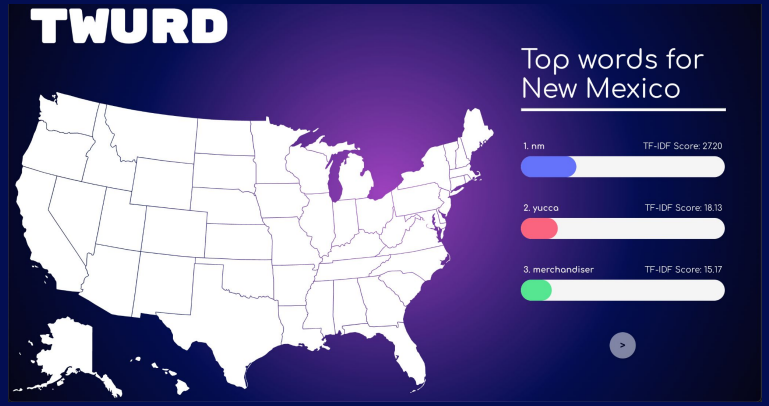
TWURD

John Lee, Isha Karki, Amey Erdenebileg,
David Gao, Calista Nguyen, Vananh Le

Objective

Collect a corpus of US tweets, run TF-IDF to find most relevant words from each state.



Twitter API

11,033

tweets from USA collected

- Gathered geo-tagged tweets only
- Removed emojis & symbols
- Filtered english tweets


```
{
  "_id": {
    "$oid": "6261e5b1650bd46d38ecff74"
  },
  "data": {
    "author_id": "329310886",
    "geo": {
      "place_id": "01fbe706f872cb32"
    },
    "id": "1517280982730162177",
    "text": "I am TIRED and MENTALLY Drained!"
  },
  "includes": {
    "users": [
      {
        "id": "329310886",
        "name": "Ro'chelle Williams",
        "username": "Prof_RWilliams"
      }
    ],
    "places": [
      {
        "country": "United States",
        "country_code": "US",
        "full_name": "Washington, DC",
        "id": "01fbe706f872cb32"
      }
    ]
  }
}
```

Algorithm

Run TF-IDF: Term Frequency - Inverse Document Frequency.

Combination of relevance of word in document times by how 'rare' the word is in the general context.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

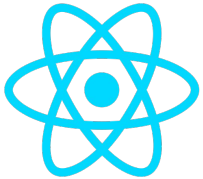
DEMO

Tech Stack

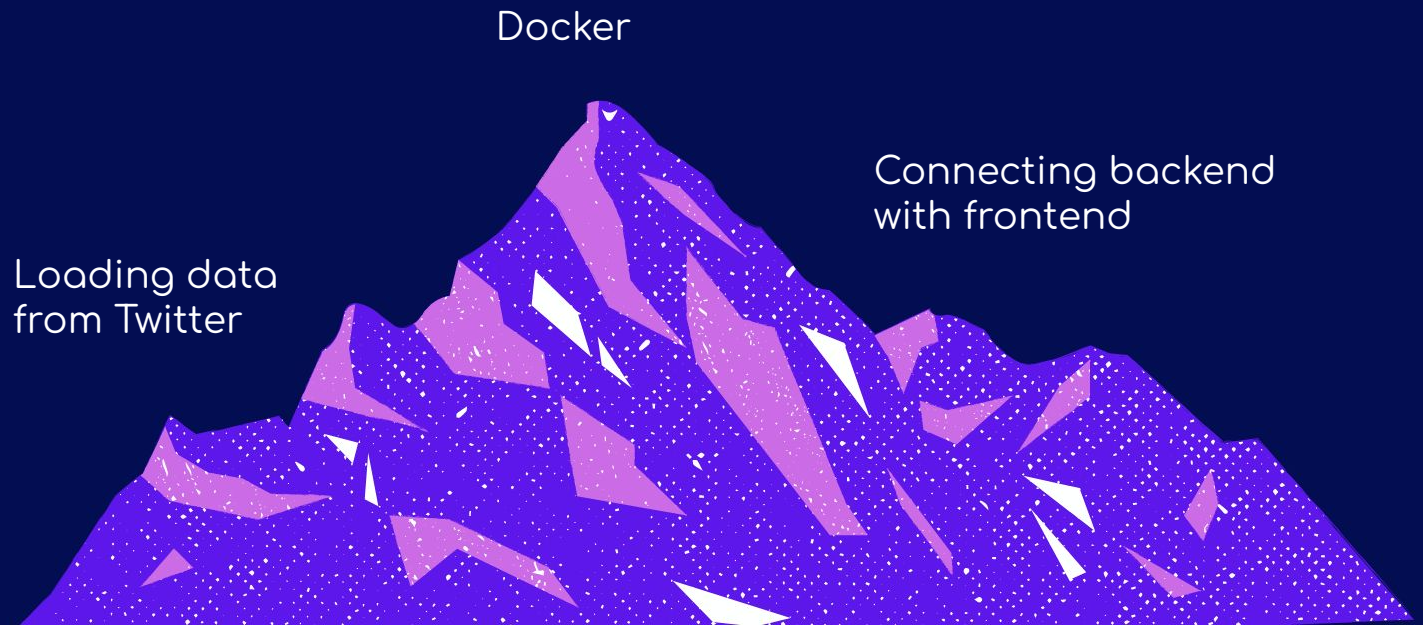
Backend

Frontend

Twitter API



Challenges


Docker
Connecting backend with frontend
Loading data from Twitter

Future Work

1 Add technologies for Big Data:

MapReduce: Hadoop, Cache: Redis

2 Host on Cloud

3 Mobile Friendly

Thank You