

in the name of god.

Mohammad jamad zamani

Dr fotoohi

در این پروژه قصد داریم قیمت دو سهام اس اند پی و دیزنی را با مدل گارچ پیش بینی کنیم

```
In [153... import pandas_datareader.data as web
import yfinance as yf
from datetime import datetime, timedelta
import pandas as pd
import matplotlib.pyplot as plt
from arch import arch_model
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.api import VAR
import numpy as np
```

پس از فراخوانی کتابخانه های مورد نیاز از سایت یاهو فایننس اطلاعات دو شرکت را برای بازه زمانی مد نظرممان بارگیری میکنیم

```
In [154... start = datetime(2017, 10, 1)
end = datetime(2024, 10, 1)
```

```
In [155... dis = yf.download('DIS', start = '2017-10-1', end = '2024-10-1')
```

```
[*****100%*****] 1 of 1 completed
```

از آنجایی که نوسان قیمت در طول روز نسبت به قیمت بسیار کم است و از طرفی هدف نهایی ما پیش بینی سود و زیان سرمایه می باشد بنابراین از درصد بازده سهام بجای خود قیمت استفاده میکنیم که حول صفر نوسان دارد

```
In [156... returns = 100 * dis.Close.pct_change().dropna()
```

در این مرحله برای اطمینان از ایستایی سری زمانی از آزمایش دیکی فولر افزوده اسفاده میکنیم

```
In [157... def perform_adf_test(series):
    result = adfuller(series)
    print('ADF Statistic: %f' % result[0])
    print('p-value: %f' % result[1])
```

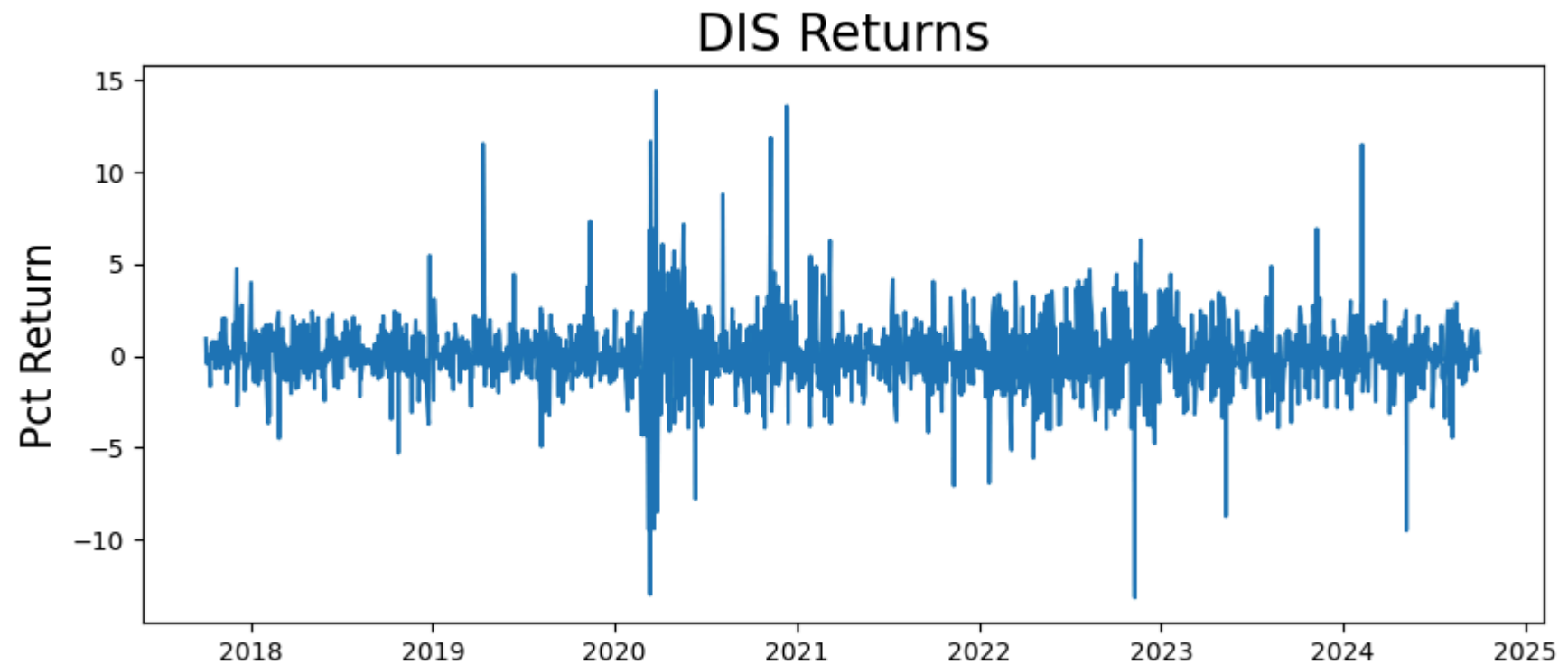
```
In [158... perform_adf_test(returns)
```

ADF Statistic: -13.542533
p-value: 0.000000

نمودار سهام دیزنی را رسم میکنیم

```
In [159... plt.figure(figsize=(10,4))  
plt.plot(returns)  
plt.ylabel('Pct Return', fontsize=16)  
plt.title('DIS Returns', fontsize=20)
```

