STU Bratislava, Fakulta informatiky a informačných technológií

Umelá inteligencia

Zadanie 4a – klasifikácia

Dokumentácia

Meno: Andrii Rybak

AIS ID: 105840

Cvičiaci: Ing. Ivan Kapustík

Akademicky rok: 2021/2022

Andrii Rybak
ID: 105840

Content

Algorithm and data representation 3

Time and space complexity 3

Tests (experiments) 4

Experiment #1 4

Experiment #2 5

Experiment #3 6

Experiment #4 7

Conclusion 8

Andrii Rybak
ID: 105840

Algorithm and data representation

The solution was developed in Python 3.9 in PyCharm.

Generated points are represented as two lists. First list contains lists with x and y coordinates, and the second list contains types (colors) of the points. This was made in order to optimize the solution.

Classified points are stored in one dimensional list and are represented as list with 3 elements: [x, y, color]. Classified points are also duplicated in 3-dimentional list with name grid. First and second dimensions represent x and y axis (coordinates of the square) in the grid, and the third dimension represents points that are contained in certain square.

First 4000 points are classified by counting colors of k nearest neighbors. In this case the distance is calculated to all points in the grid without optimizations to reduce the complexity. The new point's color is set according to the most common color among the neighbors.

Next 16000 points are classified using complexity optimization. In this case, the whole surface is divided into squares, that contain points. So, there is no need to calculate distance to every point in the surface, but it is enough to get distances to points that are located in current and neighbor squares. However, the neighbor squares are got by calculating distances between current and all other squares.

Size of every square is 400x400 points, so the whole surface is divided into 25 squares.

Generated coordinates are unique, and two points generated in a row are not of the same color.

The reason why first 4000 points are classified using brute force approach, is that it provides the most correct results, and the further classification using grid will be fast without significant decrease of success rate.

Time and space complexity

Time complexity – O(n^2), where n is number of points to classify.

Space complexity – O(n), where n is number of points to classify.

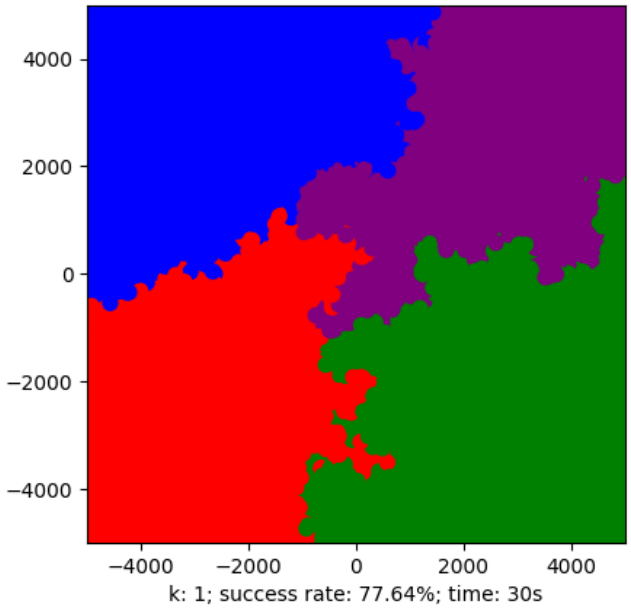
Andrii Rybak
ID: 105840

Tests (experiments)

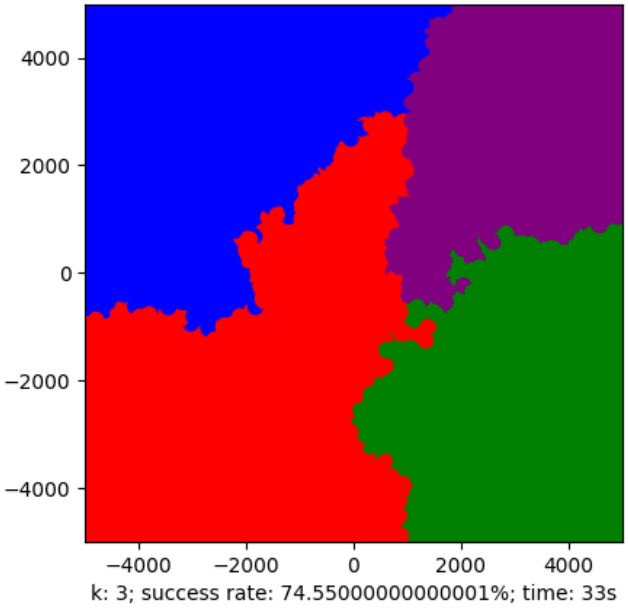
Every experiment is performed by classifying the same points with different k (number of neighbors).

