INTERSECTIONALITY & FAIRNESS IN ALGORITHMIC CLUSTERING

Jeremiah Mensah, Muno Siyakurima, Brie Sloves, Avery Hall, Victor Huang, Sophie Boileau, Armira Nance

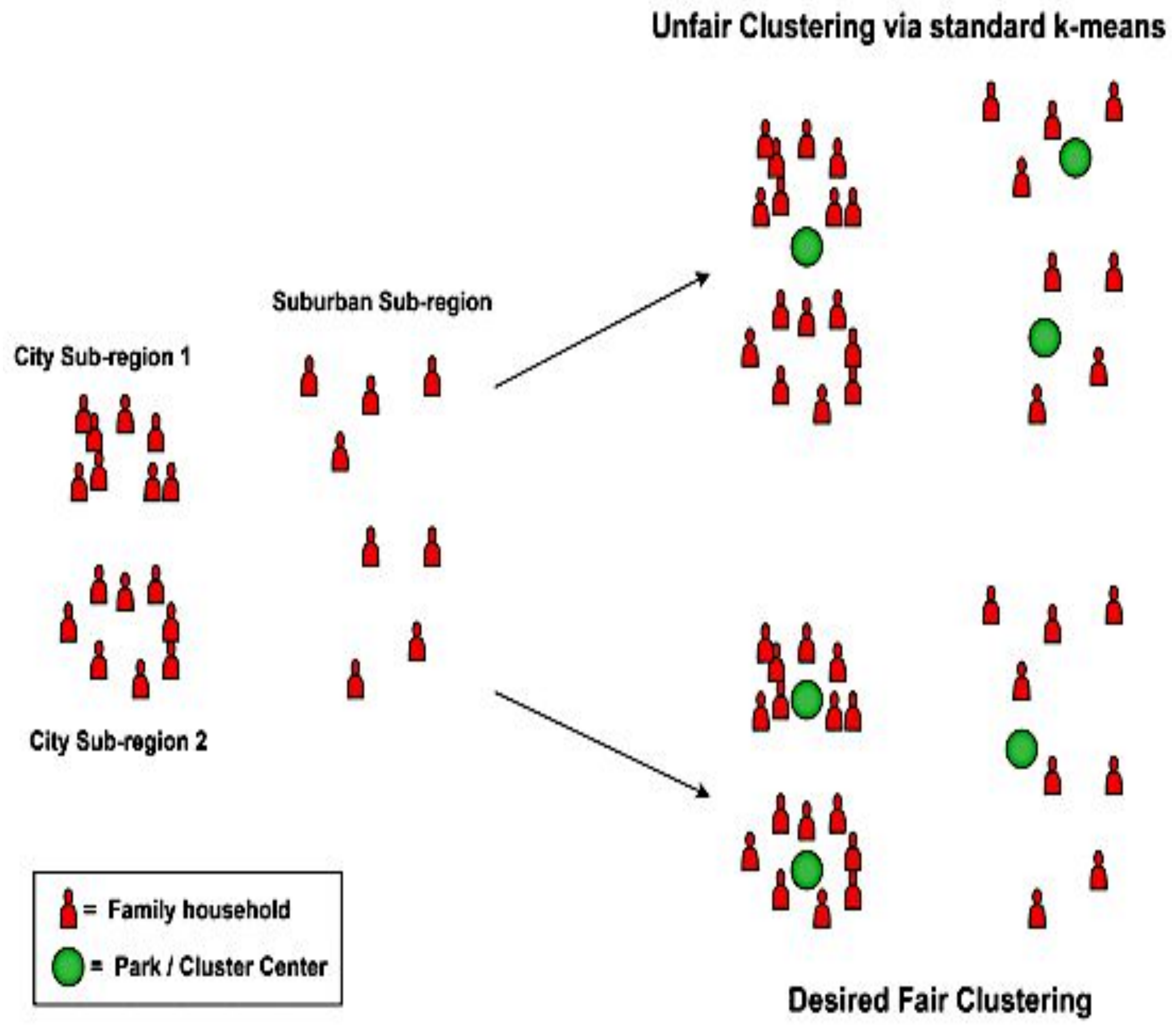
Advised by: Professor Layla Oesper

BACKGROUND

Clustering is a fundamental concept in machine learning and data analysis. It partitions *n* data points into *k* clusters, with the **primary aim** of uncovering inherent patterns or structures within a dataset. For our group's project, we want to