

智能金融科技

股价预测

股价预测



数据采集与处理



➤ 数据采集

➤ 数据清洗

➤ 错误数据剔除

➤ 数据处理

➤ 股价复权

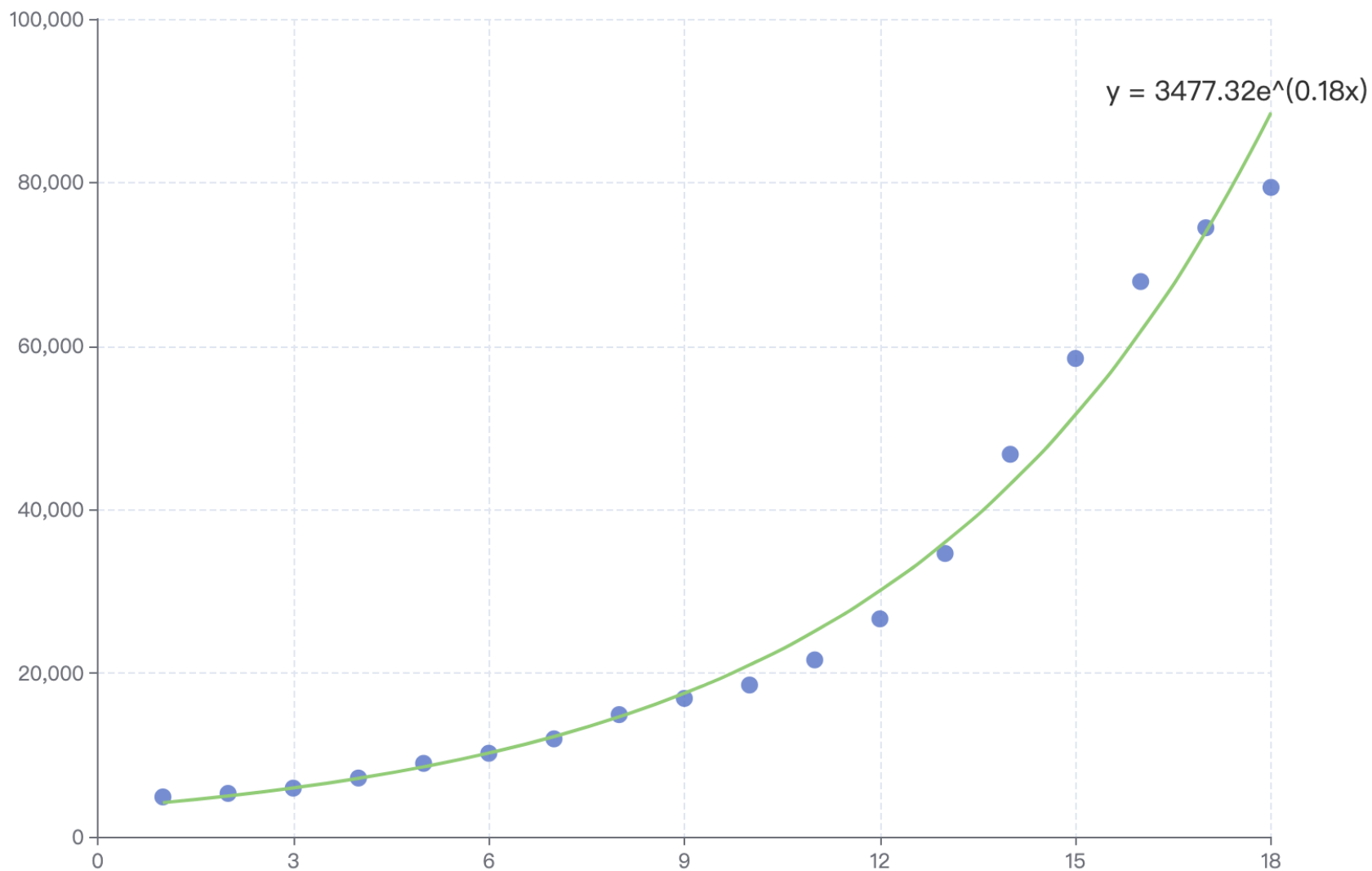
➤ 数据类型

➤ 股票、债券、基金、衍生品、指数、宏观行业等

代码	成交量(手)
交易日期	成交金额(千元)
货币代码	复权昨收盘价(元)
昨收盘价(元)	复权开盘价(元)
开盘价(元)	复权最高价(元)
最高价(元)	复权最低价(元)
最低价(元)	复权收盘价(元)
收盘价(元)	复权因子
涨跌(元)	均价(VWAP)
涨跌幅(%)	交易状态

