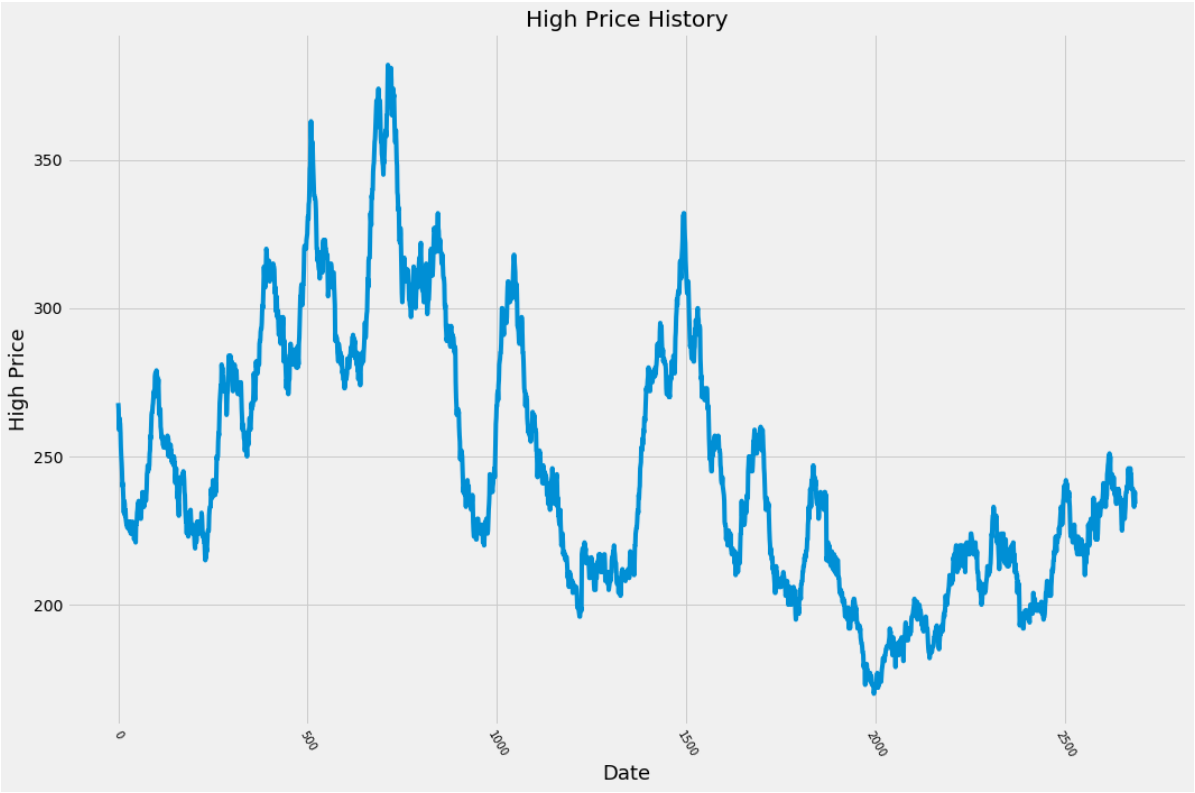
Reinforcement Learning for the High Price

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
from sklearn.preprocessing import MinMaxScaler
```

In [3]:
```
from keras.models import Sequential
from keras.layers import Dense, LSTM
plt.style.use('fivethirtyeight')
```

In [4]:
```
df = pd.read_csv('15_mins_complete_03.csv')
```

In [5]:
```
#Visualising the Close Price History
plt.figure(figsize=(15,10))
plt.title('High Price History')
plt.plot(df['High'])
plt.xlabel('Date', fontsize=18)
plt.ylabel('High Price', fontsize=18)
plt.xticks(rotation=-60, fontsize=10)
plt.tight_layout()
plt.show()
```



In [6]:
```
#New dataframe with only a particular column and convert it to nump
y array
data = df.filter(['High'])
```

In [7]:
```
dataset = data.values
```

In [8]:
```
#Convert data to a numpy array
training_data_len = math.ceil(len(dataset) * 0.8)
print(training_data_len)
```

