به نام خدا

عنوان: پیاده‌سازی الگوریتم SOStream

درس: داده کاوی

استاد درس: مهندس قاسمی

مهدی دهقانی

۲۲۰۷۹۷۰۴۸

mehdi.dehghani@ut.ac.ir

mahdiazadi18@yahoo.com

کلاس MicroCluster

این کلاس نشان دهنده خوشه‌های برنامه است که ویژگی هایی مثل تعداد داده‌ها، شعاع، مرکز، زمان ساخت، آخرین زمان تغییر و لیست داده‌های درون آن دارد.

این کلاس از هفت متد تشکیل شده است:

۱. update_last_edited_timel
این متد آخرین زمان تغییر خوشه را به روز رسانی می‌کند.

۲. insert
این متد داده جدید را به خوشه اضافه می‌کند.

۳. merge_data_points
این متد برای ادغام دو لست داده به کار می‌رود.

۴. fading
این متد برای محاسبه f(t) که مقدار از بین رفتن خوشه است به کار می‌رود.

۵. get_radius
متد گرفتن مقدار شعاع خوشه

۶. set_radius
متد برای مقداردهی یا تغییر شعاع خوشه

۷. get_centroid
متد گرفتن مرکز خوشه

کلاس SOStream

این کلاس الگوریتم SOStream را پیاده‌سازی می‌کند و شامل متدهایی است که شبه کدهای آن‌ها را در فایل مقاله دیده‌ایم

این کلاس شامل متد های زیر می‌باشد:

۱. find_neighbors
این متد برای پیدا کردن همسایه‌های خوشه برنده است.

۲. find_overlap
این متد برای پیدا کردن همسایه‌های خوشه خوشه برنده که با آن همپوشانی دارند به کار می‌رود.

۳. merge_clusters
این متد برای ادغام خوشه برنده با خوشه‌هایی که با آن همپوشانی دارند به کار می‌رود.

۴. update_cluster
به روز رسانی خوشه برنده و همسایگان آن هنگام اضافه شدن داده جدید

۵. fading_all
این متد برای از بین بردان خوشه‌ها با استفاده از محاسبه مقدار f(t) و مقایسه آن با fade_threshold به کار می‌رود.

۶. adjust_centroid
این متد برای تنظیم مرکز خوشه به کار می‌رود که در متد update_cluster نیز استفاده شده است.

۷. insert
این متد برای درج یک خوشه در لیست خوشه‌ها به کار می‌رود.

۸. get_centroids_of_clusters
این متد برای گرفتن لیست مراکز خوشه‌ها به کار می‌رود.

۹. calculate_purity
این متد برای محاسبه خلوص خوشه‌ها و داده‌ها در این الگوریتم به کار می‌رود.

۱۰. باقی متدها
بقیه متدها برای گرفتن مقادیر مورد استفاده در الگوریتم هستند.

main.py

این اسکریپت برای اجرای الگوریتم SOStream استفاده می‌شود. ابتدا دیتای مورد نظر را از فایل مربوطه می‌خوانیم و یک جایگشت رندوم روی آن اجرا می‌کنیم تا در زمان از بین بردن خوشه‌ها که نسبت به زمان ساخت و اضافه شدن خوشه‌ها و داده‌ها کار می‌کند به مشکلی بر نخوریم.

سپس مقادیر موردنیاز، نظیر آلفا، minPts، merge_threshold، لاندا و fade_threshold را نسبت به شرایط مختلف مقداردهی می‌کنیم و الگوریتم را به صورت زیر اجرا می‌کنیم.

هر داده موجود در دیتاست را بصورت تکی به برنامه وارد می‌کنیم و نسبت به حالت فعلی داده‌ها و خوشه‌ها، نسبت به به روز رسانی یا ادغام یا موارد دیگر اقدام می‌کنیم. هر ۲۰ مرحله (هر بار که ۲۰ داده به برنامه وارد شده باشد) نیز از بین بردن خوشه‌ها را طبق fading_all انجام می‌دهیم و همچنین هر ۲۵ مرحله نیز تعداد خوشه‌ها و مقدار خلوص را ذخیره می‌کنیم تا در انتها در نتیجه کار تغییرات آن‌ها را ببینیم.

