In [19]:
```
pip install sklearn
```

```
Requirement already satisfied: sklearn in /usr/local/lib/python3.6/dist-packages
(0.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.6/dist-pac
kages (from sklearn) (0.21.3)
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.6/dist-pa
ckages (from scikit-learn->sklearn) (1.16.4)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.6/dist-pac
kages (from scikit-learn->sklearn) (0.13.2)
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.6/dist-pa
ckages (from scikit-learn->sklearn) (1.3.0)
WARNING: You are using pip version 19.1.1, however version 19.2.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

In [20]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-packages
(0.25.0)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-pa
ckages (from pandas) (1.16.4)
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.
6/dist-packages (from pandas) (2.8.0)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-pac
kages (from pandas) (2019.2)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from
python-dateutil>=2.6.1->pandas) (1.11.0)
WARNING: You are using pip version 19.1.1, however version 19.2.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

In [21]:
```
pip install keras
```

```
Requirement already satisfied: keras in /usr/local/lib/python3.6/dist-packages
(2.2.4)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.6/dist-pac
kages (from keras) (1.16.4)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/pyth
on3.6/dist-packages (from keras) (1.1.0)
Requirement already satisfied: six>=1.9.0 in /usr/lib/python3/dist-packages (fro
m keras) (1.11.0)
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.6/dist-pack
ages (from keras) (1.3.0)
Requirement already satisfied: keras-applications>=1.0.6 in /usr/local/lib/pytho
n3.6/dist-packages (from keras) (1.0.8)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.6/dist-packages
(from keras) (5.1.2)
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (f
rom keras) (2.9.0)
WARNING: You are using pip version 19.1.1, however version 19.2.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

In [22]:
```
pip install chart-studio
```

```
Requirement already satisfied: chart-studio in /usr/local/lib/python3.6/dist-pac
kages (1.0.0)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-package
s (from chart-studio) (2.22.0)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from chart
-studio) (1.11.0)
Requirement already satisfied: plotly in /usr/local/lib/python3.6/dist-packages
(from chart-studio) (4.0.0)
Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.6/dist-
packages (from chart-studio) (1.3.3)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/l
ocal/lib/python3.6/dist-packages (from requests->chart-studio) (1.25.3)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6
/dist-packages (from requests->chart-studio) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in /usr/lib/python3/dist-packages
(from requests->chart-studio) (2.6)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/di
st-packages (from requests->chart-studio) (2019.6.16)
WARNING: You are using pip version 19.1.1, however version 19.2.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

In [23]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in /usr/local/lib/python3.6/dist-packages
(4.0.0)
Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.6/dist-
packages (from plotly) (1.3.3)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from plotl
y) (1.11.0)
WARNING: You are using pip version 19.1.1, however version 19.2.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

In [24]:
```
from keras.layers.core import Dense, Dropout
from keras.layers.recurrent import LSTM
from keras.models import Sequential
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
#from sklearn.linear_model import LinearRegression
import sklearn
import tensorflow as tf
import pandas
import numpy
import keras
```

In [25]:
```
#Step 1a Load Data //CSV needs to be in coloumbs
csv_path = "/tf/GOLDDaily.csv"
dataOPEN = pandas.read_csv(csv_path, usecols=[1])
dataHIGH = pandas.read_csv(csv_path, usecols=[2])
dataLOW = pandas.read_csv(csv_path, usecols=[3])
dataCLOSE = pandas.read_csv(csv_path, usecols=[4])
```

In [26]:
```
#Step 1b Scale data betwen 0 and 1
scaler = MinMaxScaler(feature_range = (0, 1))
openScaled = scaler.fit_transform(dataOPEN)
highScaled = scaler.fit_transform(dataHIGH)
lowScaled = scaler.fit_transform(dataLOW)
closeScaled = scaler.fit_transform(dataCLOSE)
```

In [27]:

```
#SAVE!
numpy.savetxt('openScaled.csv', openScaled, delimiter=',')
numpy.savetxt('highScaled.csv', highScaled, delimiter=',')
numpy.savetxt('lowScaled.csv', lowScaled, delimiter=',')
numpy.savetxt('close_scaled.csv', closeScaled, delimiter=',')
```

In [28]:

```
#VARIABLES
#allValues for total set of values
#splitUnit for values per split
splitUnit = 26
allValues = 1999
```

In [29]:

```
#Step 1c load data into appropriate values, convert to numpy and reshape
features_set_open = []
labels_open = []
for i in range(splitUnit, allValues):
    features_set_open.append(openScaled[i-splitUnit:i, 0])
    labels_open.append(openScaled[i, 0])
features_set_open, labels_open = numpy.array(features_set_open), numpy.array(labels_open)
features_set_open = numpy.reshape(features_set_open, (features_set_open.shape[0], features_set_open.shape[1], 1))

features_set_high = []
labels_high = []
for i in range(splitUnit, allValues):
    features_set_high.append(highScaled[i-splitUnit:i, 0])
    labels_high.append(highScaled[i, 0])
features_set_high, labels_high = numpy.array(features_set_high), numpy.array(labels_high)
features_set_high = numpy.reshape(features_set_high, (features_set_high.shape[0], features_set_high.shape[1], 1))

features_set_low = []
labels_low = []
for i in range(splitUnit, allValues):
    features_set_low.append(lowScaled[i-splitUnit:i, 0])
    labels_low.append(lowScaled[i, 0])
features_set_low, labels_low = numpy.array(features_set_low), numpy.array(labels_low)
features_set_low = numpy.reshape(features_set_low, (features_set_low.shape[0], features_set_low.shape[1], 1))

features_set_close = []
labels_close = []
for i in range(splitUnit, allValues):
    features_set_close.append(closeScaled[i-splitUnit:i, 0])
    labels_close.append(closeScaled[i, 0])
features_set_close, labels_close = numpy.array(features_set_close), numpy.array(labels_close)
features_set_close = numpy.reshape(features_set_close, (features_set_close.shape[0], features_set_close.shape[1], 1))
```

In [30]:

```
#Step 1d spliting the dataset into training and test set
open_x_train, open_x_test, open_y_train, open_y_test = train_test_split(features_set_open, labels_open, test_size=0.05)
high_x_train, high_x_test, high_y_train, high_y_test = train_test_split(features_set_high, labels_high, test_size=0.05)
low_x_train, low_x_test, low_y_train, low_y_test = train_test_split(features_set_low, labels_low, test_size=0.05)
close_x_train, close_x_test, close_y_train, close_y_test = train_test_split(features_set_close, labels_close, test_size=0.05)
```

In [31]:

```
#VARIABLES
nodes =10
epochUnit = 100
dropout = 0.2
```

In [32]:

```
#Step 2a Build Model For Open
modelOpen = Sequential()
modelOpen.add(LSTM(nodes, return_sequences=True, input_shape=(features_set_open.shape[1],1)))
modelOpen.add(Dropout(dropout))

modelOpen.add(LSTM(nodes, return_sequences=True))
modelOpen.add(Dropout(dropout))

modelOpen.add(LSTM(nodes))

modelOpen.add(Dense(1))

modelOpen.compile(loss='mse', optimizer='adam')
modelOpen.fit(open_x_train, open_y_train, batch_size=26, epochs=epochUnit, validation_split=0.1)
```

