Learning challenge

For this assignment you will participate in groups of 4 students in a learning challenge. You will need to submit your predictions, as well as a report describing your solutions and the code which you used to generate the predictions.

This assignment is worth 30% of your course grade.

The assignment grade will be based on the quality of your work as judged by the instructor based on your report and code. Additionally, you will get a bonus based on your ranking on the leaderboard of the shared task.

Specifically:

- if your rank first, you will receive 2 bonus points;
- if your score is no better than the provided baseline your will receive no bonus;
- for intermediate ranks the bonus points will be linearly interpolated.

The performance of the baseline solution is shown on Codalab (submitted by account `gchrupal`.)

Report

Your report should be **2 page maximum**, in **PDF format** and should include the following:

Page 1

- Description of your computational learning experiments, including:
  - feature engineering
  - learning algorithm(s)
  - hyperparameter tuning
  - discussion of the performance of your solution

Page 2

- The name of the account under which you submitted your results to the competition on Codalab (see below)
- Detailed specification of the work done by group members
- References or appendices (if applicable)

Code

Your code should be a **plain Python script** (`.py`, not a notebook) which can be run to generate your predictions. You should not include any data files.

Codalab submission

You will need to submit your prediction file to the competition server. The team leader will need to get a codalab account (using a `tiburguniversity.edu` email), and will be responsible for submitting your solution. Indicate the name of this account in your report. For further details about the format of the prediction file, see section **Submission to Codalab**.

IN SUMMARY: submission consists of the following items:

1. Report (`.pdf`, Canvas)
2. Code (`.py`, Canvas)
3. Prediction file (`.zip`, Codalab)

Group work

Your report needs to contain a detailed description of who did what, so make sure to keep track of this information.

Note: it is **not acceptable** to just say *All members worked together and contributed equally.*

If there are any problems with collaboration, such as serious disagreements, a group member not contributing, or a group dissolving, make sure inform the course coordinator as soon as possible via email.

Code reuse rules

Remember this assignment is group work. You are **not allowed** to collaborate or share code with students outside your group.

Submissions will be checked for plagiarism.

If you are found breaking the above rules you will be reported to the Board of Examiners for fraud.

You **are allowed** to use:

- code examples provided by the instructor during the course, or as part of the competition;
- open source libraries available for Python;
- code found on Github, Stackoverflow or similar websites, as long as it is credited in your script with a link to the source.

Dataset

Citation count prediction

In this challenge the task is to predict the number of citations a scientific paper receives based on its abstract and metadata.

Data files

The dataset is available for download on Canvas. It contains the following files:

- `train.json`: the metadata and the citation counts of all the papers in the training data.
- `test.json`: the metadata, excluding the citation counts, of the papers in the test data.

Both of these files are in the `JSON` format. You can load them using the `json.load` function. The loaded data will consist of a list of Python dictionaries, with each dictionary corresponding to the record for one paper. The training records specify the number of citations the paper received to date under the key `citations`. For the test data this information is missing, and your task is to predict it. The other keys have descriptive names indicating the nature of the information: e.g. `title`, `abstract`, `authors`, `venue` (where the paper was published), `year` (date of publication), etc.

Evaluation metric

The evaluation metric for this task is the R^2 score computed on the log-transformed citation counts. Specifically, the following Python code is used for evaluation of the the predictions:

```
import numpy as np
import json

def score(Y_true, Y_pred):
    y_true = np.log1p(np.maximum(0, Y_true))
    y_pred = np.log1p(np.maximum(0, Y_pred))
    return 1 - np.mean((y_true-y_pred)**2) / np.mean((y_true-np.mean(y_true))**2)

def evaluate(gold_path, pred_path):
    gold = { x['doi']: x['citations'] for x in json.load(open(gold_path)) }
    pred = { x['doi']: x['citations'] for x in json.load(open(pred_path)) }

    y_true = np.array([ gold[key] for key in gold ])
    y_pred = np.array([ pred[key] for key in gold ])

    return score(y_true, y_pred)
```

Note that your predictions will be log-transformed within this function, so they need to be submitted as regular counts. If you decide to train your model on log-transformed labels, you will need to convert them back to counts before submitting. You may find the functions `numpy.log1p` and `numpy.expm1` useful for these conversions.

Method

You are free to use any learning algorithm or combination thereof, and any features you wish, subject to the following constraints:

- the method should be fully automatic, that is, by re-running your code it should be possible to re-create your prediction file;
- every software component used should be open-source and possible to install locally;
- while you can use external datasets in addition to the data provided, you are not allowed to use external information on citation counts or related metrics for the papers in the test data. If you wish to make use of external data in your solution, ask the instructor via the course forum to confirm that this data is allowed.

Some hints:

- Use part of the provided training data as a validation set.
- Only submit to Codalab after validating your results on this validation data.
- In the simplest case, you can rely solely on the provided files to extract all your features. However, it may be possible to improve performance by collecting additional information about the papers, such as for example the content of the paper or the affiliation of the authors.

Submission format

The submission format is a file named `predicted.json` with a list of dictionaries, each dictionary containing two keys: `doi` (the identifier of a paper) and `citations` (the predicted citation count). They dictionary may contain additional keys which will be ignored. You must supply a prediction for each paper in the test data: missing predictions will cause an error. The following is the beginning of an example prediction file:

```
[
  {
    "doi": "10.18653/v1/2021.findings-acl.255",
    "citations": 3
  },
  {
    "doi": "10.18653/v1/2020.acl-main.200",
    "citations": 7
  },
  {
    "doi": "10.18653/v1/W18-0211",
    "citations": 2
  },
```

The competition is hosted on Codalab at the following URL: https://bit.ly/2ZdDRXQ

You can submit your results in the **Participate** link.

Over the course of the competition you can make 7 submissions.

Note that if you submission fails for some reason such as incorrect format, this is still counted as one of the 7 submissions.

The results from all the participating teams will be displayed in the **Results** tab.

The submission file should be a `.zip` file with a file named `predicted.json` in it. (Make sure there are not additional subdirectories in the zip file.) Your file needs to use a valid JSON format.