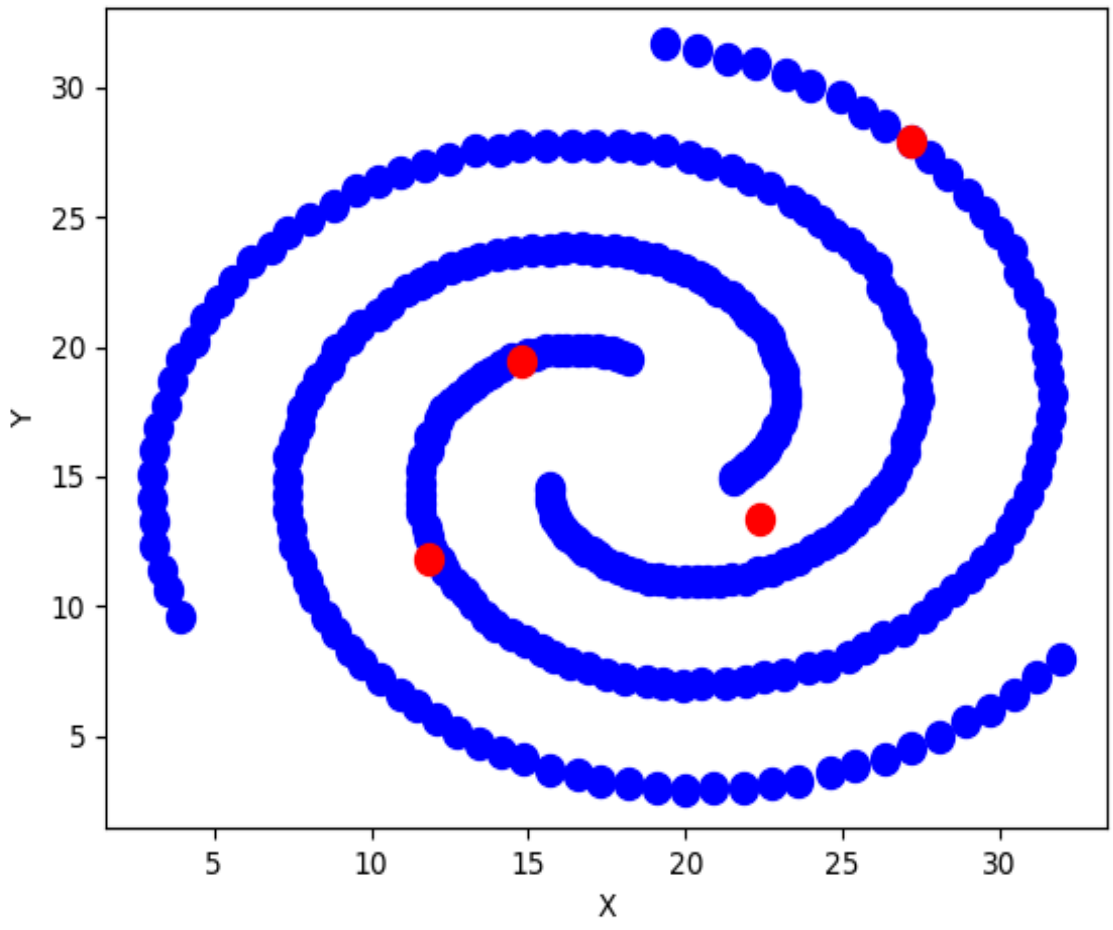
با توجه به محاسبه شدن موارد مختلف در الگوریتم و ذخیره کردن تعداد خوشه‌ها و مقدار خلوص در هر ۲۵ مرحله، می‌توانیم نمودار های خطی و میله‌ای متناسب با آن‌ها را نمایش بدهیم و همچنین بعد از آن، نتایجی مثل زمان پردازش، تعداد خوشه‌های نهایی، تعداد خوشه‌های از بین رفته، تعداد خوشه‌های ادغام شده و میانگین خلوص را نشان داده و بررسی میکنیم.

