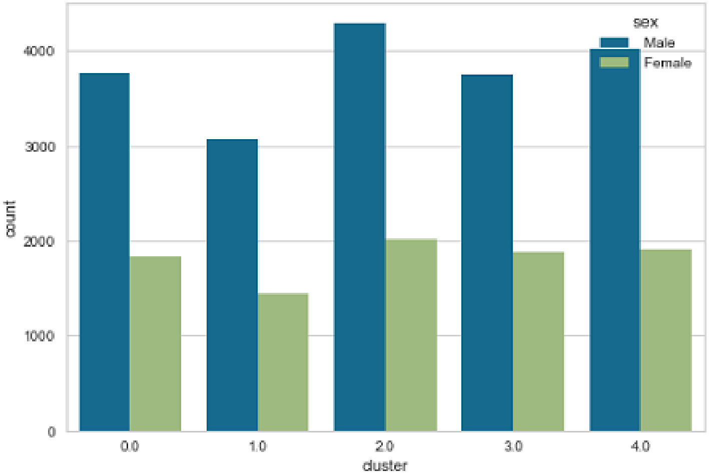

Figure 10: discriminative behaviour of K_mdoids with respect to *race*

(c) DBSCAN

We applied the DBSCAN Algorithm to cluster the adult data set.

This clustering algorithm groups together the data points that are close to each other based on the distance between the points and marks the data points as outliers which are away from high density of data points.

There are two parameters which play an important role in implementation of DBSCAN:

- Epsilon value
- Minimum sample

(i) Parameter Tuning

Determining the epsilon value:-
A distance measure that will be used to locate the points clustered together for a region to be considered dense. Epsilon value is calculated using elbow method. Based on the graph plotted below , the epsilon value is estimated

around 0.5.

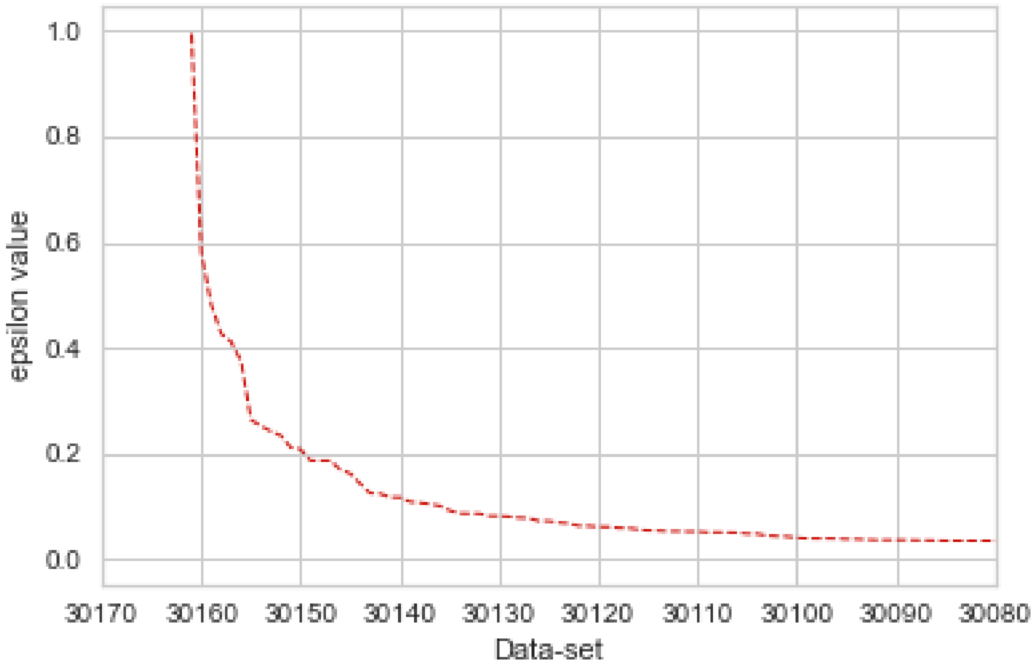

Figure 11: Optimum epsilon for DBSCAN

Minimum sample :-
The number of samples in a neighborhood for a point to be considered as a core point. This includes the point itself.
As there is no automatic way to determine the Min-points for DBSCAN. The Min-points value should be set using domain knowledge and based on the Data set provided. We have selected Minimum samples as 50,100,300.
We have also plotted a graph the shows how silhouette score decreases as the value of minimum sample increases.

