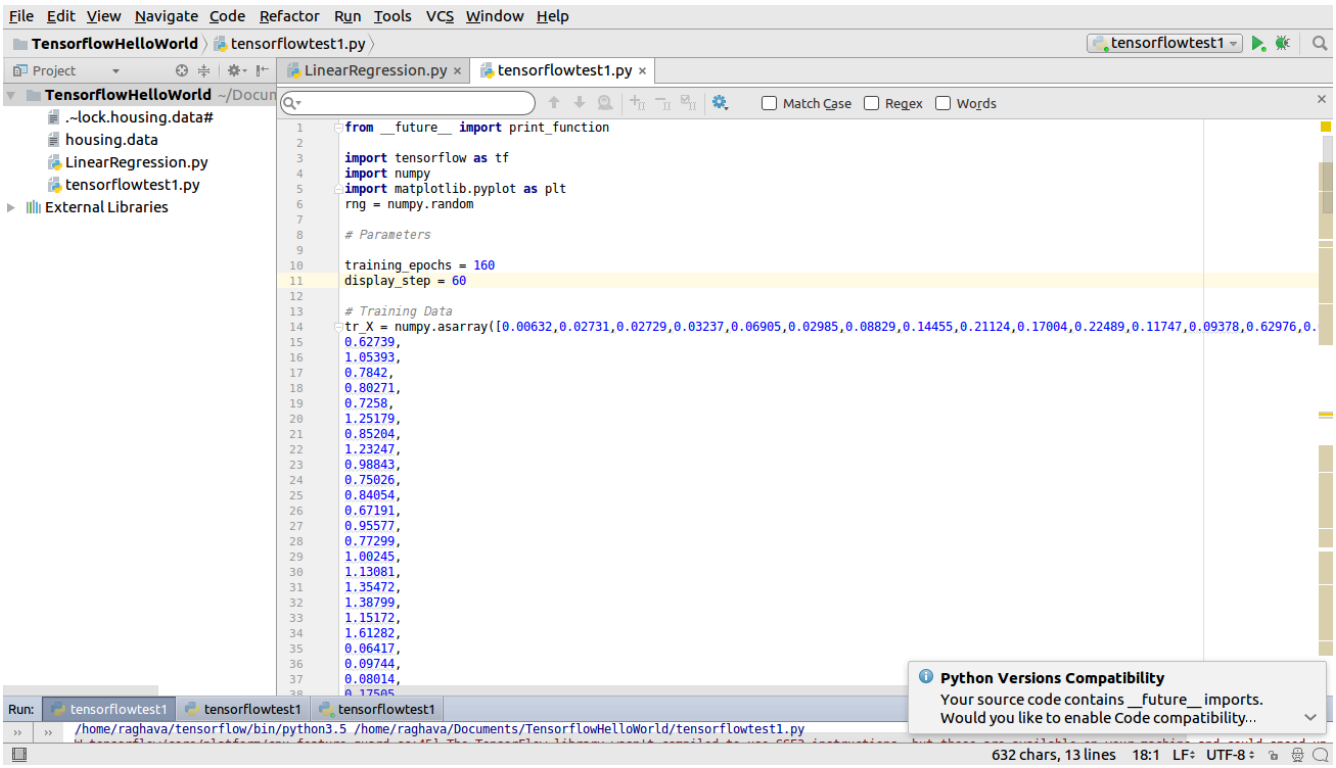
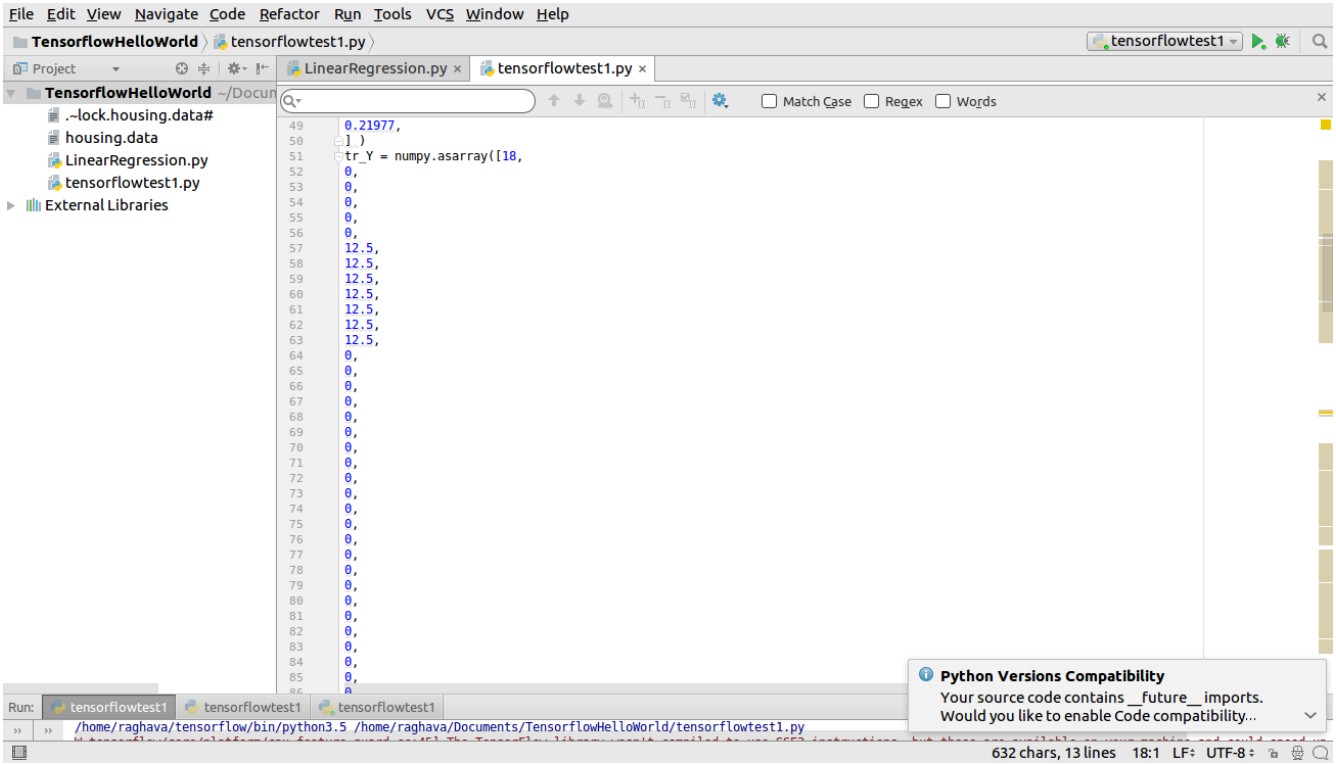
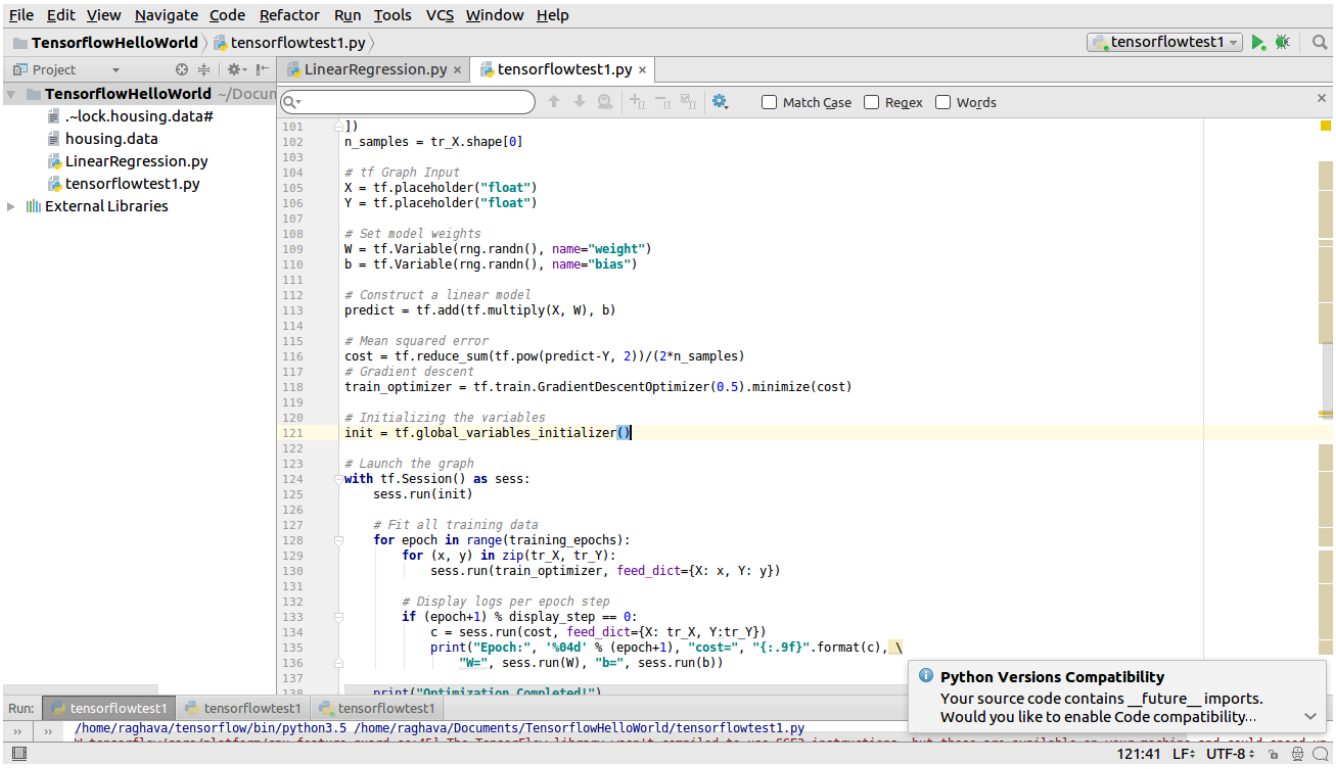
CS5542 Big Data Analytics and App
LAB ASSIGNMENT #7