explore the roles of **demographic bias** and **intersectionality** in k-means clustering and clustering through fairlets and see which trends and patterns emerge from our dataset *High School Longitudinal Study of 2009*.

FAIRNESS DEFINITION

In this project, we adopt the fairness definition from the paper 'Fair Clustering Through Fairlets'. **Fair clusters** are those that maintain the same attribute ratio as the original dataset. For instance if gender is the attribute, a fair cluster would retain the same male to non-male ratio found in the original dataset.



OBJECTIVE & METHODS

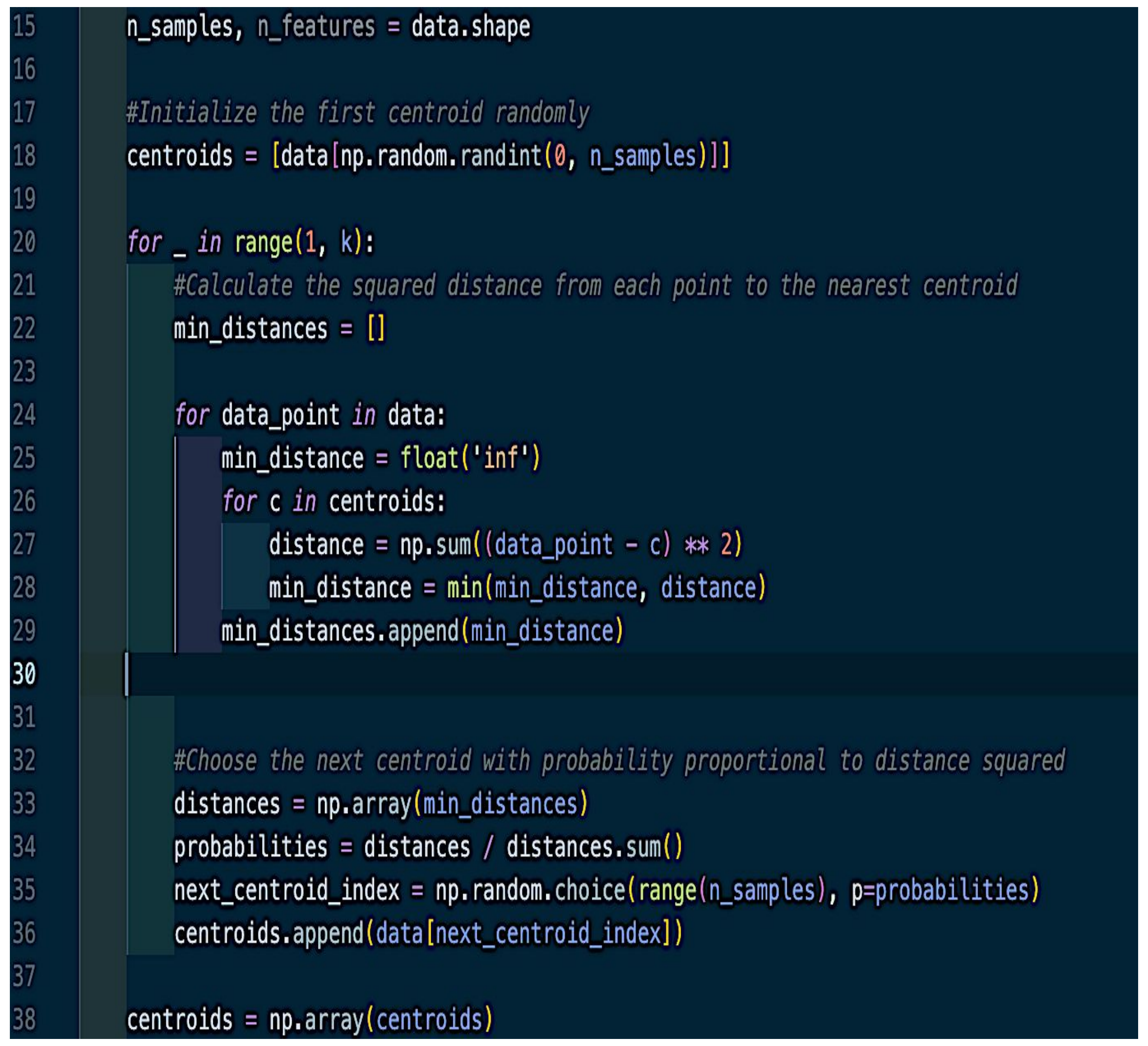
We'll analyze four key attributes that have high levels of intersectionality: **Parent Education, Socioeconomic Status, Household Members per Income, and Hours of Extracurricular Activity** using clustering methods, including **K-Means, K-Means++, MCF Fairlets, and Vanilla Fairlets**. We'll examine how **Race and Gender** affect the balance value and k-center cost.