```
2149
```

In [9]:
```
#Scailing of the data
scalar = MinMaxScaler(feature_range=(0,1))
```

In [10]:
```
scaled_data = scalar.fit_transform(dataset)
```

In [11]:
```
#Scaled training dataset
train_data = scaled_data[0:training_data_len, :]
```

In [12]:
```
#Splitting into Xtrain, Y_train
X_train = []
Y_train = []

for i in range(101, len(train_data)):
    X_train.append(train_data[i-101:i, 0])
    Y_train.append(train_data[i, 0])
```

In [13]:
```
#Convert xtrain and ytrain to numpy arrays
X_train, Y_train = np.array(X_train), np.array(Y_train)
```

In [14]:
```
#Reshape Xtrain as LSTM takes 3d input
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1],
1))
```

In [15]:
```
X_train.shape
```

Out[15]:
```
(2048, 101, 1)
```

In [16]:
```
#Building the LSTM model
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape = (X_train.sh
ape[1], 1)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(10))
model.add(Dense(1))
```

```
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versi
ons/3.7/lib/python3.7/site-packages/keras/backend/tensorflow_backe
nd.py:68: The name tf.get_default_graph is deprecated. Please use
tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versi
ons/3.7/lib/python3.7/site-packages/keras/backend/tensorflow_backe
nd.py:504: The name tf.placeholder is deprecated. Please use tf.co
mpat.v1.placeholder instead.

WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versi
ons/3.7/lib/python3.7/site-packages/keras/backend/tensorflow_backe
nd.py:3828: The name tf.random_uniform is deprecated. Please use t
f.random.uniform instead.
```

In [17]:
```
#Compiling the model
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versi
ons/3.7/lib/python3.7/site-packages/keras/optimizers.py:744: The n
ame tf.train.Optimizer is deprecated. Please use tf.compat.v1.trai
n.Optimizer instead.
```

In [18]:
```
#Training the model
model.fit(X_train, Y_train, batch_size=1, epochs=1)
```

```
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versi
ons/3.7/lib/python3.7/site-packages/tensorflow/python/ops/math_gra
d.py:1250: add_dispatch_support.<locals>.wrapper (from tensorflow.
python.ops.array_ops) is deprecated and will be removed in a futur
e version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versi
ons/3.7/lib/python3.7/site-packages/keras/backend/tensorflow_backe
nd.py:973: The name tf.assign_add is deprecated. Please use tf.com
pat.v1.assign_add instead.

WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versi
ons/3.7/lib/python3.7/site-packages/keras/backend/tensorflow_backe
nd.py:960: The name tf.assign is deprecated. Please use tf.compat.
v1.assign instead.

Epoch 1/1
2048/2048 [==============================] - 177s 86ms/step - loss
: 0.0020
```

Out[18]:
```
<keras.callbacks.History at 0x7fd3cbe9b908>
```

In [20]:
```
#Creating the test dataset
test_data = scaled_data[training_data_len - 101: , :]
```

In [21]:
```
len(test_data)
```

Out[21]:
```
638
```

In [22]:
```
#Splitting into X_test, Y_test
X_test = []
Y_test = dataset[training_data_len:, :]

