BIG DATA PROJECT - REPORT

Project Title :- Spam mail Detection using Machine Learning and Spark

Design details:-

1. The RDD streamed by the stream.py is in JSON format, so it was first converted from json to dictionary.
2. Then that json file should be converted to a dataframe so that we can use the data to apply machine learning statistics.
3. It was necessary to label and clean the features and hence we have used
    - Tokenizer
    - StopWordsRemover
    - CountVectorizer
    - IDF - Inverse Document Frequency
4. Then we have applied Multiple Models such as :-
    - Logistics Regression
    - SGD classification
    - Random Forest Classifier
5. Pipelining has been used to serially implement all the pre-processing and the final Model classifier to show various outputs in the terminal.
6. At last we deliver the performance of model using metrics like:-
    - F-1
    - Accuracy
    - Precision
    - Recall
7. K-Means Clustering has been used to analyze the clusters obtained

Surface level implementation details:-

1) SparkContext running on 2 threads :- local[2]
2) Streaming spark with time for each batch as 5 seconds
3) SparkSession Builder called
4) RDD is received using Streaming Spark and socketTextSTream function which accesses localhost:6100
5) forEachRDD helps to use each RDD in DStream and RDDtoDF function is used.
6) RDDtoDF function inturn calls rdd_to_json which converts the json RDD to dictionary
7) Then the conversion of output to dataframe and 'cleaning' function is being called to implement the pipeline.
8) Second Column in the DataFrame or 'Feature 1' holds the data for evaluation so we apply the pipeline in this fashion:-
   Tokenizer -> StopWordsRemover -> CountVectorizer -> Inverse Document Frequency -> StringIndexer and for final values -> VectorAssembler.
    a) Tokenizer :- tokenizes the values from the feature column
    b) StopWordsRemover :- removes the stopwords from the tokenized values
    c) CountVectorizer :- converts the stop words to a count vector

d) IDF :- finds the frequency of each word

e) String Indexer :- converts each string to an unique integer value

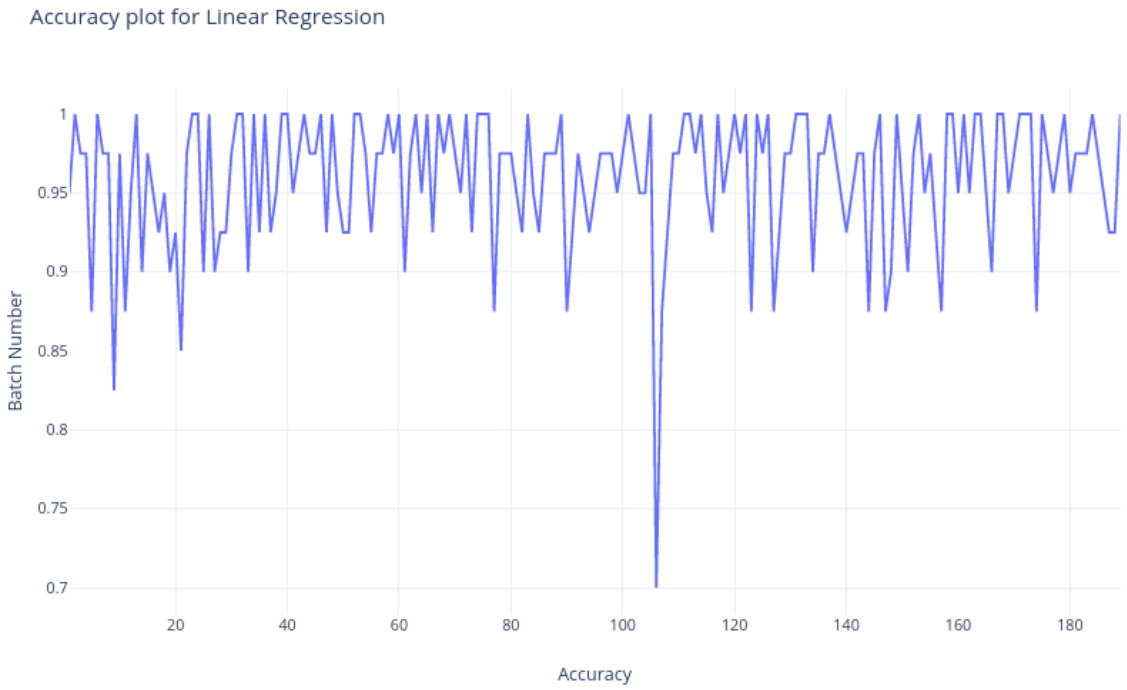
f) Vector Assembler :- gives out the final output in the pipeline