ALGORITHM: K-MEANS++

What is it?

K-means++ represents an enhanced version of the basic k-means clustering algorithm

1. Start by selecting a random data point from the dataset as the first centroid.
2. Compute the distance between each data point and the nearest existing centroid.
3. Square the calculated distances from step 2
4. Select the next centroid based on the probabilities calculated in step 3. Data points with higher squared distances (farther from existing centroids) have a higher probability of being chosen as the next centroid
5. Repeat steps 2 to 4 iteratively until 'K' centroids have been chosen.



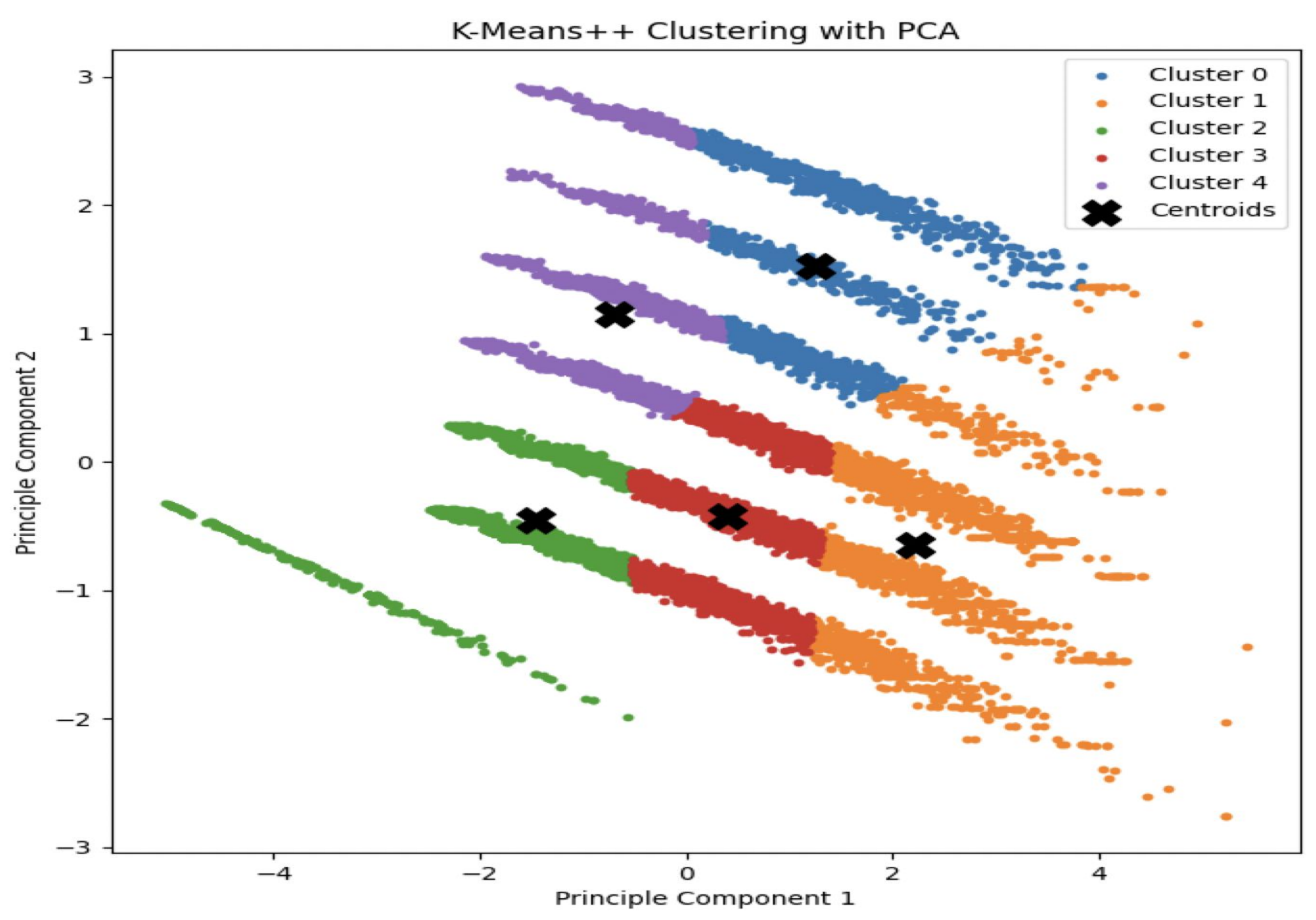
```
n_samples, n_features = data.shape

#Initialize the first centroid randomly
centroids = [data[np.random.randint(0, n_samples)]]

for _ in range(1, k):
    #Calculate the squared distance from each point to the nearest centroid
    min_distances = []

    for data_point in data:
        min_distance = float('inf')
        for c in centroids:
            distance = np.sum((data_point - c) ** 2)
            min_distance = min(min_distance, distance)
        min_distances.append(min_distance)


    #Choose the next centroid with probability proportional to distance squared
    distances = np.array(min_distances)
    probabilities = distances / distances.sum()
    next_centroid_index = np.random.choice(range(n_samples), p=probabilities)
    centroids.append(data[next_centroid_index])

centroids = np.array(centroids)
```

Why?

