

BIG DATA PROGRAMMING

Final Project Report

COVID-19 Data Analysis

BY

Neha Navgale

Christian Barlow

Andrew Poitras

Krishnapriya Akula

Jayadeep Kumar Reddy Singareddi

GitHub:

[https://github.com/NehaNavgale/Covid
_Data_Analysis](https://github.com/NehaNavgale/Covid_Data_Analysis)

YouTube Video:

https://youtu.be/uh7Yodmw_Ag

Introduction:

Covid-19 has impacted each and every person across the globe and every one is taking necessary measures to reduce its impact. Our project is to study and analyze its impact on different sectors and geographic locations and provide users with interactive visualizations.

Background:

There are some projects out there which shows analysis and visualization related to covid data. The scope of analysis is limited to identifying the number of active, recovered cases and deaths per country/region.

<https://www.tableau.com/covid-19-coronavirus-data-resources>

<https://www.worldometers.info/coronavirus/#countries>

<https://coronavirus.jhu.edu/map.html>

Goals and Objectives:

Motivation:

As the coronavirus is spreading rapidly, everyone looks for a place where they can find meaningful insights on its impact on different geographical locations, health care facilities and on the economy. In order to provide information, we are motivated to generate new insights in support of the ongoing fight against this infectious disease using big data techniques.

Significance/Uniqueness:

Every single person on the planet has been affected by COVID-19 and have been doing everything they can to avoid it. This project could help people avoid hot zones by analyzing the correlation between geographical location and the number of confirmed cases of the virus. The research gathered will also allow us to identify locations that may not be following proper social distancing and mask regulations as they are going to have a higher population of people with the virus. Insights gathered will also allow us to see in what age groups the disease has been most commonly spreading. We will also be able to identify the impact on different business sectors and by looking into trends we can identify its impact in the future. This project will serve as a single place to understand its overall impact.

Objectives:

Our objective is to understand the datasets and utilize latest big data techniques to extract meaningful insights which can help to understand the trend of coronavirus and the correlation of the number of cases with locations, age groups, business and economy.

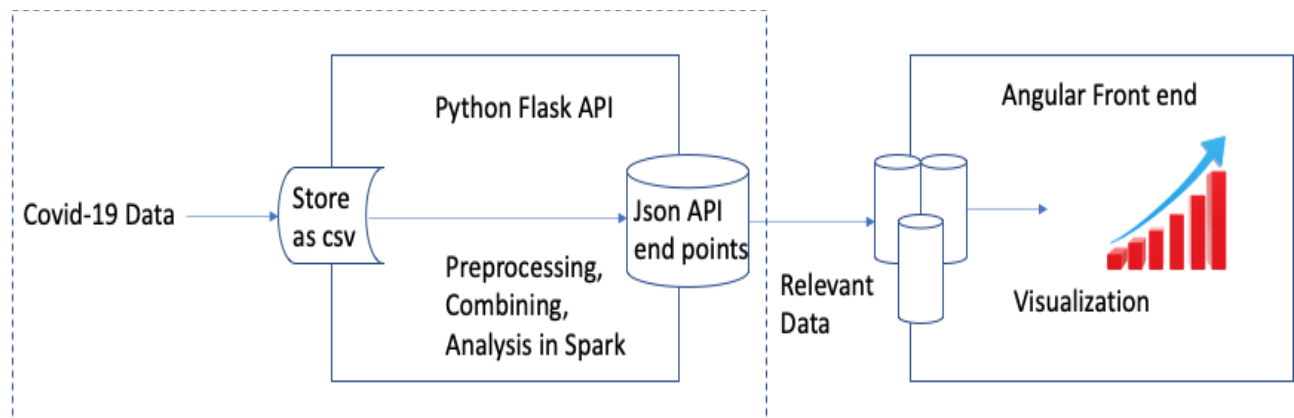
Features:

Our project will include the following features:

1. Collect real time data of covid-19 for each state of the USA
2. Identify answers to below questions based on data analysis:
 - a. Identify the total number of cases, deaths, total cases per million, icu_patients, total hospital patients in state of USA
 - b. Find the total number of cases and deaths for each USA state.
 - c. Find single day max death for each state and the date when it happened
 - d. Find top 10 states with max single day rise of cases and the date when it happened
 - e. Find rate of increase of covid cases only for Kansas State
 - f. Find the rate of increase of covid cases in the USA for Nov month.
 - g. Find the list of States for which Stay at Home order is lifted.
 - h. Find the total number of States in which Quarantine is not in place for Travelers.
 - i. Find the list of States where Bars have been reopened along with the mask requirement and gathering restriction data.
3. Develop interactive front end applications for users to understand the analysis.
4. Create graphical visualization of all the analysis completed in feature 2.

Model:

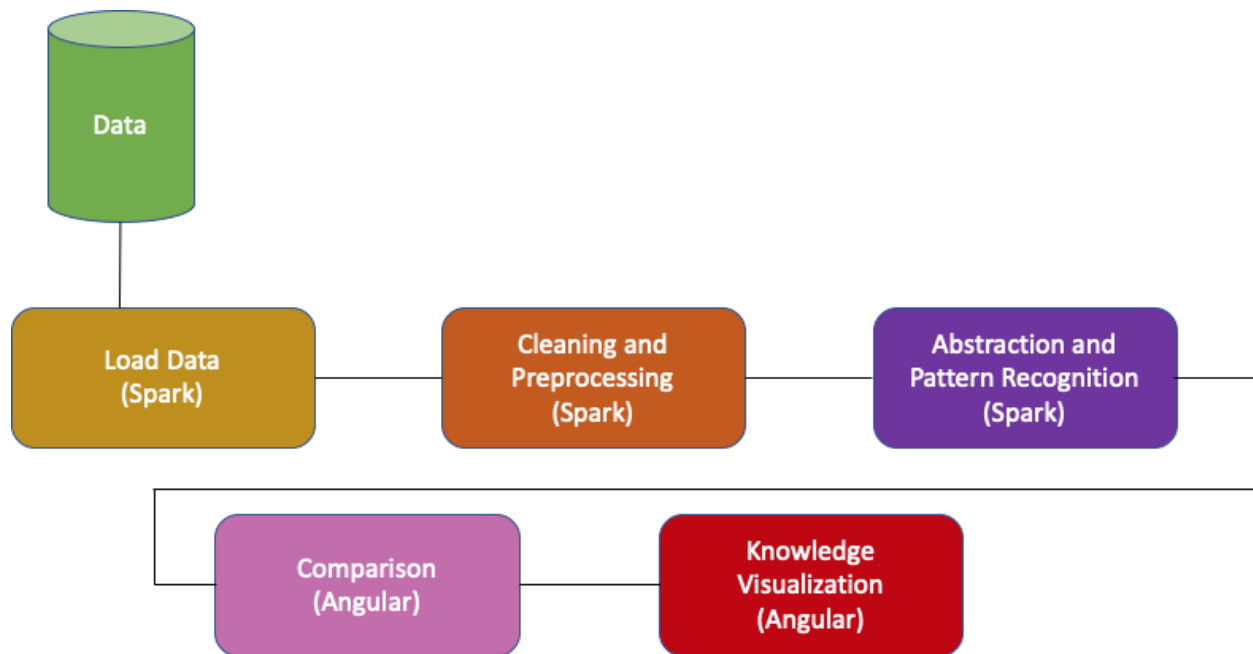
Architecture Diagram:



Explanation:

1. The data is collected from different sources in csv format and imported in a python flask project.
2. All preprocessing, data cleaning, analysis will be done in Spark.
3. The output of all the analysis will be in tabular format so to make it more user friendly, we have integrated the python project with an angular front end application.
4. All the relevant data out of big dataset will be extracted and shared with angular front end application via API
5. All visualizations are handled in angular.