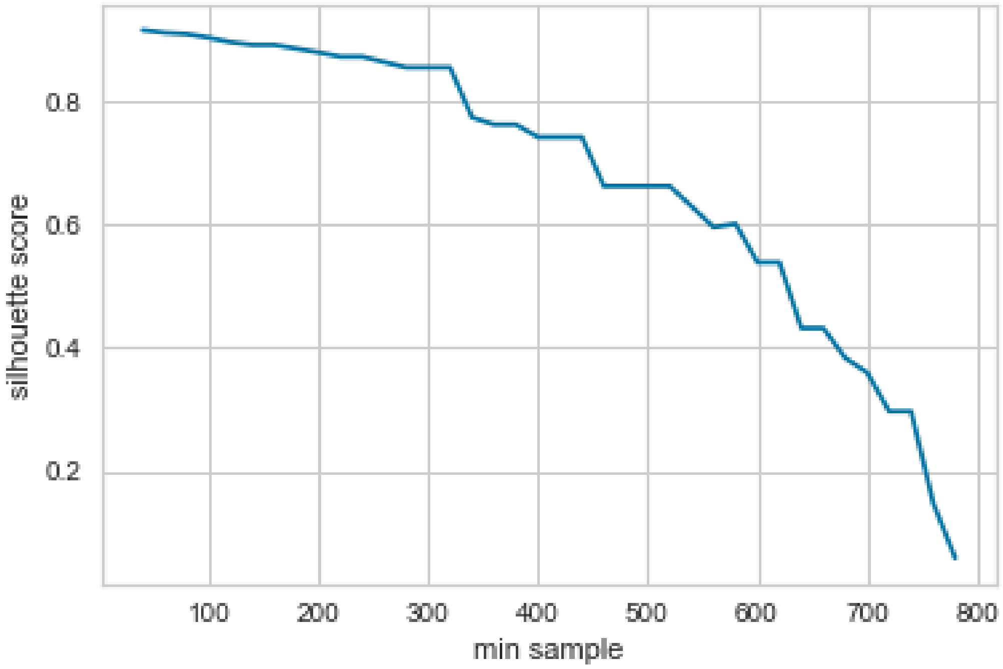

Figure 12: Minimum sample for DBSCAN

(ii) Internal evaluation measures
The Silhouette Coefficient is calculated using the mean intra-cluster distance, and the mean nearest-cluster distance for each sample. For the silhouette coefficients of different parameters, see visualisation part. Generally, we can say that the model manages to obtain a high silhouette score.

(iii) External evaluation measures In this study, we implemented a function that can calculate *Purity* and *Entropy* to evaluate the clustering externally. In this data set, *Purity* and *Entropy* are 0.01 and 0.71 respectively.

(iv) Discriminative behaviour

In this model, we assume that *sex* and *race* are protected features. Models supposed to avoid cluster instances based on these features and the number of instances in these clusters should be distributed equally in order to avoid Discriminative behaviour.