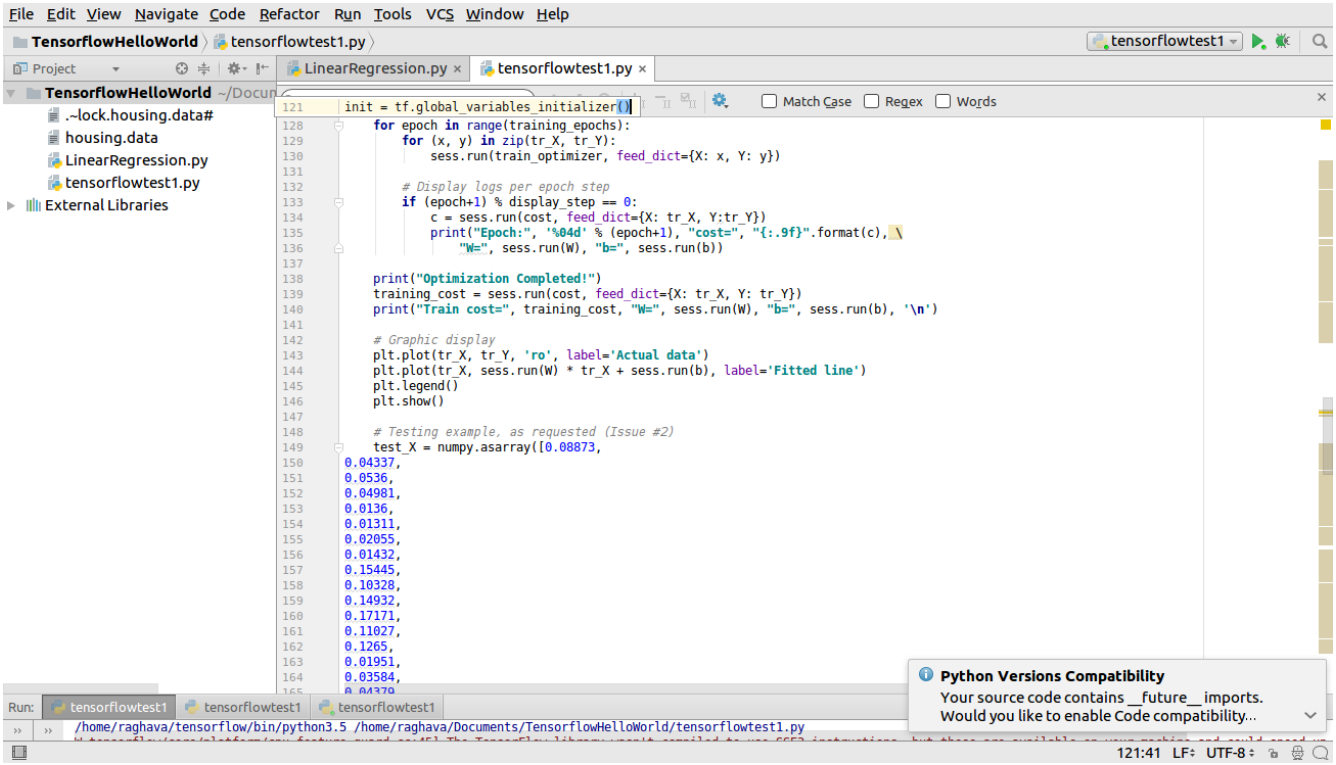
Linear Regression using Tensor flow:
[1]In statistics, **linear regression** is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple **linear regression**.