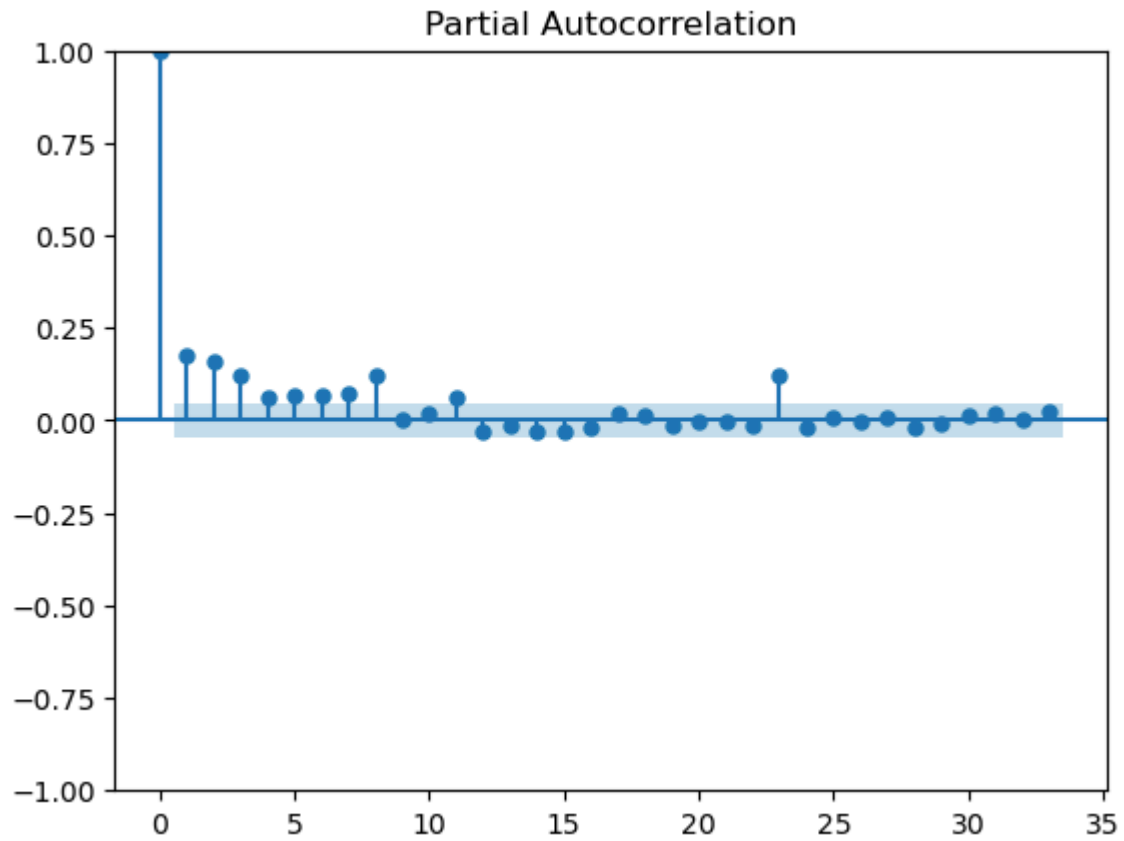
```
Out[159... Text(0.5, 1.0, 'DIS Returns')
```



با توجه به نوسانات ناهمگون در سال های 2020 و 2023 تا 2025 مدل گارچ را برای این سهم پیشنهاد میکنیم و برای بدست آوردن مقدار پارامترها از نمودار ضریب همبستگی جزئی کمک میگیریم

```
In [160... plot_pacf(returns**2)
```

```
plt.show()
```



```
In [161... model = arch_model(returns, p=3, q=3)
```

با توجه به نمودار بالا سه لگ اول در قیمت سهام نقش دارند، وابستگی تا سه روز قبل، حال مدل را با پارامترهای بدست آمده برازش میکنیم

```
In [162... model_fit = model.fit()
```

```
Iteration:      1,  Func. Count:      10,  Neg. LLF: 7827.039233026659
Iteration:      2,  Func. Count:      21,  Neg. LLF: 1116171.7662005096
Iteration:      3,  Func. Count:      32,  Neg. LLF: 3842.053543290488
Iteration:      4,  Func. Count:      42,  Neg. LLF: 3539.7307117418654
Iteration:      5,  Func. Count:      52,  Neg. LLF: 3540.0095053270798
Iteration:      6,  Func. Count:      62,  Neg. LLF: 3749.556952768361
Iteration:      7,  Func. Count:      73,  Neg. LLF: 3516.347169963872
Iteration:      8,  Func. Count:      82,  Neg. LLF: 3572.8776711962496
Iteration:      9,  Func. Count:      92,  Neg. LLF: 3537.274144881866
Iteration:     10,  Func. Count:     102,  Neg. LLF: 3515.5567694468646
Iteration:     11,  Func. Count:     111,  Neg. LLF: 3515.9298321309407
Iteration:     12,  Func. Count:     121,  Neg. LLF: 3515.8155100070044
Iteration:     13,  Func. Count:     131,  Neg. LLF: 3515.5161652084143
Iteration:     14,  Func. Count:     140,  Neg. LLF: 3515.8583012265426
Iteration:     15,  Func. Count:     151,  Neg. LLF: 3515.5141907995394
Iteration:     16,  Func. Count:     160,  Neg. LLF: 3515.5139460886667
Iteration:     17,  Func. Count:     169,  Neg. LLF: 3515.513904740098
Iteration:     18,  Func. Count:     178,  Neg. LLF: 3515.5139038331067
```

```
Optimization terminated successfully (Exit mode 0)
Current function value: 3515.5139038331067
Iterations: 18
Function evaluations: 178
Gradient evaluations: 18
```

با توجه به پی مقدارهای بدست آمده برای آلفا یک تا سه و بتا یک تا سه از ضرایب دو و دو برای برازش استفاده میکنیم

```
In [163... model_fit.summary()
```

Out[163...

