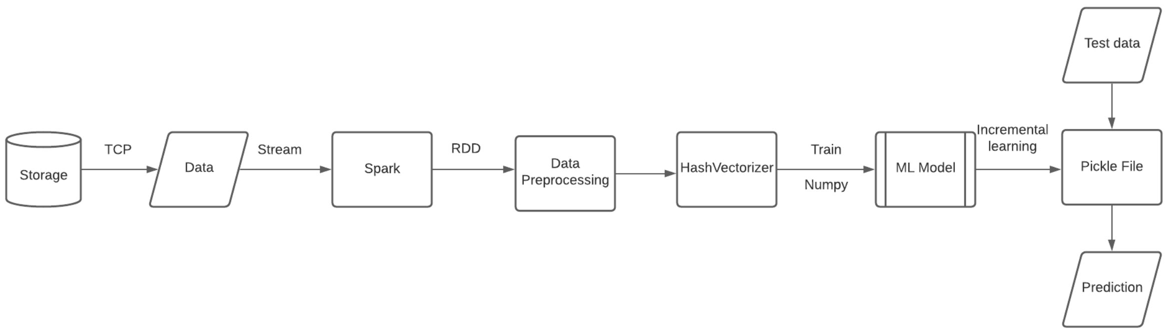
BIG DATA PROJECT REPORT
UE19CS322

Team details:
Abhiram Puranik : PES1UG19CS018
Smruthi BT : PES1UG19CS487
Vishisht Rao : PES1UG19CS572
Shashank Navad : PES1UG19CS601

PROJECT TITLE CHOSEN : **Machine Learning with Spark MLlib**

DESIGN DETAILS :

1. 4 client files for training each model, namely MNBClassifier, SGBClassifier, MLPClassifier, Kmeans
2. 4 pickle files containing the trained final model weights
3. 4 test files for testing the model, with the final weights obtained from the pickle files, to get the prediction of the binary classification.
4. Plots and visualisation



IMPLEMENTATION DETAILS :

Step 1:
Streaming the dataset:
We were given a stream.py which streamed the spam dataset with the specified batch size.We wrote a client.py which makes a TCP connection with localhost to fetch data and put it into a pyspark sql dataframe.

Step 2:
Preprocessing:
We used Hashing Vectorizer, which converts the collection of documents into a sparse matrix, which holds the token occurrence counts.

Step 3:
Training:
We trained the models MLPClassifier, SGDClassifier and MNBClassifier on the training data and saved the incrementally trained model in a pickle file, which can be later loaded for prediction in the testing phase.

Step 4:
Testing:
We trained the models MLPClassifier, SGDClassifier and MNBClassifier on the training data and saved the incrementally trained model in a pickle file, which can be later loaded for prediction in the testing phase.

Step 5:
Testing:
We tested the model on the testing data and obtained the final accuracies for the classification.

DESIGN DECISION :

Choice of preprocessing using HashingVectorizer:
HashingVectorizer has several advantages over other preprocessing techniques like CountVectorizer and tf-idf

1. It is very low in terms of memory usage, scalable to large datasets as there is no need to store a vocabulary dictionary in memory.
2. Faster to pickle and unpickle as it holds no state besides the constructor parameters.
3. It can be used in streaming, with partial fit as there is no state computed during fit, unlike tf-idf where the number of columns in matrix corresponds to the number of unique words, which changes with every batch and hence cannot be incremented.

Choice of models:

1. MNBClassifier - The multinomial Naive Bayes classifier is suitable for classification with discrete features.The multinomial distribution normally require integer feature counts
2. MLPClassifier - MLPs are suitable for classification prediction problems where inputs are assigned a class or label.They are suitable for regression prediction problems where a real-valued quantity is predicted given a set of inputs.
3. SGDClassifier - The model optimizes the log loss function using stochastic gradient descent.

TAKE AWAY FROM THE PROJECT :

We learnt about the working of spark streaming,which is scalable and fault tolerant and how we can stream data in batches and perform computation. We learnt how to incrementally train the model on a set of rdd's(dstream) and obtain the final weights, which was used for prediction on the test dataset. We also learnt how to fine-tune the batch-size which different values affect the data prediction and accuracy. We learnt to implement MiniBatchKmeans and visualised how clustering/unsupervised learning works for big data.