BIG DATA ANALYTICS

Assignment 1

Balamanikandan Gopalakrishnan

Rachan Hegde

Contents

Data Analysis 2

Data Normalization 3

Performance Metrics 3

SciPy – Classification Algorithm 3

SciPy – Regression Algorithm 4

Apache Spark - Classification Algorithm 5

Apache Spark - Regression Algorithm 7

SciPy libraries vs Apache Spark 9

Appendix 9

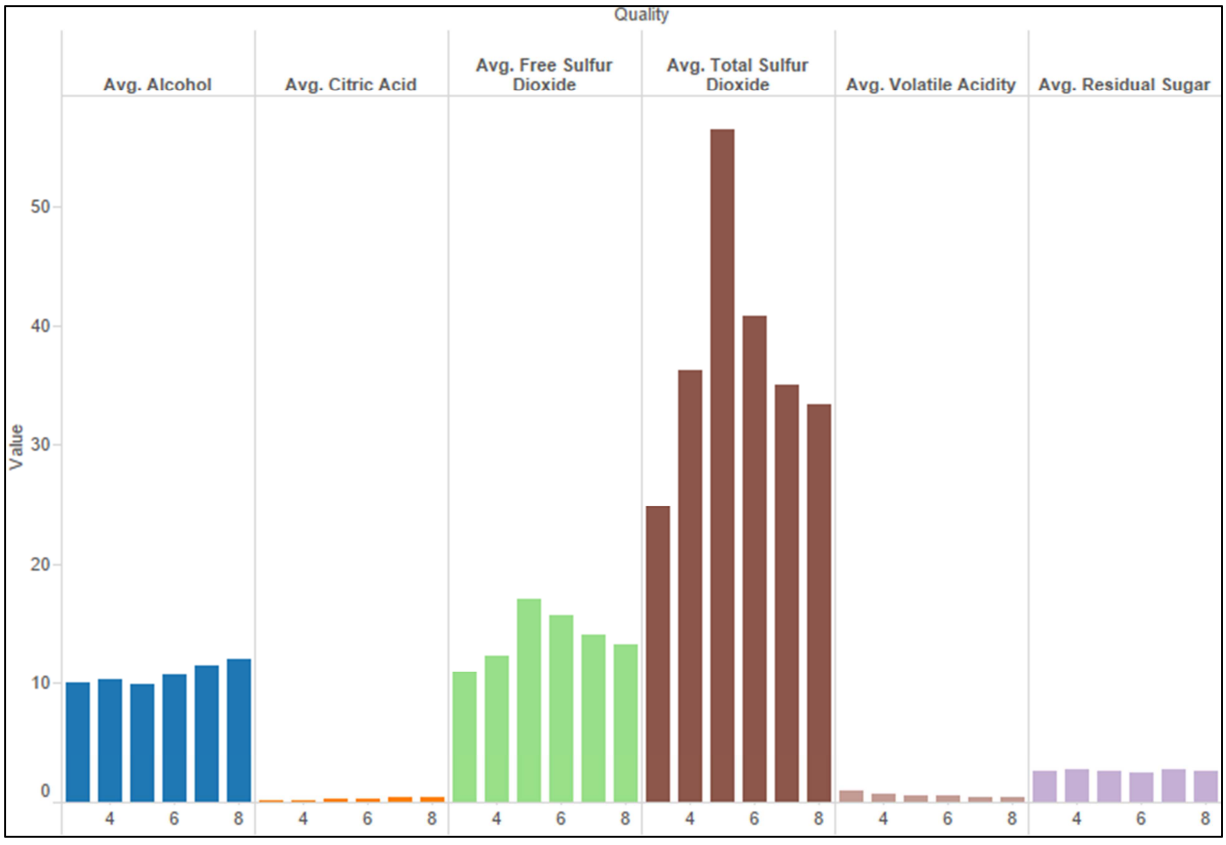
Instructions to execute 9

Data Analysis

The dataset was complete and didn't require additional cleansing. Visual analysis was performed to understand the relationship between features and the prediction variable to identify most significant features.

The following features seemed to have the most impact the prediction variable from our initial data analysis. The variance in other variables were minimal.