Constant Mean - GARCH Model Results

Dep. Variable:	DIS	R-squared:	0.000
Mean Model:	Constant Mean	Adj. R-squared:	0.000
Vol Model:	GARCH	Log-Likelihood:	-3515.51
Distribution:	Normal	AIC:	7047.03
Method:	Maximum Likelihood	BIC:	7090.81
No. Observations:			1759
Date:	Sat, Feb 01 2025	Df Residuals:	1758
Time:	11:12:30	Df Model:	1

Mean Model

	coef	std err	t	P> t 	95.0% Conf. Int.
mu	8.6537e-03	4.193e-02	0.206	0.836	[-7.353e-02,9.084e-02]

Volatility Model

	coef	std err	t	P> t 	95.0% Conf. Int.
omega	0.3436	0.294	1.170	0.242	[-0.232, 0.919]
alpha[1]	0.0870	7.331e-02	1.187	0.235	[-5.667e-02, 0.231]
alpha[2]	0.0631	0.103	0.611	0.542	[-0.139, 0.266]
alpha[3]	4.9727e-04	3.969e-02	1.253e-02	0.990	[-7.730e-02,7.829e-02]
beta[1]	0.0515	0.143	0.361	0.718	[-0.228, 0.331]
beta[2]	0.0000	0.156	0.000	1.000	[-0.305, 0.305]
beta[3]	0.7066	9.502e-02	7.437	1.033e-13	[0.520, 0.893]

Covariance estimator: robust

```
In [164... model = arch_model(returns, p=2, q=2)
```

```
In [165... model_fit = model.fit()
```

```
Iteration:      1,  Func. Count:      8,  Neg. LLF: 10610.572947415372
Iteration:      2,  Func. Count:     18,  Neg. LLF: 1578389.5385628422
Iteration:      3,  Func. Count:     27,  Neg. LLF: 3611.905552227384
Iteration:      4,  Func. Count:     35,  Neg. LLF: 3571.7830685568724
Iteration:      5,  Func. Count:     43,  Neg. LLF: 3544.263041442879
Iteration:      6,  Func. Count:     51,  Neg. LLF: 4979.594049937861
Iteration:      7,  Func. Count:     59,  Neg. LLF: 3560.387694594904
Iteration:      8,  Func. Count:     67,  Neg. LLF: 3536.207208596763
Iteration:      9,  Func. Count:     75,  Neg. LLF: 3519.2656067338157
Iteration:     10,  Func. Count:     83,  Neg. LLF: 3518.4374593039465
Iteration:     11,  Func. Count:     90,  Neg. LLF: 3518.3202444171784
Iteration:     12,  Func. Count:     97,  Neg. LLF: 3518.312483530952
Iteration:     13,  Func. Count:    104,  Neg. LLF: 3518.307877058359
Iteration:     14,  Func. Count:    111,  Neg. LLF: 3518.30557607813
Iteration:     15,  Func. Count:    118,  Neg. LLF: 3518.3055319024697
Iteration:     16,  Func. Count:    125,  Neg. LLF: 3518.3055307615377
Iteration:     17,  Func. Count:    131,  Neg. LLF: 3518.30553076003
```

```
Optimization terminated successfully (Exit mode 0)
    Current function value: 3518.3055307615377
    Iterations: 17
    Function evaluations: 131
    Gradient evaluations: 17
```

```
In [166... model_fit.summary()
```

Out[166...

Constant Mean - GARCH Model Results

Dep. Variable:	DIS	R-squared:	0.000
Mean Model:	Constant Mean	Adj. R-squared:	0.000
Vol Model:	GARCH	Log-Likelihood:	-3518.31
Distribution:	Normal	AIC:	7048.61
Method:	Maximum Likelihood	BIC:	7081.45
No. Observations:			1759
Date:	Sat, Feb 01 2025	Df Residuals:	1758
Time:	11:12:30	Df Model:	1

Mean Model

	coef	std err	t	P> t 	95.0% Conf. Int.
mu	4.2195e-03	0.133	3.168e-02	0.975	[-0.257, 0.265]

Volatility Model

	coef	std err	t	P> t 	95.0% Conf. Int.
omega	0.1697	2.808	6.043e-02	0.952	[-5.335, 5.674]
alpha[1]	0.0851	0.526	0.162	0.871	[-0.945, 1.116]
alpha[2]	3.3041e-13	1.962	1.684e-13	1.000	[-3.846, 3.846]
beta[1]	0.4670	19.304	2.419e-02	0.981	[-37.369, 38.303]
beta[2]	0.4039	17.160	2.354e-02	0.981	[-33.229, 34.037]

Covariance estimator: robust

حال برای پیش بینی دقیقتر مدل از تکنیک رولینگ استفاده میکنیم


```
In [167... rolling_predictions = []
test_size = 365

for i in range(test_size):
    train = returns[:-(test_size-i)]
    model = arch_model(train, p=2, q=2)
    model_fit = model.fit(dis='off')
    pred = model_fit.forecast(horizon=1)
    rolling_predictions.append(np.sqrt(pred.variance.values[-1,:][0]))
```

