Lab3: Parameterized clustering and analysis of global events on GDELT structured dataset

UBIT Name: bthorat Person number: 50290581
UBIT Name: harshalg Person number: 50290606

Problem statement:

Build an interactive web based data driven application which enables users to visualize impact of an event in interest to an event in future which is a consequence of that event. E.g. Users wants to analyze the effect of recent acquisitions by a company on it's shares in global market. We will gather the events related to the company's shares for a period of time specified by user (typically 1 year) after the acquisition date, across various geographic locations.

Here, measure the effect of event under subject for the time period that user has specified. Provide user the ability to to set different parameters related to events he is interested in and generate output based on those parameters. These parameters will be geographic locations the user is focussing on, the specific keywords to look for or focus on certain level of positive or negative sentiment across populace of various locations.

We are going to collect event samples that we are interested in from a free public structured dataset of events called Global Database of Events Language and Tones (GDELT). To fetch samples based on user preference or filtering, we are going to build SQL queries on Google BigQuery platform.

In order to make most sense out of the data so collected, run clustering algorithms like K-Means GDELT dataset collects events from all the printed media across the world. The data gathered related to a certain event across the globe for long periods can be tremendous. To apply iterative algorithms like K-Means clustering on such a big dataset can be computationally demanding. Hence, we are going to utilize the power of Spark. The clustering algorithm divides the dataset into distinct set of clusters which can visualized in a multi-dimensional space.

The clustered visualization and representation can provide users insights on particular focus areas they need to put emphasis on in order to facilitate an improved decision process. Each sample of event is plotted as a distinct point in a multi-dimensional space. Each axis in this space represents a feature from the dataset. Depending on the centers of these clusters, users can assess the most dominant parameters related to an event (features in the dataset) that are affected. This can help decision makers in an organization to narrow down their scope of analysis to a specific set of attributes related to the event.

The GDELT dataset:

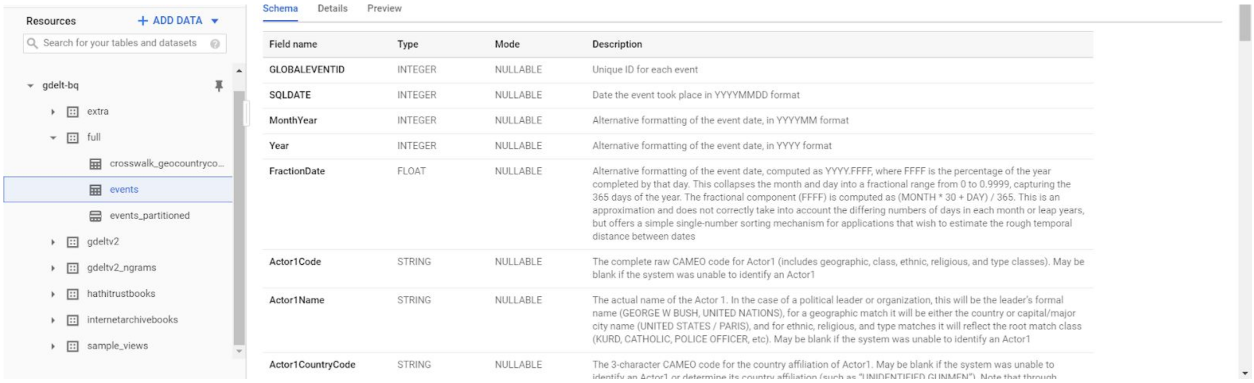
Global Database of Events Language and Tones (GDELT) is a free public dataset available on the internet which monitors the world's broadcast, print, and web news from nearly every corner of every country in over 100 languages and identifies the people, locations, organizations, themes, sources, emotions, counts, quotes, images and events, creating a free open platform for computing on the entire world. The dataset is a structured dataset, which means that it has a fixed size of schema with fixed number of features.

Easiest and best source to collect GDELT data samples is the Google BigQuery platform. We can run SQL queries to collect and filter data according to our needs. We can build SQL queries based on values of various attributes user has entered or selected from the interactive web interface. We can execute these pre-built queries on BigQuery interface and save the results in CSV or JSON format.