股票相关数据属性

回归



线性回归：<https://echarts.apache.org/examples/zh/editor.html?c=scatter-linear-regression>

指数回归：<https://echarts.apache.org/examples/zh/editor.html?c=scatter-exponential-regression>

多项式回归：<https://echarts.apache.org/examples/zh/editor.html?c=scatter-polynomial-regression>

LSTM模型特点



超参数少



调参相对便捷



可拓展性强



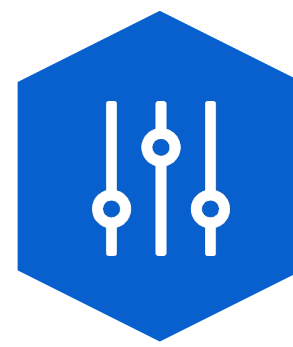
方便接入其他模型



长期依赖丢失



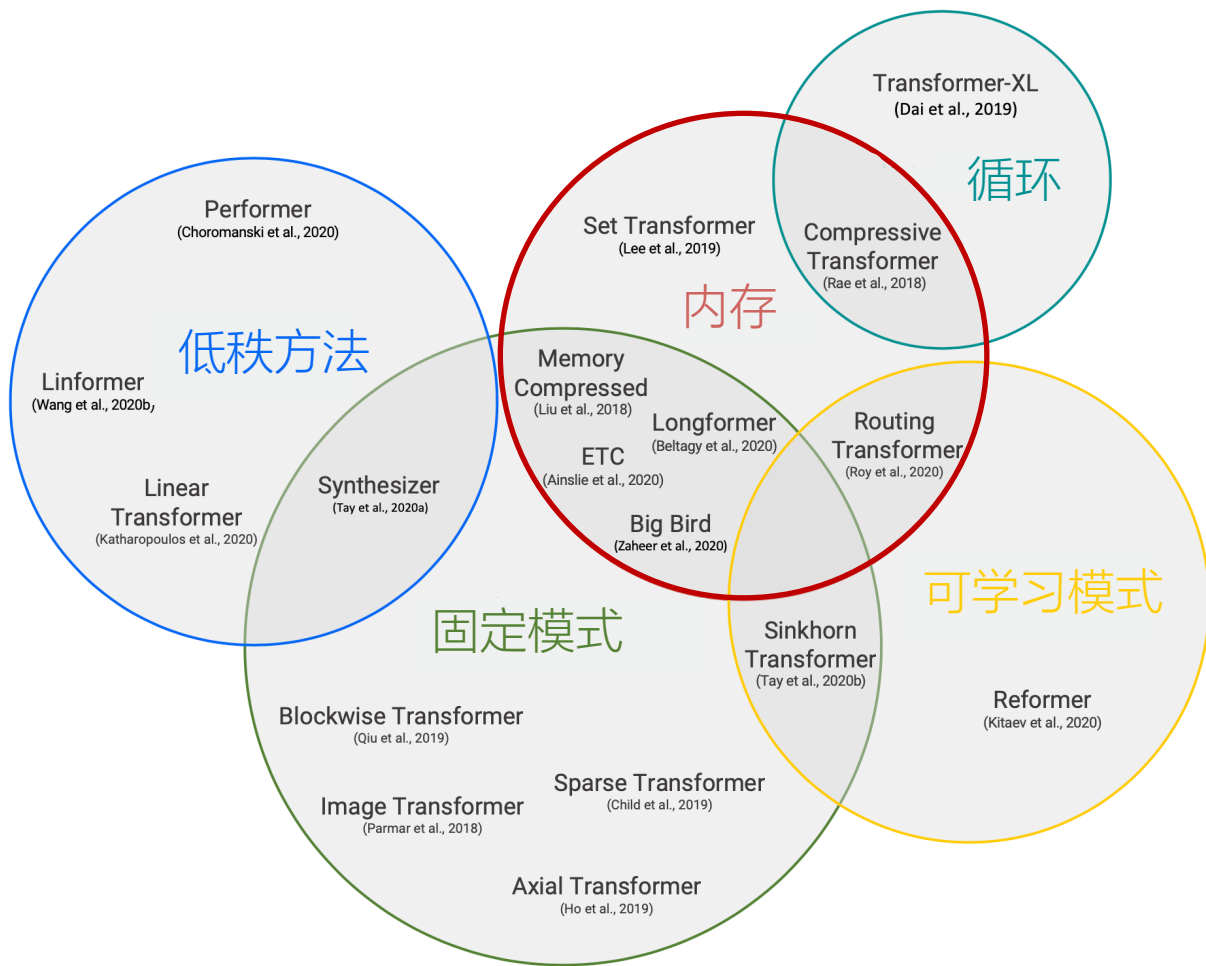
难以进行长期预测



难以捕获周期性



预测结果不够准确



注意力机制模型 (AI)

