

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CNTT & TT



**BÁO CÁO BÀI TẬP LỚN
LƯU TRỮ VÀ XỬ LÝ DỮ LIỆU LỚN**

**Đề tài
PHÂN TÍCH DATA GOOGLE TREND VỀ ĐẠI DỊCH
COVID-19**

Nhóm sinh viên thực hiện:

Đỗ Hà Thủy

MSSV: 20175246

Bùi Minh Hiếu

MSSV: 20173114

Trần Hữu Hiếu

MSSV: 20180078

Nguyễn Như Hoàng

MSSV: 20164850

Hà Nội, 06/2021

Mục lục

Mục lục	1
Chương 1: Giới thiệu đề tài	2
Chương 2: Triển khai cài đặt	2
Khái quát mô hình hệ thống	2
Dữ liệu đầu vào	3
Worker	3
MongoDB	3
Apache Spark, Jupyter Notebook	4
Elastic Cloud, Kibana	5
Chương 3: Kết quả cài đặt	6
Trực quan hóa dữ liệu	6
Visualizing “data trend” by Kibana	6
Visualizing “Real covid data” by kibana	9
Spark Machine Learning	11
Mô tả bài toán	11
Các bước tiến hành:	12
Tài liệu tham khảo	16

1. Chương 1: Giới thiệu đề tài

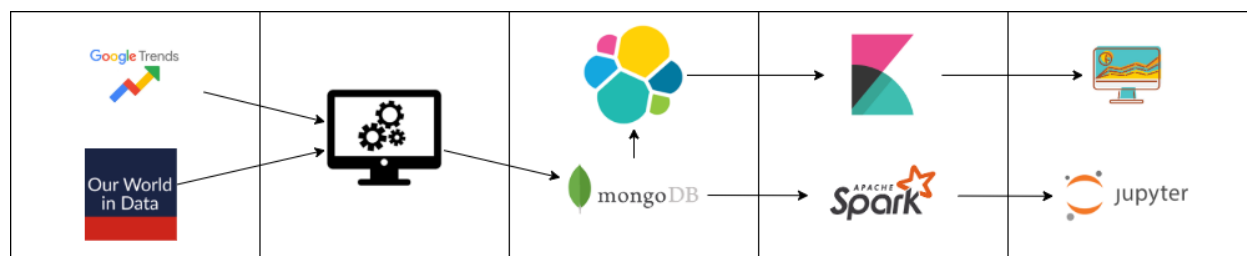
Kể từ khi bùng phát coronavirus vào tháng 12 năm 2019 ở Vũ Hán, Trung Quốc, nó đang lây lan theo cấp số nhân và đã ảnh hưởng đến gần như mọi quận trên thế giới, lây nhiễm cho hàng triệu người và gây ra hơn hàng chục nghìn ca tử vong trên khắp thế giới (tính đến ngày 16 tháng 3, 2020). Nó đã gây ra thiệt hại kinh tế và xã hội vô cùng thảm khốc trên khắp thế giới. Theo ước tính của Cục Dự trữ Liên bang, Coronavirus có thể mất việc làm tổng cộng 47 triệu người, tỷ lệ thất nghiệp có thể lên tới 32%. Để dự đoán số lượng bệnh nhân bị nhiễm là rất quan trọng đối với sự chuẩn bị của cả cá nhân và người ra quyết định, và để làm phẳng các đường cong. Tuy nhiên, làm thế nào để dự đoán chính xác số lượng bệnh nhân mắc bệnh không bao giờ là một việc đơn giản. Có rất nhiều yếu tố góp phần vào sự lan truyền của vi rút này, chẳng hạn như khả năng di chuyển của quần thể, nhiệt độ và tình trạng y tế.

Ngày nay, ngày càng có nhiều người truy cập internet và tìm kiếm những thông tin liên quan mật thiết đến cuộc sống, cảm xúc và suy nghĩ hàng ngày của họ. Người ta ước tính rằng có khoảng 63.000 lượt tìm kiếm trên Google mỗi giây. Một người bình thường thực hiện khoảng ba hoặc bốn lần tìm kiếm mỗi ngày. Google Trend là một trang web được Google tài trợ phân tích mức độ phổ biến của các truy vấn tìm kiếm hàng đầu trong Google Search trên nhiều khu vực và ngôn ngữ khác nhau. Trang web đáng tin cậy để tìm kiếm thông tin mà người dùng quan tâm. Do đó, các xu hướng của Google đang bộc lộ và có thể tạo cơ hội để xem xét mối quan tâm của mọi người cũng như các chủ đề nóng mà họ quan tâm. Các nhà nghiên cứu đã sử dụng dữ liệu của Google Trend để điều tra một số nghiên cứu, trong đó có dự đoán sự bùng phát dịch bệnh.

Trong bài báo này, ta khám phá dữ liệu của Google Trend để tìm ra mối quan hệ của nó với sự lan truyền COVID-19. Thay vì tập trung vào dự đoán dựa trên mô hình, chúng tôi đề xuất sử dụng dữ liệu của Google Trend và kết hợp với chuỗi thời gian lịch sử để dự đoán các ca mắc trong tương lai. Cách tiếp cận của chúng tôi là theo hướng dữ liệu thuần túy và bỏ qua mô hình toán học phức tạp, điều này làm giảm đáng kể độ phức tạp của thuật toán. Chúng tôi đã thực hiện các thử nghiệm toàn diện và áp dụng nhiều mô hình dự đoán phổ biến trên dữ liệu toàn cầu để xem mối tương quan giữa xu hướng tìm kiếm và các trường hợp bị nhiễm bệnh. Các thử nghiệm của chúng tôi đã chứng minh rằng có mối quan hệ chặt chẽ giữa các trường hợp bệnh nhân bị nhiễm bệnh và dữ liệu xu hướng của Google, và có thể được sử dụng với các kỹ thuật phân tích khác để hiểu rõ hơn về sự lây lan của căn bệnh này.

