Ryan Ballenger
CSCIE63 HW11 Solutions

Problem 1. Remove the header of the attached Samll_Car_Data.csv file and then import it into Spark. Randomly select 10% of you data for testing and use remaining data for training. Look initially at horsepower and displacement. Treat displacement as a feature and horsepower as the target variable. Use MLlib linear regression to identify the model for the relationship. Use test data to illustrate accuracy of your ability to predict the relationship. Create a diagram using D3 which presents the model (straight line), original test data and predictions of your analysis. Please label your axes and use different colors for original data and predicted data.

```
// Anaconda is installed from the downloaded file
[cloudera@localhost Desktop]$ bash Anaconda2-4.0.0-Linux-x86_64.sh
Welcome to Anaconda2 4.0.0 (by Continuum Analytics, Inc.)
In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
…

// The ANACONDA environment variable is created within the bash_profile and added
// to the PATH in order to conveniently access it.
[cloudera@localhost Desktop]$ vim /home/cloudera/.bash_profile

ANACONDA=/home/cloudera/anaconda2
export ANACONDA

PATH=$PATH:$HOME/bin:$JAVA_HOME/bin:$SPARK_HOME/bin:$M2_HOME/bin:$ANACONDA/bin

// The .bash_profile is sourced to load the ANACONDA variable into the PATH.
[cloudera@localhost Desktop]$ source /home/cloudera/.bash_profile

// The following step is skipped from the notes since Anaconda is sufficient for the analysis.
# $ sudo yum -y install python2.7

// The numpy and scipy packages are installed with yum. These libraries are necessary
// to perform the machine learning analysis.
[cloudera@localhost Desktop]$ sudo yum install numpy scipy
Loaded plugins: fastestmirror, refresh-packagekit, security
Setting up Install Process
```

```
Loading mirror speeds from cached hostfile
 * base: centos.firehosted.com
 * epel: mirror.cogentco.com
 * extras: yum.tamu.edu
 * updates: dallas.tx.mirror.xygenhosting.com
Package numpy-1.4.1-9.el6.x86_64 already installed and latest version
Package scipy-0.7.2-8.el6.x86_64 already installed and latest version
Nothing to do

// The ipython installation is confirmed.
[cloudera@localhost Desktop]$ pip install ipython
Requirement already satisfied (use --upgrade to upgrade): ipython in
/home/cloudera/anaconda2/lib/python2.7/site-packages

// The following lines are added to the .bash_profile. They set the libraries to use when
// pyspark is run including pylab which supports MATLAB type interfaces.
[cloudera@localhost Desktop]$ vim /home/cloudera/.bash_profile

IPYTHON=1
export IPYTHON
IPYTHON_OPTS="--pylab"
export IPYTHON_OPTS

// .bash_profile is sourced to load the new settings.
[cloudera@localhost Desktop]$ source /home/cloudera/.bash_profile

// The ipython notebook changes were not made and notebook is not used on the
//  Tornado server

// The header of labels is removed from Small_Car_Data which prevents the machine learning
// processes from treating the header as a row of data.
[cloudera@localhost Desktop]$ sed 1d Small_Car_Data.csv > Small_Car_Data_noheader.csv

// pyspark is invoked to provide the console where the programs will be run.
[cloudera@localhost Desktop]$ pyspark
Python 2.7.11 |Anaconda 4.0.0 (64-bit)| (default, Dec  6 2015, 18:08:32)
Type "copyright", "credits" or "license" for more information.
IPython 4.1.2 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
Using matplotlib backend: Qt4Agg
```