Workflow Diagram:



Explanation:

1. The data is collected in csv format and stored in Python Flask application
2. PySpark library is used in Python to perform data analysis using Spark.
3. Data from csv is loaded into spark dataframe.
4. Cleaning of data, removing duplicates and creating schemas are while loading data.
5. Spark queries are executed to extract only meaningful and relevant information as per requirement
6. Output of queries are stored in file and converted to json.
7. Separate API endpoints are created for each query.
8. Queries output are accessed in an angular application using API.
9. AmCharts angular library is used to convert json data into interactive graphs
10. Each query output can be visualized in the form of graphs in-browser.

Dataset

USA Data:

<https://data.cdc.gov/Case-Surveillance/United-States-COVID-19-Cases-and-Deaths-by-State-o/9mfq-cb36>

Hospitalization Data of USA - <https://covidtracking.com/data>

World Data - <https://ourworldindata.org/coronavirus-source-data>

Social Distancing -

<https://www.kff.org/coronavirus-covid-19/issue-brief/state-data-and-policy-actions-to-address-coronavirus/>

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	date	state	dataQuality	death	deathConfirm	deathIncrease	deathProbab	hospitalized	hospitalized	hospitalized	hospitalized	icuCumulat	icuCurrent	negative	negativeInc	negativeTest	negativeTest	negativeTest	onVentilator	onVentilator	pending
2	10/10/20	AK	A	60	60	0				66	0			488825	5931			489481		10	
3	10/10/20	AL	A	2664	2508	11	156	18179	18179	792	190	1889		1059419	6867				1065		
4	10/10/20	AR	A+	1552	1405	49	147	5900	5900	554	95			234	1049968	24422		1049968	730	97	
5	10/10/20	AS	D	0	0	0				0					1616	0					
6	10/10/20	AZ	A+	5759	5472	13	287	20229	20229	685	30			145	1335929	11731				71	
7	10/10/20	CA	B	16500		72				3084	0			732	15035669	137778					
8	10/10/20	CO	A	1998		1		7855	7855	380	21			924468	10031	160671					
9	10/10/20	CT	B	4530	3631	0	899	11845	11845	134	0			1736501	0						
10	10/10/20	DC	A+	636		2				99	0			27	413181	6799				14	
11	10/10/20	DE	A+	653	573	2	80			103	0			21	285647	3107					
12	10/10/20	FL	A	15372	15372	0		46201	46201	2077	0			4789241	0	487730	475588	6895066			4176
13	10/10/20	GA	A	7393		45		29611	29611	1646	101	5508		2802290	18693			2823640			
14	10/10/20	GU	B	59		1		59		62	0			13	51760	254					
15	10/10/20	HI	B	166		2		911	911	106	11			29	295153	0					18
16	10/10/20	IA	A+	1455		18				450	0			101	719807	4193	58468				40
17	10/10/20	ID	A	506	464	3	42	2013	2013	192	16	475		45	288702	3653					
18	10/10/20	IL	A	9221	8975	30	246			1807	0			406	5927212	63351				166	
19	10/10/20	IN	A+	3782	3555	21	227	13756	13756	1180	108	2751		339	1341228	9285				115	
20	10/10/20	KS	A+	763		0		3185	3185	469	0	877		119	494382	0			271	34	
21	10/10/20	KY	B	1249	1236	7	13	6094	6094	652	61	1586		170	1467733	10104					
22	10/10/20	LA	A	5635	5442	0	193			582	0				2276681	0				78	
23	10/10/20	MA	A+	9587	9372	10	215	12861	12861	531	15			86	2259954	14573				28	
24	10/10/20	MD	A	3995	3850	5	145	16006	16006	383	90			91	1589007	11799					
25	10/10/20	ME	A+	143	142	0	1	462	462	7	2			5	485386	0	10015		483143	0	
26	10/10/20	MI	A+	7219	6891	19	328			862	0			220	3860845	43645		3860845		85	
27	10/10/20	MN	A	2184	2131	10	53	8302	8302	471	51	2277		133	1450275	16045					
28	10/10/20	MO	B	2422		27				1313	0				1303437	42982	76509	71709	1908333		
29	10/10/20	MP	D	2	2	0		4	4		0				15121	0					
30	10/10/20	MS	A+	3096	2825	16	271	6053	6053	600	0			136	704655	0					59
31	10/10/20	MT	C	209		3		885	885	280	17				370532	1185					
32	10/10/20	NC	A+	3765	3731	18	34			1034	0			281	3121063	37209					
33	10/10/20	ND	B	234	230	0	4	1062	1062	219	25	248		42	230558	1402	9655				
34	10/10/20	NE	A	519		5		2499	2499	299	18				453288	8880					
35	10/10/20	NH	B	455		6		750	750	21	3	238			283359	4809					
36	10/10/20	NJ	B	16171	14383	7	1788	23868	23868	641	47			156	3701144	34169					48
37	10/10/20	NM	B	902		3		3678	3678	133	26				947812	5623					
38	10/10/20	NV	A+	1659		2				519	0			133	649544	3903					64

We collected the data in csv format. The datasets contain data from March to present date for each state of the USA. There are some fields which we have not considered in scope of this project. Below are the definition of fields that are in scope.

Data Definitions:

1. Submission_date: The date when data was recorded in the system.
2. State: State of the USA
3. Tot_cases: The total number of cases for each date and for each state.
4. New_Case: Indicates the daily increase in the number of cases in each state.
5. total_death: Total number of deaths recorded in each state.
6. New_death: Daily increase in total number of deaths.
7. Total_cases_per_million: Ratio of cases per million population
8. icu_patients: Count of patients admitted to ICU due to covid
9. hosp_patients: Count of individuals currently hospitalized because of covid.
10. Status of Reopening: Shows the status of reopening like reopened, paused, new restrictions imposed
11. Stay at Home Order: Shows different stay home orders for each state
12. Mandatory Quarantine for Travelers: Indicates if quarantine is needed or lifted
13. Non-Essential Business Closures: Policy for Non-Essential Businesses
14. Restaurant Limits: Policy on restaurants dine-in and drive through facilities also shows the number of people allowed in restaurants
15. Face Covering Requirement: Shows if face covering is required in a particular state.

