**Report (10 points)**

The report should have at least 2500 words with the following structure:

Project Title: Come up with an attractive project title (see this [page](https://blog.hubspot.com/marketing/a-simple-formula-for-writing-kick-ass-titles-ht#sm.0000072w4nzrqmfn2vwd6y30afmyo) for some tips);

**Building a Popular Tweet: Tag Prediction and Sentiment Analysis**

**(option 2) TagTrends: AI-Powered Tweet Tag Prediction and Sentiment Analysis**

Motivation and Background: Who cares about this project? Any related work?

**Motivation and Background**

Social media platforms like Twitter have become one of the most popular sources of information and communication, especially among young people. Millions of tweets are posted every day on various topics ranging from news, politics, sports, entertainment, and more. These tweets contain a wealth of information, including people's opinions, thoughts, and sentiments about various topics.

Understanding the sentiment of tweets is essential for companies and individuals to make informed decisions. For instance, companies can use sentiment analysis to evaluate customer feedback on their products and services. They can also use it to monitor the sentiment of their brand and their competitors.

Similarly, tagging tweets with relevant topics can help users find relevant content and discover new information. For instance, if someone is interested in sports, they can use tags like #football, #basketball, or #tennis to search for tweets related to those topics. These tags can also help tweets become “hot” or trending on social media, increasing their visibility and reach.

However, analyzing the sentiment of tweets and predicting relevant tags is challenging due to various reasons. Firstly, tweets are often short, unstructured, and contain slang, abbreviations, and emoticons, making it difficult to understand the context and the sentiment. Secondly, tweets often contain noise and irrelevant information, making it challenging to predict the relevant tags accurately.

Therefore, our project aims to address these challenges by developing a system that can perform sentiment analysis and tag prediction on tweets accurately. We believe that this project can provide valuable insights to companies, individuals, and researchers by allowing them to make informed decisions and discover new information.

Related Work:

There has been extensive research on sentiment analysis and tag prediction for tweets. Numerous studies have employed machine learning algorithms, such as Support Vector Machines (SVM), Naive Bayes, and Decision Trees, for sentiment analysis tasks. These approaches have focused on classifying tweets as positive, negative, or neutral based on their textual content.

In the domain of tag prediction, various methods have been explored, including Latent Dirichlet Allocation (LDA), k-Nearest Neighbors (k-NN), and neural network-based approaches like Recurrent Neural Networks (RNN) and Transformer models. Researchers have sought to predict relevant tags by analyzing the semantic and syntactic information present in tweets.

Our work builds upon these existing methods by combining state-of-the-art neural network models for both tag prediction, focusing on hot trending topics. By leveraging the capabilities of the advanced model, we aim to provide a more comprehensive understanding of the sentiment and tags associated with popular discussions on social media platforms like Twitter.

Problem Statement: What questions do you want to answer? Why are they challenging?

**Problem Statement**

The primary objective of our project is to develop a comprehensive solution that addresses the following questions:

1. Can we present the sentiment analysis data (such as retweet, like, and view metrics) in a visually appealing and easily interpretable format to provide content insights for users?
2. How can we accurately predict relevant tags for tweets related to hot trending topics, facilitating better content organization and discoverability for users?
3. How can we effectively analyze the sentiment of tweets, providing insights into trending discussions?

These questions pose several challenges:

1. Developing an efficient scraping process for handling large volumes of tweets to provide a foundation for up-to-date insights on sentiment and tag predictions.
2. Data cleaning. Handling the complex and diverse nature of language used in tweets, including informal expressions, abbreviations, and slang, can be a major obstacle for both tag prediction and sentiment analysis.
3. The dynamic nature of trending topics requires a robust and adaptable neural network model that can learn and generalize effectively from data.
4. Ensuring that the visualization of sentiment analysis data is both informative and engaging for users, while maintaining clarity and simplicity in presentation.

Data Science Pipeline: What's your data-science pipeline like? Describe each component in detail.

Data Science Pipeline:

Our data science pipeline consists of the following components:

Data Collection: To acquire data on hot trending topics, we scraped tweets and their metadata using the python library beautifulsoup.

(describe what topics were scraped, how big are the the datasets)

Data Preprocessing: We reorganized our tweets and metadata, and stored them into .csv files. Then, we preprocessed that data using NLTK to clean the text and extract relevant features.

