Phase I - Describe dataset

1. Explain data (i.e.., simple exploratory analysis of various fields, such as the semantic as well as intrinsic meaning of ranges, null values, categorical/numerical, mean/std.dev to normalize and/or scale inputs). Identify any missing or corrupt (i.e., outlier) data.
2. Define the outcome (i.e., the evaluation metric and the target) precisely, including mathematical formulas.
3. How do you ingest the CSV files and represent them efficiently? (think about different file formats)
4. Join relevant datasets
5. Split the data train/validation/test - making sure that no leaks occur, for example: normalize using the training statistics.

Phase II - Scalability, Efficiency, Distributed/parallel Training and Scoring Pipeline

1. What are the proposed features, and how are they derived from underlying data?
2. Do you need any dimensionality reduction?
3. Specify the feature transformations for the pipeline and justify these features given the target (ie, hashing trick, tf-idf, stopword removal, lemmatization, tokenization, etc..)
4. Other feature engineering efforts, ie interaction terms, Brieman's method, etc...)

Phase III - Baseline Algorithm

1. Create a baseline model (ie, logistic regression) and write a gap analysis against the Leaderboard.
2. Fine tune your baseline model

a. Is there a difference in performance? Is it related to features? Is it related to noise? What is impacting the model performance?

Phase IV - Select optimal algorithm and fine tune

1. Consider more sophisticated models: RF, GBT, something else?
2. Hyperparameter tuning using cross validation (think about bias/variance tradeoff, regularization, etc..)
3. Feature refinement

Phase V - Novel approaches

1. Continue fine-tuning
2. You may or may not pivot at this stage to a different algorithm, or try novel approaches.

Phase VI - Write up & Presentations

- Clean up your code a. **Make sure you have an end-to-end pipeline** does the data change frequently? Do we need real-time results? How frequently will this model need to be retrained?

Presentation

- Introduce the business case
- What worked, what didn't? Include a comparison chart of things tried.
- What did you learn?
- What kind of scalability issues did you encounter, and how did you solve them?
- How long did your model take to run?
- What did you do to optimize the training time?
- Given the dataset and problem you chose, how important is the training time? Ie,
- Introduce the dataset
- Summarize EDA and feature engineering
- Summarize algorithms tried, and justify final algorithm choice
- Discuss evaluation metrics in light of the business case
- Discuss performance and scalability concerns
- Summarize limitations, challenges, and future work.