2. Chương 2: Triển khai cài đặt

2.1. Khái quát mô hình hệ thống



2.2. Dữ liệu đầu vào

Hai nguồn dữ liệu chính được sử dụng để giải quyết bài toán đã đặt ra bao gồm:

- Google Trends: Google Trends là một dịch vụ web công cộng của Google. Nó giúp cung cấp thông kê về kết quả tìm kiếm Google Search trên toàn cầu kể từ năm 2004.
- Our World in Data (<https://ourworldindata.org/>): là một ấn phẩm trực tuyến khoa học tập trung vào các vấn đề toàn cầu lớn như nghèo đói, bệnh tật, đói kém, biến đổi khí hậu, chiến tranh, rủi ro hiện hữu và bất bình đẳng. Đây là một dự án của Phòng thí nghiệm Dữ liệu Thay đổi Toàn cầu, một tổ chức từ thiện đã đăng ký ở Anh và xứ Wales, và được thành lập bởi Max Roser, một nhà sử học xã hội và nhà kinh tế phát triển. Nhóm nghiên cứu có trụ sở tại Đại học Oxford.

2.3. Worker

Nhiệm vụ của Worker là đọc dữ liệu từ các nguồn dữ liệu nêu trên và insert vào database MongoDB. Đối với dữ liệu Covid thực tế, ta có thể đọc trực tiếp từ các file dữ liệu được cung cấp và insert lần lượt vào MongoDB. Tuy nhiên đối với Google Trends API, vấn đề nằm ở giới hạn về số lượng request trên mỗi IP của Google, giải pháp của nhóm là sử dụng các dịch vụ VPS.

Source code: https://github.com/Mhieul4/covid_trend_data_collector

2.4. MongoDB

MongoDB được triển khai trên dịch vụ MongoDB Atlas. Dữ liệu trong mỗi Cluster ở Atlas được lưu trữ theo cơ chế Replication, với 3 nodes: 1 master (primary) và 2 slaves (secondary). Tổ chức lưu trữ trong MongoDB bao gồm các collection như sau:

Dữ liệu về quốc gia (country):

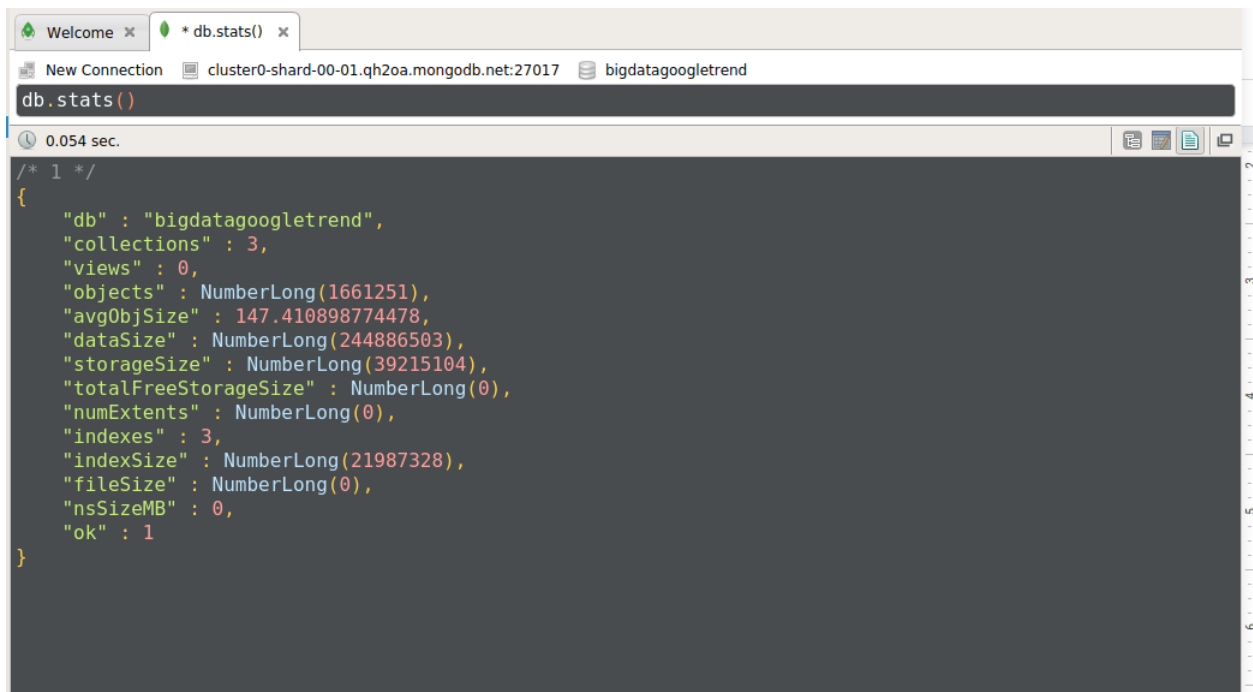
```
const country = Schema({
  Code_2: String,
  Code_3: String,
  Code_num: String,
  Crawled: { type: Number, default: 0 },
}, { timestamps: false, collection: 'country' });
```

Dữ liệu thống kê của Google Trend (data_trend):

```
const dataTrend = Schema({
  country_code_2: String,
  country_code_3: String,
  date_statistic: Date,
  key_word: String,
  value: { type: Number, default: 0 },
}, { timestamps: false, collection: 'data_trend' });
```

Dữ liệu về các ca nhiễm Covid thực tế (data_covid):

```
const dataCovid = Schema({
  country_code_2: String,
  country_code_3: String,
  date_statistic: Date,
  total_cases: Number,
  new_cases: Number,
  total_cases_per_million: Number,
  new_cases_per_million: Number,
  stringency_index: Number,
}, { timestamps: false, collection: 'data_covid' });
```