There were made 4 experiments as required by the task description.

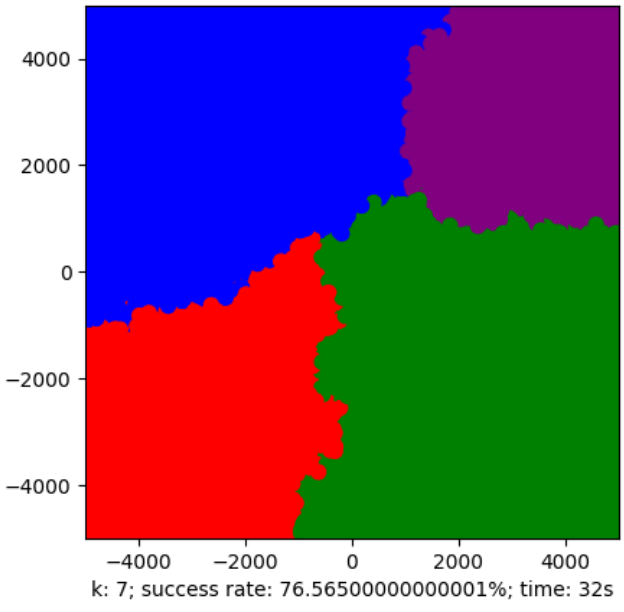
Experiment #1



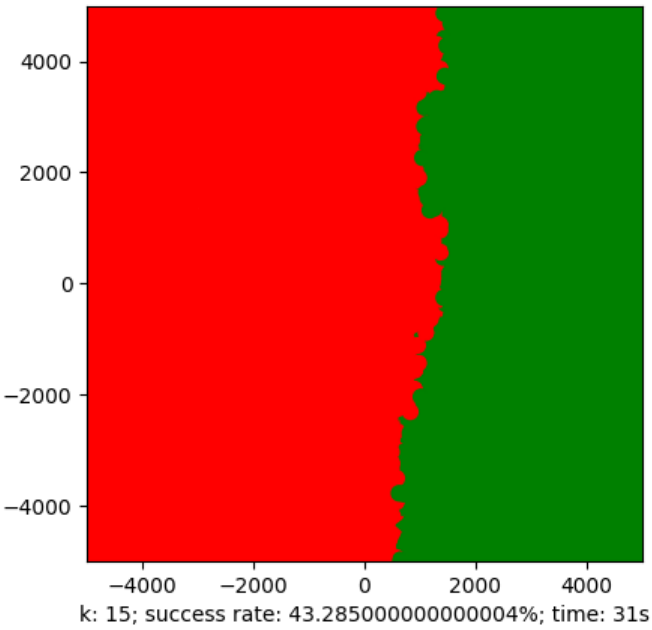
k: 1; success rate: 77.64%; time: 30s



k: 3; success rate: 74.55000000000001%; time: 33s



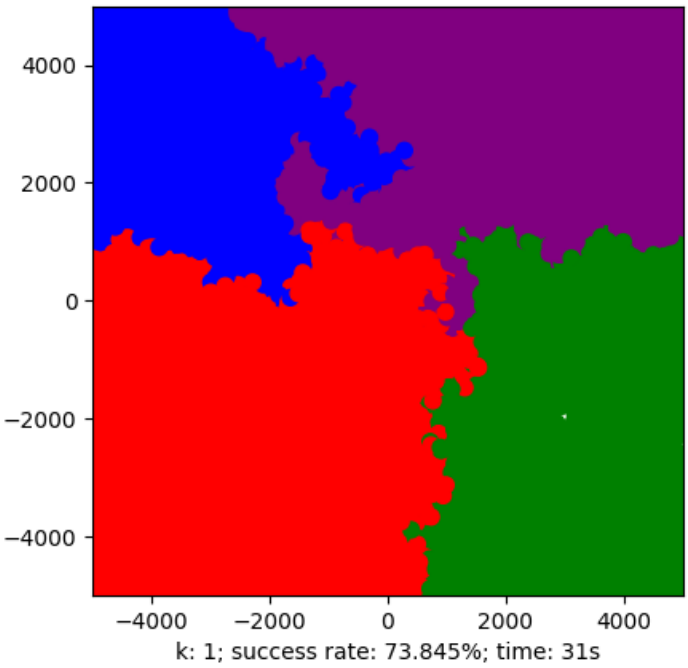
k: 7; success rate: 76.56500000000001%; time: 32s



