Requirements for documenting Kaggle submissions

For the NSS DS6 Kaggle competition, you and your teammates will be testing out different models to predict the winning team in a League of Legends match. Documenting how you prepare your data and train / tune / validate your models enables your team to stay in sync and iterate quickly. The guidelines below will set you on the path to a smooth collaborative process throughout the competition.

1. Version your submissions

- For each submission, give the CSV file you upload to Kaggle a unique name and version number
- e.g. “submission_YYYY-MM-DD_v01.csv”
- Why: this will enable you and your teammates to refer back to high-performing submissions later in the project (particularly useful for ensemble modeling, but also for general iteration)

2. Connect your GitHub commits and Kaggle submissions

- For each submission, there must be a unique GitHub commit hash with the exact version of the notebook that generated the predictions
- e.g. a sample commit message: “notebook for submission_YYYY-MM-DD_v01.csv, logistic regression using all predictor variables, macro f1 score 0.8” (see notes on documentation below)
- Why: adding the submission filename into the GitHub commit message will enable you and your teammates to quickly find and refer back to your code for specific models throughout the project
- NB: do not push data up to GitHub (add your submissions folder to your .gitignore file!); if you need to swap CSVs, use a file sync / share platform for your team

3. Document how you generate each submission

- In your team project planning document, add a new section titled “Submissions”
- Use the following outline to keep track of metadata for each submission:

    1. ***Submission ID***
        - e.g. “submission_YYYY-MM-DD_v01.csv GH commit XXXXX”
        - Why: you need a way to connect the predictions you submitted to Kaggle, the notebook version you pushed to GitHub, and the documentation you are writing in your team project doc
    2. ***Model Type***
        - e.g. “Logistic Regression”, “Random Forest”
        - Why: you will be testing out many different models throughout the process and will want to compare how they perform

3. ***Variables***
    - Why: the variables you choose (and the logic for how you make those choices) may inform variable selection for future models / iterations

4. ***Data Preprocessing***
    - e.g. “log transformation of x, y, and z variables”
    - Why: how you preprocess your data will directly impact the performance of your model

5. ***Hyperparameter Settings***
    - e.g. “max_depth = 15”, “n_estimators = 80”
    - Why: knowing what hyperparameter settings work well / less well can give you a head start in iterating over your model

6. ***Model Validation***
    - e.g. “5-fold cross-validation”, “60/20/20 train/test/validation split”
    - Why: proper model validation practices ensure you are not overfitting or underfitting the training data, and different techniques may result in different predictions

7. ***Evaluation metric(s) and scores***
    - e.g. “macro f1 score = 0.8”
    - Why: keeping track of what performs well / less well as you go is a good way to measure your progress

8. ***Additional notes***
    - e.g. “v6 model modifies v2 by utilizing k-fold cross-validation instead of train/test split, moved kaggle public leaderboard macro f1 score from 0.7 to 0.75”
    - Why: sometimes, there will be a distinguishing feature to call out, a relationship to another submission that might be useful to learn from, or something that you discovered in the process of building the model / submission that you will want to record, so make a note to share with your team

4. Curate your GitHub Repo

- At the end of the competition, you and your team will curate your competition GitHub repo. In addition to merging down and pruning any branches and cleaning up / documenting your code, you will use the notes you generated throughout the competition to update the project readme.md file with a clear narrative of the problem statement, what you did, what worked (well and not-so-well), and what insights you gained through the process.