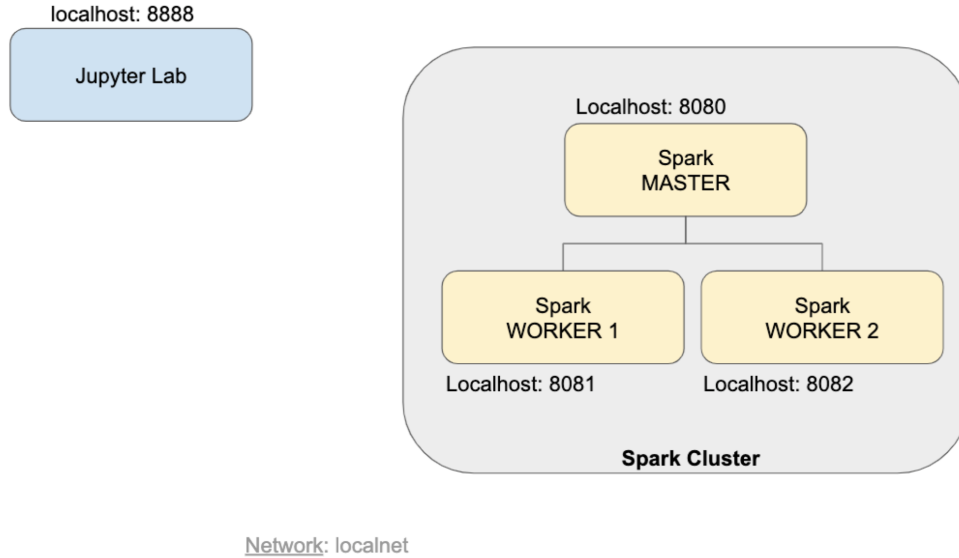
The screenshot shows a MongoDB CLI window with the following details:

- Tab: `* db.stats() *`
- Connection: `cluster0-shard-00-01.qh2oa.mongodb.net:27017`
- Database: `bigdatagoogletrend`
- Command: `db.stats()`
- Execution time: `0.054 sec.`
- Output (JSON):

```
{
  "db" : "bigdatagoogletrend",
  "collections" : 3,
  "views" : 0,
  "objects" : NumberLong(1661251),
  "avgObjSize" : 147.410898774478,
  "dataSize" : NumberLong(244886503),
  "storageSize" : NumberLong(39215104),
  "totalFreeStorageSize" : NumberLong(0),
  "numExtents" : NumberLong(0),
  "indexes" : 3,
  "indexSize" : NumberLong(21987328),
  "fileSize" : NumberLong(0),
  "nsSizeMB" : 0,
  "ok" : 1
}
```

2.5. Apache Spark, Jupyter Notebook

Cụm Spark cài đặt trên docker bao gồm 4 container là JupyterLab, một master, hai worker.



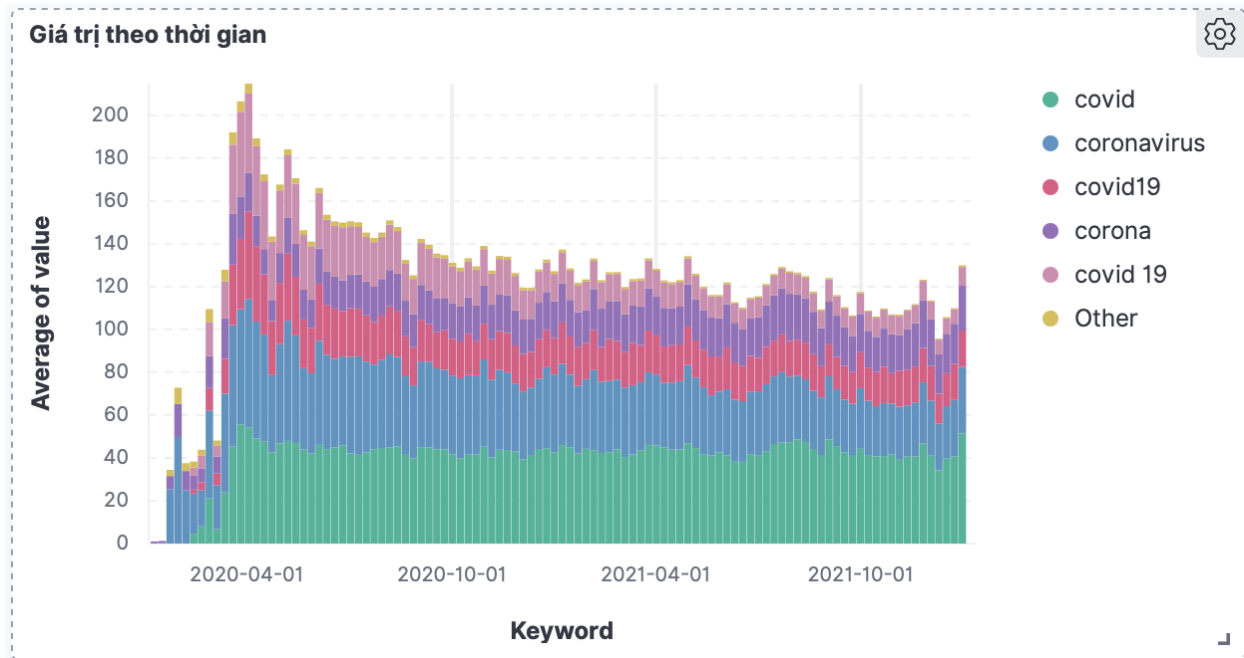
2.6. Elastic Cloud, Kibana

```
1  version: "3.0"
2  ∨ services:
3  ∨   elasticsearch:
4  |     container_name: es-container
5  |     image: docker.elastic.co/elasticsearch/elasticsearch:7.11.0
6  |     environment:
7  |       - xpack.security.enabled=false
8  |       - "discovery.type=single-node"
9  |     networks:
10 |       - es-net
11 |     ports:
12 |       - 9200:9200
13 |   kibana:
14 |     container_name: kb-container
15 |     image: docker.elastic.co/kibana/kibana:7.11.0
16 |     environment:
17 |       - ELASTICSEARCH_HOSTS=http://es-container:9200
18 |     networks:
19 |       - es-net
20 |     depends_on:
21 |       - elasticsearch
22 |     ports:
23 |       - 5601:5601
24 |   networks:
25 |     es-net:
26 |       driver: bridge
27
```