Analysis of data

Data Preprocessing

1. Imported csv data and created the dataframe on data

```
spark = SparkSession.builder.appName("Covid Analysis").getOrCreate()

# Loading CSV file to dataframes
dfWorld = spark.read.option("header", True).csv("Dataset/owid-covid-data.csv")

# Obtaining Summary Statistics
dfWorld.show()
dfWorld.printSchema()
```

2. Spark reads all columns as string. To assign the correct data type to all columns so that group by, aggregation and other operations can be performed, we have created schema for csv.

```
schemaUSA = StructType([
    StructField("submission_date", DateType(), True),
    StructField("state", StringType(), True),
    StructField("tot_cases", IntegerType(), True),
    StructField("conf_cases", IntegerType(), True),
    StructField("prob_cases", IntegerType(), True),
    StructField("new_case", IntegerType(), True),
    StructField("pnew_case", IntegerType(), True),
    StructField("tot_death", IntegerType(), True),
    StructField("conf_death", IntegerType(), True),
    StructField("prob_death", IntegerType(), True),
    StructField("new_death", IntegerType(), True),
    StructField("pnew_death", IntegerType(), True),
    StructField("created_at", DateType(), True),
    StructField("consent_cases", StringType(), True),
    StructField("consent_deaths", StringType(), True)
])

# Loading CSV file to dataframes
df = spark.read.schema(schemaUSA).option("header", True).csv("Dataset/United_States_COVID-19.csv")
```

3. Checked for duplicates and removed

```

# Loading CSV file to dataframes
dfWorld = spark.read.option("header", True).csv("Dataset/owid-covid-data.csv")

# Obtaining Summary Statistics
dfWorld.show()
dfWorld.printSchema()

# remove duplicates
dfWorld.dropDuplicates()

```

4. Applied filter on dataframe to get only USA data

```

# Obtaining Summary Statistics
dfWorld.show()
dfWorld.printSchema()

# remove duplicates
dfWorld.dropDuplicates()

# Filter operation to take only USA data
dfUSA = dfWorld.filter(col("iso_code") == "USA")

```

5. Create a view on dataframe

```

# Filter operation to take only USA data
dfUSA = dfWorld.filter(col("iso_code") == "USA")

# create view
dfUSA.createOrReplaceTempView("covidUSA")

```

Implementation

Technologies/Platform used: Hadoop, Spark, Python Flask, Angular 8.

Backend

1. Designed Python Flask application and collected covid-19 global and USA data in csv format.

2. Performed analysis on data and came up with spark sql queries to extract meaningful information from data.
3. Used pyspark library to create covid-19 view in Spark SQL, executed queries on view and generated the output.
4. Output is sent in json format to Angular 7 frontend application to draw visualization.

Frontend

1. Designed Angular 7 application to visualize all the executed queries.
2. Used am4charts library to draw different graphs.

Testing

1. Performed Unit testing on both Frontend and backend applications.
2. Performed integration testing of frontend and backend.

Feature 1

Collect data of covid-19 for each state of the USA

- Analyzed different datasets and finalized a few datasets out of all.
- Collected the data in csv format for each state.

Feature 2

Identify answers to below questions based on data analysis:

We have imported different datasets to perform different type analysis on total cases, hospitalization and age group affected more.

1. Identify the total number of cases, deaths, total cases per million, icu_patients, total hospital patients in state of USA

- Loaded the data into spark.
- Filtered the data to get only USA data.
- Applied max operation to get the latest total.
- Saved the data into csv and convert csv into json and passed the data to angular using API endpoints.

```
# Query 1: USA Total records
query1 = spark.sql("select max(total_cases) as total_cases, max(total_deaths) as total_deaths, max(total_cases_per_million) "
                  " as total_cases_per_million, max(icu_patients) as icu_patients, max(hosp_patients) as hosp_patients "
                  " from covidUSA group by iso_code")
query1.show()
pd = query1.toPandas()
pd.to_csv('static/Input/byUSATotal.csv', index=False)
```



```
def load_USAData():
    csv_data = pd.read_csv("static/Input/byUSATotal.csv", sep=',')
    return csv_data

@app.route('/api/byUSATotal')
def byUSATotal():
    data = load_USAData()
    return data.to_json(orient='records')
```

2. Find the total number of cases and deaths for each USA state.

- Loaded the data into spark.
- Applied max operation to get the latest total grouped by state.
- Saved the data into csv and convert csv into json and passed the data to angular using API endpoints.

```
# Query 2: State with positive cases and total death
query2 = spark.sql("SELECT state, max(tot_cases) tot_cases, max(tot_death) as tot_death FROM covid group by "
                  " state order by tot_cases desc")
query2.show()
pd = query2.toPandas()
pd.to_csv('static/Input/byCases.csv', index=False)
```

```
def load_positiveCaseData():
    csv_data = pd.read_csv("static/Input/byCases.csv", sep=',')
    return csv_data

@app.route('/api/byCases')
def byCases():
    data = load_positiveCaseData()
    return data.to_json(orient='records')
```

3. Find top 10 states with max cases as of today

- Loaded the data into spark.
- Applied max operation to get the latest total grouped by state.
- Used limit to get only top 10 results
- Saved the data into csv and convert csv into json and passed the data to angular using API endpoints.

```
# Query 3: State with most positive cases and max death
query3 = spark.sql("SELECT state, max(tot_cases) as tot_cases FROM covid group by "
                  " state order by tot_cases desc limit 10")
query3.show()
pd = query3.toPandas()
pd.to_csv('static/Input/byCasesTop10State.csv', index=False)
```

```
def load_top10StateCaseData():
    csv_data = pd.read_csv("static/Input/byCasesTop10State.csv", sep=',')
    return csv_data

@app.route('/api/byCasesTop10State')
def byTop10StateCases():
    data = load_top10StateCaseData()
    return data.to_json(orient='records')
```

4. Find single day max death for each state and the date when it happened

- Loaded the data into spark.
- Applied max operation on new death recorded each day and grouped by state.
- Join the query with max query to find the date for respective record.
- Saved the data into csv and convert csv into json and passed the data to angular using API endpoints.

```
# Query 3: State with max single day death with date
query3 = spark.sql("Select c.state, c.new_death as death, c.submission_date as date from covid c join"
                  " (SELECT state, max(new_death) as death FROM covid group by state) m on "
                  " c.state = m.state and c.new_death = m.death and c.new_death > 0")
query3.show()
pd = query3.toPandas()
pd.to_csv('static/Input/byDeath.csv', index=False)
```

```
def load_deathData():
    csv_data = pd.read_csv("static/Input/byDeath.csv", sep=',')
    return csv_data

@app.route('/api/byDeath')
def byDeath():
    data = load_deathData()
    return data.to_json(orient='records')
```

5. Find top 10 states with max single day rise of cases and the date when it happened

- Loaded the data into spark.

