2017

Health and Diet Analysis

SPARK-ANALYSIS USING JAVA

NIKET PATEL

NIKET PATEL | A20384264

Table of Contents

1. Introduction

Adapting a healthy Lifestyle can be both challenging and rewarding. In Today's fast lifestyle, people become lazy and unaware about their dietary problems and its bad effect on their life. Due to the fast life, people prefer junk food over healthy food and avoid exercises, which can be cause of many dangerous disease to them. So, In the real time, it is required to analysis for a country their people health awareness and their diary.

The main agenda of this project to do analysis of dietary and health awareness of people using twits collected from twitter and show it into graphical way. To visualize the health pattern across the globe, I have collected 400k tweets data (size: 1GB+) from the twitter using Amazon AWS server and stored in JSON file. With these collected data, I will analyze the habit of people about health, food, sickness and their preference for food and exercise.

2. System Setup:

Operating System:	Windows 10
RAM:	12 GB
Tool/IDE:	Eclipse Neon
Server:	Amazon AWS
Browser:	Google
Language:	Java, Maven Repository
Visualization:	D3.js, High Charts, Amcharts
Big Data Technology:	Apache Spark
Web Technology:	HTML, JSP, Servlet, Bootstrap, AJAX, JQuery, CSS.

3. Technology Overview:

1. Apache Spark

Apache Spark is an open source, Hadoop-compatible, fast and general purpose cluster-computing platform. It also includes Spark SQL, Spark Streaming, Spark MLib (used for Machine Learning), GraphX (used for graph processing). Spark is capable of processing a program 100x faster than Hadoop MapReduce in Memory and 10x faster in Disk. Spark is written in scala and runs on JVM (Java Virtual Machine).

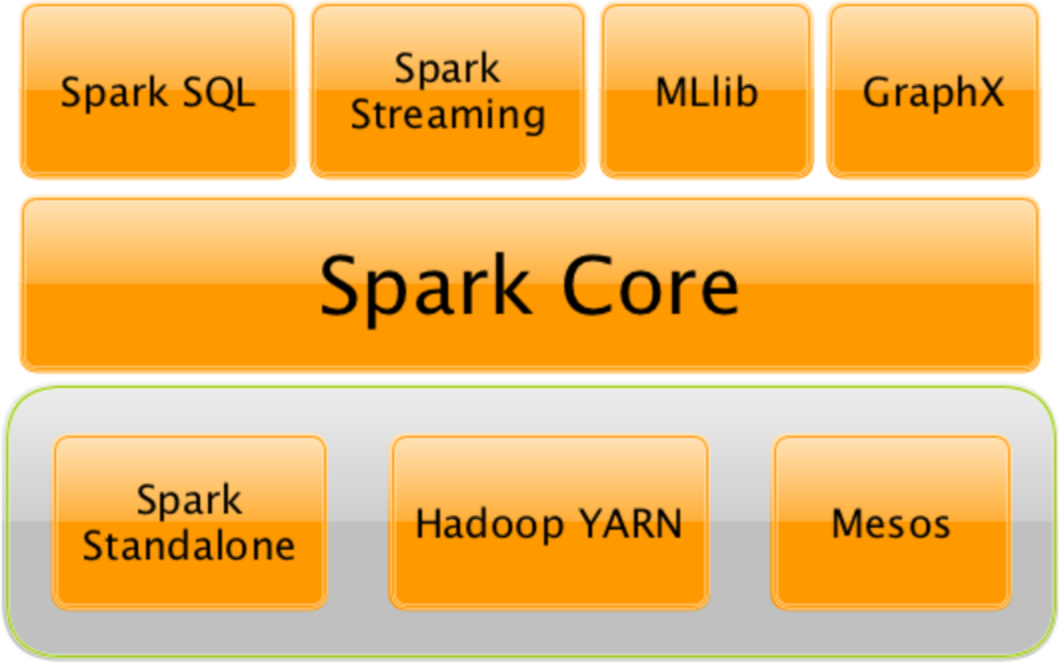


Figure 1 Spark Architecture

Spark contains application programming interface that defines Resilient Distributed Dataset(RDD) which represents collection of items distributed on multiple cluster node and compute nodes in parallel. It was developed due to inefficiency of Hadoop MapReduce cluster computing paradigm. Which forces linear dataflow structure on distributed programs. MapReduce program reads data from disk, map a function on the data, reduce the results of the map and stores a result back to disk.

Following are some of the MapReduce shortcomings:

Aspects	MapReduce	Spark
Difficulty	MapReduce is difficult to program and needs abstraction	Spark is easy to program and does not require abstraction.
Interactive mode	There is no such mode.	Bach Processing and interactive mode available.
Streaming	Hadoop MapReduce just get to process a batch of large stored data.	Spark can be used to modify in real time though spark streaming.
Performance	MapReduce doesn't leverage the memory of the Hadoop cluster to the maximum	Spark has been said to execute batch processing jobs about 10x to 100x faster than MapReduce.
Latency	MapReduce is disk oriented completely.	Spark ensures the lower latency communications by catching the partial results across its memory of distributed workers.
Ease of Coding	Writing Hadoop MapReduce pipelines is complex and lengthy process.	Writing spark code is always more compact.

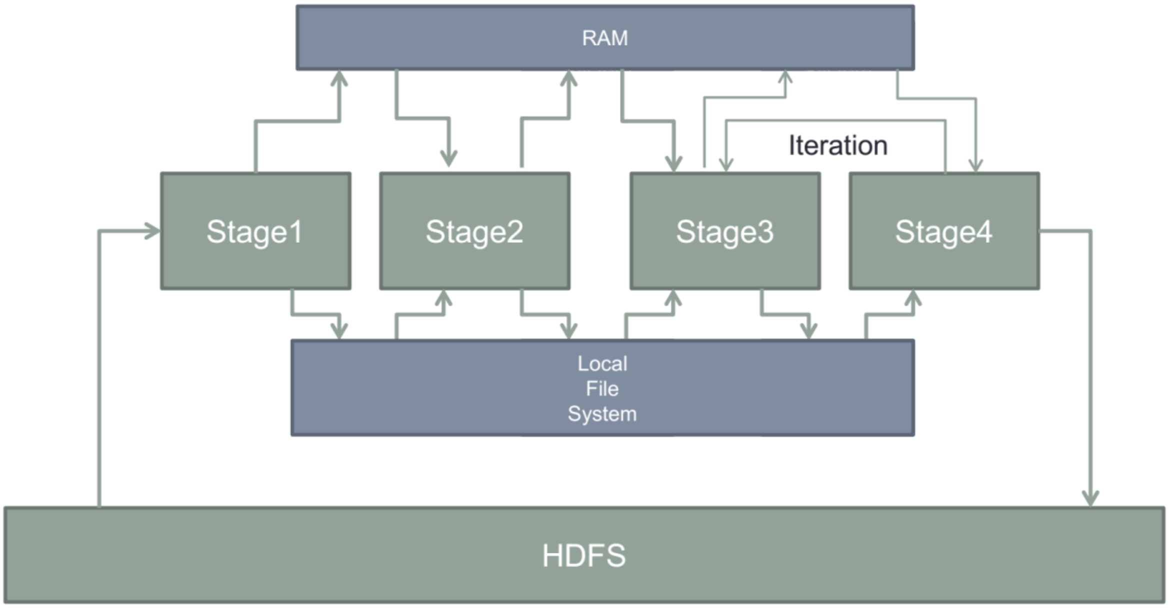
Datasets:

A Dataset is a distributed collection of data. A Datasets can be constructed using JVM Objects and then manipulated using functional transformation such as Map, flatMap, filter etc.

DataFrame:

DataFrame is a Datasets organized into named columns. It is conceptually similar to relational database. DataFrames can be constructed from structured data files, tables in Hive, external databases or existing RDDs.

Spark File Based Processing diagram.



Sample Code:

```
SparkConf sparkConf = new SparkConf().setAppName("Country  Servlet" ).setMaster(
"local").set("spark.driver.allowMultipleContexts", "true");

SparkContext ctx = new SparkContext(sparkConf);
             SQLContext sc = new SQLContext(ctx);

DataFrame df = sc.read().json(inputFile);
df.registerTempTable("tweets");

DataFrame data = sc.sql("select place.country from tweets");
JavaPairRDD<String, Integer> ones = data.toJavaRDD().mapToPair(new PairFunction<Row,
String, Integer>() {
                    @Override
                    public Tuple2<String, Integer> call(Row r) throws Exception {
                           // TODO Auto-generated method stub
```

```
                String i = r.getString(0);
                return new Tuple2<String, Integer>(i, 1);
            }
        });

        JavaPairRDD<String, Integer> counts = ones.reduceByKey(new
Function2<Integer, Integer, Integer>() {

            @Override
            public Integer call(Integer i1, Integer i2) throws Exception {
                // TODO Auto-generated method stub
                return i1 + i2;
            }

        });
```

Map Function will find the country and convert it to tuple <"Country",1> form and Reduce function will sum the country grouped by key field.

2. Maven

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Maven's primary goal is to allow developer to comprehend the complete state of development effort in the shortest period of time.

Maven's Objective:

➔Making the build process easy

➔providing uniform build system

➔Providing quality project information

➔Allowing transparent migration to new features

Sample pom.xml file:

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>TwitterAnalysis</groupId>
    <artifactId>TwitterAnalysis</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <build>
        <sourceDirectory>src</sourceDirectory>
        <plugins>
            <plugin>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.5.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
            <plugin>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.0.0</version>
                <configuration>
                    <warSourceDirectory>WebContent</warSourceDirectory>
                    <failOnMissingWebXml>false</failOnMissingWebXml>
                </configuration>
            </plugin>
        </plugins>
    </build>
    <dependencies>
        <dependency>
            <groupId>com.sparkjava</groupId>
            <artifactId>spark-core</artifactId>
            <version>2.3</version>
        </dependency>
        <dependency> <!-- Spark dependency -->
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_2.10</artifactId>
            <version>1.5.1</version>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-sql_2.10</artifactId>
            <version>1.5.1</version>
        </dependency>
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-csv</artifactId>
            <version>1.0</version>
        </dependency>
    </dependencies>
</project>
```

3. Data Visualization

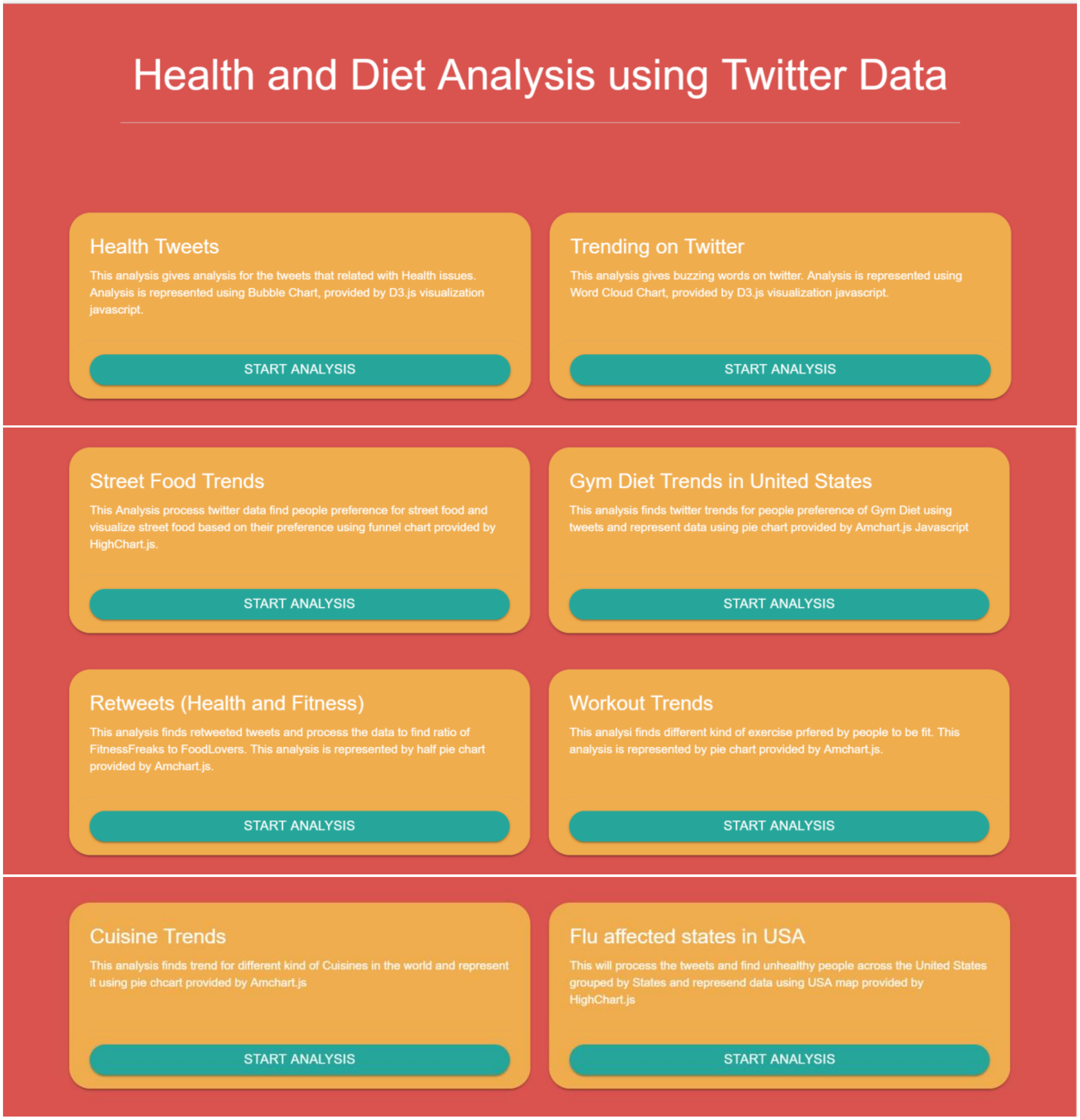
➔D3.js: D3 stands for Data Driven Documents (D3) is a JavaScript library for producing dynamic, interactive data visualization in web browser. It uses SVG, HTML and CSS standards.

➔HighChart.js: HighChart.js is provided by HighCharts ompany. It is a charting library written in pure JavaScript. It also provides data visualization in web browser.

➔Amchart.js: It is provided by AmCharts. It is also a charting library which helps to visualize data in the form of charts and Maps.

4. Implementation

1. Home Page:



Health and Diet Analysis using Twitter Data

Health Tweets

This analysis gives analysis for the tweets that related with Health issues. Analysis is represented using Bubble Chart, provided by D3.js visualization javascript.

Trending on Twitter

This analysis gives buzzing words on twitter. Analysis is represented using Word Cloud Chart, provided by D3.js visualization javascript.

Street Food Trends

This Analysis process twitter data find people preference for street food and visualize street food based on their preference using funnel chart provided by HighChart.js.

Gym Diet Trends in United States

This analysis finds twitter trends for people preference of Gym Diet using tweets and represent data using pie chart provided by Amchart.js Javascript

Retweets (Health and Fitness)

This analysis finds retweeted tweets and process the data to find ratio of FitnessFreaks to FoodLovers. This analysis is represented by half pie chart provided by Amchart.js.

Workout Trends

This analysi finds different kind of exercise prfered by people to be fit. This analysis is represented by pie chart provided by Amchart.js.

Cuisine Trends

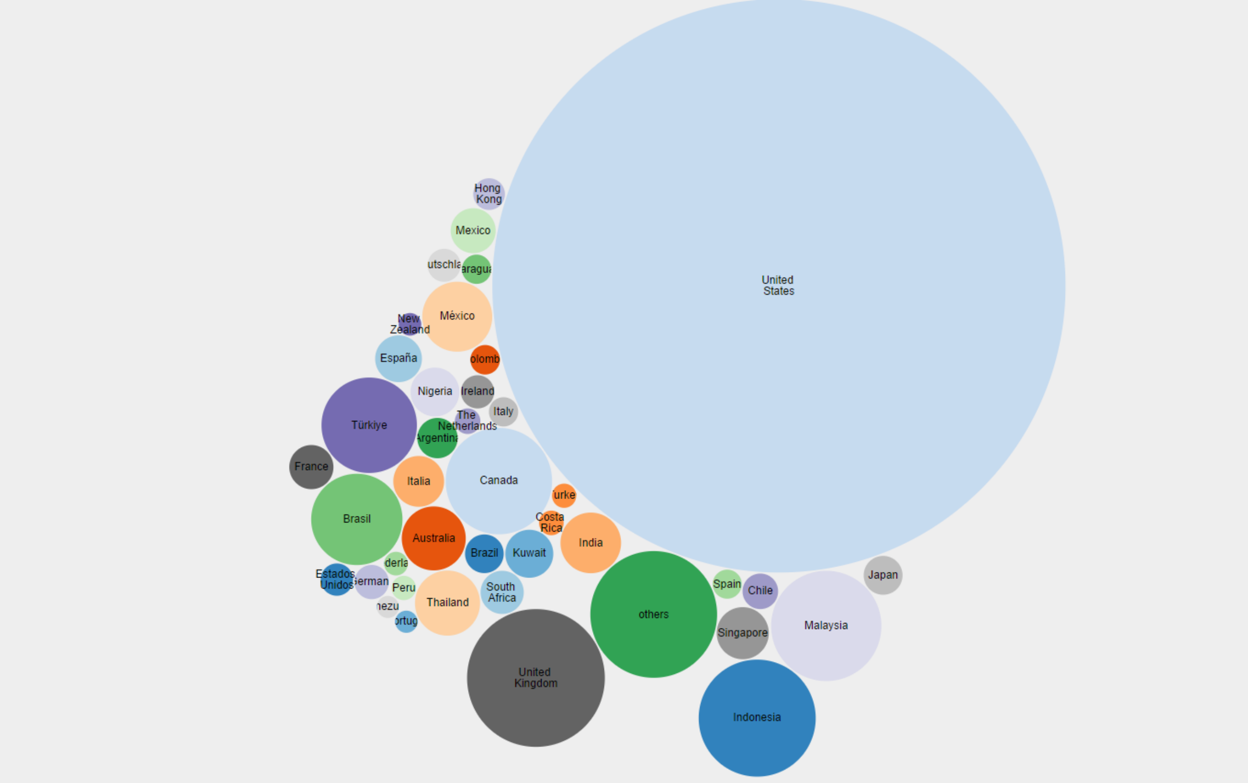
This analysis finds trend for different kind of Cuisines in the world and represent it using pie chcart provided by Amchart.js

Flu affected states in USA

This will process the tweets and find unhealthy people across the United States grouped by States and represend data using USA map provided by HighChart.js

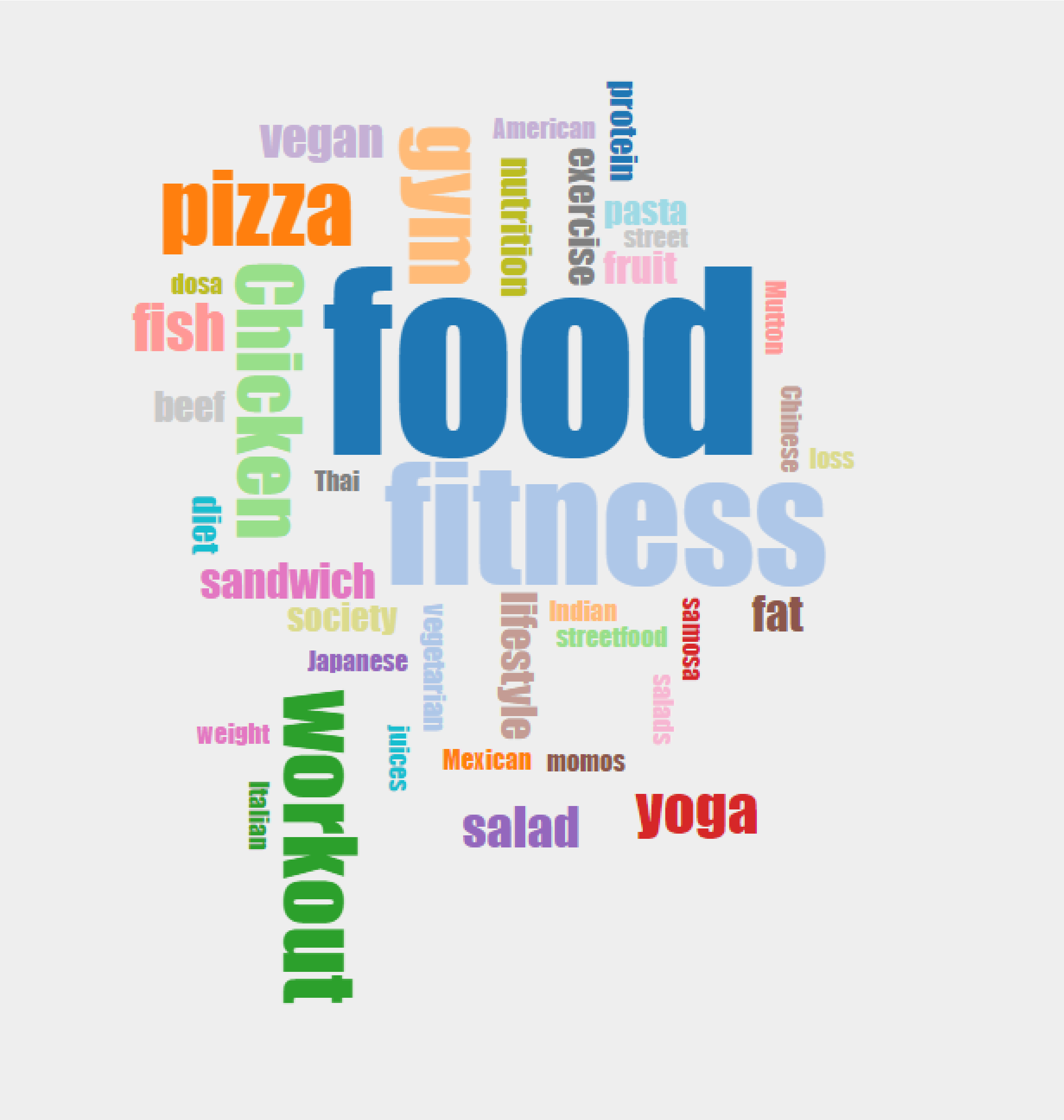
2. Health Tweets.

Goal: This analysis gives analysis for the tweets that related with Health issues. Analysis is represented using Bubble Chart, provided by D3.js visualization JavaScript.



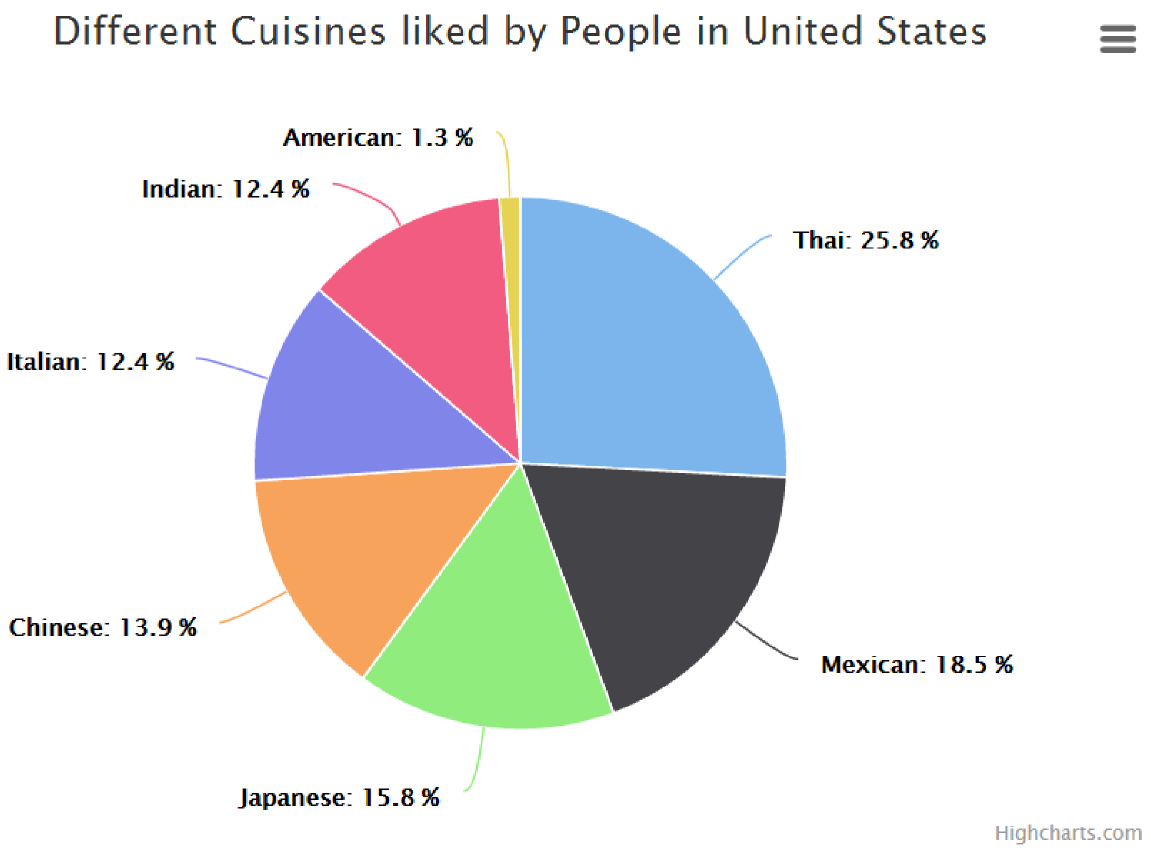
3. Trending on Twitter

Goal: This analysis gives buzzing words on twitter. Analysis is represented using Word Cloud Chart, provided by D3.js visualization JavaScript.



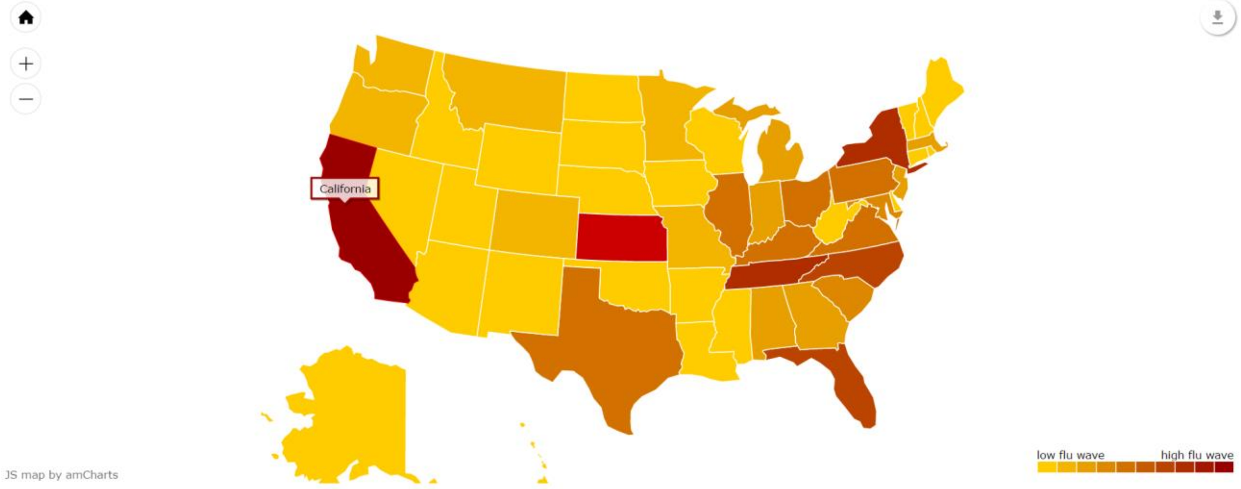
4. Cuisine Trends

Goal: This analysis finds trend for different kind of Cuisines in the world and represent it using pie chcart provided by Amchart.js



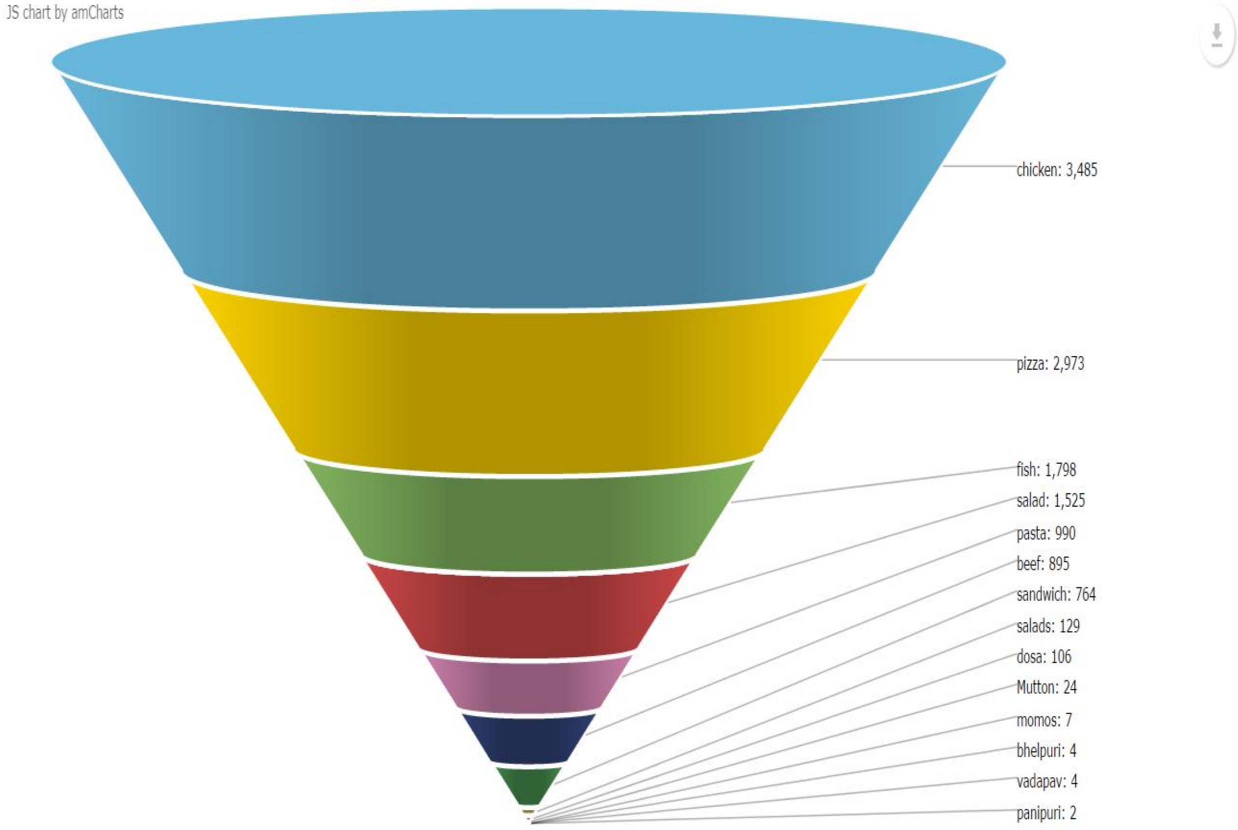
5. Flu affected states in USA

Goal: This will process the tweets and find unhealthy people across the United States grouped by States and represent data using USA map provided by HighChart.js



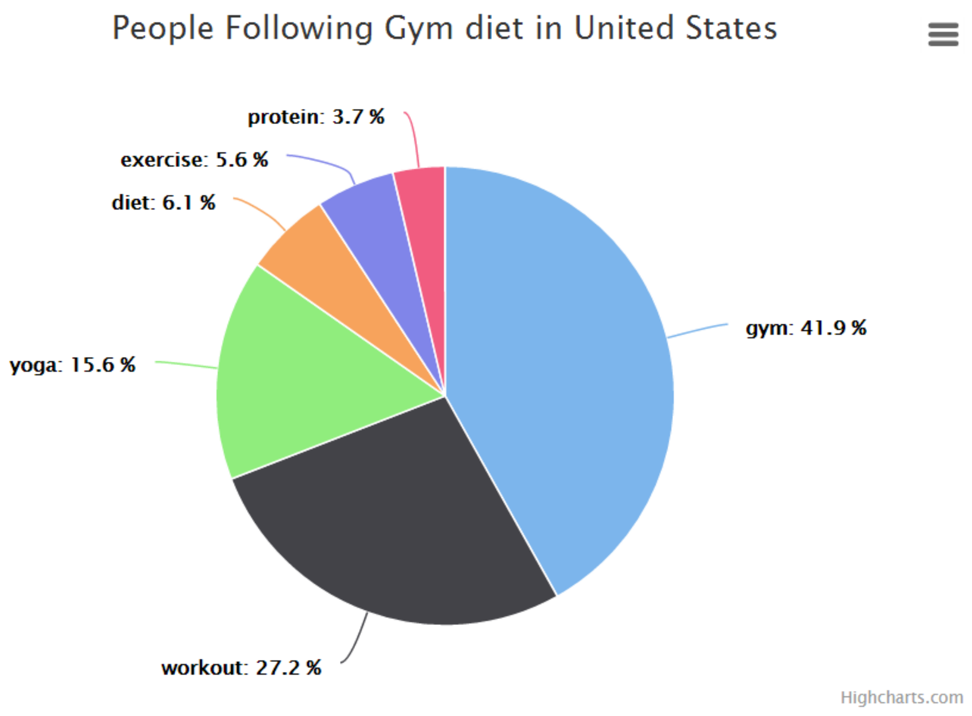
6. Street Food Trends

Goal: This Analysis process twitter data find people preference for street food and visualize street food based on their preference using funnel chart provided by HighChart.js.



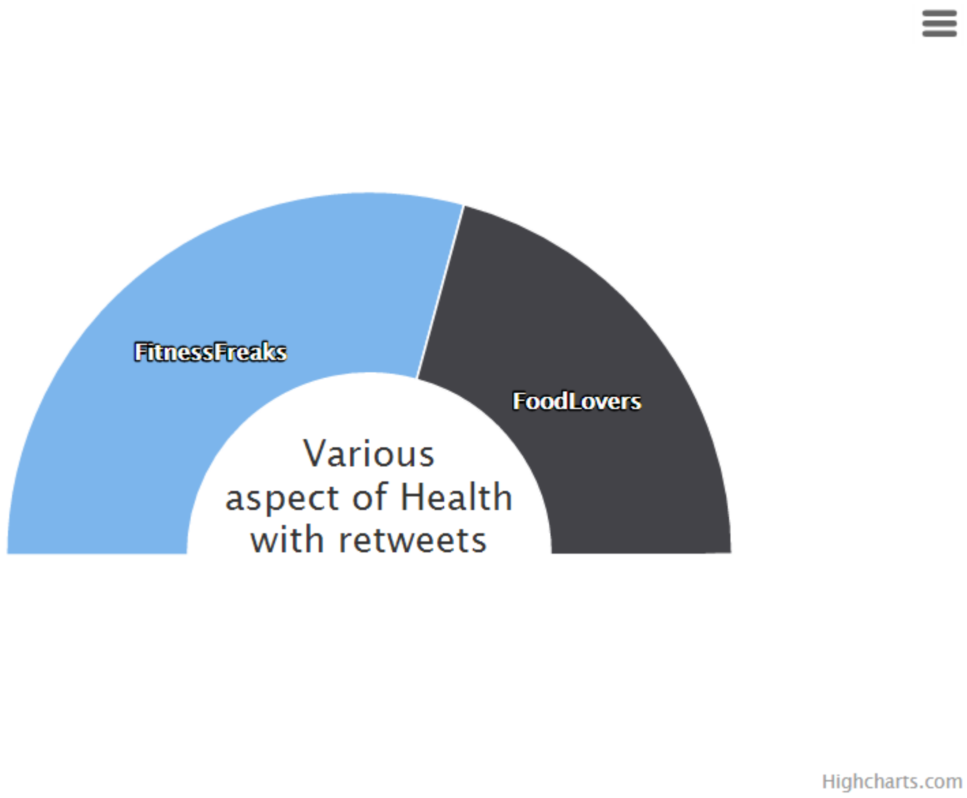
7. Gym Diet Trends in United States

Goal: This analysis finds twitter trends for people preference of Gym Diet using tweets and represent data using pie chart provided by Amchart.js Javascript



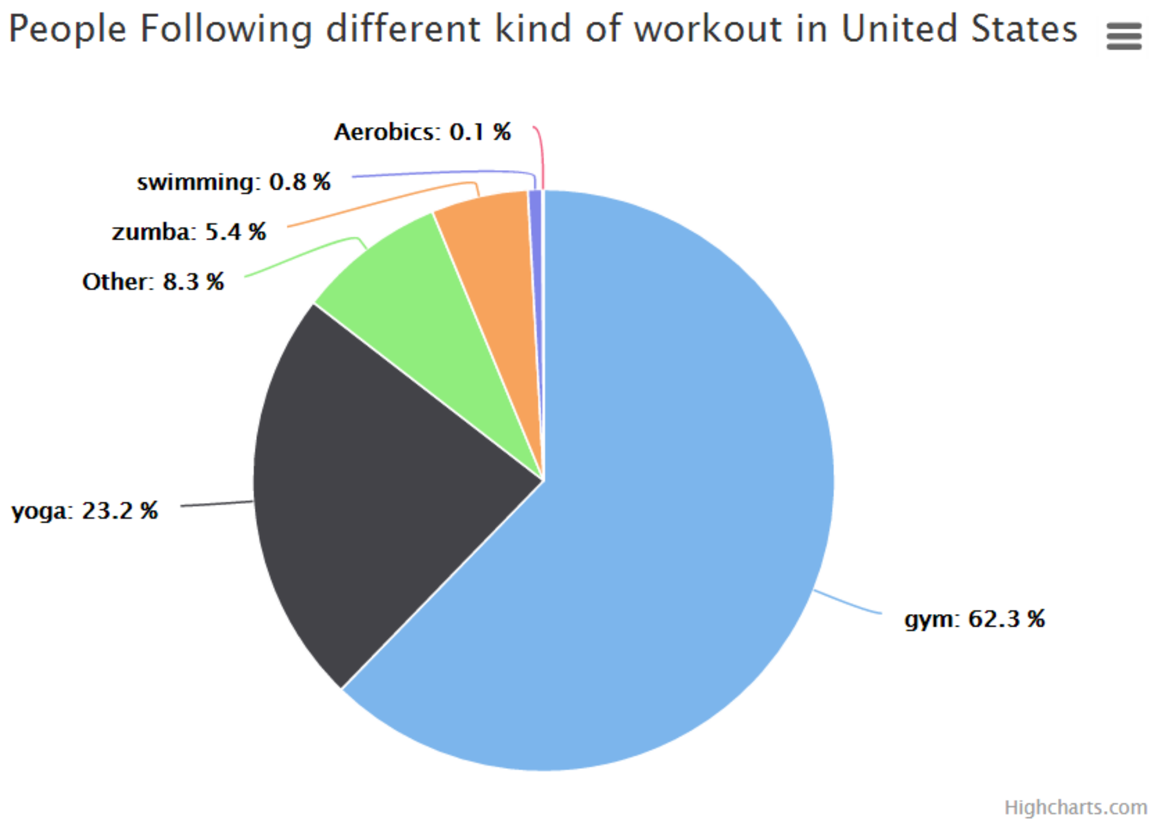
8. Retweets (Health and Fitness)

Goal: This analysis finds retweeted tweets and process the data to find ratio of FitnessFreaks to FoodLovers. This analysis is represented by half pie chart provided by Amchart.js.



9. Workout Trends

Goal: This analysis finds different kind of exercise preferred by people to be fit. This analysis is represented by pie chart provided by Amchart.js.



6. Project Manual

- **Project Structure:**

1) TwitterData:
    a. TwitterDataMining.Java

    A java code to collect twitter tweets using keywords related to the project.

2) TwitterAnalysis:

    Java Dynamic Web Project, runs on Apache Tomcat and developed using Eclipse IDE.

- **How to run Project?**

A. To run TwitterDataMining project on Amazon AWS server as Background service:

```
nohup java -cp twitter4j-stream-4.0.4.jar:twitter4j-media-support-4.0.4.jar:twitter4j-examples-4.0.4.jar:twitter4j-core-4.0.4.jar:twitter4j-async-4.0.4.jar: TwitterDataMining &
```

B. To import zip project (TwitterAnalysis) in Eclipse, (make sure Eclipse has installed Maven and Dynamic Web Project plugins)

Steps:

1. Go to File -> Import. The following dialog will appear.

2. Select **Existing Projects into Workspace**.
3. Click the radio button next to **Select archive file** and click the **Browse**.
4. Find the archive file on your hard disk. Click Open to select it.
5. If you have selected an archive file containing an entire Eclipse project, the project name will appear in the box below, already checked. Click Finish to perform the import.
6. Congratulations, the project should now appear in your Project Explorer view!

Note:

- *Run the project on Apache Tomcat v9.0.*
- *Put the twitter.json file collected by TwitterDataMining.java, in following directory It is a working project directory.*

```
..\workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\TwitterAnalysis
```

References:

[1] https://maven.apache.org/maven-features.html

[2] https://maven.apache.org/what-is-maven.html

[3] https://maven.apache.org/

[4]https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-overview.html

[5] https://www.amcharts.com

[6] https://www.highcharts.com

[7] https://d3js.org

[8] https://spark.apache.org/docs/latest/api/java/index.html

[9] https://spark.apache.org/docs/latest/programming-guide.html