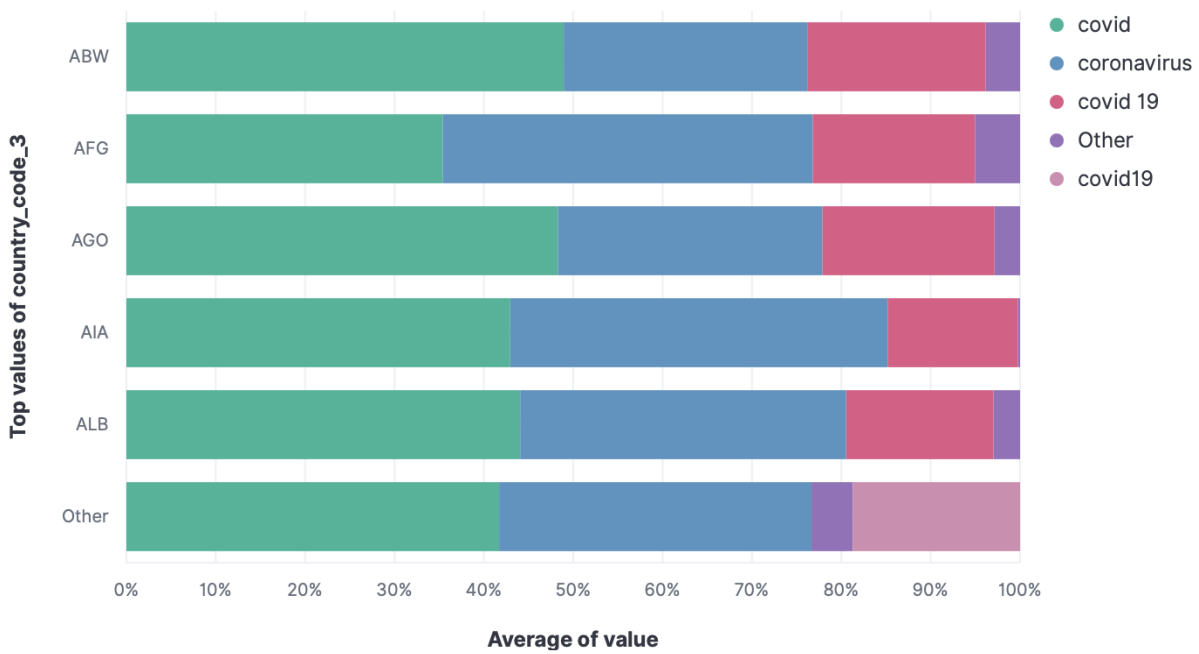
3. Chương 3: Kết quả cài đặt

3.1. Trực quan hóa dữ liệu

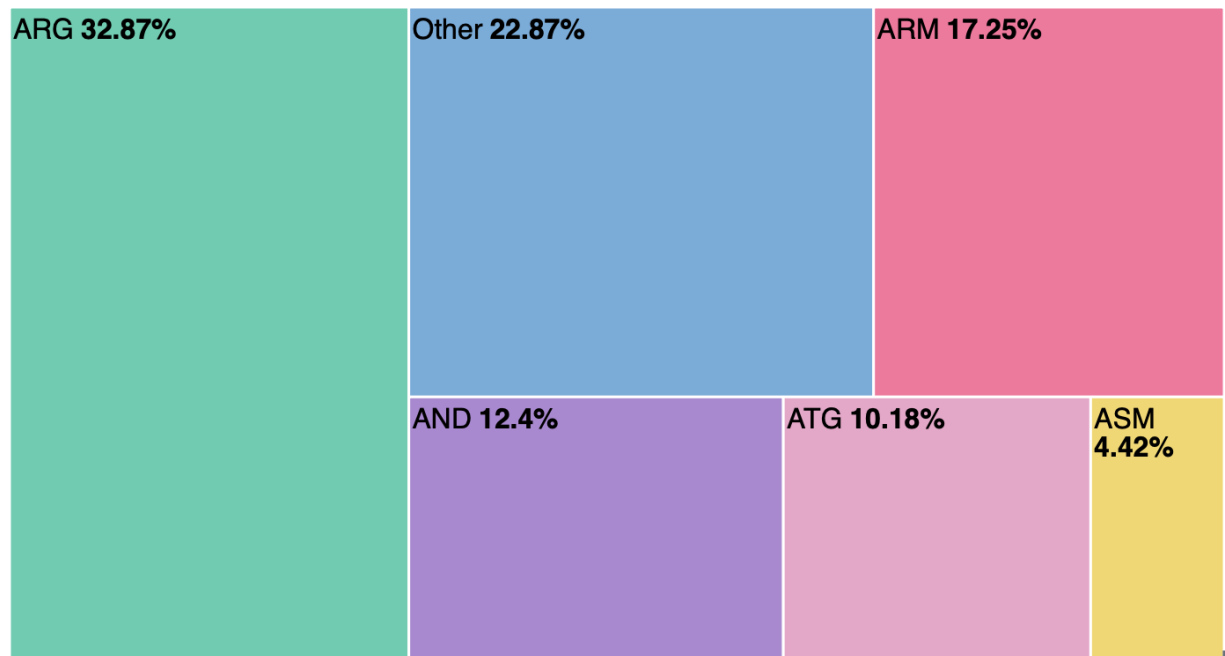
3.1.1. Visualizing “data trend” by Kibana



