Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning

Instructor: Purushottam Kar
Date: 15 April, 2023
Total: 60 marks

ASSIGNMENT 3

1 What should I submit, where should I submit and by when?

Your submission for this assignment will be one PDF (.pdf) file and one ZIP (.zip) file. Instructions on how to prepare and submit these files is given below.

Assignment Package:
`https://www.cse.iitk.ac.in/users/purushot/courses/ml/2022-23-w/material/assn/assn3.zip`
Deadline for all submissions: 04 May, 2023, 9:59PM IST
Code Validation Script: `https://colab.research.google.com/drive/1XpVkd7c7vC2cytvonlLseA-DwzAf2OPP?usp=sharing`
Code Submission: `https://forms.gle/9WVHHCtQujjLfbTEA`
Report Submission: on Gradescope
There is no provision for "late submission" for this assignment

1.1 How to submit the PDF report file

1. The PDF file must be submitted using Gradescope in the *group submission mode*. Note that this means that auditors may not make submissions to this assignment.
2. Make only one submission per assignment group on Gradescope, not one submission per student. Gradescope allows you to submit in groups - please use this feature to make a group submission.
3. Ensure that you validate your submission files on Google Colab before making your submission (validation details below). Submissions that fail to work with our automatic judge since they were not validated will incur penalties.
4. Link all group members in your group submission. If you miss out on a group member while submitting, that member may end up getting a zero since Gradescope will think that person never submitted anything.
5. You may overwrite your group's submission (submitting again on Gradescope simply overwrites the old submission) as many times as you want before the deadline.
6. Do not submit Microsoft Word or text files. Prepare your report in PDF using the style file we have provided (instructions on formatting given later).

1.2 How to submit the code ZIP file

1. Your ZIP file should contain atleast one Python (.py) file called `submit.py`, your learnt model (in raw form or pickled form) and any supporting files e.g. non-standard libraries, additional code etc.

2. The main Python file sitting inside the ZIP file must be named "submit.py" and it must implement a method called `my_predict` that takes in a pandas dataframe containing test features and gives two sets of outputs (see below as well as the `dummy_submit.py` present in the assignment package).

3. Do not submit Jupyter notebooks or files in other languages such as C/Matlab/Java. We will use an automated judge to evaluate your code which will not run code in other formats or other languages (submissions in other languages may simply get a zero score).

4. Password protect your ZIP file using a password with 8-10 characters. Use only alphanumeric characters (a-z A-Z 0-9) in your password. Do not use special characters, punctuation marks, whitespaces etc in your password. Specify the file name properly in the Google form.

5. Remember, your file is not under attack from hackers with access to supercomputers. This is just an added security measure so that even if someone guesses your submission URL, they cannot see your code immediately. A length 10 alphanumeric password (that does not use dictionary phrases and is generated randomly e.g. 2x4kPh02V9) provides you with more than 55 bits of security. It would take more than 1 million years to go through all $> 2^{55}$ combinations at 1K combinations per second.

6. Make sure that the ZIP file does indeed unzip when used with that password (try
`unzip -P your-password file.zip`
on Linux platforms).

7. Upload the password protected ZIP file to your IITK (CC or CSE) website (for CC, log on to `webhome.cc.iitk.ac.in`, for CSE, log on to `turing.cse.iitk.ac.in`).

8. Fill in the Google form linked above to tell us the exact URL for the file as well as the password.

9. Make sure that when we visit the URL you have given, there is actually a file to be downloaded. Test your URL on a browser window to verify that you have not submitted a corrupted URL.

10. Do not host your ZIP submission file on file-sharing services like GitHub or Dropbox or Google drive. Host it on IITK servers only. We will autodownload your submissions and GitHub, Dropbox and Google Drive servers often send us an HTML page (instead of your submission) when we try to download your file. Thus, it is best to host your code submission file locally on IITK servers.

11. While filling in the form, you have to provide us with the password to your ZIP file in a designated area. Write just the password in that area. For example, do not write "Password: helloworld" in that area if your password is "helloworld". Instead, simply write "helloworld" (without the quotes) in that area. Remember that your password should contain only alphabets and numerals, no spaces, special or punctuation characters.

12. While filling the form, give the complete URL to the file, not just to the directory that contains that file. The URL should contain the filename as well.

    (a) Example of a proper URL:
    `https://web.cse.iitk.ac.in/users/purushot/mlassn1/my_submit.zip`

(b) Example of an improper URL (file name missing):
`https://web.cse.iitk.ac.in/users/purushot/mlassn1/`

(c) Example of an improper URL (incomplete path):
`https://web.cse.iitk.ac.in/users/purushot/`

13. We will use an automated script to download all your files. If your URL is malformed or incomplete, or if you have hosted the file outside IITK and it is difficult for us to download automatically.

