Home assignment
Machine Learning position

Kendaxa

Abstract—The home assignment is split into three parts. The first is a simple algorithmic assignment with the goal to evaluate programming skills and basic skills with working with graphs. The second and third are more open and their goal is to evaluate analysis skill on a real world data.

1 Introduction

The goal of these short assignments is to help us evaluate your proficiency with python, algorithm design, machine learning and data science in general. The output is expected to be both a formal report and a runnable solution reproducing your results.

2 Algorithmic task

2.1 Problem definition

Let $\mathcal{G}$ be a connected undirected graph without cycles (a tree graph), where each leaf is colored either using white, blue or red color. The task is to find the maximum number of different pairs of leaves, where one leaf is red and the other blue such that all pairs can connected by mutually disjunctive paths — these paths connects leaves of each pair and these path has no common node.

2.2 Task

Your goal is to write a program in Python 3, that takes the input describing the graph from the standard input/file and returns a single number on the standard output with the answer to the problem. The program has to be efficient and quick (e.g., brute-forcing is inadmissible). **The solution should follow all coding standards common in python** (including docstrings for building documentation), comments, etc.

The solution should also be reproducible with as least work as possible. Ideally, you prepare a docker image which can be used for running the program; if you are unfamiliar with the docker, usage of the Anaconda virtual environment is recommended (provide also the file defining the environment). You are given some tests inputs, please incorporate them to your unit tests but do not limit the unit tests to them, also test other parts you think should be tested using unit tests. Your program should contain a basic CLI. The CLI can be used , for example, for setting the verbosity, for setting whether the input is taken from the standard input or a provided file, etc.

You are also supposed to write a brief report with illustrations describing your approach and algorithms you have used to tacking the problem. The report should clearly contain the information about the asymptotic complexity of your solution with explanation. More details about the report requirements are in the section 5.

2.3 Input

- The first line contains 3 numbers (integers); denoted M, R, and B, where
  - M — the number of edges,
  - R — number of red leaves,
  - B — number of blue leaves.
- Then there are M rows, where each row represents an edge. It contains two numbers n_1 and n_2 — indices of nodes that are connected by the edge. The nodes are indexed from 1.
- Then you get one row of R numbers — indices of leaves colored red.
- Then you get one row of B numbers — indices of leaves colored blue.

You can expect $3 \leq N \leq 10^6$ and that there is at least one blue leaf and at least one red leaf.

2.3.1 *Example input*

```
6 2 2
1 2
2 3
4 2
5 7
6 5
4 5
1 3
7 6
```

The graph listed above is also shown in fig. 1.Another example (Case 3) is shown in fig. 2.

2.3.2 *Expected output*

```
1
```

2.4 Additional testcases

The additional testcases are located in the provided archive. The expected running time is at most few seconds for each test case.

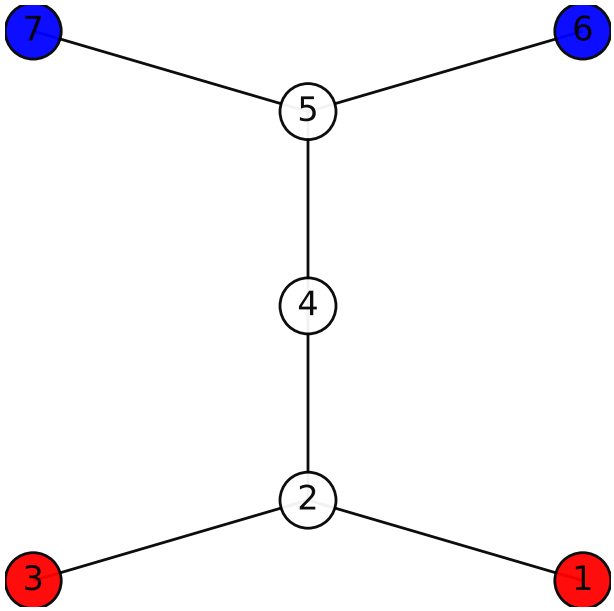

Figure 1. The input graph for the Case 1

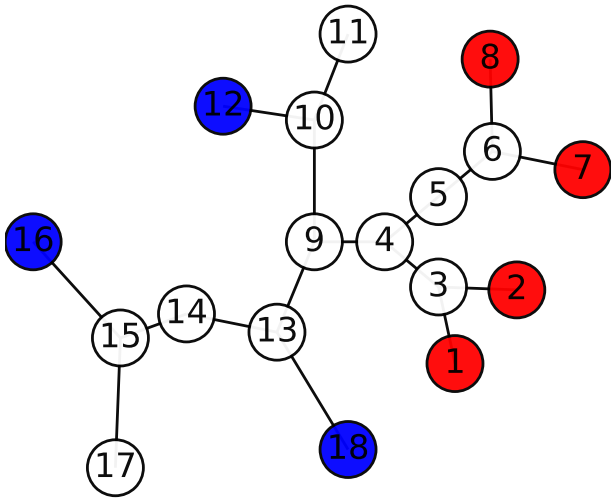

Figure 2. The input graph for the Case 3

3 Classification Task

Your goal is to perform **exploratory data analysis** (EDA) and to train and compare few models on a classification task using the dataset Amphibians (https://archive.ics.uci.edu/ml/datasets/Amphibians). Your task is to evaluate the prediction of the presence of individual amphibians, namely *green frogs*, *brown frogs*, *common toad*, *fire-bellied toad*, *tree frog*, *common newt*, and *great crested new*. You are not to use the *ID* and `MV` features.

