DSCI 552: Machine Learning for Data Science

Problem Set 5

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

Deadline: Thursday, April 1, 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 fourth task. This one was tough! You don't even know how you survived the last two weeks.

You hoped to have a moment of rest – however, the rest was not given to you. The CEO called you again. You enter the conference room. Moment later the technical manager and the developer enter the room as well.

The CEO said: "You did a solid job. We are happy to have you in our company. However, we have another task for you. We got a federal contract to develop a Crisis Management System. One of the component is supposed to monitor the epidemiological situation in the country. We have a various teams working on this project. Each team works with different data sources – some public, some proprietary. We would like *you* to propose a model, that can be used to analyze Twitter messages.

You will get a collection of (labeled) tweets describing or commenting the local Covid situation. We would like you to build a classifier, that can sort the messages into 5 categories: Extremely Negative (0), Negative (1), Neutral (2), Positive (3), Extremely Positive (4). When we have it, later, we will be able to build a system that measure how the perception of the epidemiological situation change in various regions – and this can be helpful in planning the next moves by various federal agencies.

I would like to see a preliminary report in two weeks. We will meet again on Thursday, April 1, at 10 am."

Hint: This task is to build an text classifier. You have to first tokenize all words. There are various libraries that offer Tokenizers, e.g., NLTK (Natural Language Toolkit). TensorFlow have also a tokenizer, that you can use. You can also test, if stemming (read about it) can help you.

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: "You have several choices how to approach this problem. You can use a Naive Bayes Classifier for start. Or you can train your own word embedding. If you feel like you can, you can also try recurrent models or you can use transfer learning techniques, by taking a pre-trained word embedding. You can also try to clean the text data by e.g., stemming the words."

Hint: You do not have to use transfer learning techniques. This is just an optional extension. The same about recurrent models. It is just an option for people who would like to do a little more. As always, you should validate your model choice – by for example, comparing the performance of some candidate-models on some validation tests.

The senior developer took you aside and said: "My task will be to maintain your code. Please, 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

Then, you will be able to find the dataset at `https://www.kaggle.com/c/usc-dsci552-section-32415d-spring-2021-ps5/data`. There are two data files. The first file, `ps5_tweets_text.csv`, contains labeld Tweet messages. For your convenience, I prepared the labels in two formats: text and numeric, see `ps5_tweets_labels.csv` and `ps5_tweets_labels_as_numbers.csv`.

The second data file `ps5_tweets_text_for_the_kaggle_competition.csv` contains Tweet messages for the (optional) 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, ..., 3797 (intigers). Second column is called `Predicted` and should contain your prediction if the message was Extremely Negative (0), Negative (1), Neutral (2), Positive (3), Extremely Positive (4).

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/CCIIYXnL`.

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 fifth 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 `https://www.tensorflow.org/tutorials` or optionally, the Chapter 16 from "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 2nd Edition by Aurélien Géron.

If something is not clear, ask questions on Piazza.