Ðauphine UNIVERSITÉ PARIS | PSL

Classifying Relationship-Oriented Subreddit Posts Using NLP and Machine Learning

Author:
Mehyar MLAWEH
mehyar.mlaweh@dauphine.tn

Contents

Problem Statement

The core problem this project aimed to address was:

- Can we accurately classify posts as belonging to either `r/dating_advice` or `r/relationship_advice` based solely on the text content of the posts?

Sub-problems included:

- What are the most common topics and keywords in each subreddit?
- What linguistic features best differentiate posts between the two subreddits?
- Which machine learning algorithms perform best for this classification task?
- What insights can we gain about the nature of dating vs relationship advice from this analysis?

Introduction

This report details an extensive natural language processing (NLP) project focused on classifying posts from two popular relationship-oriented subreddits: `r/dating_advice` and `r/relationship_advice`. The primary goal was to scrape data from these subreddits, perform comprehensive exploratory data analysis, and build sophisticated classification models to predict which subreddit a given post originated from based solely on its text content.

Reddit, as one of the largest and most diverse social media platforms globally, hosts a myriad of highly active communities discussing an extensive range of topics. The `r/dating_advice` and `r/relationship_advice` subreddits stand out as particularly popular forums where users seek guidance on romantic relationships and interpersonal issues. By meticulously analyzing the language patterns and key topics prevalent in these communities, we aim to gain valuable insights into the types of relationship challenges people commonly face and how these challenges differ between individuals in the dating phase and those in established relationships.

This project leveraged advanced web scraping techniques to gather a substantial corpus of post data, employed state-of-the-art natural language processing methods for in-depth text analysis, and utilized cutting-edge machine learning classification algorithms to construct robust predictive models. The primary evaluation metric for our models was classification accuracy, supplemented by additional performance measures to ensure a comprehensive assessment of model efficacy.

Motivation

The motivation behind this project stems from several key factors:

1. **Understanding Relationship Dynamics:** By analyzing the content of these subreddits, we can gain valuable insights into the evolving nature of modern relationships, the challenges people face, and how these issues vary across different stages of relationships.

2. **Enhancing Support Systems:** Identifying common themes and concerns in these forums can help inform the development of more targeted and effective support resources for individuals navigating relationships.

3. **Improving Online Community Management:** The ability to accurately classify posts can aid in the efficient moderation and organization of online

forums, ensuring users receive appropriate advice and support.

4. **Advancing NLP Techniques:** This project serves as a practical application of NLP and machine learning techniques in the domain of social media analysis, potentially uncovering new methodologies or insights applicable to similar text classification tasks.

5. **Sociological Insights:** The patterns and trends identified through this analysis may provide valuable data for sociologists and relationship experts studying contemporary interpersonal dynamics.

The code for this project is available in the GitHub repository: **reddit-relationship-classifier repository**.

Chapter 1

Data Collection

1.1 Scraping

Web scraping was performed using the Python `requests` library to access the Reddit API. Three separate scraping passes were conducted to gather a diverse dataset:

1. Initial scrape of `r/relationship_advice`
   - Scraped 981 posts
   - Used the "hot" sorting algorithm
   - Timeframe: July 6, 2023
2. Second scrape of `r/relationship_advice/new`
   - Scraped 999 posts
   - Used the "new" sorting algorithm to get more recent posts
   - Timeframe: July 8, 2023
3. Scrape of `r/dating_advice`
   - Scraped 1865 posts
   - Used default sorting

The scraping process utilized the following code structure:

```
url = 'https://www.reddit.com/r/relationship_advice.json'
headers = {"User-agent": "not-robot-mehyar"}
res = requests.get(url, headers=headers)
the_json = res.json()
```

Data was iteratively collected using a loop to paginate through results:

```
posts = []
after = None
for i in range(40):
    if after == None:
        params = {}
    else:
        params = {'after': after}
    url = "https://www.reddit.com/r/relationship_advice.json"
    res = requests.get(url, params=params, headers=headers)
    if res.status_code == 200:
        the_json = res.json()
        for post in the_json['data']['children']:
            posts.append(post['data'])
        after = the_json['data']['after']
    else:
        print(res.status_code)
        break
    time.sleep(1)
```

Data was stored in pandas DataFrames and exported to CSV files for further processing. Ethical considerations ensured only publicly available post data was collected, with appropriate delays between requests to avoid overloading the API.