```
Train on 1686 samples, validate on 188 samples
Epoch 1/100
1686/1686 [==============================] - 3s 2ms/step - loss: 0.0532 - val_lo
ss: 0.0043
Epoch 2/100
1686/1686 [==============================] - 2s 965us/step - loss: 0.0040 - val_
loss: 0.0033
Epoch 3/100
1686/1686 [==============================] - 2s 962us/step - loss: 0.0036 - val_
loss: 0.0028
Epoch 4/100
1686/1686 [==============================] - 2s 969us/step - loss: 0.0034 - val_
loss: 0.0035
Epoch 5/100
1686/1686 [==============================] - 2s 963us/step - loss: 0.0034 - val_
loss: 0.0026
Epoch 6/100
1686/1686 [==============================] - 2s 978us/step - loss: 0.0031 - val_
loss: 0.0025
Epoch 7/100
1686/1686 [==============================] - 2s 970us/step - loss: 0.0032 - val_
loss: 0.0025
Epoch 8/100
1686/1686 [==============================] - 2s 986us/step - loss: 0.0030 - val_
loss: 0.0024
Epoch 9/100
1686/1686 [==============================] - 2s 965us/step - loss: 0.0029 - val_
loss: 0.0031
Epoch 10/100
1686/1686 [==============================] - 2s 990us/step - loss: 0.0027 - val_
loss: 0.0029
Epoch 11/100
1686/1686 [==============================] - 2s 982us/step - loss: 0.0027 - val_
loss: 0.0029
Epoch 12/100
1686/1686 [==============================] - 2s 967us/step - loss: 0.0025 - val_
loss: 0.0021
Epoch 13/100
1686/1686 [==============================] - 2s 982us/step - loss: 0.0025 - val_
loss: 0.0023
Epoch 14/100
1686/1686 [==============================] - 2s 975us/step - loss: 0.0024 - val_
loss: 0.0021
Epoch 15/100
1686/1686 [==============================] - 2s 990us/step - loss: 0.0024 - val_
loss: 0.0035
Epoch 16/100
1686/1686 [==============================] - 2s 961us/step - loss: 0.0025 - val_
loss: 0.0020
Epoch 17/100
1686/1686 [==============================] - 2s 983us/step - loss: 0.0022 - val_
loss: 0.0020
Epoch 18/100
1686/1686 [==============================] - 2s 970us/step - loss: 0.0022 - val_
loss: 0.0017
Epoch 19/100
1686/1686 [==============================] - 2s 973us/step - loss: 0.0020 - val_
loss: 0.0017
Epoch 20/100
1686/1686 [==============================] - 2s 962us/step - loss: 0.0020 - val_
loss: 0.0016
Epoch 21/100
1686/1686 [==============================] - 2s 960us/step - loss: 0.0020 - val_
loss: 0.0016
```

```
Out[32]: <keras.callbacks.History at 0x7f8e5d32add8>
```

In [33]:

```
#Step 2b Build Model For High
modelHigh = Sequential()
modelHigh.add(LSTM(nodes, return_sequences=True, input_shape=(features_set_high.shape[1], 1)))
modelHigh.add(Dropout(dropout))

modelHigh.add(LSTM(nodes, return_sequences=True))
modelHigh.add(Dropout(dropout))

modelHigh.add(LSTM(nodes))

modelHigh.add(Dense(1))

modelHigh.compile(loss='mse', optimizer='adam')
modelHigh.fit(high_x_train, high_y_train, batch_size=26, epochs=epochUnit, validation_split=0.1)
```

