Sentiment Analysis and Content Recommendations on Reddit

Venkataramanan Venkateswaran
vv2265
New York University
New York City
vv2265@nyu.edu

Simardeep Singh Mehta
sm11377
New York University
New York City
sm11377@nyu.edu

Dhanesh Baalaji Srinivasan
ds7636
New York University
New York City
ds7636@nyu.edu

***Abstract*—The project titled "Sentiment Analysis and Content Recommendations on Reddit" is dedicated to mining insights from Tensorflow's Reddit dataset using advanced big data tools. Utilizing Pyspark, SparkML, and MongoDB, this project aims to conduct comprehensive sentiment analysis and develop tailored content recommendations for various Reddit communities. By analyzing millions of textual interactions, the project seeks to uncover the prevailing sentiments, identify authors with most posts, and enhance user engagement on the platform. The application of these sophisticated big data technologies not only enriches the user experience but also provides valuable insights into the collective mindset and preferences of Reddit's user base, demonstrating the power of combining big data analytics with machine learning in understanding and interpreting complex online community dynamics.**

I. Why is this a Big Data Problem?

The dataset under consideration contains 3,848,330 posts with an average length of 270 words for content, and 28 words for the summary. The overall dataset size is 18.09GB. Processing such a large dataset on a single machine would take a considerable amount of time. As such, it is more practical to scale the data processing using big data techniques horizontally. Horizontal scaling is more cost-efficient than vertical scaling, as it allows for multiple machines to work in coordination with accuracy. This approach can significantly reduce the time taken to process the data. This is where PySpark comes in. It is an incredibly useful tool that can preprocess data and perform necessary actions on it. SparkML is highly valuable for analyzing the TensorFlow Reddit dataset due to its scalability and comprehensive machine learning capabilities. It excels in processing large volumes of data, like those from Reddit, across distributed environments. SparkML offers a wide array of algorithms for tasks such as sentiment analysis and trend detection, essential for this dataset. Its seamless integration with Apache Spark ensures efficient data preprocessing and real-time processing, which is crucial for the dynamic and varied nature of Reddit's data. Additionally, SparkML's compatibility with other big data tools like MongoDB enhances its utility in handling complex big data challenges. MongoDB is particularly useful for managing the TensorFlow Reddit dataset due to its flexibility in handling diverse data types and its scalability. As a NoSQL database, MongoDB excels in storing unstructured data common in social media datasets, like text, user interactions, and metadata. Its schema-less nature allows for easy incorporation of various data formats without rigid structure constraints. Additionally, MongoDB's ability to handle large volumes of data and its efficient querying capabilities enable quick data retrieval and analysis, essential for big data projects. Its scalability, both vertically and horizontally, ensures that it can grow with the dataset, making it an ideal choice for the dynamic and expanding nature of data from a platform like Reddit. Overall, by horizontally scaling the data processing, we can reduce the time taken to process the data, while maintaining accuracy and cost efficiency. This will allow us to make effective use of the large dataset provided by the tensorflow's reddit dataset and extract valuable insights that can aid in our sentiment analysis and recommendation tasks.

II. Dataset

The TensorFlow's Reddit dataset comprises approximately 4 million records. Each record has an average length of 270 words for content and 28 words for the summary.

A. Dataset Features

The main features of the dataset include:

- **Dataset URL:** https://www.tensorflow.org/datasets/catalog/reddit
- **Size:** 18.09 GiB
- **Schema:** normalizedBody, subreddit_id, subreddit, summary, body, content, author, id.

B. Description of Features

- **normalizedBody:** Contains the entire post with formatting for newline character.
- **subreddit_id:** Each subreddit is given a unique id.
- **subreddit:** Contains the name of the subreddit the post belongs to.
- **summary:** Contains a summary of the post.
- **body:** Contains the entire Reddit post.
- **content:** Contains the content of the post without "TLDR".
- **author:** Contains the username of the person who created the post.
- **id:** Each author is given a unique id.