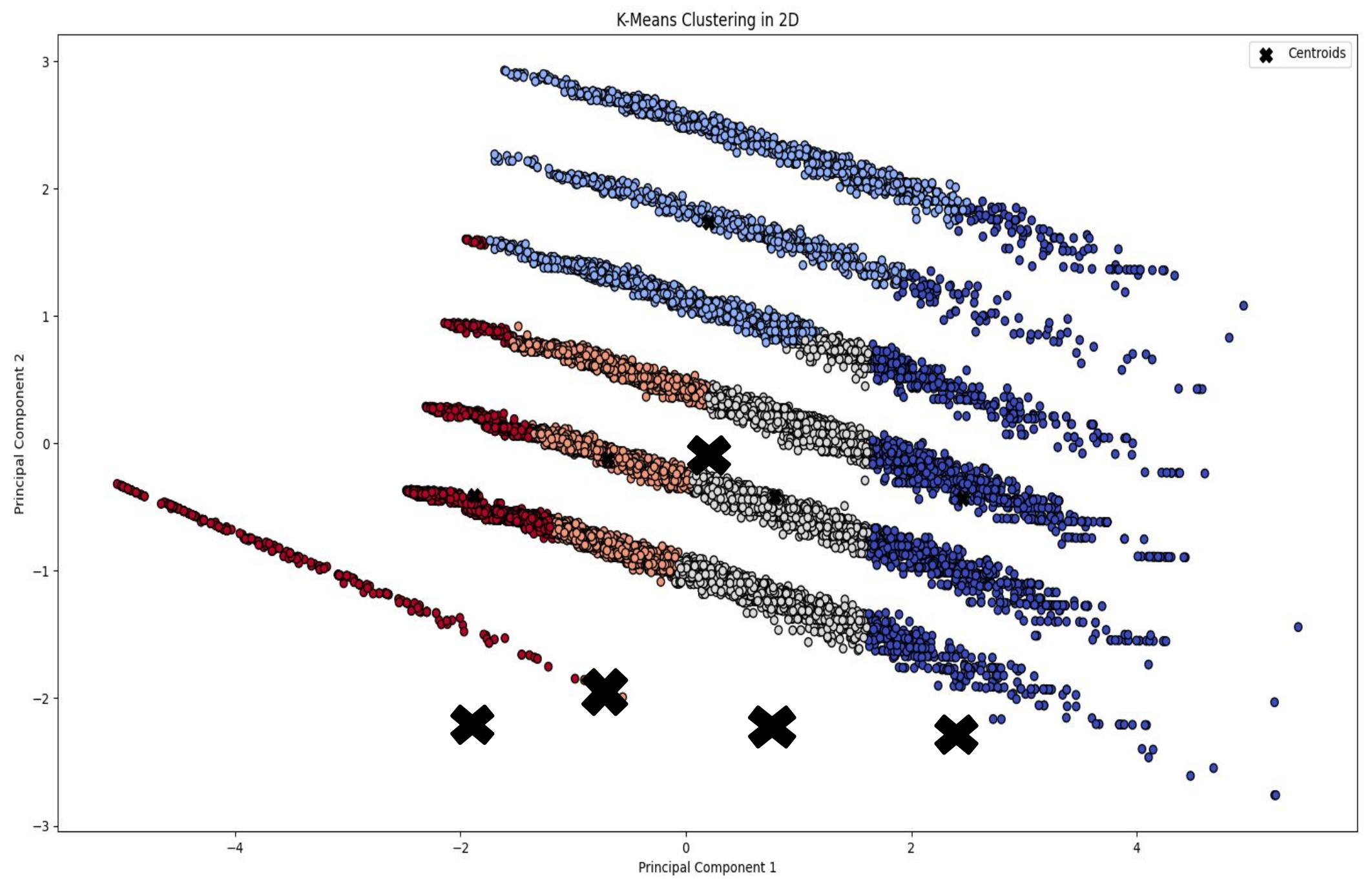
- One significant limitation of the basic k-means algorithm lies in its method of centroid initialization. It resorts to randomly selecting K data points as initial centroids and then proceeds with the clustering process. This initialization approach has several drawbacks such as: instability, higher computational costs, and suboptimal clustering.
- K-means++, on the other hand, addresses these issues by initializing its centroids in a more systematic and efficient manner. This initialization process offers several advantages such as: stable clustering, even distribution, and higher quality clustering.

