Chapter 19 - exercise 2: Bitcoin Prediction

- Cho dữ liệu bitcoin_price.csv. Áp dụng mô hình HoltWinters để dự báo Close price of bitcoin cho 3 tháng tiếp theo.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

Đọc dữ liệu, kiểm tra/định dạng thời gian

In [2]:

```
df = pd.read_csv('bitcoin_price.csv',
                 parse_dates=['Date'],
                 index_col='Date')
```

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1655 entries, 2017-11-07 to 2013-04-28
Data columns (total 6 columns):
Open          1655 non-null float64
High          1655 non-null float64
Low           1655 non-null float64
Close         1655 non-null float64
Volume        1655 non-null object
Market Cap    1655 non-null object
dtypes: float64(4), object(2)
memory usage: 90.5+ KB
```

In [4]:

```
df.head()
```

Out[4]:

	Open	High	Low	Close	Volume	Market Cap
Date						
2017-11-07	7023.10	7253.32	7023.10	7144.38	2,326,340,000	117,056,000,000
2017-11-06	7403.22	7445.77	7007.31	7022.76	3,111,900,000	123,379,000,000
2017-11-05	7404.52	7617.48	7333.19	7407.41	2,380,410,000	123,388,000,000
2017-11-04	7164.48	7492.86	7031.28	7379.95	2,483,800,000	119,376,000,000
2017-11-03	7087.53	7461.29	7002.94	7207.76	3,369,860,000	118,084,000,000

In [5]:
```
df = df[['Close']]
df.head()
```

Out[5]:

	Close
Date	
2017-11-07	7144.38
2017-11-06	7022.76
2017-11-05	7407.41
2017-11-04	7379.95
2017-11-03	7207.76

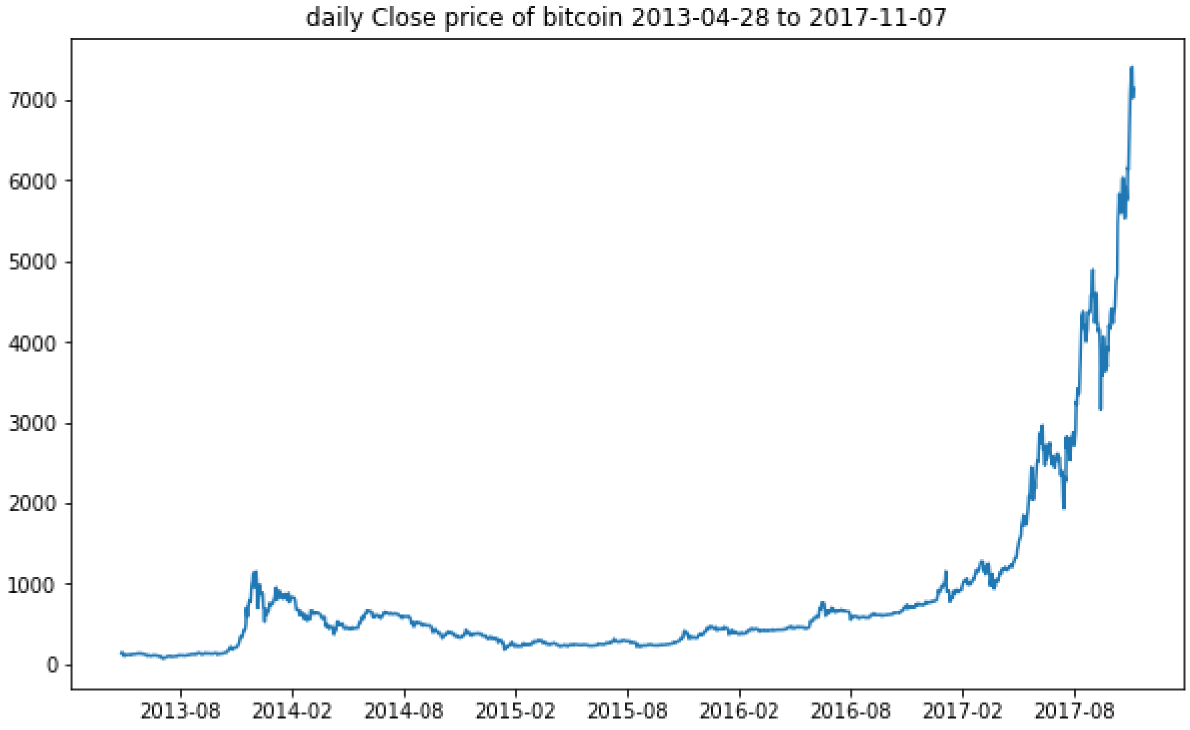
In [6]:
```
df.tail()
```

Out[6]:

	Close
Date	
2013-05-02	105.21
2013-05-01	116.99
2013-04-30	139.00
2013-04-29	144.54
2013-04-28	134.21

In [7]:

```
plt.figure(figsize=(10,6))
plt.plot(df)
plt.title("daily Close price of bitcoin 2013-04-28 to 2017-11-07")
plt.show()
```



Decomposition

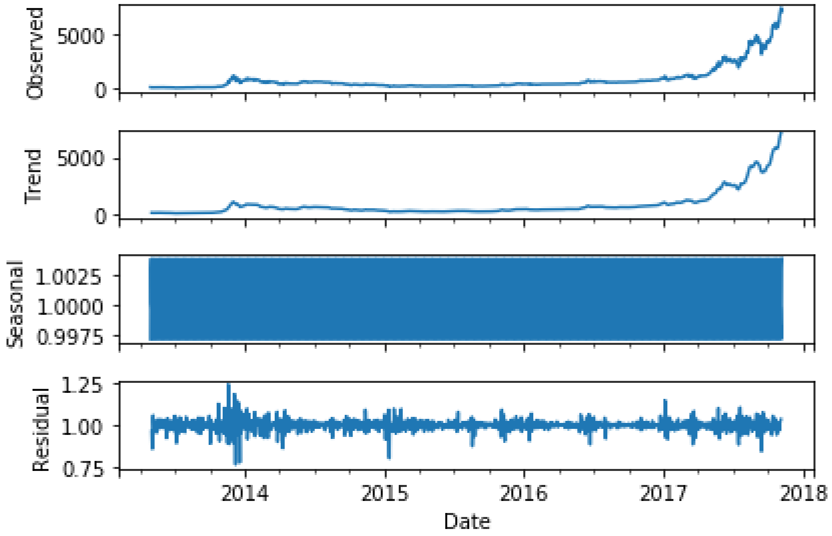
In [8]:

```
from statsmodels.tsa.seasonal import seasonal_decompose
result = seasonal_decompose(df, model='multiplicative')
result
```

Out[8]:

```
<statsmodels.tsa.seasonal.DecomposeResult at 0x1a47fec1dd8>
```

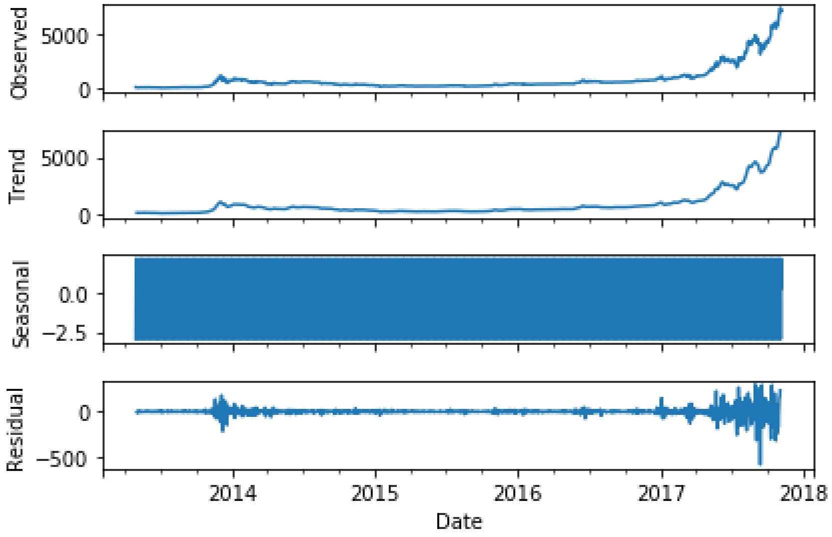
In [9]:
```
result.plot()
plt.show()
```



In [10]:
```
result1 = seasonal_decompose(df, model='additive')
result1
```

Out[10]:
```
<statsmodels.tsa.seasonal.DecomposeResult at 0x1a40542cd30>
```

In [11]:
```
result1.plot()
plt.show()
```



Chia dữ liệu train/test => Áp dụng mô hình