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in
[jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in
[jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
…
```

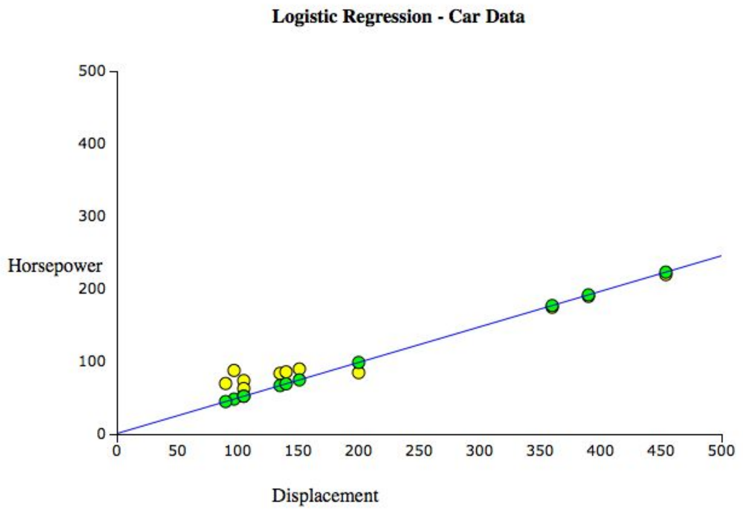
// one.py is run. The program loads the data, creates test and training datasets, creates a linear
// model, makes predictions on the test data, and measures the error. Please view one.py to
// see the entire process and comments on all the steps.

```
In [61]: execfile('/home/cloudera/Desktop/one.py')
...
16/04/22 18:53:13 INFO executor.Executor: Running task 0.0 in stage 1253.0 (TID 1231)
16/04/22 18:53:13 INFO storage.BlockManager: Found block rdd_1368_0 locally
16/04/22 18:53:13 INFO python.PythonRDD: Times: total = 45, boot = -197, init = 241, finish = 1
16/04/22 18:53:13 INFO executor.Executor: Finished task 0.0 in stage 1253.0 (TID 1231). 2545
bytes result sent to driver
16/04/22 18:53:13 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1253.0 (TID
1231) in 48 ms on localhost (1/1)
16/04/22 18:53:13 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1253.0, whose tasks
have all completed, from pool
16/04/22 18:53:13 INFO scheduler.DAGScheduler: ResultStage 1253 (runJob at
PythonRDD.scala:361) finished in 0.048 s
16/04/22 18:53:13 INFO scheduler.DAGScheduler: Job 706 finished: runJob at
PythonRDD.scala:361, took 0.056597 s
```

Linear Model predictions: [(220.0, 223.40250192175964), (190.0, 192.05093752896795), (175.0, 177.35489171984685), (85.0, 98.975980737867616), (88.0, 48.519556793218484)]
Linear Model: (weights=[0.489868193637], intercept=1.002342010393577)
Linear Model - Mean Squared Error: 341.0954
Linear Model - Mean Absolute Error: 15.1460
Linear Model - Root Mean Squared Log Error: 0.2741

// The results of the error analysis show the predictions are effective. An absolute error of 15.1
// confirms that the model is accurately predicting horsepower values that are typically
// in the range of 50 to 200 HP.

// The test data and predictions are saved as CSV files and plotted with D3. The test data is
// in yellow, the predicted values are green, and the model is the blue line.



Problem 2. Treat: cylinders, displacement, manufacturer, model_year, origin and weight as features and use linear regression to predict two target variable: horsepower and acceleration. Please note that some of those are categorical variables. Use test data to assess quality of prediction for both target variables. Which of two target variables is easier to predict, in the sense that predicted values differ less from the original values.

```
// two_acceleration.py is executed. It loads the data, creates test and train datasets, maps the
// categorical features to floats, creates a linear model, makes predictions, and measures error.
// Please see two_acceleration.py for the entire process and comments on each step.
In [68]: execfile('/home/cloudera/Desktop/two_acceleration.py')
...
16/04/22 19:17:37 INFO python.PythonRDD: Times: total = 50, boot = -70, init = 117, finish = 3
16/04/22 19:17:37 INFO executor.Executor: Finished task 0.0 in stage 2432.0 (TID 2414). 2545
bytes result sent to driver
16/04/22 19:17:37 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 2432.0 (TID
2414) in 55 ms on localhost (1/1)
16/04/22 19:17:37 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2432.0, whose tasks
have all completed, from pool
16/04/22 19:17:37 INFO scheduler.DAGScheduler: ResultStage 2432 (runJob at
PythonRDD.scala:361) finished in 0.051 s
16/04/22 19:17:37 INFO scheduler.DAGScheduler: Job 1272 finished: runJob at
PythonRDD.scala:361, took 0.065297 s
Linear Model predictions: [(12.0, 15.534528937812402), (10.0, 13.731885045708598), (15.0,
10.493341973298689), (14.0, 20.44535200688761), (15.5, 12.964952507239234)]
Linear Model - Mean Squared Error: 22.5791
Linear Model - Mean Absolute Error: 4.2820
Linear Model - Root Mean Squared Log Error: 0.3060