Sentiment Analysis:

(describe Sentiment Analysis process,)

Regarding predicting the sentiment of COVID-related tweets, we trained a random forest regression model using scikit-learn in Python. Our dataset consisted of approximately 3,500 tweets with its predicted tag from output of tag prediction model and other features including number of comments, number of retweets, number of likes and polarity scores ranging from -1 to 1, where -1 represents extremely negative sentiment and 1 represents extremely positive sentiment.

We preprocessed the text data by tokenizing each tweet using the NLTK library's word\_tokenize function and transforming the resulting tokens into a numerical feature matrix using scikit-learn's CountVectorizer. We then combined the resulting text features with other relevant features such as the number of comments, retweets, views, and timestamp information. After splitting the data into training and testing sets, we fit the random forest model to the training data using 100 decision trees and a random state of 30. We then used the trained model to make predictions on the test data.To predict the sentiment of a single new tweet, we first preprocessed the text in the same way as before. We then extracted relevant features such as the number of comments, retweets, views, and timestamp information, and combined these features with the preprocessed text features into a single feature vector.

We passed this feature vector into our trained random forest model to obtain a predicted compound sentiment score. This score represents the overall sentiment of the tweet, with higher scores indicating more positive sentiment and lower scores indicating more negative sentiment. By using this approach, we were able to accurately predict the sentiment of COVID-related tweets with a mean squared error of 0.17 on our test set.

Tag Prediction:

(describe Tag Prediction process)

Since string data can’t be directly used for training, we conducted tokenization， truncation, and tensorization to make sure our Pytorch model could utilize data in a proper way.

Results?

Tag Prediction: To decode the tensors our model returned, we need to perform the opposite decoding operation to the encoding done during preprocessing, which can give us strings. To make this string more like tags, we split it by commas, strip nonsense and duplicated words, and return a list of tags that is easier to use.

Methodology: What tools or analysis methods did you use? Why did you choose them? How did you apply them to tackling each problem?

**Methodology**

We used Python as our primary programming language due to its flexibility, readability, and extensive support for data science and machine learning tasks. To address the challenges outlined in the Problem Statement and develop an efficient solution, several libraries were employed in the development process, including:

1. Natural Language Toolkit (NLTK): Widely used for Natural Language Processing (NLP) tasks, we utilized NLTK for text preprocessing, including tokenization, stopword removal, and stemming.
2. Scikit-learn: A popular machine learning library, Scikit-learn was employed for data preprocessing, model selection, training, and evaluation.
3. PyTorch
4. Streamlit: An open-source library for creating custom web applications, Streamlit enabled us to develop an interactive user interface for our project.