9) Then the Models are called :- predict and fit functions of each classifier

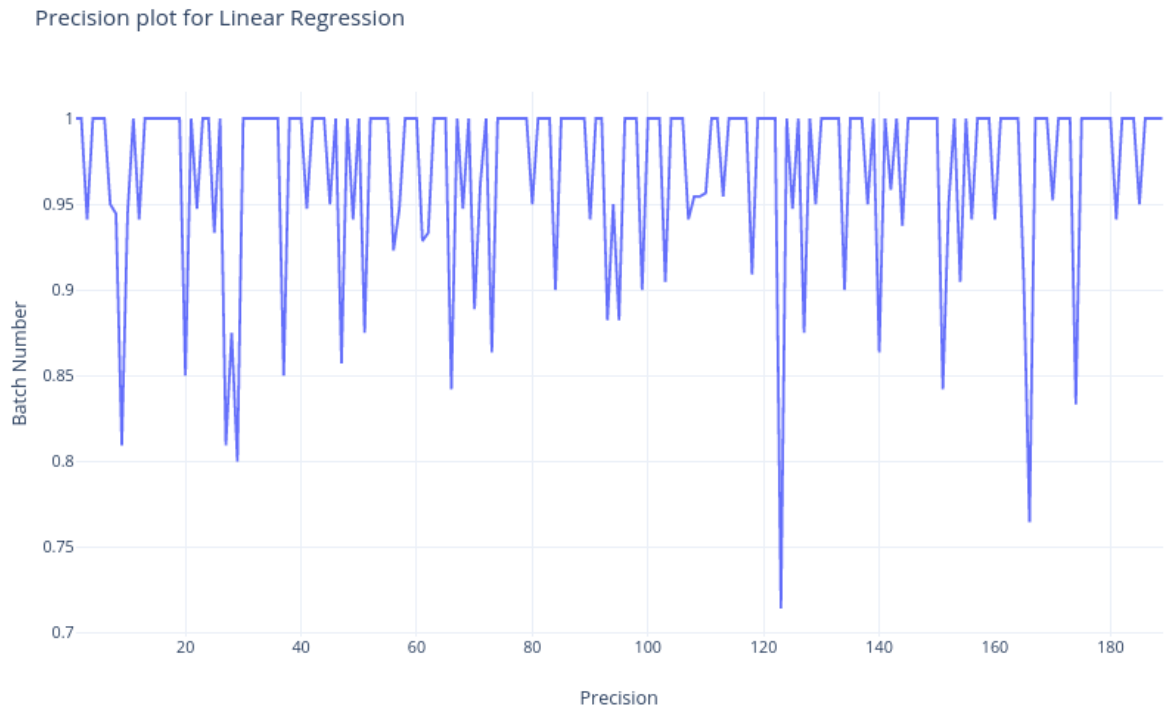
10) Performance Scores like Accuracy,F1 , Precision and recall are printed and plots are built using matplotlib.

Plots for Linear Regression vs it's metrics :

Accuracy :