- Alcohol
- Citric Acid
- Free Sulfur Dioxide
- Total Sulfur Dioxide
- Volatile Acid
- Residual Sugar



On predicting the values by just considering these features, predictions were skewed and so all the input features were considered for the data analysis.

Data Normalization

Data normalization was performed by using the MLIB inbuilt normalization techniques to analyze the impact of normalization. On applying classification/ regression models to the normalized features, the predictions were skewed. Manual data normalization was performed and models were applied to cross check this behavior and the results were skewedness consistent. So data provided was used as is for creating the model.

Performance Metrics

SciPy – Classification Algorithm

The following performance metrics were used in evaluating the classification algorithm

- Accuracy
- Precision
- Recall

	L0	L2	L1
SGD (Hinge)	Training Accuracy: .8 Precision: .58 Recall: .27 ROC: .61 Test Error: .71 Precision: .41 Recall: .65 ROC: .69	Training Accuracy: .79 Precision: .53 Recall: .11 ROC: .54 Test Error: .75 Precision: .44 Recall: .37 ROC: .62	Training Accuracy: .81 Precision: .64 Recall: .32 ROC: .63 Test Error: .74 Precision: .44 Recall: .61 ROC: .69
SGD (Log)	Training Accuracy: .80 Precision: .59 Recall: .27 ROC: .61 Test Error: .72 Precision: .42 Recall: .65 ROC: .69	Training Accuracy: .79 Precision: .53 Recall: .19 ROC: .57 Test Error: .73 Precision: .43 Recall: .51 ROC: .66	Training Accuracy: .81 Precision: .65 Recall: .27 ROC: .61 Test Accuracy: .74 Precision: .45 Recall: .56 ROC: .68
Logistic Regression (LBFGS)	NA	Training Accuracy: .81 Precision: .63 Recall: .32 ROC: .63 Test Accuracy: .73 Precision: .44 Recall: .58 ROC: .68	Training Accuracy: .82 Precision: .63 Recall: .32 ROC: .63 Test Accuracy: .73 Precision: .44 Recall: .58 ROC: .68

SciPy Classification Configurations

The following configurations were used for the algorithms tested using

Data Split: 60, 40

The data was randomly split and 60% was used for training the model and 40% was used for testing the created model.

Fit Intercept: True

This optimization attribute was set to true to enable algorithm to use intercepts while creating the model which reduced the error. This is also enabled to reduce the prediction skewedness.

Alpha: .01

Alpha is a constant used to multiply the regularization term which is defaulted to 1E-4. On increasing this to .01, the accuracy was improved.

Tolerance: .001

Tolerance is accepted stopping criteria in LBFGS algorithms. The value .001 returned better predictions

Shuffle: False

Disabled the option to shuffle the data after each epoch. This is set to true by default and makes it hard to validate against models created.

Iterations: 200

Maximum of the actual number of iterations across all classes.

Note: Few of these configurations are not applicable for some of the algorithms.

SciPy – Regression Algorithm

The following performance metrics were used in evaluating the classification algorithm

- Mean Squared Error

	L0	L2	L1
Linear Regression	Training RMS Error: .59 Test RMS Error: .54	NA	NA
SGD Regressor	Training RMS Error: 1.17 Test RMS Error: 8.9	Training RMS Error: 1.61 Test RMS Error: 1.29	Training RMS Error: 1.55 Test RMS Error: 1.26
Ridge	NA	Training RMS Error: .59 Test RMS Error: .54	NA
Lasso	NA	NA	Training RMS Error: .6

			Test RMS Error: .55

SciPy Regression Configurations

The following configurations were used for the algorithms tested using

Data Split: 60, 40

The data was randomly split and 60% was used for training the model and 40% was used for testing the created model.

Fit Intercept: True

This optimization attribute was set to true to enable algorithm to use intercepts while creating the model which reduced the error. This is also enabled to reduce the prediction skewedness.

Alpha: .01

Alpha is a constant used to multiply the regularization term which is defaulted to 1E-4. On increasing this to .01, the accuracy was improved.

Tolerance: .001

Tolerance is accepted stopping criteria in LBFGS algorithms. The value .001 returned better predictions

Shuffle: False

Disabled the option to shuffle the data after each epoch. This is set to true by default and makes it hard to validate against models created.

