TASK 1B

Regression (graded)

You are in the group Siuuu consisting of angarcia (angarcia@student.ethz.ch (mailto://angarcia@student.ethz.ch)).

1. READ THE TASK DESCRIPTION

2. SUBMIT SOLUTIONS

3. HAND IN FINAL SOLUTION

1. TASK DESCRIPTION

This task is about **linear regression**: given an input vector **x**, your goal is to predict a value **y** as a **linear** function of a set of feature transformations, $\phi(\mathbf{x})$.

DATA DESCRIPTION

Download handout (/static/task1b_ql4jfi6af0.zip)

In the handout for this project, you will find the the following files:

- **train.csv** - the training set
- **sample.csv** - a sample submission file in the correct format
- **template_solution.py** - a template file that will guide you through the implementation of the solution

Each line in train.csv represents one data instance by an id, its label y, and its features x1-5:

```
Id,y,x1,x2,x3,x4,x5
0,3.5796196352704994,0.019999999999999907,0.04999999999999993,-0.09000000000000008,-0.43000000000000005,-0.0800000000000000
...
```

FEATURES DESCRIPTION

You are required to use the following features (in the following order) to make your predictions:

- Linear

$$\phi_1(\mathbf{x}) = x_1,\ \phi_2(\mathbf{x}) = x_2,\ \phi_3(\mathbf{x}) = x_3,\ \phi_4(\mathbf{x}) = x_4,\ \phi_5(\mathbf{x}) = x_5,$$

- Quadratic

$$\phi_6(\mathbf{x}) = x_1^2,\ \phi_7(\mathbf{x}) = x_2^2,\ \phi_8(\mathbf{x}) = x_3^2,\ \phi_9(\mathbf{x}) = x_4^2,\ \phi_{10}(\mathbf{x}) = x_5^2,$$

- Exponential

$$\phi_{11}(\mathbf{x}) = e^{x_1},\ \phi_{12}(\mathbf{x}) = e^{x_2},\ \phi_{13}(\mathbf{x}) = e^{x_3},\ \phi_{14}(\mathbf{x}) = e^{x_4},\ \phi_{15}(\mathbf{x}) = e^{x_5}$$

- Cosine

$$\phi_{16}(\mathbf{x}) = \cos(x_1),\ \phi_{17}(\mathbf{x}) = \cos(x_2),\ \phi_{18}(\mathbf{x}) = \cos(x_3),\ \phi_{19}(\mathbf{x}) = \cos(x_4),\ \phi_{20}(\mathbf{x}) = \cos(x_5)$$

- Constant

$$\phi_{21}(\mathbf{x}) = 1$$

where we indicate the whole input vector with **x** and we use x_i to denote its i^{th} component.

Your predictions are calculated as a linear function of the features above according to the following formula:

$$\hat{y} = w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \cdots w_{21}\phi_{21}(\mathbf{x})$$

We provide a template solution file that suggests a structure for how you can solve the task, by filing in the TODOs in the skeleton code. It is not mandatory to use this solution template but it is recommended since it should make getting started on the task easier. You are also encouraged (but not required) to implement regression solutions from scratch, for a deeper understanding of the course material.

SUBMISSION FORMAT

You are required to submit the weights of your linear predictor in a .csv file.

The file should contain 21 lines containing a float each. The i-th line indicates the i-th weight of your linear predictor. For your convenience, we further provide a sample submission file:

```
1
2
...
```

Notice that, to compute your prediction on the test data, the raw features of the test data are transformed according to the transformations introduced in the previous section and their dot products with your submitted weight vector are computed. This means that the first entry of your weight vector is multiplied by $\phi_1(\mathbf{x})$, the second entry is multiplied by $\phi_2(\mathbf{x})$ and so on. As a consequence, it is important to submit the weight vector in the **correct order**.

Please keep in mind that, as a group, you have a limited number of submissions as stated on the submissions page.

EVALUATION

The evaluation metric for this task is the **Root Mean Squared Error** which is the square root of the mean/average of the square of all of the error.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where y_i are the ground truth labels and $\hat{y}_i$ estimations made by your regressor.
Your goal is to minimize the RMSE i.e. bring your estimation as close as possible to our ground truth value.