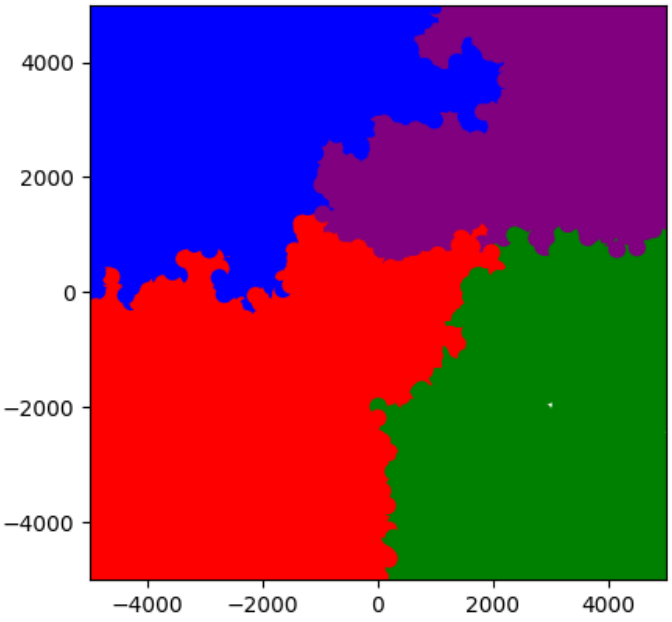
k: 15; success rate: 43.285000000000004%; time: 31s

