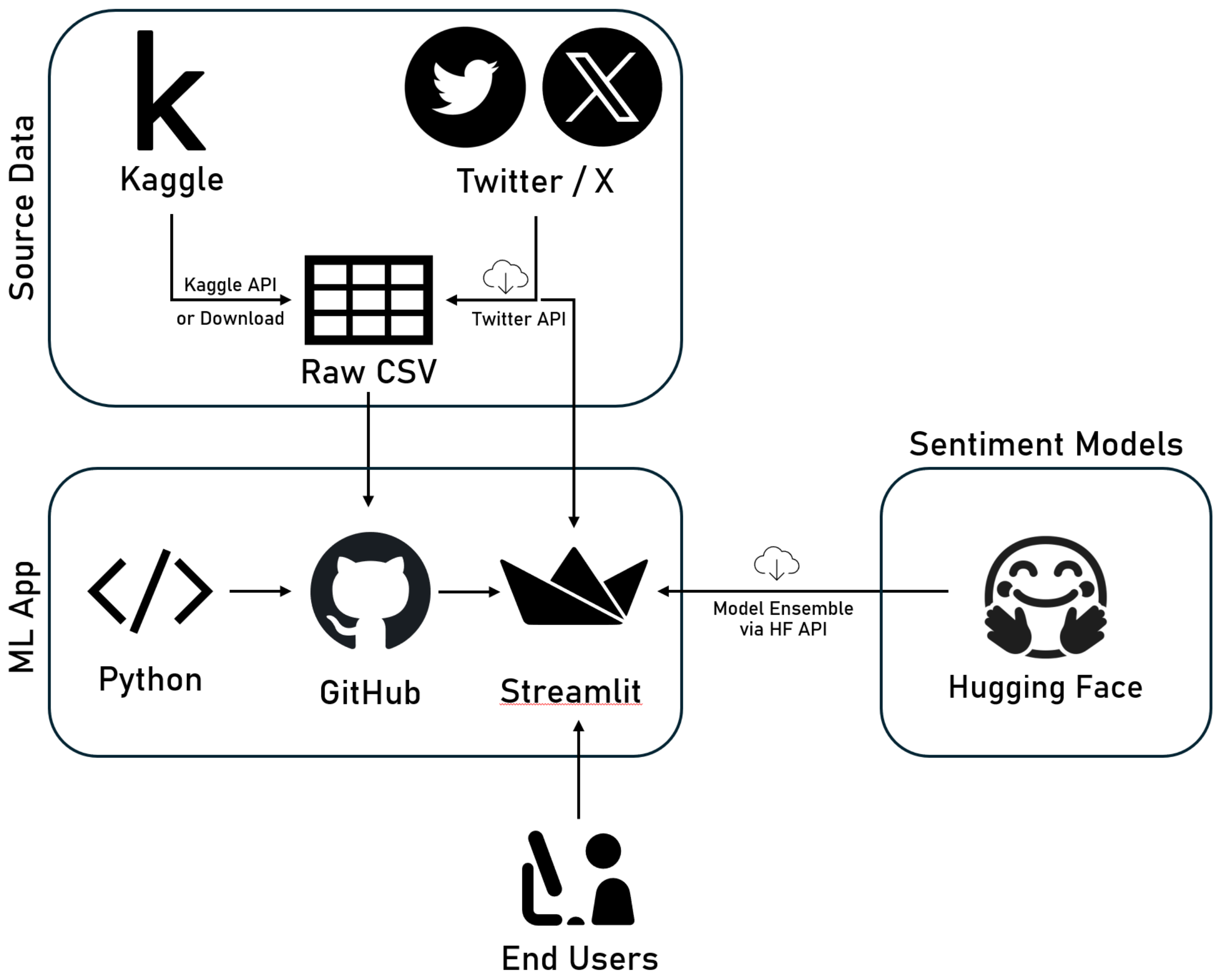
Ilya Shafir
UCSD AI/ML Bootcamp - March 2024 Cohort

33.4.1 Capstone Project Deployment Architecture

31st March 2025

Sentiment Analysis Architecture Overview

Below is the planned architecture for my Twitter Sentiment Analysis project.



Data Sources

Twitter/X data source would initially be static, coming from existing datasets, such as from Kaggle, in CSV format, stored in the same GitHub repo. In future deployments, ideally, would be able to gather recent X posts via the [X API] and daily stream to a CSV update file which could then be processed for analysis. The X API is rather costly at the moment so would look to third party aggregators first. I may expand data sources to Reddit as the methodology is the same.

Data Cleaning

The datasets specifically for this project include Twitter data with users from all over the world. Since I am analyzing tweets about the 2020 US Election candidates to “predict” the likely winner, it makes sense to pair down the data to only US users. While not all users will be eligible to vote, some of the non-voters may influence actual voters. Analyzing non-US user tweets may be of interest as a form of sentiment comparison but not in scope for the final project itself, especially since reducing the dataset down significantly from ~1.7 million tweets for performance and compute cost reasons is critically important.

Machine Learning App

This has been first developed on a desktop computer in Python and saved on my GitHub repos for initial prototyping and then public deployment. GitHub provides the updated repo to Streamlit, my chosen deployment platform. Once any updates to the repo are committed, Streamlit immediately picks up the changes and updates the app in place. This makes the development platform entirely flexible and can be performed from any device.

Interface

Streamlit Community provides a free, cloud-based compute and UI development for my ML App. GitHub interfaces directly with Streamlit. Here is a screenshot of the app that provides a simple, guided workflow for the end user:

Twitter/X Sentiment Analysis for Two Election Candidates

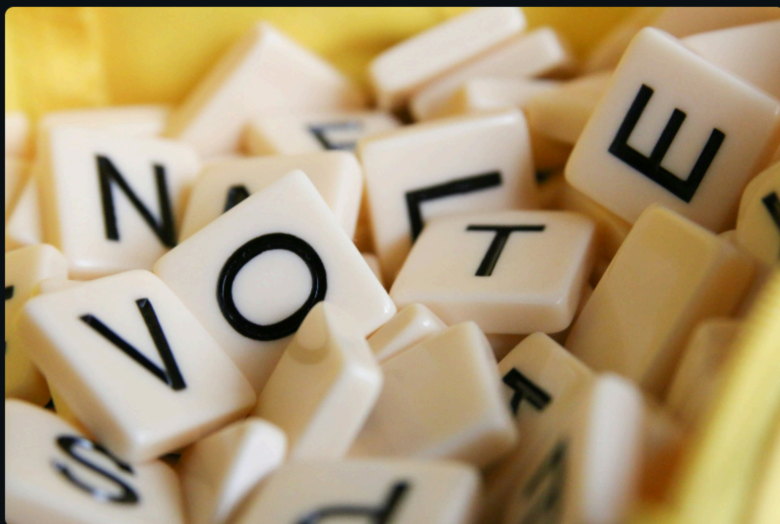


Image licensed with Unsplash

Can we predict the likely outcome of elections by performing sentiment analysis on the tweets/X posts about each candidate?

Use the sample data from 2020 US Elections or upload two CSV files—one per candidate. The app merges tweets from both candidates, cleans them, and sends batches to the Hugging Face Inference API for sentiment analysis. Results are ensembled across the RoBERTa, DistilBERT and SieBERT inference models.

Note: Each CSV must include a `tweet` column.

Step 1: Choose Data Source

Choose data source:

- (●) Use included sample data
- ( ) Upload your own CSV files

Load Sample Data

Sample data loaded.

Sampled Data Preview

	cleaned_tweets	candidate
81627	thank you for JoeBiden from Austin Texas	biden
30036	Photo of Joe Biden not wearing a mask on a plane is from , fact-checkers report VoteBlueTo	biden
9415	JoeBiden Blue collar, authored Violence Against Women Act (VAWA), pro LGBTQ gay marriag	biden
66044	YOU ALREADY WON WHATS THE HOLD UP DonaldTrump PresidentialElection	trump
46598	Looks like this is BCE - before the Covid era. Yet another example of how trump fell short of l	trump
14177	You got it Uncle JoeBiden! LordHearOurPrayers	biden
12395	Youre not helping your dad with your nasty Tweets. Have you learned nothing in your privil	biden
75502	Watch Live: President Trump Holds MAGA Rally in Nevada Sunday //: President Donald Tru	trump
26146	And Biden received ,, and won. BidenHarris JoeBidenKamalaHarris TrumpTantrum lost the	biden
32140	Lo que debes saber antes de que acabe el de octubre de : Un ltimo debate intenso entre Tr	biden

Step 2: Choose Sample Size

Limitations and Alternate Deployments

The free Streamlit Community Cloud App is resource limited, unfortunately:

- **CPU:** 0.078 cores minimum, 2 cores maximum
- **Memory:** 690MB minimum, 2.7GBs maximum
- **Storage**: No minimum, 50GB maximum

One can apply for a temporary resource upgrade if it's for a "good-for-the-world" use case, such as education.

For a production deployment it's possible to run Streamlit using varied cloud providers. This will certainly reduce resource limitations and improve performance, but will incur significant costs due to additional compute resources, especially as the number of user requests increase. Of course, saving analysis results will help, but not if the user brings in their own data.

Sentiment Analysis Ensemble

For the first iteration of the ML App, I'll be using three existing Hugging Face models to create an ensemble that evaluates the final sentiment of each tweet. The models in the ensemble are all derivatives of the original BERT model, which performs Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

- **roBERTa** - This model was specifically trained on ~124M tweets from January 2018 to December 2021, and fine tuned for sentiment analysis with the TweetEval benchmark. It's the only model out of the three that includes not only positive and negative sentiments, but also neutral.
- **DistilBERT** - This is a large english model which is a cousin twice-removed from the original BERT, but "distilled" for faster performance.
- **sieBERT** - SieBERT is a fine-tuned checkpoint based on the roBERTa large model which itself was pretrained on a large corpus of English raw text in a self-supervised fashion (i.e., no human labeling).

Ensemble Execution – Best of 3

The App will interface with the Hugging Face Inference API to load all the models above. Each tweet will be processed by all three models. It will drop all tweets that are judged "neutral" by roBERTa so we are left with a binary result of either "positive" or "negative" sentiment. Because the results are binary and we have three models, then at least two out of three models must agree on the sentiment. This majority agreement will be our final sentiment of each tweet.

Additionally it will store the confidence score of each tweet, where we can filter further the results based on this confidence. The final confidence score is the average of the two or three models that agree on the sentiment.

Finally, to evaluate the accuracy and correlation of the models, it is of interest to keep track of which two or three models won the majority vote for each sentiment so we will store these votes per tweet as well.

Ensemble Improvements

It is worth researching and evaluating newer sentiment models and/or sentiment models trained on more recent data to consider replacing any of the three models above to improve the accuracy of the sentiment results. Replacing the models requires minimal code changes when using the HF API.

Another improvement would be to provide the end user with a choice of models to generate the ensemble and make the analysis more dynamic that way. The Streamlit UI makes this easy, and again, code changes would be minimal as long as the model choices given have been tested with the App prior to deployment.

App System Monitoring

System monitoring for this Streamlit Community edition App is rudimentary and mostly manual right now. The only analytics it provides is the number of unique visitors to your App. There is a python plugin to send push alerts from Streamlit but that's only if you're in the same browser as your Streamlit app. Finally, we have the poor man's method of observing the Streamlit app console to mitigate issues. At this point, we'd likely need to move this to a hosted Streamlit App on Snowflake to get advanced monitoring capabilities.

End Goal

With the major controversies surrounding, not just the recent US elections, but elections and political situations worldwide, performing sentiment analysis across social media and various mainstream media outlets would be invaluable for research, especially understanding political polarization.

Election polls only provide a very small sample view of people's sentiment. Using ML sentiment analysis offers millions, if not billions more data points to reference. Clearly data filtering and cleansing is important prior to any sentiment analysis (e.g. identifying and removing AI/bot generated content, irrelevant content, etc.). Then choosing the right sentiment models for accuracy, and those trained on recent Twitter feeds would be better.

My goal is to keep improving and scaling this app for anyone to gain a better understanding of what (real) people's sentiments are towards high-profile political figures.