Distributed Systems

Assignment 5 - Comparison of Hadoop MapReduce and Spark
Assignment given on: 20th Nov 2020
Due Date: 30th Nov 2020

Objective

The objective of this assignment is to understand the performance of Hadoop MapReduce and Spark for distributed computing. You need implementing Logistic Regression to train the model to predict the Heart disease.

Dataset

The data set is given in .csv format(heart_disease.csv). The classification goal is to predict whether the patient has 10-years risk of future coronary heart disease (CHD). The dataset provides the patients' information. It includes around 4000 records and 16 attributes. TenYearCHD(16^{th}) Attribute is the Class Label.

male: male or female(Nominal)
age: Age of the patient;(Continuous — Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)
education: Education of the patient.
Current Smoker: whether or not the patient is a current smoker (Nominal)
Cigs Per Day: the number of cigarettes that the person smoked on average in one day.(can be considered continuous as one can have any number of cigarettes, even half a cigarette.)
BP Meds: whether or not the patient was on blood pressure medication (Nominal)
Prevalent Stroke: whether or not the patient had previously had a stroke (Nominal)
Prevalent Hyp: whether or not the patient was hypertensive (Nominal)
Diabetes: whether or not the patient had diabetes (Nominal)
Tot Chol: total cholesterol level (Continuous)
Sys BP: systolic blood pressure (Continuous)
Dia BP: diastolic blood pressure (Continuous)
BMI: Body Mass Index (Continuous)
Heart Rate: heart rate (Continuous — In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values.)
Glucose: glucose level (Continuous)
TenYearCHD: 10 year risk of coronary heart disease CHD (binary: "1", means "Yes", "0" means "No")

Implementation

This program should be implemented and executed in both Spark and Hadoop. Train the model by analyzing the existing data. The trained model can be used to predict if users suffer from heart disease.
Note: We will exclude the prediction(testing) part.

To understand Logistic Regression refer the following link

http://rasbt.github.io/mlxtend/user_guide/classifier/LogisticRegression/#logistic-regression

A sample spark code for **Logistic Regression Gradient** show below

```
from pyspark import SparkContext
import numpy as np
import numpy . random as nr
from operator import add
n =10000
x = nr . randn (n )
y = [ nr . binomial (1 ,1/(1+ np . exp ( -1 - xi )))*2 -1 for xi in x]
xy = np . column_stack ((x ,y )). tolist ()
sc = SparkContext () #initialize the spark context
data = sc . parallelize ( xy ). cache ()
iterations = 100; P = 2 # Number of dimensions
# Initialize parameter w to a random value
w = 2 * nr . ranf ( size = P) - 1
# Compute logistic regression gradient for each data point
def gradient ( line , w ):
        Y = np . array ( line [1])
        X = np . array ([1 , line [0]])
# For each point (x, y), compute gradient function
return ((1.0 / (1.0 + np . exp ( -Y * X . dot ( w ))) -1.0 ) * Y * X. T)
for i in range ( iterations ):
        w -= data . map ( lambda m : gradient (m , w )). reduce ( add ) / n
sc . stop ()
```

You need to use different TRAIN_SET_COUNT(iterations) range from 10 to 100 to train the model, record the running time of the program for both Spark and Hadoop code.

Technical Report:

- Draw a graph for running time of Spark and hadoop code, x axis – number of iterations (TRAIN_SET_COUNT) and y axis – Running Time.
- Also include your observations.

What to Submit?

Please submit a zipped file which includes source code, output file, readme file, technical report.

Grading – 20 Marks

Source Code – 15 Marks

Technical Report and Readme – 5 Marks

Note:

- Penalty of 10% per day will be issued if the deadline is not met
- If found copied, you will fetch 0 score.

Submission Details:

1. Please read the questions carefully and complete it.

2. Make a directory with name <Your_Roll_Number> and copy your all program (source code) and output file to that folder.

3. You can implement using any programming language.

4. Please screen shot of output, name it with its question number and put it in a same folder.

5. Test well before submission. Follow some coding style uniformly. Provide proper comments in your code.

6. Submit only through moodle and well in advance. Any hiccups in the moodle/Internet at the last minute is never acceptable as an excuse for late submission. Submissions through email will be ignored.

7. Please zip your folder and submit throgh moodle within 30-11-2020 (23:55 PM)