# DS-670 Capstone: BigData & Business Analytics

# Data Load and Data Summary

### Data Summary:

Data file given to me and need to be worked on is Social events data from the link [http://iot.ee.surrey.ac.uk:8080/datasets.html#eventss](http://iot.ee.surrey.ac.uk:8080/datasets.html" \l "eventss). The time frame of the data of 2 years which is from June 2012 to June 2014. This data is survey data of UK, which have RSS feeds from municipality department. Data is downloaded from the citypulse website. This website provides with various semantical datasets which can be used for various purpose. This data is collected from various partners of CityPulse EU and this data is relevant resource for smart city data. There are so many papers which of them cited this website.

This website basically collects the data from the surrey county council webcasts at <http://surreycc.public-i.tv/core/portal/home>, this have the webcast data of previous and up coming events. We can access recent ones from the link [http://surreycc.public-i.tv/core/portal/home#webcast](http://surreycc.public-i.tv/core/portal/home" \l "webcast) and previous ones can be accessed from the link <http://connect.surreycc.public-i.tv/site/webcasts.php>. The timeline from the data which usually provided by the surrey website shows the agenda of the live webcast and recent ones. In the same way it may also refer to index points from the previous ones. There are some additional tabs too which used to access the additional features like slides resources, speaker profile and chats. There is also sharing and embedded buttons which can be used to integrate in the personal or our own websites, blogs or any forms. When we open the link and look for library listings we find all the events going on at present, resent and previous ones. Which also includes with time and date of events. It also includes with small description about this event and most importantly venue of the event where it is going to happen. This helps in lot of people to find their interesting events and follow them.

So the data file which we are using for the project is of 8.8kb. It is in the format of csv. Csv stands for comma separated values. In this, data will be similar to the other data file formats but each column is separated by a comma (“,”). This is a basic simple text format where it does not take more space in hard disc which is more than enough. So this format is hugely used and it easily accessible.

First column of the data file is as shown below:

“Planning and Regulatory Committee, <http://www.surreycc.public-i.tv/core/portal/webcast_interactive/144043>, Planning and Regulatory Committee 03/09/2014 10.30 am Ashcombe Suite County Hall Kingston upon Thames Surrey KT1 2DN, <http://www.surreycc.public-i.tv/core/portal/webcast_interactive/144043>, Wed 03 Sep 2014 10:30:00 +0100”

As we can see there are five columns. This dataset came with no header with it. So I assume these column names for each column so that we can use those column names and refer them with those names when needed. First one is “Event name”, second one is “event link”, third one is “description”, fourth one is “alternative link”, fifth one is “time and date”. As we can see the column names are given relevant to the data, for example first column “Planning and Regulatory Committee” it is name of the event and second one clearly is the link to the event same way for third, forth and fifth columns.

In this 2 years eight committee’s conducted around 30 events on whole. These 8 committees are planning and regulatory committee, cabinet, communities select committee, council, guildford local committee, mole valley local committee, planning and regulatory committee, surrey police and crime panel and woking joint committee. So with these data set is of 30 records and with 5 columns.

|  |  |
| --- | --- |
| Event | Event name or event organizer |
| Link | Link to the events |
| Description | Description of this event |
| AltLink | Alternative link to the event |
| Time | Time and date of the event. |

### Data Load:

We are using Zeppelin to build our project and run the test on the dataset. Zeppelin is notebook which is web based. This is used and will provide interactive data analytic functions which can be used by data analysts. We can use SQL, Scala, R, Python and many number of languages in with this. This runs on top of spark. This zeppelin is provided with data integration, data discovery, data analytics and data visualization. Zeppelin provides in-built spark integration which help is fast access and fast processing of the data and perform analytics without any difficulties and faster than any other. Even data visualization is done with ease basic chart are provided with data retrieval so it is easy to check.Due to spark the performance is very high.

Zeppelin installation consist of steps like download the binary files from the link <https://zeppelin.apache.org/download.html>. The current latest version is 0.6.2. After downloading, extract the zip file and save it in a folder. We can also build this with source. To install zeppelin, need to run this command from terminal /bin/install-interpreter.sh –all, which installs all the interpreters available. After installation, need to goto [http://localhost:8080](http://localhost:8080/)where zeppelin web interface is provide and we can perform the actions. We can learn using the tutorials provided in the web application and seek help for any assistance.

There are many interpreters available to work on like python, spark, scala, R, sql and so on. These list of interpreters are listed in conf/interpreters-list document.

Loading dataset can be done by command **val eventText = sc.textFile("/home/scarface/Desktop/sem-3/Cape/surrey\_events.csv").** Create a class which have all the column names which are used to access and these are basically the variable names to refer in the further use. Now data file which is in the csv format is to be split and assign it to the variables from the class created in the previous step. Each record can be a new object. The data is split and assigned to the variable by using the command **val evt = eventText.map(s=>s.split(",")).map(s=>Event(s(0),s(1),s(2),s(3),s(4)))**. Make this a Dataframe using command **evt.toDF().registerTempTable("event").** Now that data is converted into dataframe, it can be accessed using the commands and even we can use the sql to look into the data. SQL query **select \* from event.**