DSCI 552: Machine Learning for Data Science

Problem Set 6

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

Deadline: Thursday, April 15, 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 fifth task.

The last months were exhausting. You really need a vacation. You went to HR and scheduled some days off in May. You were about to return to your desk, when you noticed your CEO.

"Good to see you! I have another assignment for you. Come with me!", she called you. You entered the well familiar meeting room. Your technical manager and the developer were already there.

The CEO said: "You did a splendid work. We are all happy with your performance. We should soon have a conversation about your promotion. We will get a few new summer interns, you might take one or two and train them. And for this, you should have an upgraded title – what would you think about *Lead* Data Scientist? We can talk about the details of your promotion later, maybe end of the month.

In the meantime, we have another task for you. Los Angeles Department Of Water and Power is developing a new weather predictive system. They wish to predict the temperature for the next 24 to 120 hours.

You will get historical records from our city, about 5 years. Your task is to propose a model that is capable of predicting temperature based on the recent history. You should report how good your model is – and how long in the future you can predict with a reasonable accuracy (how many hours or how many days).

I would like to see a preliminary report in two weeks. We will meet again on Thursday, April 15, at 10 am."

Hint: This task is called a time series forecasting. As a help and inspiration, read "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 2nd Edition, Chapter 15 as well as the official TensorFlow tutorial, `https://www.tensorflow.org/tutorials/structured_data/time_series`*. You can also look for other materials (just always cite the sources!).*

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. Remember, it is always important to know limits of you model. In other words, if your model doesn't work well – but you can detect it, and maybe explain the reason why it fails – such a solution will not be treated as wrong or not-completed.

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: "It is up to you, how you will approach this problem. However, my suggestion is that you should start from constructing some baselines, that exploit the seasonality of the temperature (yearly and daily). Then, you can train some (more advance) models that predict either a single timestep or several timesteps at once. The most natural choice is to use recurrent neural networks, but you can also start from some simpler architectures (not recurrent). It can be useful to have some simple models first – you will be able to use the performance of those simpler models as your reference point – so when you train your recurrent neural network, you will be able to easily tell, if you make progress or not. By the way, you do not have to test all approaches that are mentioned in the literature – just pica a few that you believe are the most suitable and compare them.

I also noticed, that the data are quite rich. You have information not only about the temperature but also about humidity, pressure, general weather conditions, etc. Your baseline models do not have to use all those information.

Finally, if you can, try to measure and report how important those different features are. It could be useful to know how useful the information about the humidity or pressure, or wind are for your predictive task (predicting the temperature)."

Hint: There are different ways how to measure the feature importance. One (easy) method is to use the permutation importance technique, see `https://scikit-learn.org/stable/modules/permutation_importance.html` *You can also train two models, one with only temperature, another with some selected features – and you can compare the precision of your predictions. If you run out of the time, you don't have to* *measure all combinations – just some, so we know that you know how to do it.*

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 work with time series and you will be able to predict future events.

Data

First, join the Kaggle competition using the following link `https://www.kaggle.com/t/50bb2ace22ee4727b85ba866b1e8afb3`. Then, you will be able to find the dataset at `https://www.kaggle.com/c/usc-dsci552-section-32415d-spring-2021-ps6/data`. There are two data files. The first file, `ps6_trainvalid.csv`, contains labeled training and validation data. You can use the second file, `ps6_test.csv`, to test the final versions of your models (to make the testing procedure less confusing, this time you *do* have access to the test set. You can also prepare your own tests sets as well – if you wish to check how the model perform in different time of the year).

I removed last 120 records of temperature from the test set (look at the file, you will find question marks instead of the values). If you wish to take part in the optional Kaggle competition, you can try to predict those missing values.

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, ..., 119 (intigers). Second column is called `Predicted` and should contain the predicted temperature in a given hour (look at the `ps6_test.csv` to see the exact time). Note, that you should report the temperature in Kelvin.

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/Yu8XGhtx`.

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 sixth 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.