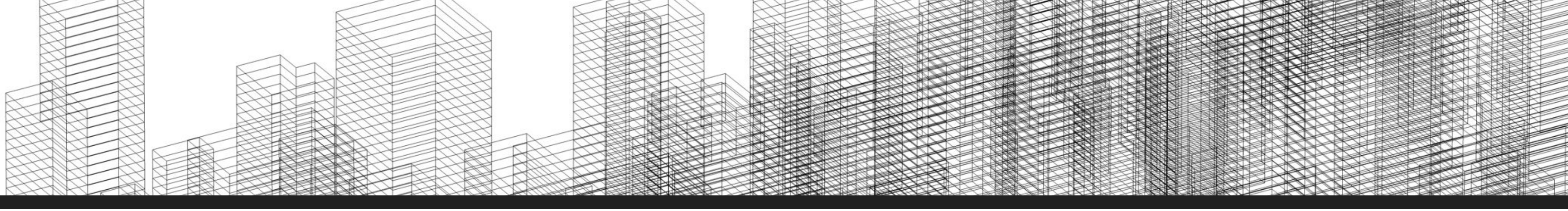


Fake Disaster Tweets Prediction

Getting Started Prediction Competition

Real or Not? NLP with Disaster Tweets

Predict which Tweets are about real disasters and which ones are not

$10,000

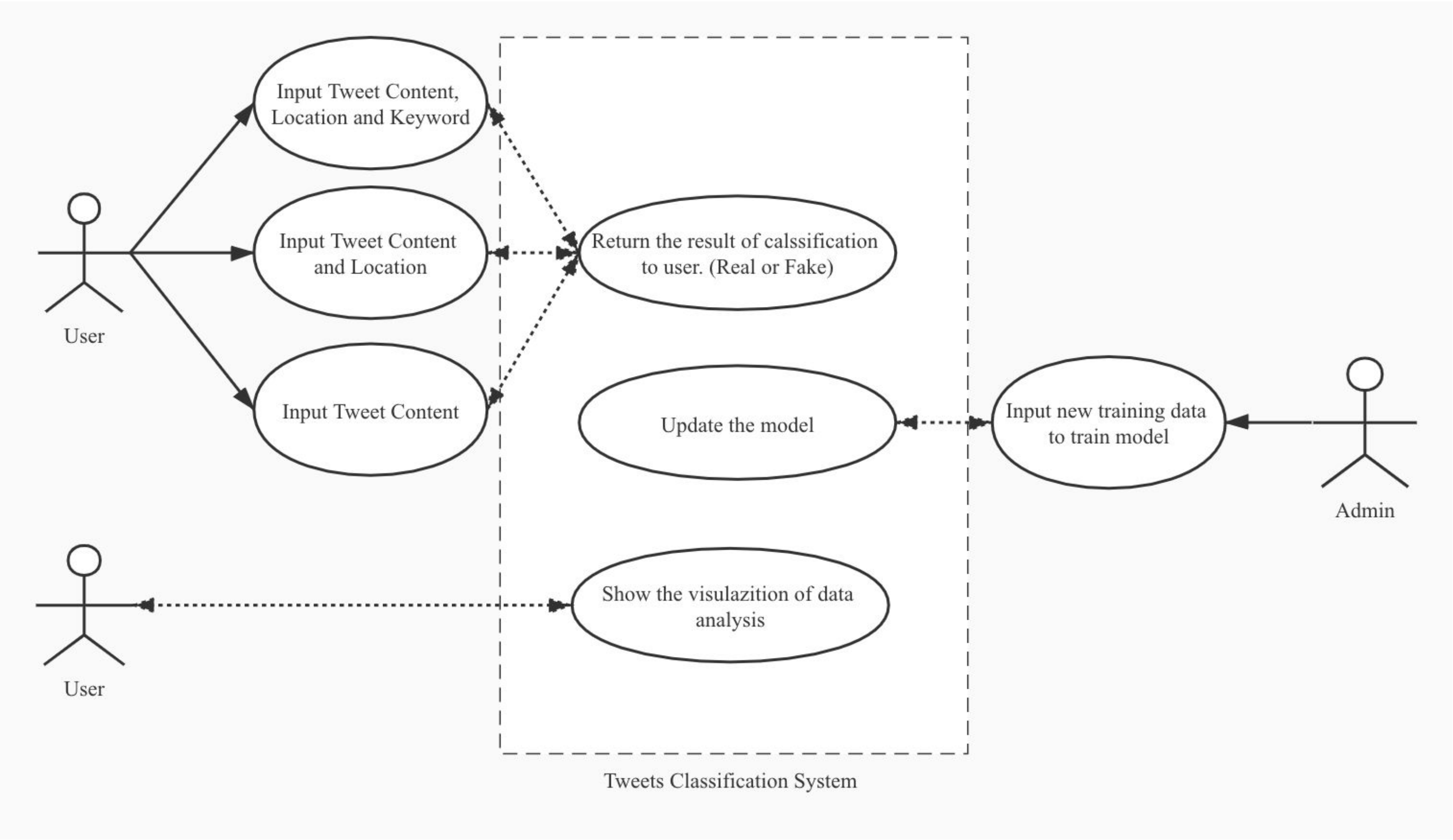
Prize Money

Kaggle · 3,347 teams · Ongoing

Group 8

Chao Ma
Yan Sun
Ruonan Ren

Use Cases


Input Tweet Content, Location and Keyword
Input Tweet Content and Location
Input Tweet Content
User
Return the result of calssification to user. (Real or Fake)
Update the model
Input new training data to train model
Admin
Show the visulazition of data analysis
User
Tweets Classification System