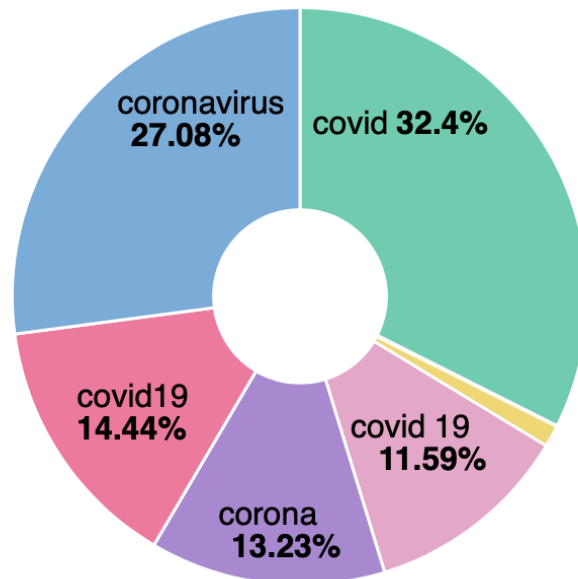
Top value of country code



Country Code Value



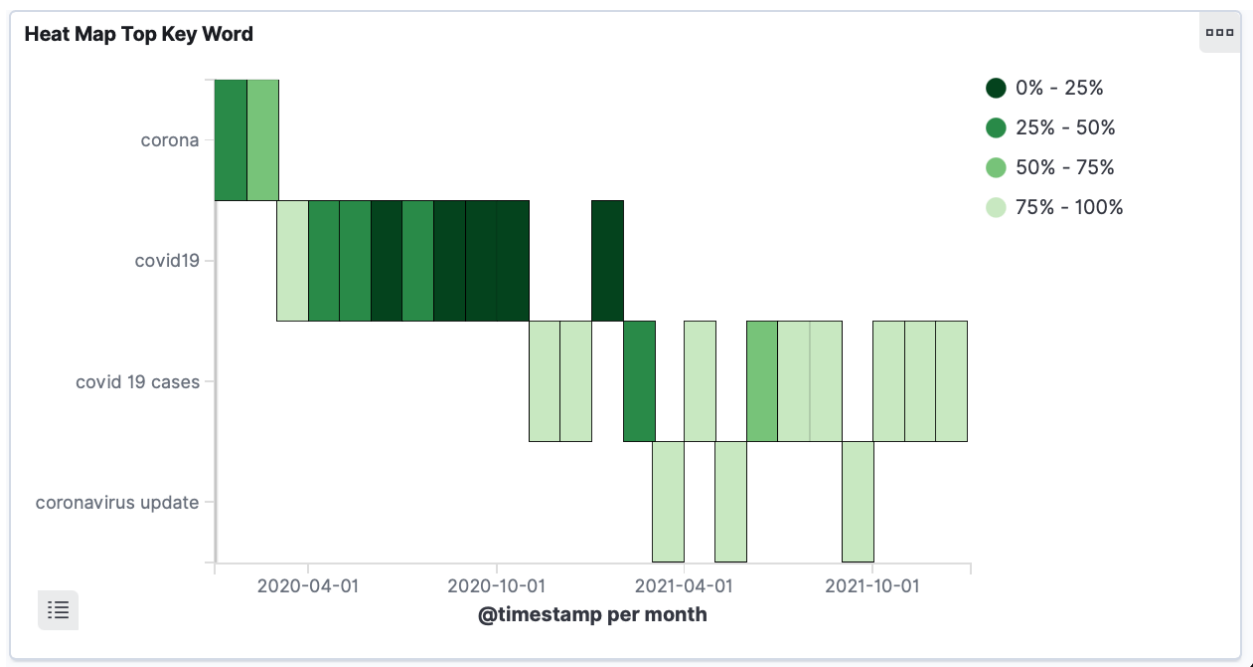
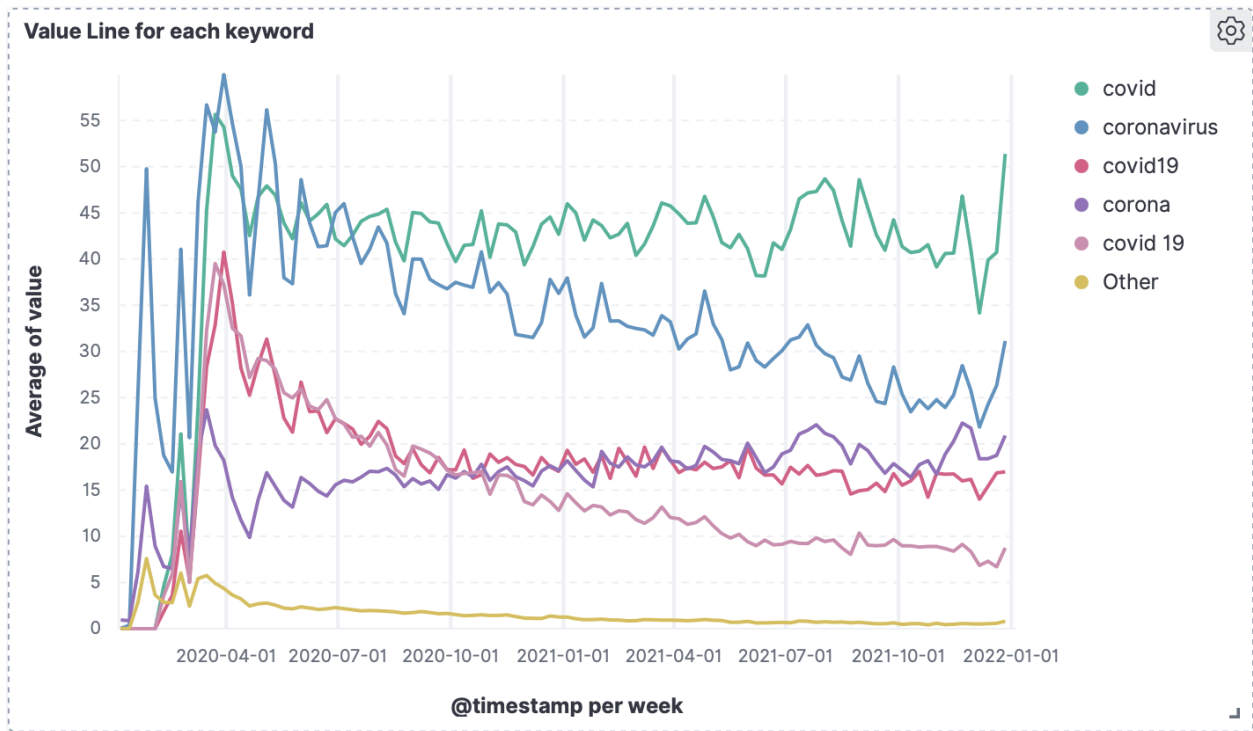
Category Keyword