III. Architecture

To efficiently manage the vast TensorFlow Reddit dataset, which includes nearly 4 million posts from a variety of subreddits, a structured approach is necessary within the JupyterHub environment. The tensorflow_datasets library is used to cache the data by first downloading the entire dataset and then converting it into a TensorFlow-compatible serialized format, known as TFRecord.

TFRecord is a flexible and efficient binary file format that enables fast data loading and processing—a key consideration when working with large-scale datasets. The TFRecord files are organized and stored in a dedicated folder, acting as a repository for the dataset's tfrecords.

Spark's ability to handle TensorFlow data is established through the Spark-TensorFlow connector. This is configured via SparkConf, which specifies the necessary library for reading TFRecord files. The SparkContext and SparkSession are then initiated to interface with this data. Using the spark.read.format("tfrecord") method, the system efficiently loads the TensorFlow Reddit dataset, which is stored in TFRecord files within a specified directory. This seamless integration ensures that the vast dataset can be processed in a distributed manner, leveraging Spark's capabilities for scalable data analysis and machine learning, crucial for the sentiment analysis and recommendation engine components of the architecture.

After loading the TensorFlow Reddit dataset into Spark, the architecture splits into two main components: Sentiment Analysis and the Recommender System. For Sentiment Analysis, we employ two methodologies: TextBlob and DistilBERT. TextBlob is a straightforward NLP library that provides a user-friendly API for common language processing tasks, including sentiment analysis. It assigns polarity scores to the text, indicating whether the sentiment is positive or negative. On the other hand, DistilBERT is a more advanced model, a lighter version of BERT, fine-tuned specifically for sentiment classification. This approach offers a deeper and state-of-the-art understanding of the emotional context within the Reddit posts. To handle these analyses at scale, we leverage Spark's machine learning capabilities, particularly through SparkML. This allows us to process the vast volume of Reddit posts efficiently and derive valuable sentiment insights from them.

For the visualization phase, Pandas and Matplotlib are employed to interpret and display the results of the sentiment analysis. Pandas is used for its powerful data manipulation capabilities, organizing the data into a suitable format for visualization. Matplotlib is then utilized to create clear and informative visual representations of the sentiment data, such as graphs and charts. These visualizations are integral for quick and comprehensible insights into sentiment trends and model performance, allowing users to interact with and understand the complex data through a more accessible and graphical format.

The persistence layer of the system, powered by MongoDB, stores outputs from both sentiment analysis and the recommender system. MongoDB's flexibility in managing diverse data types is instrumental in storing TextBlob's sentiment scores, DistilBERT's predictions, and user preference data from the recommender system. The recommender system leverages cosine similarity for aligning user profiles with content.

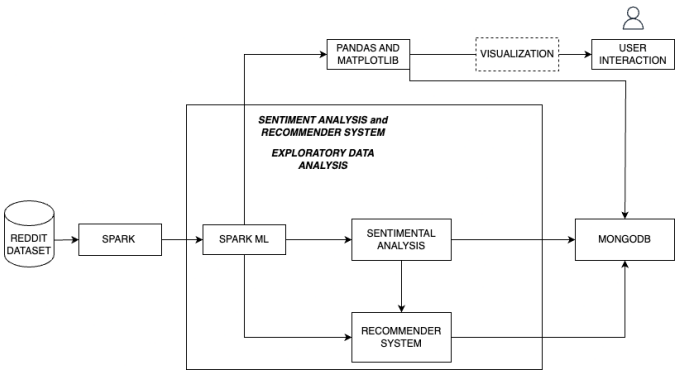

Fig. 1: Architecture diagram

IV. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical step in the data science process, focused on exploring and understanding the characteristics of data. It involves summarizing main characteristics, often with visual methods. In this section, we will discuss the Exploratory Data Analysis conducted on our reddit dataset focussing on count of posts in each subreddit and top authors based on the number of posts they have. We also display word clouds for the 4 top subreddits to analyse the most commonly used words.