1.2 Data Overview

The scraped data included various fields such as post ID, author, title, selftext, timestamp, number of comments, upvotes, and subreddit. After initial cleaning, the dataset contained 981 posts from the first scrape and 999 posts from the second scrape of r/relationship_advice.

Figure 1.1 shows the distribution of post lengths in the r/relationship_advice subreddit. The mean post length is indicated by the red dashed line.

1.3 Word Frequency Analysis

To gain insights into the most common topics and themes, we analyzed word frequencies in both post titles and body text.

Figure 1.2 displays the most frequent words appearing in post titles. Words like "relationship," "girlfriend," and "boyfriend" are unsurprisingly common.

Figure 1.3 shows the most frequent words in the body text of posts. This provides insight into the common themes and concerns discussed in the subreddit.

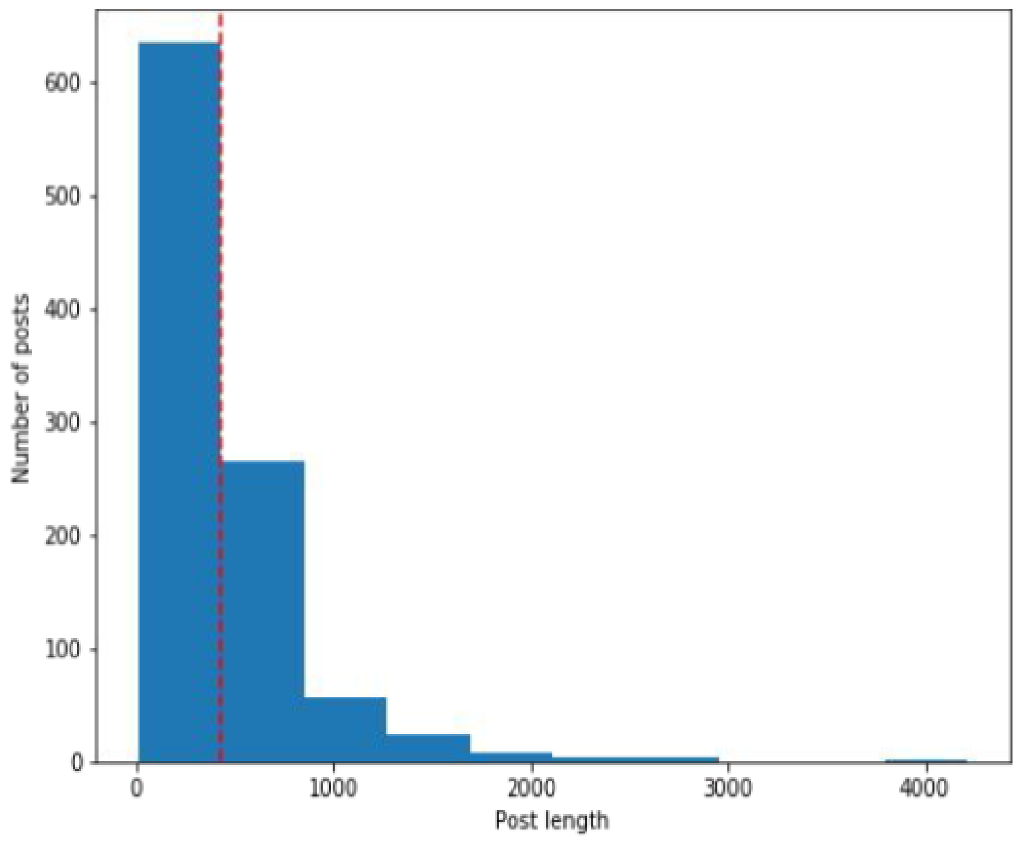

Figure 1.1: Distribution of relationship advice post lengths

1.4 Annotation

The scraped data was automatically annotated based on the subreddit of origin:

- `dating_advice` posts labeled as 0
- `relationship_advice` posts labeled as 1

This binary classification setup allows for straightforward model training and evaluation.

