Big Data Project Report

Machine Learning using Spark Streaming

Project title: Spam classifier

Spam detection is a supervised machine learning problem. We provide the machine learning model with a set of examples of spam and ham messages and let it find the relevant patterns that separate the two different categories.The dataset chosen had 30000 rows.

Github Colab

Design Details

The codebase for this project is split into 2 modules , spark_fetch.py and process_temp.py.

- **spark_fetch.py:** This module fetches the data (json strings) streamed from the localhost at port 6000 , the data received from the spark streaming is in the form of dstream. This dstream is nothing but collection of rdd's which is looped through and then json strings are parsed to get rdd of the rows , which is sent to process_temp.py file for preprocessing.
- **process_temp.py:** This module receives the rdd of rows from the previous models and converts it to spark dataframe. All the functions such as text preprocessing , model fitting and writing the result to a csv file happens in this module.

Surface level implementation details about each unit

spark_fetch.py module

→ In this module the very first step is to import all the important libraries for spark streaming and processing. After which a spark streaming context is set up which reads at the port 6100 of the localhost. From where it receives the streams of data.

→ After receiving the streams of data (here dstreams) , no operation can be informed , so thus we have to loop through the dstreams with the help of the function foreachRDD and process the rdd to parse the json strings received and convert to rows which can be used to create a spark dataframe for further analysis and processing.

→ Once the json strings have been parsed and converted to rows in rdd form , it is sent to the process_temp.py file for further text processing and model building.

process_temp.py module

→ This module has many different functions for performing various tasks like preprocessing the text data , functions for various classifier models and functions to write the performance metric score in csv files.

→ The function preprocess in this module performs the text preprocessing , which are tokenizer, stop word removal , bigram , word2vec all these preprocessing techniques have been put in a pipeline. These techniques are from the spark ml library. The features that were processed were the subject and content of the message attribute(2 features).

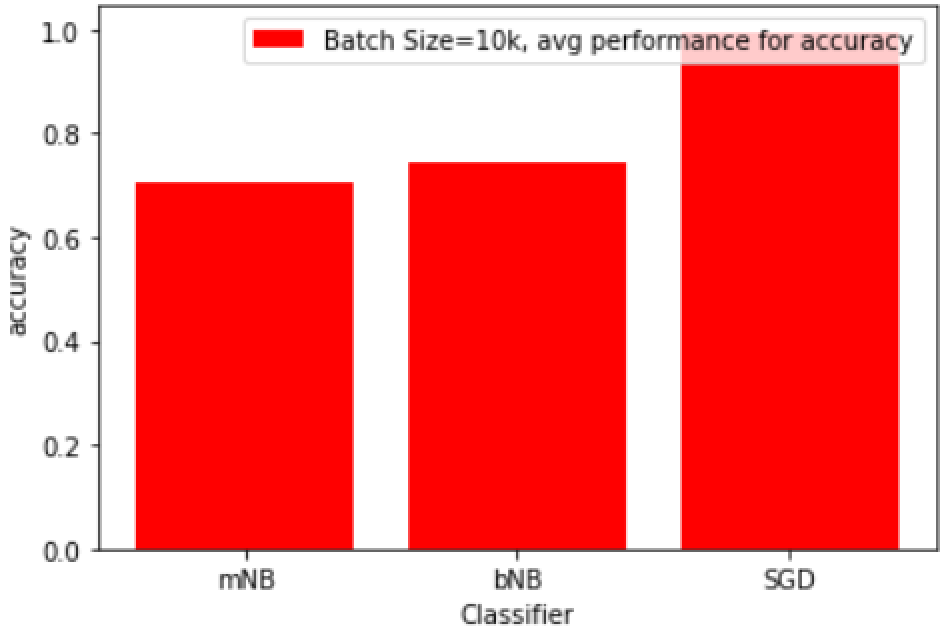
→ There were separate functions created for each of the classifier models used namely:

- The bernoulli naive bayes model
- Multinomial naive bayes model
- Stochastic gradient model

All these models support increment learning , and hence have been used.
Apart from these models , we also implemented the **K Means clustering model which is an unsupervised machine learning model.**

For all these models the way incremental learning was performed was , whenever the very first batch of data was received , the model was fit using the partial_fit method and was dumped into a pickle file (the library used here was the joblib), and from the next upcoming batches of data onwards the model was loaded from these pickle files , used the partial_fit method and dumped into the same pickle file. This process was continued until all the batches were streamed.

→ Finally the batches of data received were split into 80:20 ratio and the testing of the model was done on this.There was a separate function for calculating the performance metrics, which basically calculated the accuracy score , precision , recall and the f1-score.



From the above graph result we came to the conclusion that amongst the supervised learning model the SGD performed the best. (This was done with a batch size of 10000).

Take Away from the project

Spark is a lightning fast cluster computing technology, designed for fast computation. Since it is based on Hadoop MapReduce, it extends the MapReduce model to efficiently use it for more types of computations such as interactive queries and stream processing.
By using this with MLlib we provided the model with test and train datasets, we let the model classify between spam and ham messages. MLlib increased the efficiency and streaming since its a distributed machine learning framework it’s nine times as fast as the Hadoop disk-based version of Apache Mahout.