A. Count of posts in each subreddit

In the first step of our EDA, we identified the top subreddits based on the number of posts in them (As shown in Fig. 2). This step involves tallying the total posts in individual subreddits, providing a quantitative measure of activity and popularity. This analysis helps in identifying which communities are most active or have the most engagement on Reddit. It can be particularly useful for spotting emerging trends, popular topics, or niche areas with highly engaged user bases. When we analysed the data, we found out that AskReddit, relationships, leagueoflegends, tifu and relationship_advice were the top 5 subreddits based on the number of posts in them.

B. Average post length of posts in each subreddit

This step involves analysing textual data to understand the nature of discussions within different Reddit communities. This analysis helps in identifying the depth and detail of conversations in various subreddits (As shown in Fig. 3). Longer posts might indicate more detailed discussions or complex subjects, whereas shorter posts could suggest quick queries or less elaborate topics. This metric can reveal differences in user engagement and content types across subreddits, offering

subreddit	count
AskReddit	589947
relationships	352049
leagueoflegends	109307
tifu	52219
relationship_advice	50416
trees	47286
gaming	43851
atheism	43268
AdviceAnimals	40783
funny	40171
politics	36518
pics	35098
sex	28806
WTF	25781
explainlikeimfive	25482
todayilearned	25004
Fitness	22694
IAmA	22689
worldnews	22577
DotA2	22405

only showing top 20 rows

Fig. 2: Subreddit posts count

insights into the dynamics of each community. This step is valuable for understanding user behaviour and tailoring content strategies accordingly.

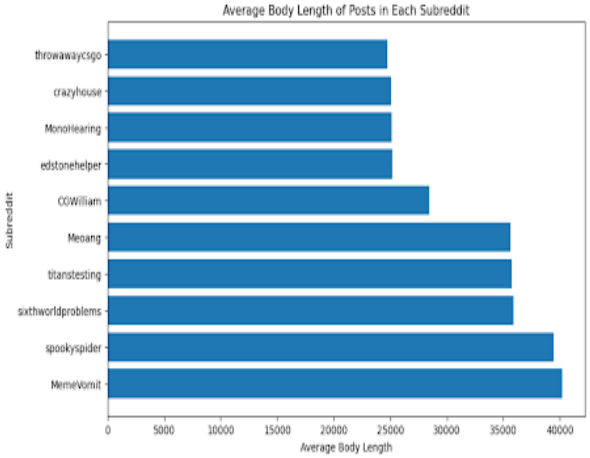

Fig. 3: Average posts length across subreddits

C. Top 10 Authors by Number of Posts

This step involves ranking authors in a dataset based on the total number of posts they have made. It aims to identify the top 10 authors who are the most active in terms of posting frequency. This analysis is significant for understanding who the key contributors are in a community, which can be insightful for recognizing influential users or content creators. It's a useful metric for understanding user engagement and the distribution of content creation within the dataset.

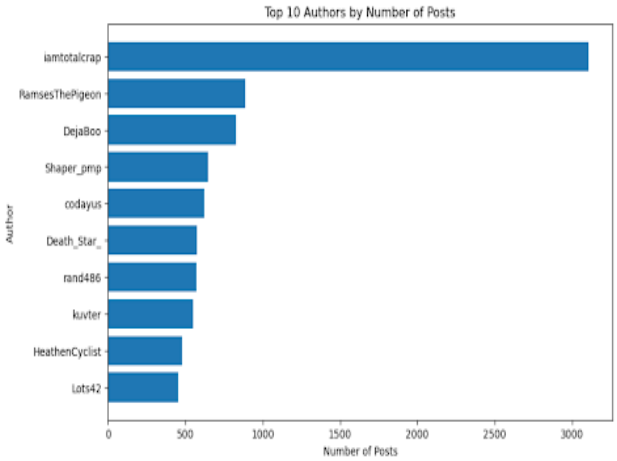

Fig. 4: Top authors by number of posts

D. Word cloud of subreddits

Creating a word cloud of subreddits in EDA helps to visually identify the most prominent communities in the dataset. Larger font sizes for certain subreddit names indicate higher frequencies or significance. This visual representation aids in quickly grasping the distribution of topics and identifying dominant themes or areas of interest within the Reddit data. It's a useful tool for getting an intuitive sense of the dataset's focus areas, guiding further detailed analysis and hypothesis formulation.

