Module 4

- By Suprateek Halsana

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score,mean_squared_error
from sklearn.model_selection import train_test_split
sns.set(style='darkgrid')
import warnings
warnings.filterwarnings('ignore')
```

Problem [4.1]

In [2]:
```
data=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\week3.csv")
data
```

Out[2]:

	Date	Symbol	Series	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	...	Deliverable Qty	% Dly Qt to Traded Qty	Month	Year	Da
0	2017-05-15	LALPATHLAB	EQ	891.15	895.0	914.15	881.00	912.00	900.60	889.35	...	174775	67.83	5	2017	
1	2017-05-16	LALPATHLAB	EQ	900.60	910.0	925.00	895.05	909.40	910.95	914.57	...	75813	72.27	5	2017	
2	2017-05-17	LALPATHLAB	EQ	910.95	913.0	925.00	909.00	912.05	911.70	917.19	...	53829	74.29	5	2017	
3	2017-05-18	LALPATHLAB	EQ	911.70	908.0	919.35	903.05	906.00	909.75	914.12	...	24836	65.87	5	2017	
4	2017-05-19	LALPATHLAB	EQ	909.75	917.0	917.00	905.80	910.00	910.25	910.61	...	69926	86.24	5	2017	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
489	2019-05-07	LALPATHLAB	EQ	1024.95	1015.7	1031.95	1006.90	1013.00	1013.10	1020.66	...	11587	57.61	5	2019	
490	2019-05-08	LALPATHLAB	EQ	1013.10	1014.0	1019.85	1001.90	1002.00	1006.10	1010.90	...	11239	62.27	5	2019	
491	2019-05-09	LALPATHLAB	EQ	1006.10	1014.0	1014.00	977.70	979.90	982.90	987.36	...	37303	61.62	5	2019	
492	2019-05-10	LALPATHLAB	EQ	982.90	989.9	994.30	963.60	980.00	980.65	980.71	...	19097	62.47	5	2019	
493	2019-05-13	LALPATHLAB	EQ	980.65	970.8	996.00	960.00	975.00	976.25	974.21	...	23772	63.93	5	2019	

494 rows × 23 columns

Creating Column 'Call'

In [3]:
```
data['Call']=''
for i in range(data.shape[0]):
    if data['Close Price'][i] < data['Lower_Band'][i] :
        data['Call'][i]='Buy'
    elif data['Close Price'][i] > data['Upper_Band'][i] :
        data['Call'][i]='Short'
    elif data['Close Price'][i] > data['Lower_Band'][i] and data['Close Price'][i] < data['14_Day_MV'][
i] :
        data['Call'][i]='Hold Buy'
    elif data['Close Price'][i] > data['14_Day_MV'][i] and data['Close Price'][i] < data['Upper_Band'][
i] :
        data['Call'][i]='Hold Short'
```

In [4]:
```
data
```

Out[4]:

	Date	Symbol	Series	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	...	% Dly Qt to Traded Qty	Month	Year	Day_Perc_chan
0	2017-05-15	LALPATHLAB	EQ	891.15	895.0	914.15	881.00	912.00	900.60	889.35	...	67.83	5	2017	0.0000
1	2017-05-16	LALPATHLAB	EQ	900.60	910.0	925.00	895.05	909.40	910.95	914.57	...	72.27	5	2017	1.1492
2	2017-05-17	LALPATHLAB	EQ	910.95	913.0	925.00	909.00	912.05	911.70	917.19	...	74.29	5	2017	0.0823
3	2017-05-18	LALPATHLAB	EQ	911.70	908.0	919.35	903.05	906.00	909.75	914.12	...	65.87	5	2017	-0.2138
4	2017-05-19	LALPATHLAB	EQ	909.75	917.0	917.00	905.80	910.00	910.25	910.61	...	86.24	5	2017	0.0549
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
489	2019-05-07	LALPATHLAB	EQ	1024.95	1015.7	1031.95	1006.90	1013.00	1013.10	1020.66	...	57.61	5	2019	-1.1561
490	2019-05-08	LALPATHLAB	EQ	1013.10	1014.0	1019.85	1001.90	1002.00	1006.10	1010.90	...	62.27	5	2019	-0.6909
491	2019-05-09	LALPATHLAB	EQ	1006.10	1014.0	1014.00	977.70	979.90	982.90	987.36	...	61.62	5	2019	-2.3059
492	2019-05-10	LALPATHLAB	EQ	982.90	989.9	994.30	963.60	980.00	980.65	980.71	...	62.47	5	2019	-0.2289
493	2019-05-13	LALPATHLAB	EQ	980.65	970.8	996.00	960.00	975.00	976.25	974.21	...	63.93	5	2019	-0.4486

494 rows × 24 columns

In [5]:
```
x=data[['Close Price','14_Day_MV','Upper_Band','Lower_Band']][13:]
y=data['Call'][13:]
```

In [6]:
```
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.25)
```

In [7]:
```
# Decision Tree Classifier
```

In [8]:
```
from sklearn.tree import DecisionTreeClassifier
d=DecisionTreeClassifier()
d.fit(xtrain,ytrain)
yd=d.predict(xtest)
a_dt = accuracy_score(ytest,yd)
print('Accuracy by DTree  :',a_dt)
#print('Mean Squared Error :',mean_squared_error(ytest,yd))
```

```
Accuracy by DTree  : 0.7603305785123967
```

In [9]:
```
# K Nearest Neighbour Classifier
```

In [10]:
```
from sklearn.neighbors import KNeighborsClassifier
k=KNeighborsClassifier()
k.fit(xtrain,ytrain)
yk=k.predict(xtest)
a_KNN = accuracy_score(ytest,yk)
print('Accuracy by KNN :',a_KNN)
#print('Mean_squared_error :',mean_squared_error(ytest,yk))
```

```
Accuracy by KNN : 0.8429752066115702
```

In [13]:
```
from sklearn.svm import SVC
sc=SVC(kernel='poly')
sc.fit(xtrain,ytrain)
ys=sc.predict(xtest)
a_sc=accuracy_score(ytest,ys)
print('Accuracy by SVC :',a_sc)
```

```
Accuracy by SVC : 0.9834710743801653
```

In [14]:
```
from sklearn.linear_model import LogisticRegression
l=LogisticRegression()
l.fit(xtrain,ytrain)
yl=l.predict(xtest)
a_log=accuracy_score(ytest,yl)
print('Accuracy by Logistic Reg :',a_log)
```

```
Accuracy by Logistic Reg : 1.0
```

In [15]:
```
# ************************************************************************** #
# Stock Chosen 'RAYMOND'
# For Prediction of Call
```

In [16]:
```
raymond = pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\Prerequisites\Small_Cap\Small_Cap\RAYMOND.csv")
raymond
```

Out[16]:

	Symbol	Series	Date	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	Total Traded Quantity	Turnover	No. of Trades	Deliverable Qty	% D Qt Trad Q
0	RAYMOND	EQ	15-May-2017	763.45	765.00	779.00	757.05	773.90	772.80	770.09	292498	2.252491e+08	9033	55467	18.
1	RAYMOND	EQ	16-May-2017	772.80	773.45	801.45	768.10	783.50	785.00	788.89	823956	6.500070e+08	18238	179377	21.
2	RAYMOND	EQ	17-May-2017	785.00	786.00	804.95	778.40	785.50	783.65	792.72	598094	4.741212e+08	13968	127637	21.
3	RAYMOND	EQ	18-May-2017	783.65	779.35	779.95	740.20	744.00	746.95	762.57	319834	2.438956e+08	7713	92200	28.
4	RAYMOND	EQ	19-May-2017	746.95	748.00	767.50	714.00	722.00	723.10	740.30	561346	4.155619e+08	13328	137648	24.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
489	RAYMOND	EQ	07-May-2019	806.05	806.05	821.60	799.25	811.15	808.95	814.61	2038484	1.660574e+09	33444	321610	15.
490	RAYMOND	EQ	08-May-2019	808.95	807.90	816.90	797.05	806.40	807.10	807.55	930010	7.510262e+08	19028	51327	5.
491	RAYMOND	EQ	09-May-2019	807.10	801.00	811.10	798.00	805.25	805.30	804.23	425205	3.419611e+08	8608	34675	8.
492	RAYMOND	EQ	10-May-2019	805.30	809.95	815.00	795.05	804.00	804.40	804.95	464539	3.739298e+08	9381	22009	4.
493	RAYMOND	EQ	13-May-2019	804.40	803.00	803.00	774.75	782.00	779.35	789.32	342185	2.700928e+08	9023	29418	8.

494 rows × 15 columns

In [17]:
```
raymond['14_Day_MV']=raymond['Close Price'].rolling(window=14).mean()
raymond['14_Day_STD']=raymond['Close Price'].rolling(window=14).std()
raymond['Upper_Band']=raymond['14_Day_MV']+raymond['14_Day_STD']*2
raymond['Lower_Band']=raymond['14_Day_MV']-raymond['14_Day_STD']*2
raymond
```

Out[17]:

	Symbol	Series	Date	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	Total Traded Quantity	Turnover	No. of Trades	Deliverable Qty	% D Qt Trad Q
0	RAYMOND	EQ	15-May-2017	763.45	765.00	779.00	757.05	773.90	772.80	770.09	292498	2.252491e+08	9033	55467	18.
1	RAYMOND	EQ	16-May-2017	772.80	773.45	801.45	768.10	783.50	785.00	788.89	823956	6.500070e+08	18238	179377	21.
2	RAYMOND	EQ	17-May-2017	785.00	786.00	804.95	778.40	785.50	783.65	792.72	598094	4.741212e+08	13968	127637	21.
3	RAYMOND	EQ	18-May-2017	783.65	779.35	779.95	740.20	744.00	746.95	762.57	319834	2.438956e+08	7713	92200	28.
4	RAYMOND	EQ	19-May-2017	746.95	748.00	767.50	714.00	722.00	723.10	740.30	561346	4.155619e+08	13328	137648	24.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
489	RAYMOND	EQ	07-May-2019	806.05	806.05	821.60	799.25	811.15	808.95	814.61	2038484	1.660574e+09	33444	321610	15.
490	RAYMOND	EQ	08-May-2019	808.95	807.90	816.90	797.05	806.40	807.10	807.55	930010	7.510262e+08	19028	51327	5.
491	RAYMOND	EQ	09-May-2019	807.10	801.00	811.10	798.00	805.25	805.30	804.23	425205	3.419611e+08	8608	34675	8.
492	RAYMOND	EQ	10-May-2019	805.30	809.95	815.00	795.05	804.00	804.40	804.95	464539	3.739298e+08	9381	22009	4.
493	RAYMOND	EQ	13-May-2019	804.40	803.00	803.00	774.75	782.00	779.35	789.32	342185	2.700928e+08	9023	29418	8.

494 rows × 19 columns

In [18]:
```
raymond['Call']=''
for i in range(data.shape[0]):
    if raymond['Close Price'][i] < raymond['Lower_Band'][i] :
        raymond['Call'][i]='Buy'
    elif raymond['Close Price'][i] > raymond['Upper_Band'][i] :
        raymond['Call'][i]='Short'
    elif raymond['Close Price'][i] > raymond['Lower_Band'][i] and raymond['Close Price'][i] < raymond[
'14_Day_MV'][i] :
        raymond['Call'][i]='Hold Buy'
    elif raymond['Close Price'][i] > raymond['14_Day_MV'][i] and raymond['Close Price'][i] < raymond['U
pper_Band'][i] :
        raymond['Call'][i]='Hold Short'
```

In [19]:
```
raymond
```

Out[19]:

	Symbol	Series	Date	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	Total Traded Quantity	Turnover	No. of Trades	Deliverable Qty	% D Qt Trad Q
0	RAYMOND	EQ	15-May-2017	763.45	765.00	779.00	757.05	773.90	772.80	770.09	292498	2.252491e+08	9033	55467	18.
1	RAYMOND	EQ	16-May-2017	772.80	773.45	801.45	768.10	783.50	785.00	788.89	823956	6.500070e+08	18238	179377	21.
2	RAYMOND	EQ	17-May-2017	785.00	786.00	804.95	778.40	785.50	783.65	792.72	598094	4.741212e+08	13968	127637	21.
3	RAYMOND	EQ	18-May-2017	783.65	779.35	779.95	740.20	744.00	746.95	762.57	319834	2.438956e+08	7713	92200	28.
4	RAYMOND	EQ	19-May-2017	746.95	748.00	767.50	714.00	722.00	723.10	740.30	561346	4.155619e+08	13328	137648	24.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
489	RAYMOND	EQ	07-May-2019	806.05	806.05	821.60	799.25	811.15	808.95	814.61	2038484	1.660574e+09	33444	321610	15.
490	RAYMOND	EQ	08-May-2019	808.95	807.90	816.90	797.05	806.40	807.10	807.55	930010	7.510262e+08	19028	51327	5.
491	RAYMOND	EQ	09-May-2019	807.10	801.00	811.10	798.00	805.25	805.30	804.23	425205	3.419611e+08	8608	34675	8.
492	RAYMOND	EQ	10-May-2019	805.30	809.95	815.00	795.05	804.00	804.40	804.95	464539	3.739298e+08	9381	22009	4.
493	RAYMOND	EQ	13-May-2019	804.40	803.00	803.00	774.75	782.00	779.35	789.32	342185	2.700928e+08	9023	29418	8.

494 rows × 20 columns

For the Prediction Logistic Model is Used as it gives best Prediction

In [20]:
```
x_ray_test=raymond[['Close Price','14_Day_MV','Upper_Band','Lower_Band']][13:]
y=l.predict(x_ray_test)
raymond['Call_Pred']=''
raymond['Call_Pred'][13:]=y
```

In [21]:
```
raymond[['Call','Call_Pred']][13:]
```

Out[21]:

	Call	Call_Pred
13	Hold Buy	Hold Buy
14	Hold Buy	Hold Buy
15	Hold Short	Hold Short
16	Hold Short	Hold Short
17	Hold Short	Hold Short
...	...	...
489	Hold Short	Hold Short
490	Hold Short	Hold Short
491	Hold Short	Hold Short
492	Hold Short	Hold Short
493	Hold Short	Hold Short

481 rows × 2 columns

In [22]:
```
# Count of the Prediction exactly true out of 494 values
len(raymond[raymond['Call']==raymond['Call_Pred']])
```

Out[22]: 488

Problem [4.2]

In [23]:
```
# Stock Chosen : 'ASHOKA'
```

In [24]:
```
ashoka=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\Prerequisites\Small_Cap\Small_Cap\ASHOKA.csv")
ashoka
```

Out[24]:

	Symbol	Series	Date	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	Total Traded Quantity	Turnover	No. of Trades	Deliverable Qty	% Dly Qt to Traded Qty
0	ASHOKA	EQ	15-May-2017	210.35	211.00	217.50	206.15	216.30	216.05	212.63	159997	34020520.45	3506	88509	55.32
1	ASHOKA	EQ	16-May-2017	216.05	212.30	216.00	211.00	214.00	214.90	213.24	213214	45466299.10	2860	151941	71.26
2	ASHOKA	EQ	17-May-2017	214.90	214.30	218.70	212.10	216.55	217.00	215.97	383131	82743402.10	3297	261939	68.37
3	ASHOKA	EQ	18-May-2017	217.00	216.00	216.60	208.00	210.00	209.65	211.74	78709	16665707.55	1372	35458	45.05
4	ASHOKA	EQ	19-May-2017	209.65	212.10	213.90	205.05	206.05	206.25	209.95	398725	83713500.05	2060	301747	75.68
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
489	ASHOKA	EQ	07-May-2019	116.05	118.00	124.15	117.45	117.50	118.10	119.09	68586	8167796.10	1690	25784	37.59
490	ASHOKA	EQ	08-May-2019	118.10	117.00	120.45	115.00	115.10	115.45	117.13	67803	7941511.00	1843	46000	67.84
491	ASHOKA	EQ	09-May-2019	115.45	115.45	117.00	114.80	115.05	115.30	115.53	33182	3833502.00	817	18750	56.51
492	ASHOKA	EQ	10-May-2019	115.30	115.50	118.00	114.35	117.90	117.45	116.64	43958	5127319.60	1490	22355	50.86
493	ASHOKA	EQ	13-May-2019	117.45	117.95	119.70	116.75	116.75	117.15	117.36	46832	5496355.60	1527	32612	69.64

494 rows × 15 columns

In [199]:
```
ashoka['per_oc']=''
ashoka['per_hl']=''
for i in range(ashoka.shape[0]):
    ashoka['per_oc'][i]=1-(ashoka['Close Price'][i]/ashoka['Open Price'][i])
    ashoka['per_hl'][i]=1-(ashoka['Low Price'][i]/ashoka['High Price'][i])
ashoka['5_Day_MV']=(ashoka['Close Price'].pct_change()).rolling(window=5).mean()
ashoka['5_Day_std']=(ashoka['Close Price'].pct_change()).rolling(window=5).std()
```

In [200]:
```
ashoka
```

Out[200]:

	Symbol	Series	Date	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	Total Traded Quantity	Turnover	No. of Trades	Deliverable Qty	% Dly Qt to Traded Qty
0	ASHOKA	EQ	15-May-2017	210.35	211.00	217.50	206.15	216.30	216.05	212.63	159997	34020520.45	3506	88509	55.32
1	ASHOKA	EQ	16-May-2017	216.05	212.30	216.00	211.00	214.00	214.90	213.24	213214	45466299.10	2860	151941	71.26
2	ASHOKA	EQ	17-May-2017	214.90	214.30	218.70	212.10	216.55	217.00	215.97	383131	82743402.10	3297	261939	68.37
3	ASHOKA	EQ	18-May-2017	217.00	216.00	216.60	208.00	210.00	209.65	211.74	78709	16665707.55	1372	35458	45.05
4	ASHOKA	EQ	19-May-2017	209.65	212.10	213.90	205.05	206.05	206.25	209.95	398725	83713500.05	2060	301747	75.68
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
489	ASHOKA	EQ	07-May-2019	116.05	118.00	124.15	117.45	117.50	118.10	119.09	68586	8167796.10	1690	25784	37.59
490	ASHOKA	EQ	08-May-2019	118.10	117.00	120.45	115.00	115.10	115.45	117.13	67803	7941511.00	1843	46000	67.84
491	ASHOKA	EQ	09-May-2019	115.45	115.45	117.00	114.80	115.05	115.30	115.53	33182	3833502.00	817	18750	56.51
492	ASHOKA	EQ	10-May-2019	115.30	115.50	118.00	114.35	117.90	117.45	116.64	43958	5127319.60	1490	22355	50.86
493	ASHOKA	EQ	13-May-2019	117.45	117.95	119.70	116.75	116.75	117.15	117.36	46832	5496355.60	1527	32612	69.64

494 rows × 19 columns

In [201]:
```
ashoka['Action']=0
for i in range(1,ashoka.shape[0]):
    if(ashoka['Close Price'][i]>ashoka['Close Price'][i-1]):
        ashoka['Action'][i]=1
    else:
        ashoka['Action'][i]=-1
```

In [202]:
```
ashoka
```

Out[202]:

	Symbol	Series	Date	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	Total Traded Quantity	Turnover	No. of Trades	Deliverable Qty	% Dly Qt to Traded Qty
0	ASHOKA	EQ	15-May-2017	210.35	211.00	217.50	206.15	216.30	216.05	212.63	159997	34020520.45	3506	88509	55.32
1	ASHOKA	EQ	16-May-2017	216.05	212.30	216.00	211.00	214.00	214.90	213.24	213214	45466299.10	2860	151941	71.26
2	ASHOKA	EQ	17-May-2017	214.90	214.30	218.70	212.10	216.55	217.00	215.97	383131	82743402.10	3297	261939	68.37
3	ASHOKA	EQ	18-May-2017	217.00	216.00	216.60	208.00	210.00	209.65	211.74	78709	16665707.55	1372	35458	45.05
4	ASHOKA	EQ	19-May-2017	209.65	212.10	213.90	205.05	206.05	206.25	209.95	398725	83713500.05	2060	301747	75.68
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
489	ASHOKA	EQ	07-May-2019	116.05	118.00	124.15	117.45	117.50	118.10	119.09	68586	8167796.10	1690	25784	37.59
490	ASHOKA	EQ	08-May-2019	118.10	117.00	120.45	115.00	115.10	115.45	117.13	67803	7941511.00	1843	46000	67.84
491	ASHOKA	EQ	09-May-2019	115.45	115.45	117.00	114.80	115.05	115.30	115.53	33182	3833502.00	817	18750	56.51
492	ASHOKA	EQ	10-May-2019	115.30	115.50	118.00	114.35	117.90	117.45	116.64	43958	5127319.60	1490	22355	50.86
493	ASHOKA	EQ	13-May-2019	117.45	117.95	119.70	116.75	116.75	117.15	117.36	46832	5496355.60	1527	32612	69.64

494 rows × 20 columns

In [203]:
```
x=ashoka[['per_oc','per_hl','5_Day_MV','5_Day_std']][5:]
y=ashoka['Action'][5:]
```

In [204]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.25)
```

In [205]:
```
# Random Forest Classifier used for prediction
```

In [206]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(xtrain,ytrain)
yrfc=rfc.predict(xtest)
print('Accuracy Score :',accuracy_score(ytest,yrfc))
```

```
Accuracy Score : 0.8780487804878049
```

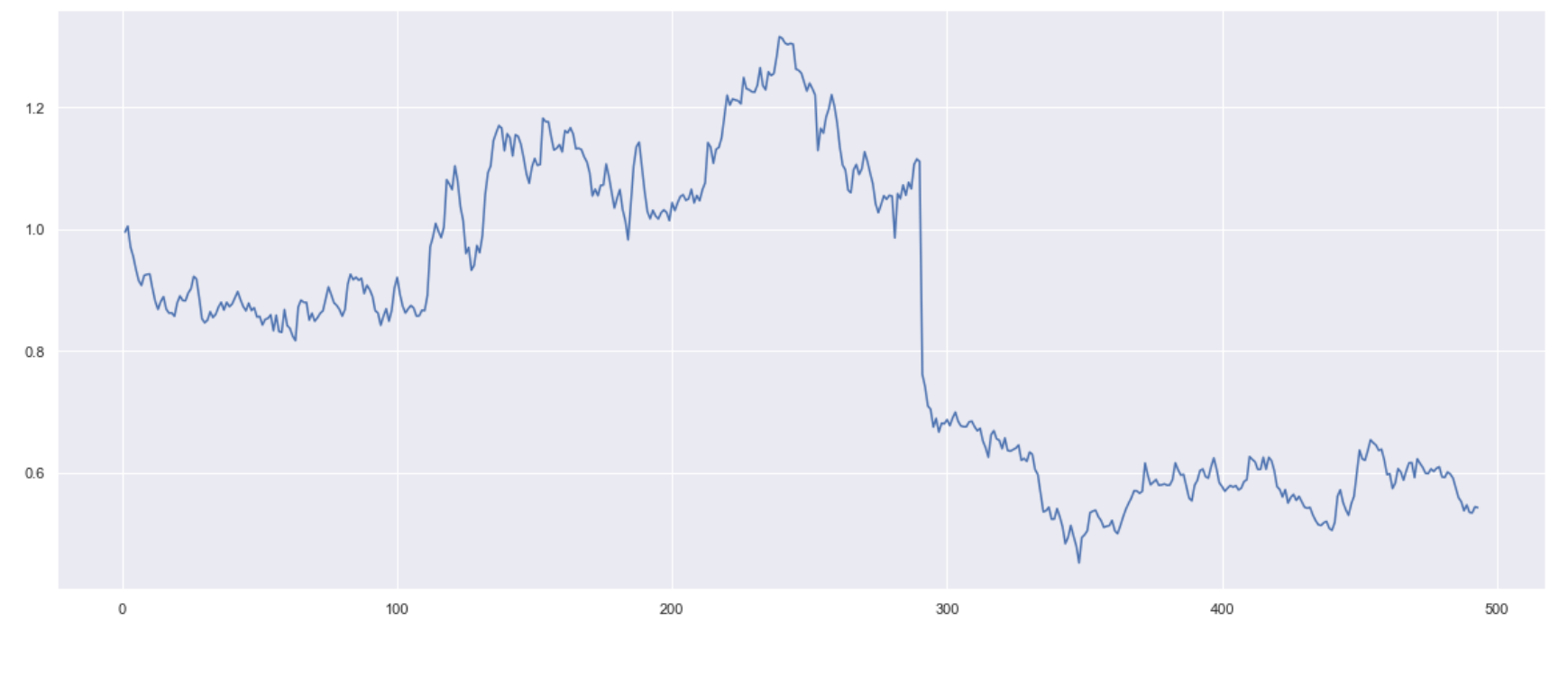
In [207]:
```
# Cummulative Return Visualization
```

In [208]:
```
daily_pc=ashoka['Close Price'].pct_change()[1:]
```

In [209]:
```
cum_ret=(1+daily_pc).cumprod()
```

In [210]:
```
plt.figure(figsize=(20,8))
cum_ret.plot()
```

Out[210]: <matplotlib.axes._subplots.AxesSubplot at 0x1ef398072c8>



The Above is the Cummulative Return Plot of the Stock Ashoka