```
In [12]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1655 entries, 2017-11-07 to 2013-04-28
Data columns (total 1 columns):
Close    1655 non-null float64
dtypes: float64(1)
memory usage: 25.9 KB
```

```
In [13]: df = df.sort_index()
```

```
In [14]: train, test=df.iloc[0:1500,0], df.iloc[1500:, 0]
```

```
In [15]: train[0:5]
```

```
Out[15]: Date
         2013-04-28    134.21
         2013-04-29    144.54
         2013-04-30    139.00
         2013-05-01    116.99
         2013-05-02    105.21
         Name: Close, dtype: float64
```

```
In [16]: test[0:5]
```

```
Out[16]: Date
         2017-06-06    2863.20
         2017-06-07    2732.16
         2017-06-08    2805.62
         2017-06-09    2823.81
         2017-06-10    2947.71
         Name: Close, dtype: float64
```

```
In [17]: #https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.Exponential
```

```
In [18]: #https://www.statsmodels.org/stable/generated/statsmodels.tsa.holtwinters.Exponent
         model = ExponentialSmoothing(train, seasonal='add', trend='add', seasonal_periods=
```

```
c:\program files\python36\lib\site-packages\statsmodels\tsa\base\tsa_model.py:
171: ValueWarning: No frequency information was provided, so inferred frequenc
y D will be used.
  % freq, ValueWarning)
```

```
In [19]: test.index[0]
```

```
Out[19]: Timestamp('2017-06-06 00:00:00')
```

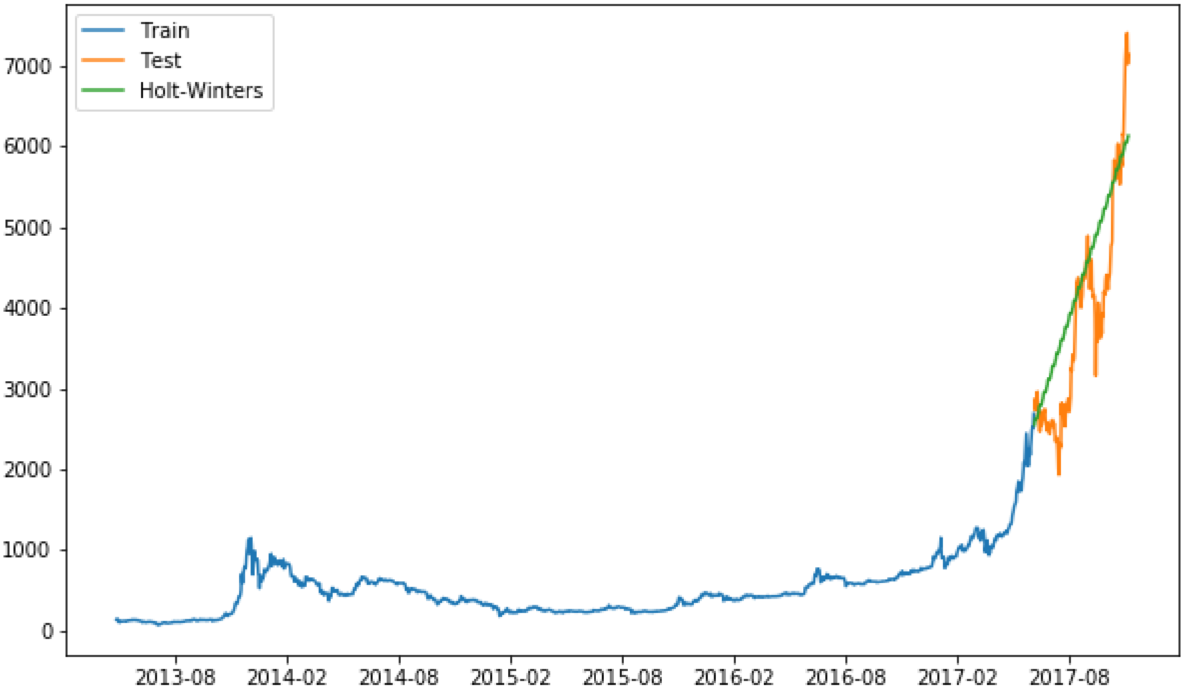
```
In [20]: test.index[-1]
```