We are only interested in following parameters of the events table:

```
SQLDATE ,MonthYear, Year, FractionDate, Actor1Name, Actor2Name,
Actor1Geo_CountryCode ,AvgTone , GoldsteinScale, NumArticles,
NumMentions
```

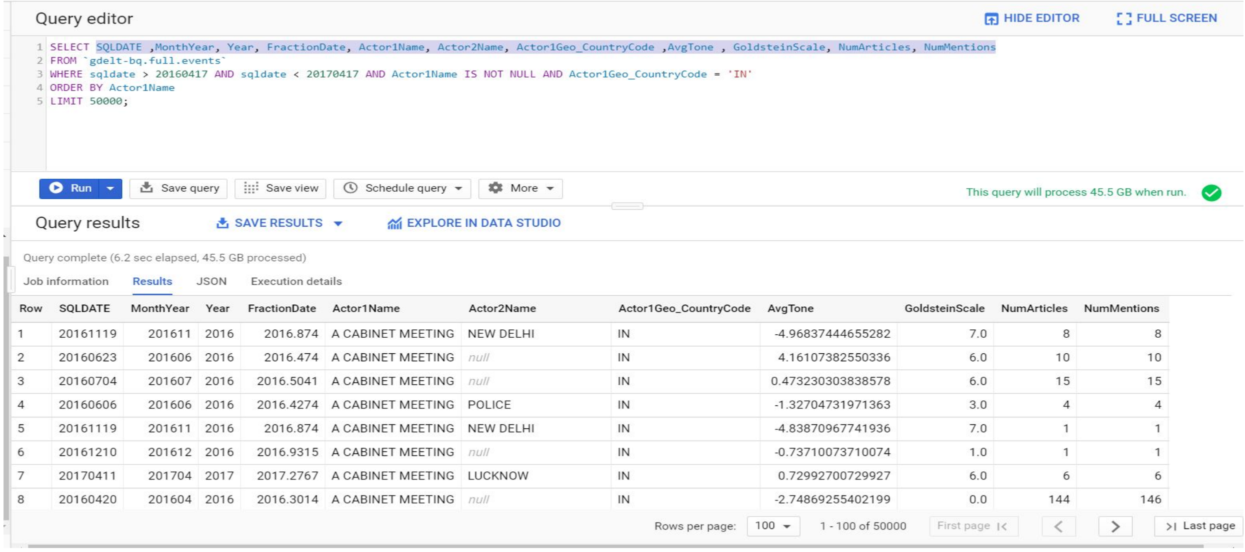
Additional details about these parameters can be found by clicking on the events table in BigQuery as follows:



Field name	Type	Mode	Description
GLOBALEVENTID	INTEGER	NULLABLE	Unique ID for each event
SQLDATE	INTEGER	NULLABLE	Date the event took place in YYYYMMDD format
MonthYear	INTEGER	NULLABLE	Alternative formatting of the event date, in YYYYMM format
Year	INTEGER	NULLABLE	Alternative formatting of the event date, in YYYY format
FractionDate	FLOAT	NULLABLE	Alternative formatting of the event date, computed as YYYY.FFFF, where FFFF is the percentage of the year completed by that day. This collapses the month and day into a fractional range from 0 to 0.9999, capturing the 365 days of the year. The fractional component (FFFF) is computed as (MONTH * 30 + DAY) / 365. This is an approximation and does not correctly take into account the differing numbers of days in each month or leap years, but offers a simple single-number sorting mechanism for applications that wish to estimate the rough temporal distance between dates
Actor1Code	STRING	NULLABLE	The complete raw CAMEO code for Actor1 (includes geographic, class, ethnic, religious, and type classes). May be blank if the system was unable to identify an Actor1
Actor1Name	STRING	NULLABLE	The actual name of the Actor 1. In the case of a political leader or organization, this will be the leader's formal name (GEORGE W BUSH, UNITED NATIONS), for a geographic match it will be either the country or capital/major city name (UNITED STATES / PARIS), and for ethnic, religious, and type matches it will reflect the root match class (KURD, CATHOLIC, POLICE OFFICER, etc). May be blank if the system was unable to identify an Actor1
Actor1CountryCode	STRING	NULLABLE	The 3-character CAMEO code for the country affiliation of Actor1. May be blank if the system was unable to identify an Actor1 or determine its country affiliation (such as "UNIDENTIFIED GUNMEN"). Note that through

Following is the sample query and its output on BigQuery. This output can be saved locally or on google drive in CSV or JSON format. The query extracts dates in multiple formats, Actor attributes, geological location, goldstein scale, number of mentions and articles, average tone for a particular event filtered by SQLDATE:

```
SELECT SQLDATE ,MonthYear, Year, FractionDate, Actor1Name,
Actor2Name, Actor1Geo_CountryCode ,AvgTone , GoldsteinScale,
NumArticles, NumMentions
FROM `gdelt-bq.full.events`
WHERE sqldate > 20160417 AND sqldate < 20170417 AND Actor1Name IS NOT
NULL AND Actor1Geo_CountryCode = 'IN' AND Actor1Name = ’Apple Inc’
ORDER BY Actor1Name
LIMIT 50000;
```



```
SELECT SQLDATE ,MonthYear, Year, FractionDate, Actor1Name, Actor2Name, Actor1Geo_CountryCode ,AvgTone , GoldsteinScale, NumArticles, NumMentions
FROM `gdelt-bq.full.events`
WHERE sqldate > 20160417 AND sqldate < 20170417 AND Actor1Name IS NOT NULL AND Actor1Geo_CountryCode = 'IN'
ORDER BY Actor1Name
LIMIT 50000;
```

This query will process 45.5 GB when run.

