INF 554 - Introduction to Machine and Deep Learning
Data Challenge 2020

COVID19 Retweet Prediction

1. Introduction

In this challenge, you will work in *teams of at most three students*. For this challenge, **your mission is to accurately predict the number of retweets a tweet will get**. The provided dataset contains tweet related information, such as the text and the number of hashtags, mentions and URLs contained in the tweet, and user related information, such as the number of followers and tweets that the user has published. Your solution can be based on supervised or unsupervised techniques or on a combination of both. You should aim for a minimum **Mean Absolute Error (MAE),** i.e., the MAE will be the loss function we use to evaluate your models.

This data challenge is hosted on Kaggle as an in-class competition. In order to access the competition you must have a Kaggle account. If you do not have an account you can create one for free. The URL to register for the competition and have access to all necessary material is the following:

https://www.kaggle.com/t/9661636150ea4f768178b34ba0693478

2. File Description

train.csv - 665.777 tweets for which we know the number of retweets. Each row has the following fields, 'id', 'timestamp', 'retweet_count', 'user_verified', 'user_statuses_count', 'user_followers_count', 'user_friends_count', 'user_mentions', 'urls', 'hashtags' and 'text'.

evaluation.csv - 285.334 tweets. The number of retweets is not available (your task is to predict it). Each row has the following fields, an 'id', 'timestamp', 'user_verified', 'user_statuses_count', 'user_followers_count', 'user_friends_count', 'user_mentions', 'urls', 'hashtags' and 'text'.

mean_predictions.csv, constant_predictions.csv - sample submission files in the correct format (the predictions have been generated by dummy baselines). Please ensure that your submissions follow the format of these files.

dummy_baseline.py - a Python script with two baselines, one that given each row predicts the

number of retweets is equal to the mean value of the training dataset and one that constantly predicts zero. The MAE is 284.57 and 161.03 respectively.

baseline.py - a Python script containing a baseline method that only uses the text of the tweet. The method transforms the text using the TF-IDF vectors and then trains a Gradient Boosting Regressor. The MAE score of this baseline is approximately 275,69 .

3. Evaluation Metric

For each tweet in the test set, your model should predict the number of retweets it will get after its publication. The evaluation metric for this competition is Mean Absolute Error (MAE). The MAE metric is calculated by dividing the sum of absolute differences of the predicted number of retweets (p_i) and the observed number of retweets (a_i) by the number of observations (N), i.e.,

$$MAE = \frac{1}{N} \sum_{t=1}^{N} |p_i - a_i|$$

4. Grading Scheme

Grading will be out of 100 points in total. Each team should deliver:

A submission on the Kaggle competition webpage. (20 points) will be allocated based on raw performance only, provided that the results are reproducible. That is, using only your code, the data provided on the competition page and any additional resources you are able to reference and demonstrate understanding of, the jury should be able to train your final model and use it to generate the predictions you submitted for scoring.

A zipped folder in moodle (30 points) including:
1) A folder named "code" containing all the scripts needed to reproduce your submission. (Please do not submit any data.)
2) A README file with brief instructions on how to run your code and where it expects the original data files.
3) A report (.pdf file), of max 3 pages, excluding the cover page and references. In addition to your self-contained 3-page report, you can use up to 3 extra pages of appendix (for extra explanations, algorithms, figures, tables, etc.). Please ensure that both your real name(s) and the name of your Kaggle team appear on the cover page.

The 3-page report should include the following sections (in that order):

- **Section 1: feature selection/extraction (15 points).** Independent of the prediction performance achieved, the jury will reward the research effort done here. Best submissions will capture both tweet features, user and text information. You are expected to:
  1) Explain the motivation and intuition behind each feature. How did you come up with the feature (e.g., are you following the recommendation of a research paper)? What is it intended to capture?
  2) Rigorously report your experiments about the impact of various combinations of features on predictive performance, and, depending on the regressor, how you tackled the task of feature selection.

- **Section 2: model tuning and comparison (10 points).** Best submissions will:
  1) Compare your model against different regression models (e.g., Support Vector Regression, Random Forest, Boosting...).
  2) For each regressor, explain the procedure that was followed to tackle parameter tuning and prevent overfitting.

Report and code completeness, organization and readability will be worth **5 points**. Best submissions will (1) clearly deliver the solution, providing detailed explanations of each step, (2) provide clear, well organized and commented code, (3) refer to research papers.

You are free to search for relevant papers and articles and try to incorporate their ideas and approaches into your code and report as long as (a) it is clearly cited within the report, (b) it is not a direct copy of code and (c) you are able to demonstrate understanding of the content you incorporated.

An oral presentation of your project and the achieved results (50 points)

Oral presentations will be scheduled in the week of the 14th December. Please submit, (1) your team member names and (2) the exam and lecture dates and hours of your team members during the week of the 14th December, to us via direct message on slack by **Monday 23rd November**, so that we can schedule the presentation of your group. More details on the presentations will follow once they are scheduled.

Finally, note that the test set has been randomly partitioned into public and private. Scores on the leaderboard are based on the public set, but final scores (based on which grading will be performed) will be computed on the private set. This removes any incentive for overfitting the test set.

5. Submission Process

Submission files should be in **.csv format**, and contain two columns respectively named "TweetID" and "NoRetweet". The "TweetID" column should contain the tweet id as given in the evaluation.csv. The “NoRetweet" column should contain the predictions (no negative integer). Note that a sample submission file is available for download **(mean_predictions.csv)**. You can use it to test that everything works fine.

The competition ends on **Thursday 10th December at 23:59**. This is the deadline for you to submit a compressed file containing your source code and final report, explaining your solutions and discussing the scores you have achieved. *Until then, you can submit your solution to Kaggle and get a score at most 5 times per day.*

There must be one final submission per team.

Also, do not forget to include the name of your team and all team members’ real names **on the cover page of the report .**