UE19CS322 BIG DATA PROJECT 2

MACHINE LEARNING WITH SPRAK STREAMING

PES UNIVERSITY

NAME	SRN
RAVI AISHWARYA	PES2UG19CS322
PAVITRA K	PES2UG19CS279
NANDINI N	PES2UG19CS248
RASHMI	PES2UG20CS808

GitHub Repository : https://github.com/pavitrak8/BD2_248_279_322_808_SparkML

I. PROJECT TITLE

Machine Learning with Spark ML Lib**: SENTIMENTAL ANALYSIS**

II. DETAILED DESIGN

Apache Spark is a data processing framework that can handle big data sets quickly and distribute processing jobs across numerous computers, either on its own or in conjunction with other distributed computing tools.

Apache Spark is a distributed processing solution for big data workloads that is open-source. For quick analytic queries against any size of data, it uses in-memory caching and efficient query execution.

Spark MLlib is utilized to perform machine learning in Apache Spark.

III. SURFACE LEVEL IMPLEMENTATION DETAILS ABOUT EACH UNIT

UNIT 1– Streaming the data from stream.py

The streaming component in Apache Spark, which runs on top of Spark, allows for strong interactive and analytical applications across both streaming and historical data while preserving Spark's ease of use and fault tolerance.The data is streamed using stream.py The data is huge so we will make it as batches with batch_size and run . Data is received using a TCP socket and receive byte is interpreted using SocketTextStream and local host with port number 6100

UNIT 2- Reading the streaming data and converting it to dataframe

Now the dataset is streaming in batches .

RDD is created in SparkContext and it connects to spark cluster and DataFrame is created in SparkSession which is an entry point to start spark programming with df and dataset.

The batch sizes can be controlled accordingly and the accuracy or the other performance metrics depends on baatch_size as well

The DataFrame is created from streaming nested json .

Each RDD is a single batch which is converted to a DataFrame on which the Mllib operations/functions are performed

UNIT3 –Text Classification Preprocessing

Text preprocessing is a technique for cleaning text data and preparing it for use in a model. Text data comprises noise in the form of emotions, punctuation, and text in a different case, among other things.

RegexTokenizer:

This class allows us to vectorize a text corpus by converting each text into either a sequence of integers each integer representing the index of a token in a dictionary or a vector of each coefficient representing a token in a dictionary based on word count and based on tf-idf.This allows us to specify a pattern in the text to tokenize.

StopwordsRemover:

Stop words are commonly used in Text Mining and Natural Language Processing (NLP) to eliminate words that are so commonly used that they carry very little useful information.Stop words are available in abundance in any human language. By removing these words, we remove the low-level information from our text in order to give more focus to the important information.

CountVectorizer:

Produces a CountVectorizerModel by extracting a vocabulary from document collections. The goal of the CountVectorizerModel is to assist in the conversion of a set of text documents into token count vectors.

StringIndexer:

Converts a single column to an index column using the StringIndexer function.

A label indexer that converts a string column of labels into an ML column of label indexes. If the input column is numeric, it is converted to a string and the values are indexed.

Unit 5-Model Selection

pipeline:ML Pipelines provide a standardised set of high-level APIs built on top of DataFrames to assist users in building and tuning actual machine learning pipelines. In order to specify an ML workflow, a Pipeline connects many Transformers and Estimators.

ParamGrid:

Builder for a param grid that is used to choose models via grid search. Sets the settings in this grid to their default values.

Cross-Validation:

Model selection, or using data to determine the optimum model or parameters for a given job, is an important issue in machine learning. Tuning is another term for this. Individual Estimators, such as LogisticRegression, can be fine-tuned, as can complete Pipelines that comprise numerous algorithms, featurization, and other processes. Instead of tuning each element in the Pipeline individually, users can tune the entire Pipeline at once.

Model selection in MLlib is made possible by tools like CrossValidator and TrainValidationSplit We did train test split as it is the classification problem which helps us in finding out the solutions real-quick

The following elements are required to use these tools:

Estimator: a tuning method or pipeline

SetofParamMaps:Parameters to choose from, commonly referred to as a "parameter grid" to search over

Evaluator: a statistic for determining how well a fitted Model performs on test data that has been held out.

UNIT 4 – Model Building

1.Naive Bayes

2.Logistic Regression

3.Random Forest

4.Clustering Algorithm

UNIT 5-Performance Metrics

We have performed Confusion Matrix,Accuracy,Precision,Recall to know how accurate our model is and how good are the predictions are .

Inferences:

We also compare all the 4 different models and decide which model gives best accuracy .

We found out thaat naïve bayes is the best model followed by Logistic Regression since naïve bayes predicts using probabilties for each attribute and much efficient,robust.

IV. REASON BEHIND DESIGN DECISIONS

1.**Naive Bayes** : It can be used for Binary as well as Multi-class Classifications. It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner. It is used in Text classification such as Sentiment Analysis and Spam detection as well. Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in **text classification and with problems having multiple classes**.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

2.**Logistic Regression:** Logistic regression is more straightforward to use, interpret, and train it

It classifies unfamiliar records really quickly.

The categorical dependent variable is predicted using logistic regression. In other words, it's used when making a categorical prediction, such as yes or no, true or false, 0 or 1,spam or ham etc.

It can use model coefficients to determine the importance of different features.In the case of several explanatory variables, logistic regression is utilized to get the odds ratio.

The impact of each variable on the chances ratio of the observed event of interest is calculated as a result.

3.**Random Forest:**

To generate a more accurate and reliable prediction, random forest creates numerous decision trees and blends them together. You can use the algorithm's regressor to cope with regression tasks with random forest. While growing the trees, the random forest adds more randomness to the model.However for a much higer dimensional sparse data the performance of this model is not the best one.

4.**Clustering Algorithm: K means clustering**

It is an unsupervised Machine Learining algorithm.

To locate groups that haven't been explicitly identified in the data, the K-means clustering technique is utilised. This ensures convergence and can be used to confirm business assumptions about what types of groups exist or to identify unknown groups in large data sets.

We took k=3 (Always take k as odd to avoid ambiguity in deciding the cluster)

V. TAKEAWAY FROM THE PROJECT

From this sentimental analysis project, we learnt how to stream the given dataset and how to convert to dataframe from streaming nested json . Preprocessing the dataset –Text classification,training the model,building the models and testing the model by making the predictions. We learnt spark Mllib RDD operations,preprocessing functions such as Regextokenizer,StopWordsRemover,VountVectorizer,StringIndexer.we learnt how we to apply machine learning algorithms using the PySpark library.we noticed which model performs better and with which batch_size by experimneting with different batch sizes.

The batch sizes are changed everytime and experimented on it to find out which performs best on the training batch.

We tried with 10,100,1000,10K batch_size and noticed that the more the batch size,more it is faster and showing the best results.