```
Out[20]: Timestamp('2017-11-07 00:00:00')
```

```
In [21]: # https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.Exponentia
         pred = model.predict(start=test.index[0], end=test.index[-1])
```

```
In [22]: plt.figure(figsize=(10,6))
         plt.plot(train.index, train, label='Train')
         plt.plot(test.index, test, label='Test')
         plt.plot(pred.index, pred, label='Holt-Winters')
         plt.legend(loc='best')
```

```
Out[22]: <matplotlib.legend.Legend at 0x1a40572e550>
```



```
In [23]: from sklearn.metrics import mean_squared_error, r2_score
         from math import sqrt
         mse = mean_squared_error(test, pred)
         mse
```

```
Out[23]: 579087.9539912037
```

```
In [24]: r2 = r2_score(test,pred)
         r2
```

```
Out[24]: 0.6673148735510708
```

Dự đoán

In [25]:
```
df.tail()
```

Out[25]:

	Close
Date	
2017-11-03	7207.76
2017-11-04	7379.95
2017-11-05	7407.41
2017-11-06	7022.76
2017-11-07	7144.38

In [26]:
```
import datetime
s = datetime.datetime(2017, 11, 7)
e = datetime.datetime(2018, 2, 7)
```

In [27]:
```
s
```

Out[27]:
```
datetime.datetime(2017, 11, 7, 0, 0)
```

In [28]:
```
e
```

Out[28]:
```
datetime.datetime(2018, 2, 7, 0, 0)
```

In [29]:
```
pred_next_3_month = model.predict(start= s, end=e)
```

In [30]:
```
pred_next_3_month[90:]
```

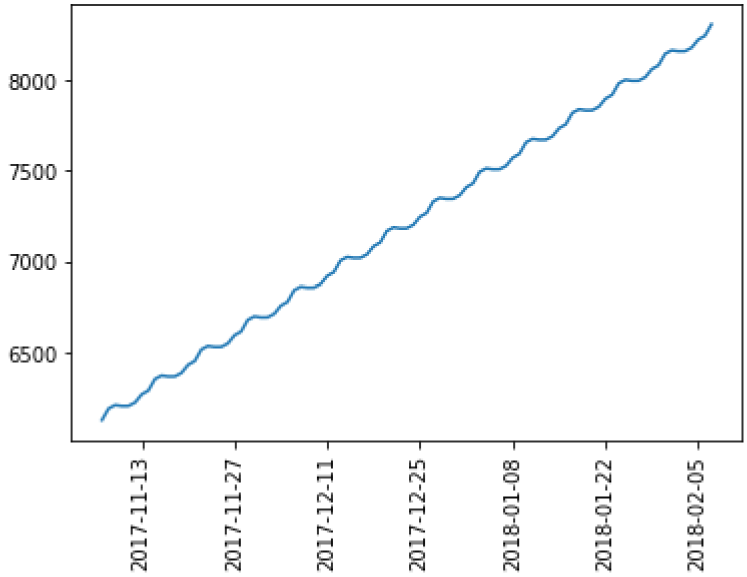
Out[30]:
```
2018-02-05    8224.151316
2018-02-06    8245.241554
2018-02-07    8308.857712
Freq: D, dtype: float64
```

In [31]:
```
x = pd.Series(pred_next_3_month)
type(x)
```

Out[31]:
```
pandas.core.series.Series
```

In [32]:

```
plt.plot(x.index, x.values)
plt.xticks(rotation = 'vertical')
plt.show()
```



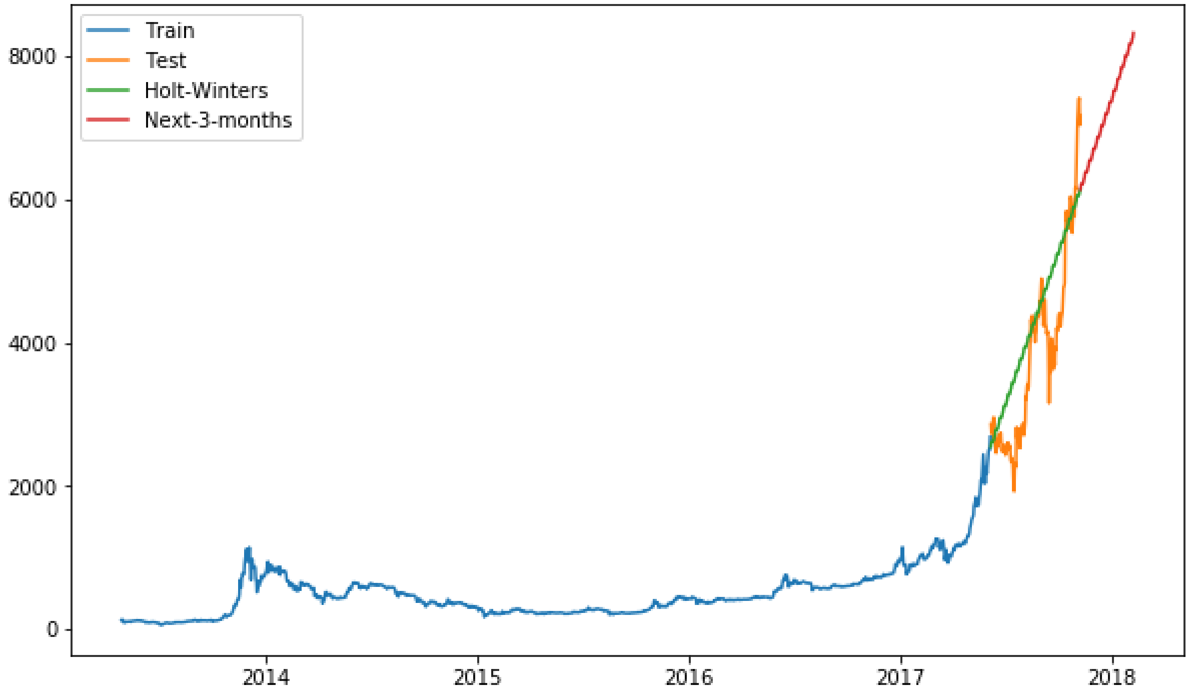
Trực quan hóa dữ liệu

In [33]:

```
plt.figure(figsize=(10,6))
plt.plot(train.index, train, label='Train')
plt.plot(test.index, test, label='Test')
plt.plot(pred.index, pred, label='Holt-Winters')
plt.plot(x.index, x.values, label='Next-3-months')
plt.legend(loc='best')
```

Out[33]:

```
<matplotlib.legend.Legend at 0x1a405bf5978>
```


Train
Test
Holt-Winters
Next-3-months