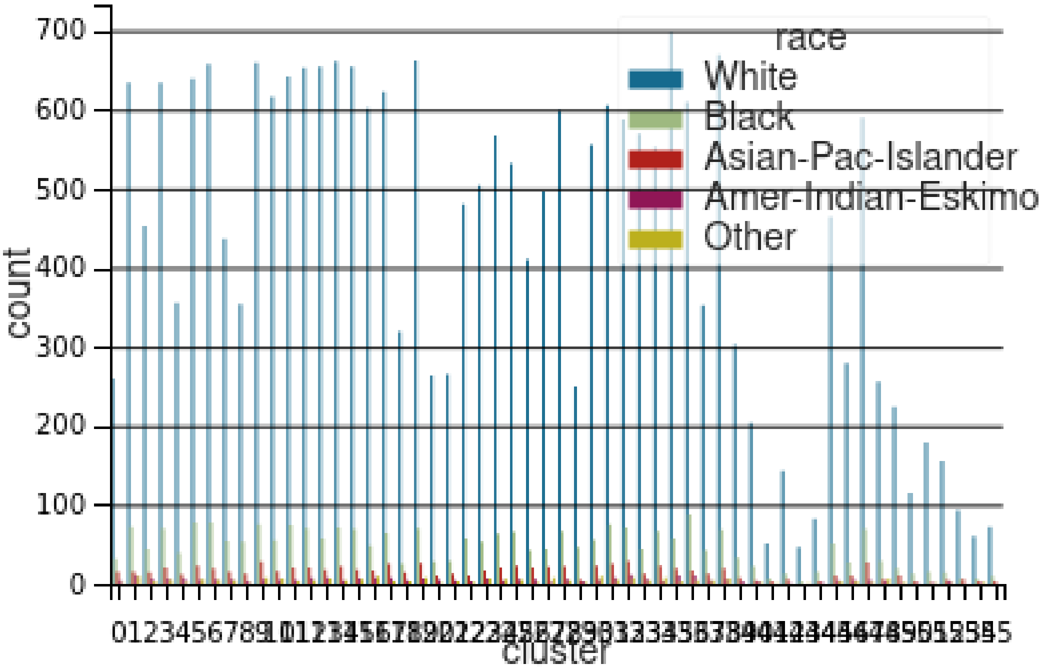

Figure 13: discriminative behaviour of K_mdoids with respect to *sex*

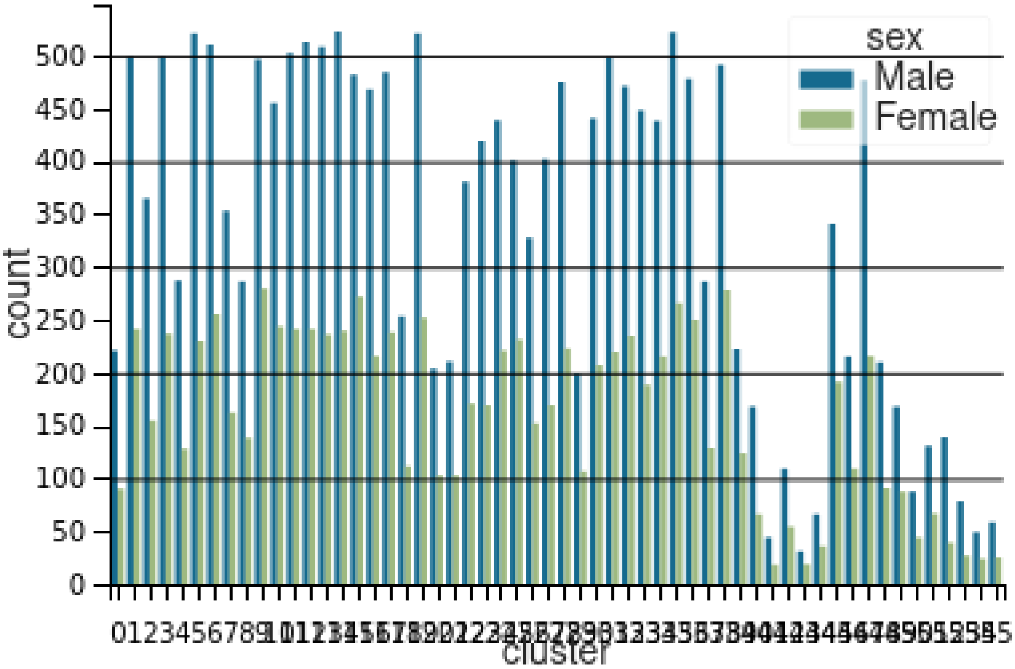

Figure 14: discriminative behaviour of K_mdoids with respect to *race*

Visualization

In order to visualize our K-Means and K-Metoids models, we use an technique, called dimension reduction, to have a better understanding of decision boundaries in different classifier. Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can

be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable (https://en.wikipedia.org/wiki/Dimensionality_reduction).
Dimension reduction is computationally expensive, so we used just 5000 instances of test dataset. The following graphs are the ability of different classifiers to with respect to two dimensions.

