Big Data Project Report

BD_057_058_213
ANAGHA H M, ANANYA UPPAL, KARTIKA NAIR
06TH DECEMBER 2021

Project Title Chosen

The chosen project title is “Machine Learning with Spark Streaming” for the “Sentiment Analysis” dataset.

Design Details

The “main” file initially calls the “PreProcessing” file in order to clean the data received via Spark Streaming. Following this, the Logistic Regression, Naive Bayes, SVM, and K-Means Clustering models are called within the “main” file from the “models” directory. The “__init__.py” file in this directory is to ensure smooth operation. The output in the terminal includes the accuracies of the Logistic Regression, Naive Bayes, and SVM models, along with the error of the K-Means Clustering model. In a similar fashion, the incremental versions of the aforementioned models were created via scikit-learn and run via the “incrementalMain” file.

Surface Level Implementation

The pre-processing of the data is done by using RegEx replacement methods to remove hyperlinks, usernames, and whitespaces, as well as to ensure matching of the same word in different cases (capital and small letters) for the same.

The Logistic Regression, Naive Bayes, SVM, and K-Means Clustering models are implemented by first creating a pipeline consisting of a tokenizer, the hashed term frequencies, the inverse document frequency, and the string index labels. The streamed dataframe is then split for training and validation, followed by fitting and transformation of the same via the PySpark MLlib module functions for the respective models. Finally the accuracy or loss is obtained through an appropriate evaluator.

For incremental learning, the models were once again passed via a pipeline. However, this time, they were dumped into a file and loaded for testing via joblib.

Reasons behind Design Decisions

Regular Expressions were used for the pre-processing of the data as it allowed for more flexibility in the substrings to be filtered, which was crucial owing to a great deal of variety in the existing usernames and URLs. It also significantly reduced the number of lines of code required for pre-processing, making it simpler and cleaner.

The Logistic Regression, Naive Bayes, SVM, and K-Means Clustering models were chosen owing to ease of implementation and comparison in both the non-incremental and incremental methods.

The Binary Classification Evaluator was used for the Logistic Regression, Naive Bayes, and SVM models as they are classification algorithms, and the given dataset has only two labels (0 and 4). However, the K-Means Clustering model used a Clustering Evaluator as it is a clustering algorithm, not a classification algorithm.

Take Away from the Project

The main take away from this project was knowledge regarding error messages and debugging PySpark code. Another discovery made was that PySpark does not support incremental learning, which is why scikit-learn was used for the same.

Is clustering a form of unsupervised classification?

Yes, clustering is a form of unsupervised classification.

Does increasing the batch size increase performance?

No, the models perform better with a smaller batch size.

Which hyperparameters provide maximum changes to your model's performance? Why?

The change in batch size provided maximum changes to the model's performance. This is because the entire training procedure, and the number of times the procedure is conducted, both depend on the batch size, and hence, this is more effective than any change within the model itself.