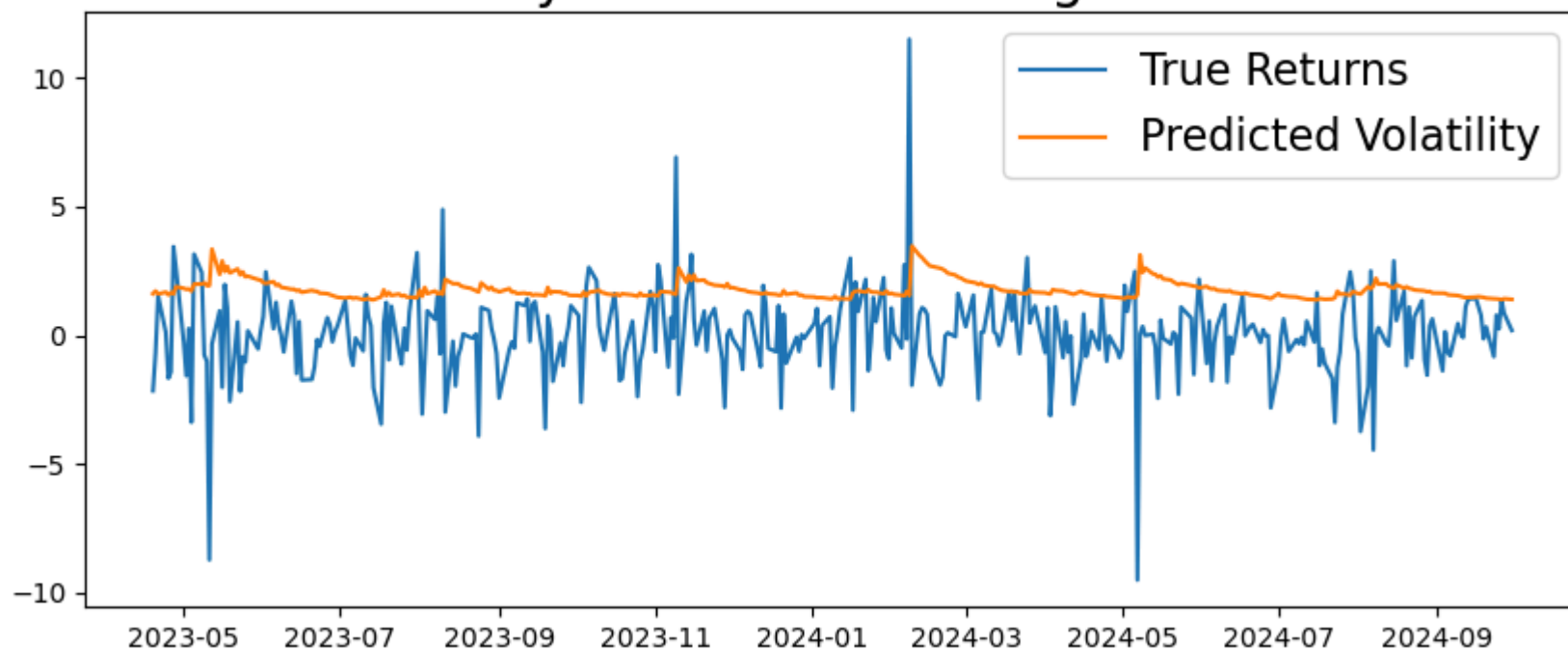
```
In [168... rolling_predictions = pd.Series(rolling_predictions, index=returns.index[-365:])
```

مدل آموزش یافته بازده و مقدار اصلی را رسم میکنم

```
In [169... plt.figure(figsize=(10,4))
true, = plt.plot(returns[-365:])
preds, = plt.plot(rolling_predictions)
plt.title('Volatility Prediction - Rolling Forecast', fontsize=20)
plt.legend(['True Returns', 'Predicted Volatility'], fontsize=16)
```

```
Out[169... <matplotlib.legend.Legend at 0x24ecc0c5760>
```

Volatility Prediction - Rolling Forecast



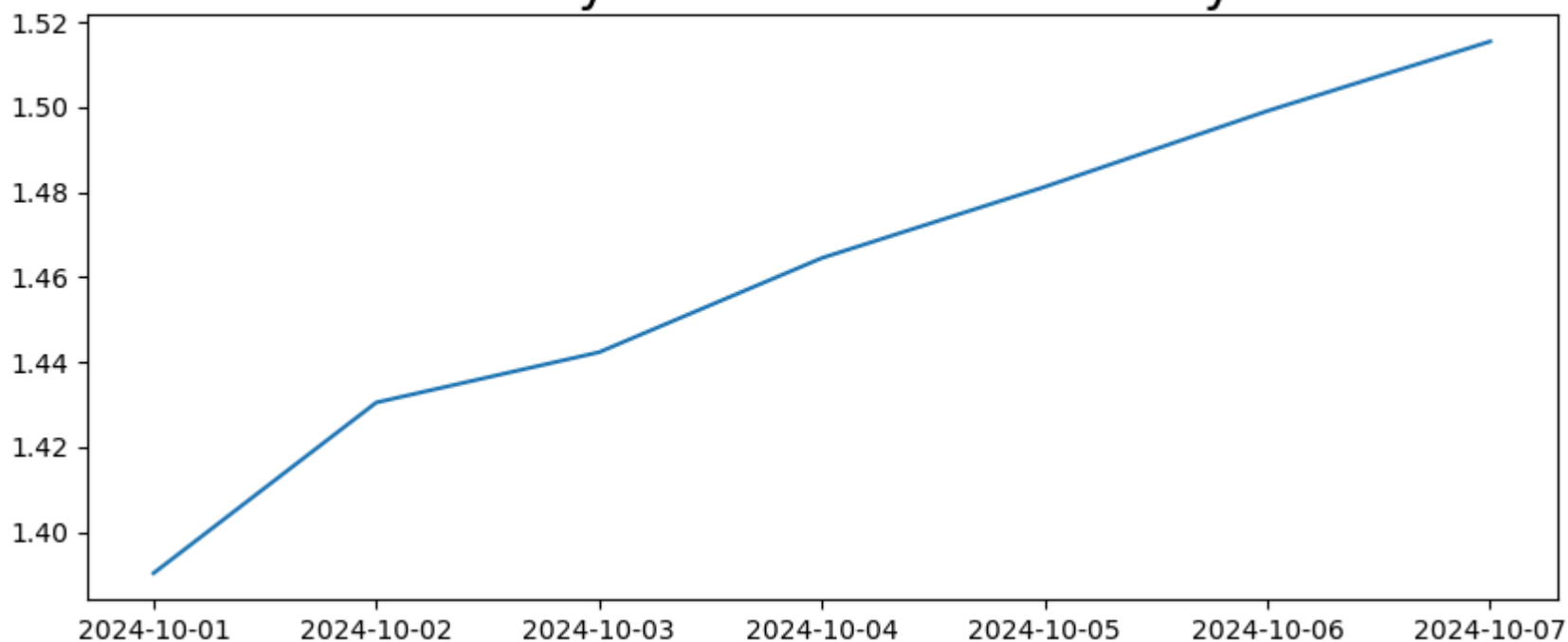
حال برای هفت روز آینده بازده سهم را پیش بینی میکنیم

```
In [170...] pred = model_fit.forecast(horizon=7)
future_dates = [returns.index[-1] + timedelta(days=i) for i in range(1,8)]
pred = pd.Series(np.sqrt(pred.variance.values[-1,:]), index=future_dates)
```

```
In [171...] plt.figure(figsize=(10,4))
plt.plot(pred)
plt.title('Volatility Prediction - Next 7 Days', fontsize=20)
```

```
Out[171...] Text(0.5, 1.0, 'Volatility Prediction - Next 7 Days')
```