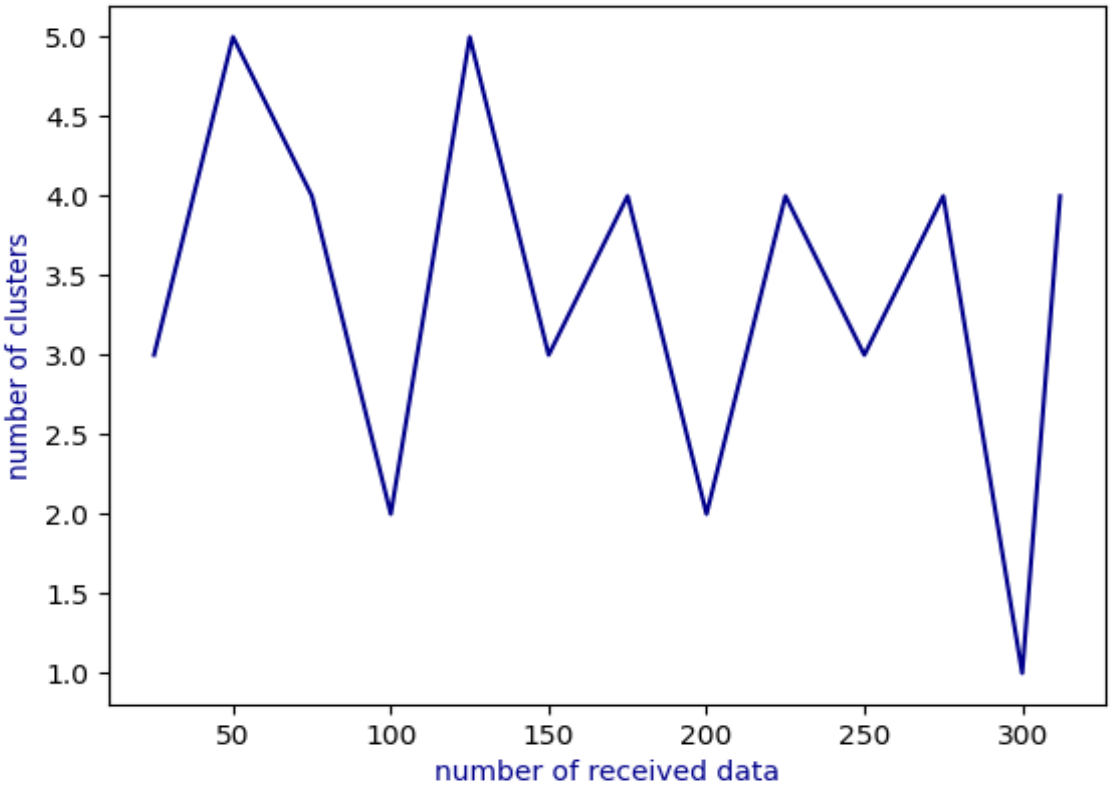
در ادامه مثال هایی از اجرا برنامه روی دو دیتاست قرار داده شده را میبینیم. بدیهی است که با تغییر مقادیر آلفا، لاندا، merge_threshold یا ...، خروجی متفاوت خواهد شد. همچنین بخاطر جایگشت رندومی که در ابتدای کار روی دیتاست اعمال می‌شود نیز هر بار خروجی متفاوت خواهد بود.

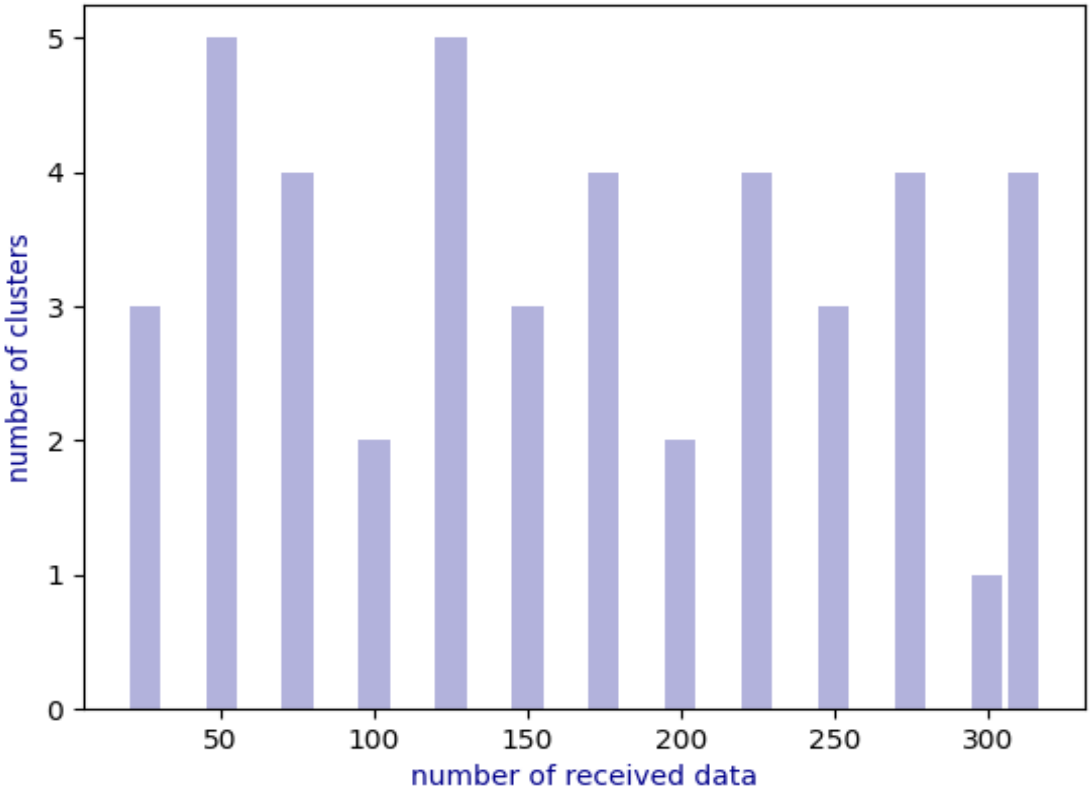
در نمودارهای اول نقاط آبی رنگ، داده‌های ورودی و نقاط قرمز رنگ، مراکز خوشه‌های نهایی هستند.

