Jahanna Chronicle, a technology company based in Lunakick, has reached out to you with a task that involves predicting the impact of a tweet and prepare a report on your analysis. You could think about ‘impact’ as a value that could help Jahanna Chronicle decide if the tweet could go viral.

Chronicle’s data team has worked hard and prepared a dataset for you. They compiled the data and decided to share it via HTTP. So here is the link, directly from the Chronicle’s data team!

Dataset: Attached as part of email

There are **15 features** and 1 dependent variable (also called as the output variable; Here it is named as ‘impact’).

**Post Content** - The text in the tweet

Sentiment score - Ranges from -20 to +20 (0 - neutral)

Post Length - The length of the tweet

Hashtag Count - The number of hashtags used in the tweet

Content URL Count - The number of URLs mentioned in the tweet

Tweet Count - The total number of tweets posted by the author of the tweet

Followers Count - The number of followers of the author of the post

Listed Count - the number of lists the post author is a part of

Media Type - The media type of the post (Text, image, video)

Published Datetime - The published time of the tweet

Mentions Count - The number of user mentions in the tweet

Post Author Verified - 1 if author is a verified user

Likes - Likes received for the tweet

Shares - Retweets received for the tweet

Comments - Number of comments for the tweet

For some weird reason, Chronicle’s engineering team is adamant that you explore modeling using a decision tree (with some form of boosting and pruning), neural network and linear regression but give the model that fits best.

To summarize here are the tasks required by Jahanna Chronicle:

● Give the model and code (by uploading it to a public repo) that can best predict the impact score

● Prepare a report of not more than 6 pages of your findings of data. It should include analysis like the training and testing error rates you obtained running the various learning algorithms on your problems, graphs that show performance on both training and test data as a function of training size, Why did you get the results you did? How fast were they in terms of wall clock time? How much performance was due to the problems you were given? How about the values you chose for learning rates, stopping criteria, pruning methods, and so forth (and why doesn't your analysis show results for the different values you chose?)? Which algorithm performed best? How do you define best? Be creative and think of as many questions you can, and as many answers as you can.

● All the code that was used to prepare a report (also by uploading it to a public repo)

Note from Jahanna Chronicle: The engineers at Chronicle must be able to recreate all experiments/graphs/results. (So you may want to include a readme on how to run the code)