- Removed null rows (2% of data)
- Combined post titles and body text
- Binarized target variable (0: dating_advice, 1: relationship_advice)
- Tokenization of titles and post text
- Removal of stop words

The preprocessing steps resulted in a clean dataset ready for further analysis and modeling. The tokenization process revealed:

- 15,215 total words in titles, with 2,734 unique words
- Maximum title length of 31 words

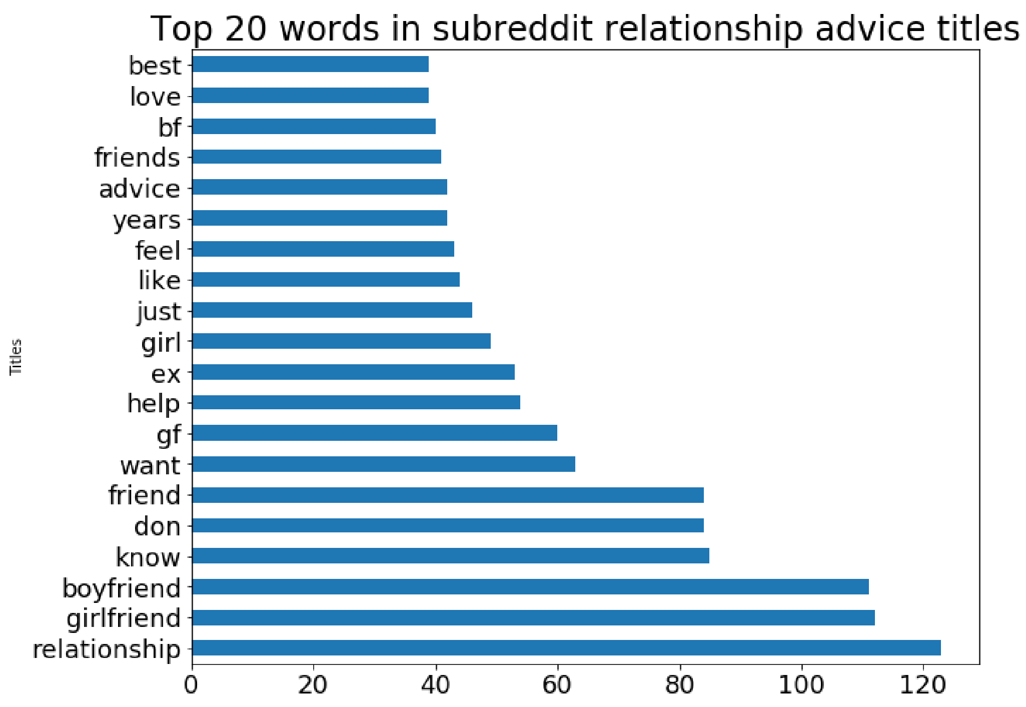

Figure 1.2: Top 20 words in subreddit relationship advice titles

- 1,053,774 total words in post text, with 25,392 unique words
- Maximum post length of 9,751 words

These statistics provide valuable insights into the complexity and diversity of the collected data, setting the stage for more advanced natural language processing tasks in subsequent stages of the project.