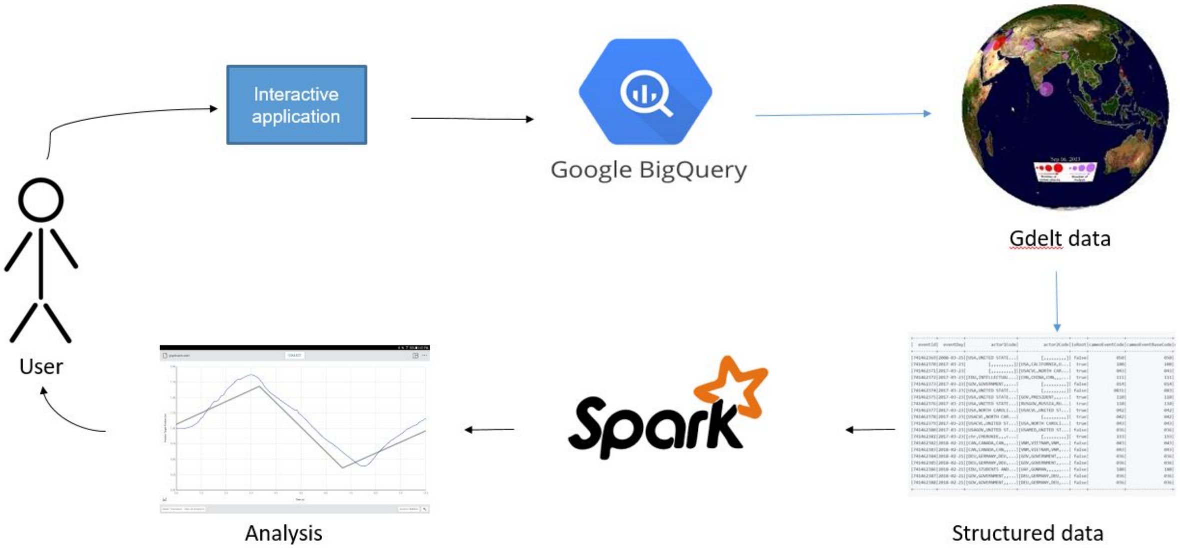
Query complete (6.2 sec elapsed, 45.5 GB processed)

Row	SQLDATE	MonthYear	Year	FractionDate	Actor1Name	Actor2Name	Actor1Geo_CountryCode	AvgTone	GoldsteinScale	NumArticles	NumMentions
1	20161119	201611	2016	2016.874	A CABINET MEETING	NEW DELHI	IN	-4.96837444655282	7.0	8	8
2	20160623	201606	2016	2016.474	A CABINET MEETING	*null*	IN	4.16107382550336	6.0	10	10
3	20160704	201607	2016	2016.5041	A CABINET MEETING	*null*	IN	0.473230303838578	6.0	15	15
4	20160606	201606	2016	2016.4274	A CABINET MEETING	POLICE	IN	-1.32704731971363	3.0	4	4
5	20161119	201611	2016	2016.874	A CABINET MEETING	NEW DELHI	IN	-4.83870967741936	7.0	1	1
6	20161210	201612	2016	2016.9315	A CABINET MEETING	*null*	IN	-0.73710073710074	1.0	1	1
7	20170411	201704	2017	2017.2767	A CABINET MEETING	LUCKNOW	IN	0.72992700729927	6.0	6	6
8	20160420	201604	2016	2016.3014	A CABINET MEETING	*null*	IN	-2.74869255402199	0.0	144	146

Rows per page: 100 1 - 100 of 50000 First page Last page

The solution model:

Block diagram for **Parameterized clustering and analysis of global events on GDELT structured dataset**



Above is the architectural block diagram of the proposed solution. User interacts to a web based application to set parameters related to an event he/she is interested in. User can also specify the number of clusters that the user wants to clusterize the data into. User hits an analyze button to generate the analysis. In the back-end, the application builds a SQL query based on the parameters that user has set. E.g. User can set geolocation, time frame, average tone, number of mentions etc. The query so built is executed using Google BigQuery api. The api provides the resultant output in a data frame. The data frame can be stored in a csv file. This csv file be read in a scala program using Spark.