// The linear model test data predictions and errors confirm the model is successfully
// predicting the accelerations. The acceleration varies from approximately 8 to 22 and the MAE
// is 4.2. Basically, the model is able to predict whether a car has a low, average, or high
// acceleration. It does not precisely predict the acceleration though considering the range is
// approximately 14 and the MAE is 4.2. This suggests the model does wrongly predict the
// acceleration but does not predict it with great accuracy either.

// two_horsepower.py is run from pyspark. This program is similar to the previous on
// except that the target variable is horsepower instead of acceleration.

In [77]: execfile('/home/cloudera/Desktop/two_horsepower.py')
...
16/04/22 19:26:10 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 2650.0 (TID
2623, localhost, partition 0,PROCESS_LOCAL, 2164 bytes)
16/04/22 19:26:10 INFO executor.Executor: Running task 0.0 in stage 2650.0 (TID 2623)
16/04/22 19:26:10 INFO storage.BlockManager: Found block rdd_2968_0 locally
16/04/22 19:26:10 INFO python.PythonRDD: Times: total = 45, boot = -71, init = 112, finish = 4
16/04/22 19:26:10 INFO executor.Executor: Finished task 0.0 in stage 2650.0 (TID 2623). 2545
bytes result sent to driver
16/04/22 19:26:10 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 2650.0 (TID
2623) in 50 ms on localhost (1/1)
16/04/22 19:26:10 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2650.0, whose tasks
have all completed, from pool
16/04/22 19:26:10 INFO scheduler.DAGScheduler: ResultStage 2650 (runJob at
PythonRDD.scala:361) finished in 0.049 s
16/04/22 19:26:10 INFO scheduler.DAGScheduler: Job 1409 finished: runJob at
PythonRDD.scala:361, took 0.061699 s
Linear Model predictions: [(130.0, 133.68804852741812), (95.0, 90.234732173862596),
(120.0, 145.24672318005159), (84.0, 96.105195493440505), (96.0, 101.43451294604785)]
Linear Model - Mean Squared Error: 245.1263
Linear Model - Mean Absolute Error: 13.2628
Linear Model - Root Mean Squared Log Error: 0.1494

// The first 5 predicted values and the MAE of 13.2 confirms that the model is successfully
// predicting horsepower values. The horsepower values range from approximately 40 to 220
// meaning that predictions with a MAE of 13.2 are accurate. In general, the model is able
// to predict whether the horsepower is low, medium, or high in terms of the other cars and is
// able to precisely identify the horsepower within 13 hp on average.
```

Which of two target variables is easier to predict, in the sense that predicted values differ less from the original values.

// The MAE and other error measurements are smaller for the acceleration predictions.
// However, the range of values for the horsepower is much larger (~40 to 220) than the range of
// accelerations (8 to 22). For that reason, I believe the model is better at predicting the
// horsepower. For a range of approximately 180, predictions with a MAE of 13 are very
// accurate. Also the RMSLE is smaller for the horsepower since this measurement takes into
// account the range by converting to a log scale.

Problem 3. Repeat above analysis with decision tree method. Compare predicting ability/quality of this technique with that of the linear regression.

// three_acceleration.py is run. The decision tree model is created with the training data
// and used to predict values on the test data.

