

## 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 a machine learning enthusiast without formal education in machine learning or computer science. I work as an early childhood educator professionally.

2. What motivated you to compete in this challenge?

The simple data structure and interesting problem setup.

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

100% CNN models since I have been learning deep learning recently.

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

I am not sure if I can do that but the most impactful part of my solution is mainly inspired by this paper: <https://arxiv.org/abs/1408.5882>.

5. Please provide the machine specs and time you used to run your model.

- CPU (model): i9 9900K
- GPU (model or N/A): Nvidia 2080ti
- Memory (GB): 128GB
- OS: Linux Ubuntu 18
- Train duration: 14 days
- Inference duration: 1 hour

6. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?

Tried different Kmers but found that 8mers worked the best.

7. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?

No.

8. How did you evaluate performance of the model other than the provided metric, if at all?

The models were solely evaluated with the provided metric.

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

Nothing other than what is specified in readme.

10. Do you have any useful charts, graphs, or visualizations from the process?

No.

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?

Transformers or their variants might be something that I will try if I continue working on this problem.