Volatility Forecasting

S&P 500 ETF SPY

Yuman Lin, Algo Depth Project

Asset Volatility

- Important feature in quant finance
- Useful: portfolio optimization, option pricing, risk management
- Some popular methods for volatility prediction:
- (1) History volatility models, such as Exponential Weighted Moving Average;
- (2) Time series models: ARMA, GARCH family of models;
- (3) Machine learning models: RNN, neutral network, decision tree, etc.

Project Description

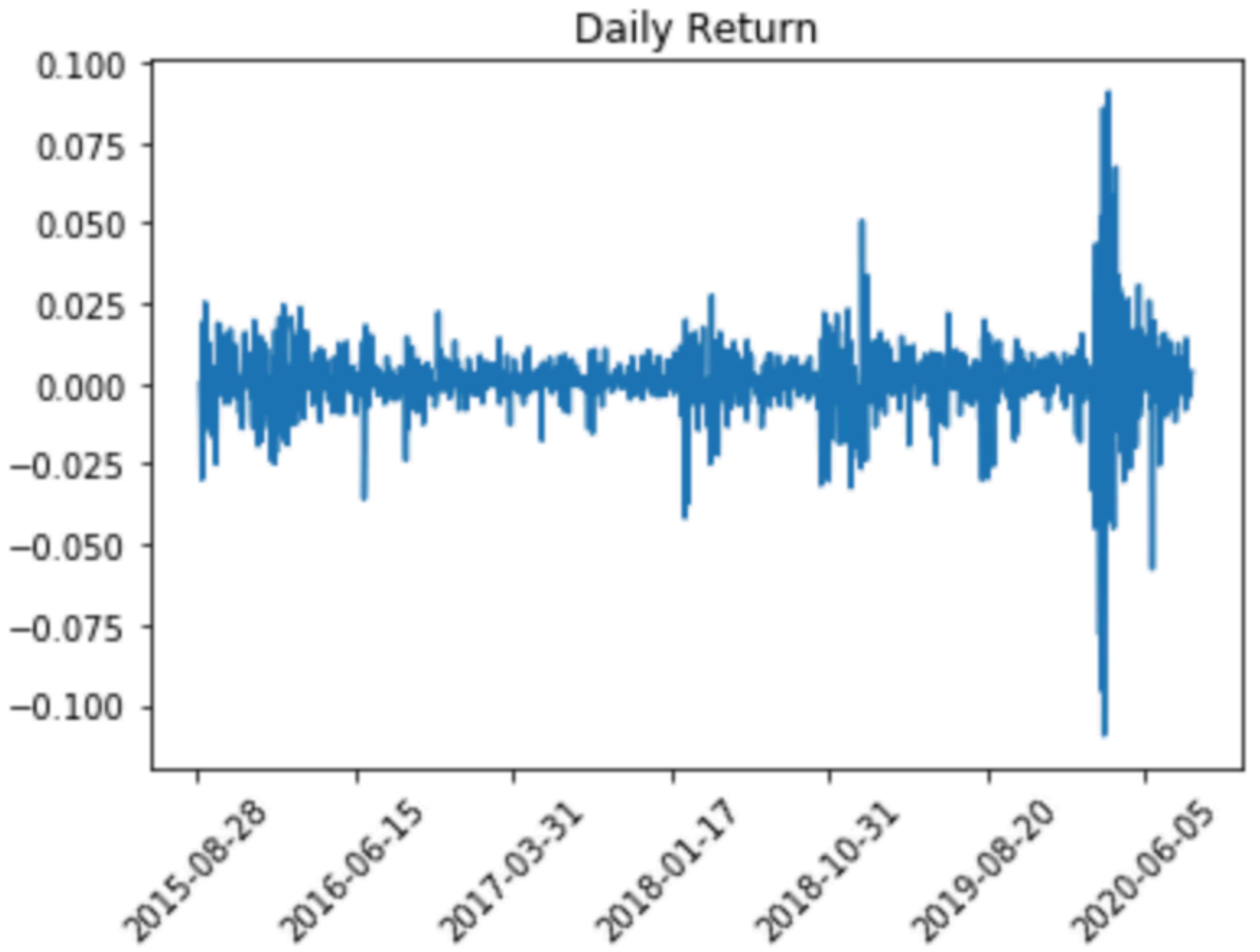
S&P 500 ETF SPY

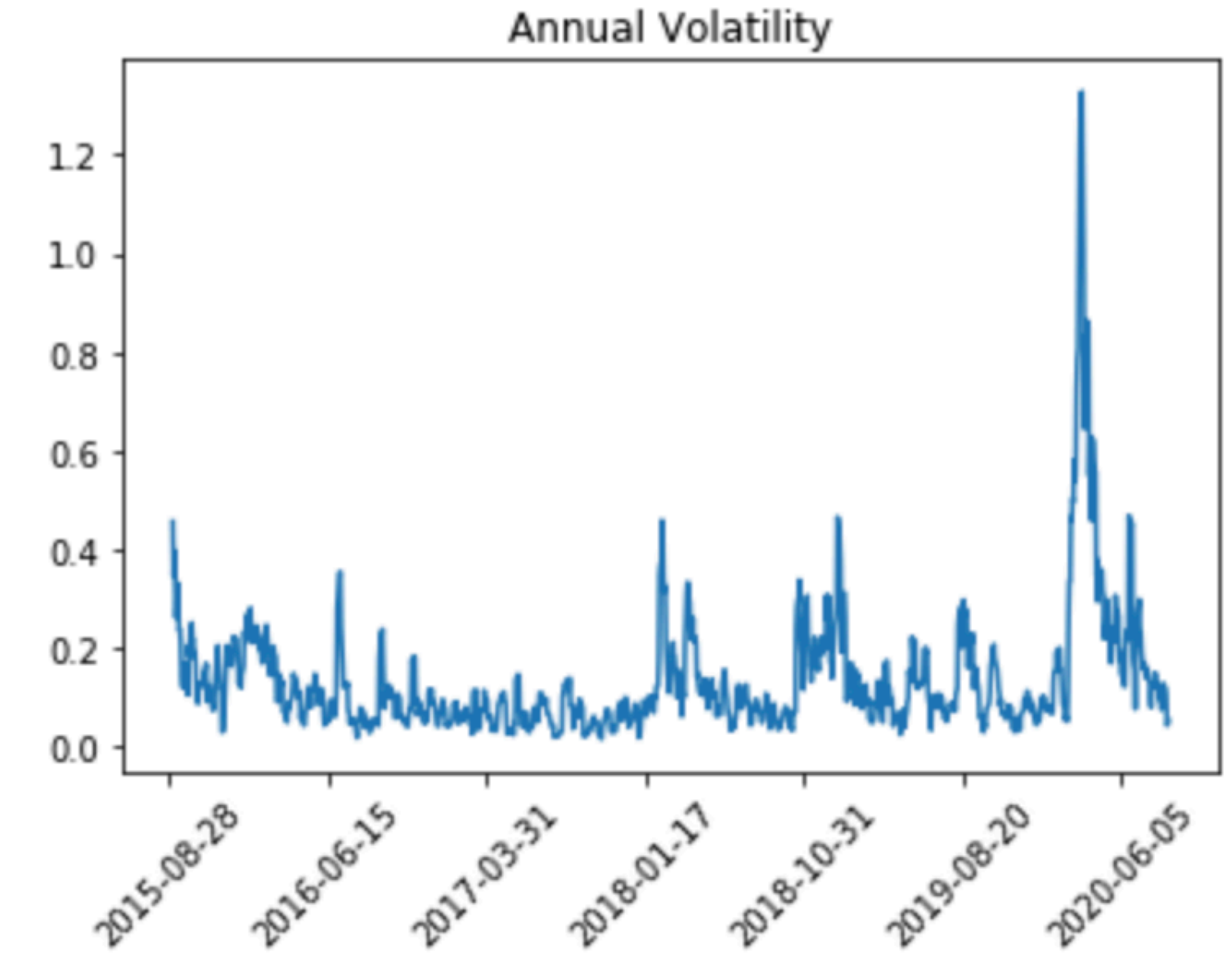
- Data: time series data of S&P 500 ETF SPY, from 2015-08-21 to 2020-08-21
- Open, High, Low, Close, Adjust Close, Volume

$$\text{daily return} = \frac{P_t - P_{t-1}}{P_{t-1}}$$

$$\text{annual volatility} = \sqrt{52} * \sqrt{(r_t^2 + r_{t-1}^2 + r_{t-2}^2 + r_{t-3}^2 + r_{t-4}^2}$$