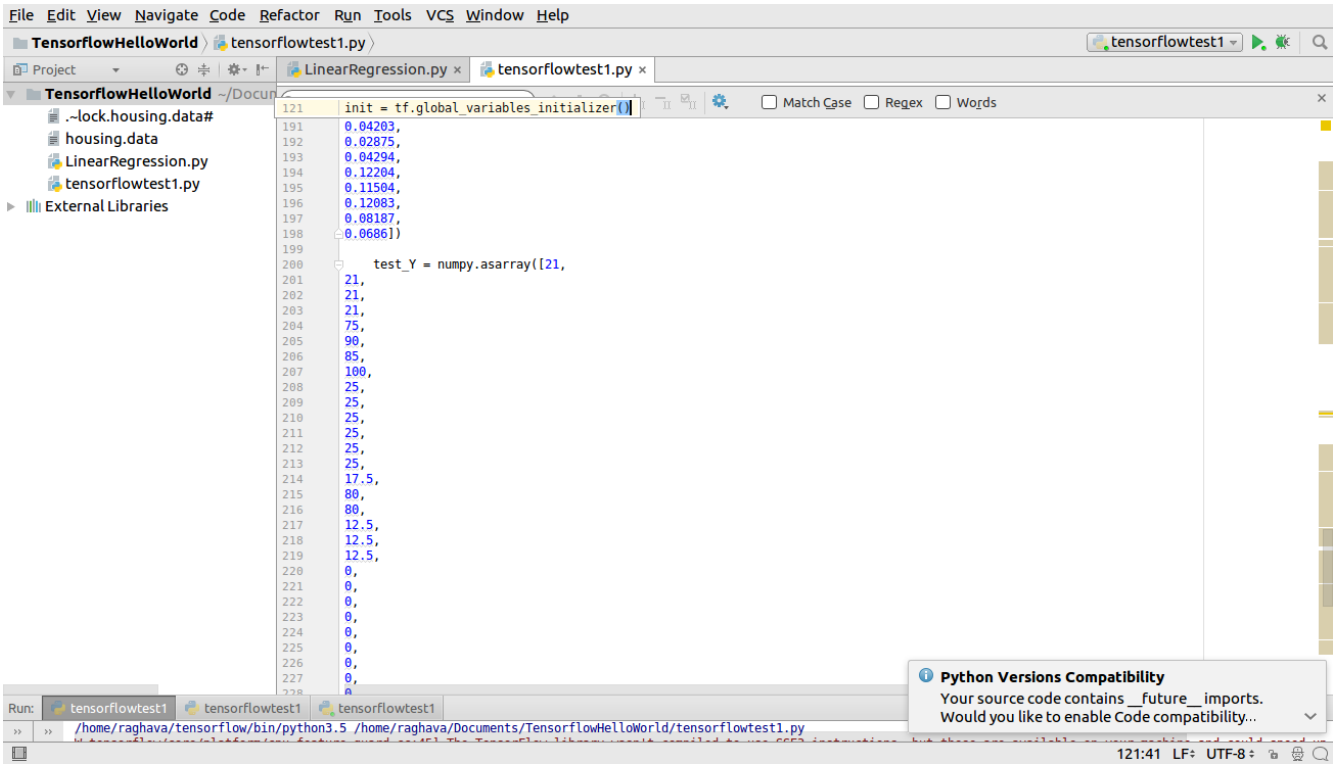
Implementing Linear Regression through Tensor Flow:
Housing data set from the UCI machine Learning Repository is used. For this lab assignment only **50** rows from the data set are used for **training** and another **50** rows are used for the **test data**. Here are the screen shots:

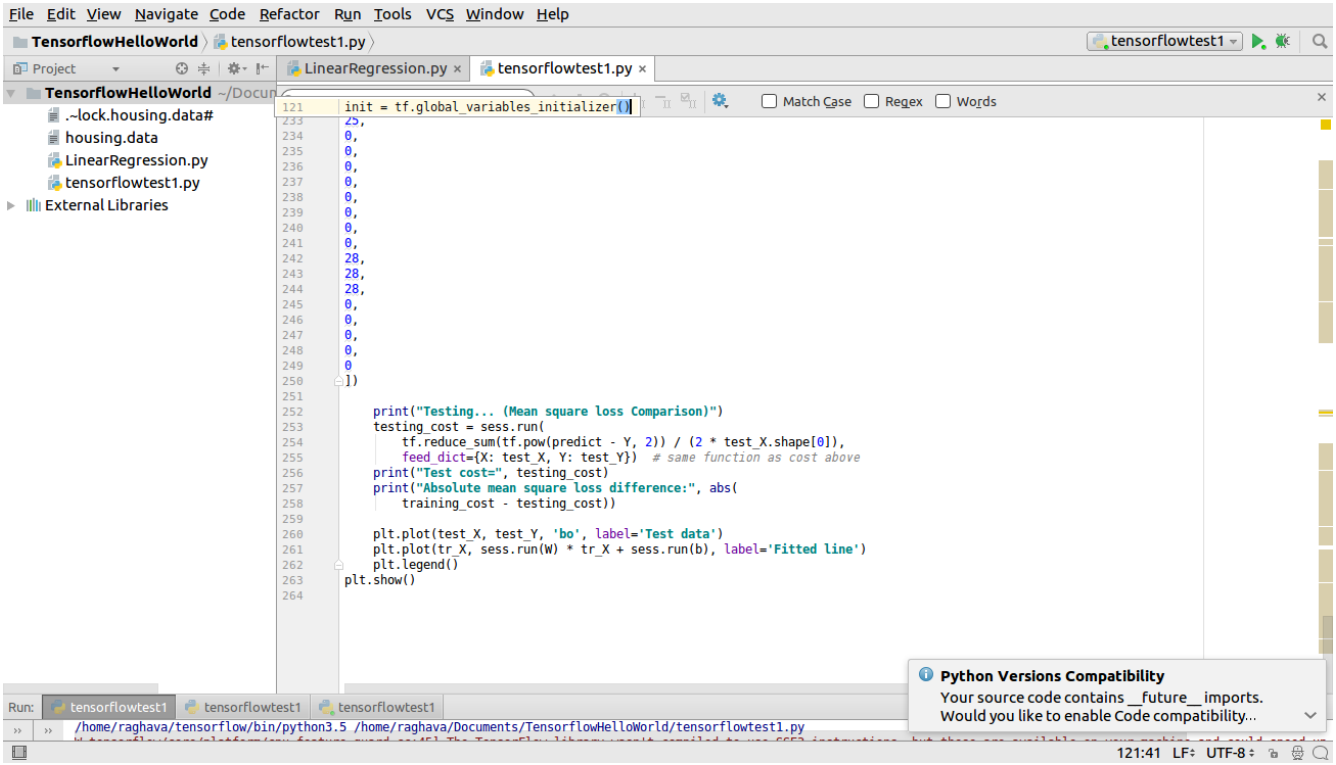



File Edit View Navigate Code Refactor Run Tools VCS Window Help
TensorflowHelloWorld 〉 tensorflowtest1.py 〉
tensorflowtest1
Project
LinearRegression.py ×
tensorflowtest1.py ×
Match Case
Regex
Words
TensorflowHelloWorld ~/Docun
.~lock.housing.data#
housing.data
LinearRegression.py
tensorflowtest1.py
External Libraries
49 0.21977,
50])
51 tr_Y = numpy.asarray([18,
52 0,
53 0,
54 0,
55 0,
56 0,
57 12.5,
58 12.5,
59 12.5,
60 12.5,
61 12.5,
62 12.5,
63 12.5,
64 0,
65 0,
66 0,
67 0,
68 0,
69 0,
70 0,
71 0,
72 0,
73 0,
74 0,
75 0,
76 0,
77 0,
78 0,
79 0,
80 0,
81 0,
82 0,
83 0,
84 0,
85 0,
Python Versions Compatibility
Your source code contains __future__ imports.
Would you like to enable Code compatibility...
Run: tensorflowtest1 tensorflowtest1 tensorflowtest1
/home/raghava/tensorflow/bin/python3.5 /home/raghava/Documents/TensorflowHelloWorld/tensorflowtest1.py
632 chars, 13 lines 18:1 LF: UTF-8:

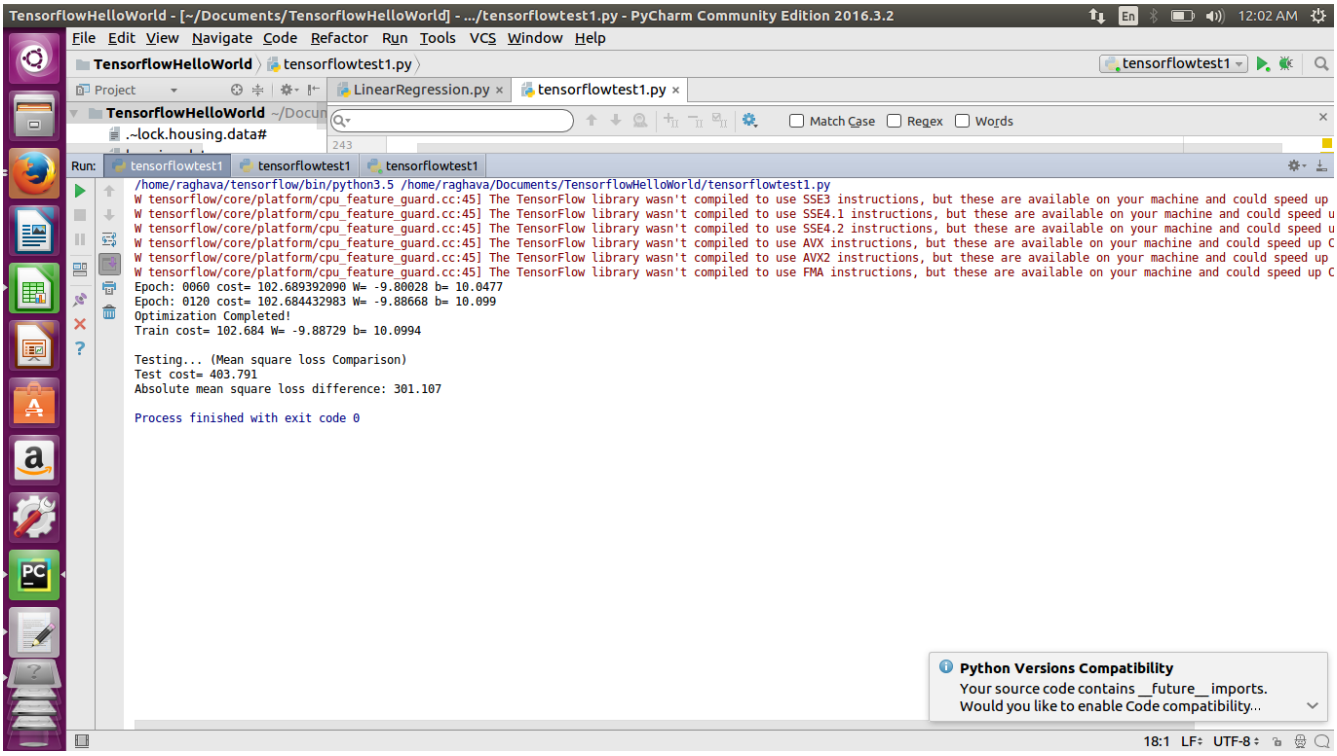

File Edit View Navigate Code Refactor Run Tools VCS Window Help
TensorflowHelloWorld 〉 tensorflowtest1.py 〉
tensorflowtest1
Project
LinearRegression.py ×
tensorflowtest1.py ×
Match Case
Regex
Words
TensorflowHelloWorld ~/Docun
.~lock.housing.data#
housing.data
LinearRegression.py
tensorflowtest1.py
External Libraries
101])
102 n_samples = tr_X.shape[0]
103
104 # tf Graph Input
105 X = tf.placeholder("float")
106 Y = tf.placeholder("float")
107
108 # Set model weights
109 W = tf.Variable(rng.randn(), name="weight")
110 b = tf.Variable(rng.randn(), name="bias")
111
112 # Construct a linear model
113 predict = tf.add(tf.multiply(X, W), b)
114
115 # Mean squared error
116 cost = tf.reduce_sum(tf.pow(predict-Y, 2))/(2*n_samples)
117 # Gradient descent
118 train_optimizer = tf.train.GradientDescentOptimizer(0.5).minimize(cost)
119
120 # Initializing the variables
121 init = tf.global_variables_initializer()
122
123 # Launch the graph
124 with tf.Session() as sess:
125 sess.run(init)
126
127 # Fit all training data
128 for epoch in range(training_epochs):
129 for (x, y) in zip(tr_X, tr_Y):
130 sess.run(train_optimizer, feed_dict={X: x, Y: y})
131
132 # Display logs per epoch step
133 if (epoch+1) % display_step == 0:
134 c = sess.run(cost, feed_dict={X: tr_X, Y:tr_Y})
135 print("Epoch:", '%04d' % (epoch+1), "cost=", "{:.9f}".format(c), \
136 "W=", sess.run(W), "b=", sess.run(b))
137
Python Versions Compatibility
Your source code contains __future__ imports.
Would you like to enable Code compatibility...
Run: tensorflowtest1 tensorflowtest1 tensorflowtest1
/home/raghava/tensorflow/bin/python3.5 /home/raghava/Documents/TensorflowHelloWorld/tensorflowtest1.py
121:41 LF: UTF-8:

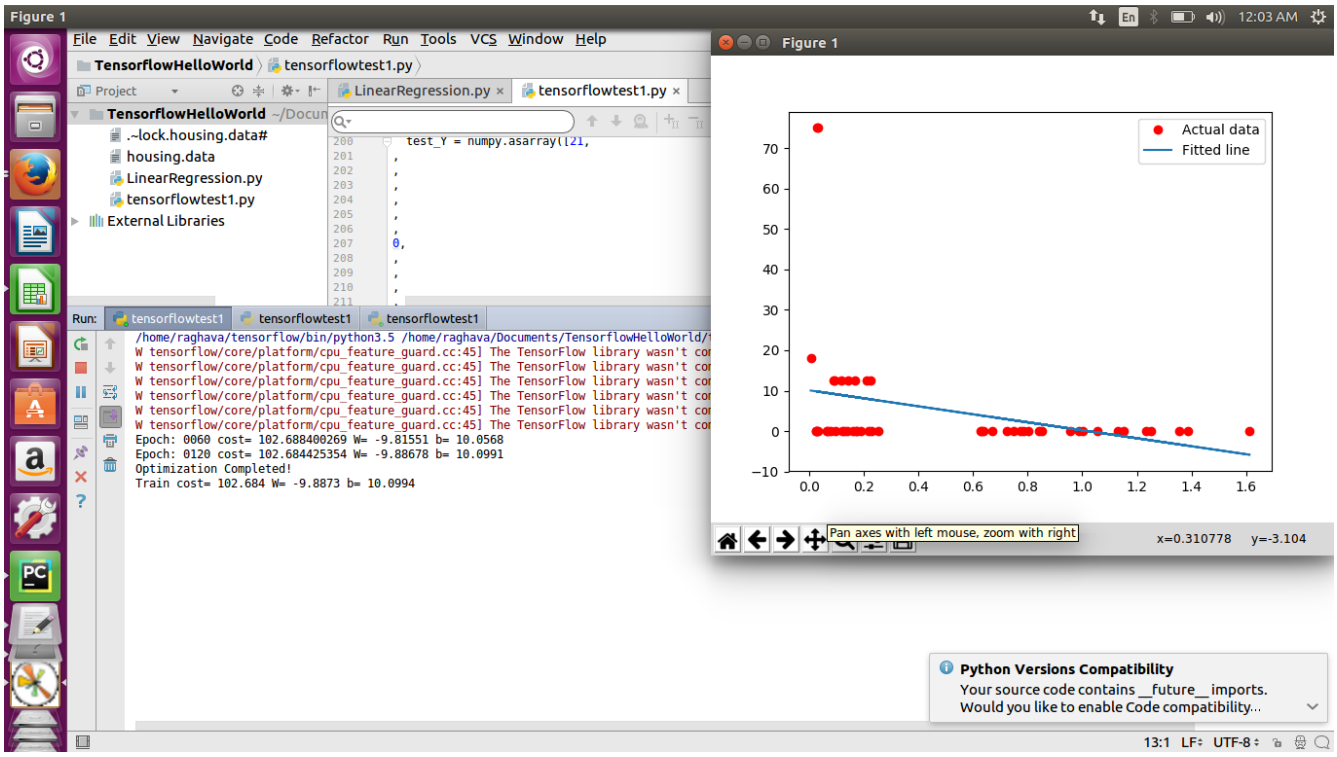

File Edit View Navigate Code Refactor Run Tools VCS Window Help
TensorflowHelloWorld 〉 tensorflowtest1.py 〉
tensorflowtest1
Project
LinearRegression.py ×
tensorflowtest1.py ×
TensorflowHelloWorld ~/Docum
.~lock.housing.data#
housing.data
LinearRegression.py
tensorflowtest1.py
External Libraries
Match Case
Regex
Words
121 init = tf.global_variables_initializer()
128 for epoch in range(training_epochs):
129 for (x, y) in zip(tr_X, tr_Y):
130 sess.run(train_optimizer, feed_dict={X: x, Y: y})
131
132 # Display logs per epoch step
133 if (epoch+1) % display_step == 0:
134 c = sess.run(cost, feed_dict={X: tr_X, Y:tr_Y})
135 print("Epoch:", '%04d' % (epoch+1), "cost=", "{:.9f}".format(c), \
136 "W=", sess.run(W), "b=", sess.run(b))
137
138 print("Optimization Completed!")
139 training_cost = sess.run(cost, feed_dict={X: tr_X, Y: tr_Y})
140 print("Train cost=", training_cost, "W=", sess.run(W), "b=", sess.run(b), '\n')
141
142 # Graphic display
143 plt.plot(tr_X, tr_Y, 'ro', label='Actual data')
144 plt.plot(tr_X, sess.run(W) * tr_X + sess.run(b), label='Fitted line')
145 plt.legend()
146 plt.show()
147
148 # Testing example, as requested (Issue #2)
149 test_X = numpy.asarray([0.08873,
150 0.04337,
151 0.0536,
152 0.04981,
153 0.0136,
154 0.01311,
155 0.02055,
156 0.01432,
157 0.15445,
158 0.10328,
159 0.14932,
160 0.17171,
161 0.11027,
162 0.1265,
163 0.01951,
164 0.03584,
Python Versions Compatibility
Your source code contains __future__ imports.
Would you like to enable Code compatibility...
Run: tensorflowtest1 tensorflowtest1 tensorflowtest1
/home/raghava/tensorflow/bin/python3.5 /home/raghava/Documents/TensorflowHelloWorld/tensorflowtest1.py
121:41 LF: UTF-8

