Spark Streaming for Machine Learning

Apache Spark is an open-source unified analytics engine for large-scale data processing. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Apache Spark has its architectural foundation in the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines. It facilitates the implementation of both iterative algorithms, which visit their data set multiple times in a loop, and interactive/exploratory data analysis, i.e., the repeated database-style querying of data. Apache Spark requires a cluster manager and a distributed storage system. For cluster management, Spark supports standalone (native Spark cluster, where you can launch a cluster either manually or use the launch scripts provided by the install package.
Spark Streaming uses Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and performs RDD transformations on those mini-batches of data. This design enables the same set of application code written for batch analytics to be used in streaming analytics. Spark Streaming has support built-in to consume from Kafka, Flume, Twitter, ZeroMQ, Kinesis, and TCP/IP sockets.

Design Details

Our chosen dataset is the spam detection dataset. We have streamed the dataset, followed by pre-processing and understanding it, after which it has been converted into a schema, followed by pipelining. The main libraries used in our project are pickle, sklearn and pyspark and the models we have implemented are the Logistic regression, SGD Classifier and MLP Classiifier.

Surface Level Implementation Details

First we have created a sparkcontext, streaming context and sqlcontext instance. Then we have converted it into json readable file. After that, we reated a datarame of the same, post which we used RegexTokenizer, StopWordsRemover and Word2Vec to process the data before using this for pipelining. Finally, we have trained the models - Logistic regression, SGD Classifier and MLP Classiifier. At the end, we have used pickle to dump and save the models.

Reason Behind Design Decisions

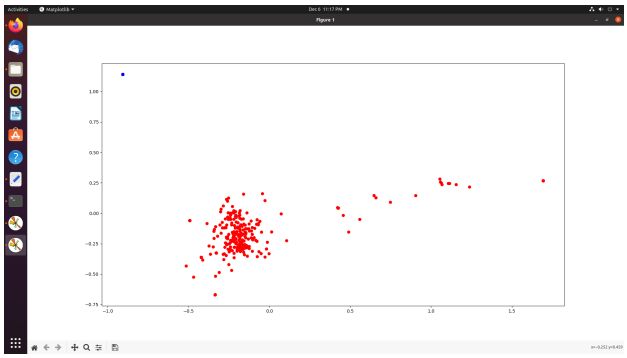
We have used pyspark and sklearn for the following reasons:

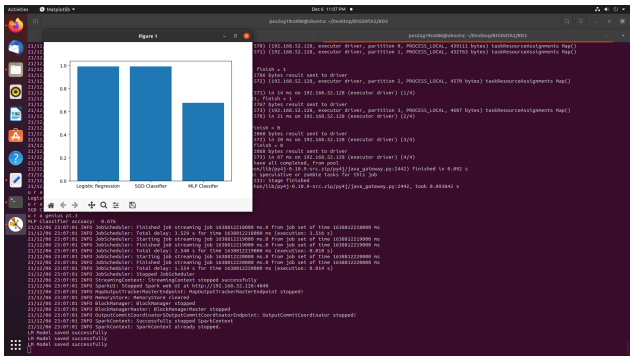
- Sklearn has extensive and easily comprehensible documentation
- Pyspark has been discussed in class which gives us the support of live demo
- The inbuilt functions in these libraries for machine learning models are easy to use and understand
- They are publicly available and are not subject to legal bindings or copyrights

Takeaways From The Project

The project enables us to stream data not just from a dataset in batches, it also helps gain information on how to use an API to stream data in real-time and analyse the legitimacy of the mail. It also helped us gain knowledge on how to run python modules for machine learning, spark streaming and Hadoop, while integrating it with the pre-requisite Java jdk. The hands-on experience gives us an edge as beginners in the field of big data handling.

Outputs





Team Members

1. Name: Toshani Rungta
   SRN: PES2UG19CS433
   Section: G

2. Samriddhi Vishwakarma
   SRN: PES2UG19CS359
   Section: F

3. Sohan Beela
   SRN: PES2UG19CS397
   Section: F