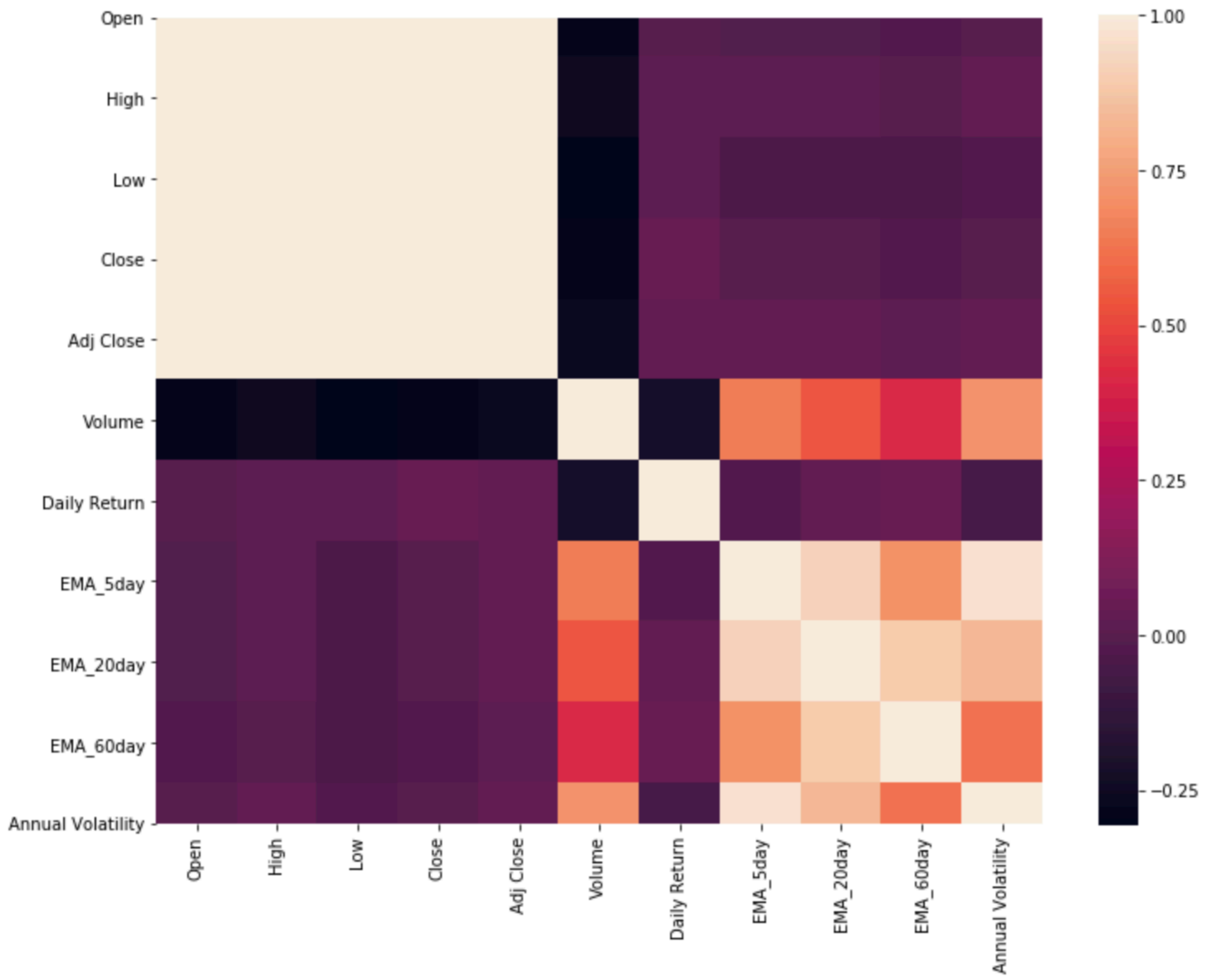
-
-





Feature Engineering

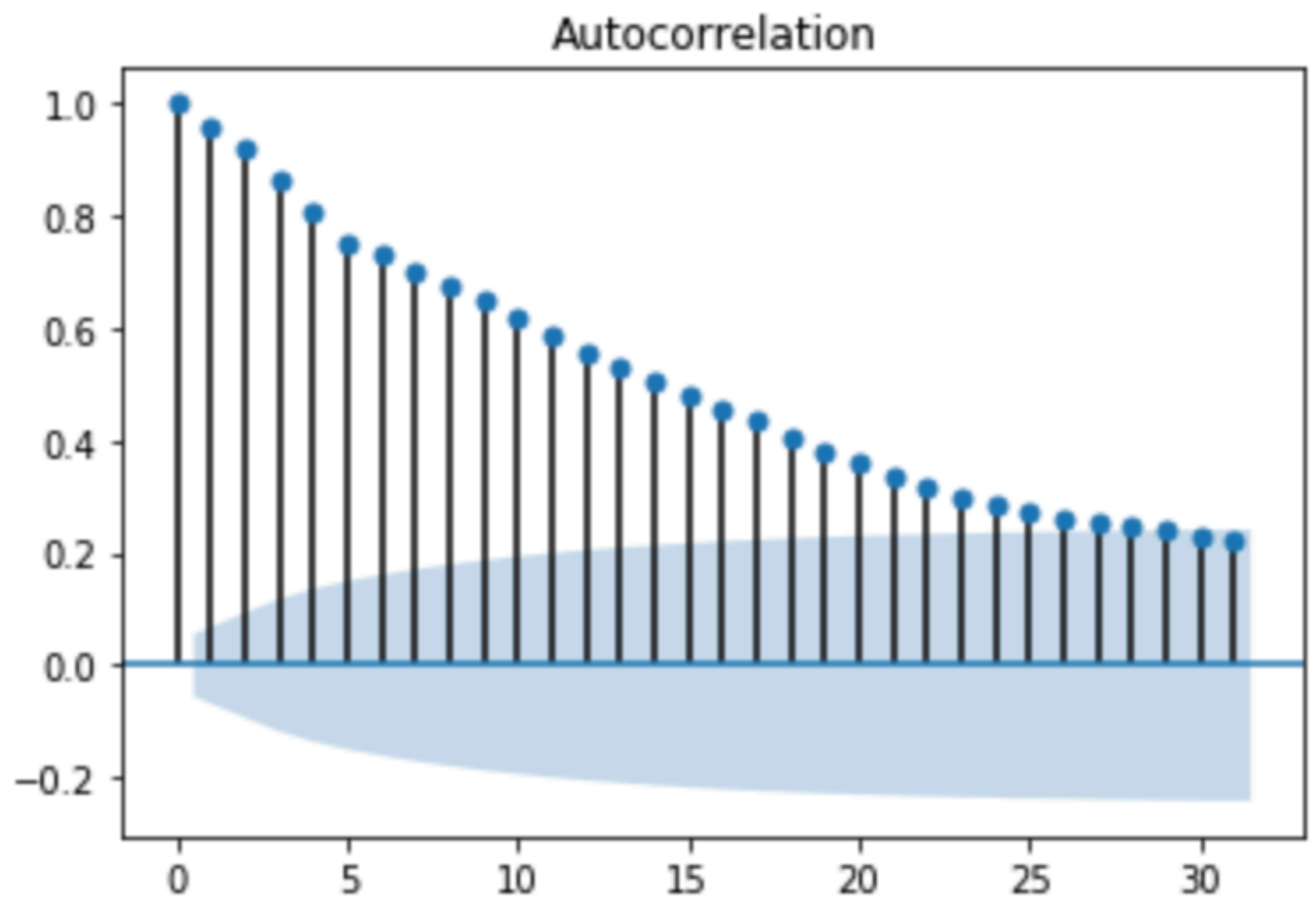
- Moving average smoothing: exponential weighted functions for annual volatility column, span = 5 (weekly), span = 20 (monthly) and span = 60 (seasonly)
- Close correlation between annual volatility and volume: single moving average for volume, rolling window = 5 (weekly)
- Lag feature: lag shift of annual volatility, lag = 1 and lag = 5
- Null values: replace with the most recent value or mean value

