Recencia Papers

1. **Model Explanation**

Background

1. Name: Felipe Nunez
2. City/Country: Villa Alemana, Chile
3. Email: f.nunezb@gmail.com
4. What studies have you done that are related to data science and machine learning?
    a. Msc. Data Science
5. What relevant experience do you have in data science and/or machine learning?
    a. University and 6 months as Senior DS in Evalueserve Chile.
6. Do you have experience related to the problem of this competition?
    a. NLP at university and company.
7. How much time have you invested in the competition?
    a. Around 40 hours.

Model:

- My approach uses transformers to encode the Abstract text into numerical features and then Catboost to generate the final predictions, based on the transformer embedding.

1. Split the input data into 3 subsets: one for each language. Cleaned the abstracts (removed additional languages, for example) and then generated embeddings, using the Huggingface Transformers library (a different model for each language).
2. Concatenate the embeddings with the year of publication. Use this final vector as input for the Catboost regression model.
3. Given the lack of validation set, split the data using kFolds: trained 5 models for each language, using an 80/20 subsplit. I used bayesian optimization to find the best hyperparameters. Ended up with 15 model weight files (5 for each language).
4. For the test set, generate embeddings and using the trained model weights, generate the prediction by taking the average prediction for each one of the models (e.g. for a portuguese text, predict with each one of the 5 portuguese models and output the average of these 5 models).

What takes more time is the embedding generation (Transformers). With GPU activated on a Google Colab notebook instance, it took ~11 minutes to generate all embeddings for the training set and 5 additional minutes to train the Catboost models.

For the predictions of the competition test set, the embedding generation took ~5 minutes, with 20 additional for predictions.

For the final test set, the prediction time was nearly the same as for the competition test set.

To be fair, my approach was pretty simple and therefore I can't talk much about what helped with good results.