- Applied max operation on new cases recorded each day and grouped by state.
- Join the query with max query to find the date for respective record.
- Limited the record for 10.
- Saved the data into csv and convert csv into json and passed the data to angular using API endpoints.

```
query4 = spark.sql("select c.state, c.new_case, c.submission_date as date from covid c join "
                  " (select state, max(new_case) as new_case from covid group by state) "
                  " m on c.state = m.state and c.new_case = m.new_case and c.new_case > 0 limit 10")
query4.show()
pd = query4.toPandas()
pd.to_csv('static/Input/bySingleDayRise.csv', index=False)
```

```
def load_SingleDayRise():
    csv_data = pd.read_csv("static/Input/bySingleDayRise.csv", sep=',')
    return csv_data

@app.route('/api/bySingleDayRise')
def bySingleDayRise():
    data = load_SingleDayRise()
    return data.to_json(orient='records')
```

6. Find rate of increase of covid cases only for Kansas State.

- Loaded the data into spark.
- Filtered the data on state to find all records of only Kansas state.
- Saved the data into csv and convert csv into json and passed the data to angular using API endpoints.

```
# Query 5: Kansas Cases by Date
query5 = spark.sql("SELECT submission_date as date, tot_cases FROM covid where state = 'KS'")
query5.show()
pd = query5.toPandas()
pd.to_csv('static/Input/byDateKS.csv', index=False)
```

```
def load_ByDateKS():
    csv_data = pd.read_csv("static/Input/byDateKS.csv", sep=',')
    return csv_data

@app.route('/api/byDateKS')
def byDateKS():
    data = load_ByDateKS()
    return data.to_json(orient='records')
```

7. Find the rate of increase of covid cases in the USA for Nov month.

- Loaded the data into spark.
- Filtered the date to include only nov records
- Saved the data into csv and convert csv into json and passed the data to angular using API endpoints.

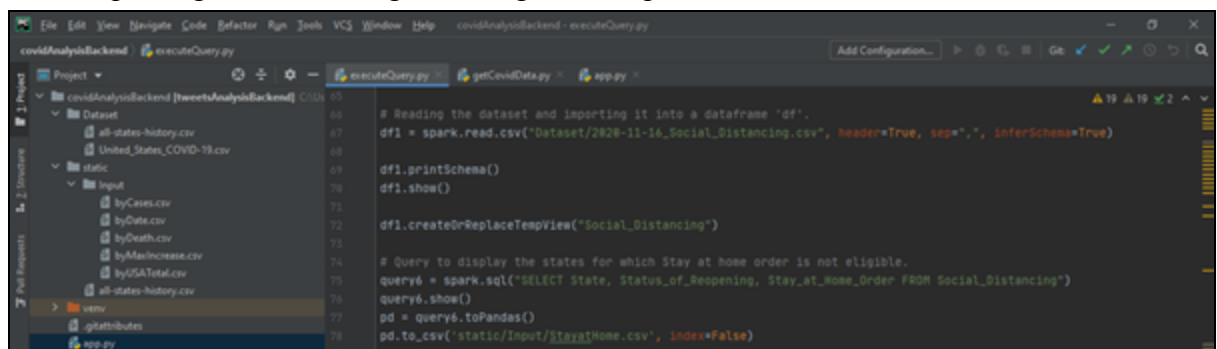
```
# Query 6: USA nov cases
query6 = spark.sql("select total_cases, total_deaths, date from covidUSA where date between '2020-11-01' and '2020-11-31'")
query6.show()
pd = query6.toPandas()
pd.to_csv('static/Input/byNovUSACases.csv', index=False)
```

```
def load_ByNovUSACases():
    csv_data = pd.read_csv("static/Input/byNovUSACases.csv", sep=',')
    return csv_data

@app.route('/api/byNovUSACases')
def byNovCases():
    data = load_ByNovUSACases()
    return data.to_json(orient='records')
```

8. Find the list of States in which Stay at Home order is lifted.

- Loaded the CSV dataset into the spark data frame.
- Filtered the data to fetch all the States where Stay at Home order is not in place and stored it in a data frame.
- Storing the data frame records into a CSV and converting CSV into JSON and passing the data to angular using API endpoints.



```
# Reading the dataset and importing it into a dataframe 'df'.
df1 = spark.read.csv("Dataset/2020-11-16_Social_Distancing.csv", header=True, sep=",", inferSchema=True)
df1.printSchema()
df1.show()
df1.createOrReplaceTempView("Social_Distancing")

# Query to display the states for which Stay at home order is not eligible.
query6 = spark.sql("SELECT State, Status_of_Respending, Stay_at_Home_Order FROM Social_Distancing")
query6.show()
pd = query6.toPandas()
pd.to_csv('static/Input/StayatHome.csv', index=False)
```

State	Status_of_Reopening	Stay_at_Home_Order
Alabama	Paused	Lifted
Alaska	Proceeding with R...	Lifted
Arizona	New Restrictions ...	Lifted
Arkansas	Paused	-
California	Proceeding with R...	Statewide
Colorado	New Restrictions ...	Lifted
Connecticut	Proceeding with R...	Lifted
Delaware	Proceeding with R...	Lifted
District of Columbia	Proceeding with R...	Lifted
Florida	Proceeding with R...	Lifted

```

19
20 def load_StayatHome():
21     csv_data = pd.read_csv("static/Input/StayatHome.csv", sep=',')
22     return csv_data
23

```

```

46
47 @app.route('/api/StayatHome')
48 def StayatHome():
49     data = load_StayatHome()
50     return data.to_json(orient='records')
51

```

9. Find the total number of States in which Quarantine is not in place for Travelers.

- Loaded the CSV dataset into the spark data frame.
- Filtered the data to fetch the total number of States where Quarantine is not in place for Travelers and stored it in a data frame.
- Storing the data frame records into a CSV and converting CSV into JSON and passing the data to angular using API endpoints.

```

80 # Query to display the total number of states with No Quarantine for Travelers.
81 query7 = spark.sql("SELECT COUNT(*) AS Number_of_States FROM Social_Distancing WHERE "
82                    "Mandatory_Quarantine_for_Travelers = 'Lifted'")
83 query7.show()
84 pd = query7.toPandas()
85 pd.to_csv('static/Input/StateswithNOQuarantine.csv', index=False)
86

```

Number_of_States
16

```

24 def load_StateswithNOQuarantie():
25     csv_data = pd.read_csv("static/Input/StateswithNOQuarantine", sep=',')
26     return csv_data
27

```

```

51
52 @app.route('/api/StateswithNOQuarantine')
53 def StateswithNOQuarantine():
54     data = load_StateswithNOQuarantie()
55     return data.to_json(orient='records')
56

