TASK 1B

Classification (graded)

You are in the group Frieren consisting of ddoebel (ddoebel@student.ethz.ch (mailto://ddoebel@student.ethz.ch)), mkryukov (mkryukov@student.ethz.ch (mailto://mkryukov@student.ethz.ch)) and tsciolla (tsciolla@student.ethz.ch (mailto://tsciolla@student.ethz.ch)).

1. READ THE TASK DESCRIPTION | 2. SUBMIT SOLUTIONS | 3. HAND IN FINAL SOLUTION

1. TASK DESCRIPTION

This task is about **logistic regression**: given an input vector **x**, your goal is to predict the value of a binary random variable **y** where the logits of y can modelled as a **linear** function of a set of feature transformations, $\phi(\mathbf{x})$. In other words, the labels Y given X can be modelled using a logistic regression mo where the inputs are a feature-transformed 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
- **template_solution.ipynb** - a template file in jupyter notebook format that will guide you through the implementation of the solution

You are free to use either jupyter notebook or the .py template file.

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
0,1,0.019999999999999907,0.04999999999999993,-0.09000000000000008,-0.43000000000000005,-0.08000000000000007
...
```

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 model the logits of y as a linear function of the features above according to the following formula:

$$\hat{P}(y = 1 \mid \mathbf{x}) = \sigma\left(w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \cdots w_{21}\phi_{21}(\mathbf{x})\right)$$

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 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) implement logistic 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 provi 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 the previous section and their dot products with your submitted weight vector are computed, before taking the Sigmoid of the scalar result to produce probability. 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 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 **F1 Score**, which is the harmonic mean of precision and recall. This metric balances the model's ability to correc identify whether a positive instance is positive (Precision) and its ability to classify all positive instances as positive (Recall).

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Where:

- $\mathrm{Precision} = \frac{TP}{TP+FP}$
- $\mathrm{Recall} = \frac{TP}{TP+FN}$

We abbreviated True Positives (TP), False Positives (FP) and False Negatives (FN) in the formulas above.

To compute these, your continuous probability estimates $\hat{y}_i = \hat{P}(y = 1 \mid \mathbf{x}_i)$ are converted by us into hard labels (0 or 1) using a threshold of **0.5**. Your go is to maximize the F1 score, i.e., achieve the best balance between precision and recall.

GRADING

In this task you will submit a weight vector. We compute the performance of the resulting predictor on a test set. When handing in the task, you need to sele which of your submissions will get graded and provide a short video description of your approach. This has to be done **individually by each member** of t team. You achieve a **pass (6.0)** if you achieve a better score than the single baseline, and a **fail (2.0)** otherwise. For the pass/fail decision, we also consid the code and the video submission explaining your solution. The following **non-binding** guidance provides you with an idea on what is expected to pass t project: If you hand in a proper video submission, your source code is runnable and reproduces your submitted csv, and your submission performs bet than the baseline, 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 gro (including work produced by students in previous versions of this course). Publishing project solutions online is not allowed and use of solutions fro previous years in any capacity is considered plagiarism. Among the code and the reports, including those of previous years, we search for similar solution reports in order to detect plagiarism. Although not strictly forbidden, we discourage the use of Github Copilot or similar code/language generation tools writing code. We expect that if such tools are used, this is clearly stated in the video submission explaining the solution. While it will have no effect on yo grade or if a solution passes or fails, it may affect the awarding of prizes for best solutions. We discourage these tools because we feel that the best way understand the material is to write the code yourself referring to just the lecture material, source papers and documentation of any libraries used. For t purposes of disclosing what generative AI tools you used to write code, we don't need you to disclose using e.g. basic code autocompletion such as tho used in the default setup of Sublime Text 3. If we find strong evidence for plagiarism, we reserve the right to let the respective students or the entire group f in the IML 2025 course and take further disciplinary actions. By submitting the solution, you agree to abide by the plagiarism guidelines of IML 2025.

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 publi 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/eule 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 baseline was 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 name your main file as *main.py* (possibly with other extension corresponding to your chosen programming language).

⦿ 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 a 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 (pass/fail) part of the class, **we cannot help you solve them**. However, feel free to ask general questions about the course mater 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 I 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 all grades before the exam the latest.