# Assignment 3

## Genre Model

For the genre model, I used an interpretation of tf-idf weighting for modeling genre vectors.

- As term frequency, I used genre frequency rather than the vote counts themselves, their weights are bound to the number of votes.
    - Since $\log(m^n) = n. \log(m)$, the magnitude of genre votes is counted, the number of total vote delimits its effect within document
- For inverted document frequency, the weighting is the same as in tf-idf model. The total occurrences of each genre is stored with the genre vocabulary. And it is divided with the number of genres in the vocabulary

```
genre_frequency = math.log10(total_vote**vote)

inverted_doc_freq = math.log10(len(dictionary.keys())/dictionary[genre])

count_dictionary[genre] = abs(genre_frequency * inverted_doc_freq)
```

## Parameters

- There are no thresholds for the restrictions on term weights are implemented within model
- # of terms: The number of genre in vocabulary is 450. So any minimum threshold might turn the vectors more sparse than it should be.
    - Any supra-genre is counted as a separate genre. Their magnitude is accounted in this point
- With the multiple test runs using $\alpha$ parameter, I obtained a rough optimal value as $\alpha$: 0.35 .
    - For the most query runs, $\alpha$ parameter at $0.25 \leq \alpha \leq 0.65$, the function returns non-zero precision values
    - Then after $\alpha$ 0.35, decrease in precision is observed.