Volatility Prediction - Next 7 Days

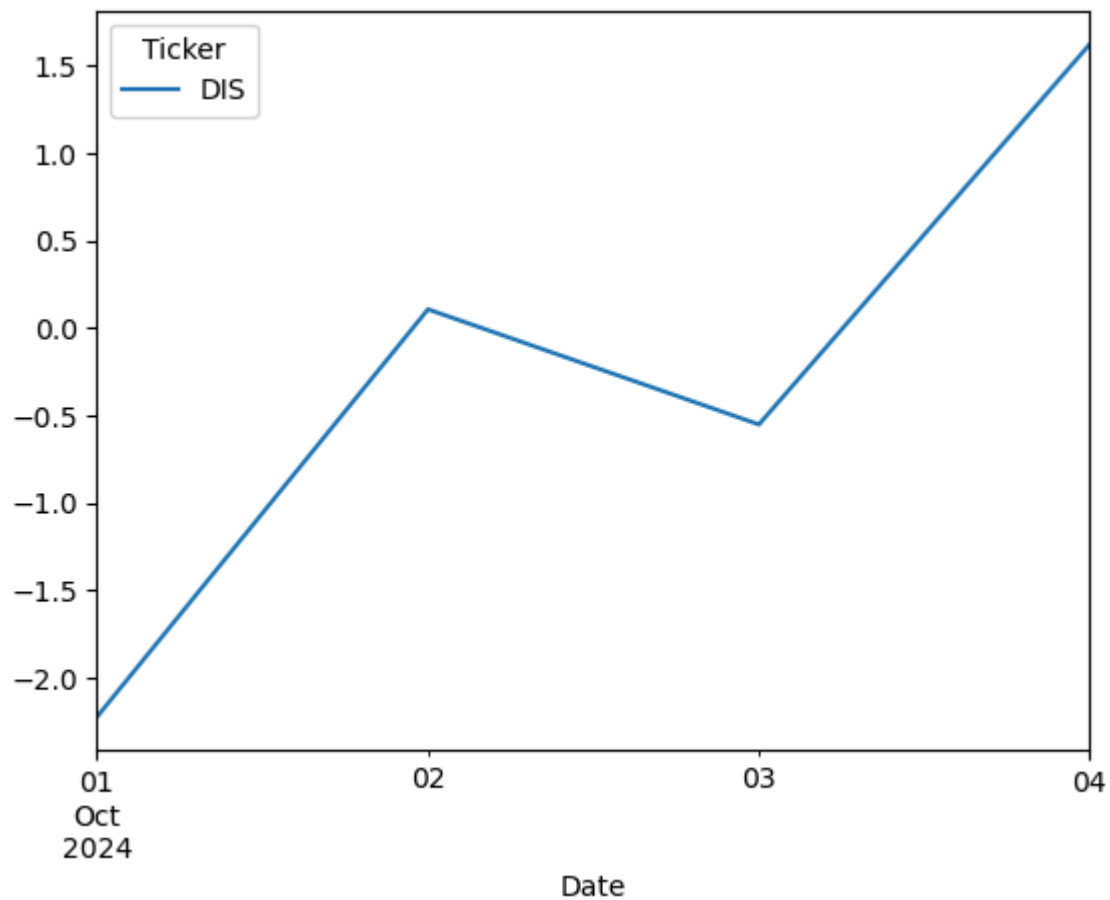


مقدار پیش بینی شده را با مقدار بازده اصلی مقایسه میکنیم

```
In [172... dis = yf.download('DIS', start = '2024-9-29', end = '2024-10-7')
returns0 = 100 * dis.Close.pct_change().dropna()
returns0.plot()
```

```
[*****100%*****] 1 of 1 completed
```

```
Out[172... <Axes: xlabel='Date'>
```



تمام مراحل بالا را برای سهام اس اند پی 500 انجام می‌دهیم

```
In [174... start = datetime(2017, 10, 1)
end = datetime(2024, 10, 1)
```

```
In [175... spy = yf.download('SPY', start = '2017-10-1', end = '2024-10-1')
```

```
[*****100%*****] 1 of 1 completed
```

