DSCI 552: Machine Learning for Data Science

Problem Set 4

Instructor: Dr. Marcin Abram – email: mjabram@usc.edu

Deadline: Thursday, March 18, 2020 at 10 am PDT

You can submit the solutions (pdfs) on Blackboard. The deadline to submit solutions is always the second Thursday at 10 am (PDT). As long as the problem set is open, you will be able to send multiple answers (the last submitted answer will matter).

Task (20 point)

You are a new hire in a mid-size company. You have just completed your third task. Your boss took your report and she went to meet the executive director of the hospital. You sit in the common room and wait for the meeting to be over. If you have learned anything in the past weeks, it's that you will be called as soon as the meeting is done. Therefore, you try to enjoy that little moment of freedom you have now.

It was already late afternoon, when you got the call. "Good job", you hear the voice of your boss in your phone, "The executive director of the hospital want to see you. Come here, please", she asked.

You enter the conference room. You see your CEO, your technical manager, the developer (why they are allways sooner then me?) and an another person – introduced as an executive director of the city hospital.

The executive director of the hospital said: "You did a fantastic job! With all the people you were able to find, there is enough material to conduct a large-scale study. What you did, really matters.

I'm here, to ask for your help with another task. Since the pandemic started, the number of patients with various lung complications increased dramatically. The radiologists and other specialists who are trained to interpret x-ray or CT-scans are overwhelmed. There is an urgent need for a system that can help us to sort the scans. Here are some examples, so you can see what I'm talking about". She gives you a folder with three images.

Pre-existing Conditions

Healthy

Effusion/Mass

"We would like you to build a classifier, that can help us to sort the scans into three categories. The first category are the healthy patients. The second category are patients with some pre-existing conditions, like aortic enlargement, cardiomegaly or pulmonary fibrosis. The third category are various, serious lung conditions that require our immediate attention, like pleural effusion.

There is a board meeting on Thursday, March 18, at 10 am. Will you be ready by then?"

Hint: This task is to build an image classifier. You have three classes. You have to propose a few CNN architectures, train your models, tune the hyperparameters, and test your models' performance. It is very important, that users of your models understand how good (or bad) your models are. This is one of the central part of your job – you must know how to evaluate your models – and how to communicate those results. When we grade, we will always look at the section that describe your testing methodology.

Remember, we do not grade on a curve. You don't have to be better then your collegues to get 100% from that task. Your model can have a moderate performance – but as long as your training and testing methodologies are good, and as long as everything is well described, you can get 100% for this problem set.

We added some baselines on Kaggle, to help you gauge if you move into a good direction. You do not have to beat all the baselines to pass this exercise. However, if your score is below the low baseline, it might indicate that your solution has still a room for improvement.

Your Technical Manager said: "I looked at the data. I noticed, that the labels are quite unbalanced. Be careful how you train your model and how you measure the performance. Depending on how you construct your test set (how balenced or un-balanced that test set will be), your measured accuracy (or micro/macro AUC or F1 score – whatever you decide to use) might be very different. In your report, describe not only how good is your final model, but also, in detail, how you measured the performance."

Hint: If you don't know what to do: read chapter 14 from "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (2019).

The senior developer took you aside and said: "My task will be to maintain your code. However, remember that I'm not a specialist in neural network – like you. Write comments and try to explain any nontrivial part of your code!"

Hint: If you use Jupyter Notebooks, remember that you can also add special "markdown cells". You can use it to split your notebook into a few logical parts – see also post @87 on Piazza.

Learning Objective: You will be able to train an image classifier. You will be able to work with un-balanced labels.

Data

First, enter the competition by accepting the terms at `https://www.kaggle.com/t/91a75288c54e48dbb2f9904318af27dd`. Then, you will be able to find the dataset at `https://www.kaggle.com/c/usc-dsci552-section-32415d-spring-2021-ps4/data`. There are two files. The first file `ps4_trainvalid_images.npy` contains the xrays (gathered together in one large numpy array). You can find the labels in the corresponding csv file, `ps4_trainvalid_images.csv`. Use those images and labels to train and to validate/assess your models. The second data file `ps4_kaggle_images.npy` contains images for the Kaggle competiotion.

Kaggle Competition

The Kaggle submission file should have only two columns. First column is called `Id` and should contain the row number: 0, 1, 2, ..., 299 (intigers). Second column is called `Predicted` and should contain your prediction if the patient is healthy (0), has pre-existing conditions (1) or has Effusion/Mass in the lungs (2).

You don't have to beat all the baselines for your solution to be correct. There were created just to give you a sense of what is possible – and to give you an opportunity to check if you go in right direction (score significantly lower then the lower baseline might indicate, that your solution has still a room for improvement).

Report

To help you, I prepared a template for you, see `https://www.overleaf.com/read/qcbcsdscvcdb`. You are encouraged to use the template, but if you prefer to use another editor – or to made some modifications – this is acceptable. Just remember to always export your report to the pdf format at the end.

Report Submission

Submit your report (a pdf file) using Blackboard (check the “Assignments” section on Blackboard page of our class).

Code Submission

To accept the assignment, use the secret link `https://classroom.github.com/a/uVs88uox`.

Upload your code to a private GitHub repository linked to our GitHub Classroom, `https://classroom.github.com/classrooms/77759207-usc-dsci552-32415d-spring2021-classroom`.

Please, remember that your GitHub name is linked to your student profile in our GitHub Classroom. If you use Python, you can commit `.py` or `.ipynb` file(s). If you use other programming languages, like `R`, `C++`, `Java`, `Julia`, etc., prepare a description that would explain how to compile (if necessary) and run your code. If you submit more than one file (or one Jupyter Notebook), include `Readme.md` file, where you would explain what each file is about (so we know in which order we should read the files).

Grading Rules

In order to grade your work, we will role-play the following situation. We will assume, that you are a new-hire in our company. You were asked to provide a comprehensive technical report, that illustrates your findings. We will evaluate it from the perspective of three people.

- Your CEO (she would like to hear the high-level stuff; She will probably read only the abstract and the conclusions, maybe she will also look at the main figure or the main plot). *(4 points for the report)*

- Your manager (he would like to see a detailed report; he might also look at some parts of the code). *(6 points for the report and 2 points for the code)*

- A senior developer (they would like to see your code; they might not read the report at all). *(8 points for the code)*

Your final score is the sum of scores given by each person. Cumulatively, there are 10 points for the report and 10 points for the code.

Each person (the CEO, the manager, the developer) will use the following grade rubrics.

Grade Component	Meets Expectations	Approaches Expectations	Needs Improvement
Completeness			
Clarity and Support			
Validity			

By "Completeness" we mean, that all parts of the question are addressed. By "Clarity" we mean, that the text and the code is written in accessible way. By "Support", we mean that you provide sufficient evidences to back-up your statements. You must cite any source you use (even if you happened to copy or adapt a snippet of a code from the internet – you should still treat it as a citation. You should clearly mark how large is that snipped and provide adequate references).

Don't Panic

Don't panic. We understand, that this is a large, open-ended task. We also understand, that this might be only your fourth technical report that you were tasked to write. We are dedicated to help you do the best work – and while we keep high standards for you (and for us) – in the same time, we acknowledge, that you have limited time and limited resources to complete your task. This report doesn't have to be perfect to give you 100% score.

If you don't know where to start, read the Chapters 11-15 from "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 2nd Edition by Aurélien Géron. Check also the *Appendix B. Machine Learning Project Checklist* from that book.

If something is not clear, ask questions on Piazza.