for i in range(101, len(test_data)):
    X_test.append(test_data[i-101 : i, 0])
```

In [23]:
```
#Converting data to numpy array and then 3d
X_test = np.array(X_test)
```

In [24]:
```
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
```

In [25]:
```
X_test.shape
```

Out[25]:
```
(537, 101, 1)
```

In [26]:
```
#Predictions
predictions = model.predict(X_test)
```

In [27]:
```
predictions = scalar.inverse_transform(predictions)
```

In [28]:
```
#RMSE
rmse = np.sqrt(np.mean(predictions - Y_test) ** 2)
```

In [29]:
```
rmse
```

Out[29]:
```
0.10373390808673766
```

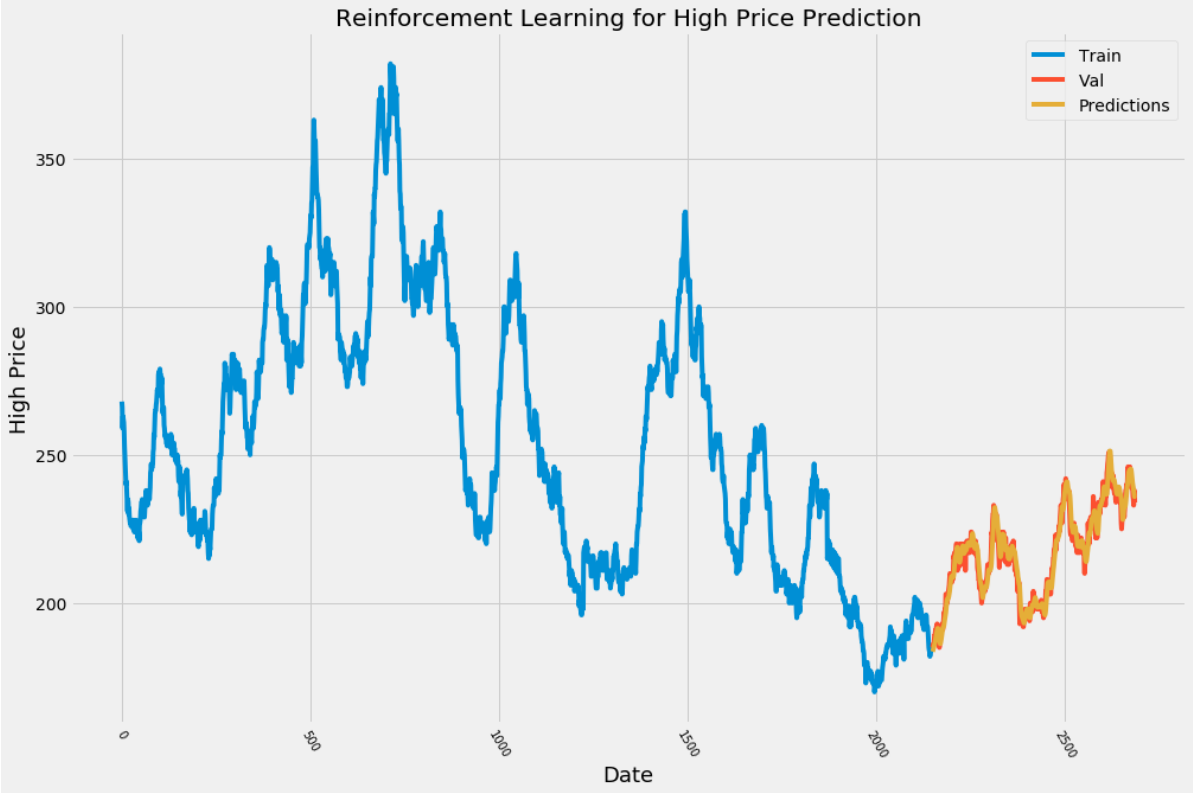
In [30]:
```
#Plotting the data
train = data[ : training_data_len]
valid = data[ training_data_len : ]

valid['Predictions'] = predictions
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/si
te-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
-copy
  """
```

In [31]:
```
plt.figure(figsize=(15,10))
plt.title('Reinforcement Learning for High Price Prediction')
plt.xlabel('Date', fontsize=18)
plt.ylabel('High Price', fontsize=18)
plt.plot(train['High'])
plt.plot(valid[['High', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'], loc='upper right')
plt.xticks(rotation=-60, fontsize=10)
plt.show()
```



In [32]:
```
valid
```

Out[32]:

	High	Predictions
2149	185.0	183.617279
2150	186.0	183.965225
2151	186.0	184.460876
2152	186.0	184.951996
2153	187.0	185.351822
...	...	...
2681	235.0	238.926605
2682	233.0	237.840820
2683	236.0	236.516678
2684	238.0	235.888779
2685	234.0	236.134048

537 rows × 2 columns

In [ ]:
```

```