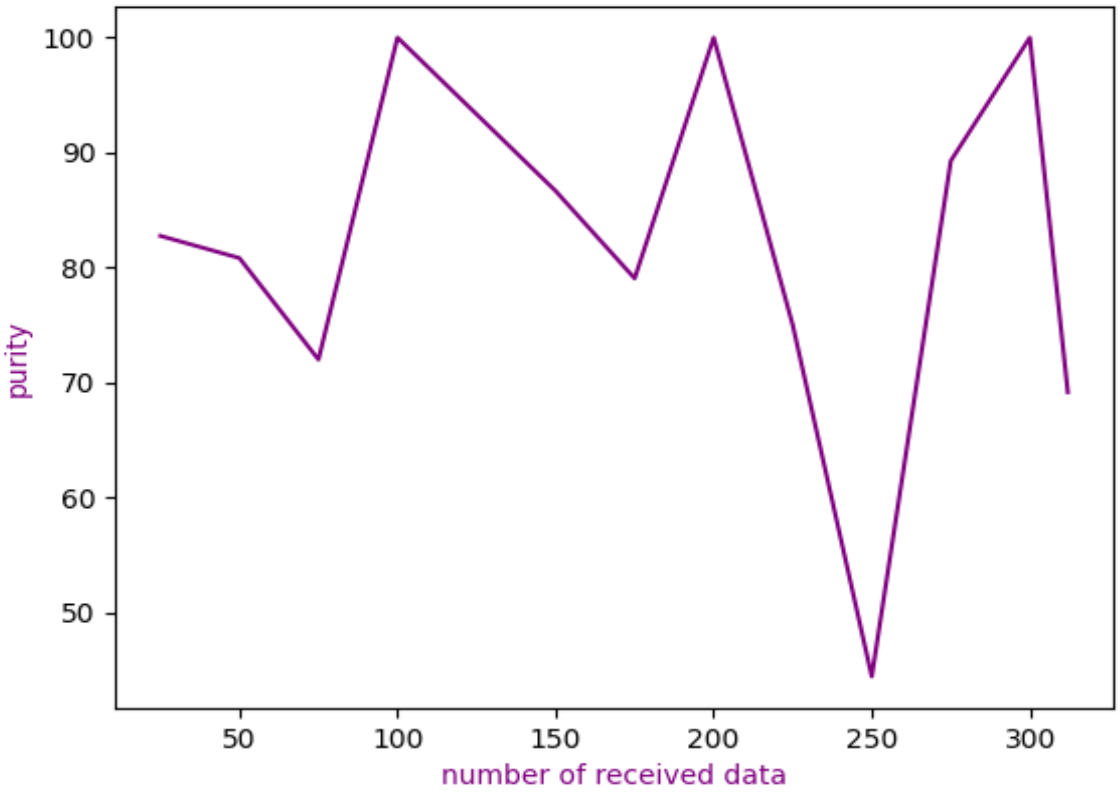
Dataset_۱.csv

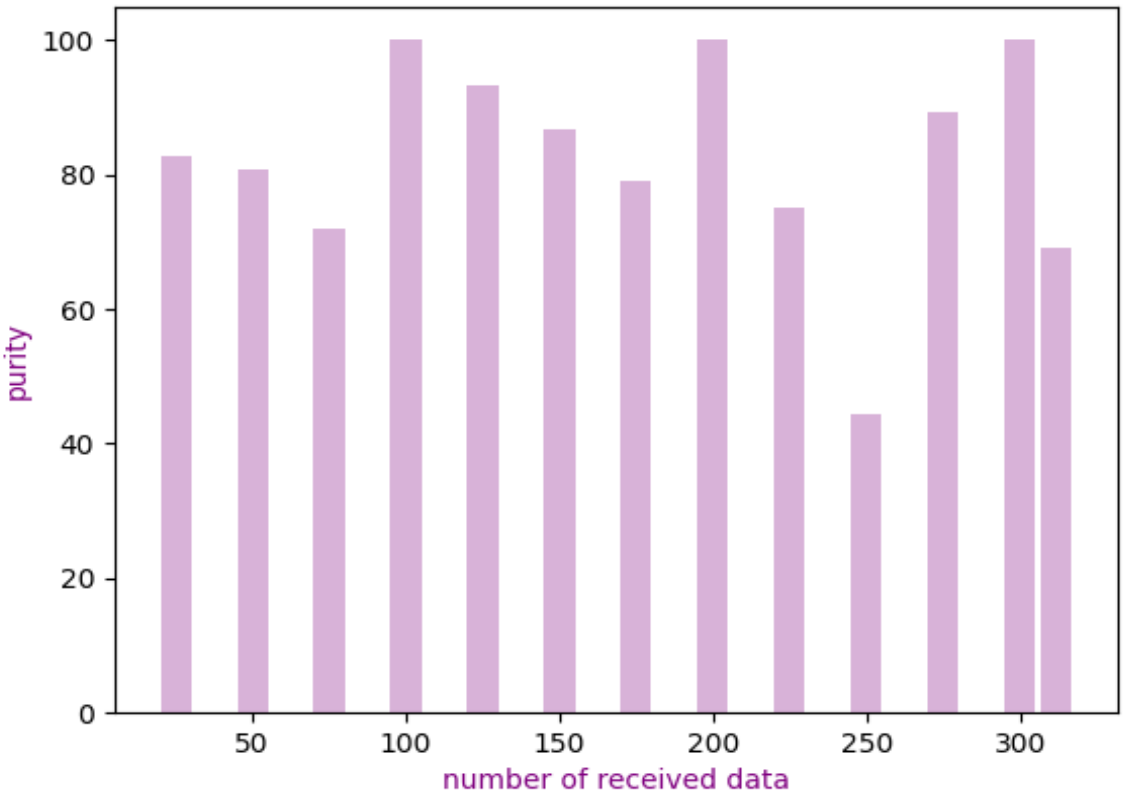
```
# Create SOSTream object
sostream = SOStream(data=data_frame,
                    alpha=0.1,
                    min_pts=3,
                    merge_threshold=7.75,
                    lambda_t=0.2,
                    fade_threshold=6.5
                    )
```




number of clusters
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
50
100
150
200
250
300
number of received data