Iterations: 200

Maximum of the actual number of iterations across all classes.

Note: Few of these configurations are not applicable for some of the algorithms.

Apache Spark - Classification Algorithm

The following performance metrics were used in evaluating the classification algorithm

- Error
- Precision
- Recall
- Confusion Matrix

	Scala		Python		
	L2	L1	L2	L1	L0
SVM With SGD	Training Error: .23 Precision: .76 Recall: .76 Precision (1): .13 Recall (1): .018 Precision (0): .78 Recall (0): .96 ROC: .49 Test Error: .24 Precision: .75 Recall: .75 Precision (1): .13 Recall (1): .018 Precision (0): .78 Recall (0): .96 ROC: .49	Training Error: .23 Precision: .76 Recall: .76 Precision (1): .13 Recall (1): .018 Precision (0): .78 Recall (0): .96 ROC: .49 Test Error: .24 Precision: .75 Recall: .75 Precision (1): .13 Recall (1): .018 Precision (0): .78 Recall (0): .96 ROC: .49	Training Error: .24 Precision: .75 Recall: .75 Precision (1): .07 Recall (1): .009 Precision (0): .77 Recall (0): .96 ROC: .48 Test Error: .23 Precision: .76 Recall: .76 Precision (1): .11 Recall (1): .019 Precision (0): .76 Recall (0): .96 ROC: .49	Training Error: .24 Precision: .75 Recall: .75 Precision (1): .07 Recall (1): .009 Precision (0): .77 Recall (0): .96 ROC: .48 Test Error: .23 Precision: .76 Recall: .76 Precision (1): .11 Recall (1): .019 Precision (0): .76 Recall (0): .96 ROC: .49	Training Error: .24 Precision: .75 Recall: .75 Precision (1): .07 Recall (1): .009 Precision (0): .77 Recall (0): .96 ROC: .48 Test Error: .23 Precision: .76 Recall: .76 Precision (1): .11 Recall (1): .019 Precision (0): .78 Recall (0): .96 ROC: .49
LR With LBFGS	Training Error: .21 Precision: .78 Recall: .78 Precision (1): .55 Recall (1): .007 Precision (0): .78 Recall (0): .99 ROC: .50 Test Error: .21 Precision: .78 Recall: .78 Precision (1): .55 Recall (1): .007 Precision (0): .78 Recall (0): .99 ROC: .49	Training Error: .13 Precision: .78 Recall: .78 Precision (1): .53 Recall (1): .071 Precision (0): .79 Recall (0): .98 ROC: .52 Test Error: .21 Precision: .78 Recall: .78 Precision (1): .53 Recall (1): .071 Precision (0): .79 Recall (0): .98 ROC: .52	Training Error: .19 Precision: .80 Recall: .80 Precision (1): .67 Recall (1): .23 Precision (0): .81 Recall (0): .96 ROC: .6 Test Error: .19 Precision: .80 Recall: .80 Precision (1): .59 Recall (1): .21 Precision (0): .82 Recall (0): .96 ROC: .58	Training Error: .21 Precision: .78 Recall: .78 Precision (1): .62 Recall (1): .06 Precision (0): .78 Recall (0): .98 ROC: .52 Test Error: .19 Precision: .80 Recall: .80 Precision (1): .75 Recall (1): .07 Precision (0): .80 Recall (0): .99 ROC: .53	Training Error: .19 Precision: .80 Recall: .80 Precision (1): .63 Recall (1): .29 Precision (0): .85 Recall (0): .95 ROC: .62 Test Error: .19 Precision: .80 Recall: .80 Precision (1): .55 Recall (1): .27 Precision (0): .83 Recall (0): .94 ROC: .60
LR With SGD	Training Error: .27 Precision: .72 Recall: .72 Precision (1): .22 Recall (1): .10 Precision (0): .78 Recall (0): .90 ROC: .50	Training Error: .17 Precision: .72 Recall: .72 Precision (1): 21 Recall (1): .095 Precision (0): .78 Recall (0): .90 ROC: .49	Training Error: .23 Precision: .76 Recall: .76 Precision (1): .025 Recall (1): .0015 Precision (0): .77 Recall (0): .98	Training Error: .23 Precision: .76 Recall: .76 Precision (1): .05 Recall (1): .003 Precision (0): .77 Recall (0): .98 ROC: .49	Training Error: .23 Precision: .76 Recall: .76 Precision (1): .025 Recall (1): .0015 Precision (0): .77 Recall (0): .98 ROC: .49