The scala program which is executed using Spark, runs K-means clustering algorithm on the read csv file. Spark's MLlib library provides API for clustering algorithms like K-Means. We will be using this API to cluster out data into distinct clusters and calculate their centers. The output so generated will be shown to the user on the application output screen.

Pre-processing of data:

Please note that certain features of dataset like `Actor1Name`, `Actor1Geo_CountryCode` don't have a numerical data type. Before running clustering algorithm, we will have to represent these string values to numeric values. We can substitute each character in a string with it's lexicographical position in the alphabet. E.g. a -> 1, e -> 5. In this way, each string can be represented with a unique number. The features so represented are now ready to be fed as input to K-means api of MLlib.

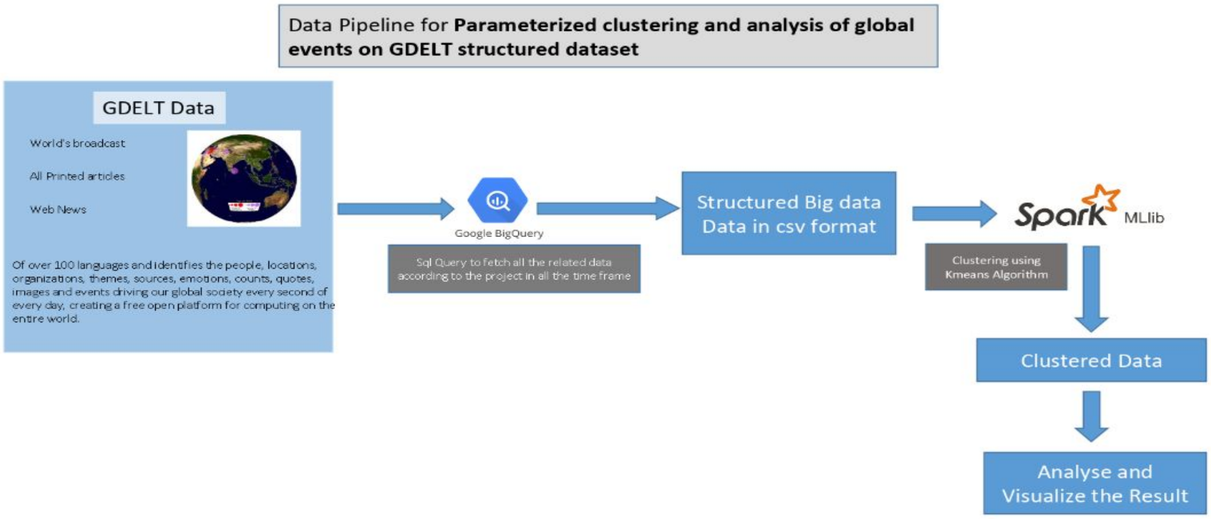
Visualization:

The number of parameters that user want to analyze for a particular events can be greater than 3. This means that after gathering of data, the plotting of data samples can involve multi-dimensional feature space. After running clustering algorithm, it is difficult to visualize clusters in a space with more than 3 dimensions. Hence, we will ask user to select the three parameters that the user want the visualization on. Hence, we will reduce the number of axes for the data points to plot on and these axes will be specified by the user.

In the case that the user does not specify any parameters to visualize on, the application will just display the coordinates of centers of different clusters. The application will also display the dimensions of clusters like width along a particular axis.

The visualization and the displayed cluster centers and dimensions will enable user to analyze the the most dominant parameters related to an event through a certain time frame. Based on parameters of these dominant features, decision makes of an corporation can make more informed and calculated future decisions.

Following is the data-flow pipeline of proposed application:



Data Pipeline for **Parameterized clustering and analysis of global events on GDELT structured dataset**
GDELT Data
World's broadcast
All Printed articles
Web News
Of over 100 languages and identifies the people, locations, organizations, themes, sources, emotions, counts, quotes, images and events driving our global society every second of every day, creating a free open platform for computing on the entire world.
Google BigQuery
Sql Query to fetch all the related data according to the project in all the time frame
Structured Big data Data in csv format
Spark MLlib
Clustering using Kmeans Algorithm
Clustered Data
Analyse and Visualize the Result

SPARK code snippet (SCALA) :

