

CASE STUDY ON LIVE STREAMING SOCIAL MEDIA DATA INTO SAP HANA PERFORM TEXT ANALYSIS AND CLUSTERING

Thakur Navneetha Singh (M.tech Scholar)

Vasavi college of Engineering, t.neetusingh@gmail.com

Abstract— The main objective of this project is to feed the real-time data feeds from social media (twitter) and this data then can be used to perform the text analytics for sentiment analysis and as well as perform Segmentation using SAP HANA clustering. In this case study I am going to focus on how to access real-time social media information using Twitter. Based on our interested trend that we want to analyze , we pull out the specific tweets in real-time using Twitter API, then process and loaded this data into a SAP HANA database. To talk with the Twitter API and load the data into SAP HANA , We can make use node.js which is a free open source Google Chrome run-time. Here In this case study i have used SAP HANA instance hosted in cloud but one can also use the SAP HANA database will be hosted on the developer trail edition of the SAP HANA Cloud Platform. This project is done by using SAP HANA (in memory database platform) and Its Predictive Analysis Library algorithms in a practical approach. Once the data has been loaded we will perform further analysis. Sentiment Analysis is performed by finding whether people are tweeting positive or negative statements. Then different types of tweeters can then be grouped and segmented by estimating the influence of tweeter using the Predictive Analysis Library. For example, we can determine who is influential due to a high number of followers and re-tweets and then group the influential tweeters who are expressing negative thoughts about a product together. We can then target that group with an educational outreach program. But as a case study here I am not going to focus on a particular product because its upto the user requirement who want to focus on which kind of data from social media. So here as I am just focusing on how to feed the real-time data feeds from social media (twitter) and how this data , then can be used to perform the text analytics for sentiment analysis and as well as Segmentation using SAP HANA clustering as an example I focus on tweets related to entrepreneur 2015.

Keywords— Live Streaming , Twitter Streaming, real-time data , SAP HANA, Sentiment Analysis, Clustering, Automated Predictive Library, PAL - Predictive Analysis Library.

INTRODUCTION

I am going to access some real time social media information (example twitter) so we are going to get information on specific tweets that we want to look for and pull those out in real time from twitter API, Process those tweets and load that tweets in real time to HANA database.

HANA database will be hosted on cloud platform in Linux SUSE or trial HANA cloud platform Developer edition can be used by registering for it. Then once we load data in HANA database we can do further analysis on it say for example sentiment analysis to understand what people are saying, liking or not liking whatever product or service tweets that we want to analyze. Then we may want to segment or group different types of persons (twitter users) based on their tweets. There may be people who may be saying many negative things about the products or service and there may be some influential since lot of people may follow and retweets , we then target such people take decision accordingly educate them etc. So we do some predictive analysis in order to segment groups sending similar kinds of tweets together by using SAP Predictive Analytics Library.

We can connect to SAP HANA Database hosted in cloud through eclipse IDE where we install and configure SAP HANA studio. Then using this HANA studio we access HANA database and do operations required such as SQL operation or modeling etc using SQL editor from HANA studio.

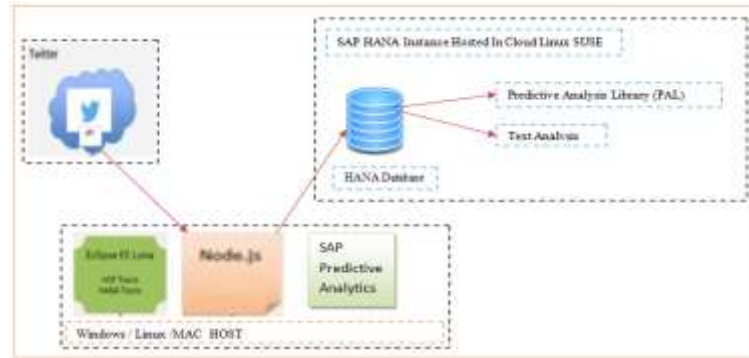


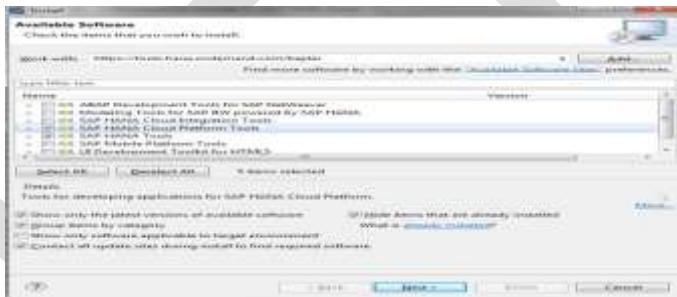
fig: Architecture of Live Streaming into SAP HANA

PROCEDURE TO PERFORM REAL TIME DATA LOAD INTO SAP HANA

Prior to start with the this case study of loading data into SAP HANA one should installed HANA database in LINUX SUSE or Register at hanatrial.ondemand.com for the trial HANA cloud platform for developer and create HANA XS instance. Once on successful hosting of HANA Database follow the below steps to load data in to sap hana in real time from social media:

STEP 1: INSTALL SAP HANA CLOUD PLATFORM TOOLS

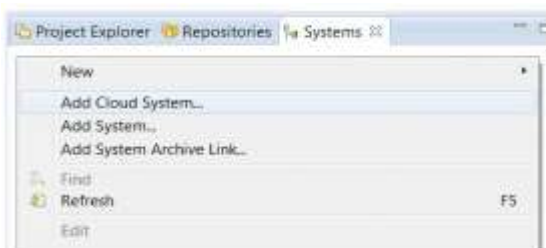
1. Open HANA Studio i.e Eclipse IDE.
2. Go to Help - Install New Software
3. Depending on Eclipse version, enter one of the following URLs:
For Eclipse Kepler (4.3) - <https://tools.hana.ondemand.com/kepler>
For Eclipse Luna (4.4) - <https://tools.hana.ondemand.com/luna>
4. Select "SAP HANA Tools" and "SAP HANA Cloud Platform Tools".



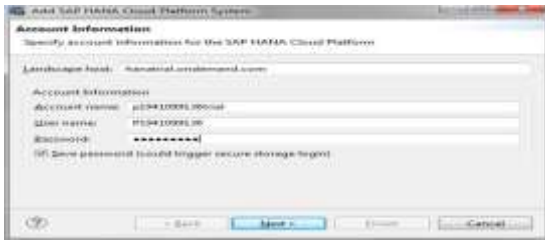
5. Click Next...Next and finally Finish. You will need to restart the HANA Studio.

STEP 2: CONNECT TO HANA INSTANCE FROM HANA STUDIO

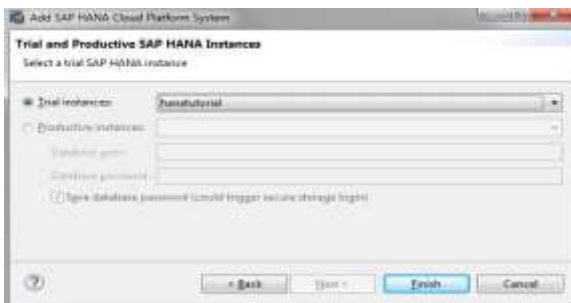
1. Open HANA Studio.
2. In System view, right click and select "Add Cloud System".



3. Enter Account Name, User Name and Password. To know more about Account Name and User Name, refer to the article [Free Access to SAP HANA Cloud](#) a Account Dashboard



4. Select the HANA Instance that you have created in step [Free Access to SAP HANA Cloud](#) à Create a trial SAP HANA Instance.



5. Click on Finish and HANA Cloud system will be added.



STEP 3: REGISTER AT TWITTER

We can find the information about the twitter steaming API's under documentation at <https://dev.twitter.com>.



If you don't already have an account then register at twitter i.e apps.twitter.com where we get the following keys using which we can connect to twitter and fetch data:



Click on Create New App and fill the application details and accept the agreement and hit create application. Once the application is created then you will find the page as below, note down the consumer key you see under application settings.



Click on Keys And Access Tokens tabs , scroll down and click create my access token button.



Now **get Access Token**: This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

- Access Token , Access Token Secret , Access Level Read and write and Owner

Note down all the keys under application settings and access token.

STEP 4: CREATE TABLE DEFINITION IN SAP HANA

Create Table in HANA where you want to store the data that we capture from live streaming using the sql command as shown below:

```
--EntrepreneurTweets--
CREATE COLUMN TABLE "Entrepreneur2015Tweets" (
  "id" VARCHAR(256) NOT NULL,
  "created" TIMESTAMP,
  "text" NVARCHAR(256),
  "lang" VARCHAR(256),
  "user" VARCHAR(256),
  "replyUser" VARCHAR(256),
  "retweetedUser" VARCHAR(256),
  "lat" DOUBLE,
  "lon" DOUBLE,
  PRIMARY KEY ("id")
);
```

STEP 5: DOWNLOAD NODE.JS

Download the node.js from the this location <https://github.com/saphanaacademy/Live3HCP>.