GRADING

In this task you will submit a weight vector. We compute the performance of the resulting predictor on a test set. The samples in the test set are split into a *public* set and a *private* set of the same size. The RMSE of the predictor resulting from your weight vector will be computed on each split of the test set to compute a *public score* and *private score*. You only receive feedback about your performance on the public part in the form of the public score, while the private leaderboard remains secret. The purpose of this division is to prevent overfitting to the public score. Your model should generalize well to the private part of the test set. When handing in the task, you need to select which of your submissions will get graded and provide a short description of your approach. This has to be done **individually by each member** of the team. We will then compare your selected submission to our baselines.This project task is graded with grades between **2.0 - 6.0**. Your grade is calculated using a weighted sum of your private and public score. We do not share the weights used for the grading, again to prevent overfitting to the public score. The weights are selected such that:

- To achieve the best grade (6.0), you need to perform better than the hard baseline in both private and public score.
- To pass the project (grade: 4.0), you need to perform better than the easy baseline in both private and public score.

The medium baseline is only used for your reference and is not considered in the grading. In addition, for the grading, we consider the code and the description of your solution that you submitted. The following **non-binding** guidance provides you with an idea on what is expected to pass the project: If you hand in a properly-written description, your source code is runnable and reproduces your weight vector, and your submission performs better than the baselines, you can expect to have passed the assignment.

⚠ Make sure that you properly hand in the task, otherwise you may obtain zero points for this task.

PLAGIARISM

The use of open-source libraries is allowed and encouraged. However, we do not allow copying the work of other groups / students outside the group (including work produced by students in previous versions of this course). Publishing project solutions online is not allowed and use of solutions from previous years in any capacity is considered plagiarism. Among the code and the reports, including those of previous years, we search for similar solutions / reports in order to detect plagiarism. Use of GPT3 Copilot or similar code/language generation tools in any capacity for writing code or reports will be considered and treated as plagiarism in the context of this course. Basic code autocompletion such as those used in the default setup of Sublime Text 3 are permitted. If we find strong evidence for plagiarism, we reserve the right to let the respective students or the entire group fail in the IML 2023 course and take further disciplinary actions. By submitting the solution, you agree to abide by the plagirism guidelines of IML 2023.

FREQUENTLY ASKED QUESTIONS

⊙ WHICH PROGRAMMING LANGUAGE AM I SUPPOSED TO USE? WHAT TOOLS AM I ALLOWED TO USE?

You are free to choose any programming language and use any software library. However, **we strongly encourage you to use Python**. You can use publicly available code, but you should specify the source as a comment in your code.

⦿ WHAT TO DO IF I CAN'T RUN THE CODE/SETUP AN ENVIRONMENT ON MY PC?

If you are having trouble running your solution locally, consider using the ETH Euler cluster to run your solution. Please follow the Euler guide (/static/euler-guide.md). The setup time of using the cluster means that this option is only worth doing if you really can't run your solution locally.

⦿ AM I ALLOWED TO USE MODELS THAT WERE NOT TAUGHT IN THE CLASS?

Yes. Nevertheless, the baselines were designed to be solvable based on the material taught in the class up to the second week of each task.

⦿ IN WHAT FORMAT SHOULD I SUBMIT THE CODE?

You can submit it as a single file (main.py, etc.; you can compress multiple files into a .zip) having max. size of 1 MB. If you submit a zip, please make sure to name your main file as *main.py* (possibly with other extension corresponding to your chosen programming language).

⦿ IN WHAT FORMAT SHOULD I SUBMIT THE REPORT?

The handin page of the submission server contains a simple textbox in which you should insert your report. It should consist of a couple of sentences explaining the main ideas and concepts of your solution. Every student writes and submits the report independently.

⦿ WILL YOU CHECK / RUN MY CODE?

We will check your code and compare it with other submissions. We also reserve the right to run your code. Please make sure that your code is runnable and your predictions are reproducible (fix the random seeds, etc.). Provide a readme if necessary (e.g., for installing additional libraries).

⦿ SHOULD I INCLUDE THE DATA IN THE SUBMISSION?

No. You can assume the data will be available under the path that you specify in your code.

⦿ CAN YOU HELP ME SOLVE THE TASK? CAN YOU GIVE ME A HINT?

As the tasks are a graded part of the class, **we cannot help you solve them**. However, feel free to ask general questions about the course material during or after the exercise sessions.

⦿ CAN YOU GIVE ME A DEADLINE EXTENSION?

⚠ We do not grant any deadline extensions!

⦿ CAN I POST ON MOODLE AS SOON AS HAVE A QUESTION?

This is highly discouraged. Remember that collaboration with other teams is prohibited. Instead,

- Read the details of the task thoroughly.
- Review the frequently asked questions.
- If there is another team that solved the task, spend more time thinking.
- Discuss it with your team-mates.

⦿ WHEN WILL I RECEIVE THE PRIVATE SCORES? AND THE PROJECT GRADES?

We will publish the private scores, and corresponding grades before the exam the latest.