DAT Sample Modeling Challenge

Welcome to this week's bonus assignment! If you choose, you can spend time this week cutting your teeth on a sample modeling challenge.

Your Objective: With Linear Regression as your hammer, see if you can transform your dataset into a nail that it can pound to accurately predict the amount of money a particular movie will be able to gross. This is the **revenue** column in the training set.

How Your Model Will Be Evaluated: At the start of class on December 2nd, I'm going to run a script that grades everyone's final submissions live in class, to determine who the winner is.

How You Should Submit Your Answers: Your final answers should be in a 2-column csv file, that have the following column headers:

id: the values of the 'id' column in the test set

prediction: your predictions for revenue on the test set.

How you will be graded: Submissions will be graded using the r_squared value on the test set.

Key Points to Pay Attention To:

- See if you can find a coherent way to extract information out of the many categorical columns in this dataset. You can't work with all of these, but pick at least one that looks meaningful to you and see if you can do something to make it more useful.
- Start with a simple model and iteratively add to that, as opposed to starting with something very complicated
- There are empty values…..what are the most likely causes for these? Should we interpret them as being randomly generated, or do they mean something specific? Are they specific to particular periods of time?
- There is a date column. Can you extract information about what season it was released? If it was done on a Friday? Wednesday? How long was its opening weekend?
- Things like popularity and budget have meaningful correlations to final revenue, but what about movies that have higher or lower budgets compared to their categorical average?
- Maybe dramas tend to be rated more highly than other films, so columns that compare stats about a film to its category average are often useful ways to extract information out of different features.
- How do different validation strategies compare to one another? The data is sorted according to date…..are your scores consistent across time?

Again…….you probably won't have enough time to go through **all of these,** but see if you can find at least one way to meaningfully improve your score and make sense of the modeling process.

About the Data Set:

The dataset is a list of over of over 10,000 movies dating back from 1960, and sorted sequentially. Ie, the most recent values are in the test set.

It was taken from the movies database, and you can read more about its api here: https://www.themoviedb.org/documentation/api

The dataset contains the following columns:

- **id:** A numeric index. Does not represent anything.
- **popularity:** A fairly holistic measure that measures search intent for a particular movie. More can be read about it here: https://developers.themoviedb.org/3/getting-started/popularity
- **budget:** Budget for the movie, in nominal (non-inflation adjusted) dollars
- **budget_adj:** Inflation adjusted budget for the movie
- **original_title:** Title of the movie
- **cast:** List of the starring actors in the film
- **homepage:** Official homepage for the movie.
- **director:** Movie's main director.
- **tagline:** Official slogan of the movie
- **keywords:** Search keywords associated with the movie, as determined by TMDB
- **overview:** Official description of the movie.
- **runtime:** Official length of the movie
- **genres:** List of genres associated with the movie
- **production_companies:** Companies associated with the making of the movies
- **release_date:** When the movie was first released
- **vote_count:** Number of ratings that a movie received from users
- **vote_average:** Average value of a user review, from 0-10
- **budget_adj:** Inflation adjusted budget for the movie
- **revenue(train only):** This is the inflation adjusted revenue that the movie made. This is also the target variable. It's what you're trying to predict on the test set.