```

10. Find the list of States where Bars has been reopened along with the mask requirement and gathering restriction data.

- Loaded the CSV dataset into the spark data frame.
- Filtered the data to fetch the total number of States where Quarantine is not in place for Travelers and stored it in a data frame.
- Storing the data frame records into a CSV and converting CSV into JSON and passing the data to angular using API endpoints.

```

86
87 # Query to display the states with Bars Reopened along with the face mask requirement and gathering ban data.
88 query8 = spark.sql("SELECT State, Large_Gatherings_Ban, Face_Covering_Requirement FROM Social_Distancing "
89                    "WHERE Bar_Closures = 'Reopened'")
90 query8.show()
91 pd = query8.toPandas()
92 pd.to_csv('static/Input/BarsReopened.csv', index=False)
93

```

State	Large_Gatherings_Ban	Face_Covering_Requirement
Alabama	Lifted	Required for Gene...
Alaska	Lifted	Required for Cert...
Arkansas	Lifted	Required for Gene...
Delaware	Expanded Limit to...	Required for Gene...
Florida	Lifted	Required for Cert...
Georgia	Expanded Limit to...	Required for Cert...
Indiana	New Limit on Lang...	Required for Gene...
Iowa	New Limit on Lang...	Required for Cert...
Kansas	Lifted	Required for Gene...
Minnesota	>10 People Prohib...	Required for Gene...

```

27
28 def load_BarsReopened():
29     csv_data = pd.read_csv("static/Input/BarsReopened", sep=',')
30     return csv_data
31

```

```

57 @app.route('/api/BarsReopened')
58 def BarsReopened():
59     data = load_BarsReopened()
60     return data.to_json(orient='records')
61

```

11. Find total number of hospitalizations by state

- Loaded CSV into spark data frame

- Grouped records by state and summed hospitalizations
- Stored data frame in csv for quick look up
- Created api endpoint to access hospitalization data

```
#Query 7: State with most hospitalizations
query7 =
df.groupBy("state").agg({"hospitalized":
"sum"}).withColumnRenamed("sum(hospitalized)",
"totalHospitalizations").orderBy("totalHospital
izations", ascending=False)
query7.show()
pd = query7.toPandas()
pd.to_csv('static/Input/byHospitalizationTotal.
csv', index=False)
```

```
87
88 @app.route('/api/byHospitalizationTotal')
89 def byHospitalizationTotal():
90     data = load_byHospitalizationTotal()
91     return data.to_json(orient='records')
92
```

12. Find total deaths by date

- Loaded CSV into spark data frame
- Grouped records by date and summed deaths
- Stored data frame in csv for quick look up
- Created api endpoint to access total death date data

```
#Query 8: Date with most deaths
query8 = df.groupBy("date").agg({"death":
"sum"}).withColumnRenamed("sum(death)",
"totalDeaths").orderBy("totalDeaths",
ascending=False)
query8.show()
pd = query8.toPandas()
pd.to_csv('static/Input/byDateDeaths.csv',
index=False)
```

```
93 @app.route('/api/byDateDeaths')
94 def byDateDeaths():
95     data = load_byDateDeaths()
96     return data.to_json(orient='records')
97
```


13. Find max increase of tests given by state

- Loaded CSV into spark data frame
- Grouped records by state and summed tests
- Stored data frame in csv for quick look up
- Created api endpoint to access hospitalization data

```
#Query 9: States with largest increase in testing
query9 =
df.groupBy("state").agg({"totalTestResultsIncrease":
"max"}).withColumnRenamed("max(totalTestResultsIncrease)",
"maxTestIncrease").orderBy("maxTestIncrease",
ascending=False)
query9.show()
pd = query9.toPandas()
pd.to_csv('static/Input/byLargestTestingIncrease.csv', index=False)
```

```
98 @app.route('/api/byLargestTestingIncrease')
99 def byLargestTestingIncrease():
100     data = load_byLargestTestingIncrease()
101     return data.to_json(orient='records')
102
```

Feature 3

Develop interactive front end applications for users to understand the analysis

1. Developed Angular 8 front end app to display the output of queries in graphical formats.
2. Queries outputs are accessed via python API endpoints in json format.

Feature 4

Create graphical visualization of all the analysis completed in feature 2

1. Separate visualization are created am4Charts angular library for each and every query endpoints.
2. Graphs are interactive hence can be further sorted and filtered.
3. This helps to draw an output out of the graph.

Results Evaluation

1. USA Covid records:

Shows the output of the 1st query (Feature 2) on browser.

Covid-19 Pandemic Data Analysis				
USA Covid Record				
Total Cases		Total Death		
994265		99214		
Total Cases per million	Total Hospitalized	Total ICU Patients		
9960.869	98691	9994		

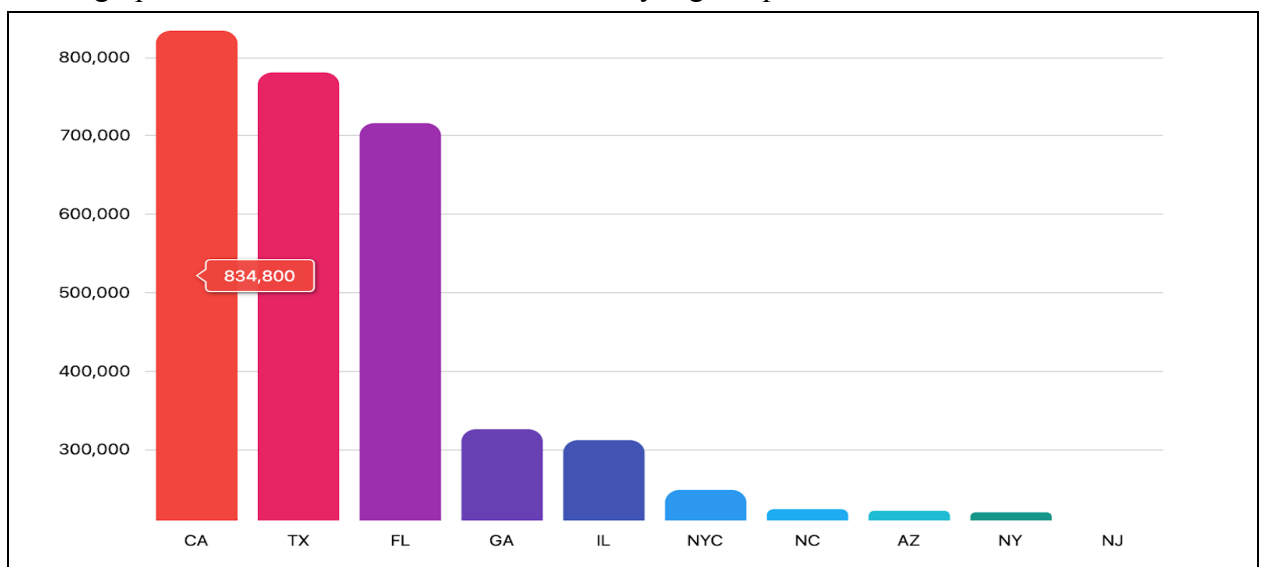
2. Find the total number of cases and deaths for each USA state.

