INTRODUCTION TO PROGRAMMING FOR DATA SCIENCE FINAL PROJECT REPORT CODEFORCES COLORISM?

R12922109 Chien-Yi Chien

National Taiwan University

ABSTRACT

This short essay addresses Codeforces' "Colorism," where high-rated individuals receive undue tolerance for offensive remarks. Recently, a Codeforces blogger BledDest critiques the normalization of disrespectful language and advocates for responsible discourse. Our research employs web crawling and statistical analysis, including scatter plots and sentiment analysis to quantify the prevalence of Colorism. The goal is to shed light on this issue by examining objective investigations.

1. INTRODUCTION AND BACKGROUND

Codeforces, established in 2010, stands as a renowned competitive programming platform that attracts a diverse community of programmers and problem-solving enthusiasts worldwide. It serves as a hub for hosting competitive programming contests and fostering a collaborative environment for individuals to enhance their algorithmic and coding skills.

One distinctive feature of Codeforces is its rating system, a dynamic metric that reflects a user's performance in contests. Users are assigned different colors based on their ratings, creating a visible hierarchy that signifies expertise. This color-coded system not only recognizes accomplishments but also inadvertently shapes perceptions within the community.

Beyond its primary function as a competitive programming platform, Codeforces also incorporates social networking elements. Users can engage in discussions, share insights, and comment on each other's work. However, this integration of social interaction introduces an unintended consequence—Colorism. The correlation between a user's color, representing their skill level, and the perceived intelligence in their communication has led to instances where individuals with higher ratings receive preferential treatment.

In this context, we delve into an exploration of Colorism on Codeforces, examining how the platform's inherent features contribute to the differential treatment of users based on their rating-related color. The following sections will present an in-depth analysis, incorporating web crawling and statistical methods, to shed light on the extent of this phenomenon and its potential impact on the Codeforces community.

2. METHODOLOGY

2.1. Data Collection

We gathered comments from Codeforces blogs using the official API. To ensure a focus on recent trends, we randomly selected blogs with IDs ranging from 100,000 to 120,000. Subsequently, we extracted the usernames of these commentators and utilized the API to retrieve their respective ratings. This process allowed us to analyze the interactions of users with varied ratings, shedding light on the recent dynamics within the Codeforces community.

2.2. Data Preprocessing

During the data preprocessing stage, we identified redundant entries in the raw API output, such as comment ID and timestamp. These entries were deprecated, streamlining each comment to include only the commentator's handle, comment rating, and content. Likewise, for each user, we retained essential information by keeping the rating, friendOfCount, and maxRating fields. This refinement ensures that our dataset is focused on pertinent details, facilitating a more efficient and meaningful analysis of Codeforces user interactions.

2.3. Statistical Analysis

In the process of statistical analysis, we honed in on comments with distinctive opinions by filtering for those where the difference between upvotes and downvotes surpassed $+50$ or dipped below -50. This focused approach allowed us to concentrate on comments that elicited strong reactions within the Codeforces community. Furthermore, we explored the correlation between a user's rating and their comment rating, as well as the correlation between the number of a user's friends and their comment rating. Correlation coefficients were calculated for both of these relationships.

In tandem with correlation investigations, we conducted sentiment analysis to discern the prevailing sentiment associated with words used in comments. Leveraging the Python library TextBlob, as introduced in our lectures, we categorized words as either more affirming or more negative. This facet of the analysis contributes practical insights to Codeforces users

by offering guidance on language preferences within the community.

This multifaceted approach enhances our understanding of user interactions on Codeforces, providing valuable information about the dynamics between user ratings, comments, and sentiment expressions.

3. RESULTS

3.1. User-Based Analysis

In this subsection, we explore the correlation between user information and their comment ratings. Two scatter plots are presented to illustrate these relationships. The first scatter plot (Fig. 3.1) juxtaposes user rating on the x-axis against comment rating on the y-axis. The second scatter plot (Fig. 3.2) introduces a logarithmic transformation (base 2) of the number of user's friends on the x-axis, plotting it against comment rating on the y-axis. These visualizations aim to elucidate patterns and relationships between user attributes and the perceived impact of their comments within the Codeforces community.