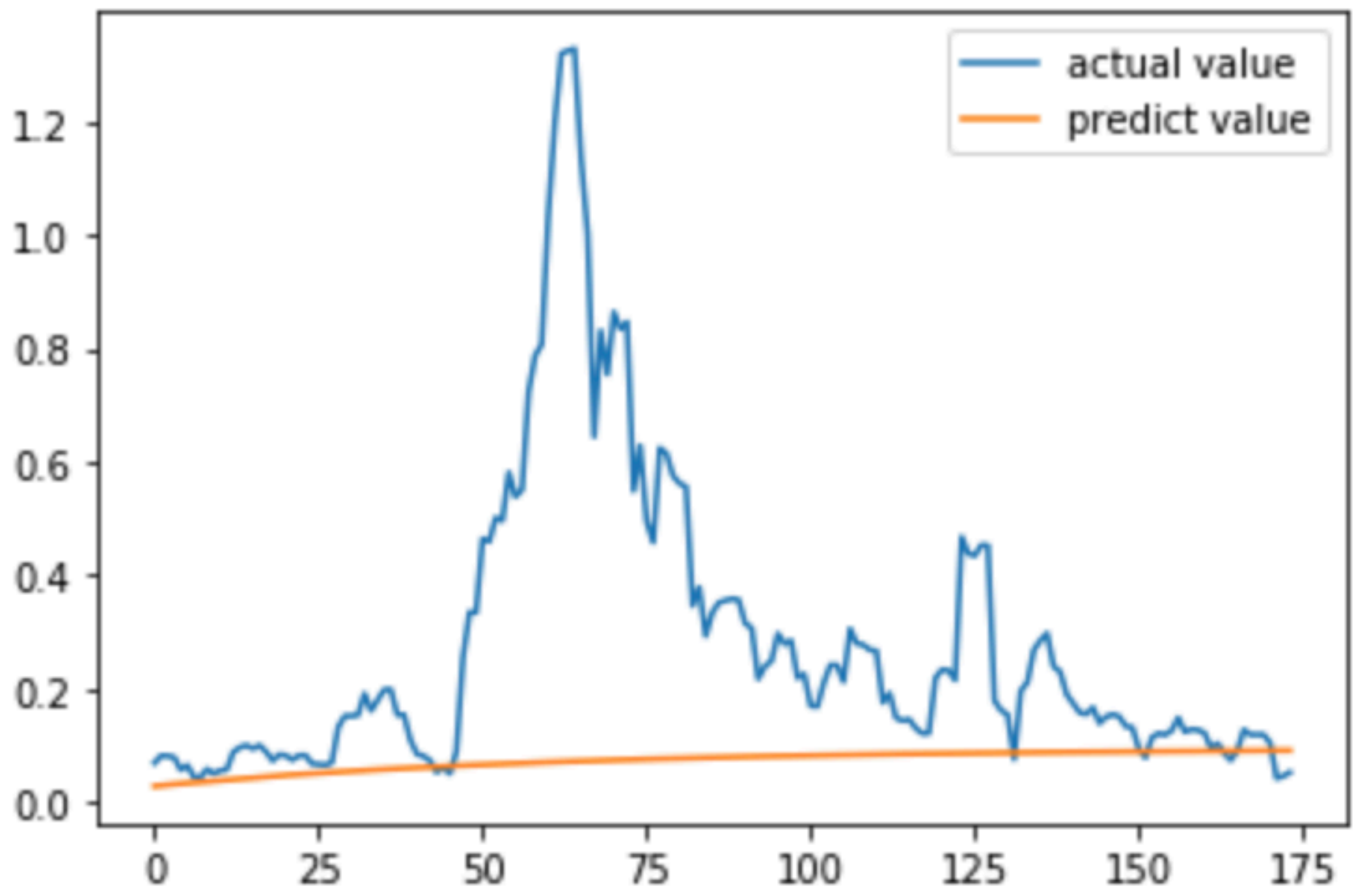


	Date	Open	High	Low	Close	Adj Close	Volume	Daily Return	EMA_5day	EMA_20day	EMA_60day	SMA_5	Lag_Volatility1	Lag_Volatility5	Annual Volatility
5	2015-08-28	198.500000	199.839996	197.919998	199.279999	180.415207	160414400	0.000050	0.455920	0.455920	0.455920	178414140.0	0.136702	0.136888	0.455920
6	2015-08-31	198.110001	199.130005	197.009995	197.669998	178.957611	163298800	-0.008079	0.418965	0.445362	0.452285	178414140.0	0.455920	0.136888	0.345054
7	2015-09-01	193.119995	194.770004	190.729996	191.770004	173.616119	256000400	-0.029848	0.411886	0.440825	0.450497	178414140.0	0.345054	0.136888	0.397728
8	2015-09-02	194.619995	195.460007	192.419998	195.410004	176.911545	160269300	0.018981	0.380143	0.428999	0.446108	178414140.0	0.397728	0.136888	0.316655
9	2015-09-03	196.259995	198.050003	194.960007	195.550003	177.038300	152087800	0.000716	0.340659	0.413065	0.440062	178414140.0	0.316655	0.136888	0.261691