This represents the total of cases and deaths recorded so far in each state. The table provides facility to sort and filter the data.

state	cases	deceased
state	cases	deceased
CA	834800	16361
TX	781794	16334
FL	717148	15068
GA	327407	7294
IL	313439	9159
NYC	250037	23879
NC	225397	3722
AZ	223401	5743
NY	221408	9092
NJ	211148	16161

3. Find top 10 states with max cases as of today

Below graph shows the states where there is a very high impact of covid.



4. Find single day max death for each state and the date when it happened

This table shows the single day max death recorded for each state and also shows the date when it was recorded.

state	deceased ▼	date
state	deceased	date
NYC	4585	4/15/20
NY	2185	4/6/20
NJ	1877	6/25/20
PA	678	5/22/20
TX	675	7/27/20
IL	382	5/21/20
MA	334	5/14/20
MI	270	6/5/20
CT	260	4/15/20
FL	257	7/31/20

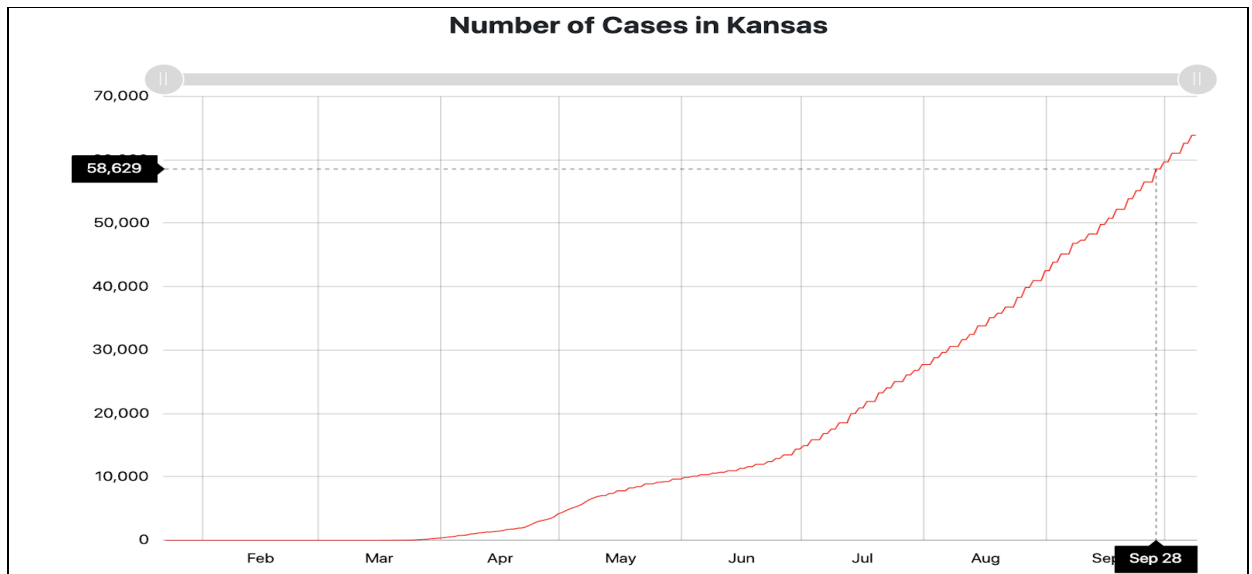
5. Find top 10 states with max single day rise of cases and the date when it happened

This table shows the single day max cases recorded for each state and also shows the date when it was recorded.

state	case ▼	date
state	case	date
NY	17844	4/6/20
TX	17820	9/22/20
FL	15135	7/12/20
CA	12807	7/22/20
NYC	8593	4/10/20
NC	6142	9/25/20
IL	5594	9/4/20
MI	5298	6/5/20
MA	4946	4/24/20
AZ	4877	7/1/20

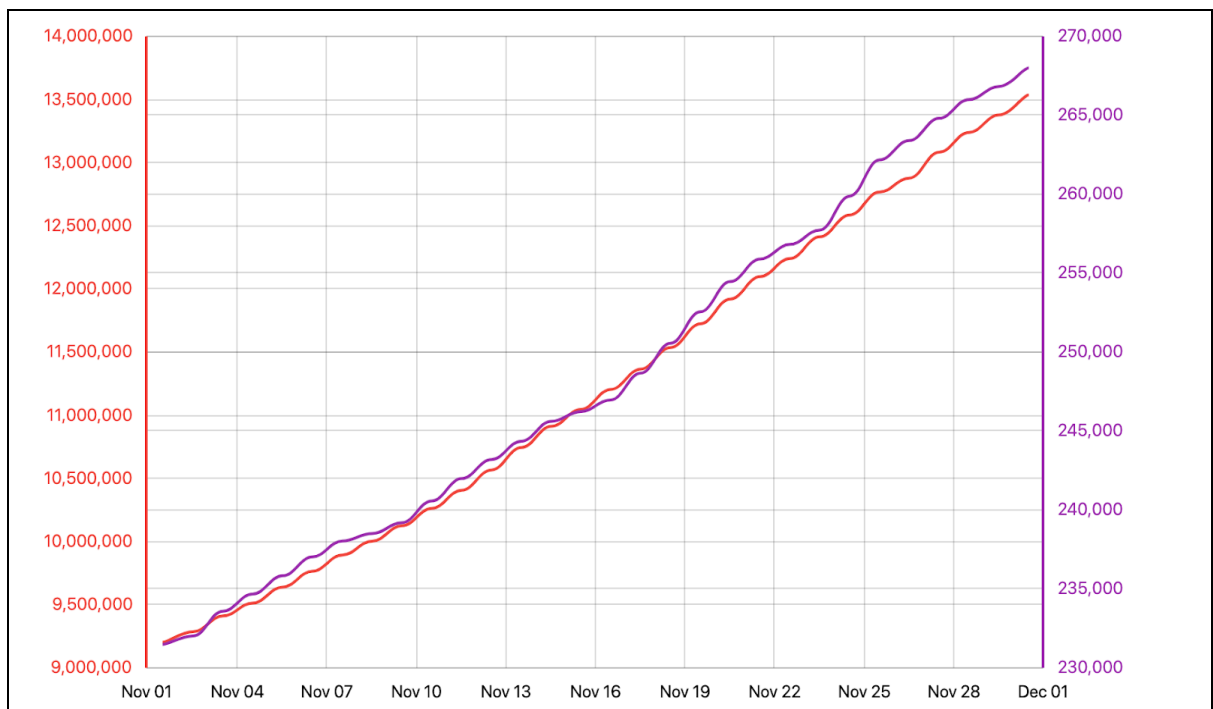
6. Find rate of increase of covid cases only for Kansas State.

The graph shows how covid impact increases linearly in Kansas state.



7. Find the rate of increase of covid cases and deaths in the USA for Nov month.

This shows the comparison between new cases and deaths recorded in the USA in the month of november.