Since your goal is to evaluate and compare several models along with finding the best one, you have to use some kind of cross-validation as the dataset is quite small (which is very common for some of the real world datasets).

If you find it applicable, use statistical tests in the EDA and comparison to distinguish between insignificant differences and significant ones.

3.1 Required output

The required output is a Jupyter notebook with the EDA along and the model comparison (if, for some reason, you are unable or do not want to use Jupyter notebooks, provide the EDA as a python script but describe it in much finer detail in report and add the relevant plots there as well). If the models are trained outside the notebook, please attach the codes for the model training as well. You are also required to briefly summarize the results of the model comparison in the report. More details about the report requirements are in the section 5.

4 Regression Task

Your goal is to perform **exploratory data analysis** (EDA) and to train and compare few models on a regression task. Your task is to predict day-ahed daily volumes of the S&P 500 index using any available information from the past; i.e., you are going to predict the volume v_{t+1} using the information available on days $t, t-1, \ldots$.

Evaluate the models performance on out-of-sample data using data from 2017 and 2018 (i.e., January 1st, 2017 – December 31st , 2018).

Do not forget that you can use (and should) data outside the series itself — for example, calendar with known events.

Since your goal is to evaluate and compare several models along with finding the best, you have to use some kind of cross-validation as the dataset is quite small (which is very common for some of the real world datasets).

If you find it applicable, use statistical tests in the EDA and comparison to distinguish between insignificant differences and significant ones.

4.1 Required output

The required output is a Jupyter notebook with the EDA along and the model comparison (if, for some reason, you are unable or do not want to use Jupyter notebooks, provide the EDA as a python script but describe it in much finer detail in report and add the relevant plots there as well). If the models are trained outside the notebook, please attach the codes for the model training as well. You are also required to briefly summarize the results of the model comparison in the report. More details about the report requirements are in the section 5.

5 Report

You are required to write a brief report in the PDF format (LaTeXusage is recommend) summarizing the approaches and presenting the results for all three subtasks. It is recommend to use figures and plots where it will help you make your point. The report should contain all the necessary details to understand what approach you have undertaken, what were the results and how you interpret them.

5.1 Algorithmic task

Briefly describe your solution and your algorithm including the asymptotic complexity of your algorithm together with explanation of the complexity. Also provide some visualization of how your algorithm works (ideally output generated by the algorithm itself so it can be generated for different inputs).

Provide a simple benchmark of your algorithm for all the provided test cases measuring each test case individually. Focus mostly on timings but you can also discuss memory consumption, etc., do not rely on a single run. You can also provide a figure depicting the benchmark. Also discuss the asymptotic complexity of the algorithm with respect to size of input.

5.2 Classification and regression tasks

Your report should summarize the main results of your EDA but it is sufficient to have the details of the EDA only in the Jupyter notebook. Briefly (very briefly) introduce the used models. You should compare the models with regards to more than one metric each with explanation when is each of the metrics preferable. You should also state your trust in the individual models — e.g., that even if some model gives you very good results, you still might not trust it because it is sensitive to the data changes. Compare the models also with respect to their robustness and interpretability. Interpret the few models you will select as your top ranking candidates, show which features they are relying the most, etc. Where applicable, perform formal statistical tests to support your results.

Please, also state the limitations of your work and directions, in which it can be expanded — it is expected that you will not be able to exhaust all possible approaches in the limited time. Please state which of the possible expansions are most promising and why.

5.3 Scope

The report is expected to have about 5–12 pages when using two-column format with figures but there are no hard limits as the **completeness of the presented information is the goal** (as long as there are no empty sentences or fillers, the length will not be evaluated).

6 Deadline

The deadline for submission is March 18th, 2021.

Appendix A
Notes

A.1 LaTeX editor

Unless you have installed LaTeX locally, it is recommended to use Overleaf v2 which provides an online editor and also has many predefined templates and also allows online collaboration (which might be useful for your other projects). The Overleaf v2 is the result of a merge of Overleaf v1 and ShareLatex several years ago.

If you have LaTeX installed **locally**, it is recommended to set the `matplotlib` with LaTeX which allows using LaTeX code inside the figure — e.g. for the legend or axis labels.

A.2 Using Unittests

While the tests can be launched from a terminal, you can launch them also directly from the PyCharm IDE and Visual Studio Code.

A.3 Saving figures

You should save the figures in a vectorized format, which is better for publication. This can be done easily in python using

```
plt.savefig('your_filename.pdf', dpi=500,
    transparent=True)
```

A.4 Questions

If you have any questions regarding the task, do not hesitate to contact me (v.kunc@kendaxa.com), I'll will try to answer within two days.