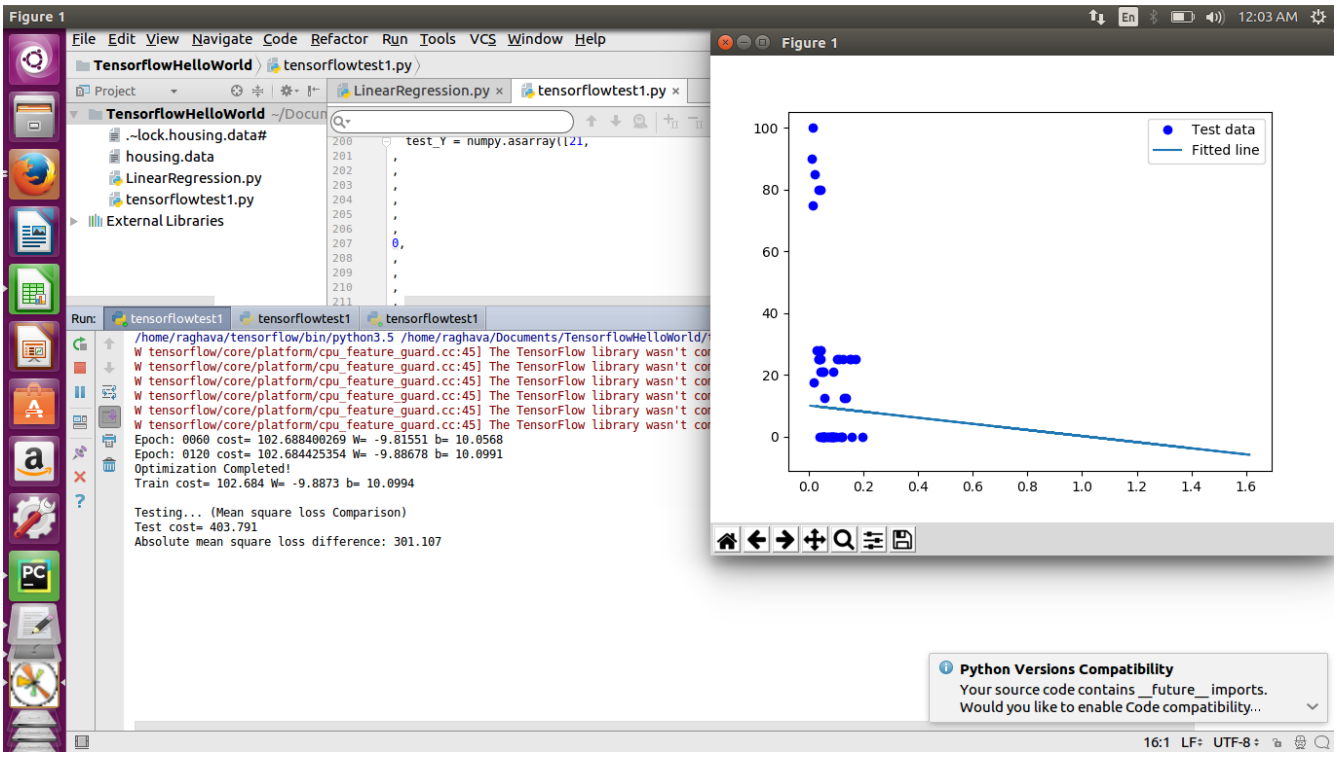

File Edit View Navigate Code Refactor Run Tools VCS Window Help
TensorflowHelloWorld 〉 tensorflowtest1.py 〉
tensorflowtest1
Project
LinearRegression.py ×
tensorflowtest1.py ×
TensorflowHelloWorld ~/Docum
.~lock.housing.data#
housing.data
LinearRegression.py
tensorflowtest1.py
External Libraries
Match Case
Regex
Words
121 init = tf.global_variables_initializer()
191 0.04203,
192 0.02875,
193 0.04294,
194 0.12204,
195 0.11504,
196 0.12083,
197 0.08187,
198 0.0686])
199
200 test_Y = numpy.asarray([21,
201 21,
202 21,
203 21,
204 75,
205 90,
206 85,
207 100,
208 25,
209 25,
210 25,
211 25,
212 25,
213 25,
214 17.5,
215 80,
216 80,
217 12.5,
218 12.5,
219 12.5,
220 0,
221 0,
222 0,
223 0,
224 0,
225 0,
226 0,
227 0,
Python Versions Compatibility
Your source code contains __future__ imports.
Would you like to enable Code compatibility...
Run: tensorflowtest1 tensorflowtest1 tensorflowtest1
/home/raghava/tensorflow/bin/python3.5 /home/raghava/Documents/TensorflowHelloWorld/tensorflowtest1.py
121:41 LF: UTF-8



