AIML427 Big Data: Assignment 3

1. Pseudocode:

**DecisionTree Pseudocode:**

Set start time

Create new spark session

Read data as csv

Add column headers to data

Set all but the last column to Double type, with String for the last

Split data into 70% training and 30% testing

Create labelled points for each row in the data

Generate decision tree model

Calculate the test and training error

Set the duration from the start time

Print results

Save results to file

**Logistic Regression Pseudocode**

Configure new spark session and spark context

Load the data as RDD string

Convert RDD to javaRDD string.

Randomly split the data by 7:3 ratio using random seed

Use 70% of the data as training set and use the rest for testing the model

Convert the class label for both test and training set into binary form for class prediction

Configure new Logistic regression model

Train the model using the labelled training data.

Test the model using the labelled test data.

Calculate accuracy and test error of the Logistic Regression model.

1. Read Me file is attached.

|  |  |
| --- | --- |
|  | Spark’s Decision Tree algorithm |
| Run 1 | Train Accuracy: 0.9958091360833383  Train Error: 0.0041908639166616775  Test Accuracy: 0.9862526168876483  Test Error: 0.01374738311235171  Duration: 22ms |
| Run 2 | Train Accuracy: 0.9960365944512323  Train Error: 0.003963405548767768  Test Accuracy: 0.9860356865787432  Test Error: 0.013964313421256789  Duration: 21ms |
| Run 3 | Train Accuracy: 0.9968334578043316  Train Error: 0.003166542195668409  Test Accuracy: 0.9863964658859827  Test Error: 0.01360353411401725  Duration: 24ms |
| Run 4 | Train Accuracy: 0.995685393258427  Train Error: 0.004314606741573033  Test Accuracy: 0.9857948610820974  Test Error: 0.014205138917902652  Duration: 22ms |
| Run 5 | Train Accuracy: 0.996297522651335  Train Error: 0.003702477348665001  Test Accuracy: 0.9863589390285911  Test Error: 0.013641060971408887  Duration: 22ms |
| Run 6 | Train Accuracy: 0.9964632805302107  Train Error: 0.003536719469789283  Test Accuracy: 0.9864433245794592  Test Error: 0.013556675420540848  Duration: 21ms |
| Run 7 | Train Accuracy: 0.9958177744585511  Train Error: 0.004182225541448843  Test Accuracy: 0.9866769511254471  Test Error: 0.013323048874552977  Duration: 21ms |
| Run 8 | Train Accuracy: 0.9974318392211903  Train Error: 0.002568160778809687  Test Accuracy: 0.9869464523826234  Test Error: 0.013053547617376657  Duration: 22ms |
| Run 9 | Train Accuracy: 0.9959031100478469  Train Error: 0.00409688995215311  Test Accuracy: 0.9871992165640738  Test Error: 0.012800783435926133  Duration: 22ms |
| Run 10 | Train Accuracy: 0.9967978452790663  Train Error: 0.0032021547209337125  Test Accuracy: 0.9863138049018924  Test Error: 0.013686195098107675  Duration: 23ms |

|  |  |
| --- | --- |
|  | Logistic Regression algorithm |
| Run 1 | Train Accuracy: 0.888696534638689  Train Error: 0.11130346536131097  Test Accuracy: 0.8888498559685238  Test Error: 0.11115014403147616  Duration: 19ms |
| Run 2 | Train Accuracy: 0.8879194027165602  Train Error: 0.11208059728343978  Test Accuracy: 0.8914146680570039  Test Error: 0.10858533194299613  Duration: 21ms |
| Run 3 | Train Accuracy: 0.8908443327047978  Train Error: 0.10915566729520221  Test Accuracy: 0.8841884816753927  Test Error: 0.11581151832460734  Duration: 20ms |
| Run 4 | Train Accuracy: 0.8890255610224409  Train Error: 0.1109744389775591  Test Accuracy: 0.8871146903637879  Test Error: 0.11288530963621213  Duration: 21ms |
| Run 5 | Train Accuracy: 0.8871759564150152  Train Error: 0.11282404358498477  Test Accuracy: 0.8893230983949756  Test Error: 0.11067690160502441  Duration: 20ms |
| Run 6 | Train Accuracy: 0.8886305371728372  Train Error: 0.11136946282716276  Test Accuracy: 0.8846370683579986  Test Error: 0.1153629316420014  Duration: 25ms |
| Run 7 | Train Accuracy: 0.8860425353211147  Train Error: 0.11395746467888535  Test Accuracy: 0.889436766442114  Test Error: 0.110563233557886  Duration: 22ms |
| Run 8 | Train Accuracy: 0.8862539349422875  Train Error: 0.11374606505771245  Test Accuracy: 0.887351366386204  Test Error: 0.11264863361379596  Duration: 31ms |
| Run 9 | Train Accuracy: 0.8876163206871869  Train Error: 0.11238367931281312  Test Accuracy: 0.8915399774774775  Test Error: 0.10846002252252251  Duration: 23ms |
| Run 10 | Train Accuracy: 0.8877807726864331  Train Error: 0.11221922731356693  Test Accuracy: 0.8884706538407918  Test Error: 0.11152934615920818  Duration: 23ms |

**DT Training Std dev** = 0.000053

**DT Testing Std dev** = 0.00038

**LR Training Std dev** = 0.00132

**LR Testing Std dev** = 0.0025

1. Compare and discuss the results of the two models

In this report, we have compared two models i.e. Spark’s Decision Tree algorithm and Logistic Regression algorithm. We can see from the output that the training and testing accuracy of Spark’s Decision Tree algorithm is higher than the other model. When computing the accuracy, , we have used the data that is divided into 70% training set and 30% testing set. The decision tree prediction is ----% and Logistic regression is ----%.

When coming one of the run that have taken the highest timings from both algorithms as shown in the table below. We can see that the training accuracy of decision tree is higher than the logistic regression that shows that the training error decision tree is very low. The testing accuracy of decision tree is also higher than the logistic regression.

|  |  |
| --- | --- |
| Decision Tree Run | Train Accuracy: 0.9968334578043316  Train Error: 0.003166542195668409  Test Accuracy: 0.9863964658859827  Test Error: 0.01360353411401725  Duration: 24ms |
| Logistic Regression Run | Train Accuracy: 0.8862539349422875  Train Error: 0.11374606505771245  Test Accuracy: 0.887351366386204  Test Error: 0.11264863361379596  Duration: 31ms |

In the plot, it can be seen that Decision Tree algorithm performed better than the Logistic Regression algorithm, it can be seen by computing the mean.

Chart, line chart

Description automatically generated

**DT Training Std dev** = 0.000053

**DT Testing Std dev** = 0.00038

**LR Training Std dev** = 0.00132

**LR Testing Std dev** = 0.0025