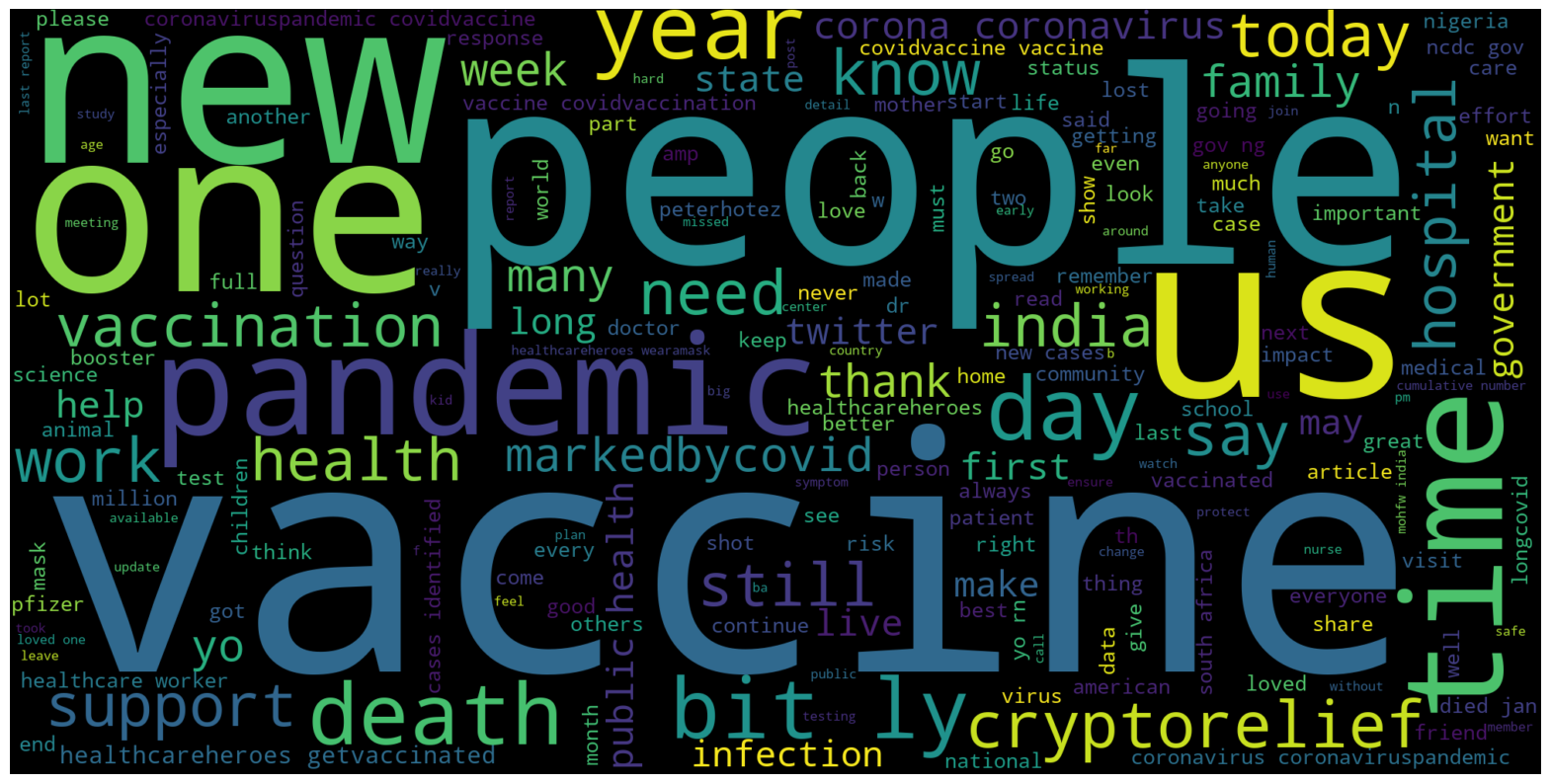
We chose these libraries due to their widespread use in the field of NLP, their ease of integration, and the variety of functionalities they provide for text preprocessing, data visualization, and machine learning. They were utilized in the following processes of the project:

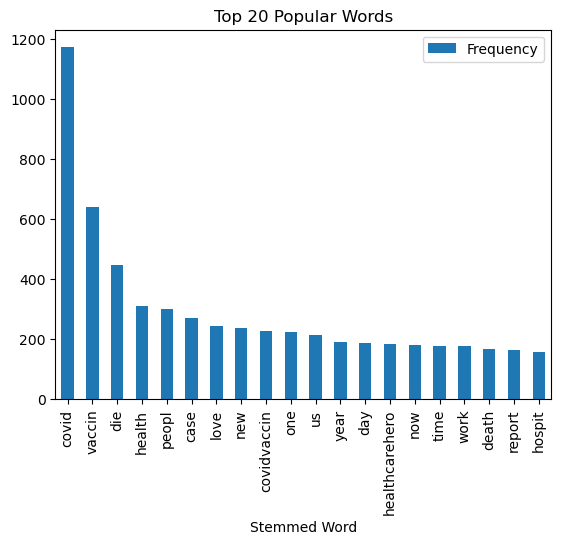
Tag Prediction:

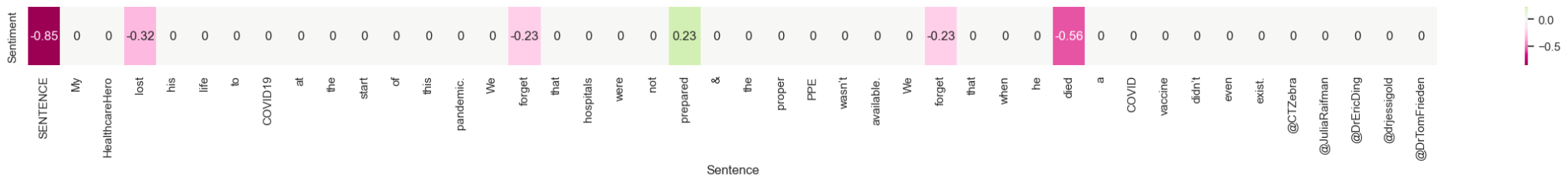
For predicting relevant tags, we trained a neural network model using Python library Scikit-learn. the preprocessed tweet data as input. We experimented with different architectures, such as RNNs, Transformers, and fine-tuned T5 model to determine the most effective approach for our dataset. Once the model was trained, we used it to predict tags for new tweets related to the selected trending topics.