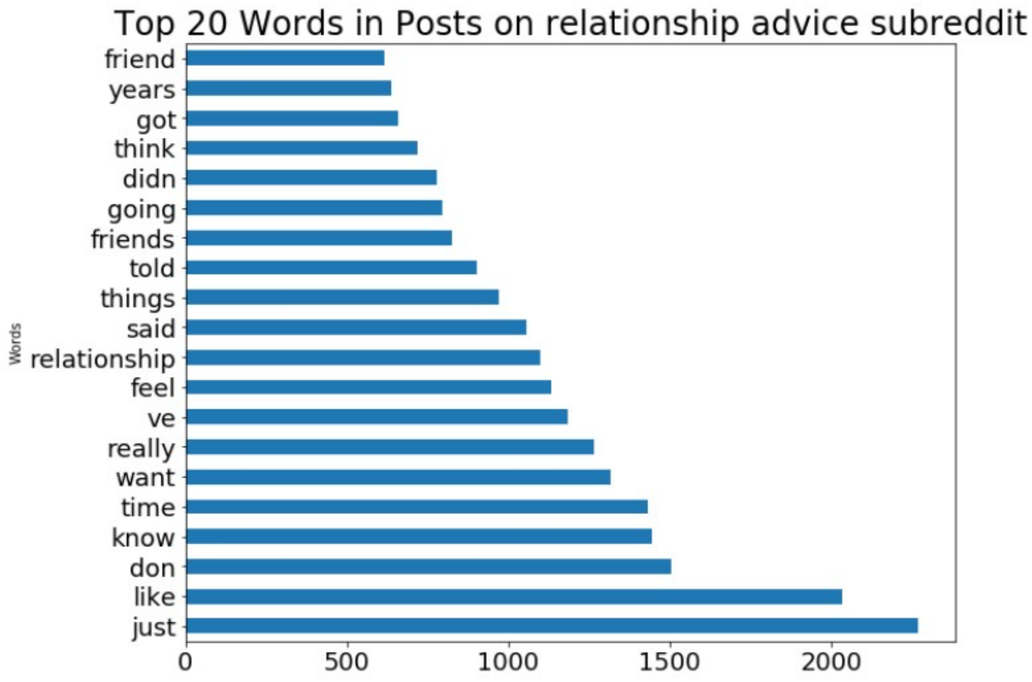

Figure 1.3: Top 20 Words in Posts on relationship advice subreddit

Chapter 2

Exploratory Data Analysis

2.1 Data Processing and Initial Analysis

After the initial data processing steps, further analysis was conducted to understand the distribution of the data across subreddits and explore the relationship between subreddits and the number of comments.

2.1.1 Distribution of Posts Across Subreddits

To visualize the distribution of posts across the two subreddits, a histogram was created:

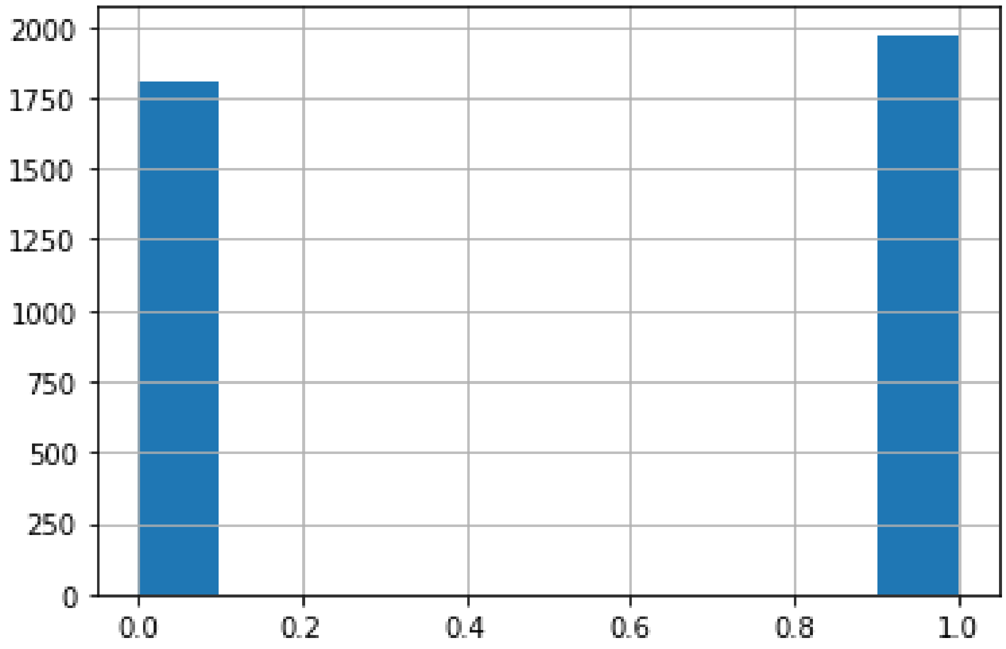

Figure 2.1: Distribution of posts across subreddits

This histogram shows the frequency of posts in each subreddit, allowing us to see if there's a balance or imbalance in the dataset between 'datingadvice' and 'relationshipadvice' subreddits.

2.1.2 Binarization of Target Variable

For further analysis and modeling, the subreddit column was binarized:

- 'datingadvice' was mapped to 0
- 'relationshipadvice' was mapped to 1

This binarization allows for easier statistical analysis and prepares the data for binary classification models.

2.1.3 Number of Comments Analysis

To compare the distribution of the number of comments between the two subreddits, a box plot was created:

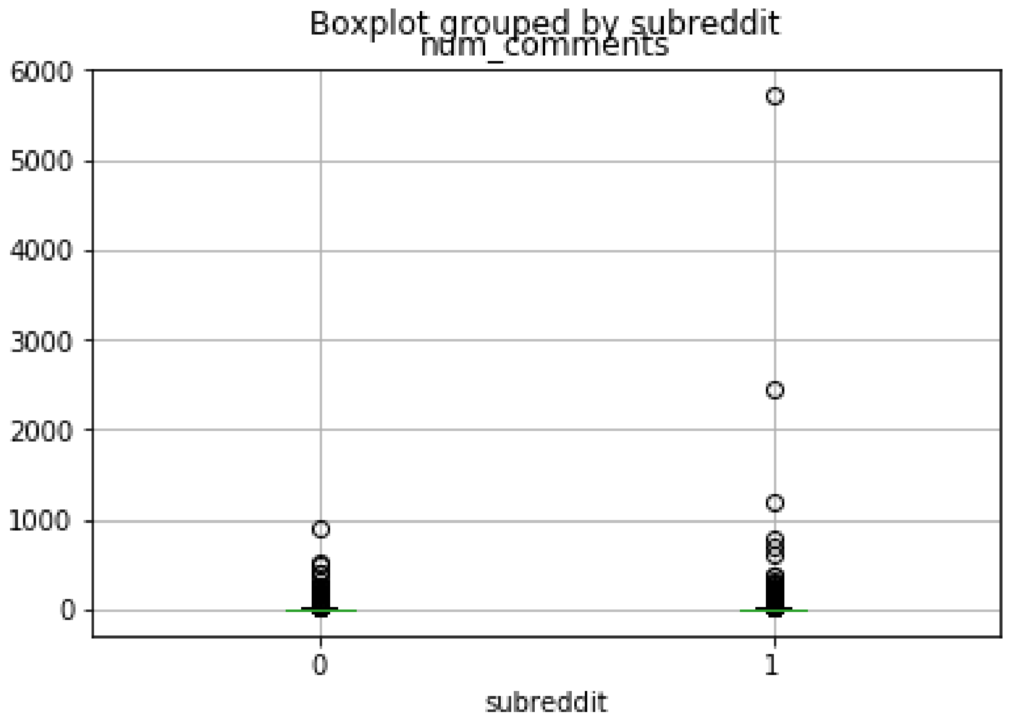

Figure 2.2: Distribution of number of comments by subreddit

This box plot allows us to compare the distribution of the 'numcomments' column between the two subreddits. Key observations from this plot include:

- Median number of comments for each subreddit
- Spread of the number of comments
- Presence of outliers in comment numbers
- Any significant differences between the two subreddits in terms of comment engagement

These visualizations and analyses provide insights into the engagement levels and distribution of posts across the two subreddits, which can be valuable for understanding the nature of the data and informing further modeling decisions.