Andrii Rybak
ID: 105840

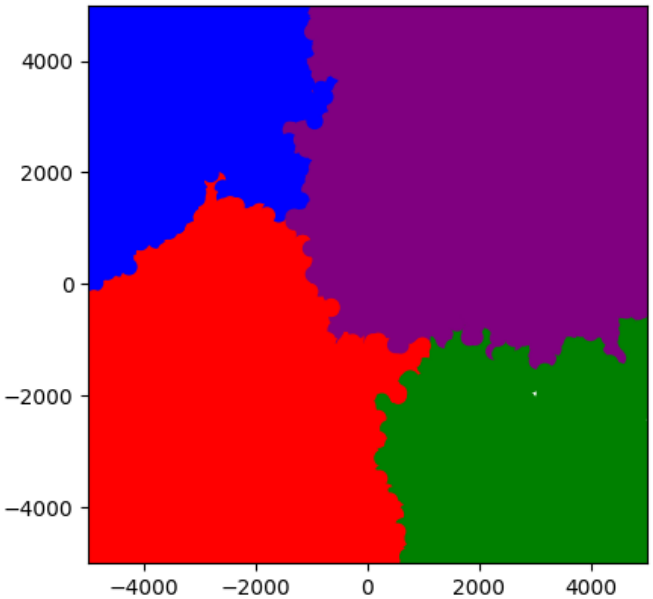
Experiment #2



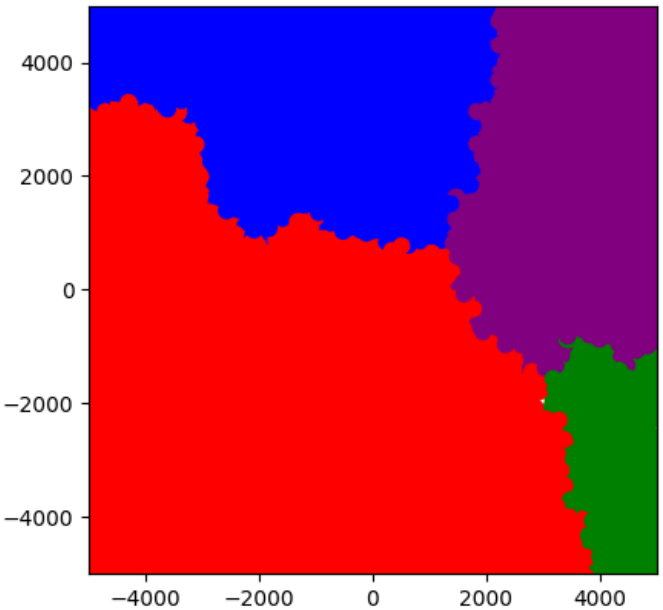
k: 1; success rate: 73.845%; time: 31s



k: 3; success rate: 75.53999999999999%; time: 31s



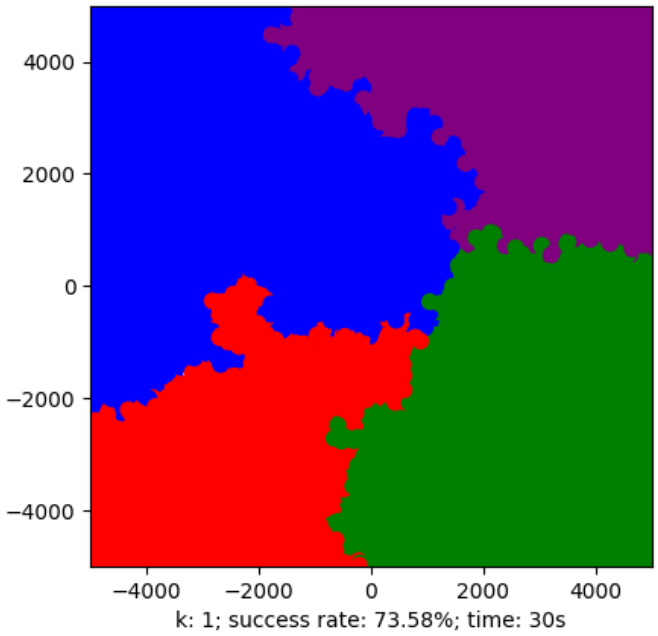
k: 7; success rate: 74.77000000000001%; time: 30s