```
In [136]: execfile('/home/cloudera/Desktop/three_acceleration.py')
...
16/04/22 19:59:08 INFO storage.BlockManager: Found block rdd_3180_0 locally
16/04/22 19:59:08 INFO python.PythonRDD: Times: total = 56, boot = 18, init = 33, finish = 5
16/04/22 19:59:08 INFO python.PythonRDD: Times: total = 133, boot = 101, init = 1, finish = 31
16/04/22 19:59:08 INFO python.PythonRDD: Times: total = 63, boot = 7, init = 56, finish = 0
16/04/22 19:59:08 INFO python.PythonRDD: Times: total = 208, boot = 175, init = 33, finish = 0
16/04/22 19:59:08 INFO executor.Executor: Finished task 0.0 in stage 2795.0 (TID 2788). 2263 bytes result sent to driver
16/04/22 19:59:08 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 2795.0 (TID 2788) in 212 ms on localhost (1/1)
16/04/22 19:59:08 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2795.0, whose tasks have all completed, from pool
16/04/22 19:59:08 INFO scheduler.DAGScheduler: ResultStage 2795 (runJob at PythonRDD.scala:361) finished in 0.211 s
16/04/22 19:59:08 INFO scheduler.DAGScheduler: Job 1486 finished: runJob at PythonRDD.scala:361, took 0.238597 s
Decision Tree predictions: [(11.0, 12.0), (8.5, 10.5), (8.0, 11.5), (8.0, 10.5), (14.0, 11.0)]
Decision Tree - Mean Squared Error: 5.2509
Decision Tree - Mean Absolute Error: 1.7619
Decision Tree - Root Mean Squared Log Error: 0.1781
```

// The first 5 predictions and the small values for the other errors show the decision tree model is
// able to predict acceleration well.

// three_horsepower.py is executed. The decision tree model is created with the horsepower
// train data and then, the model predicts the test data horsepower values. Error measurements
// are given the output.

```
16/04/22 20:02:05 INFO storage.BlockManager: Found block rdd_3239_0 locally
```

```
16/04/22 20:02:05 INFO python.PythonRDD: Times: total = 63, boot = 38, init = 15, finish = 10
16/04/22 20:02:05 INFO python.PythonRDD: Times: total = 122, boot = 98, init = 1, finish = 23
16/04/22 20:02:05 INFO python.PythonRDD: Times: total = 71, boot = 37, init = 33, finish = 1
16/04/22 20:02:05 INFO python.PythonRDD: Times: total = 206, boot = 191, init = 15, finish = 0
16/04/22 20:02:05 INFO executor.Executor: Finished task 0.0 in stage 2832.0 (TID 2830). 2263 bytes result sent to driver
16/04/22 20:02:05 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 2832.0 (TID 2830) in 210 ms on localhost (1/1)
16/04/22 20:02:05 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2832.0, whose tasks have all completed, from pool
16/04/22 20:02:05 INFO scheduler.DAGScheduler: ResultStage 2832 (runJob at PythonRDD.scala:361) finished in 0.209 s
16/04/22 20:02:05 INFO scheduler.DAGScheduler: Job 1506 finished: runJob at PythonRDD.scala:361, took 0.228058 s
```

Decision Tree predictions: [(140.0, 140.0), (165.0, 151.25), (175.0, 202.5), (88.0, 75.0), (79.0, 85.0)]
Decision Tree - Mean Squared Error: 210.3306
Decision Tree - Mean Absolute Error: 11.3306
Decision Tree - Root Mean Squared Log Error: 0.1419

// The first 5 predicted values and the small error measurements confirm the decision tree is
// effectively predicting the horsepower values in the test data.

// As with the linear regression, the error measurements are smaller for the acceleration.
// However, the acceleration has a much smaller range. The decision tree is able to very
// acurately predict both target variables. The acceleration is predicted within 1.7 in a range of
// approximately 14. The horsepower is predicted within 11.3 on average in a range of
// approximately 180. For that reason, I argue the horsepower is more effectively and accurately
// predicted. The feature variables are enabling this model to very accurately predict horsepower
// within a wide range. In terms of MAE and MSE, acceleration predictions are closer to the
// actual values though.

// The decision tree model is more accurate than the linear regression model. The decision tree
// is able to recognize and use patterns in the data to more precisely predict the acceleration and
// horsepower. The decision tree model is able to learn more effectively from the training data in
// order to predict the test data. The method of creating a tree to follow to a predicted value is
// more effectively with this dataset than using a linear regression equation to compute the
// predicted values.
// Specifically, the decision tree is better which is illustrated by comparing the errors. For
// acceleration the decision tree has a MAE of 1.8 while the linear regression model has a MAE
// of 4.2. The decision tree is much more accurate. For the horsepower, the decision tree has an
// MAE of 11.3 while the linear regression model has an MAE of 13.2. Although the models are

// closer in predicting horsepowers than acceleration, the decision tree again does better.