Fig. 5: Word cloud of subreddits

V. Sentiment Analysis

A. TextBlob

Sentiment analysis using TextBlob involves evaluating the emotional tone of text data. TextBlob, a straightforward natural language processing (NLP) tool, assesses text and assigns a polarity score that indicates whether the content is positive, negative, or neutral. This method is efficient for quickly gauging the overall sentiment of large volumes of text.

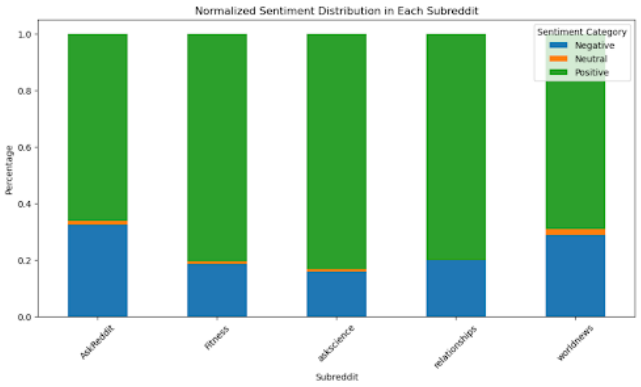

Fig. 6: Sentiment analysis - TextBlob

B. BERT

Sentiment analysis using DistilBERT-base-uncased-finetuned-sst-2-english involves a more advanced approach, leveraging a distilled version of the BERT model that is fine-tuned for sentiment analysis. This method efficiently processes text to understand nuanced emotional contexts, providing accurate sentiment classification into positive or negative categories. DistilBERT's refined model offers high accuracy while being less resource-intensive than its full-sized counterparts.

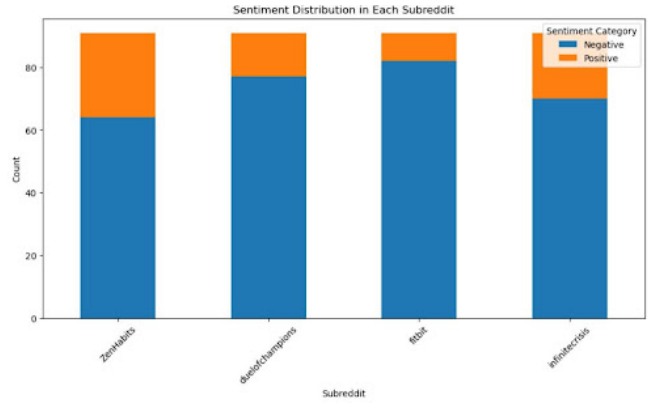

Fig. 7: Sentiment analysis - BERT

VI. Recommendation System

The Subreddit Recommendation System is developed on the robust framework of Apache Spark, renowned for its proficiency in large-scale data management, this system proficiently processes an expansive dataset comprising approximately 3.8 million subreddit entries. Its capability lies in how it turns this complex data into tailored recommendations.

At the core of this system is the implementation of cosine similarity. This technique gauges the alignment between a user's specific interests and the broad spectrum of available subreddits. This technique is pivotal in accurately identifying those entries that most closely resonate with the user's preferences.

A. Data Loading and Preprocessing

Utilizing Apache Spark's data processing power, the system efficiently handles a large dataset of approximately 3.8 million subreddit entries. This stage involves cleaning and standardizing the data by removing special characters, handling missing values, and ensuring consistency.

