# III. Model documentation and write-up

Information included in this section may be shared publicly with challenge results. You can respond to these questions in an e-mail or as an attached file. Please number your responses.

1. Who are you (mini-bio) and what do you do professionally? If you are on a team, please complete this block for each member of the team.

I am doing research at the University of Montreal in the field of computer graphics. Before I received my master's degree in mathematics from Novosibirsk State University in Russia. Recently, I became interested in machine learning, so I was enrolled in the Yandex School of Data Analysis and Computer Science Center. Machine learning is my passion and I often participate in competitions.

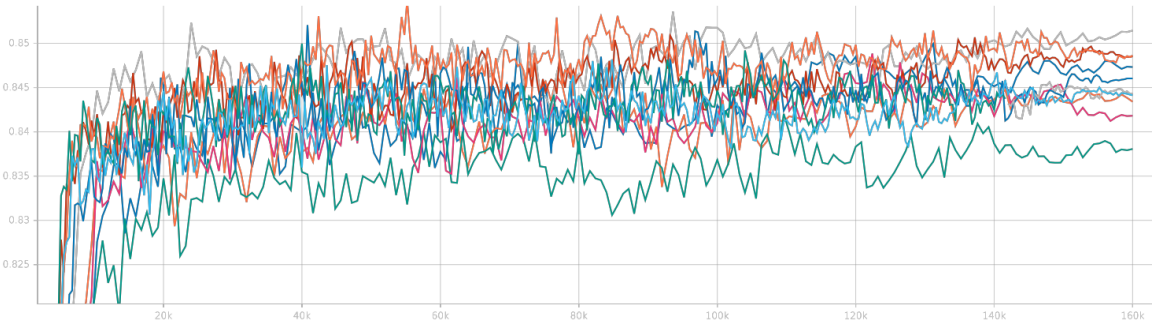
2. What motivated you to compete in this challenge?

I love competitions and solving new problems. This competition allowed me to get my first touch and play with LLMs, although they don't perform any better than regular logistic regression for this task.

3. High level summary of your approach: what did you do and why?

We addressed a multilabel text classification problem for long documents using BigBird and Longformer models. Pretraining BigBird as a masked language model with a 512 block size proved effective, while attempts to pretrain Longformer led to NaN values. Fine-tuning employed the symmetric Lovász loss, with binary cross-entropy and focal losses yielding comparable performance. We trained with a learning rate of 1e-5, a linear scheduler with 2 warmup epochs, batch sizes of 4 (BigBird) and 2 (Longformer) per GPU, and for 200 and 400 epochs, respectively. Optimal thresholds were determined for each binary variable, and a 5-fold ensemble improved generalization.

4. Do you have any useful charts, graphs, or visualizations from the process?
   Typical learning curves of the model (CDC F1 on 5 folds)




5. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.
   1. Custom trainer

```python
class CustomTrainer(Trainer):
    def compute_loss(self, model, inputs, num_items_in_batch=None, return_outputs=False):
        labels = inputs.pop("labels")
        outputs = model(**inputs)
        logits = outputs.logits
```

```
        logits_bin, logits_1, logits_2 = logits.split([len(BIN_COLS), N_CAT_INJ,
N_CAT_WEAP], dim=1)
        labels_bin, labels_1, labels_2 = labels.split([len(BIN_COLS), 1, 1], dim=1)
        labels_bin = labels_bin.float()
        loss_bin = (
            #binary_cross_entropy_with_logits(logits_bin, labels_bin)
            #sigmoid_focal_loss(logits_bin, labels_bin)
            symmetric_lovasz(logits_bin, labels_bin)
        )
        loss_1 = cross_entropy(logits_1, labels_1.squeeze(1))
        loss_2 = cross_entropy(logits_2, labels_2.squeeze(1))
        loss = loss_bin + loss_1 + loss_2

        return (loss, outputs) if return_outputs else loss
```

2. Best threshold

```python
def threshold_search(y_train, y_train_hat):
    thresholds = np.linspace(0.0, 1.0, 100 + 1)
    scores = []
    for thresh in thresholds:
        score = get_score(y_train, y_train_hat, thresh)
        scores.append(score)

    thresh_ind = np.argmax(scores)
    thresh = thresholds[thresh_ind].item()
    score = scores[thresh_ind]

    return thresh, score
```

3. Model ensembling

```python
# predictions.shape == (n_models, N, C)
# threshes.shape == (n_models, C)

threshes = np.mean(threshes, axis=0)       # (C, )
predictions = np.mean(predictions, axis=0)  # (N, C)
predictions = predictions > threshes        # (N, C)
```

6. Please provide the machine specs and time you used to run your model.
   - CPU (model): Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz
   - GPU (model or N/A): Nvidia Tesla V100 32GB
   - Memory (GB): 16Gb
   - OS: GNU/Linux Ubuntu 22
   - Train duration: 10 days on 2 GPUs V100 32Gb
   - Inference duration: ~60 min

7. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?

No

8. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?
   No
9. How did you evaluate performance of the model other than the provided metric, if at all?
   No
10. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?

First I tried logistic regression and LightGBM on Tf-Idf. Then I tried different LLM prompt engineering that perform worse than logistic regression.

11. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?
    Use new variables from the adjacent track of this competition.
12. What simplifications could be made to run your solution faster without sacrificing significant accuracy?
    Use one model instead of an ensemble. Another option is to apply knowledge distillation.