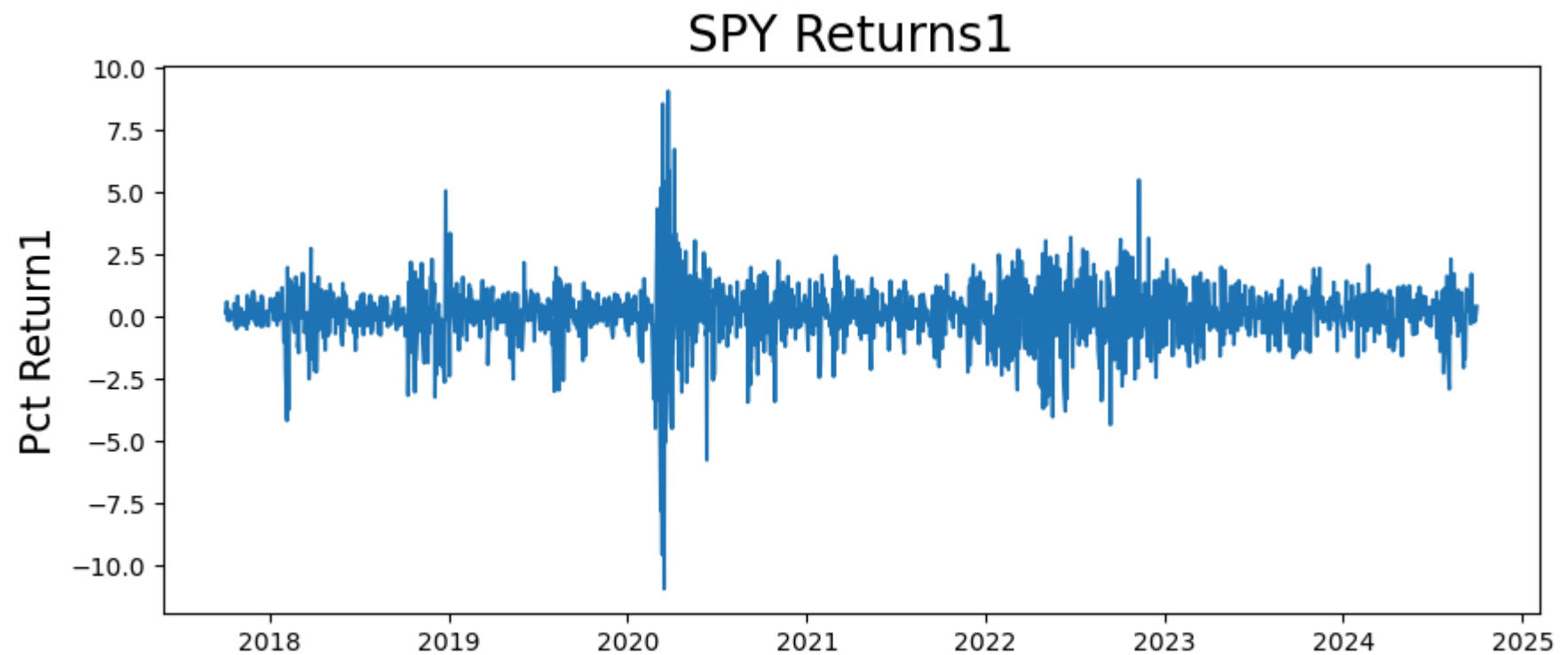
```
In [176... returns1 = 100 * spy.Close.pct_change().dropna()
```

```
In [177... perform_adf_test(returns1)
```

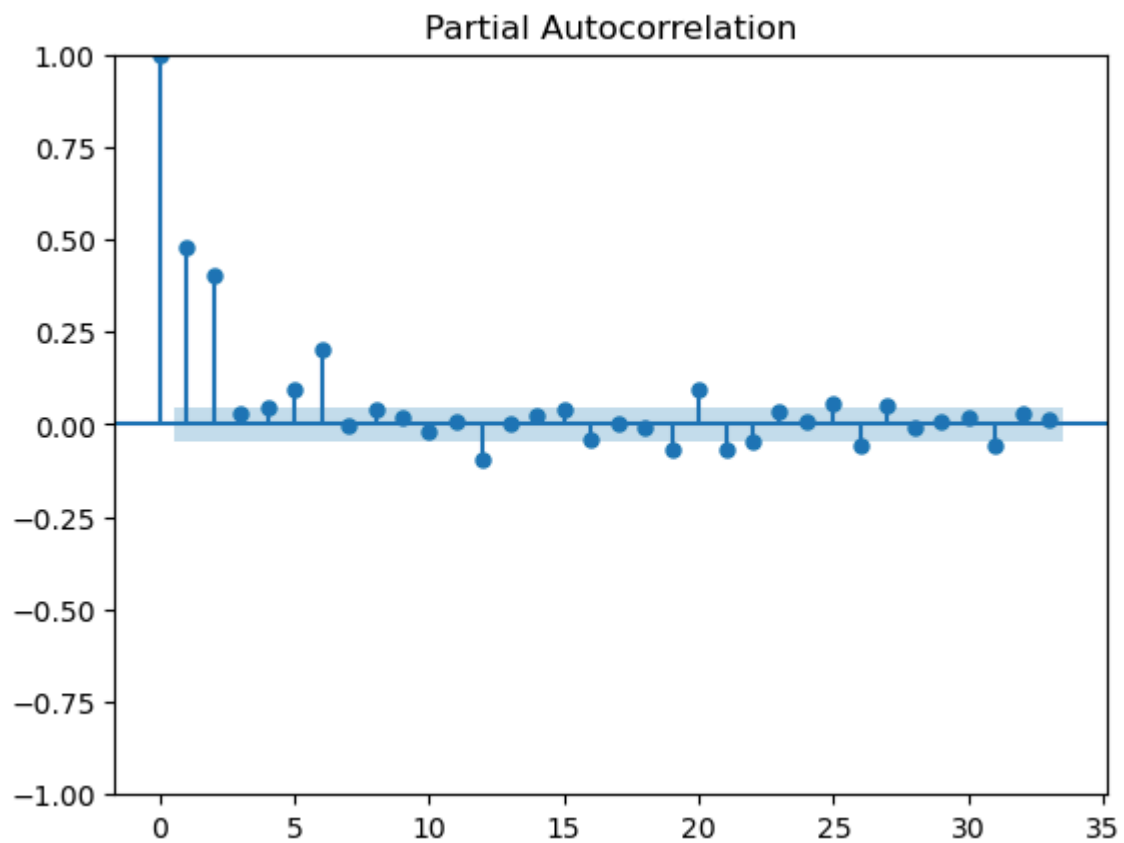
ADF Statistic: -12.833366
p-value: 0.000000

```
In [178... plt.figure(figsize=(10,4))  
plt.plot(returns1)  
plt.ylabel('Pct Return1', fontsize=16)  
plt.title('SPY Returns1', fontsize=20)
```

```
Out[178... Text(0.5, 1.0, 'SPY Returns1')
```



```
In [179... plot_pacf(returns1**2)  
plt.show()
```



```
In [180...] model = arch_model(returns1, p=2, q=2)
```

```
In [181...] model_fit = model.fit()
```

```
Iteration:      1,  Func. Count:      8,  Neg. LLF: 11467.837127551458
Iteration:      2,  Func. Count:     20,  Neg. LLF: 678597.335641531
Iteration:      3,  Func. Count:     29,  Neg. LLF: 3103.082821871381
Iteration:      4,  Func. Count:     38,  Neg. LLF: 4055.8707897456534
Iteration:      5,  Func. Count:     47,  Neg. LLF: 2835.4304784793076
Iteration:      6,  Func. Count:     56,  Neg. LLF: 2395.1645778939837
Iteration:      7,  Func. Count:     65,  Neg. LLF: 2385.3119762096167
Iteration:      8,  Func. Count:     73,  Neg. LLF: 2385.202203498992
Iteration:      9,  Func. Count:     81,  Neg. LLF: 2385.219172651314
Iteration:     10,  Func. Count:     89,  Neg. LLF: 2385.0417942395006
Iteration:     11,  Func. Count:     96,  Neg. LLF: 2385.7761512941493
Iteration:     12,  Func. Count:    104,  Neg. LLF: 2385.011259111091
Iteration:     13,  Func. Count:    111,  Neg. LLF: 2385.0099234756617
Iteration:     14,  Func. Count:    118,  Neg. LLF: 2385.009792079388
Iteration:     15,  Func. Count:    125,  Neg. LLF: 2385.0097850177426
Iteration:     16,  Func. Count:    132,  Neg. LLF: 2385.009784258321
```