模型可以选择地聚焦于输入的某些部分
因此推理更加高效

Source 我在周日看了一本书



Target I read a book on Sunday

机器翻译



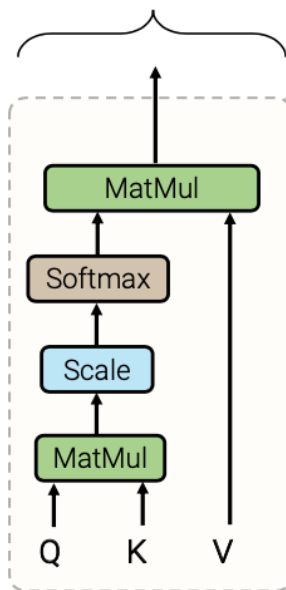
音乐生成



视频修复

AComputational
and Memory
Complexity

$$\mathcal{O}(n^2)$$

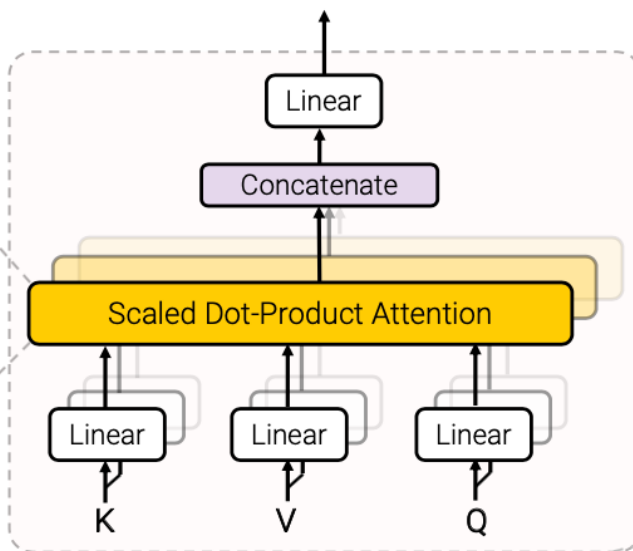
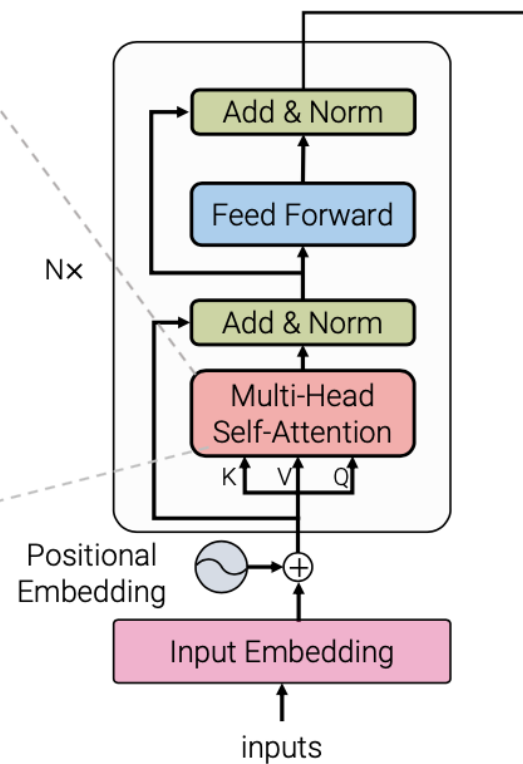
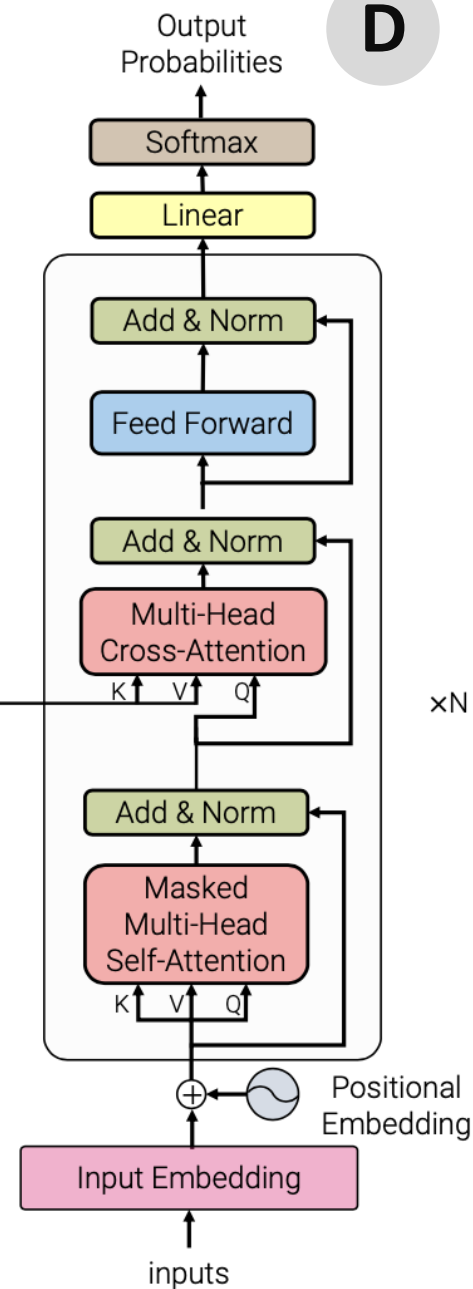