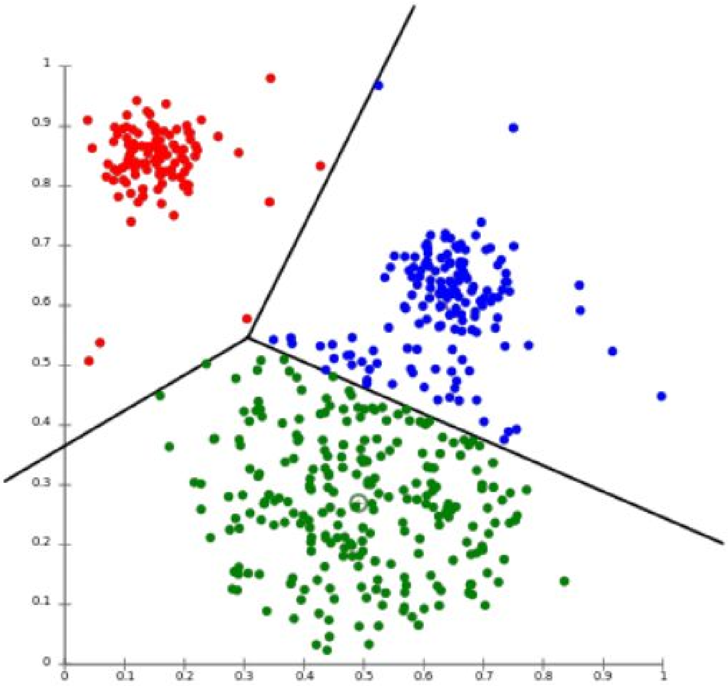
```
package org.apache.spark.examples.mllib

import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.mllib.clustering.{KMeans, KMeansModel}
import org.apache.spark.mllib.linalg.Vectors

object KMeansExample {

  def main(args: Array[String]) {

    val conf = new SparkConf().setAppName("KMeansExample")
    val sc = new SparkContext(conf)

    // $example on$
    // Load and parse the data
    val data = sc.csvFile("data/mllib/gdelt_data.csv")
    val parsedData = data.map(s => Vectors.dense(s.split('
').map(_.toDouble))).cache()

    // Cluster the data into threeclasses using KMeans
    val numClusters = 3    val numIterations = 20
    val clusters = KMeans.train(parsedData, numClusters, numIterations)

    // Evaluate clustering by computing Within Set Sum of Squared Errors
    val WSSSE = clusters.computeCost(parsedData)
    println(s"Within Set Sum of Squared Errors = $WSSSE")

    // Save and load model
    clusters.save(sc,
"target/org/apache/spark/KMeansExample/KMeansModel")
    val sameModel = KMeansModel.load(sc,
"target/org/apache/spark/KMeansExample/KMeansModel")
    // $example off$

    sc.stop()
  }
}
```

Sample visualizations and output:

The parameters of cluster center coordinate points can be features of GDELT dataset. Upon inspecting a particular center point, user can know the values of different event parameters corresponding to the cluster center.

Cluster centers:

```
Cluster 1                       center (25,25,100)
Width along dim 1: 10  Width along dim2: 15          Width  along  dim
3: 25
Cluster 2                       center (75,50,10)
Width along dim 1: 5  Width along dim2: 45      Width along dim 3: 30
Cluster 3                       center (130,70,90)
Width along dim 1: 5  Width along dim2: 50      Width along dim 3: 35
```



Sample visualization

Summary:

An analysis can be performed using data clustering to enable user make more informed decisions about a company aspect. Structured data related to an event in subject can be gathered from across the internet and after preprocessing, it can be clusterized using K-Means algorithm on spark. These clustering results can be integrated into an web application for user analysis.

Take-away points:

- Learnt data gathering from public dataset like GDELT.
- Learnt to leverage the power of Spark to run various algorithms on a dataset.
- Learnt to integrate the Spark output into a data driven interactive web application.

Future scope:

GDELT dataset is very vast and easily accessible. So, with further analysis in each field, by customizing to the needs of the user, we can modify our search query of the Gdelt dataset and get suitable results. These type of project on Gdelt dataset are currently on-going and further research is being done. Hence, this interactive application we believe, can be implemented.

REFERENCES:

1. https://www.gdeltproject.org/data.html
2. https://blog.gdeltproject.org/did-the-arab-spring-really-spark-a-wave-of-global-protests/
3. https://spark.apache.org/docs/latest/
4. https://www.gdeltproject.org/data.html#googlebigquery
5. https://cloud.google.com/bigquery/docs/
6. https://spark.apache.org/docs/latest/ml-guide.html
7. https://github.com/aamend/spark-gdelt
8. https://blog.gdeltproject.org/getting-started-with-gdelt-google-cloud-datalab-simple-network-visualizations/