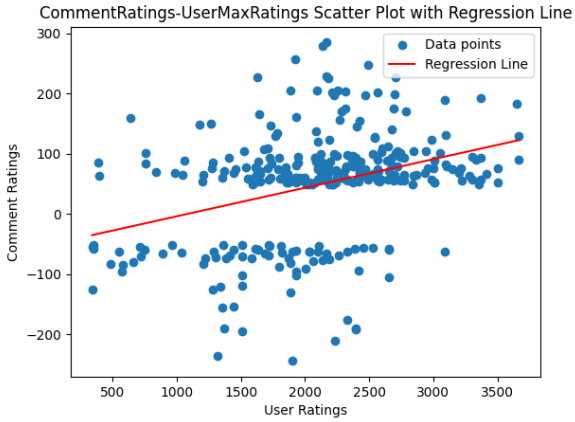

Fig 3.1. CommentRatings-UserMaxRatings Scatter Plot with Regression Line

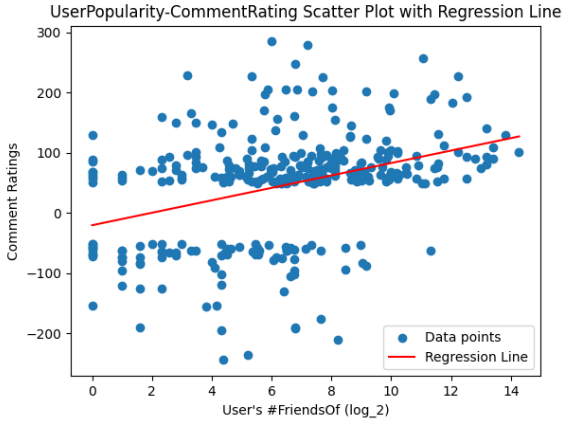

Fig 3.2. CommentRatings-UserPopularity Scatter Plot with Regression Line

Observing both scatter plots, a discernible positive correlation is evident. This leads to the conclusion that users with higher ratings are more likely to receive upvotes on their comments. Additionally, the second plot, depicting the relationship between the logarithm (base 2) of the number of user's friends and comment rating, suggests that users with a larger social network tend to garner more upvotes. This infers that users receiving increased attention are likely to be more conscientious in their comments, steering clear of controversial speech, consequently resulting in higher comment ratings.

3.2. Sentiment Analysis

In this section, our anticipation was that users incorporating proper nouns like "binary search" or "dynamic programming" in their comments—indicative of instructive content—might receive more upvotes. Surprisingly, the top 5 words associated with positive sentiments turned out to be "good," "also," "would," "many," and "best" while the top 5 words linked to negative sentiments were "bad," "wrong," "hate," "sorry," and "worst." Regrettably, these results closely resemble patterns typically observed in sentiment analysis across various social media platforms, deviating from our initial hypothesis.

4. CONCLUSION

In conclusion, our investigation affirms the presence of Colorism on Codeforces. Users receiving greater attention demonstrate a heightened awareness of their speech, contributing to a positive correlation with comment ratings. Furthermore, sentiment analysis results reveal that patterns observed in language expression on a competitive programming contest platform closely parallel those found in broader social media contexts. These findings underscore the importance of fostering inclusivity within the Codeforces social network, emphasizing the need for a mindful and equitable community culture.

5. RECOMMENDATIONS FOR FUTURE WORKS

For future investigations into the presence of Colorism on Codeforces, a potential avenue involves integrating machine learning techniques, which could be explored in more advanced courses. One approach could be to establish models capable of predicting comment ratings independently from user information. Subsequently, these models could be scrutinized to determine their accuracy, especially when applied to comments from high-rated users. This method provides a nuanced perspective and serves as a compelling direction for researchers intrigued by the Colorism issue on Codeforces, offering an avenue for more in-depth exploration and analysis.

6. TOOLS AND REFERENCES

- Codeforces blog: I don't want to get this normalized
- Codeforces official API
- Python libraries
  - `requests` library for web crawling
  - `json` library for parsing results from API into a JSON file
  - `matplotlib.pyplot` for plotting scatter plots
  - `scipy.stats` for sketching regression lines
  - `nltk` for parsing comments
  - `TextBlob` for sentiment analysis
- Overleaf for this report completion environment
- ChatGPT for rephrasing and expressing ideas in a more accurate and professional way