```
Train on 1686 samples, validate on 188 samples
Epoch 1/100
1686/1686 [==============================] - 3s 2ms/step - loss: 0.0498 - val_lo
ss: 0.0061
Epoch 2/100
1686/1686 [==============================] - 2s 932us/step - loss: 0.0038 - val_
loss: 0.0027
Epoch 3/100
1686/1686 [==============================] - 2s 937us/step - loss: 0.0037 - val_
loss: 0.0026
Epoch 4/100
1686/1686 [==============================] - 2s 943us/step - loss: 0.0034 - val_
loss: 0.0026
Epoch 5/100
1686/1686 [==============================] - 2s 934us/step - loss: 0.0033 - val_
loss: 0.0024
Epoch 6/100
1686/1686 [==============================] - 2s 968us/step - loss: 0.0030 - val_
loss: 0.0024
Epoch 7/100
1686/1686 [==============================] - 2s 957us/step - loss: 0.0029 - val_
loss: 0.0023
Epoch 8/100
1686/1686 [==============================] - 2s 955us/step - loss: 0.0029 - val_
loss: 0.0022
Epoch 9/100
1686/1686 [==============================] - 2s 938us/step - loss: 0.0029 - val_
loss: 0.0022
Epoch 10/100
1686/1686 [==============================] - 2s 957us/step - loss: 0.0028 - val_
loss: 0.0022
Epoch 11/100
1686/1686 [==============================] - 2s 937us/step - loss: 0.0026 - val_
loss: 0.0021
Epoch 12/100
1686/1686 [==============================] - 2s 951us/step - loss: 0.0026 - val_
loss: 0.0021
Epoch 13/100
1686/1686 [==============================] - 2s 939us/step - loss: 0.0025 - val_
loss: 0.0020
Epoch 14/100
1686/1686 [==============================] - 2s 948us/step - loss: 0.0025 - val_
loss: 0.0019
Epoch 15/100
1686/1686 [==============================] - 2s 933us/step - loss: 0.0024 - val_
loss: 0.0019
Epoch 16/100
1686/1686 [==============================] - 2s 975us/step - loss: 0.0024 - val_
loss: 0.0019
Epoch 17/100
1686/1686 [==============================] - 2s 948us/step - loss: 0.0022 - val_
loss: 0.0018
Epoch 18/100
1686/1686 [==============================] - 2s 972us/step - loss: 0.0022 - val_
loss: 0.0017
Epoch 19/100
1686/1686 [==============================] - 2s 951us/step - loss: 0.0021 - val_
loss: 0.0018
Epoch 20/100
1686/1686 [==============================] - 2s 951us/step - loss: 0.0021 - val_
loss: 0.0021
Epoch 21/100
1686/1686 [==============================] - 2s 991us/step - loss: 0.0021 - val_
loss: 0.0017
```

```
Out[33]: <keras.callbacks.History at 0x7f8e3ee850f0>
```

In [34]:

```
#Step 2c Build Model For Low
modelLow = Sequential()
modelLow.add(LSTM(nodes, return_sequences=True, input_shape=(features_set_low.shape
[1], 1)))
modelLow.add(Dropout(dropout))

modelLow.add(LSTM(nodes, return_sequences=True))
modelLow.add(Dropout(dropout))

modelLow.add(LSTM(nodes))

modelLow.add(Dense(1))

modelLow.compile(loss='mse', optimizer='adam')
modelLow.fit(low_x_train, low_y_train, batch_size=26, epochs=epochUnit, validation_
split=0.1)
```