A 自注意力机制的计算过程

B Transformer中的多头注意力机制

C Transformer中的编码器结构，其中N为编码器重复的次数

D Transformer中的解码器结构。解码器的输出为价值的概率。

B**C****D**

Transformer模型特点



推理能力强



预测相对准确



可拓展性强



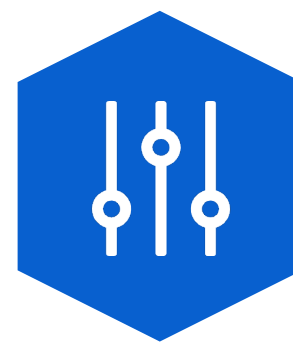
方便接入其他模型



超参数多



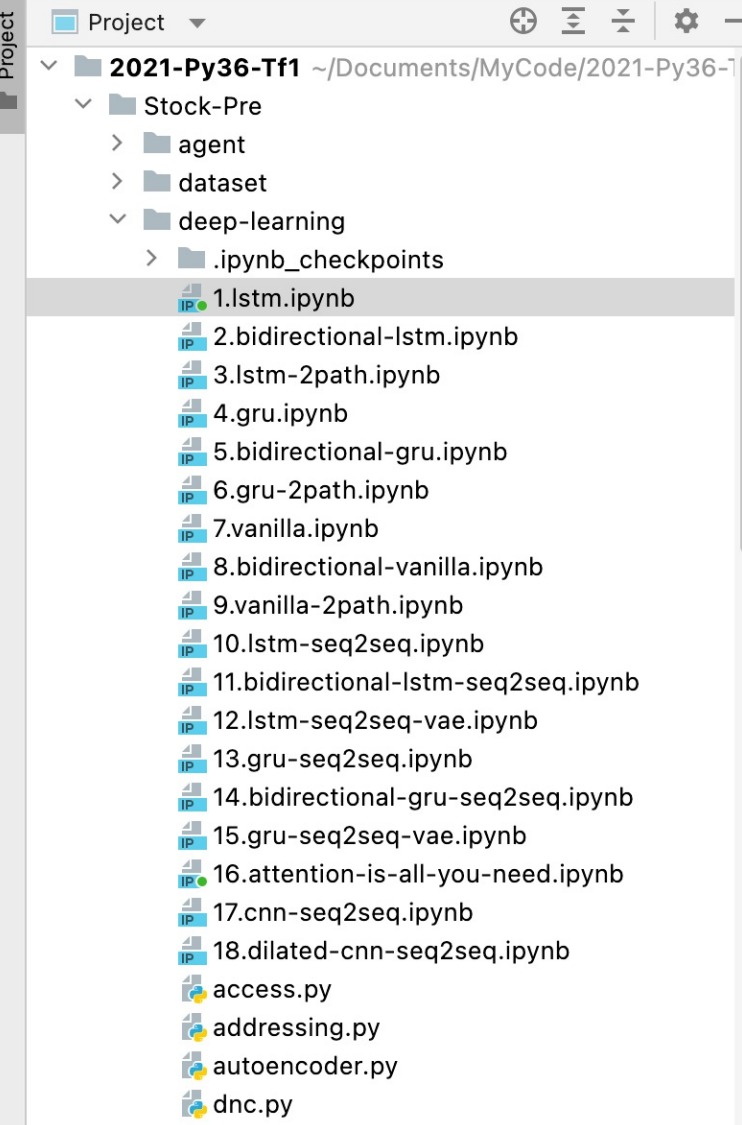
调参耗时



模型参数多



训练时间长



```
80     output_predict = minmax.inverse_transform(output_predict)
81     deep_future = anchor(output_predict[:, 0], 0.3)
82
83     return deep_future[-test_size:]

In _ 1 results = []
      2 for i in range(simulation_size):
      3     print('simulation %d'%(i + 1))
      4     results.append(forecast())

simulation 1