number of clusters
5
4
3
2
1
0
50
100
150
200
250
300
number of received data

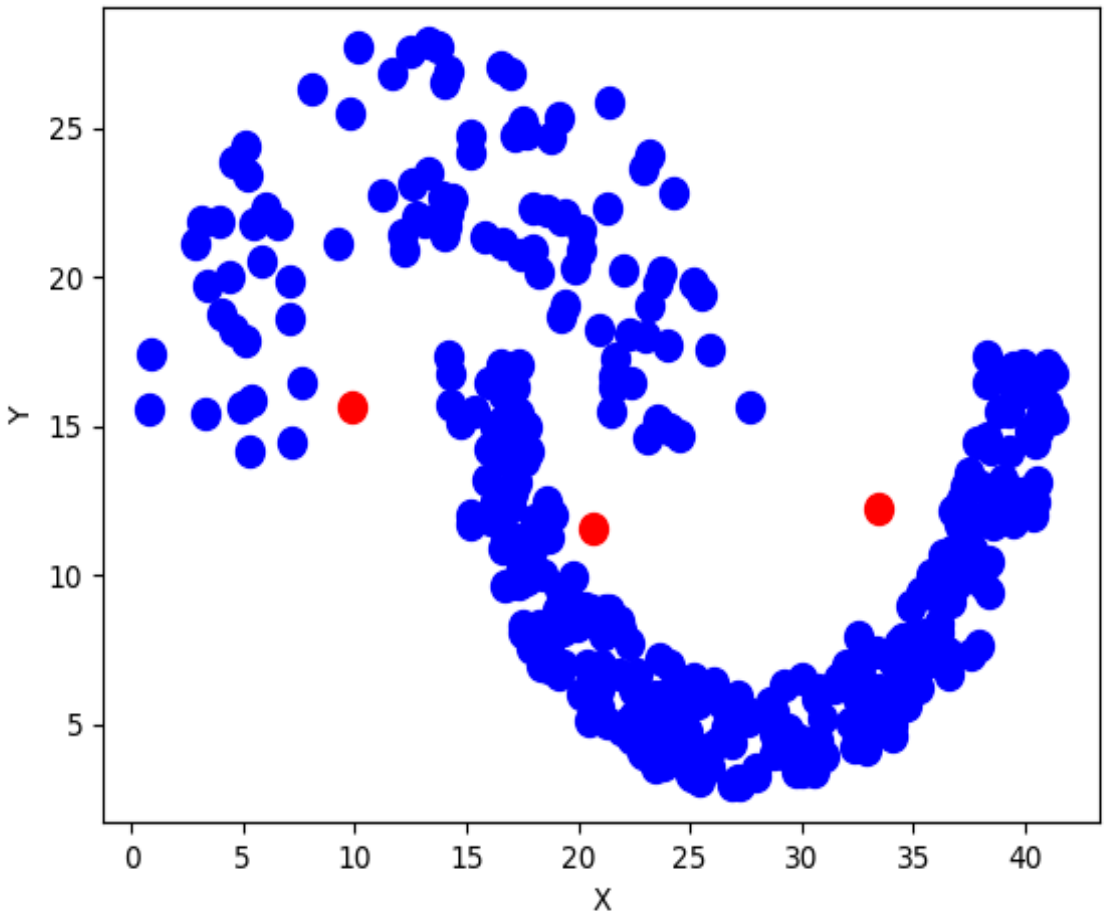

100
90
80
70
60
50
purity
50
100
150
200
250
300
number of received data


100
80
60
40
20
0
purity
50
100
