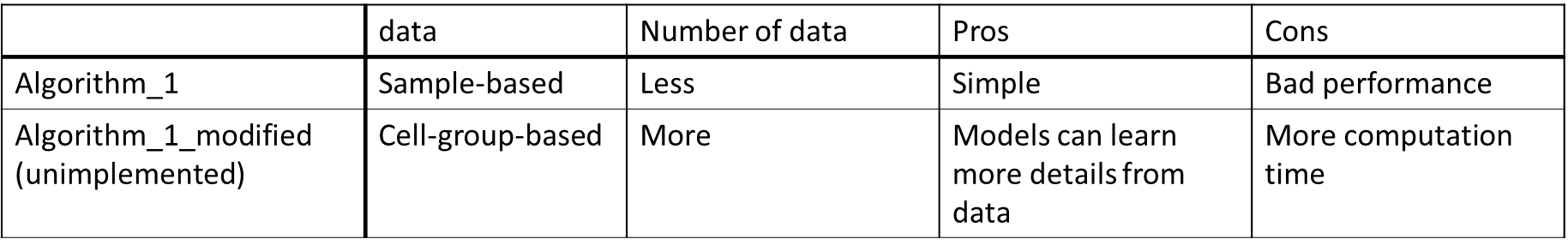
1. Algorithm\_1
2. Aggregate cells in each sample into one data point by calculating the max, min, quartiles, mean, and standard deviation for each feature. This results in 40 data points (samples).
3. Separate the data into a train set and a test set for model evaluation.
4. Consider whether Principal Component Analysis (PCA) is needed. If so, determine the number of components for PCA.
5. Implement machine learning models, fit the data and fine-tune the models.
6. Evaluate the performance of the models using metrics such as accuracy, precision, recall, F1-score, and AUROC (Area Under the Receiver Operating Characteristic curve).
7. Use feature engineering techniques such as applying domain knowledge or feature selection using feature importance estimated by tree-based models.
8. Verify the dataset and remove any bad data.
9. Repeat steps 1-6 to improve the performance of the models.
10. Perform online validation to evaluate the model's performance on new unseen data.

Algorithm\_1\_modified (unimplemented)

1. The first step in this machine learning process is to use the K-means method to separate the cells of each sample into groups. This is done to identify patterns and clusters within the data that can be used for further analysis and modeling.
2. Once the cells have been grouped, the data from each group is aggregated into data points by calculating various statistics such as the max, min, quartiles, mean, and standard deviation for each feature. This results in a set of (40\*# of groups) data points that can be used for further analysis and modeling.
3. Next, the data is separated into a train set and a test set for model evaluation. The train set is used to train and fine-tune the models, while the test set is used to evaluate the performance of the models.
4. In this step, the need for Principal Component Analysis (PCA) is considered. If it is determined that PCA is necessary, the number of components to be used is determined. PCA is used to reduce the dimensionality of the data and make it easier to work with.
5. In this step, machine learning models are implemented and fitted to the data. The models are fine-tuned to optimize their performance. The inference result of the test data is determined by voting on the result of each group.
6. The performance of the models is evaluated using metrics such as accuracy, precision, recall, F1-score, and AUROC (Area Under the Receiver Operating Characteristic curve). These metrics provide a way to measure the performance of the models and compare them to one another.
7. Feature engineering techniques are used to improve the performance of the models. This can include applying domain knowledge or feature selection using feature importance estimated by tree-based models.
8. The dataset is verified and any bad data is removed to ensure that the models are working with accurate and reliable data.
9. Steps 1-6 are repeated in order to improve the performance of the models. This may involve experimenting with different models, fine-tuning parameters, or changing the dataset.
10. Finally, online validation is performed to evaluate the model's performance on new, unseen data. This helps to ensure that the models are robust and can generalize well to new data.
11. To identify the local density of cells in an image, we will use two-dimensional kernel density (2D-kde) estimates. This method calculates the density of cells in a specific area by analyzing the distribution of cells within a specified kernel.
12. After obtaining the local density estimates, we will set a threshold to keep only cells that are in regions of high density. These cells will then be separated into a few different segmentations.
13. The marked cells in the image appear to be located near the point (0.6, 0.0). To further analyze this specific segmentation, we will use one of the following methods:
    1. Find the centers of each segmentation, and calculate the distance between the center and (0.6,0.0)
    2. Use a region growing method and set (0.6, 0.0) as the seed point. This will allow us to identify the specific segmentation that is located near (0.6, 0.0).