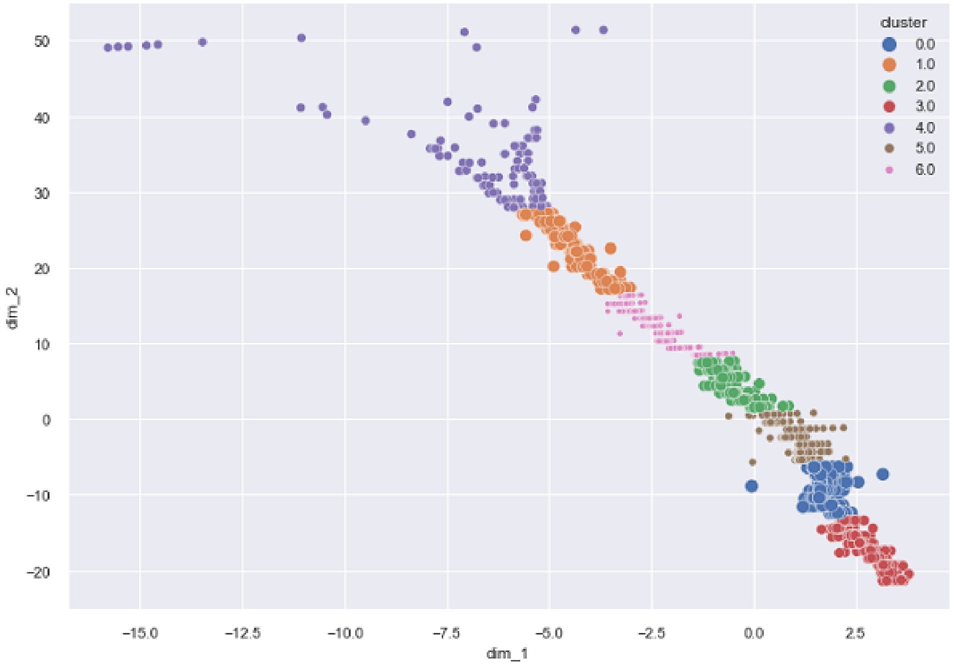

Figure 15: K_means Clustering

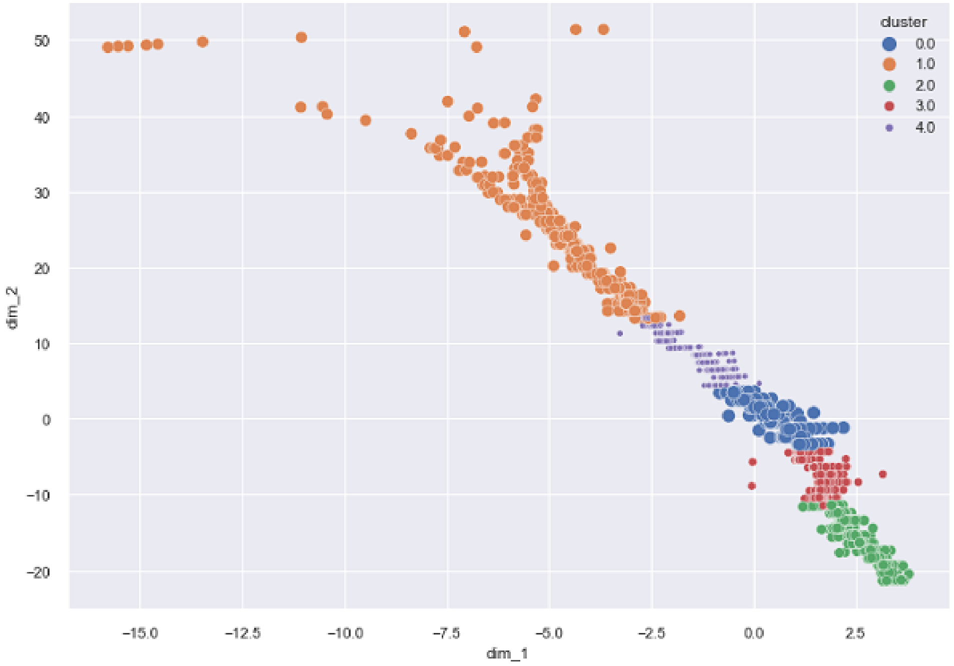

Figure 16: K_medoids Clustering

DBSCAN modelling and visualization :-
We have used DBSCAN module from sk learn by providing with the values of Epsilon and Min-points and we have used distance metric as Euclidean distance.
By doing so , DBSCAN classifies all the noise points as [-1] and other cluster numbers as [1,2,3..etc].

We have used (ggplot) for the visualization of clusters returned by DBSCAN. We have implemented DBSCAN algorithm with multiple min-samples . we have shown in the results section how results differ with different values of min-samples .