Sentiment Analysis:

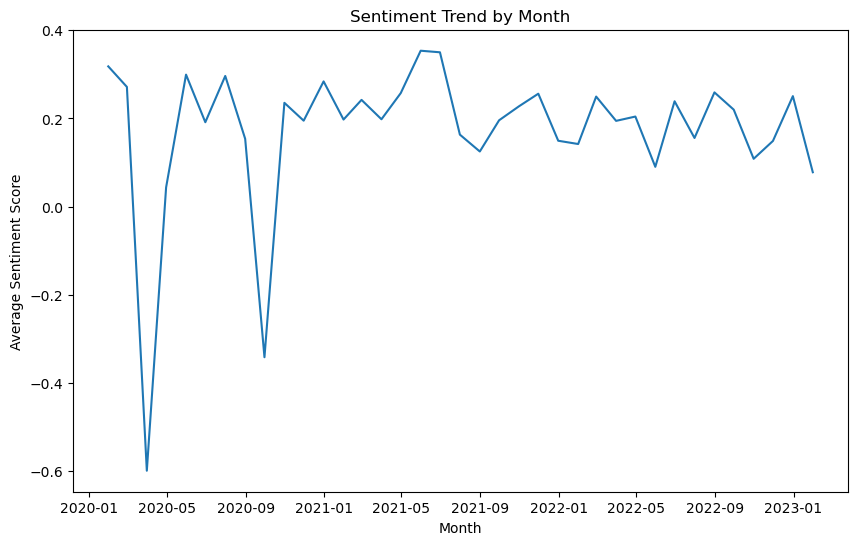
We preprocessed the data using NLTK to clean the text and extract relevant features for analysis.