Acceptance Criteria

As a user, I am able to input Disaster Tweet content, location and keyword to get the prediction if the tweet is fake:

- The prediction accuracy for complete input data should be over 70%
- The time to respond should be under 5 seconds

As a user, I am able to show the visualization of data analysis:

- The time to respond should be under 5 seconds

Goals

- Create a reactive page to detect fake news on twitter.

- Create a reactive page to analyze the characteristics of fake tweets.

- Get a well trained model for fake tweets prediction.

Methodology

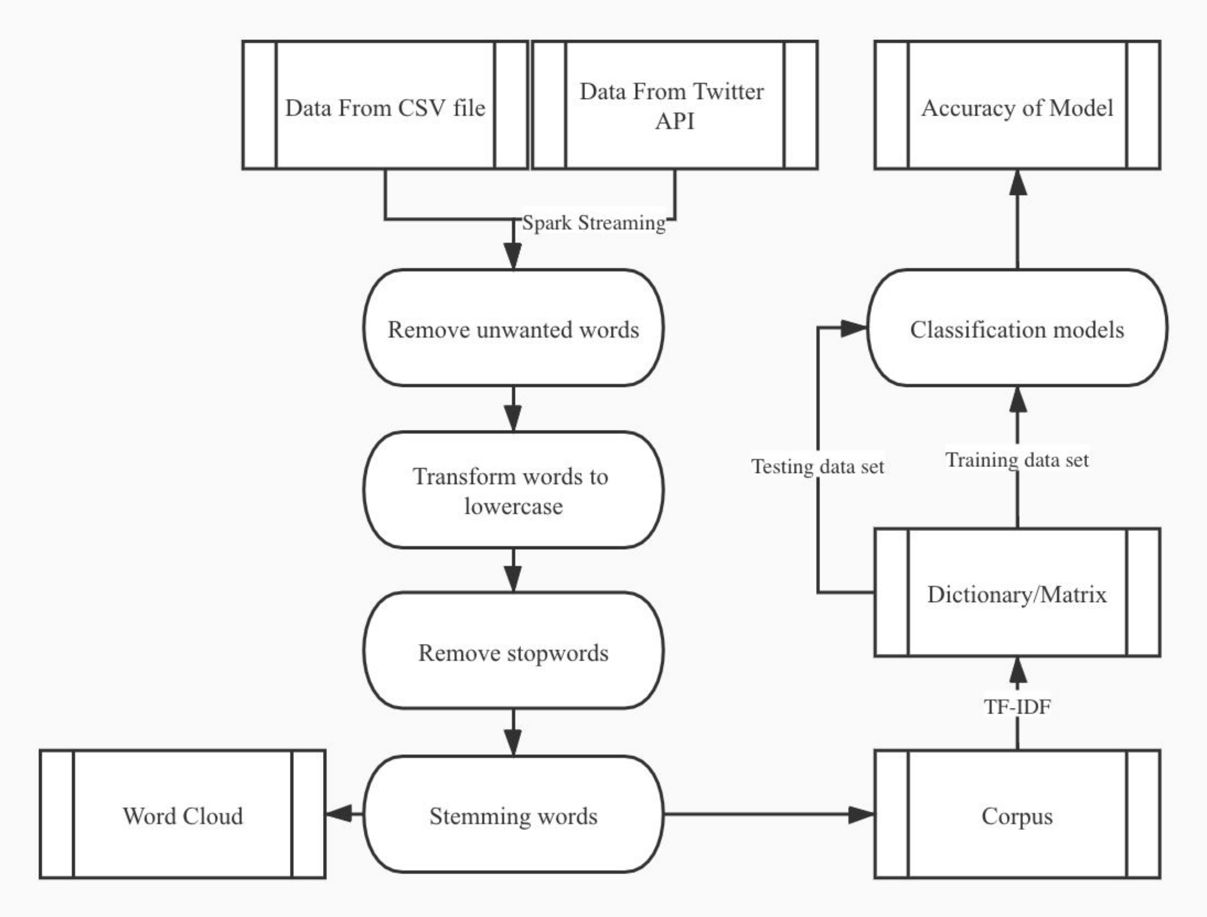
- Spark Streaming deals with data from Twitter API