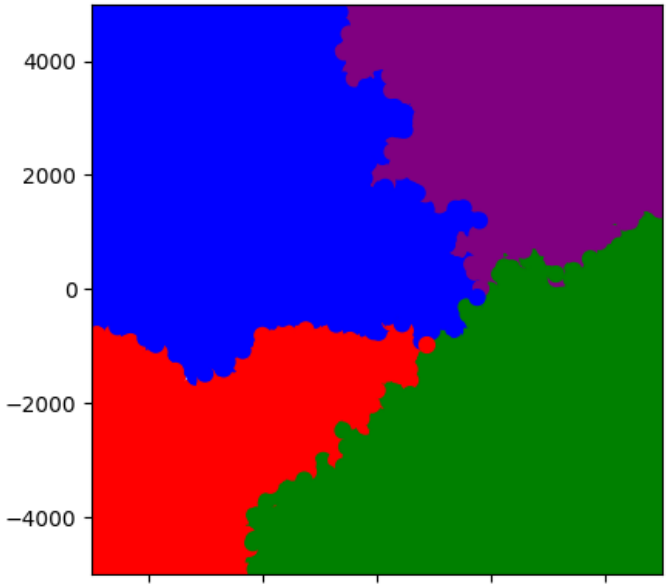
k: 15; success rate: 58.225%; time: 31s

Andrii Rybak
ID: 105840

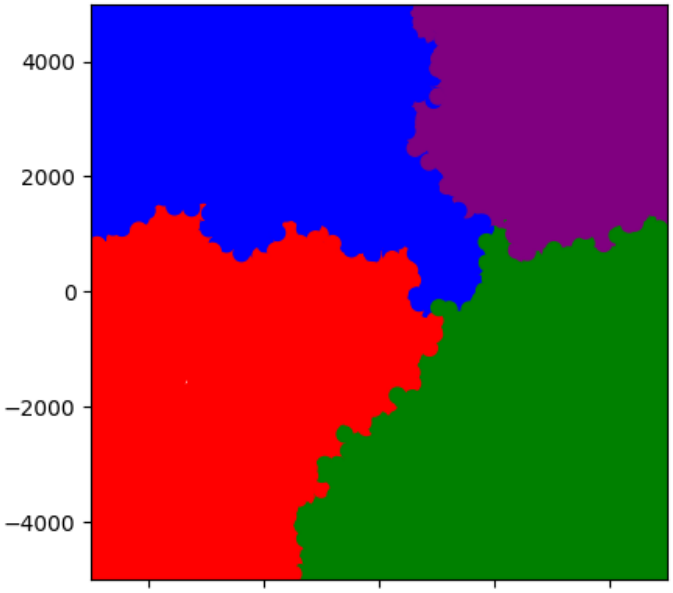
Experiment #3



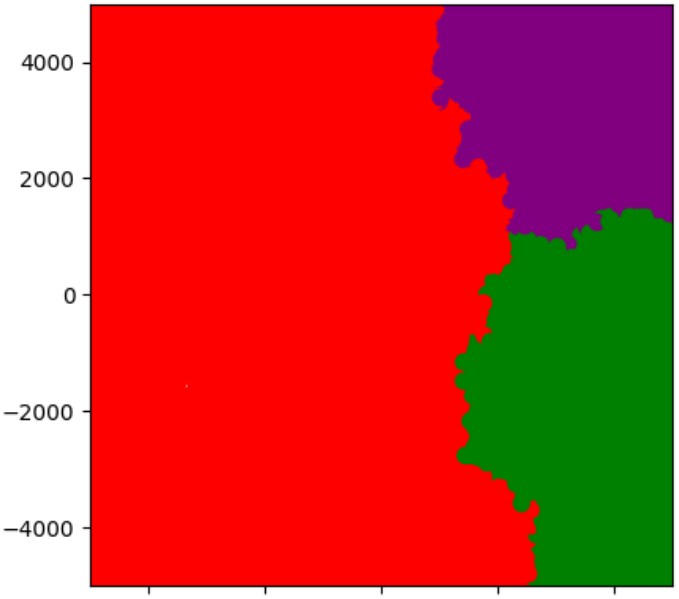
k: 1; success rate: 73.58%; time: 30s



k: 3; success rate: 74.28%; time: 31s



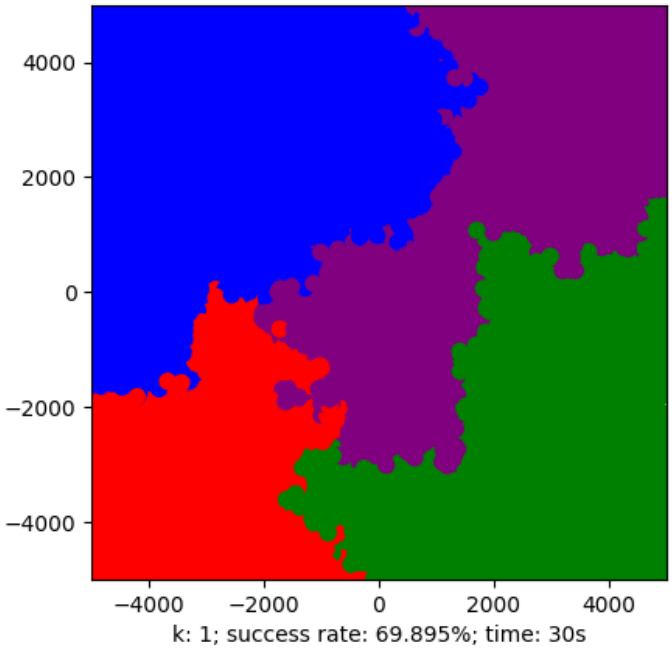
k: 7; success rate: 74.37%; time: 30s



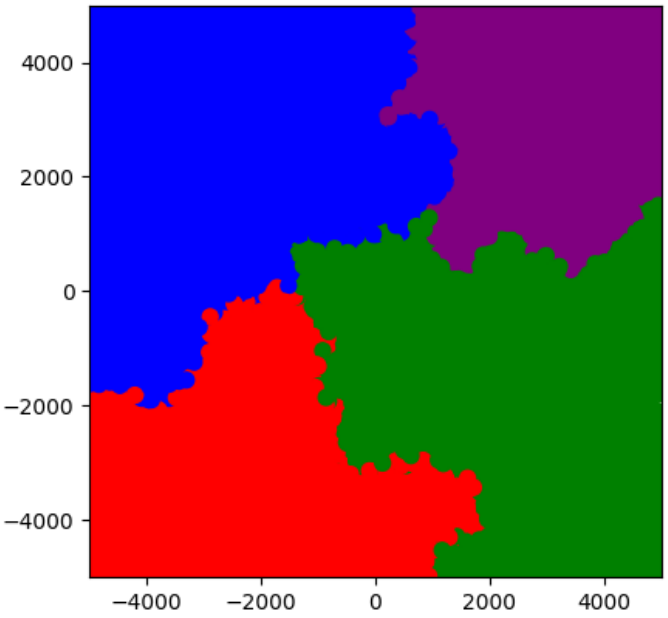
k: 15; success rate: 50.015%; time: 31s

Andrii Rybak
ID: 105840

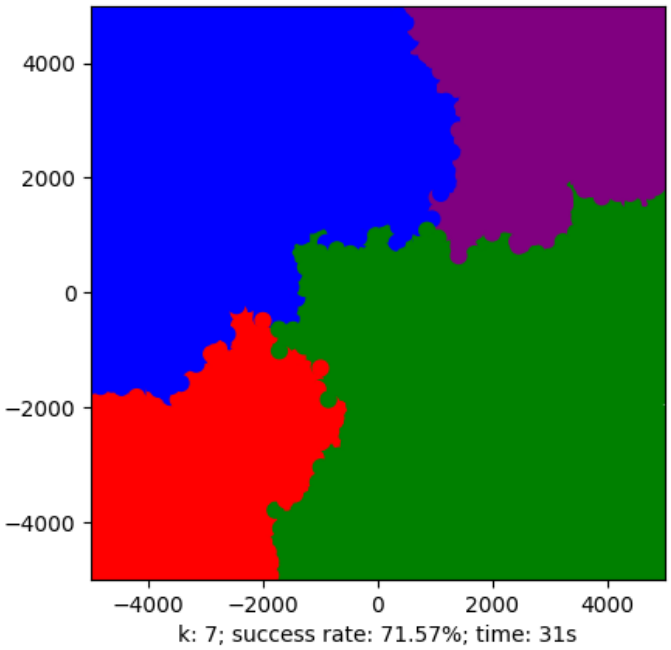
Experiment #4



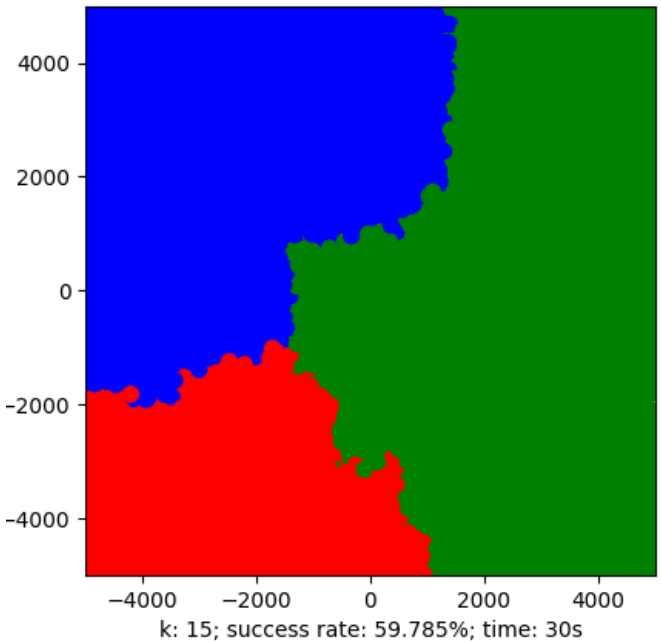
k: 1; success rate: 69.895%; time: 30s



k: 3; success rate: 74.275%; time: 30s



k: 7; success rate: 71.57%; time: 31s



k: 15; success rate: 59.785%; time: 30s

Andrii Rybak
ID: 105840

Conclusion

As can be seen from the experiments above, the average success rates are:

- k = 1: 73,775%
- k = 3: 74,625%
- k = 7: 74,31%
- k = 15: 52,875%

The success rate is relatively high and is similar when k is 1, 3, 7, but significantly decreases when k is 15. Such situation can be explained in such way, that there are not enough initial representatives of groups on the surface, so the success rate decreases. Such problem can be solved by adding more initial points in same amount for every group.

Generally, all experiments show similar results.

Execution time does not depend on k and is 31 seconds in average.

This solution is much faster that simple brute force approach and does not significantly decrease the success rate. Classifying 20000 point would take about 330-350 seconds for brute-force to execute, while this solution takes only 31 seconds.