150
200
250
300
number of received data

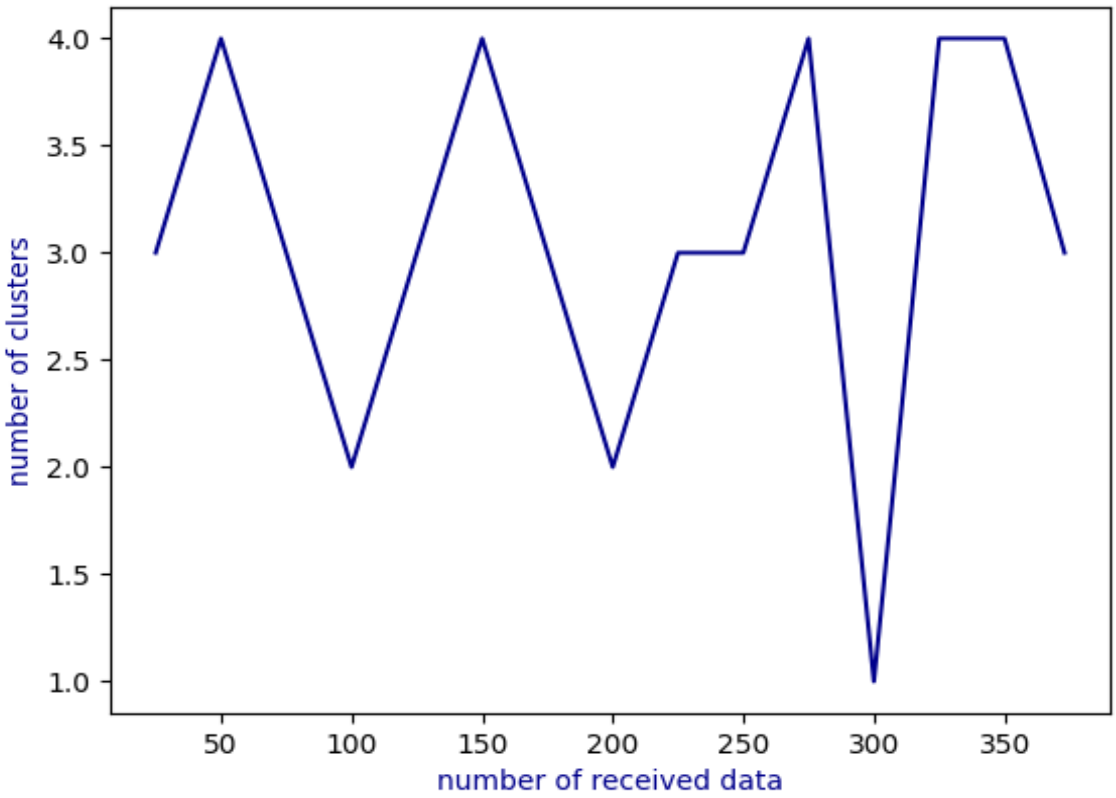
```
Process time =:  1.99
Final number of clusters:  4
Number of faded clusters:  31
Number of merged clusters:  156
Average purity:  82.50
```