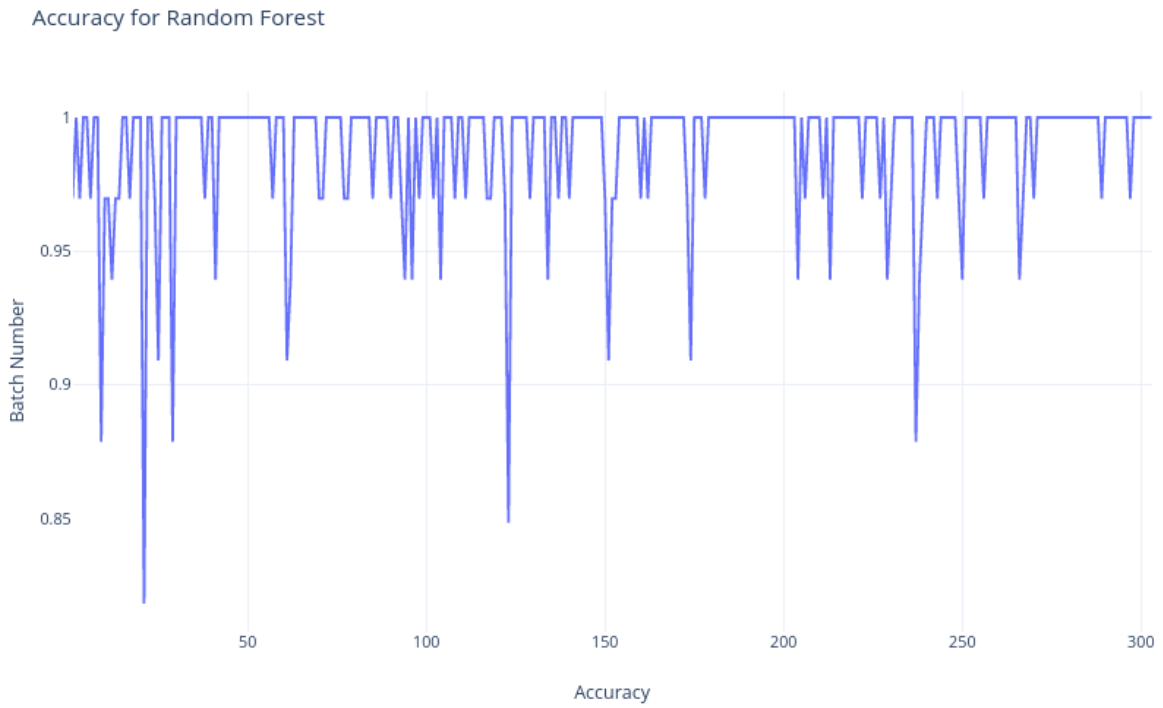


Precision :

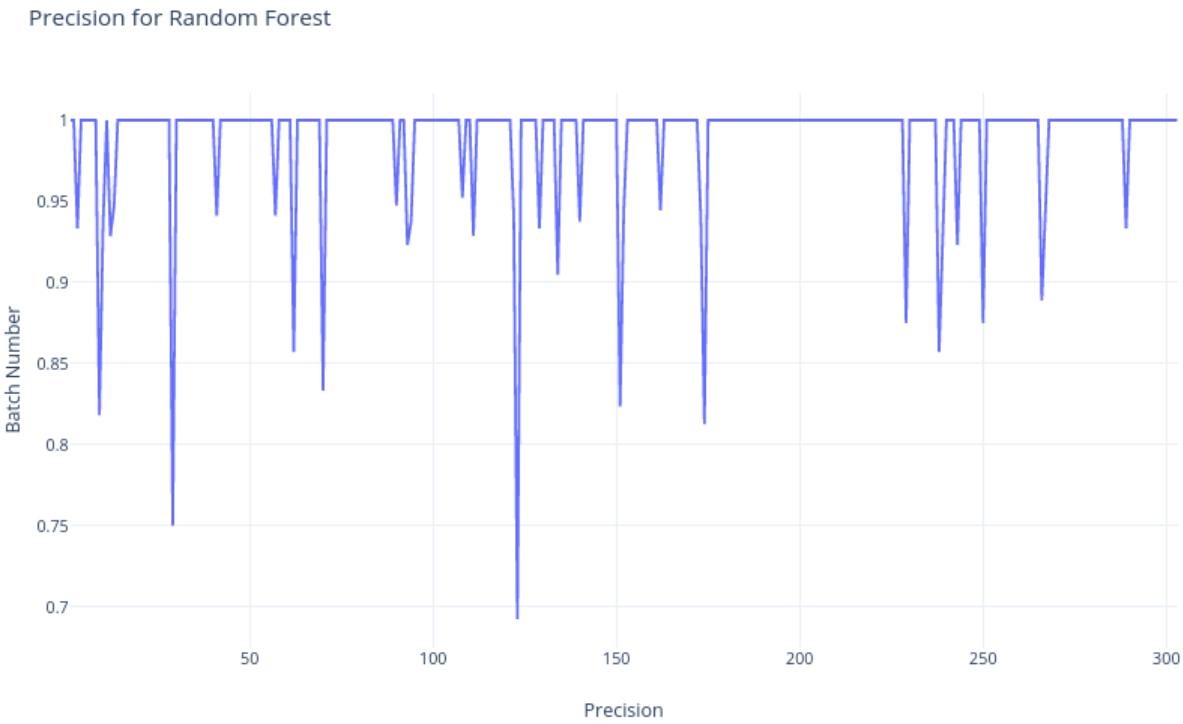


Plots for Random Forest vs it’s metrics:

Accuracy Plot :

