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.

The assignment grade will be based on the quality of your work as judged by the instructor, based on your report and code.

The quality of the report and the code will be evaluated according to the quality of the solution and the report (see assignment rubric for details).

You can think of the report as a mini research paper: the evaluation criteria will be similar.

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.)

Note that your best scoring submission should be **submitted to and visible** on the Codalab leaderboard as soon as the submission is scored. In other words, you are not allowed to try and keep your best score secret. If you fail to follow this rule you will not receive any bonus based on your score.

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. If your use multiple modules in separate Python files, you can submit them as a zip file.

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)

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

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.

**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.

Dataset

Prediction of year of publication

In this challenge the task is to predict the year of publication of scientific papers, based on their metadata.

Data files

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

* [train.json](https://tilburguniversity.instructure.com/courses/15212/files/2752179?wrap=1)[Download train.json](https://tilburguniversity.instructure.com/courses/15212/files/2752179/download?download_frd=1): the metadata including the year of publication of all the papers in the training data.
* [test.json](https://tilburguniversity.instructure.com/courses/15212/files/2752181?wrap=1)[Download test.json](https://tilburguniversity.instructure.com/courses/15212/files/2752181/download?download_frd=1): the metadata, excluding the year of publication, 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 year of publication under the key year. 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.

Evaluation metric

The evaluation metric for this task is Mean Absolute Error. Specifically, the following Python code is used for evaluation of the predictions:

import json

from sklearn.metrics import mean\_absolute\_error

import numpy as np

import json

def evaluate(gold\_path, pred\_path):

gold = np.array([ x['year'] for x in json.load(open(gold\_path)) ]).astype(float)

pred = np.array([ x['year'] for x in json.load(open(pred\_path)) ]).astype(float)

return mean\_absolute\_error(gold, pred)

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;
* you are not allowed to access or use external paper metadata, or the full text of the papers in the dataset;
* it is allowed to use generic text datasets or pretrained text feature extraction models (such as topic models, word vectors or sentence embeddings). If case of doubt, ask via the course forum to confirm a dataset or model you want to use 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.
* Use the provided starter code ([baseline.py](https://tilburguniversity.instructure.com/courses/15212/files/2752185?wrap=1)[Download baseline.py](https://tilburguniversity.instructure.com/courses/15212/files/2752185/download?download_frd=1)) as a starting point for your work.

Submission format

The submission format is a file named predicted.json with a list of dictionaries, each dictionary containing at least the key year and the year value as a string, int or float. They dictionary may contain additional keys which will be ignored. You must supply a prediction for each paper in the test data, in the same order as in the file test.json. Missing predictions will cause an error. The following is the beginning of an example prediction file:

[

{

year: 1974

},

{

year: 2022

},

The competition is hosted on Codalab at the following URL: [https://bit.ly/3SrsLYjLinks to an external site.](https://bit.ly/3SrsLYj)

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

Over the course of the competition you can make 4 submissions.

*Note that if you submission fails for some reason such as incorrect format, this is still counted as one of the 4 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.