B. Text Processing and Vectorization

The system processes subreddit summaries by tokenizing the text into words, using Spark's Tokenizer and RegexTokenizer. Common words with low analytical value are removed via Spark's StopWordsRemover. The tokenized words are then vectorized using the TF-IDF algorithm, enabling the identification of unique and significant words in the summaries, crucial for generating precise recommendations.

C. Generating Recommendations through Cosine Similarity

The core of the recommendation logic involves calculating the cosine similarity between the vectorized user input and subreddit vectors. This metric quantifies the similarity between the user's interests and subreddit content, identifying the top 10 most relevant subreddits for personalized recommendations.

The cosine similarity, used to measure the similarity between user preferences and subreddit entries, is calculated as follows:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|}$$

A is the vectorized user input, and B is the vector representation of a subreddit entry. This similarity measure effectively quantifies the closeness of the user's interests to each subreddit, using vectors to capture and compare textual nuances. The system later identifies the top 10 subreddit entries with the highest cosine similarity scores to the user's input, ensuring the recommendations are personalized and contextually relevant.

D. Leveraging Spark and Its Libraries

The system's effectiveness is attributed to Apache Spark and its extensive libraries, which provide robust support for large-scale data processing, machine learning, and statistical analysis. Spark's scalability ensures the system's adaptability to growing data volumes and changing analytical needs, solidifying its role as an integral platform for this complex recommendation task.

E. Bottlenecks in the current system

The Subreddit Recommendation System, while efficiently processing a dataset of approximately 3.8 million entries using Apache Spark, encounters challenges in handling the sheer volume and complexity of data during preprocessing, tokenization, and vectorization. These intensive operations can impact processing speed and system efficiency, necessitating optimization for timely recommendations. Additionally, the system's reliance on cosine similarity for recommendations, despite its effectiveness in identifying textual similarities, has its limitations. It may not fully capture the evolving and nuanced interests of users, as it depends on static text data and

overlooks dynamic user behaviors and contextual variations. This can result in recommendations that are textually aligned but may not align perfectly with the user's current preferences.

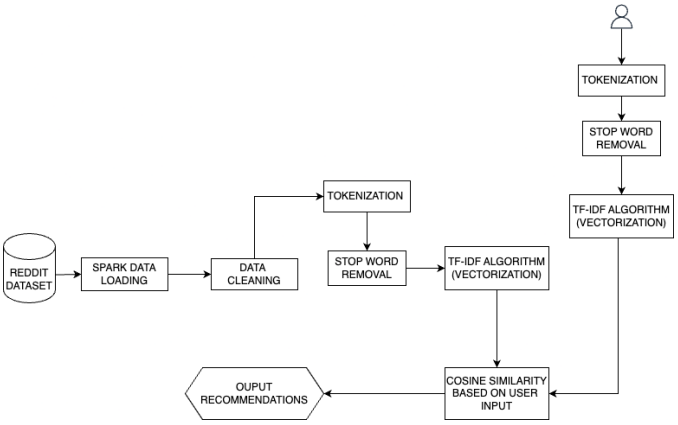

Fig. 8: Architecture - Recommendation System

VII. Results

In the analysis conducted by the recommender system, the queries "LeBron James" and "Taylor Swift" yielded distinct subreddit recommendations based on cosine similarity metrics exceeding a threshold of 0.7 (as shown in Fig. 9 and Fig. 10). For "LeBron James," the subreddits with the highest relevance included "nba," "sports," and "clevelandcavs," with frequency counts reflecting a strong association with sports and the athlete's professional and regional relevance. In contrast, the query "Taylor Swift" revealed a broader range of subreddit recommendations, with "AskReddit" showing the highest occurrence, followed by her eponymous subreddit, indicating a wider general interest as well as dedicated fan engagement. The results from both queries demonstrate the recommender system's capability to identify and aggregate thematically related digital communities based on user-generated content on the Reddit platform.

subreddit	count
AskReddit	15
TaylorSwift	3
AdviceAnimals	1
GetMotivated	1
Fitness	1