WARNING:tensorflow:<tensorflow.python.ops.rnn_cell_impl.LSTMCell object at 0x7fce58d30eb8>: Using a conc
is slower and will soon be deprecated. Use state_is_tuple=True.

train loop: 100%|██████████| 300/300 [00:41<00:00, 7.23it/s, acc=97.4, cost=0.00171]

simulation 2
WARNING:tensorflow:<tensorflow.python.ops.rnn_cell_impl.LSTMCell object at 0x7fce9119be10>: Using a conc
is slower and will soon be deprecated. Use state_is_tuple=True.

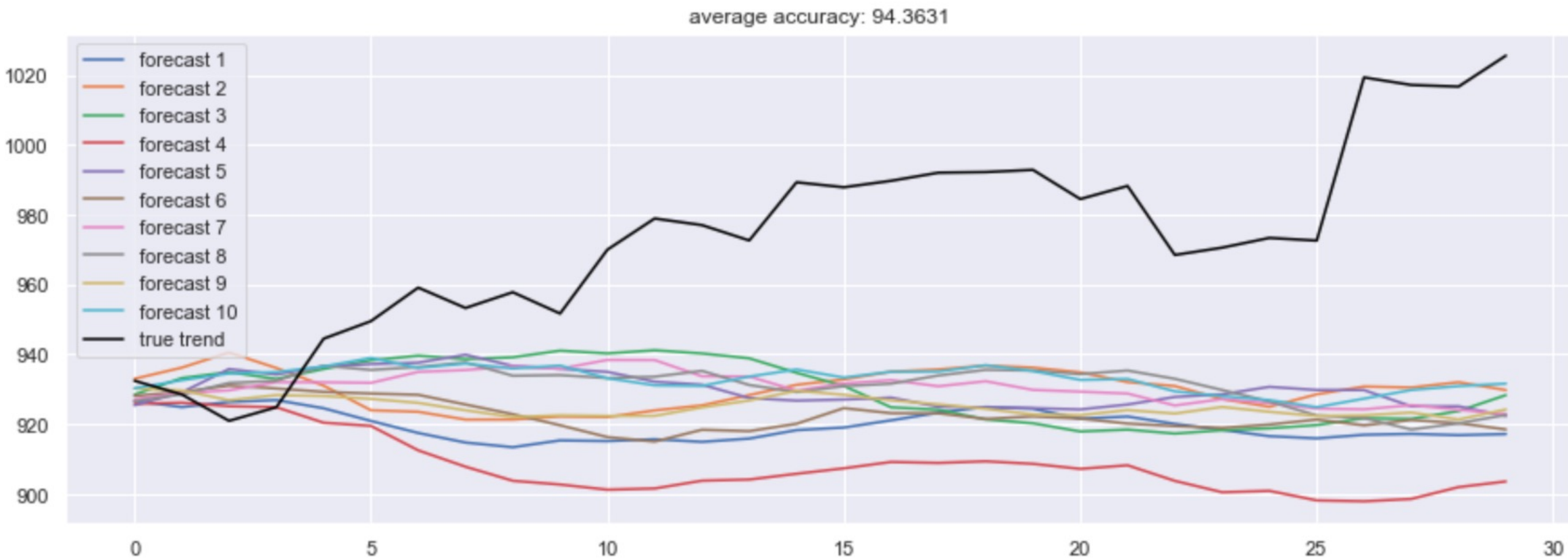
train loop: 28%|██████      | 85/300 [00:11<00:28, 7.63it/s, acc=96.6, cost=0.00323]

In _ 1 accuracies = [calculate_accuracy(df['Close'].iloc[-test_size:].values, r) for r in results]
      2
```