	Test Error: .27 Precision: .72 Recall: .72 Precision (1): .22 Recall (1): .10 Precision (0): .78 Recall (0): .90 ROC: .49	Test Error: .27 Precision: .72 Recall: .72 Precision (1): .21 Recall (1): .095 Precision (0): .78 Recall (0): .90 ROC: .49	ROC: .49 Test Error: .22 Precision: .77 Recall: .77 Precision (1): .13 Recall (1): .012 Precision (0): .79 Recall (0): .97 ROC: .49	Test Error: .22 Precision: .77 Recall: .77 Precision (1): .16 Recall (1): .014 Precision (0): .79 Recall (0): .97 ROC: .49	Test Error: .22 Precision: .77 Recall: .77 Precision (1): .13 Recall (1): .012 Precision (0): .79 Recall (0): .97 ROC: .49

Scala Classification Configurations

The following configurations were used for the algorithms tested using

Data Split: 60, 40

The data was randomly split and 60% was used for training the model and 40% was used for testing the created model.

Step Size: .00001 (For SGD)

The step size was set to .00001 and on increase or decrease of this value for SGD algorithms, the error was increasing

Convergence Tolerance (For LBFGS)

A convergence tolerance was set to .001 from a default of 1E-4 which resulted in reduced error rate.

Intercept: True

This optimization attribute was set to true to enable algorithm to use intercepts while creating the model which reduced the error. This is also enabled to reduce the prediction skewedness.

Number of Iterations: 200

Increasing the number of iterations beyond 200 had minimal impact on predictions and so 200 was chosen as iteration count.

Apache Spark - Regression Algorithm

The following performance metrics were used in evaluating the classification algorithm

- Mean Squared Error

	Scala			Python		
	L0	L2	L1	L2	L1	L0
LR with SGD	Training RMS Error: 1.291 Test RMS Error: 1.283 (Step Size: .001)	Training RMS Error: 1.291 Test RMS Error: 1.283 (Step Size: .001)	Training RMS Error: 1.291 Test RMS Error: 1.283 (Step Size: .001)	Training RMS Error: 1.16 Test RMS Error: 1.167 (Step Size: .001)	Training RMS Error: 1.167 Test RMS Error: 1.167 (Step Size: .0001)	Training RMS Error: 1.167 Test RMS Error: 1.167 (Step Size: .0001)
Ridge	NA	Training RMS Error: 1.291 Test RMS Error: 1.283 (Step Size: .001)	NA	Training RMS Error: 1.16 Test RMS Error: 1.167 (Step Size: .001)	NA	NA
Lasso	NA	NA	Training RMS Error: 1.291 Test RMS Error: 1.283 (Step Size: .001)	NA	Training RMS Error: 1.39 Test RMS Error: 1.39 (Step Size: .001)	NA

Scala Regression Configurations

The following configurations were used for the algorithms tested using

Data Split: 60, 40

The data was randomly split and 60% was used for training the model and 40% was used for testing the created model.

Step Size: .001

The step size was set to .001 and on increase or decrease of this value, the RMS Error was increasing.

Intercept: True

This optimization attribute was set to true to enable algorithm to use intercepts while creating the model which reduced the error. This is also enabled to reduce the prediction skewedness.

Number of Iterations: 100

Increasing the number of iterations beyond 100 had minimal impact on predictions and so 100 was chosen as iteration count.

SciPy libraries vs Apache Spark

On analyzing the metrics observed using SciPy and Apache Spark, it was observed that the classification metrics were almost similar from both the libraries. Some of the SciPi regression metrics were better than that of the metrics observed using Spark's regression algorithms.

In general, it is advisable to use Spark's mlib when dealing with large amounts of data. Spark's RDD's (in memory processing) enables higher response rates when compared to SciPy despite of users being able to connect to other big data platforms using SciPy libraries.

