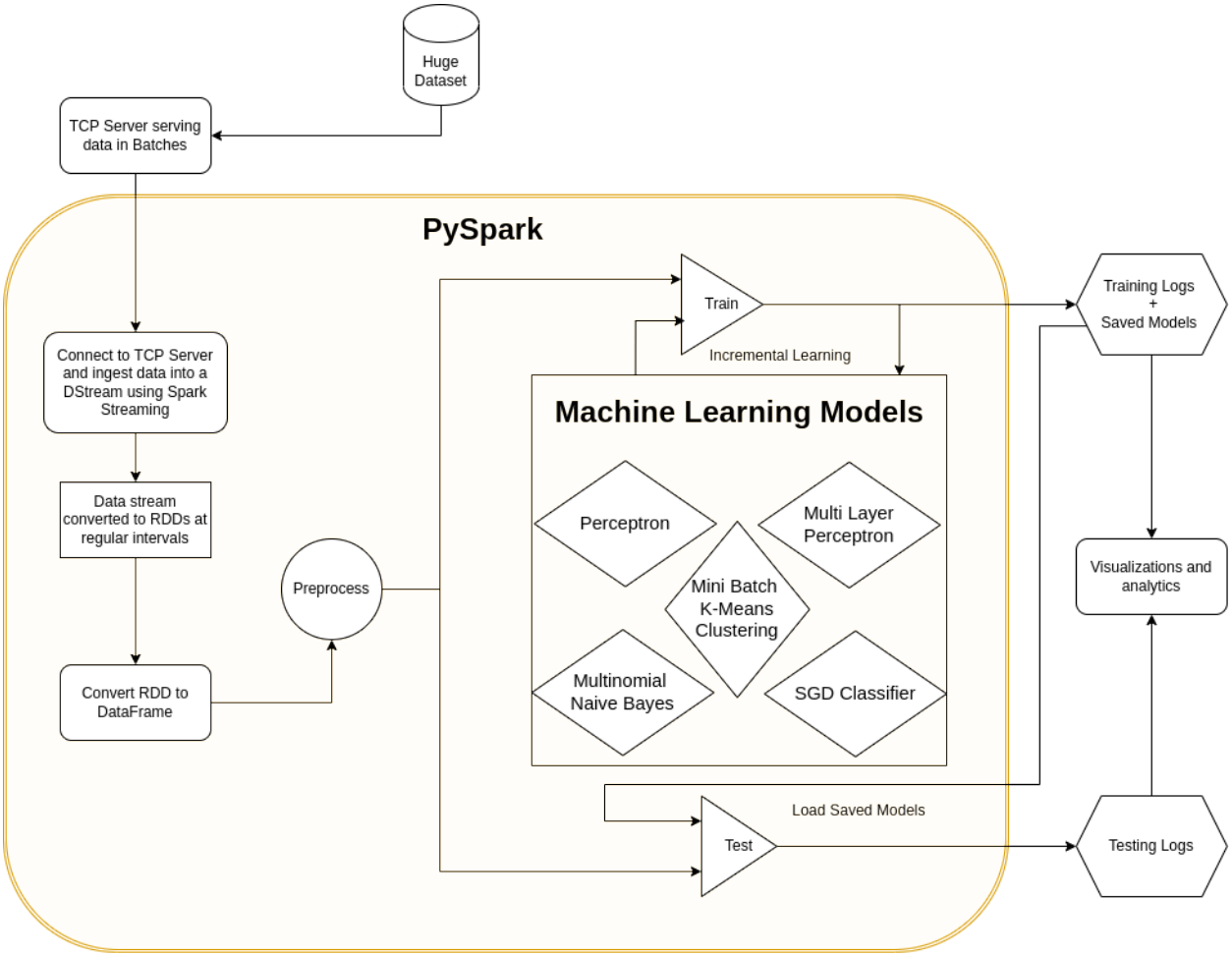
Machine Learning with Spark Streaming - BD_123_272_313_393



Spam Classification workflow

Design Details and Justifications

- Our program takes in the following command line arguments:
  - **Delay**
    - Delay in seconds before processing the next batch
  - **Mode**
    - Training mode: Used to train the models incrementally on the incoming batches of data.
    - Testing mode: Used to test the models on the incoming batches of data
  - **Clean**
    - This is used to clean the log files for a fresh start.
  - **Clustering**
    - This option is used to run only the clustering model, i.e. Training and Testing of Mini Batch K Means Clustering.
  - **Model Number**
    - Choose models (obtained at the k^{th} training batch) for testing over the test dataset.

- Stream Batches of JSONs data from a TCP server using Streaming in PySpark.
- Convert the incoming JSON data into DataFrames.
  - The pyspark framework requires the data to be an RDD object or a DataFrame object. Hence, the JSON data read is converted to a DataFrame.
- Preprocess the data
  - The data that is coming in has a lot of unfavourable characters in it. Because the models require input data in numeric format, the incoming data must be pre-processed for effective learning and prediction.
- Utilized 4 supervised learning models and 1 unsupervised clustering model

    - Multinomial Naive Bayes
    - SGD Classifier
    - Perceptron
    - Multi Layer Perceptron
    - K-means clustering
- Logged training and testing Metrics for each model
    - Logging the metrics like accuracy, precision, recall, prediction, ground truth and the batch numbers into files allows us to perform analytics and visualizations later.
- Automated the process of plotting graphs for these metrics over batches.

Surface level implementation details about each unit

1. **Streaming the data**
   The dataset has **30344** records and is small enough to fit into the memory of most modern home computers. However, we wouldn't be able to fit the entire dataset into memory in real world scenarios. Hence, we have to train models in batches. We must also take into account the constant generation of data. As a result, we conduct our analysis using Streaming Spark. Using a Python-based TCP server, we stream the **email spam classification dataset** in batches of various sizes. The Streaming Pyspark program connects to this server and uses this data for training/testing multiple machine learning models. Each Batch is a JSON of the format:

   {
   'feature0': [list of email subjects],
   'feature1': [list of email messages],
   'feature2': [list of classifications, Ham/Spam]
   }

2. **Data Ingestions using PySpark**
   We use spark streaming to read the incoming batches of data and read it into a dataframe at fixed intervals. We combine the email subject with the message as we believe both play a role in the final classification. We then pass this dataframe for text preprocessing.
3. **Text Preprocessing**
    a. New lines, multiple spaces and non alphabetical characters are replaced with a single space each.
    b. Stop Words (most frequently used words) are removed.
    c. Word stemming is performed.
    d. Hash Vectorization is used for…
4. **Model fitting/testing**
    a. For each batch of incoming data, all the models are incrementally trained.
    b. The accuracy metrics for each batch are logged and the models saved to disk.
    c. In case of testing, the specified model is loaded from disk and used for generating the predictions.
5. **Logs**
    a. This unit runs through all the Log Folders of each Model to get the recall, accuracy and precision score for both test and training and plots it against the number of batches.

Takeaways from the project

- Integration of spark streaming with advanced processing libraries.
- Learned about incremental model training for classification problems.
- Hands-on experience with pyspark.
- Exposed to distributed computing and processing of data.
- Collaborating on git.
- Debugging big data applications.