Here are the results obtained from running the logistic regression program on the data set:





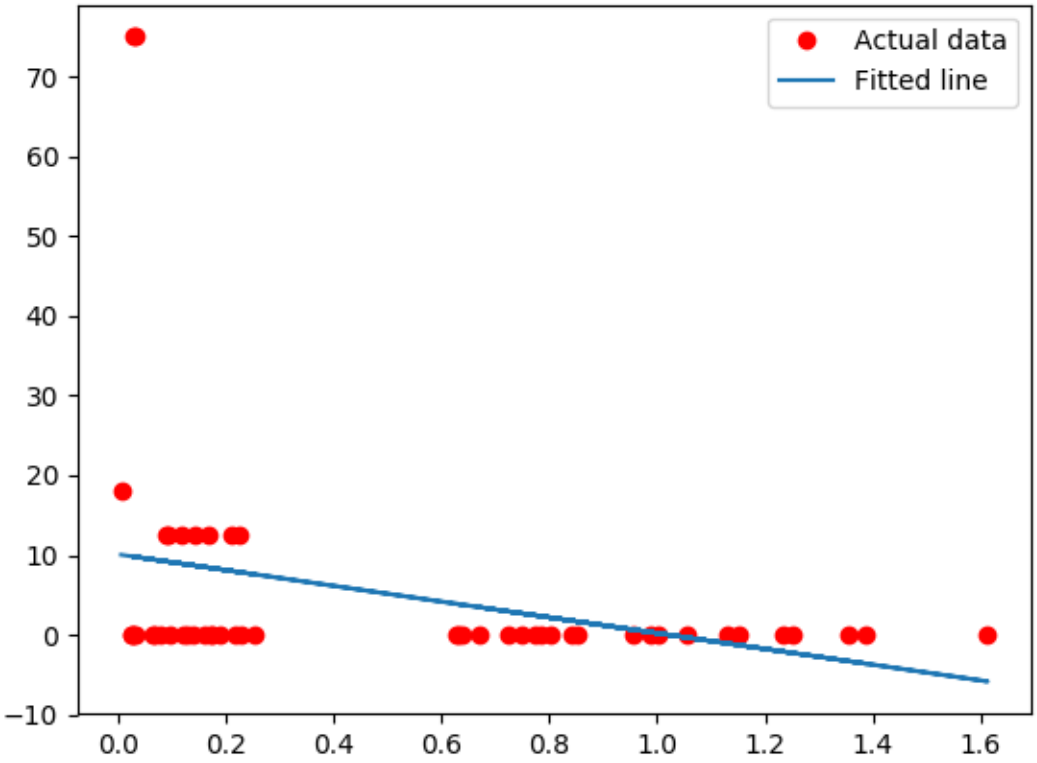


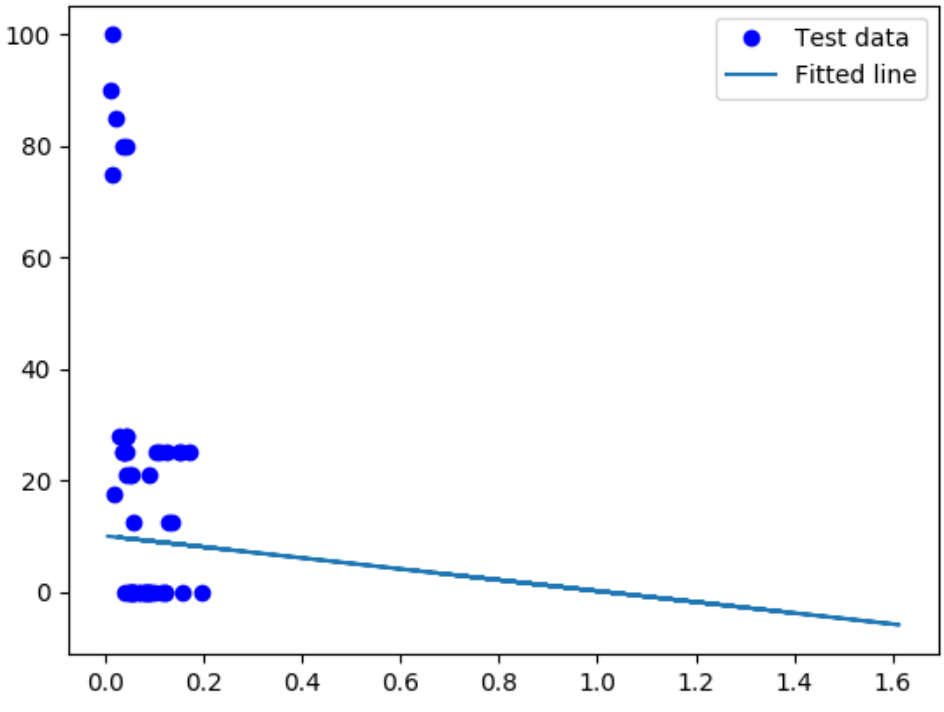
The training cost came out as **102.684**
The test cost came out as **403.791**
The Absolute mean square loss difference: **301.107**

The plots for the logistic model are:

Train:

Tes
t:





References:
[1] https://en.wikipedia.org/wiki/Logistic_regression

for an idea of how to plot the data,little reference is taken from the internet:
https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/2_BasicModels/linear_regression.py

https://blog.altoros.com/visualizing-tensorflow-graphs-with-tensorboard.html