Appendix

Instructions to execute

Python

Use the "Readme.txt" file located in folder "Python_SciPy_Models" to execute the Python code

PySpark

Use the "Readme.txt" file located in folder "Pyspark" to execute the PySpark code

Scala

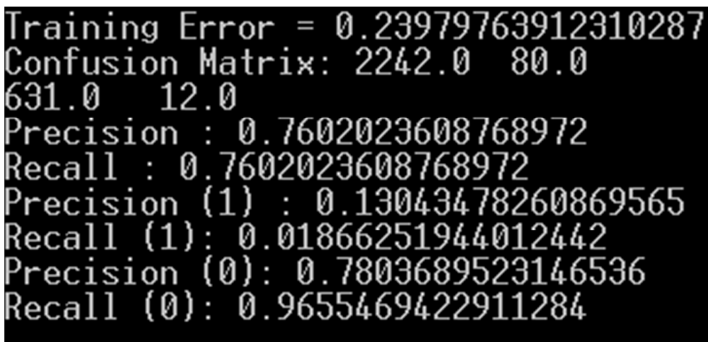
Use the "Readme.txt" file located in folder "Scala" to execute the Scala code

Some of the screenshots for predictions using SPARK

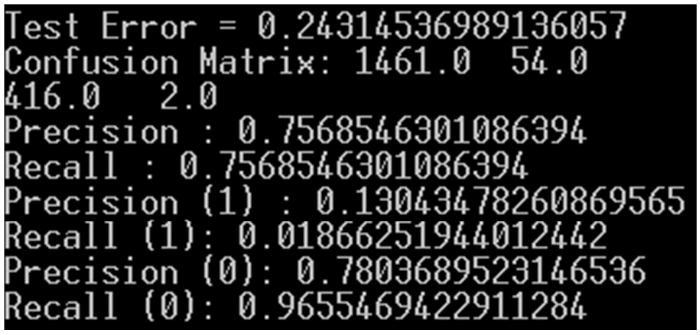
SVM with SGD - Scala

Configuration: L2

Training Metrics

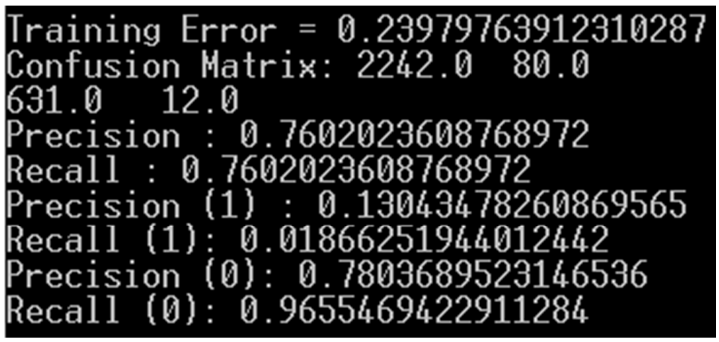


Testing Metrics

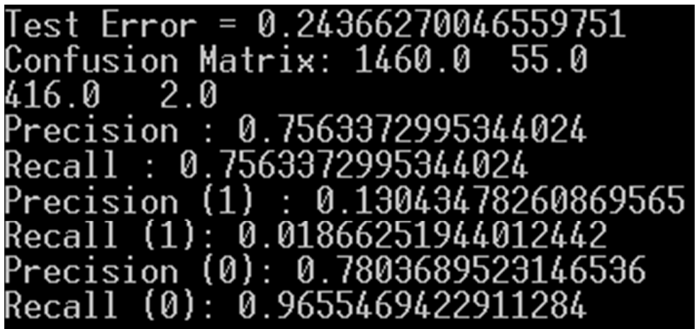


Configuration: L1

Training Metrics



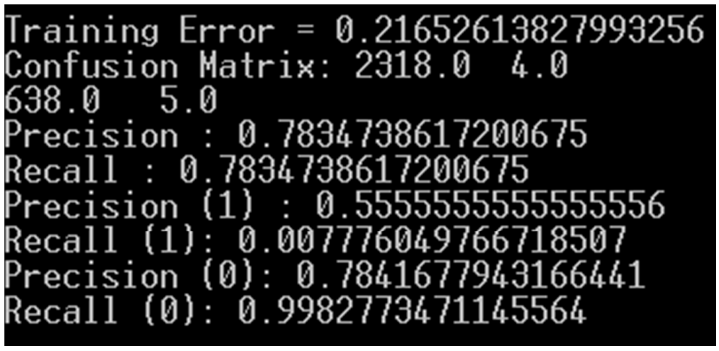
Testing Metrics



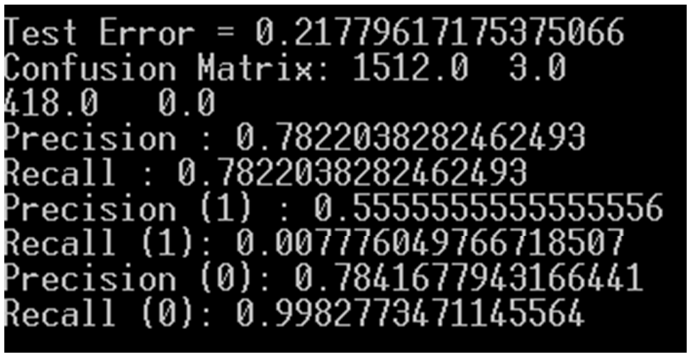
Logistic Regression with LBFGS - Scala

Configuration: L2

Training Metrics

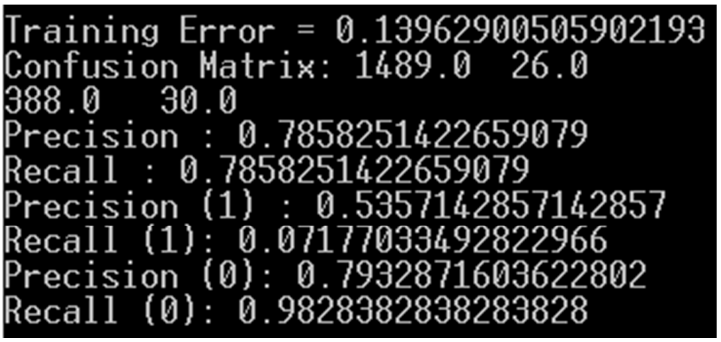


Test Metrics



Configuration: L1

Training Metrics



Test Metrics

```
Test Error = 0.2141748577340921
Confusion Matrix: 1489.0  26.0
388.0   30.0