```
Train on 1686 samples, validate on 188 samples
Epoch 1/100
1686/1686 [==============================] - 3s 2ms/step - loss: 0.0455 - val_lo
ss: 0.0046
Epoch 2/100
1686/1686 [==============================] - 2s 919us/step - loss: 0.0040 - val_
loss: 0.0034
Epoch 3/100
1686/1686 [==============================] - 2s 954us/step - loss: 0.0033 - val_
loss: 0.0033
Epoch 4/100
1686/1686 [==============================] - 2s 964us/step - loss: 0.0033 - val_
loss: 0.0031
Epoch 5/100
1686/1686 [==============================] - 2s 947us/step - loss: 0.0032 - val_
loss: 0.0034
Epoch 6/100
1686/1686 [==============================] - 2s 958us/step - loss: 0.0031 - val_
loss: 0.0034
Epoch 7/100
1686/1686 [==============================] - 2s 942us/step - loss: 0.0031 - val_
loss: 0.0029
Epoch 8/100
1686/1686 [==============================] - 2s 943us/step - loss: 0.0029 - val_
loss: 0.0028
Epoch 9/100
1686/1686 [==============================] - 2s 962us/step - loss: 0.0028 - val_
loss: 0.0027
Epoch 10/100
1686/1686 [==============================] - 2s 946us/step - loss: 0.0026 - val_
loss: 0.0025
Epoch 11/100
1686/1686 [==============================] - 2s 946us/step - loss: 0.0026 - val_
loss: 0.0025
Epoch 12/100
1686/1686 [==============================] - 2s 957us/step - loss: 0.0025 - val_
loss: 0.0024
Epoch 13/100
1686/1686 [==============================] - 2s 948us/step - loss: 0.0024 - val_
loss: 0.0023
Epoch 14/100
1686/1686 [==============================] - 2s 949us/step - loss: 0.0023 - val_
loss: 0.0024
Epoch 15/100
1686/1686 [==============================] - 2s 946us/step - loss: 0.0023 - val_
loss: 0.0023
Epoch 16/100
1686/1686 [==============================] - 2s 929us/step - loss: 0.0023 - val_
loss: 0.0022
Epoch 17/100
1686/1686 [==============================] - 2s 952us/step - loss: 0.0021 - val_
loss: 0.0021
Epoch 18/100
1686/1686 [==============================] - 2s 938us/step - loss: 0.0022 - val_
loss: 0.0020
Epoch 19/100
1686/1686 [==============================] - 2s 948us/step - loss: 0.0020 - val_
loss: 0.0020
Epoch 20/100
1686/1686 [==============================] - 2s 946us/step - loss: 0.0021 - val_
loss: 0.0020
Epoch 21/100
1686/1686 [==============================] - 2s 977us/step - loss: 0.0019 - val_
loss: 0.0021
```

```
Out[34]: <keras.callbacks.History at 0x7f8e3cfeaa20>
```

In [35]:

```
#Step 2d Build Model For Close
modelClose = Sequential()
modelClose.add(LSTM(nodes, return_sequences=True, input_shape=(features_set_close.shape[1], 1)))
modelClose.add(Dropout(dropout))

modelClose.add(LSTM(nodes, return_sequences=True))
modelClose.add(Dropout(dropout))

modelClose.add(LSTM(nodes))

modelClose.add(Dense(1, activation="linear"))

modelClose.compile(loss='mse', optimizer='adam')
modelClose.fit(close_x_train, close_y_train, batch_size=26, epochs=epochUnit, validation_split=0.1)
```

