**Problem Set**

**Kafka Team:**

1. **Real time twitter sentiment analysis using kafka**

**Description:**  Twitter provides a very convenient API to fetch tweets in a streaming manner. In this task, you will access data(twit) from the twitter (using API) and find the sentiment(positive vs negative) of the twit based on a predefined data set. Details of the project can be found here.

<https://github.com/sridharswamy/Twitter-Sentiment-Analysis-Using-Spark-Streaming-And-Kafka>

**Your Task:**

1. Download or rewrite the from the github.

2. Understand the code/functionalities (ask me if you do not understand)

3. Connect to the twitter (instructions are given how to do that)

4. Run the program with live twitter data.

Learning Challenge: how to connect to the Twitter API for analysis.

1. **Real time twitter word analysis (plot)**

**Description:** We extend the previous problem and plot the word like WordCloud. Figure1 shows a WordCloud example.



Figure 1WordCloud

You will get the required learning materials about WordCloud from the following two projects.

1. <https://www.datacamp.com/community/tutorials/wordcloud-python>
2. <https://github.com/amueller/word_cloud>

**YourTask:**

1. Access the twitter words using API in the Kafka Producer. (every 60 seconds)
2. Send the data to the Apache Streaming.
3. Store the words in the mongoDb.
4. Visualize the words like WordCloud in every 5 minutes. Plot only 100 most frequent words.
5. **Finding nearest parking spot**

In this task, we will learn and practice accessing real sensors data and some simple data preparation and visualisation. We will **use open sensor data** from Melbourne's open data platform in particular on-street parking bay sensors data. This data contains information from in-ground car parking bay sensors across the city with status indicating if a car is present or not present, the spatial coordinate of the sensor, and the street marker id. We will use this data and **present the available parking spaces in the google maps** as well as **display the closest parking spot from your location**.

References you can use:

* Accessing the On-street parking data:
  + <https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Parking-Bay-Sensors/vh2v-4nfs>
  + <https://dev.socrata.com/foundry/data.melbourne.vic.gov.au/dtpv-d4pf>
* Plotting sensor data on Google Maps
  + <https://pypi.org/project/gmplot/>
* Finding the closest parking spot. This might not be 100% accurate but will work for the purpose of the exercise.
  + <https://en.wikipedia.org/wiki/Euclidean_distance>

1. **Find the coolest and hottest place in Australia in real time.**

In the task you will extend the ideas from the home work last week. In your homework, you have plotted the temperature of the three campuses. **In this task, you will show the coolest and hottest place in real time.** You can query the open weather in real time using the API, I have given you last week. You can find the list of all the postcode of Australia in this link :<http://www.corra.com.au/australian-postcode-location-data/>. If for some reasons (e.g., your computer memory limitation or server refusal to response to large number of request) you cannot access the API, try only the postcodes of Victoria (VIC).

**Your Task:**

1. Create a list of postcodes
2. Query the API for each postcodes
3. Find Min() and Max()
4. Plot the result in every 10 minutes interval
5. **Real time Plotting of Fitbit (www.fitbit.com) data**

You will use heart rate API (<https://dev.fitbit.com/build/reference/web-api/heart-rate/>) to get heart rate data. You have to use this API in the following way to get data in data in 1 second interval.

**https://api.fitbit.com/1/user/-/activities/heart/date/today/1d/1sec/time/00:00/00:01.json**

Now transform the data in 5 minutes interval and plot. Secondly, if value is less than 90 or greater then 140 then store it in the database.

**Your Task:**

1. Write a program to access fitbit API
2. Aggregate the values and plot
3. Check for value against the range (90<value<140). If out of the range, then store the value in the database.

**Spark Team:**

1. **ML Pipelines:**This task is to introduce the idea of Estimator, Transformer and Param. Here is the details,

[**https://spark.apache.org/docs/latest/ml-pipeline.html**](https://spark.apache.org/docs/latest/ml-pipeline.html)

You do not have to submit anything for this task. This task is only for your learning. Try the examples in the link.

1. **Naïve Byes Problem:**  task is to classify a new crime description to one of 33 categories.Details of the project is here.

https://towardsdatascience.com/multi-class-text-classification-with-pyspark-7d78d022ed35

1. **Logistic Regression:** The classification goal is to predict whether the client will subscribe (Yes/No) to a term deposit. Details of the project is here.

https://towardsdatascience.com/machine-learning-with-pyspark-and-mllib-solving-a-binary-classification-problem-96396065d2aa

1. **K-Means:** You will use K-Means to analyze hacking attacks. Details of the project is here

https://medium.com/tensorist/using-k-means-to-analyse-hacking-attacks-81957c492c93

1. **Collaborative filtering:** Building a song recommendation system. Details of the project is here

https://towardsdatascience.com/building-spotifys-discover-weekly-with-spark-4370d5d0df2f