**BD PROJECT**

**MACHINE LEARNING WITH SPARK STREAMING**

* datasets - [Project - Machine Learning with Spark Streaming](https://drive.google.com/drive/folders/1hKe06r4TYxqQOwEOUrk6i9e15Vt2EZGC)
* guidelines - <https://cloud-computing-big-data.github.io/mlss.html>
* deadline - 3/12/2021
* libraries -
  1. numpy
  2. sparkdl
  3. pyspark
  4. sparkmllib

* do a dataset w text fields only?

1. Enron Email Spam Detection
2. San Francisco Crime Classification
3. Sentiment140

* do a dataset w only two/three classification labels?

1. Enron Email Spam Filter Detection
2. Sentiment140

* **MAKE REGULAR COMMITS TO REPO**
* to structure repo - [kriasoft/Folder-Structure-Conventions: Folder / directory structure options and naming conventions for software projects](https://github.com/kriasoft/Folder-Structure-Conventions)
* w every batch of data - preprocess **(can use built-in functions from SparkMLlib)**

**\****for each step, explain why + how it improves model performance later*

1. statistical techniques
2. scaling (MinMax, Robust, etc.)
3. normalising
4. feature selection, dimension reduction - extract features w highest importance
   1. find variance of each feature
   2. retain features w highest variance
5. convert all non-numerical data to numerical

* data modelling -
  + incremental due to batch size
  + each batch comes in from preprocessing and then goes to modelling **(everything in single loop?)**
  + incremental learning, re-fit on each new batch that comes in as input
  + **cannot store all batches in memory**
  + implement ~~at least~~ 3 classifiers
    - analyse performance of each
    - use *different* classifiers to check which classifier is best fit
  + hyperparameter tuning
  + modify batch size and see how performance changes
    - try on each classifier
* data testing -
  + stream in batches **(batch size might be different from modelling batch size)**
  + do prediction on all batches using all models developed
  + performance metrics after each classifier -
    - confusion matrix
    - F-1
    - accuracy
    - prediction
    - recall
  + analyse difference in performance bw each classifier
  + plot predictions -
    - across -
      * classifiers
      * hyperparameters
      * batch sizes
    - 3 plots per classifier
    - 3 plots w 3 experiments w training batch size
* clustering -
  + batches streamed in endless loop
  + fit clustering algo on each input batch - incremental learning
  + after each round of streaming all batches - compare centroids
    - if diff bw current and prev v less - stop input stream
  + **no. of clusters = no. of classes**
  + to analyse -
    - do all examples in same test class fall in same cluster?
    - is clustering a form of unsupervised classification

*References -*

1. [A deep dive into Machine learning with Apache Spark](https://blog.clairvoyantsoft.com/machine-learning-with-spark-streaming-281b2d1e4fd5)