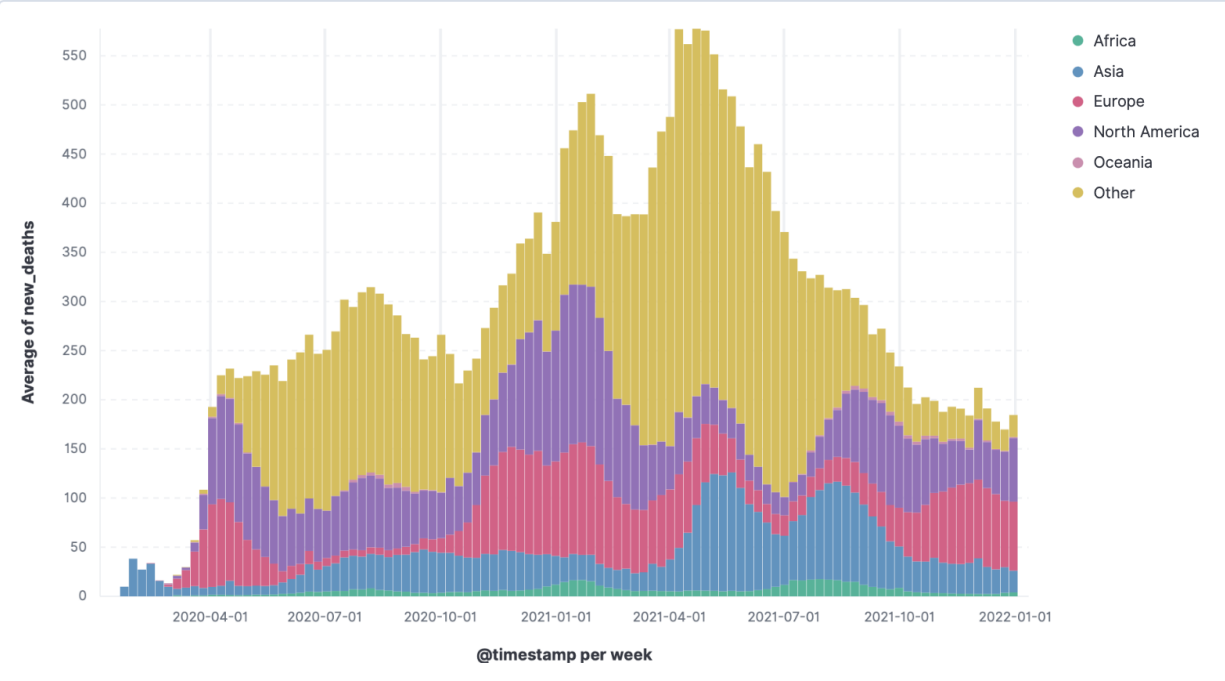
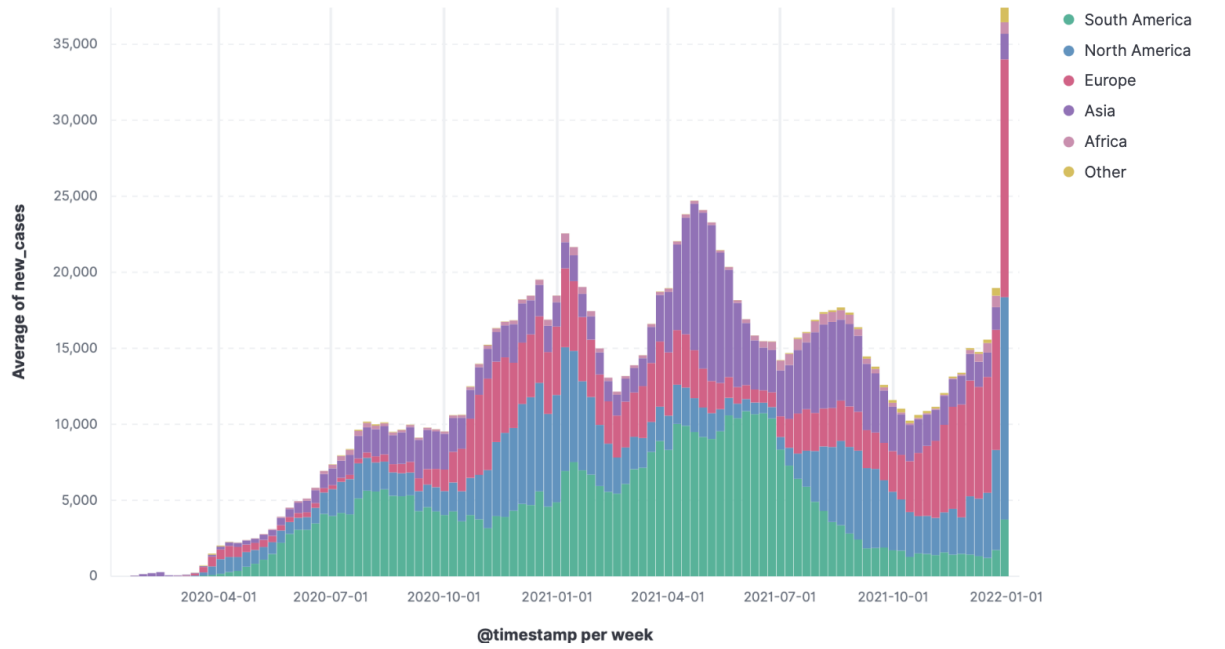
Sum of value

16,103,171

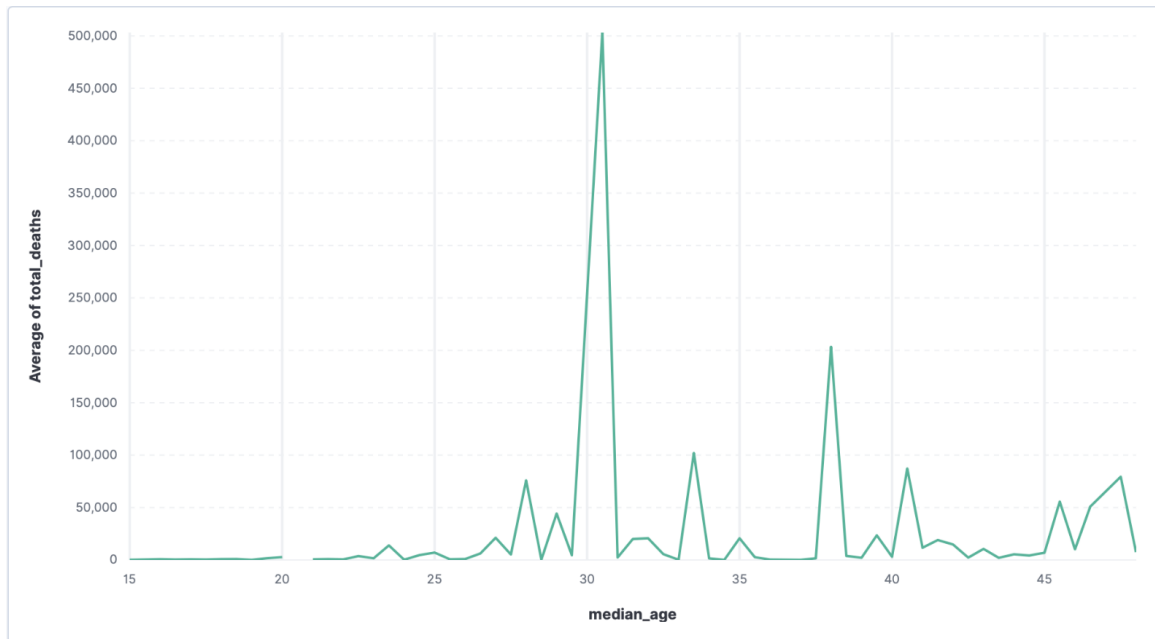
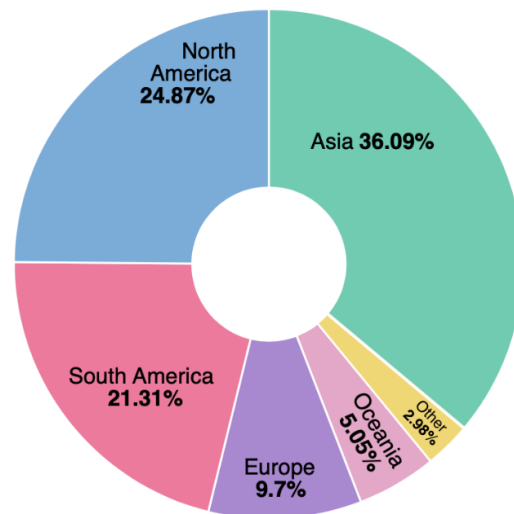
Sum of value



3.1.2. Visualizing “Real covid data” by kibana



People vaccinated Category



3.2. Spark Machine Learning

3.2.1. Mô tả bài toán

Trong phần này, ta sẽ xây dựng một mô hình dùng để dự đoán số ca mắc Covid-19 của các nước dựa trên các tham số đầu vào.

Các tham số đầu vào bao gồm:

- Keyword: “corona”, “covid19”,...
- Value
- Số ca mắc

Đầu ra sẽ là số ca mắc covid-19 của tháng sau.

3.2.2. Các bước tiến hành:

Đầu tiên tiến hành load dữ liệu:

```
df.printSchema()

root
 |-- __v: integer (nullable = true)
 |-- _id: struct (nullable = true)
 |   |-- oid: string (nullable = true)
 |-- country_code_2: string (nullable = true)
 |-- country_code_3: string (nullable = true)
 |-- date_statistic: timestamp (nullable = true)
 |-- key_word: string (nullable = true)
 |-- value: integer (nullable = true)
```