编程实践

PyCharm+Anaconda+Jupyter

<https://github.com/huseinzol05/Stock-Prediction-Models>



编程实践

```
1 test_size = 30
2 simulation_size = 2 # 10
```

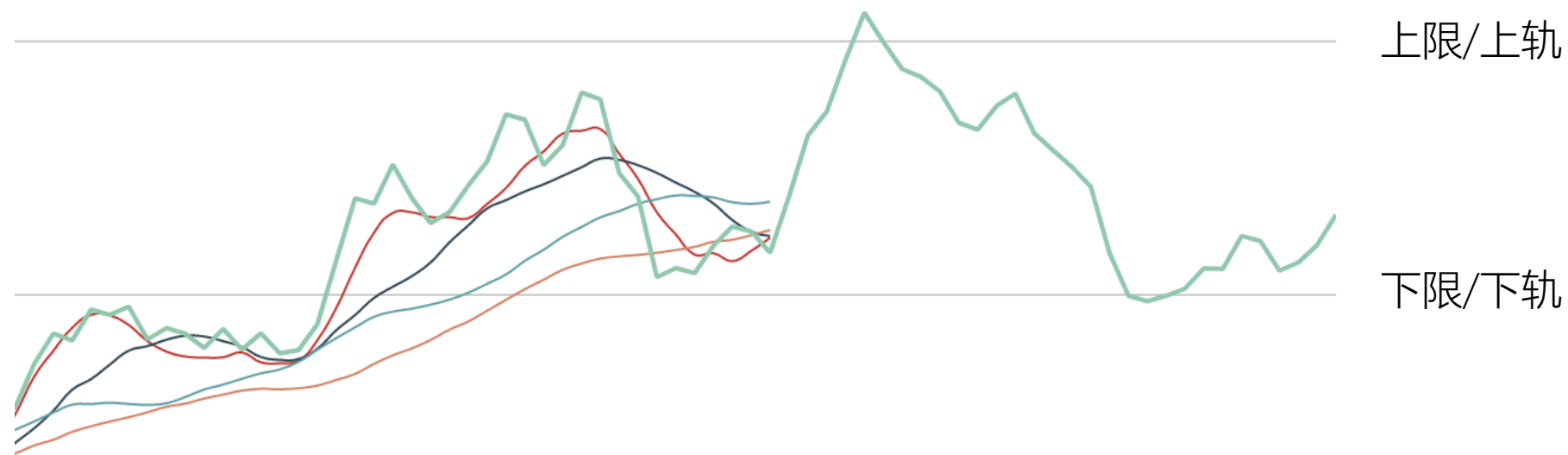
PyCharm+Anaconda+Jupyter

<https://github.com/huseinzol05/Stock-Prediction-Models>

股价区间预测



股价区间预测



| Q&A