1) For min-sample = 50:

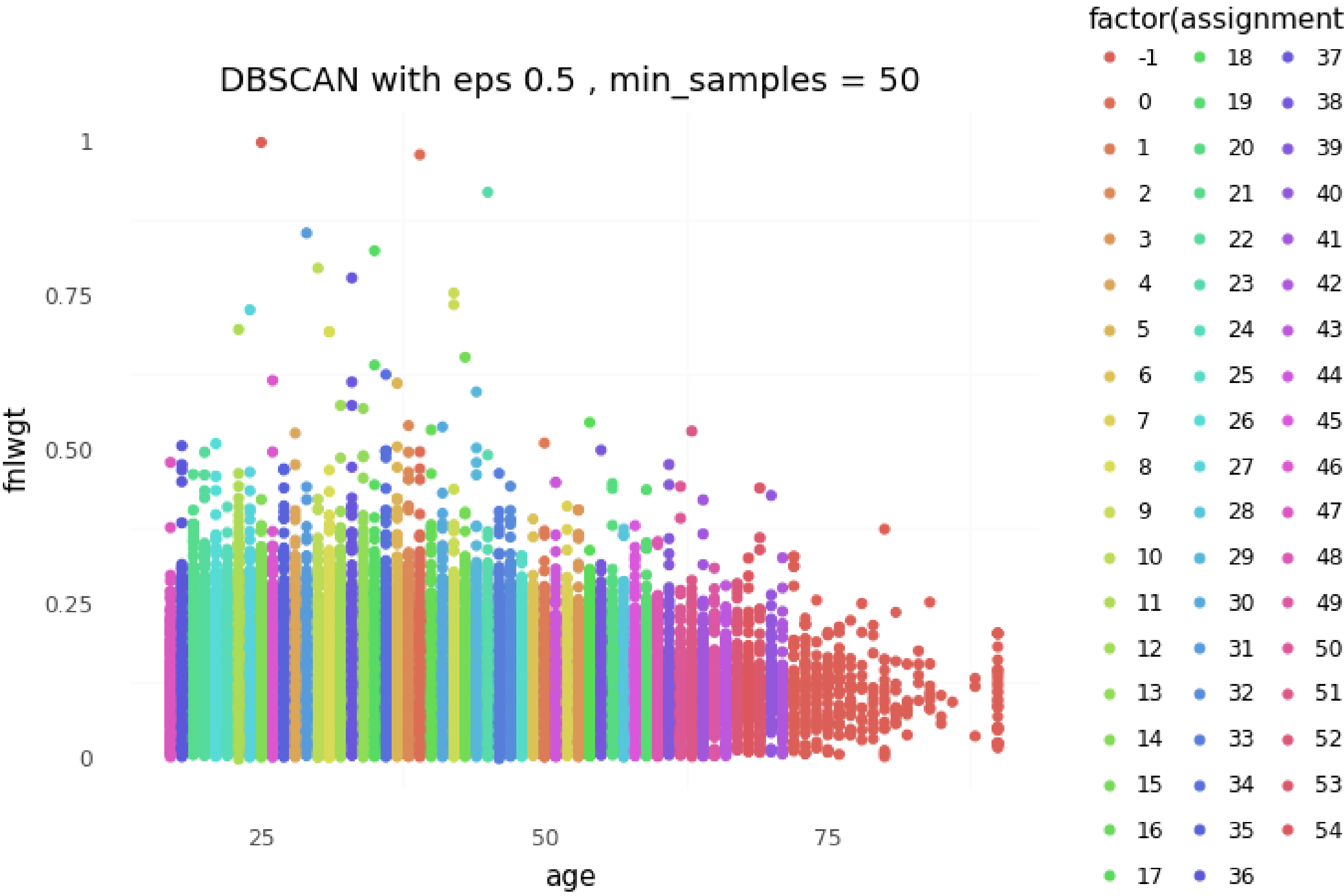

Figure 17: DBSCAN Clustering w. min-sample = 50

Estimated number of clusters: 55
Estimated number of noise points: 331
Silhouette Coefficient: 0.972

1) For min-sample = 100:

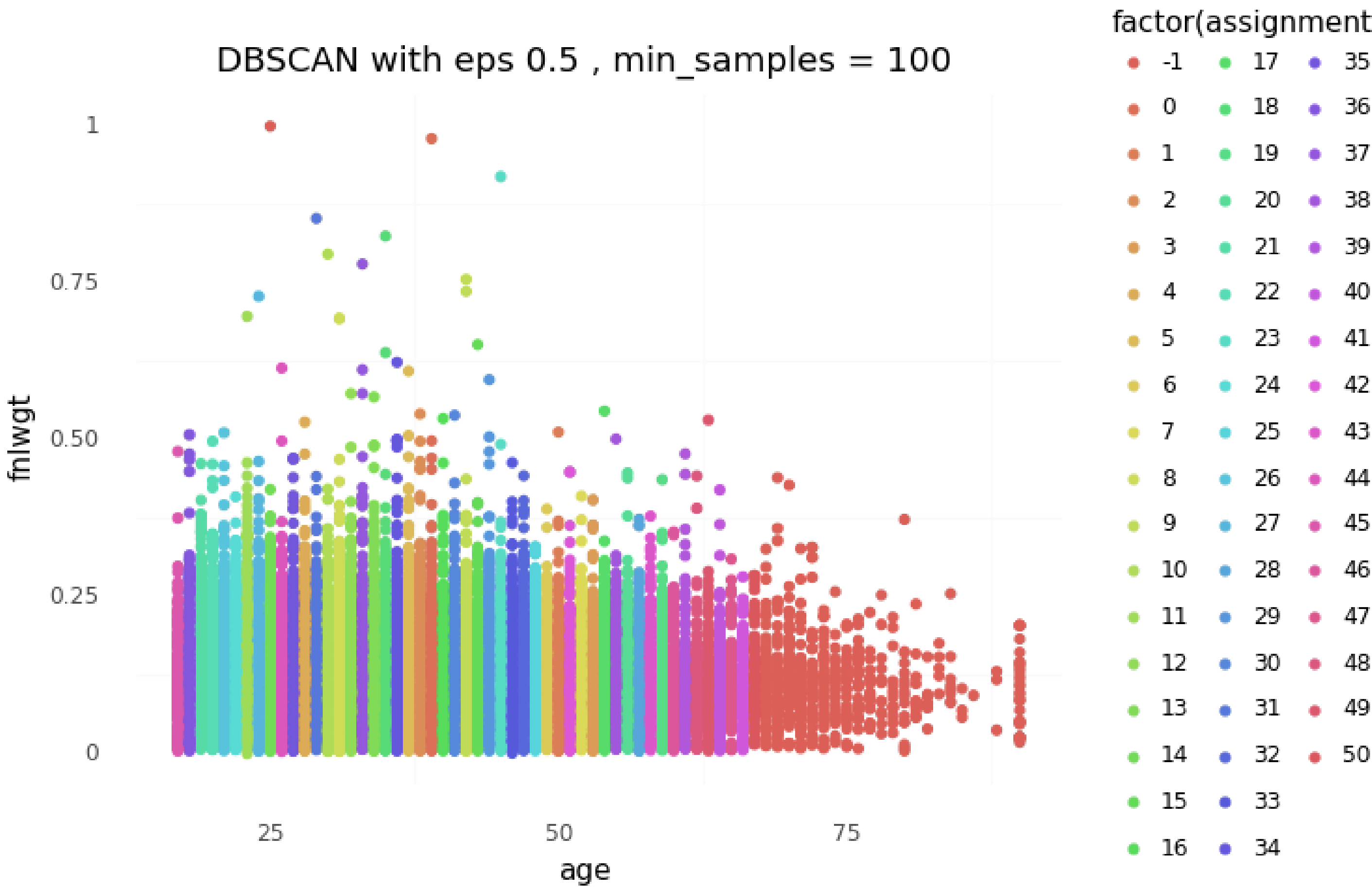

Figure 18: DBSCAN Clustering w. min-sample = 100

Estimated number of clusters: 51
Estimated number of noise points: 619
Silhouette Coefficient: 0.969

1) For min-sample = 300:

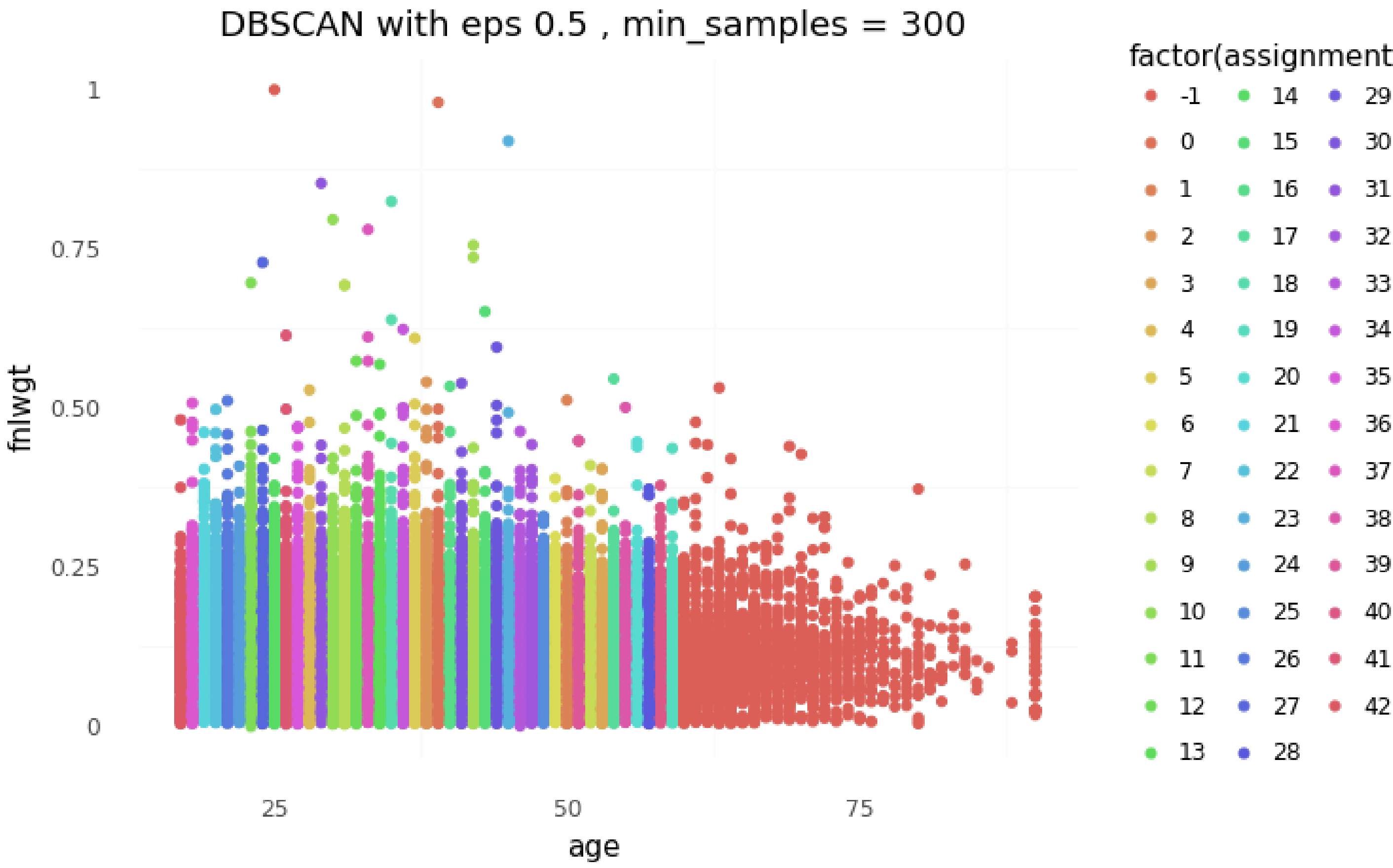

Figure 19: DBSCAN Clustering w. min-sample = 300

Estimated number of clusters: 43
Estimated number of noise points: 2083
Silhouette Coefficient: 0.942

Result and Discussion

In this study, tree different clustering models have been compared with in internal and external measures:

- K_means
- K_medoids
- DBSCAN

According to the optimum k of *K_means* and *K_medoids*, the overall silhouette score for these algorithms obtained 0.54 and 0.52 respectively. However, the number of optimum clusters for the mentioned algorithms is not the same. *K_medoids* is more expensive than *K_means* with regard to running time, but both methods cannot completely avoid discriminative behavior with respect to protected features (*sex* and *race*). By comparing these two algorithms, we can say that *K_means* can outperform *K_medoids* with respect to precision and running time.

DBSCAN

We get a Silhouette value of around 0.95 for the model, which is good. Conversely, we get a high number of clusters, no matter what parameters are chosen, although parameters can be tuned to somewhat decreas the no. of clusters, it also harms the silhouette score. For higher values of min-samples, the algorithm considers many data points, which do not seem like outliers, as outliers, thus it might be better to pick a somewhat lower min-samples.