Optimization terminated successfully (Exit mode 0)

Current function value: 2385.009784258321

Iterations: 16

Function evaluations: 132

Gradient evaluations: 16

In [182... `model_fit.summary()`

Out[182...

Constant Mean - GARCH Model Results

Dep. Variable:	SPY	R-squared:	0.000
Mean Model:	Constant Mean	Adj. R-squared:	0.000
Vol Model:	GARCH	Log-Likelihood:	-2385.01
Distribution:	Normal	AIC:	4782.02
Method:	Maximum Likelihood	BIC:	4814.85
No. Observations:			1759
Date:	Sat, Feb 01 2025	Df Residuals:	1758
Time:	11:12:57	Df Model:	1

Mean Model

	coef	std err	t	P> t 	95.0% Conf. Int.
mu	0.1077	1.872e-02	5.752	8.835e-09	[7.097e-02, 0.144]

Volatility Model

	coef	std err	t	P> t 	95.0% Conf. Int.
omega	0.0605	1.951e-02	3.101	1.931e-03	[2.225e-02, 9.872e-02]
alpha[1]	0.1705	4.070e-02	4.191	2.782e-05	[9.078e-02, 0.250]
alpha[2]	0.1603	4.378e-02	3.661	2.516e-04	[7.446e-02, 0.246]
beta[1]	0.0400	0.127	0.316	0.752	[-0.208, 0.288]
beta[2]	0.5990	0.104	5.767	8.085e-09	[0.395, 0.803]

Covariance estimator: robust


```
In [183... rolling_predictions = []
test_size = 365

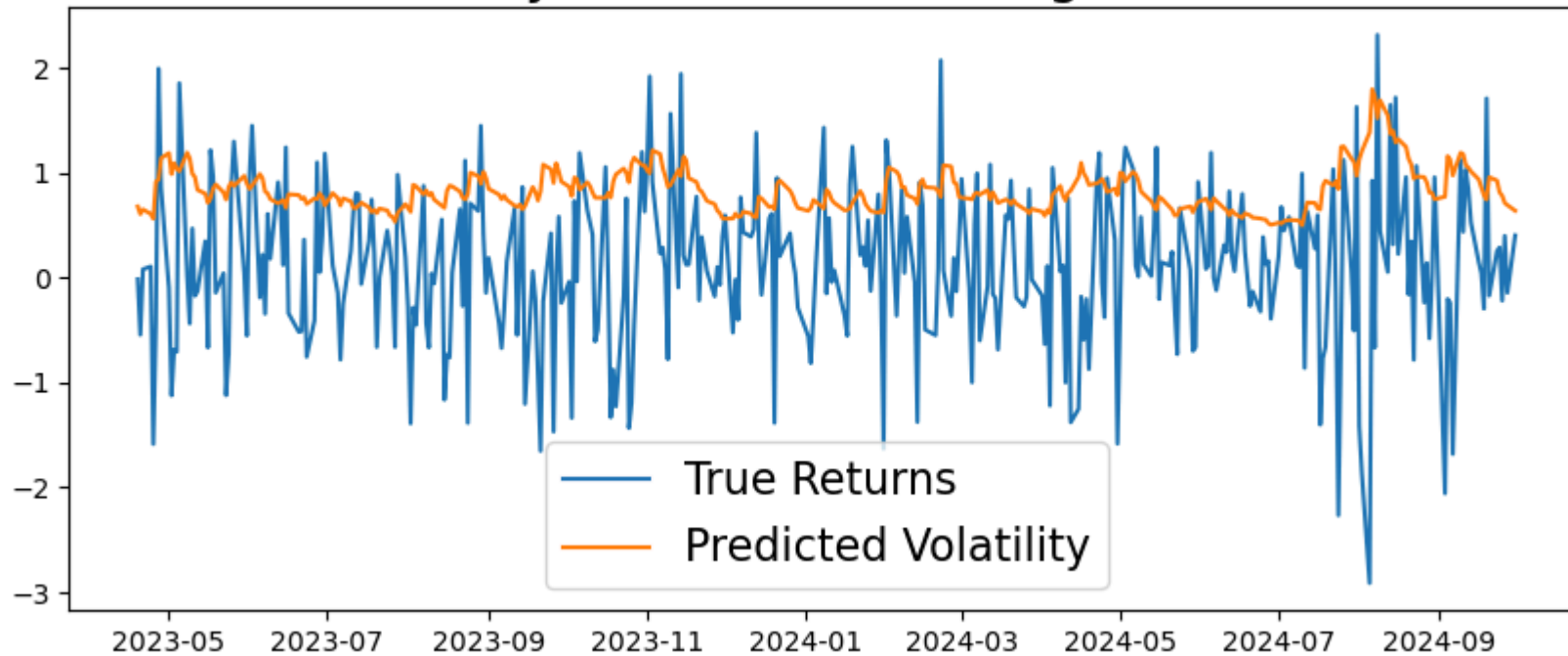
for i in range(test_size):
    train = returns1[:-(test_size-i)]
    model = arch_model(train, p=2, q=2)
    model_fit = model.fit(dispatch='off')
    pred = model_fit.forecast(horizon=1)
    rolling_predictions.append(np.sqrt(pred.variance.values[-1,:][0]))
```

```
In [184... rolling_predictions = pd.Series(rolling_predictions, index=returns1.index[-365:])
```

```
In [185... plt.figure(figsize=(10,4))
true, = plt.plot(returns1[-365:])
preds, = plt.plot(rolling_predictions)
plt.title('Volatility Prediction - Rolling Forecast', fontsize=20)
plt.legend(['True Returns', 'Predicted Volatility'], fontsize=16)
```

```
Out[185... <matplotlib.legend.Legend at 0x24ecc655ee0>
```