8. Find the list of States in which Stay at Home order is lifted

The table shows the reopen and stay at home policies for each state

Reopen policy and stay at Home order		
state	Reopen policy	Stay at home order
state	Reopen policy	Stay at home order
United States	Paused (4); New Restrictions Imposed (40); Reopened (7)	Stay at Home Order Eased or Lifted (43); New Stay at Home Order in Place (1); No Action (6)
Alabama	Paused	Lifted
Alaska	Reopened	Lifted
Arizona	New Restrictions Imposed	Lifted
Arkansas	Paused	-
California	New Restrictions Imposed	Counties of High Transmission (New)
Colorado	New Restrictions Imposed	Lifted
Connecticut	New Restrictions Imposed	Lifted
Delaware	New Restrictions Imposed	Lifted
District of Columbia	New Restrictions Imposed	Lifted

9. Find the total number of States in which Quarantine is not in place for Travelers.

The table shows the states where mandatory quarantine is required or not.

States with mandatory quarantine lifted	
state	Mandatory Quarantine
state	Mandatory Quarantine
United States	no
Alabama	no
Alaska	no
Arizona	yes
Arkansas	yes
California	no
Colorado	no
Connecticut	no
Delaware	yes
District of Columbia	no

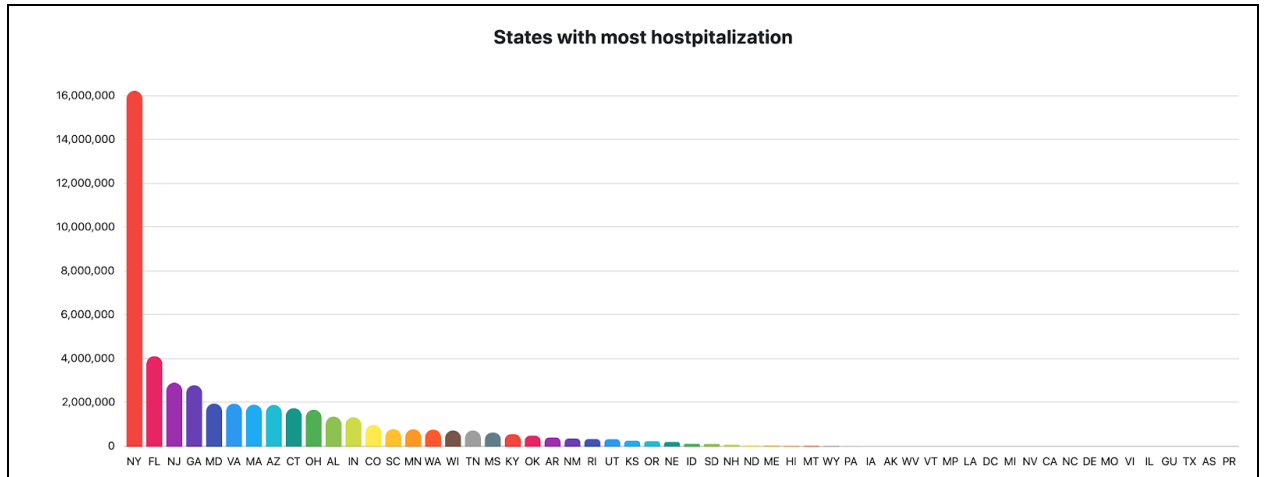
10. Find the list of States where Bars has been reopened along with the mask requirement and gathering restriction data

That table lists the policies for large gathering and face covering

Bar reopen policy		
state	Large Gathering	Face Covering
state	Large Gathering	Face Covering
Alabama	No Limit	Required for General Public
Alaska	No Limit	Required for Certain Employees
Arkansas	No Limit	Required for General Public
Florida	No Limit	-
Georgia	>50 Prohibited	Required for Certain Employees; Allows Local Officials to Require for General Public
Indiana	>25 Prohibited	Required for General Public
Iowa	>25 Prohibited	Required for General Public
Kansas	No Limit	Required for General Public
Minnesota	All Gatherings Prohibited	Required for General Public
Missouri	No Limit	-

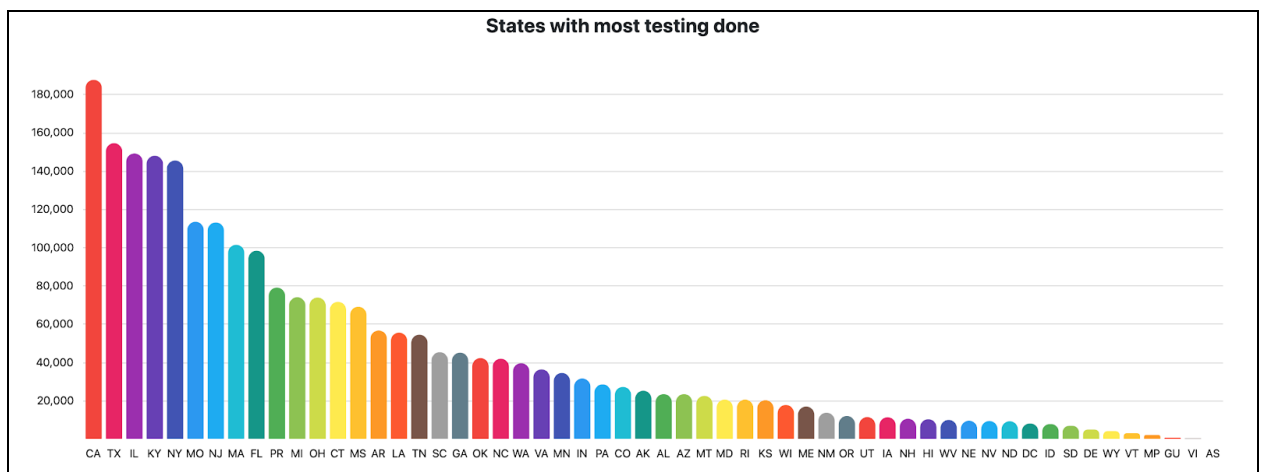
11. Find total number of hospitalizations by state

The graph shows the states where hospitalization rate is very high



12. Find max increase of tests given by state

The graph shows the states and their testing records



Conclusion

We have developed an interactive application for users to see covid impact on each state of the USA. As the impact and data is growing, it is very important to perform effective analysis on dataset. Hadoop and spark have provided an easy way to process large unstructured data and convert it into structured and perform analysis on top of that. In this project we completed analysis using Hadoop, Spark, Hive and easily transferred relevant data to the front end application.

Future Work

We have imported csv data for this project. This can be enhanced to collect real time data if provided by a government site. As the covid-19 vaccination is on its way, this project can be

enhanced to collect vaccination data and analyze how effective the vaccine is to fight with covid-19 virus.

Project Management:

Work completed:

We did analysis on different datasets available and identified the relevant dataset as per our requirement. We used spark big data technique to preprocess the data, create spark dataframe and perform different query operations to extract relevant data. In order to visualize the output in graphical format we have converted query output to json and passed it to angular front end application via API endpoints and used charts library to create graphs on data.

Responsibility (Task, Person)

Neha Navgale:

1. Identified the correct datasets.
2. Developed python flask API.
3. Preprocessed the data in spark using pyspark library
4. Wrote spark queries (mentioned in feature 2) on top of spark dataframe.
5. Developed Angular front end application for visualization.
6. Developed different graphs using angular library by consuming data from python API
7. Contributed to the project report.

Christian Barlow:

1. Create python spark queries for hospitalization by state, death by date, and tests by state
2. Create python api routes for hospitalization by state, death by date, and tests by state
3. Contributed to project report

Krishnapriya Akula:

1. Loaded the CSV dataset and imported the data into a new table(created using tmpview).
Preprocessed the same data in spark using pyspark library.
2. Developed and executed some spark queries on top of spark dataframe mentioned in Feature 2.
3. Using the created API, connected the extracted dataframe to display the data in the API.
4. Contributed to the project report.

Jayadeep Kumar:

1. Identified the dataset(Social Distancing) to get the insights of COVID regulation updates with respect to each state.
2. Preprocessed the data in spark using pyspark library and stored the insight data in individual data frames.

3. Linked these data frames with the API created by Neha.
4. Contributed to the project report.

Andrew Poitras

1. Helped find datasets for analyzation
2. Analyzed use of Apache Solr for final project
3. Developed ability to pull csv files from the internet for the relevant datasets and turned them into dataframes (couldn't get it hooked up with spark streaming properly to update data in real time, future goal)
4. Contributed to the project report

Contributions (members/percentage)

We equally contributed to the project. We divided the tasks and completed our tasks on time.

Issues/Concerns

1. Spark created all the columns as string data type while reading csv. String columns were causing issues with using groupBy agg operation on columns. This issue was resolved by creating a schema and defining data type for each column while import.

Story Telling:

Chapter 1 (Life):

Define scope or domain where the use case is relevant or prevalent?

The purpose of this study is to examine the impact of covid-19 in different regions of the USA and on different age groups. Also, to study how it has impacted health care response and economy.

What is the main story?

The main story is around what information people need related to covid-19 and its impact and how insights can help to take preventive measures.

Who are the characters or people in the main story?

I am a student in UMKC university. I enjoy attending classes and labs, hanging out with friends and meeting in the library for study. But around 7 months ago, a novel coronavirus hit the USA and since then everyone's life has changed dramatically. As the number of cases continues to rise in the USA, the coronavirus outbreak is having profound impacts on personal lives. The fear of coronavirus has made me uncomfortable attending crowded parties, eating out in restaurants, going out for grocery stores and visiting friends.

What happens?

COVID-19 has rapidly affected our day to day life, businesses, disrupted the world trade and movements. As a student I am curious to learn on how it has impacted the job market and what businesses are closing. I also need statistics to learn its trend in different regions of the USA and learn how it has impacted hospital resources.

Where?

This is a global problem. I am looking for statistics and some informative insights of covid impact on the USA.

When?

The pandemic started around March 2020 in the USA.

Why?

As the virus is contagious, the lack of social distancing has made this disease a pandemic. In order to avoid further spread of disease, strict measures have been taken in every sector which resulted in loss of business and slowing down of the economy. The places where preventive measures were not followed, have now become the hotspots. This study could help to understand the rate of spread in a particular region and its impact on the economy.

How?

As a precautionary measure, I check for covid impact before travelling to a new place. I also google for understanding the health care facilities provided near by my area. I see restaurants, business closing and can understand its impact on job opportunities. But I didn't find a reliable source of information which can provide deep insights of all at one place. Hence this project will help to provide answers to all such problems at one place.

Chapter 2 (Data):**Who:**

The dataset is about the people who were impacted due to covid 19 in different states and territories of the USA. They are the representative of the main characters of the story as this will help to understand the trend of covid -19 impact on different locations of the USA. The data does not contain any identifying information nor does it have risks of disclosing identifiable information, it is mostly anonymous geographical and medical information.

What:

The dataset records the number of people tested positive, recovered, deceased, hospitalized in each and every state on each day.

When:

The data is collected everyday from official sites of each state and placed at covidtracking.com. As part of the first and second increment, we collected the data in csv format but for next increment we will be consuming API to get the real time data. Covid-19 started spreading in the USA from March 2020, hence the data is available from March and gets updated everyday.

Where:

Global data is available but we narrowed down the scope of our project to the USA hence we have collected data for all the states and territories of the USA. The state variable in the dataset geographically separates out the data.

Why:

The data is very crucial to understand the ongoing pandemic and its effect on every sector. The data is collected to analyze, understand and identify the gaps in preventive measures taken.

Chapter 4 (User):**Who:**

The main character here is a college student who is not able to attend offline classes, cannot meet friends or go anywhere without getting impacted by novel coronavirus. This is a story of a student who seeks for the latest information on coronavirus around his area, state and country before travelling to any place.

What:

The front end application can help any user to see the latest cases updates on covid in the USA and all states of the USA. This application will help to quickly find answers to covid related questions like total cases, state affected most, how is the cases rate etc. The data is represented in the form of interactive graphs and tables on which filtering and sorting can be easily applied.

When:

This will be a web application which can be accessed at any time from any geographical location.

Where:

This application can be deployed on heroku and can be accessed through any device and any browser.

Why:

The visualization is very useful to understand the impact of covid on each and every state of the USA. Line graphs will show the rate of increase or decrease over the period of time and grid display will help to search, sort the data to check data for a particular state.

How:

The people can use the application by accessing it on any browser.

Chapter 5 (The Society):

Who:

Everyone in the world has impacted due to covid-19. The data collected is globally. The data shows the number of people impacted of different age groups. It also shows how the government is taking efforts to fight the virus. As a team we have collected the USA data and performed data sampling.

What:

The data is very generic and doesn't reveal any person's medical information. The data represents the society and different age groups collectively. The data is collected from government's verified sources

When:

Our dataset doesn't include any confidential information. So there will not be any social or cultural impact. Though, the data accuracy should be verified from time to time.

Where:

The dataset we collected is globally available, so there will not be any concerns related to primary and security of data. Also we have collected data from sources which guarantees data accuracy.

Why:

In this analysis, there will not be any concerns related to data privacy and security because the analysis doesn't include any person's confidential or medical information.

References:

<https://spark.apache.org/docs/2.2.0/sql-programming-guide.html>

https://www.tutorialspoint.com/spark_sql/spark_sql_dataframes.htm

<https://sparkbyexamples.com/spark/spark-dataframe-where-filter/>

<https://covidtracking.com/>

<https://data.cdc.gov/Case-Surveillance/United-States-COVID-19-Cases-and-Deaths-by-State-o/9mfq-cb36>

<https://ourworldindata.org/mortality-risk-covid>

<http://ocel.ai/>