```
Train on 1686 samples, validate on 188 samples
Epoch 1/100
1686/1686 [==============================] - 4s 2ms/step - loss: 0.0633 - val_lo
ss: 0.0073
Epoch 2/100
1686/1686 [==============================] - 2s 932us/step - loss: 0.0053 - val_
loss: 0.0022
Epoch 3/100
1686/1686 [==============================] - 2s 953us/step - loss: 0.0034 - val_
loss: 0.0019
Epoch 4/100
1686/1686 [==============================] - 2s 958us/step - loss: 0.0034 - val_
loss: 0.0020
Epoch 5/100
1686/1686 [==============================] - 2s 961us/step - loss: 0.0030 - val_
loss: 0.0021
Epoch 6/100
1686/1686 [==============================] - 2s 946us/step - loss: 0.0031 - val_
loss: 0.0019
Epoch 7/100
1686/1686 [==============================] - 2s 954us/step - loss: 0.0027 - val_
loss: 0.0020
Epoch 8/100
1686/1686 [==============================] - 2s 947us/step - loss: 0.0027 - val_
loss: 0.0020
Epoch 9/100
1686/1686 [==============================] - 2s 943us/step - loss: 0.0027 - val_
loss: 0.0018
Epoch 10/100
1686/1686 [==============================] - 2s 957us/step - loss: 0.0026 - val_
loss: 0.0018
Epoch 11/100
1686/1686 [==============================] - 2s 964us/step - loss: 0.0024 - val_
loss: 0.0019
Epoch 12/100
1686/1686 [==============================] - 2s 957us/step - loss: 0.0025 - val_
loss: 0.0019
Epoch 13/100
1686/1686 [==============================] - 2s 978us/step - loss: 0.0024 - val_
loss: 0.0019
Epoch 14/100
1686/1686 [==============================] - 2s 958us/step - loss: 0.0022 - val_
loss: 0.0018
Epoch 15/100
1686/1686 [==============================] - 2s 981us/step - loss: 0.0021 - val_
loss: 0.0018
Epoch 16/100
1686/1686 [==============================] - 2s 980us/step - loss: 0.0022 - val_
loss: 0.0016
Epoch 17/100
1686/1686 [==============================] - 2s 952us/step - loss: 0.0022 - val_
loss: 0.0015
Epoch 18/100
1686/1686 [==============================] - 2s 987us/step - loss: 0.0020 - val_
loss: 0.0015
Epoch 19/100
1686/1686 [==============================] - 2s 967us/step - loss: 0.0020 - val_
loss: 0.0022
Epoch 20/100
1686/1686 [==============================] - 2s 967us/step - loss: 0.0020 - val_
loss: 0.0015
Epoch 21/100
1686/1686 [==============================] - 2s 947us/step - loss: 0.0018 - val_
loss: 0.0014
```

Out[35]:
```
<keras.callbacks.History at 0x7f8e36a9ae10>
```

In [36]:
```
#Step 3a - Predict with test data!
predictions_open = modelOpen.predict(open_x_test)
predictions_high = modelHigh.predict(high_x_test)
predictions_low = modelLow.predict(low_x_test)
predictions_close = modelClose.predict(close_x_test)
```

In [37]:
```
#SAVE!
numpy.savetxt('predictions_open.csv', predictions_open, delimiter=',')
numpy.savetxt('predictions_high.csv', predictions_high, delimiter=',')