Volatility Prediction - Rolling Forecast



```
In [186... train = returns1
model = arch_model(train, p=2, q=2)

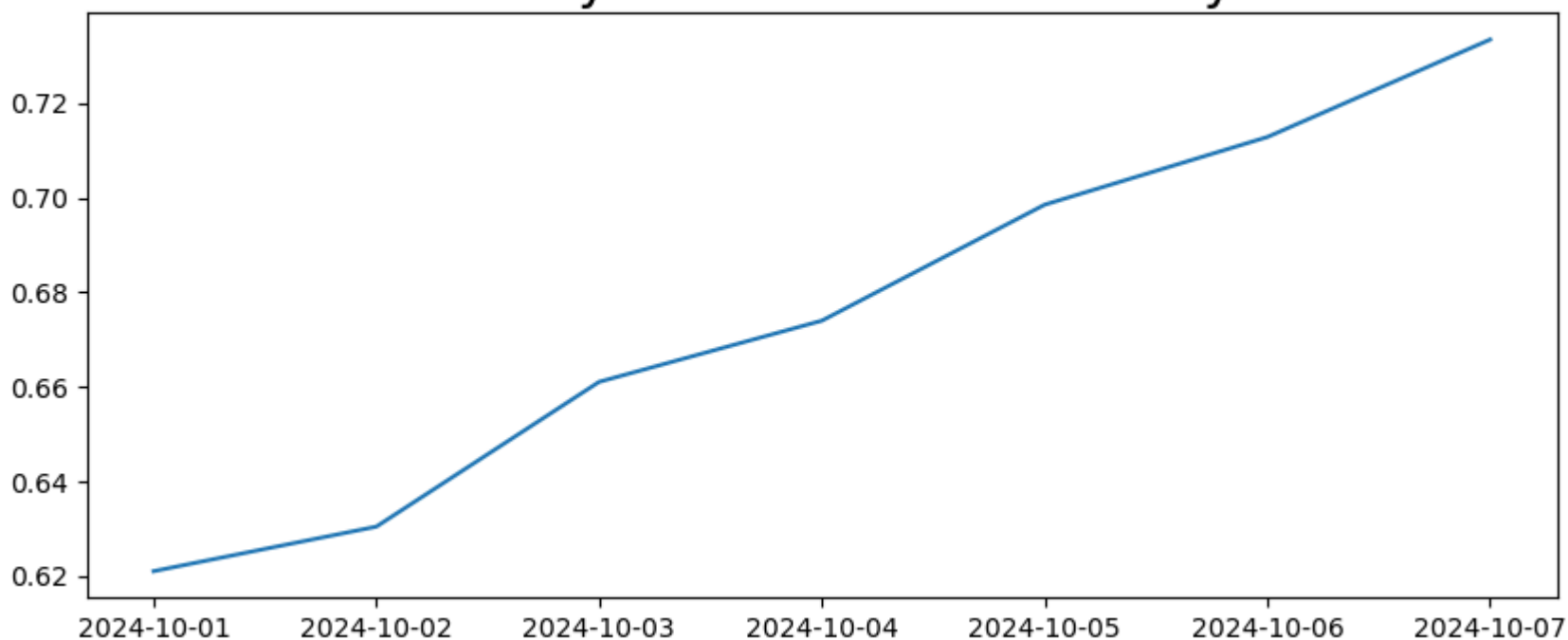
model_fit = model.fit(dispatch='off')
```

```
In [187... pred = model_fit.forecast(horizon=7)
future_dates = [returns1.index[-1] + timedelta(days=i) for i in range(1,8)]
pred = pd.Series(np.sqrt(pred.variance.values[-1,:]), index=future_dates)
```

```
In [188... plt.figure(figsize=(10,4))
plt.plot(pred)
plt.title('Volatility Prediction - Next 7 Days', fontsize=20)
```

```
Out[188... Text(0.5, 1.0, 'Volatility Prediction - Next 7 Days')
```

Volatility Prediction - Next 7 Days



```
In [189... spy1 = yf.download('SPY', start = '2024-9-29', end = '2024-10-07')
```

```
[*****100%*****] 1 of 1 completed
```

```
In [190... returns2 = 100 * spy1.Close.pct_change().dropna()  
returns2.plot()
```

```
Out[190... <Axes: xlabel='Date'>
```