STEP 6: SETUP NODE.JS

1. In browser enter <https://nodejs.org>.
2. Click on install and follow the steps as shown in below screen shots.

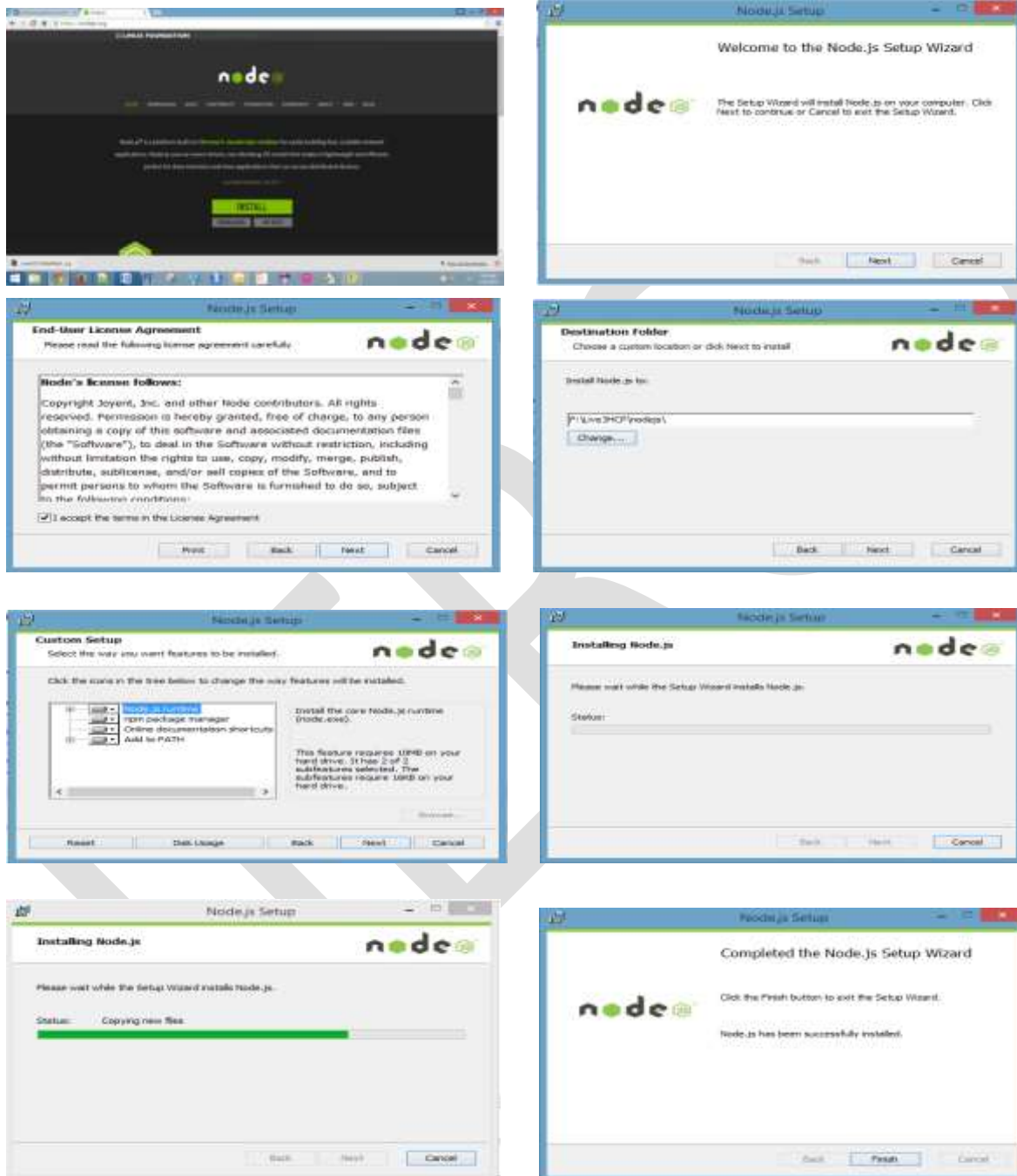


fig: representing different screenshots of node.js installation process

To test that node.js is install properly enter : `node -v` in command prompt

Open a web browser (Chrome in Philip’s demonstration) and navigate to the application by entering localhost:8888 (port number). “Cannot GET /” will be displayed because we didn’t specify a URL. Entering localhost:8888/do/start will display “Nothing to track” as we have yet to specify what we want to track. To track anything you want on Twitter (e.g. company name, product, service) enter **localhost:8888/do/start?track=entrepreneur2015** (here instead of entrepreneur2015 mention the keyword that you wish to track)



fig : Start application

Once app started: we can see trace of records or tweets that got inserted in tweets table in command prompt.

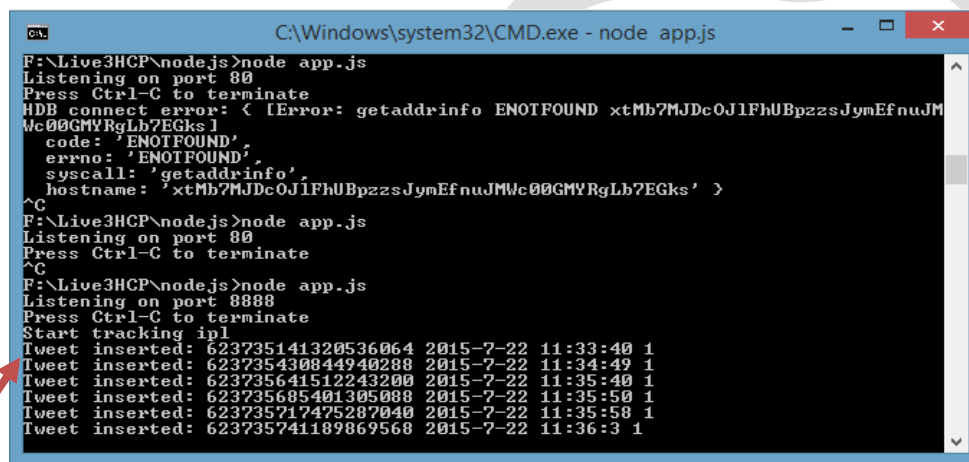


fig: real-time tweets fetching and loading in HANA table

STEP 10: CHECK THE DATA LOAD INTO HANA TABLE CREATED IN STEP 2

Now go to hana studio and check that the table that we create in step 2 i.e tweets get loaded with the data. In HANA Studio Right click on the tweets table and click data preview (or) from sql editor run the select command as : **SELECT * FROM TWEETS;**

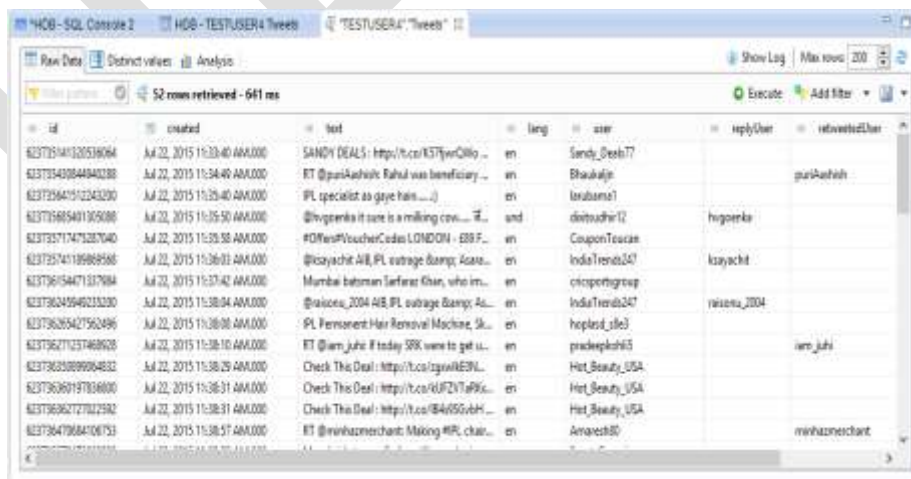


fig: Tweets Data Populated in HANA Table.

STEP 11: STOP APPLICATION

Unless we stop the application tweets will be loaded in real-time from twitter to tweets table in HANA. So in order to stop streaming the data we need to stop the application then only it stops loading data in tweets table in HANA. To stop application in the web browser if you enter **localhost:8888/do/stop** then no more Tweets will be loaded into SAP HANA.



fig : Stop application

STEP 12: PERFORM TEXT ANALYSIS FOR SENTIMENT ANALYSIS

Performing Text analysis SAP HANA allow us to identify the sentiment of persons sending tweets - for example strong positive, weak positive, weak negative, or strong negative sentiment. As we have loaded real-time Twitter data into our SAP HANA table we can view the actual text of each individual tweet. Now from sql editor of hana studio create Text Index using below code and make sure that you set the proper schema in your current session. To set schema set schema name.

```
-- CREATE TEXT ANALYSIS INDEX ON EntrepreneurTweets--  
CREATE FULLTEXT INDEX "Entrepreneur2015tweets" ON "Entrepreneur2015Tweets"("text")  
  CONFIGURATION 'EXTRACTION_CORE_VOICEOFCUSTOMER'  
  LANGUAGE COLUMN "lang"  
  LANGUAGE DETECTION ('EN','FR','DE','ES','ZH')  
  TEXT ANALYSIS ON  
  ;
```

The above code will create a full Text Index on the text column of the Tweets table. The index will be called tweets. A configuration will be set to extract the core voice of customer for the sentiment analysis part of the text analysis. The SQL specifies the language of each Tweet via the Language column of the Tweets table and also determines what languages we will do the sentiment analysis on. Sentiment analysis can be done on Tweets in the following languages English, German, French, Spanish, and Chinese. The final line of code actually turns on the text analysis. Highlight the code and click the run button to execute it. Now for all of the available data as well as for any new rows that are added in HANA table Tweets, a new table will be created automatically that will be logically seen as an index.

Overview of \$TA_Tweets Text Index: Newly created text index called \$TA_Tweets will appear in list of tables. To preview the data right click on the \$TA_Tweets table and select data preview we will see some important columns. These include TA_RULE, a column that confirms that entity extraction is occurring. The TA_TOKEN lists the piece of information that the text analysis is performed on. TA_TYPE shows what type of text analysis is it and what has been extracted. Examples include location, social media and sentiment. The text index determines the five types of sentiment a Tweet can have. For example the word “Grand” is determined to be strong positive sentiment and “Disappointed” is determined to be weak negative sentiment. Some words are labeled ambiguous and could be considered profane such as “loser.”

STEP 13: CALCULATE AND CREATE INFLUENCE AND STANCE VIEW IN HANA STUDIO

This section explains how to create a view that contains scoring for each Twitter user based on their influence and stance (attitude).

Twitter Influence Index :The influence score reflects a Tweeter’s influence based on the number of retweets and replies their Tweets have garnered.Tweets table notes the Twitter handles of users who have retweeted and replied to each individual Tweet. People are considered more influential if their Tweets are replied to and retweeted than if they are the ones who retweet and reply to others’ Tweets. Taking the total number of Tweets that you retweet and reply to and subtracting it from the total amount your personal Tweets have been retweet and replied to gives a numeric index reflecting your individual Twitter influence.

Twitter Influence = (# of times your Tweets that have been retweeted + # of times your Tweets that have been replied to) - (# of Tweets you’ve retweeted + # of Tweet you’ve replied to)

Establish a User’s Stance: Stance is based on sentiment analysis. Open the \$TA_Tweets table created in the prior. Open the Analysis tab and left click on the TA_TYPE in the String folder and choose to add it as a filter. Choose to filter on Strong Positive Sentiment.

Drag TA_TOKEN as the Labels axis and id(Count) as the Values axis. Then choose Tag Cloud as your Analytic and you will get an indication of what words are used to represent Strong Positive sentiment. We can also choose Weak Positive, Weak Negative, and Strong Negative as TA_TYPE to see the words that are classified as those particular sentiments.

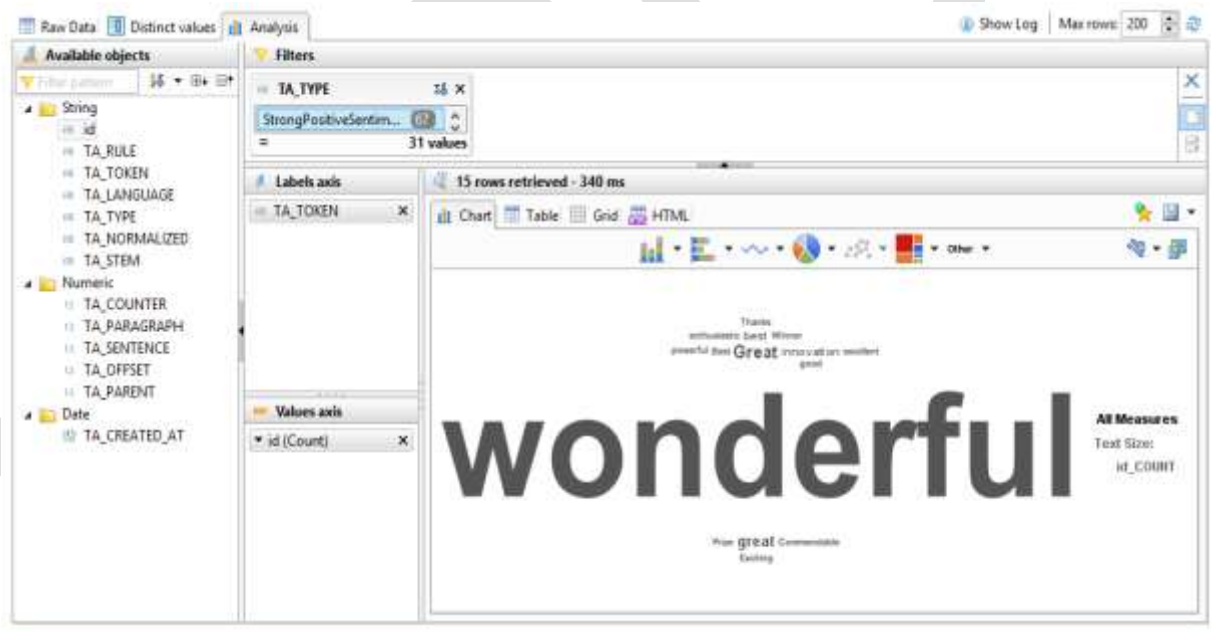


fig :Text Analysis Results

So we can add up the occurrence of the words categorized into the different sentiments for a particular user, assign those sentiment categories particular numeric values and then subtract the weak total from the strong total to get an overall indicator of a user’s stance (attitude).

Overview of SQL View Code – Influence Score

This code will create a view that will process the influence and stance calculations described above. To showcase part of the code Philip runs the four lines before the last of code (see below). Running that returns a count of the number of retweets each Tweeter has.

```
54     LEFT JOIN (  
55         SELECT "user", COUNT(*) AS RSC  
56         FROM "Tweets"  
57         WHERE "replyUser" != ''  
58         GROUP BY "user"  
59     ) rsc ON rsc."user" = t."user"  
60     LEFT JOIN (  
61         SELECT "user", COUNT(*) AS RTSC  
62         FROM "Tweets"  
63         WHERE "retweetedUser" != ''  
64         GROUP BY "user"  
65     ) rtsc ON rtsc."user" = t."user"  
66  
67
```

The section of code of above this highlighted part counts the number of replies in a similar manner. Both parts of this code are straight forward SQL select group bys. This SQL only looks at data within the time that it has been collected. The code also looks at the number times a user has been retweeted and the number of replies. So we must run these four subqueries and join the results together. We need to ensure that we capture all of the user because many user may not have preformed one of those four Twitter actions.

At the top of the code, the case statement will determine the influence score by adding together the number of received retweet (if retweets is missing then it's set to zero so propagation of missing values isn't returned) and replies and then subtracting the number of sent retweets and replies. Using a select from distinct will ensure we capture all of the Tweepers irrespective if they have used retweets or replies.

```
6 -- CREATE TWEETERS VIEW WITH INFLUENCE & STANCE SCORES  
7 CREATE VIEW "Tweepers" AS  
8     SELECT t."user",  
9         CAST(  
10            (CASE WHEN SP IS NULL THEN 0 ELSE SP * 5 END)  
11            + (CASE WHEN WP IS NULL THEN 0 ELSE WP * 2 END)  
12            - (CASE WHEN WN IS NULL THEN 0 ELSE WN * 2 END)  
13            - (CASE WHEN SN IS NULL THEN 0 ELSE SN * 5 END)  
14            AS INT) AS "stance",  
15         CAST(  
16            ((CASE WHEN RRC IS NULL THEN 0 ELSE RRC END) + (CASE WHEN RTRC IS NULL THEN 0 ELSE RTRC END)  
17            - ((CASE WHEN RBC IS NULL THEN 0 ELSE RBC END) + (CASE WHEN RTSC IS NULL THEN 0 ELSE RTSC EN  
18            AS INT) AS "influence"  
19     FROM (SELECT DISTINCT "user" FROM "Tweets") t
```

Overview of SQL View Code – Stance Score

For the stance we will run a query that will pull out the number of StrongPositive, WeakPositive, StrongNegative and WeakNegative sentiments tweeted by an individual user. Instead of having a row for each sentiment per user we transpose that data into four individual columns for each user.

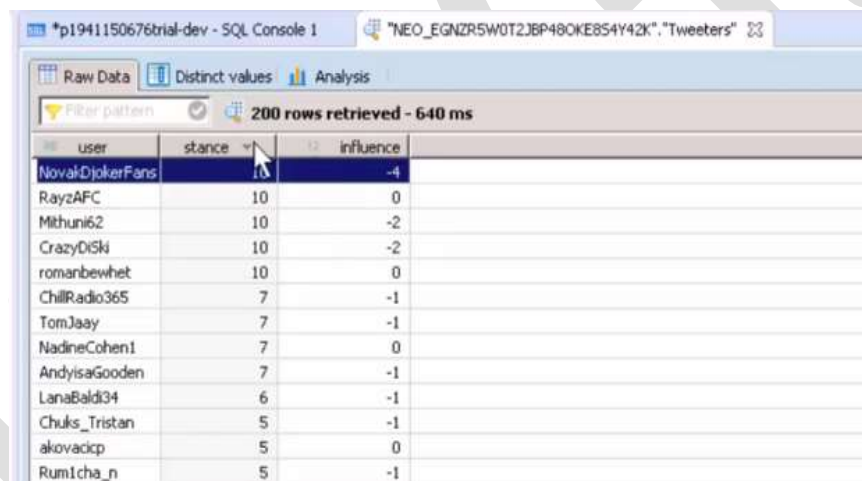
```
23     LEFT JOIN (  
24         SELECT "id",  
25         SUM(CASE TA_TYPE WHEN 'StrongPositiveSentiment' THEN "total" END) AS SP,  
26         SUM(CASE TA_TYPE WHEN 'WeakPositiveSentiment' THEN "total" END) AS WP,  
27         SUM(CASE TA_TYPE WHEN 'WeakNegativeSentiment' THEN "total" END) AS WN,  
28         SUM(CASE TA_TYPE WHEN 'StrongNegativeSentiment' THEN "total" END) AS SN  
29     FROM (  
30         SELECT "id", TA_TYPE, COUNT(*) AS "total"  
31         FROM "TA_tweets"  
32         WHERE TA_TYPE = 'StrongPositiveSentiment' OR  
33             TA_TYPE = 'WeakPositiveSentiment' OR  
34             TA_TYPE = 'WeakNegativeSentiment' OR  
35             TA_TYPE = 'StrongNegativeSentiment'  
36         GROUP BY "id", TA_TYPE  
37     )  
38     GROUP BY "id"  
39     ) i ON t."id" = i."id"  
40     GROUP BY "user"
```

Finally we will join that data together and apply a stance ratio. We have given Strong Positive a score of 5, Weak Positive a score of 2, Weak Negative a score of 2, and Strong Negative a score of 5. We will add up those scores and then subtract the overall negative value from the overall positive value to get a stance score.

```
8 SELECT t."user",
9 CAST (
10 (CASE WHEN SP IS NULL THEN 0 ELSE SP * 5 END)
11 + (CASE WHEN WP IS NULL THEN 0 ELSE WP * 2 END)
12 - (CASE WHEN WN IS NULL THEN 0 ELSE WN * 2 END)
13 - (CASE WHEN SN IS NULL THEN 0 ELSE SN * 5 END)
14 AS INT) AS "stance",
```

Running the Code to Create the Stance and Influence View

Newly created Tweepers view appear in the list of views right click on the Tweepers view and select data preview .Then we can see the data containing three columns user stance and influence score for every Tweeter. For example a user could have a very positive stance but a very low influencer score. This information is incredibly useful as we can now group together users with similar influences and stances.



user	stance	influence
NovakDjokerFans	10	-4
RayzAFC	10	0
Mithun62	10	-2
CrazyDiSkI	10	-2
romanbewhet	10	0
ChillRadio365	7	-1
TomJaay	7	-1
NadineCohen1	7	0
AndyisaGooden	7	-1
LanaBald34	6	-1
Chuks_Tristan	5	-1
akovadicip	5	0
RumIcha_n	5	-1

fig: User influence and stance

STEP 14: PERFORM SEGMENTATION

This section explains how the SAP HANA predictive analysis library (PAL) can be used to cluster similar Tweepers together based on their influence and stance scores.

Reading through the SAP HANA PAL documentation is vital for getting a full understanding of the myriad capabilities PAL offers. PAL is embedded data mining algorithms in the SAP HANA engine (where the data actually resides).Back in Eclipse do a data preview on the Tweepers table we just created. This Tweepers table will be the input table for the predictive analysis. Our id will be the Twitter users' handles and our inputs will be the stance and influence scores. Here i have used k-means clustering algorithm of PAL to perform segmentation.

Running the Code and Examining the Tables : To actually run the clustering is rather simple. First the results tables must be empty and then the stored procedure must be called. The input table/view is the Twitter view, which will send real-time data straight into the

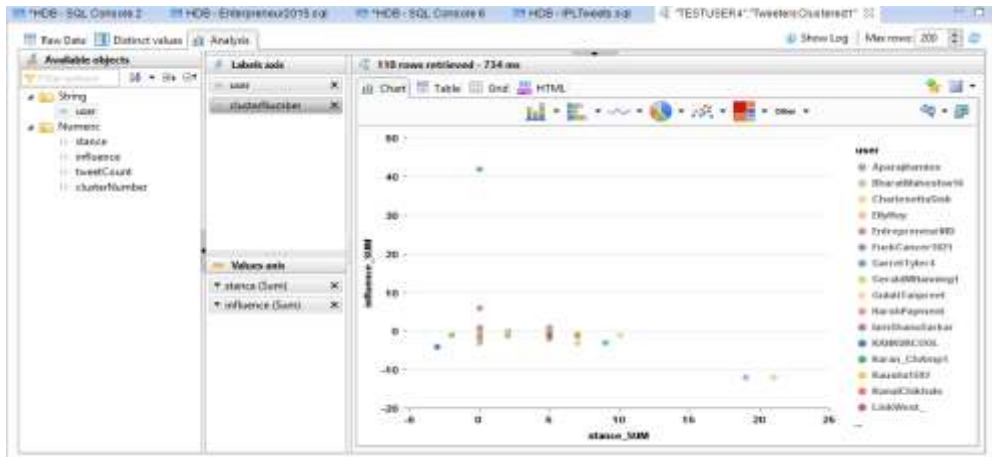


fig: User segmentation results

COMPLETE PAL PROCEDURE CODE TO PERFORM SEGEMENTATION USING K-MEANS

```
CALL"SYS".AFLLANG_WRAPPER_PROCEDURE_DROP('TESTUSER4', 'ENTERPRENEUR_CLUSTER_PROCEDURE');
CALL"SYS".AFLLANG_WRAPPER_PROCEDURE_CREATE('AFLPAL', 'KMEANS', 'TESTUSER4', 'ENTERPRENEUR_CLUSTER_PROCEDURE', PAL_SIGNATURE_TBL);
```