Tiếp theo, ta tiến hành xử lý dữ liệu:

```
In [3]: #get map countrys
data3= pd.read_csv("countries_3_digit.csv")
code_3=list(data3['Code_3'])
name_country=list(data3['Name'])
map_country={}
for i in range(len(code_3)):
    map_country[code_3[i]]=name_country[i]
map_country
```

```
Out[3]: {'AFG': 'Afghanistan',
'ALB': 'Albania',
'DZA': 'Algeria',
'ASM': 'American Samoa',
'AND': 'Andorra',
'AGO': 'Angola',
'AIA': 'Anguilla',
'ATA': 'Antarctica',
'ATG': 'Antigua and Barbuda',
'ARG': 'Argentina',
'ARM': 'Armenia',
'ABW': 'Aruba',
'AUS': 'Australia',
'AUT': 'Austria',
```

```
In [4]: data= pd.read_csv("time_series_covid19_confirmed_global.csv")
data
```

```
Out[4]:
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	12/24/21	12/25/21	12/26/21	12/27/21	12/28/21
0	NaN	Afghanistan	33.939110	67.709953	0	0	0	0	0	0	...	157887	157895	157951	157967	157967
1	NaN	Albania	41.153300	20.168300	0	0	0	0	0	0	...	207221	207542	207709	207709	208367
2	NaN	Algeria	28.033900	1.659600	0	0	0	0	0	0	...	216098	216376	216637	216930	217267
3	NaN	Andorra	42.506300	1.521800	0	0	0	0	0	0	...	21730	21730	21730	22332	22547
4	NaN	Angola	-11.202700	17.873900	0	0	0	0	0	0	...	70221	71142	71752	71752	76767
...
275	NaN	Vietnam	14.058324	108.277199	0	2	2	2	2	2	...	1620869	1636455	1651673	1666545	1680967
276	NaN	West Bank and Gaza	31.952200	35.233200	0	0	0	0	0	0	...	467682	467682	467682	468619	469467
277	NaN	Yemen	15.552727	48.516388	0	0	0	0	0	0	...	10109	10111	10115	10118	10118
278	NaN	Zambia	-13.133897	27.849332	0	0	0	0	0	0	...	228932	231581	233120	234476	238367
279	NaN	Zimbabwe	-19.015438	29.154857	0	0	0	0	0	0	...	202736	203746	204351	205449	207547

```
name contry : Congo (the)
name contry : Andorra
name contry : Moldova (the Republic of)
name contry : Montenegro
name contry : Iceland
name contry : Guinea-Bissau
name contry : Kyrgyzstan
name contry : Kiribati
name contry : Northern Mariana Islands (the)
name contry : Guam
name contry : Algeria
name contry : Isle of Man
name contry : Comoros (the)
name contry : Côte d'Ivoire
name contry : Armenia
name contry : Eritrea
name contry : Cabo Verde
name contry : Greece
name contry : Guadeloupe
```

Chia dữ liệu thành 2 tập trainingData và testVal:

```
import pickle
from sklearn import linear_model
# pickle.dump(data,open("data.pickle","wb+"))
```

```
data=pickle.load(open("data.pickle","rb"))
data_train=data["data_train"]
data_val=data["data_val"]
```

```
df_train = pd.DataFrame(data_train,columns=keywords+["value"])
df_val = pd.DataFrame(data_val,columns=keywords+["value"])
```

df_train														
	corona	cases of covid19	covid19	covid19 cases	coronavirus cases	covid	coronavirus symptoms	coronavirus news	coronavirus	covid 19 cases	covid 19	coronavirus update	coronavirus covid19	value
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...
99995	0	0	0	0	0	26	0	0	50	0	9	0	0	458219
99996	0	0	0	0	0	29	0	0	0	0	0	7	0	466169
99997	0	0	0	0	0	93	0	0	0	0	0	0	0	475105
99998	0	0	0	0	0	54	0	0	0	0	15	0	0	483710
99999	43	0	0	0	0	44	76	0	43	0	15	0	0	491904

100000 rows × 14 columns

df_val														
	covid	corona	covid 19	cases of covid19	coronavirus update	coronavirus news	covid19	coronavirus symptoms	covid 19 cases	covid19 cases	coronavirus covid19	coronavirus cases	coronavirus	value
0	52	0	7	0	7	0	0	0	0	0	0	0	43	500216
1	62	0	8	0	0	0	0	0	0	0	0	0	0	509032
2	54	0	16	0	0	0	0	0	0	0	0	0	0	517668
3	100	0	15	0	0	0	81	0	0	0	0	0	45	526837
4	85	0	17	0	0	0	0	0	0	0	0	0	0	536609
...
21227	26	0	26	0	0	0	0	0	0	0	0	0	0	964857
21228	20	0	20	0	0	0	0	0	0	0	0	50	100	965002
21229	41	0	20	0	0	0	0	0	0	0	0	0	50	965243
21230	63	0	21	0	0	0	0	0	21	0	0	0	0	965571
21231	75	0	0	0	0	0	0	0	0	0	0	0	47	966004

21232 rows × 14 columns

Training:

```

X_train = df_train[keywords]
Y_train = df_train['value']

X_val= df_val[keywords]
Y_val = df_val['value']
# with sklearn
regr = linear_model.LinearRegression()
regr.fit(X_train, Y_train)

```

LinearRegression()

Dự đoán số ca mắc của 1 nước, vd Việt Nam:

```
Test={
  "country": "Viet Nam",
  "date": "2022-01-02",
  "search": {"corona": 1,
    "cases of covid19": 2,
    "covid19": 3,
    "covid19 cases": 4,
    "coronavirus cases": 5,
    "covid": 1,
    "coronavirus symptoms": 1,
    "coronavirus news": 2,
    "coronavirus": 2,
    "covid 19 cases": 1,
    "covid 19": 1,
    "coronavirus update": 2,
    "coronavirus covid19": 1}
}

value=regr.predict([list(Test['search'].values())])[0]
print("Country "+Test["country"]+" in next date "+Test["date"]+" predict "+str(value)+" case of cov")

Country Viet Nam in next date 2022-01-02 predict 269786.11604603496 case of cov
```


Tài liệu tham khảo

- [1]. <https://ieeexplore.ieee.org/abstract/document/9377852>
- [2]. <https://www.mongodb.com/blog/post/getting-started-with-mongodb-pyspark-and-jupyter-notebook>