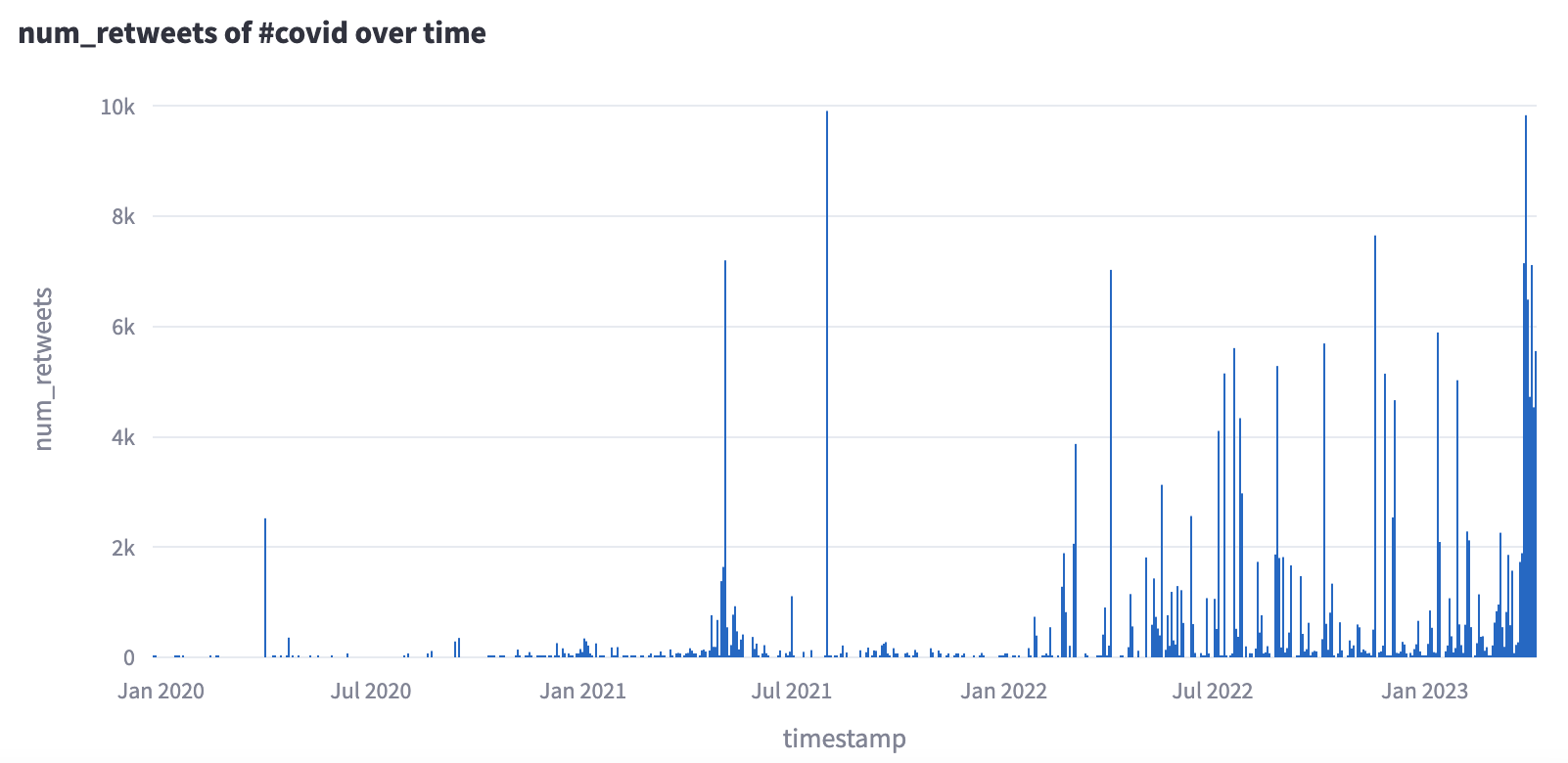






Data Visualization:





To present the sentiment analysis data in an easily interpretable format, we created several dashboards and visualizations using Streamlit. These visualizations included word clouds, bar charts, and line charts displaying the distribution of sentiment scores, retweet, view, and comment metrics over time.

Evaluation: Why is your solution good? Why does your result make sense?

**Evaluation**

To evaluate the performance of our sentiment analysis and tag prediction models, we will use the following metrics:

1. Sentiment Analysis: We used the following metrics to evaluate the performance of our sentiment analysis model: Mean Squared Error (MSE): This metric measures the average squared difference between the predicted sentiment scores and the actual sentiment scores.

Data Product: What's your data product? Please demonstrate how it works.

**Data Product**

Our data product is an intuitive web application designed to analyze the sentiment and predict tags for tweets associated with the user-specified topic and timeline. Upon receiving a tag as input, the application scrapes tweets related to the selected tag and performs sentiment analysis and tag prediction on the gathered data. The results are then presented in multiple interactive dashboards, providing users with a visually appealing and informative overview of the analyzed data.

For example, one of the dashboards displays various metrics, including the distribution of comments, retweets, and views, enabling users to gain insights into the popularity and engagement of each tag. To facilitate a tailored experience, users can select from our curated list of "Top 5 Hot Tags" and specify their desired timeline for analysis.

For sentiment analysis visualization, users could see the sentiment trend by month, the average sentiment score of each word with selected tweets. Users could choose the tweets text and also tag for a better understanding.

Also, we generated the top 20 most common words and word clouds for each tag, from there, users will get an intuitive thought about how to write their tweets.

By offering a user-friendly interface and customizable features, our data product empowers users to explore the sentiment and tags associated with trending topics on social media, ultimately enhancing their understanding of the current social media landscape.

Lessons Learnt: What did you learn from this project?

**Lessons Learnt**

1. Data preprocessing is crucial: The quality of the data and how it is cleaned, preprocessed, and transformed have a significant impact on the performance of the machine learning models. It is essential to carefully examine the data and handle missing values, outliers, and noise to avoid bias and improve accuracy.
2. Model selection and tuning matter: There are numerous machine learning algorithms, and selecting the right one(s) for the task at hand can be challenging. Moreover, hyperparameter tuning can significantly affect the performance of the models, and it requires careful tuning and validation to avoid overfitting and ensure generalization.
3. Interpretability is essential: As machine learning models become more complex and sophisticated, it becomes harder to understand their decisions and behavior. Therefore, it is crucial to incorporate interpretability methods that can help explain the models' outputs and enable their users to trust and rely on them.
4. Collaboration and communication are key: Machine learning projects often involve cross-functional teams, including data scientists, engineers, domain experts, and stakeholders. Effective collaboration and communication among team members are crucial to ensure that the project's goals and requirements are met, and the models' outputs are actionable and understandable.

Summary: A high-level summary of your project. It should be self-contained and cover all the important aspects of your project.

**Summary**