RESULTS

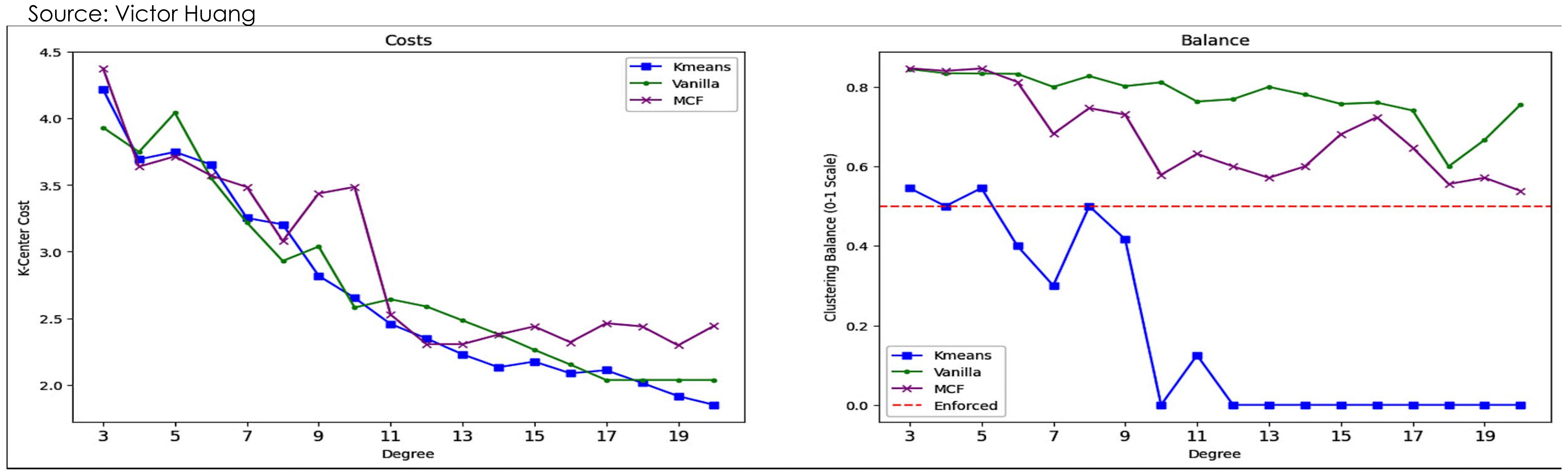
K-means++ clustering.



Basic K-means clustering. Source : Muno Siyakurima



Comparing regression of k-centers costs and balance over different degrees of K with vanilla fairlets, minimum cost flow (MCF) fairlets, and basic k-means. On the right graph, 'enforce' is a measure of fairness.
Source: Victor Huang



CONCLUSIONS

K-means vs K-means++

In our cluster analysis, we applied both K-Means++ and Basic K-Means to group data based on the attributes outlined in the Objective & Methods section. Both K-means and K-means++ resulted in similar initial centroid placements, but the final clustering outcomes differed. Notably, K-means++ yielded centroids that were more widely distributed across the data space, aligning with the advantageous properties associated with K-means++. However when running each algorithm multiple times (not pictured here) they both produced inconsistent clusterings which could be due to the random initialization step present in both algorithms.

K-center cost & Fairness Comparison

We further investigated the impact of fairness in clustering using K-Means, Vanilla Fairlets, and MCF Fairlets. On the left side of the line graphs, MCF Fairlets, K-Means, and Vanilla Fairlets exhibited similar k-center costs until degree 8 - 10 which indicates MCF and Vanilla Fairlets don't have much impact on k-center costs. On the right side of the graphs, we introduced Gender as a balancing attribute. Notably, MCF Fairlets and Vanilla Fairlets consistently outperformed K-Means in terms of enforcing fairness. These results lead us to conclude that, based on our dataset and our defined fairness criteria, MCF and Vanilla Fairlets are more effective in clustering data fairly with respect to Gender.

ACKNOWLEDGEMENTS & REFERENCES

Special thanks to Layla Oesper for guiding us throughout this project and my group mates for their excellent work throughout this project..

- https://www.geeksforgeeks.org/ml-k-means-algorithm/
- https://dl.acm.org/doi/pdf/10.5555/3295222.3295256
- "High School Longitudinal Study of 2009 (HSLS:09) - Overview." National Center for Education Statistics, nces.ed.gov/surveys/hsls09/. Accessed 5 Nov. 2023.
- https://cacm.acm.org/magazines/2020/5/244336-a-snapshot-of-the-frontiers-of-fairness-in-machine-learning/fulltext?mobile=false
- https://ieeexplore.ieee.org/document/9541160