(a) Posts in the resulting subreddit for "Taylor Swift"

subreddit	cosine_similarity
AskReddit	1.0
DoesAnybodyElse	0.9631061
gaming	0.89372706
TaylorSwift	0.87298506
AskReddit	0.8532602
AskReddit	0.8381791

(b) Recommended subreddits for "Taylor Swift"

Fig. 9: Results for "Taylor Swift"

VIII. Limitations

A. Challenges with Static Dataset

The system's dependence on a static dataset poses a significant challenge, especially for the subreddit recommendation

subreddit	count
AskReddit	15
TaylorSwift	3
AdviceAnimals	1
GetMotivated	1
Fitness	1

(a) Posts in the resulting subreddit for "Lebron James"

subreddit	cosine_similarity
AskReddit	1.0
DoesAnybodyElse	0.9631061
gaming	0.89372706
TaylorSwift	0.87298506
AskReddit	0.8532602
AskReddit	0.8381791

(b) Recommended subreddits for "Lebron James"

Fig. 10: Results for "Lebron James"

component. As Reddit is a dynamic and constantly evolving platform, the static nature of the dataset may not accurately capture the most recent trends, new subreddit creations, or shifts in user interests. This limitation could result in recommendations that are not entirely aligned with the latest Reddit content and discussions.

B. Limitations in Personalization and Complexity

While the sentiment analysis using BERT effectively captures complex user emotions, the recommendation system does not currently utilize personalized user profiling. This absence of personalization means that recommendations are generated based solely on a user's current input, without considering their past interactions or preferences on Reddit, potentially impacting the relevance and accuracy of the recommendations.

IX. Future work and improvements

A. Incorporating Real-Time Data and User Profiling in Recommendations

Future enhancements should include integrating real-time data updates in the subreddit recommendation system. This would ensure that the recommendations stay current with new and trending content on Reddit. Additionally, incorporating individual user profiles and historical interaction data could significantly refine the recommendation process, enabling more targeted and relevant subreddit suggestions that align with each user's unique interests and browsing history.

B. Exploring Advanced Techniques for Enhanced Personalization

Building on the success of BERT in sentiment analysis, future work could explore advanced machine learning and deep learning models to further enhance personalization in the recommendation system. Techniques like collaborative filtering, user behavior analysis, and context-aware recommendation algorithms could be investigated to provide a more nuanced and customized user experience.

X. Acknowledgements

We would like to express our sincere gratitude to Professor Juan Rodriguez for his invaluable feedback throughout the course.

XI. Conclusion

In conclusion, our exploratory data analysis of the TensorFlow Reddit dataset has unveiled significant patterns and insights, particularly in user engagement across various subreddits. We discovered key trends in post lengths, popular topics, and the most active authors, which offer a deeper understanding of community dynamics. Notably, our analysis using big data tools like Pyspark, SparkML, and MongoDB efficiently managed the large dataset, allowing for nuanced sentiment analysis and content recommendations. This project not only enhances the Reddit user experience but also provides valuable insights into social media interactions, demonstrating the power of advanced analytics in understanding online communities.

Our exploratory data analysis of the TensorFlow Reddit dataset, utilizing Pyspark, SparkML, and MongoDB, has provided extensive insights into the intricacies of Reddit communities. We conducted sentiment analysis to understand the emotional tone within various subreddits, revealing the prevailing moods and perspectives of users. This analysis was instrumental in tailoring a recommender system that suggests relevant content to users, enhancing engagement and personalization. The combination of sentiment analysis and the recommender system, powered by advanced analytics, has significantly contributed to a deeper understanding of user interactions and preferences on Reddit. These insights are invaluable for community moderators and marketers in strategizing content and engagement approaches. Overall, our project not only enhanced the Reddit experience but also showcased the transformative impact of big data technologies in analysing and interpreting vast, unstructured social media datasets.

References

[1] https://spark.apache.org/
[2] https://spark.apache.org/docs/1.2.2/ml-guide.html
[3] https://www.mongodb.com/
[4] https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english