14. Make sure you fill in the Google form with your file link before the deadline. We will close the form at the deadline.

15. Make sure that your ZIP file is actually available at that link at the time of the deadline. We will run a script to automatically download these files after the deadline is over. If your file is missing, we will treat this as a blank submission.

16. We will entertain no submissions over email, Piazza etc. All submissions must take place before the stipulated deadline over the Gradescope and the Google form. The PDF file must be submitted on Gradescope at or before the deadline and the ZIP file must be available at the link specified on the Google form at or before the deadline.

Problem 3.1 (Calibrate Away)**.** Monitoring air quality is of crucial importance for a country like India which is home to some of the most polluted cities in the world. India imports sensors required to measure levels of harmful pollutants like ozone O_3 and nitrogen dioxide NO_2 but these are usually manufactured in nations with distinct weather conditions like China or European countries so the sensors do not work well right out of the box in Indian conditions. To get them working, we need to perform a task called *calibration* that looks a lot like regression.

In this task, we will calibrate two sensors, one measuring the level of O_3 and another measuring the level of NO_2. Both these sensors are electrochemical in nature i.e. in response to changing levels of the pollutant they are measuring, they output two voltages called OP1 and OP2. More specifically, the O_3 sensor outputs voltages named o3op1, o3op2 whereas the NO_2 sensor outputs voltages named no2op1, no2op2.

The manufacturer of these sensors claims that these two voltages can give the true level of the pollutant using a simple linear model. However, these sensors are cross-sensitive in that the ozone sensor measures levels of not just ozone but also nitrogen dioxide. Thus, the manufacture suggests that we use all 4 voltage values o3op1, o3op2, no2op1, no2op2 along with a linear model to obtain the true value of both pollutants. Specifically, we wish to learn some real-valued constants $p_{o3}, q_{o3}, r_{o3}, s_{o3}, t_{o3}$ such that the true level of ozone is given by

$$p_{o3} \cdot \text{o3op1} + q_{o3} \cdot \text{o3op2} + r_{o3} \cdot \text{no2op1} + s_{o3} \cdot \text{no2op2} + t_{o3}$$

and for some other real-valued constants $p_{no2}, q_{no2}, r_{no2}, s_{no2}, t_{no2}$, we have the true level of nitrogen dioxide given by

$$p_{no2} \cdot \text{o3op1} + q_{no2} \cdot \text{o3op2} + r_{no2} \cdot \text{no2op1} + s_{no2} \cdot \text{no2op2} + t_{no2}$$

Your Data. We have provided you with training data in a CSV file that contains 9 columns:

1. Timestamp: this tells us the time of the day at which the measurement was taken. One measurement was taken per minute
2. OZONE: this tells us the true level of O_3 at that time stamp
3. NO2: this tells us the true level of NO_2 at that time stamp
4. temp: this tells us the temperature at that time stamp in degrees Celcius (between 0 and 100)
5. humidity: this tells us the relative humidity at that time stamp as a percentage (between 0% and 100%)
6. no2op1, no2op2, o3op1, o3op2: these tell us the voltages given by the two sensors at that time stamp

Your Task. There are three tasks you have to perform

1. Find out how well can you predict the O_3 and NO_2 using the method suggested by the manufacturer. To do this, learn the best linear model that uses just the 4 voltage values to predict O_3 and NO_2 values. Remember that for this part, you cannot use non-linear models, nor can you use temp, humidity, time stamp as features. However, you can use different loss functions e.g. least squares loss, absolute loss, ϵ-insensitive loss as well as different regularizers e.g. ridge, lasso etc. If you are trying out support vector regression for this part, remember to use the linear kernel. **Describe the method that gave**

you the best-performing linear model (in terms of MAE on training data) and write down what mean absolute error (MAE) does your model give on the training set. (10 marks)

2. Chances are that you may not get a very satisfactory result using just a linear model and just the voltage features. Thus, in this next part, develop a learning method that is free to **use temp, humidity, time stamp in addition to the voltage features** to predict the O_3 and NO_2 values. You are also free to use non-linear models e.g. decision trees, kernels, nearest-neighbors, deep-nets, etc. **Describe the method you found to work best giving all details of training strategy e.g. choice of loss function and tuning of hyperparameters.**
   Note that you may or may not find the time stamp as a useful feature since some of these pollutants are known to have a diurnal cycle e.g. Ozone is known to have high values during the daytime when sunlight is abundant and low values during night time due to darkness. (10 marks)

