Part 1.1 - Understanding APIs (5 points)

• 1.1.1 (2) How many API calls were required to collect the submissions?

* **Ans: 30. Since there is a limit of 100 for a single API call, the 1000 limit results in 10 calls. Since we made 3 praw calls for the 3 subreddits, the total calls made is 30.**

• 1.1.2 (1) Why did we set the submission limit at 1000?

* **Ans: The calls will take longer as the limit is increased. Between calls, the praw wrapper puts in a 2-second delay to confirm with API rules (Ref:** [**https://praw.readthedocs.io/en/v3.6.2/pages/getting\_started.html**](https://praw.readthedocs.io/en/v3.6.2/pages/getting_started.html) **). With the limit at 1000, there are 10 API calls with a 2-second delay which would take at-least 19 seconds.**

• 1.1.3 (2) How long, in minutes, would it take you to collect 1000 posts from 25 different subreddits? What about from 500 different subreddits? Hint: You’ll have to consider how many API requests you are allowed to make

* **Ans: With 3 subreddits, it takes at-least 57 seconds. So, for 25 subreddits, it would take at-least ~8 minutes (57 \* 25 / (3 \* 60)). For ~500 subreddits, it would take at-least ~160 minutes (57 \* 500 / (3 \* 60)).**

Part 1.2 Thinking about your sample (3 points)

• 1.2.1 (1) Do you think these posts are representative of all the posts on that subreddit?

* **Ans: No.**

• 1.2.2 (2) Why or why not? That is, if you think so, why do you think there’s not much sampling bias here? If not, what do you think might be different about these top posts than other posts?

* **Ans: A representative sample of all the posts has to adequately replicate the subreddit’s posters and selecting 1000 top posts from a subreddit does not do this because of the following factors: a) The ‘non-top’ posters are not represented in the sample, and b) The sample size would be too small compared to the total posts in most subreddits.**

2.1.6 (No code for this) Mathematically, what is the relationship between the standard deviation of the number of upvotes and the variance of upvotes?

* **Ans: Variance of upvotes is the square of the standard deviation of upvotes.**

**2.2.2** - Based on your histogram, which subreddit would you say is the *least* popular? (Note, there is more than one reasonable answer here. We are looking mostly for how you justify your response using the histogram)

* **Ans: r/learnprogramming is the least popular based on the fact that it has the lowest upvotes at ~1.78M.**

**2.2.3** - **Approximately (within 1-2 percentage points)** what percent of top posts for each of the three subreddits plotted below have less than 100,000 upvotes? (Give answers for each subreddit)

* **Ans: r/news - ~84%, r/science - ~98.5%, r/worldnews - ~79%.**

**2.2.4** - **Approximately (within 1-2 percentage points)** what is the probability that a post on each of the three subreddits plotted below has more than 70,000 upvotes? (Give answers for each subreddit)

* **Ans: P(r/news > 70K) = 0.33, P(r/science > 70K) = 0.13, P(r/worldnews > 70K) = 0.97.**
* **2.2.8** - Using what you have plotted, make an argument for which of the four subreddits is the most "up and coming" - i.e. the one that seems to be getting more popular over time. NOTE: There is more than one reasonable answer here. We are looking for how you justify your answer using the (plotted) data.

Let's start by looking at the continuous variables. Those are:

* total\_awards\_received
* downs
* gilded
* num\_comments
* num\_crossposts
* num\_reports
* created\_utc
* subreddit\_subscribers
* **2.3.1**- There are two continuous variables that are very clearly not going to be useful for our analysis. Identify them, and explain why they are not useful (**note: you do NOT need to know why these variables take on the values they do in our data. You just need to know why we don't want to use them!**)

Let's now look at our (supposedly) binary categorical variables:

* is\_crosspostable
* is\_self
* media\_only
* is\_video
* locked
* Over\_18
* **Ans:** “is\_crosspostable” and “media\_only”.
* **2.3.2**- There are two (supposedly) binary variables that are very clearly not going to be useful for our analysis. Identify them, and explain why they are not useful (**note: you do NOT need to know why these variables take on the values they do in our data. You just need to know why we don't want to use them!**)

Finally, let's look at our remaining variables, which are categorical. One of these, title (the post's title), is potentially a *very* useful feature... but we haven't yet learned how to use it. So, for now, we're not going to. The other categorical features are:

* subreddit\_id
* subreddit\_name\_prefixed
* Permalink
* **Ans:** “subreddit\_id”/”subreddit\_name\_prefixed” and “Permalink”
* **2.3.3** - Explain why we it is not useful to use *both* subreddit\_id and subreddit\_name\_prefixed in any predictive analysis of per-post upvotes.
* **2.3.4** - Explain why it is not useful to use permalink in any predictive analysis of per-post upvotes.
* **2.3.6** - Describe, briefly (a sentence) the relationship between num\_comments and upvotes.

Questions to check understanding

* **3.1.1** - Report your error on the test data, in RMSE. State what this metric means for the expected error in terms of the number of upvotes (not log upvotes!) you should expect to be off on any given prediction

Also, a few questions to target your understanding of how we set up the model:

* **3.1.2** - What did the whole one-hot encoding thing on subreddit\_name\_prefixed actually do?
* **3.1.3** - What does the argument drop = "first" do for us when we are doing that to subreddit\_name\_prefixed?
* **3.1.3** - Why did we need to add one to the outcome variable before using log?

Since some ups are 0, we need +1 to avoid np.log returning divide by zero error.

* **3.1.4** - What does the StandardScaler do? Why do we want to do that?

OK. Having looked at our RMSE, we should now realize that we have to be careful about assuming that this one statistic is actually a good estimate of how far we're going to be off on any prediction, selected at random. To see this, let's do the following:

* **3.1.5** - Provide a scatterplot that compares the true values in y\_test to the absolute value of the difference between y\_test and your predictions. **The axes should be on the original scale** (i.e. not the log scale you're predicting on.
* **3.1.6** - What does this plot suggest about how well your model fits the data as the true number of upvotes changes?

## **Part 3.2 - Exploration of regression coefficients**

Now, let's look at the effects of our variables for this last model (with the logarithms of the independent variables). Carefully re-combine your features with their labels (*hint, encoder.categories\_ will be your friend, and remember, we dropped the first category!*)

* **3.2.1** - What is the strongest positive predictor of upvotes? How many more log(upvotes+1) does a one standard deviation increase in the feature correspond to?
* **3.2.2** - What is the strongest negative predictor of upvotes? How many fewer log(upvotes+1) does a one standard deviation increase in the feature correspond to?

# **Part 3.3 - 574 Only - Attempting to Improve Your Predictions**

In class, we talked about a few things we might do to improve our model's predictions. These include adding interaction terms, including different functional forms of a feature, using a different model, etc. Here, we ask that you implement at least two of these, and then re-evaluate your model. We'll ask some of the teams with some of the more interesting/effective ideas here to come present their solutions to the class!

* **3.3.1** - Describe at least two changes you made -- at least one to the feature set, and at least one different model -- to try to improve prediction. Explain *why* you think that these changes make sense, given the Exploratory analyeses above, or any other exploratory analysis you choose to do.
* **3.3.2** - By how much did your RMSE improve? Which change that you made improved it the most? How do you know?