---TESTING---

- **CREATE PARAMETER TABLE**

```
CREATE COLUMN TABLE PAL_PARAMS1 LIKE PAL_T_PARAMS1;
INSERT INTO PAL_PARAMS1 VALUES ('THREAD_NUMBER', 2, null, null);
INSERT INTO PAL_PARAMS1 VALUES ('GROUP_NUMBER_MIN', 3, null, null);
INSERT INTO PAL_PARAMS1 VALUES ('GROUP_NUMBER_MAX', 6, null, null);
INSERT INTO PAL_PARAMS1 VALUES ('INIT_TYPE', 4, null, null);
INSERT INTO PAL_PARAMS1 VALUES ('DISTANCE_LEVEL', 2, null, null);
INSERT INTO PAL_PARAMS1 VALUES ('MAX_ITERATION', 100, null, null);
INSERT INTO PAL_PARAMS1 VALUES ('EXIT_THRESHOLD', null, 1.0E-6, null);
INSERT INTO PAL_PARAMS1 VALUES ('NORMALIZATION', 0, null, null);
```

- **CREATE OUTPUT TABLES**

```
CREATE COLUMN TABLE PAL_RESULTS1 LIKE PAL_T_RESULTS1;
CREATE COLUMN TABLE PAL_CENTERS1 LIKE PAL_T_CENTERS1;
```

- **CREATE VIEWS FOR ODATA**

```
CREATE VIEW "TweetersClustered" AS
SELECT s.*, t."tweetCount", c."clusterNumber" + 1 AS "clusterNumber" FROM "EntrepreneurTweeters" s INNER JOIN (
```



```
SELECT "user", COUNT(*) AS "tweetCount"  
FROM "Entrepreneur2015Tweets" GROUP BY "user") t ON t."user" = s."user" INNER JOIN "PAL_RESULTS1" c ON  
c."user" = s."user" ;
```

```
CREATE VIEW "CLUSTERS1" AS SELECT c."CLUSTERNUMBER" + 1 AS "CLUSTERNUMBER", c."STANCE", c."INFLUENCE",  
T."USERS"  
FROM "PAL_CENTERS1" c INNER JOIN ( SELECT "clusterNumber", COUNT(*) as "users" FROM "PAL_RESULTS1"  
GROUP BY "clusterNumber" ) t ON t."clusterNumber" = c."clusterNumber";
```

- **RUNTIME CALL THE PAL PROCEDURE CREATED**

```
TRUNCATE TABLE PAL_RESULTS1;  
TRUNCATE TABLE PAL_CENTERS1;
```

```
CALLTESTUSER4."ENTERPRENEUR_CLUSTER_PROCEDURE"("EntrepreneurTweeters",PAL_PARAMS1,  
PAL_RESULTS1, PAL_CENTERS1) WITH OVERVIEW;
```

CONCLUSION

We can make use of the power of in memory database HANA and can perform predictive analysis to take business decisions using the APL library of HANA or we can also use SAP Predictive Analytics software. This was just a sample case study explaining how live Streaming data from social sites to HANA can be performed and depending on our business requirements we can perform the analysis accordingly.

REFERENCES:

- [1] <http://saphanatutorial.com/>
- [2] http://help.sap.com/hana/sap_hana_studio_installation_update_guide_en.pdf
- [2] <http://scn.sap.com/community/predictive-analytics/blog/2015/03/02/what-is-the-sap-automated-predictive-library-apl-for-sap-hana>
- [3] <http://scn.sap.com/community/hana-in-memory/blog/2015/02/19/how-to-install-the-automated-predictive-library-in-sap-hana>
- [4] <http://scn.sap.com/community/hana-in-memory/blog/2015/03/17/introducing-the-sap-automated-predictive-library>
- [5] https://help.sap.com/hana/SAP_HANA_Predictive_Analysis_Library_PAL_en.pdf
- [6] <http://scn.sap.com/community/hana-in-memory/blog/2015/03/17/introducing-the-sap-automated-predictive-library>
- [7] <http://scn.sap.com/docs/DOC-59928>
- [8] <http://scn.sap.com/community/predictive-analytics/blog/2015/03/02/what-is-the-sap-automated-predictive-library-apl-for-sap-hana>
- [9] https://hcp.sap.com/content/dam/website/saphana/en_us/Technology%20Documents/SPS09/SAP%20HANA%20SPS%2009%20-%20Smart%20Data%20Streaming.pdf

[10] <http://saphanatutorial.com/sap-hana-text-analysis-using-twitter-data/>

[11] <http://scn.sap.com/community/developer-center/hana/blog/2015/04/09/text-analysis-in-sap-hana-integrate-with-twitter-api>

[12] <https://scn.sap.com/thread/3495915>

[13] <http://scn.sap.com/community/developer-center/hana/blog/2014/09/10/sap-hana-twitter--find-your-leads>

[14] https://hcp.sap.com/content/dam/website/saphana/en_us/Technology%20Documents/SPS09/SAP%20HANA%20SPS%2009%20-%20Smart%20Data%20Streaming.pdf

[15] <http://scn.sap.com/community/developer-center/cloud-platform/blog/2015/02/06/live3--sap-hana-cloud-platform-tutorials-for-advanced-real-time-social-media-analytics>