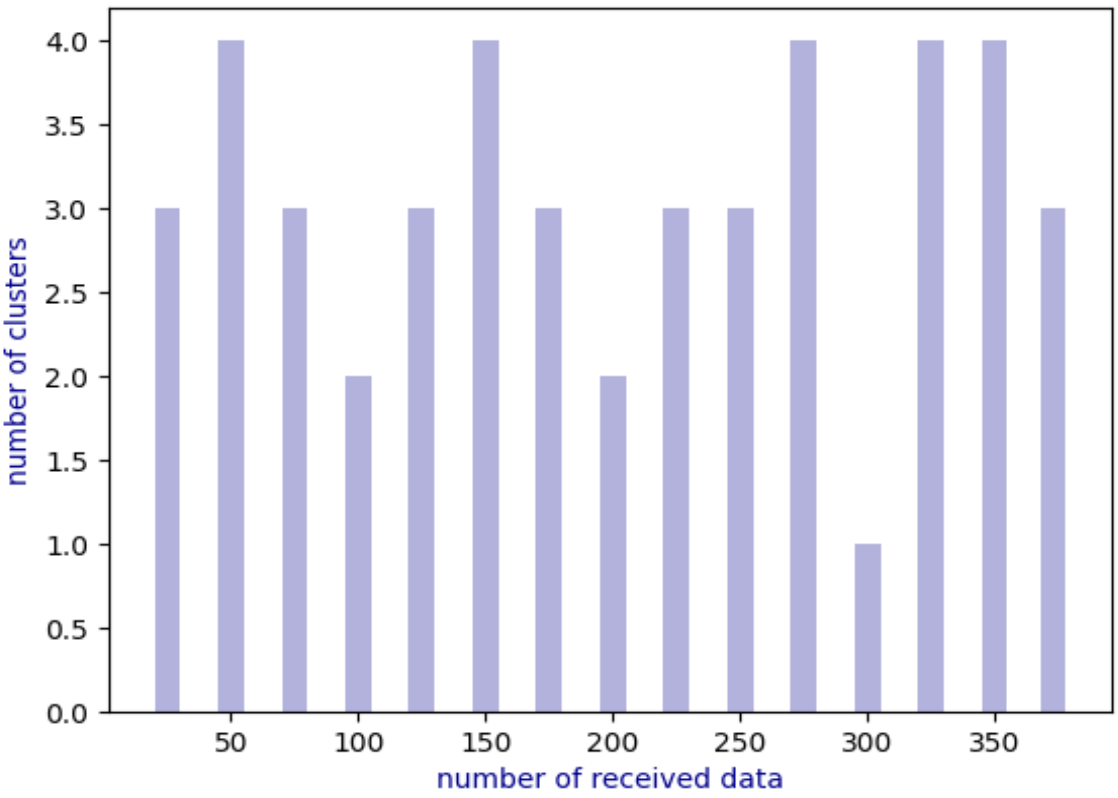
Dataset_۲.csv

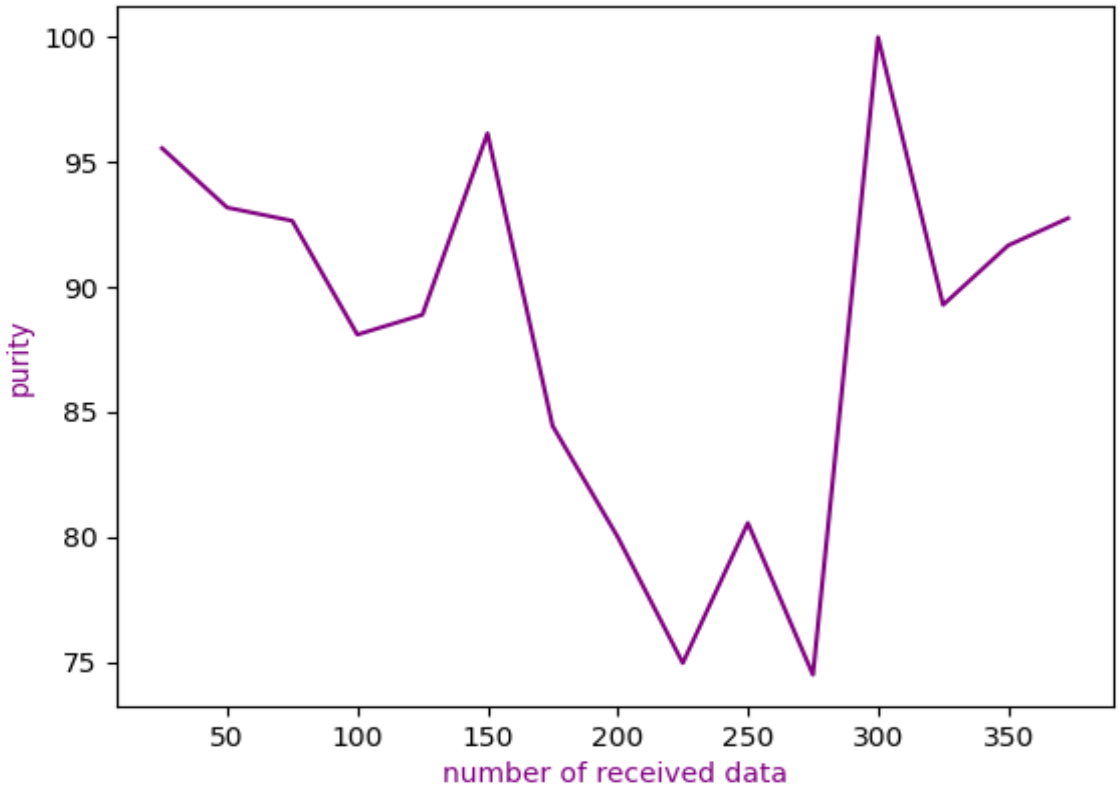
```
# Create SOSTream object
sostream = SOStream(data=data_frame,
                    alpha=0.1,
                    min_pts=3,
                    merge_threshold=9,
                    lambda_t=0.3,
                    fade_threshold=10
                    )
```