Time series model: ARCH

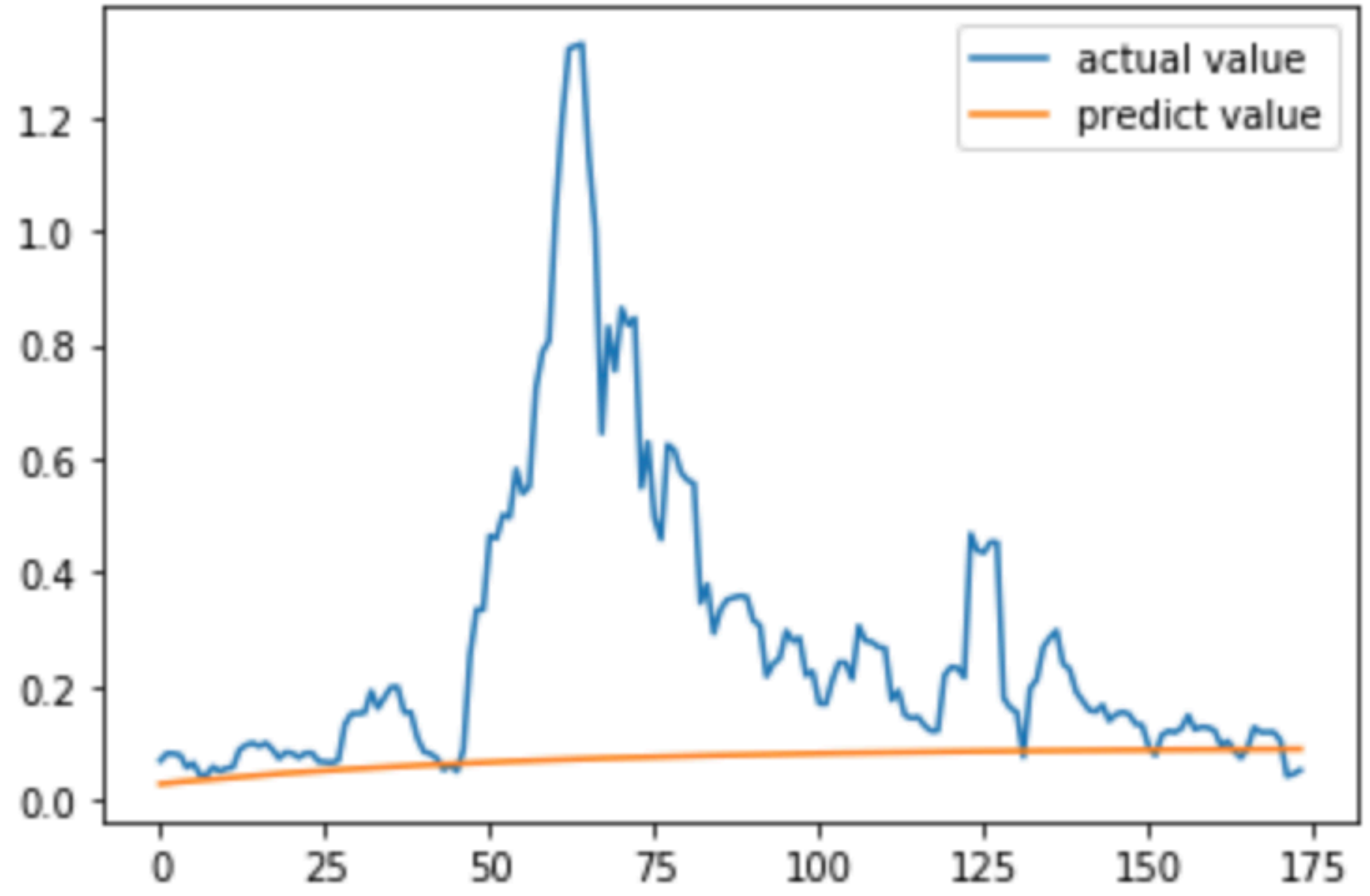
- ACF plot for autocorrelation: p = 25
- Dataframe shape: 1255*15
- Separate train set and test set (200)
- ARCH model: zero mean, p = 25
- AIC = -1852.66, BIC = -1723.03
- Test set prediction plot (second figure)
- Train all the data and predict the volatility of next week: 0.0457