Chapter 3

Classification

This chapter details the process and results of our classification task, which aims to distinguish between posts from two different subreddits.

3.1 Data Preprocessing

The preprocessing pipeline consisted of several crucial steps:

- Combining title and post text for a comprehensive analysis
- Removing HTML tags and non-letter characters to clean the text
- Converting all text to lowercase for consistency
- Tokenization to break down the text into individual words
- Removing stop words to focus on meaningful content
- Lemmatization to reduce words to their base form

A custom function `post_to_words` was implemented to perform these preprocessing steps:

```python
def post_to_words(self_text):
    soup_text = BeautifulSoup(self_text).get_text()
    letters = re.sub("[^a-zA-Z]", " ", soup_text)
    words = letters.lower().split()
    stops = set(stopwords.words('english'))
    keep_words = [w for w in words if w not in stops]
    return " ".join(keep_words)
```

3.2 Feature Extraction

Two main vectorization techniques were employed:

- **CountVectorizer**: Creates a matrix of token counts

- **TfidfVectorizer**: Computes Term Frequency-Inverse Document Frequency (TF-IDF) features

We experimented with different n-gram ranges and vocabulary sizes to optimize feature extraction.

3.3 Models

Three primary models were evaluated:

1. Logistic Regression
2. K-Nearest Neighbors (KNN)
3. Multinomial Naive Bayes

3.4 Model Performance

3.4.1 Logistic Regression

3.4.1.1 With CountVectorizer

- Train accuracy: 99.5%
- Test accuracy: 74.6%
- Cross-validation score: 74.5%

3.4.1.2 With Optimized CountVectorizer (1-2 grams)

- Test accuracy: 75.7%
- Cross-validation score: 76.3%

3.4.1.3 GridSearchCV Optimized Pipeline

- Best parameters:
  - max_df: 0.98
  - max_features: 5000
  - min_df: 2
  - ngram_range: (1, 2)
- Train accuracy: 99%
- Test accuracy: 74%

3.4.1.4 With TfidfVectorizer

- Train accuracy: 88%
- Test accuracy: 76%
- Cross-validation score: 79%

3.4.2 K-Nearest Neighbors (KNN)

- Train accuracy: 74.7%
- Test accuracy: 58% (worse than baseline)

3.4.3 Multinomial Naive Bayes

- Train accuracy: 86.5%
- Test accuracy: 73%
- Cross-validation score: 71.6%

3.5 Feature Importance

Analysis of feature importance revealed that the word 'parents' is 8.2 times more likely to predict a post from the relationship_advice subreddit.

3.6 Model Comparison and Discussion

Logistic Regression with TfidfVectorizer showed the best overall performance, with a test accuracy of 76% and a cross-validation score of 79%.

The KNN model performed poorly compared to other models and even the baseline, suggesting it may not be suitable for this particular text classification task.

The Multinomial Naive Bayes model, while not outperforming Logistic Regression, still provided reasonable results and could be considered for ensemble methods in future work.

Conclusion and Future Perspectives

Our classification task demonstrated the effectiveness of Logistic Regression, particularly when combined with TfidfVectorizer, for distinguishing between posts from different subreddits. The preprocessing steps and feature extraction techniques played crucial roles in achieving high accuracy.

Future work could explore:

- Ensemble methods combining the strengths of different models
- Deep learning approaches such as LSTM or BERT for potentially improved performance
- More advanced feature engineering techniques to capture subtle differences between subreddits

Bibliography

[cou, 2020] (2020). A practical guide to countvectorizer and tf-idf in scikit-learn. *Medium*.

[cou, 2022] (2022). *Introduction to Text Mining: CountVectorizer and TF-IDF*.

[red, 2023] (2023). Reddit scraping: From data extraction to text analysis.

[Author, 2021] Author, N. (2021). A complete exploratory data analysis and visualization for text data. *Towards Data Science*. Accessed: 2024-12-26.

[Das, 2020] Das, A. (2020). A comprehensive naive bayes tutorial using scikit-learn. *Medium*. Accessed: 2024-12-21.