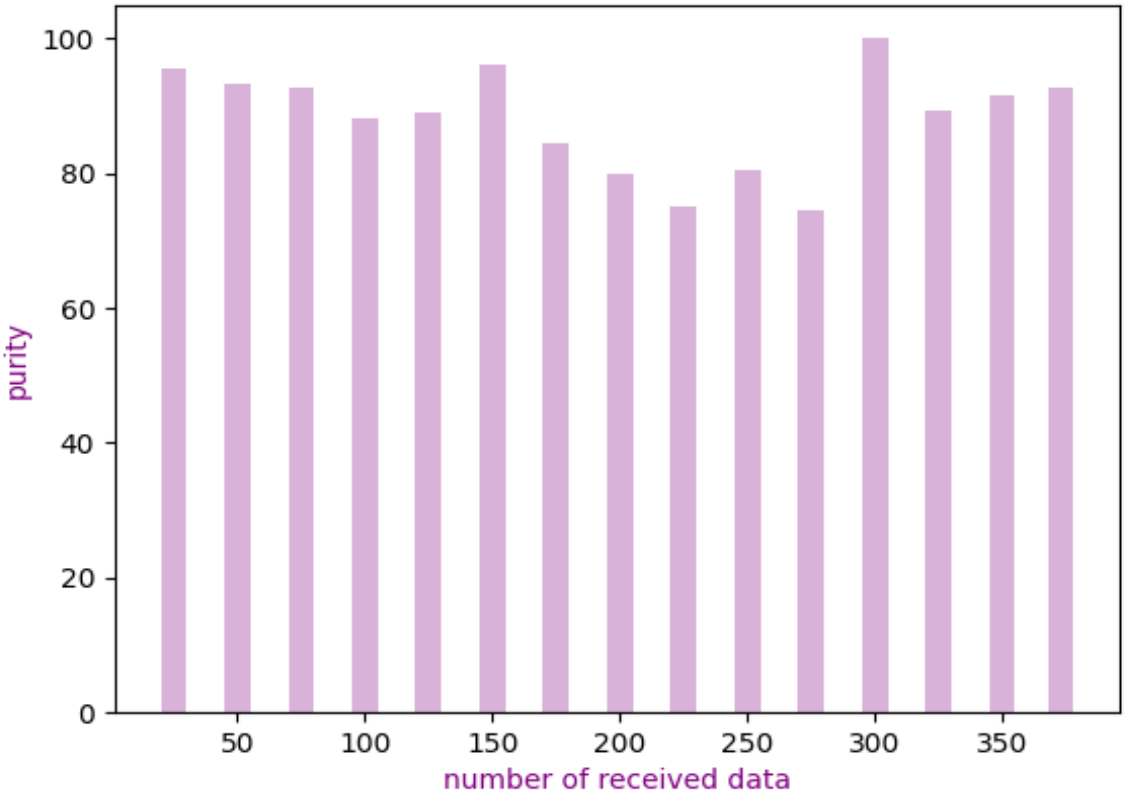


^


number of clusters
4.0
3.5
3.0
2.5
2.0
1.5
1.0
50
100
150
200
250
300
350
number of received data


number of clusters
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
50
100
150
200
250
300
350
number of received data


100
95
90
85
80
75
purity
50
100
150
200
250
300
350
number of received data


100
80
60
40
20
0
purity
50
100
150
200
250
300
350
number of received data

```
Process time =:  2.19
Final number of clusters:  3
Number of faded clusters:  37
Number of merged clusters:  205
Average purity:  88.18
```