3. Use the training data to train your model on the expanded set of features and send us that model. Also write code that can take test features (timestamp, temp, humidity, no2op1, no2op2, o3op1, o3op2) and predict the value of O_3 and NO_2 using the model you have sent us. You are allowed to use all standard Python libraries e.g. numpy, sklearn, scipy, keras etc. **However, if you are using a non-standard library that is not available via pip and which is essential for prediction on test data, you must supply that library to us in your submission.** If a library is only needed for training and not for testing, then no need to submit that library. **Do not submit training code.** Submit prediction code for your chosen method in `submit.py`. Your code must implement a `my_predict()` method that takes a dataframe as input containing the test features and returns two numpy arrays, the first numpy array containing the predictions of O_3 for each test point and the other one containing NO_2 predictions on each test point. We will evaluate your submitted model on test data that is similar to the training data provided to you (see below for details). **Please go over the Google Colab validation code and the dummy submission file** `dummy_submit.py` to clarify any doubts about data formats, protocol etc. (40 marks)

Parts 1 and 2 need to be answered in the PDF file whereas Part 3 needs to be submitted as code + model. Remember, that part 1 can use only voltage values and linear models whereas in parts 2 and 3, you can use all features and non-linear models. Please submit only one model, the one that you feel is best. **Do not submit the linear model you found to work best in Part 1** (unless you could not find a non-linear model working better).

Evaluation Measures and Marking Scheme. We have secret test data that is similar to the training data and collected using the same two sensors that generated training data. We will evaluate your method on 3 criterion.

1. How fast is your `my_predict` method able to finish prediction (10 marks)
2. What is the on-disk size of your submission (after unzipping) (5 marks)
3. What MAE does your model offer for O_3 and NO_2 predictions on test data (10 + 15 marks)

Validation on Google Colab. Before making a submission, you must validate your submission on Google Colab using the script linked below.

Link: `https://colab.research.google.com/drive/1XpVkd7c7vC2cytvonlLseA-DwzAf2OPP?usp=sharing`

Validation ensures that your `submit.py` does work with the automatic judge and does not give errors due to file formats etc. Please use IPYNB file at the above link on Google Colab and the dummy secret dictionary (details below) to validate your submission.

Please make sure you do this validation on Google Colab itself. **Do not download the IPYNB file and execute it on your machine – instead, execute it on Google Colab itself.** This is because most errors we encounter are due to non-standard library versions etc on students personal machines. Thus, running the IPYNB file on your personal machine defeats the whole purpose of validation. You must ensure that your submission runs on Google Colab to detect any library conflict. **Please note that there will be penalties for submissions which were not validated on Google Colab and which subsequently give errors with our automated judge.**

Dummy Submission File and Dummy Test Data. In order to help you understand how we will evaluate your submission using the evaluation script, we have included a dummy test data, a dummy model and a dummy submission file in in the assignment package itself. However, note that the dummy test data is just a subset of the training data.

Using Internet Resources. You are allowed to refer to textbooks, internet sources, research papers to find out more about this problem. However, if you do use any such resource, cite it in your PDF file. There is no penalty for using external resources so long as they are acknowledged but claiming someone else's work (e.g. a book or a research paper) as one's own work without crediting the original author will attract penalties.

Restrictions on Code Usage. You are allowed to use all standard Python libraries e.g. numpy, sklearn, scipy, libsvm, keras, tensorflow as well as 3rd party libraries. However, if you use someone else's code, do give them proper credit by citing them prominently in your PDF file. There is no penalty for using external code so long as they are acknowledged. **However, if you are using a non-standard library that is not available via pip and which is essential for prediction on test data, you must supply that library to us in your submission.**

(60 marks)

2 How to Prepare the PDF File

Use the following style file to prepare your report.
`https://media.neurips.cc/Conferences/NeurIPS2022/Styles/neurips_2022.sty`

For an example file and instructions, please refer to the following files
`https://media.neurips.cc/Conferences/NeurIPS2022/Styles/neurips_2022.tex`
`https://media.neurips.cc/Conferences/NeurIPS2022/Styles/neurips_2022.pdf`

You must use the following command in the preamble

```
\usepackage[preprint]{neurips_2022}
```

instead of `\usepackage{neurips_2022}` as the example file currently uses. Use proper LaTeX commands to neatly typeset your responses to the various parts of the problem. Use neat math expressions to typeset your derivations. Remember that all parts of the question need to be answered in the PDF file. All plots must be generated electronically - no hand-drawn plots would be accepted. All plots must have axes titles and a legend indicating what the plotted quantities are. Insert the plot into the PDF file using proper LaTeX `\includegraphics` commands.

3 How to Prepare the Python File

The assignment package contains a skeleton file `submit.py` which you should fill in with the code of your prediction method. This method should first load the model file that you should have included in your submission, and then make predictions.

1. Before submitting your code, make sure you validate on Google Colab to confirm that there are no errors etc.

2. You do not have to submit the evaluation script to us – we already have it with us. We have given you access to the Google Colab evaluation script just to show you how we would be evaluating your code and to also allow you to validate your code.