Precision : 0.7858251422659079
Recall : 0.7858251422659079
Precision (1) : 0.5357142857142857
Recall (1): 0.07177033492822966
Precision (0): 0.7932871603622802
Recall (0): 0.9828382838283828
```

Logistic Regression with SGD - Scala

Configuration: L2

- Intercept: True
- Data Split: 60 ,40

Training Metrics

```
Training Error = 0.2708263069139966
Confusion Matrix: 2097.0  225.0
578.0   65.0
Precision : 0.7291736930860033
Recall : 0.7291736930860033
Precision (1) : 0.22413793103448276
Recall (1): 0.10108864696734059
Precision (0): 0.7839252336448598
Recall (0): 0.9031007751937985
```

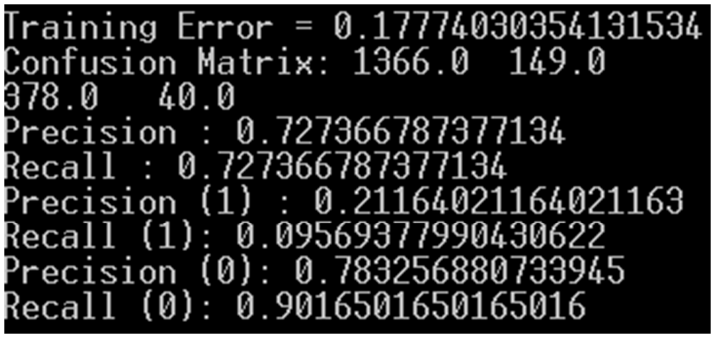
Test Metrics

```
Test Error = 0.272633212622866
Confusion Matrix: 1366.0  149.0
378.0   40.0
Precision : 0.727366787377134
Recall : 0.727366787377134
Precision (1) : 0.22413793103448276
Recall (1): 0.10108864696734059
Precision (0): 0.7839252336448598
Recall (0): 0.9031007751937985
```

Configuration: L1:

- Intercept: True
- Data Split: 60,40

Training Metrics



Test Metrics