numpy.savetxt('predictions_low.csv', predictions_low, delimiter=',')
numpy.savetxt('predictions_close.csv', predictions_close, delimiter=',')
```

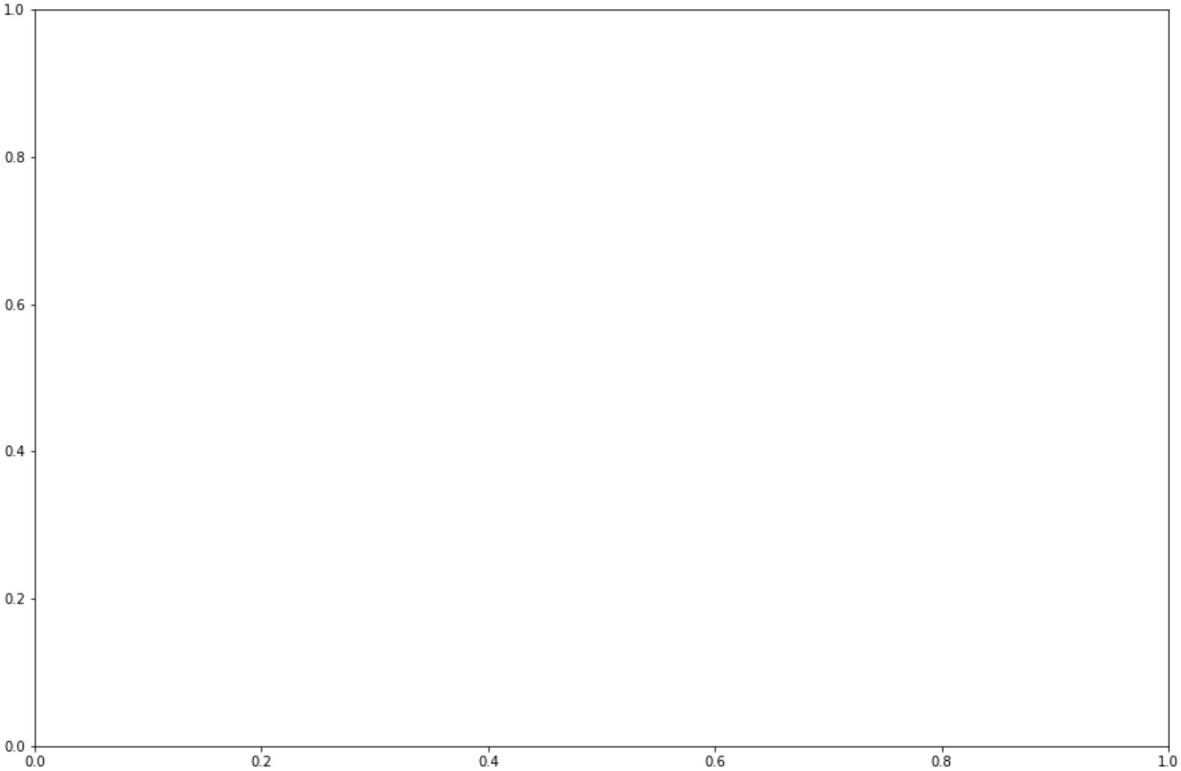
In [38]:
```
#Step 3b - Inverse scale!
inversed_open = scaler.inverse_transform(predictions_open)
inversed_high = scaler.inverse_transform(predictions_high)
inversed_low = scaler.inverse_transform(predictions_low)
inversed_close = scaler.inverse_transform(predictions_close)
```

In [39]:
```
#SAVE!
numpy.savetxt('inversed_open.csv', inversed_open, delimiter=',')
numpy.savetxt('inversed_high.csv', inversed_high, delimiter=',')
numpy.savetxt('inversed_low.csv', inversed_low, delimiter=',')
numpy.savetxt('inversed_close.csv', inversed_close, delimiter=',')
```

In [40]:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(15, 10))
plt.plot(close_x_test)
plt.plot(predictions_close)
plt.legend(['real', 'predict'])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-40-93c9dda8ec56> in <module>
      2 
      3 plt.figure(figsize=(15, 10))
----> 4 plt.plot(close_x_test)
      5 plt.plot(predictions_close)
      6 plt.legend(['real', 'predict'])

/usr/local/lib/python3.6/dist-packages/matplotlib/pyplot.py in plot(scalex, scal
ey, data, *args, **kwargs)
   2787     return gca().plot(
   2788         *args, scalex=scalex, scaley=scaley, **({"data": data} if data
-> 2789         is not None else {}), **kwargs)
   2790 
   2791 

/usr/local/lib/python3.6/dist-packages/matplotlib/axes/_axes.py in plot(self, sc
alex, scaley, data, *args, **kwargs)
   1664         """
   1665         kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D._alias_ma
p)
-> 1666         lines = [*self._get_lines(*args, data=data, **kwargs)]
   1667         for line in lines:
   1668             self.add_line(line)

/usr/local/lib/python3.6/dist-packages/matplotlib/axes/_base.py in __call__(sel
f, *args, **kwargs)
    223                 this += args[0],
    224                 args = args[1:]
--> 225             yield from self._plot_args(this, kwargs)
    226 
    227     def get_next_color(self):

/usr/local/lib/python3.6/dist-packages/matplotlib/axes/_base.py in _plot_args(se
lf, tup, kwargs)
    389             x, y = index_of(tup[-1])
    390 
--> 391         x, y = self._xy_from_xy(x, y)
    392 
    393         if self.command == 'plot':

/usr/local/lib/python3.6/dist-packages/matplotlib/axes/_base.py in _xy_from_xy(s
elf, x, y)
    271         if x.ndim > 2 or y.ndim > 2:
    272             raise ValueError("x and y can be no greater than 2-D, but ha
ve "
--> 273                              "shapes {} and {}".format(x.shape, y.shap
e))
    274 
    275         if x.ndim == 1:

ValueError: x and y can be no greater than 2-D, but have shapes (99,) and (99, 2
6, 1)
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: