MLTox: A CLOUD-BASED MACHINE LEARNING TOXICITY DETECTOR

Cloud Computing
Spring, 2019
Srihari Mohan, Lalit Varada, Benjamin Pikus, and Parth Singh

1. Introduction

- Social media has seemingly brought an increase in toxic comments
- An interesting problem, both philosophically and technically, is to in real-time assess the toxicity of a certain topic on social media
- Fortunately, this problem has a lot of data available - Twitter is responsible for generating upwards of 500,000,000 unique messages per day. The key then is to build a scalable platform that can efficiently process this large data stream
- Existing approaches focus exclusively on detecting toxicity on a message-by-message basis
- MLTox is a cloud-hosted machine learning toxicity detector that
  - 1) Determines the toxicity of a topic on Twitter in real-time
  - 2) Perform anomaly detection on toxicity post hoc
  - 3) Forecast the toxicity trends to predict how they will change
- MLTox provides a user-facing dashboard with a post-hoc toxicity trajectory with labeled anomalies and a forecasted trajectory on politics-related tweets in the United States.

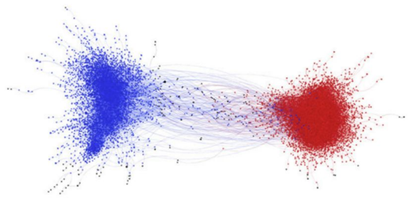

Fig. 1. Network graph of moral contagion shaded by political ideology. The graph represents a depiction of messages containing moral and emotional language, and their retweet activity, across all political topics

2. MLTox Architecture

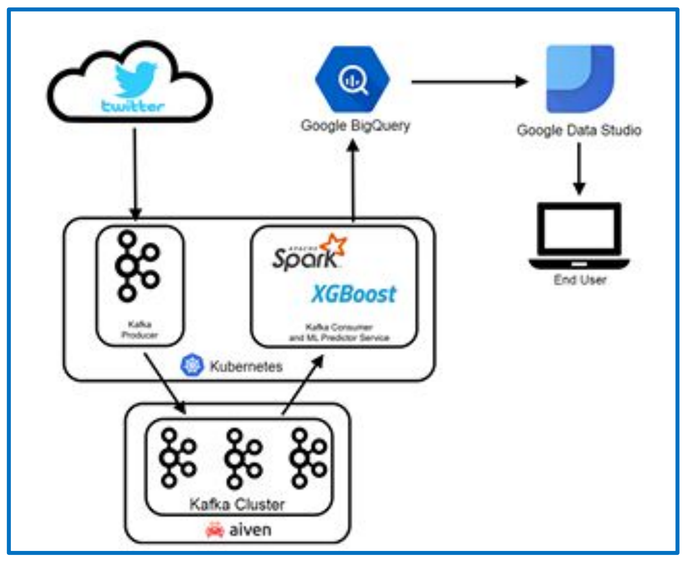

Fig. 2. MLTox streams upwards of 100,000 tweets per day through a Kafka cluster hosted on Aiven. A microservice for the Kafka producer and Kafka predictor engine are spun up as containers run on Kubernetes. The producer microservicerepeatedly fetches batches of politics-related tweets every 100s using the tweepy API, before publishing them to brokers in the Kafka cluster. The predictor microservice hosts our XGBoost ML predictor, seasonal autoregressive SSMs time-series forecaster, and LOCI anomaly detector. The microservices are hosted on Kubernetes and write theiroutput in real-time to Google BigQuery. We finally connect BigQuery to Google Data Studio to generate a user-facing and continually refreshed public dashboard of MLTox's learned toxicity trajectory, anomaly detection, and toxicity forecasting.

- **Apache Kafka**
  - Publish-subscribe messaging platform that delivers high volumes of data with low latency
  - Has 3 fundamental parts: *producer* - a client that publishes messages to a topic, *consumer* - subscriber that pulls the data from the Kafka server(s), and *brokers* - set of Kafka servers that is responsible for delivering the data to the consumer.
  - Each push/pull request is grouped by topic
  - For MLTOX - producer and consumer are microservices spun up in Docker containers on Google Kubernetes Engine (GKE)
    - producer repeatedly fetches batches of tweets every 100 seconds and pushes to brokers in MLTox's Kafka cluster hosted on Aiven's Kafka as a Service platform
    - consumers pulls the messages and streams them to a mounted disk in the Kubernetes pod on GKE

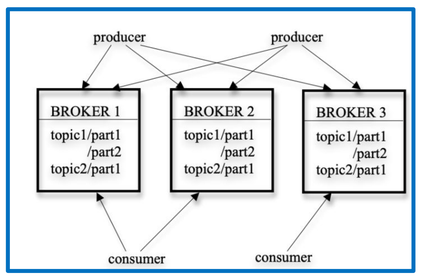

Fig. 3. Kafka architecture

- **Apache Spark**
  - Spark is a cluster computing framework that supports processing for working set of data across multiple parallel operations (based on MapReduce)
  - Scalable and fault tolerant using resilient distributed datasets (RDDs)
  - Spark's MLLib provides high-level tools for ML algorithm development, feature extraction, pipelining, and model persistence using RDDs.
  - For MLTox - model is learned offline and deployed on Kubernetes
    - use Spark's distributed system to batch processing on incoming tweets
- **Kubernetes**
  - container orchestration and managing automated application deployment that separates each service of the application with focus on high scalability
  - 20 replicas of our predictor engine are deployed onto pods on GKE
- **Google BigQuery and Google Data Studio**
  - enterprise data warehouse that allows fast interactive queries over large datasets.
  - We write toxicity predictions, detected anomalies, and forecasted trajectories into BigQuery (100,000+ records per day)
  - Google Data Studio generates visualizations of the toxicity trajectory, forecasting, and anomaly detection

4. Machine Learning

MLTox uses machine learning to provide learned toxicity trajectories, toxicity forecasts, and anomaly detection.

Toxicity Trajectory

- Input is preprocessed tweet text - output is continuous toxicity score
- Use XGBoost (gradient boosting decision tree algorithm where weak learners are combined to form a stronger model)
- Ground truth data comes from Google's Perspective API
- On a held-out test set of tweets - RMSE= 16.9 (standard deviation of learned score = 19.65).

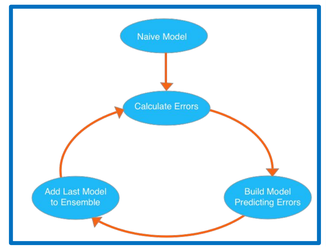

Fig. 4. High level description of iterative XGBoost learning process

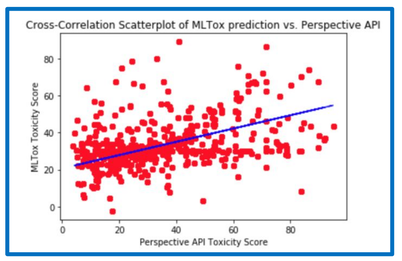

Fig. 5. Cross-Correlation between MLTox and Perspective API - correlation value r = 0.53

Anomaly Detection

- Goal: detect abnormal fluctuations in toxicity given the current trend
- Input is aggregated toxicity scores for a batch of tweets - output is whether an anomaly has been detected
- Use Local Correlation Integral (density based approach for outlier analysis) that calls something an anomaly if deviation factor is 3 times larger than the standard deviation of all toxicity scores in the batch

Toxicity Forecasting

- Goal: given past toxicity scores, predict future scores
- Use Seasonal Autoregressive state space models (each future toxicity score is connected to a hidden state and depends linearly on previous values and stochastic term)
- Forecasting 30 hours forward based off of 3 days of data gets a mean absolute percentage error of 7.5%

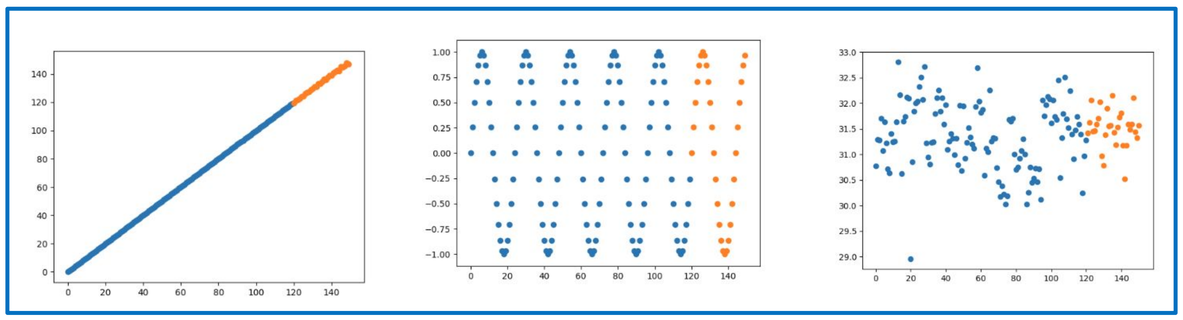

Fig. 6. Sample forecast results. The first two plots are run on simulate data, and the third is on actual data. In each plot, the blue trajectory represents the observed history, while the orange represents our forecasted trajectory. On the simulated plots, the forecasting is near perfect, with the forecaster capturing the dominant trend. On the actual data, the forecast appears to be in line with our expectations, bounded between 30 and 32 like most of the example points and exhibiting no clear trend behavior. To optimize the parameters for our SSMs, we performed a grid search on the actual data.

5. Discussion

- Have user-facing, continually refreshed public dashboard that displays the MLTox learned toxicity trajectory alongside the forecasted toxicity trajectory
- Compared to existing solutions that look at account-data, MLTox permits a greater number of tweets and discussion threads to be analyzed by looking at topics while forecasting the quality of future discussion
- *Message volume:* 368,450 tweets streamed in 4 days
  - approximately 1700 tweets per hour
  - at its peak, stream over 100,000 tweets per day through predictor service
- ML-Tox was effective in blending cloud technologies at every layer of the application stack to deliver a novel, scalable system with wide applications

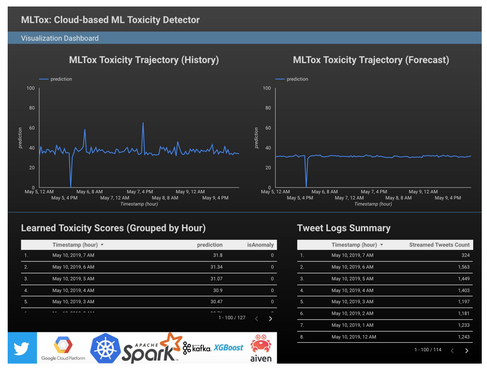

Fig. 7. MLTox Public Visualization Dashboard. Included are summary tables that mark whether a particular hour of online discussion was marked anomalous, as well as summary statistics indicating the volume of tweets streamed

6. Future Work

- Further work could be done to allow for the user to interact with the platform rather than just looking at it
  - ex: add comparison of multiple topics rather than looking at a singular topic, or letting the user choose the topic
- Also currently only streams from Twitter - could stream from Facebook, Reddit, etc
- Allow for online training so the system learns and evolves over time to get better at predicting toxicity score

7. References

[1]Jay Kreps, Neha Narkhede, and Jun Rao. Kafka: a distributed messaging system for log processing. NetDB '11,pages 1–7. ACM, Jun 12, 2011.
[2]Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computingwith working sets. page 10–10, Berkeley, CA, USA, 2010. USENIX Association.
[3]Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitterdata. New York, NY, USA. Department of Computer Science, Columbia University.
[4]Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-tasklearning. pages 2–2.
[5]Aditya Timmaraju and Vikesh Khanna. Sentiment analysis on movie reviews using recursive and recurrent neural network architectures. pages 2–2.
[6] Lauren Oyler. When did everything get so 'toxic'? New York Times, Oct. 2, 2018.
[7]William J. Brady, Julian A. Wills, John T. Jost, Joshua A. Tucker, and Jay J. Van Bavel. Emotion shapes the diffusion of moralized content in social networks. PNAS, July 11, 2017.
[8] Tianqi Chen and Carlos Guestrin. Xgboost. KDD '16, pages 785–794. ACM, Aug 13, 2016