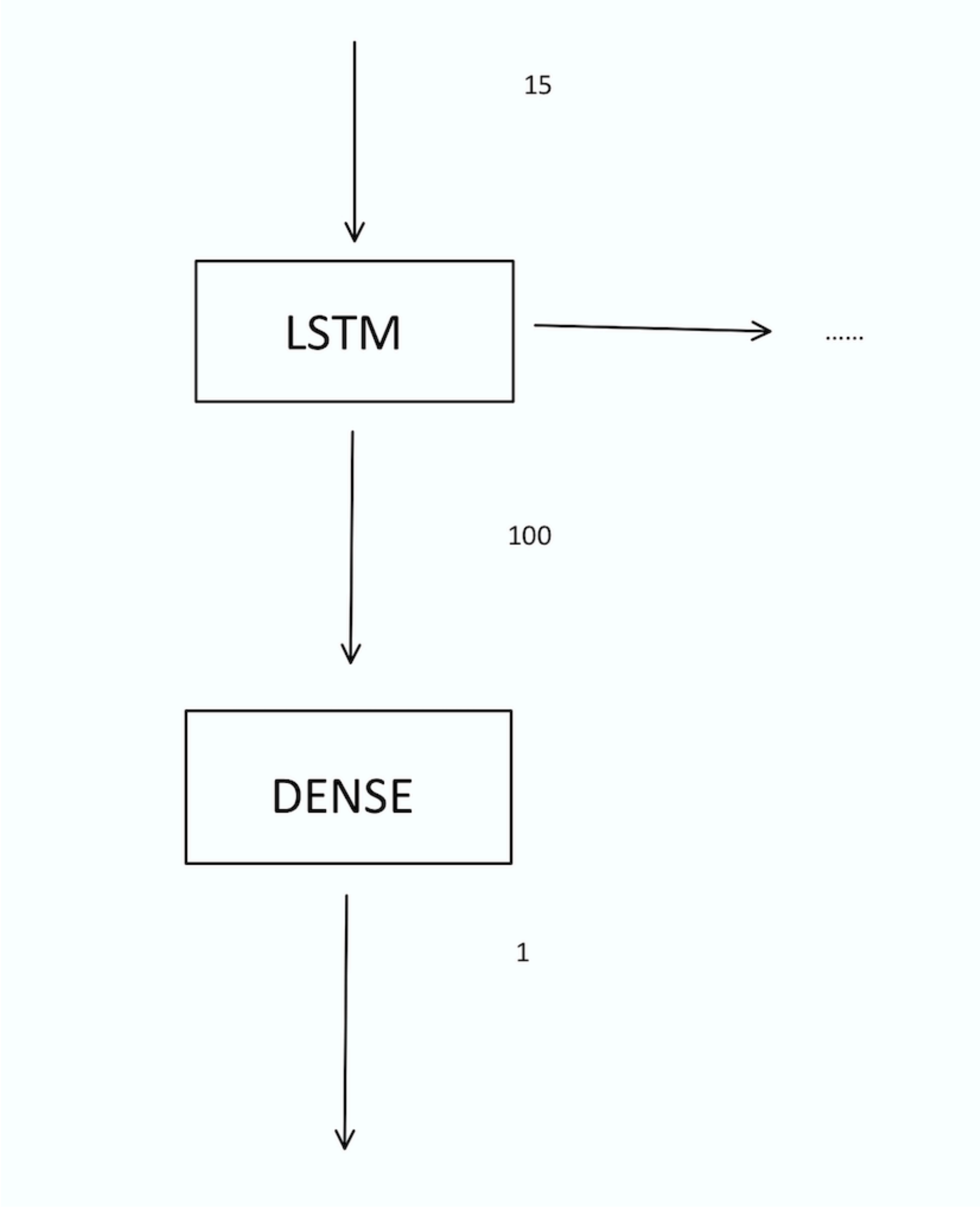
Time series model: GARCH

- GARCH model: p=20, q = 20
- AIC = -1822.47, BIC =-1618.06
- Compared with ARCH model, GARCH is worse.
- Test set prediction plot
- Train all the data and predict the volatility of next week: 0.0563