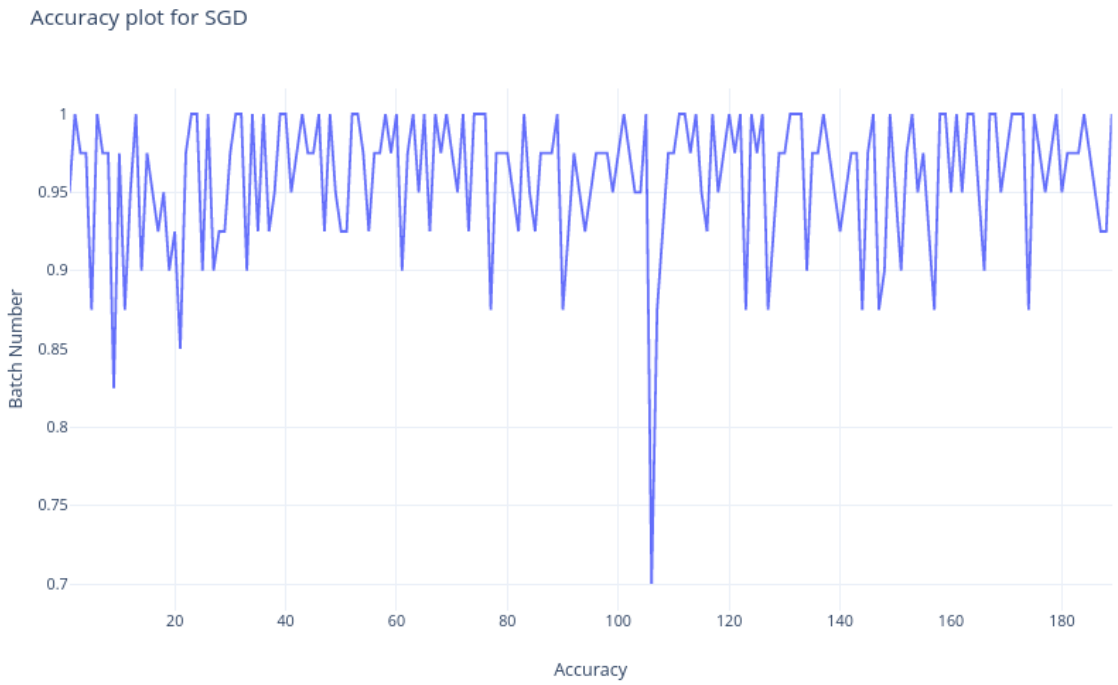


Precision :

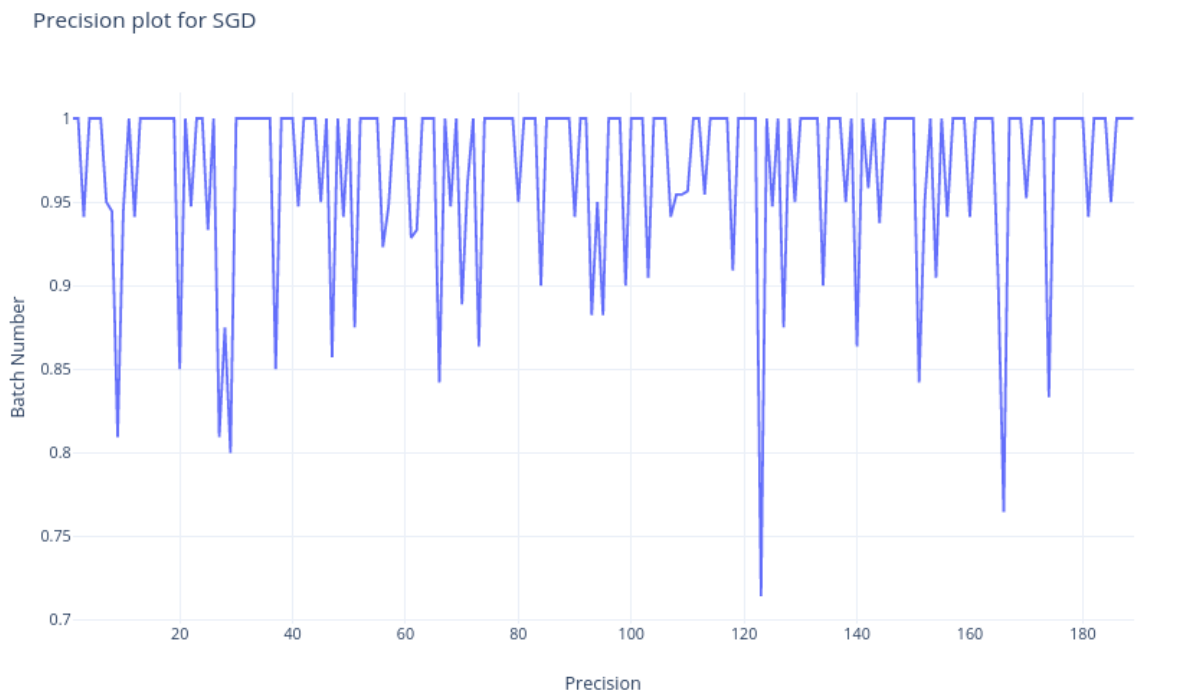


Plots for Stochastic Gradient decent for it’s metrics :

Accuracy :



Precision :



Reasons for design implementation:-

It was necessary to clean the data as we don’t know what type of characters may be used and also we had to find a relationship between the words and context commonly used in spam emails. That's why we used multiple preprocessing techniques. Also we had binary classification of Spam and no spam so Random Forest and Logistics regression were the best choices.

Takeaway from the project:-

Learned how spark streaming works and how huge data files are streamed in batches to process them in an easier way and learnt the working of machine learning models. The most accurate model was Linear regression followed by Random Forest. SGD was the weakest model and failed to display performance plots with high scores.