- BoW & TF-IDF creates corpus and dictionary for content of tweets

- Algorithms might be applied: Decision Trees/ Random Forest, SVM, Gaussian Naive Bayes, K - Nearest Neighbors

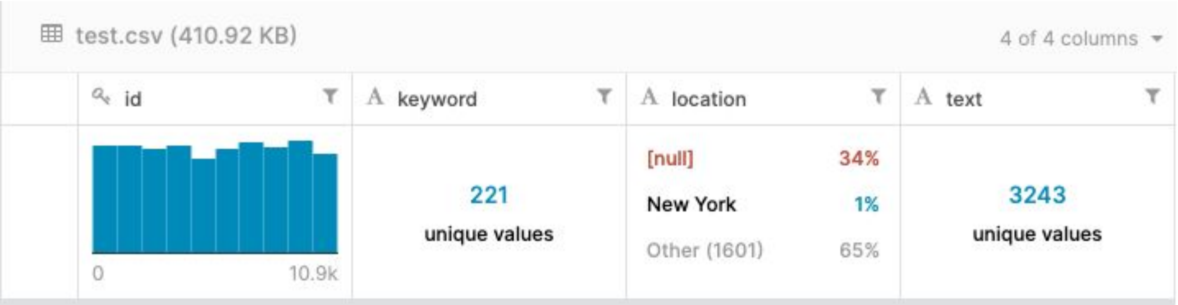
- Library: Spark MLlib, some related Java libraries

Methodology



Data Sources

- Data come from Kaggle competition and Twitter API

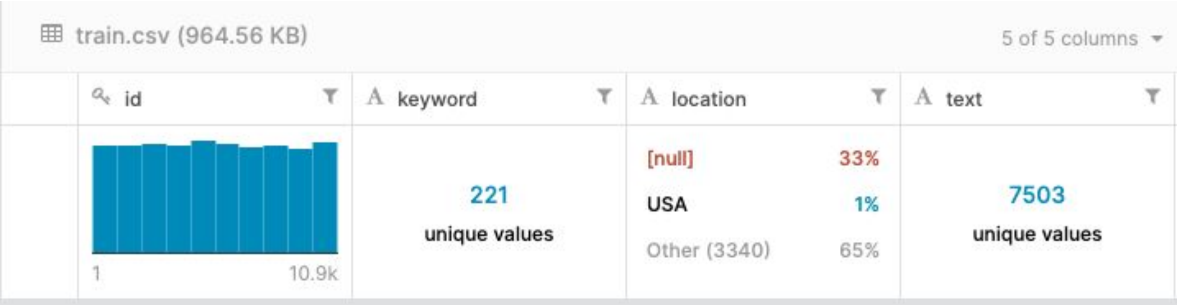


test.csv (410.92 KB)

4 of 4 columns

id	keyword	location	text
0 10.9k	221 unique values	[null] 34% New York 1% Other (1601) 65%	3243 unique values

- Data magnitude is more than 10,000 rows



train.csv (964.56 KB)

5 of 5 columns

id	keyword	location	text
1 10.9k	221 unique values	[null] 33% USA 1% Other (3340) 65%	7503 unique values

Milestones

Sprint	Milestone	Start Date	End Date
1	● Data cleaning and processing ● Unit Test	03/16/2020	03/21/2020
2	● Training machine learning model ● Unit Test	03/22/2020	03/28/2020
3	● Setup UI ● Implement visualization	03/29/2020	04/04/2020
4	● Final model and use cases testing ● System Test	04/05/2020	04/12/2020

Code

- Ingest data using Scala
- Reference Python to build ML model
- Exploratory data analysis using zeppelin
- Use MLlib in spark to train model
- Host code on GitHub Repository:

  https://github.com/SwagMC/CSYE7200FinalProject

Thank you!