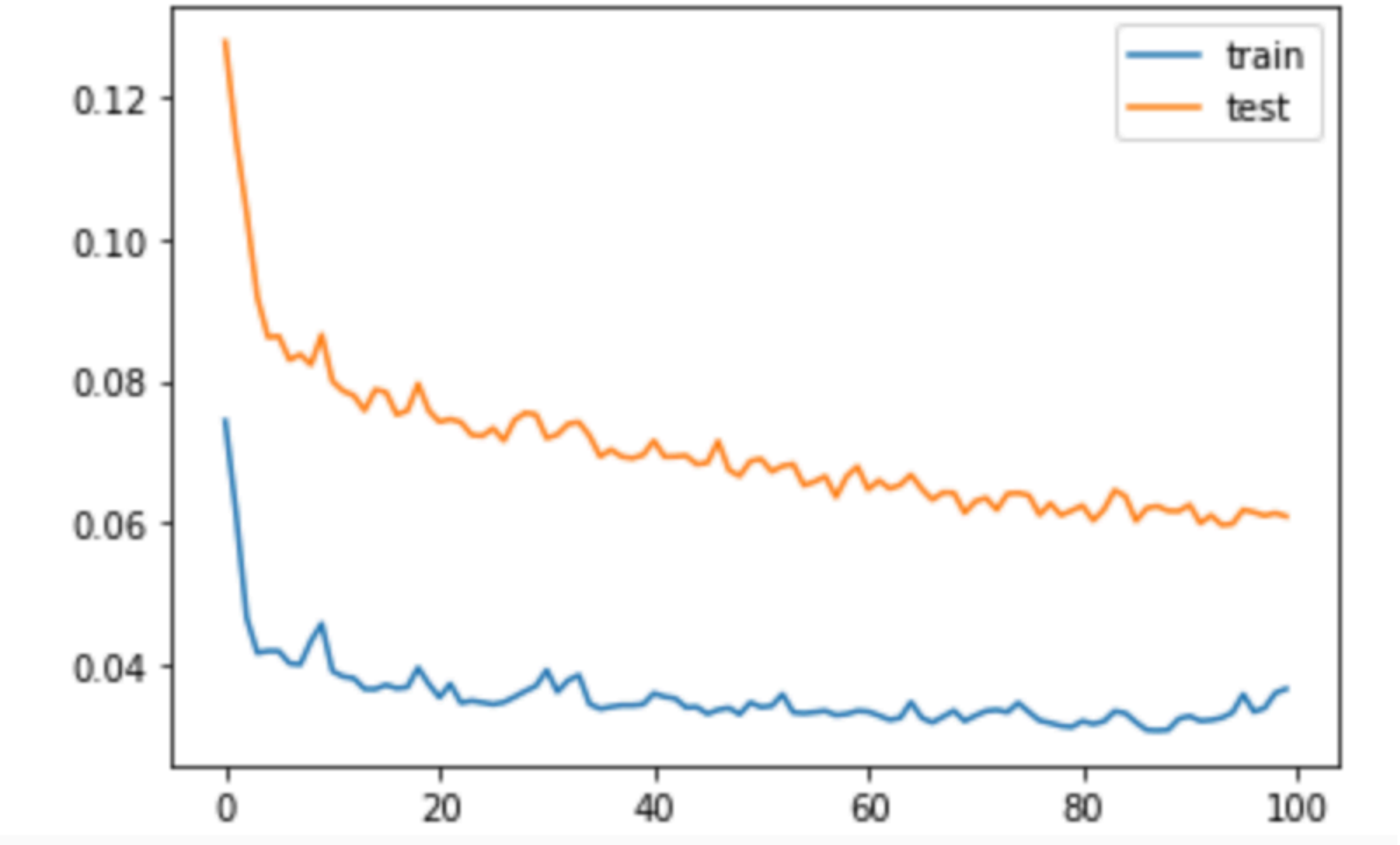
Deep Learning model: LSTM

- Long short-term memory (LSTM): artificial recurrent neural network (RNN), solve the problem of vanishing gradient
- Use (t-5) data to predict t data (weekly volatility)
- (1) Normalize features with MinMaxScaler
- (2) Convert series to supervised learning
- (3) Separate train set and test set: 1000 and 250
- (4) Layers: inputs - 100 - 1 (output)
- (5) Loss: MAE, optimizer: Adam
- (6) epoch = 100, batch size = 72
- Test RMSE: 0.165
- Retrain the model with all data and predict: 0.09574651



Deep Learning model: LSTM

- Another LSTM model:
- Previous model: one-day data in one input
- Recent model: five-day data in one input
- Previous x_train: 1250 * 15
- Recent x_train: 250 * 5 * 15




```
# design network
model = Sequential()
model.add(LSTM(150, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(32))
model.add(Dense(5))
model.compile(loss='mae', optimizer='adam')
# fit network
history = model.fit(train_X, train_y, epochs=100, batch_size=36, validation_data=(test_X, test_y), verbose=2, shuffle=F
```

Compare Result

ARCH	GARCH	LSTM1	LSTM2	Real value
0.0457	0.0563	0.0954	0.1338	0.1167

Compare Result

- LSTM is much better than traditional time series model.
- Traditional time series model only uses volatility as input.
- LSTM: different ETF prices, lag features, moving average.
- LSTM2 is better than LSTM1.
- Models suffer from overfitting problems.
- Data: COVID-19, more volatility than previous data.

- Code available: https://github.com/yumanlin34/Volatility-Forecasting