An Analysis of Reddit Comments Focusing on Movie and Actor Popularity

Gus Nard, Luke Kilgore
CS455: Introduction to Distributed Systems
Computer Science Department, Colorado State University

Background Information

- Most movie reviews and ratings are curated by or take money from companies that make movies.
- We wanted to get a less biased look at what the general public thinks
- We chose two datasets to do the analysis with:
  - Reddit Comments Dataset (12/2005 - 12/2009) - over 50GB
  - The Movies Dataset (info about 57000 movies) - around 900MB

- Using Apache Spark and Hadoop / HDFS to implement the analysis questions

Problem Characterization

- We wanted to answer the following eight questions:
  - How does reddit popularity correspond to movie rating?
  - Who is the most liked/disliked cast or crew member? - skipped
  - Which producer spends the most on making movies?
  - How many comments are there about high budget vs low budget films?
  - What are the most commented about movies per country? - changed
  - How much do plot keywords about a movie show up in comments? - skipped
  - Which actors are mentioned all the time vs actors that are only mentioned when a movie releases? - skipped

Methodology

- Question 1:
  - For each movie, found:
    - <title, popularity>
  - Filtered comments to ones that only contain a movie title.
  - Mapped each comment to:
    - <<title, popularity>, 1>
  - Reduced by key and summed values to get:
    - <<title, popularity>, mentions>
  - Sorted output by popularity.

- Question 3:
  - For each movie, found:
    - <movie id, title>
  - For each cast/crew member, found:
    - <movie id, name>
  - Found only comments that contained a name, flat mapped them to:
    - <name, movie id, 1>
  - Reduced by key, summed values, and replaced ids with titles to get:
    - <<name, title>, mentions>
  - Sorted by mentions.

Methodology

- Question 4:
  - Filtered cast/crew members to only get people who's job is 'Producer', mapped to:
    - <name, movie id>
  - Mapped the information about each movie to:
    - <movie id, budget>
  - Use these two lists to get pairs of:
    - <name, budget>
  - Reduced by keys, summed values to get final answer of:
    - <name, total spent>

- Question 5:
  - Determined a good cutoff value for what is considered a low or high budget film:
    - Low < $40,000,000
    - High >= $40,000,000
  - Mapped each movie to pairs of:
    - <title, budget>
  - Found comments that contain movie titles.
  - For each of these comments, found it's movie budget pair, and output:
    - <"High", 1> for high budget
    - <"Low", 1> for low budget
  - Reduced by keys and summed values to get final answer.

Methodology

- Question 6:
  - Changed question because of limitation in comment data
  - Flat map each movie to pairs of:
    - <production country, 1>
  - Filtered out errored data that replaced production country with production company.
  - Reduced by keys and summed values to get pairs of:
    - <production country, number of movies produced>

Performance Benchmarks

- Turnaround times for some questions were high compared with the similarly sized dataset in HW3.
- Some questions were quick, taking 1-2 minutes, others took 20-40 minutes.
- The high turnaround times can mostly be attributed:
  - Using nested loops in many places.
  - Using string contains on comment bodies inside nested loops.
- Frequent use of lambda functions may or may not contribute to high turnaround times.

Performance Benchmarks

- For a few questions, accuracy was probably quite low
- Some examples:
  - While finding the most mentioned cast or crew member, it was impossible to tell if a comment was talking about a movie or not when referring to a cast or crew member.
  - For cast / crew with more general names, especially ones that don't have their last name in the dataset, there were probably many times when the comment was not even talking about that cast or crew member.
  - For movie titles, one word movie titles are mentioned quite a lot, even though the number of these mentions that were referring to that particular movie was probably much lower than is counted.

Performance Benchmarks

- The analysis of some of the questions was infeasible due to the number of false positives that would have been generated.
- This was most prominent in the question about plot keywords:
  - Each movie's metadata line has a list of keywords relevant to the plot of the movie.
  - The problem was that the keywords were much to general, for example:
    - "friendship", "childhood", "toy", "adventure", "rescue"
  - Keywords such as these would've generated countless false positives in everyday comments that were not talking about a movie.
  - Different movies share keywords, making it very difficult to determine what movie is being referred to.
- We later decided to skip this question because of these data limitations.

Insights & Conclusions

- Some approaches to a problem might be easier or faster than others.
- Data mining is relatively easy, but finding an answer for a specific question with mined data requires more work.
- Much of our results were different than we had expected.
- Fully utilizing the distributed nature of spark made our analyses run orders of magnitude faster than they would’ve without it.
- We concluded that analyses and questions like these could be used in the future by streaming services, social media companies, and movie production companies to analyze comments on the fly in order to make better movies that more people like.