

Прогнозирование рынка: ИИ в работе с временными рядами

ИНН. ПРАКТИКУМ

А В Т О Р : В А С Ю Р И Н А В . С .

К У Р А Т О Р : М А Т В Е Е В И . А .

Введение

В данной работе мы моделируем текущую рыночную ситуацию, учитывая последние события, что позволяет автоматически исследовать нелинейное поведение рынка и взаимосвязи между разными финансовыми активами. Мы генерируем разнообразные реалистичные прогнозы будущих тенденций, учитывая текущий контекст

Цель работы

- Реализовать нейросеть на обработку временных рядов прошлого с учётом хорошей выборки
- Сильнее приблизить наше представление о предстоящих изменениях финансовых активов, чтобы подобрать оптимальную стратегию инвестирования мелких или средних сумм

Постановка задачи

Максимизировать прогноз котировок за счёт улучшенной модели Марковица

Предыдущие результаты

- Доходность считаем как сумму долей доходностей по каждой акции
- Риск считаем через матричные произведения долей и матрицы ковариации.
- Сгенерируем прогноз и выведем результат на график риск-доходность.

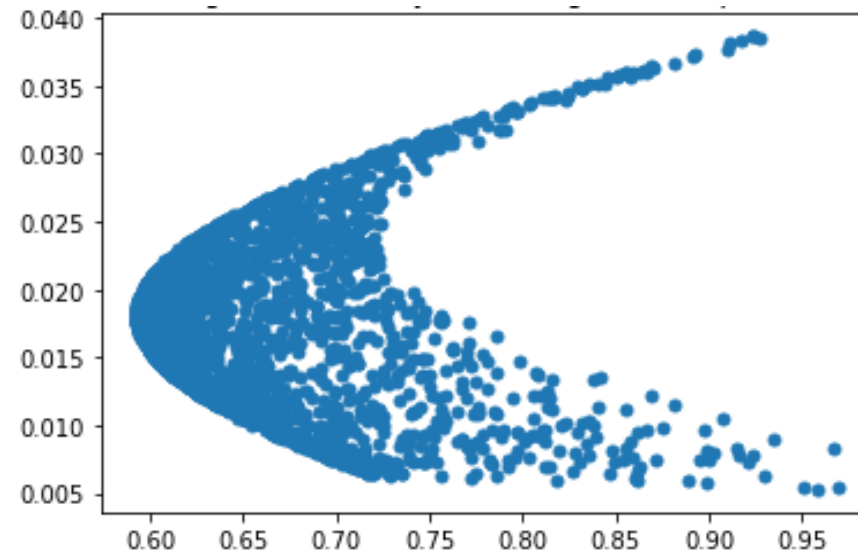


рис.1 *Модель Марковица*
риск-доход

Предыдущие результаты

Через встроенные функции реализуем прогноз, к примеру по цене закрытия

```
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

close_p = stock_ptve['Close']
train_len = math.ceil(len(close_p.values)* 0.8)

sc = MinMaxScaler(feature_range=(0,1))
sc_data = sc.fit_transform(close_p.values.reshape(-1,1))

train_data = sc_data[0: train_len, :]
x_train = []
y_train = []
for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])

x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
```

Результаты. Удобство представления датасета

```
2024-01-10,167.51750183105469,166.46499633789062,167.1500015258789,167.12000274658203,90420400,166.4460220336914
2024-01-11,167.7750015258789,167.125,167.20249938964844,167.69249725341797,78756800,167.01276397705078
2024-01-12,168.13999938964844,166.83000183105469,166.94499969482422,167.86499786376953,137310400,167.18354797363281
2024-01-13,168.82499694824219,167.73249816894531,167.86499786376953,168.7874984741211,133587600,168.09677124023438
2024-01-16,170.19750213623047,169.24500274658203,169.25,69.96499633789062,128186000,169.26245880126953
2024-01-17,170.44249725341797,169.69999694824219,169.89250183105469,170.10250091552734,114158400,169.39857482910156
2024-01-18,170.4749984741211,169.77999877929688,169.94999694824219,169.93499755859375,116028400,169.23274993896484
2024-01-19,170.29499816894531,169.73750305175781,169.875,170.00499725341797,98369200,169.30204772949219
2024-01-20,170.6624984741211,169.63999938964844,170.55750274658203,169.86000061035156,275978000,169.15850830078125
2024-01-23,171.0625,170.09249877929688,170.13249969482422,171.0,98572000,170.28706359863281
2024-01-24,171.22250366210938,170.7300033569336,171.17250061035156,171.06749725341797,48478800,170.3538818359375
2024-01-26,172.49500274658203,171.17500305175781,171.20500183105469,172.47750091552734,93121200,171.7497329711914
2024-01-27,173.49250030517578,172.02999877929688,172.77999877929688,172.44999694824219,146266000,171.72248840332031
2024-01-30,173.17250061035156,171.30500030517578,172.36499786376953,172.87999725341797,144114400,172.14817810058594
2024-01-31,173.41999816894531,172.37999725341797,172.48249816894531,173.4124984741211,100805600,172.67533874511719
```

- Investing.com российской версии (исходные данные)
- Переформатирование с учётом удаления колонки amount из котировок, перенос close столбца в конец

```
[ ] import pandas as pd
data = ts.get_h_data(stock_index, start=start_date, autype=None)
csv_data = 'stock-{}'.format(stock_index)
start_date = '2024-01-10'
data = data.reset_index(drop=True)
col_list.remove('close')
col_list.remove('amount')
col_list.append('close')
data.to_csv(csv_data, index=False)
data.head()
```

| | open | high | low | volume | close |
|---|-------|-------|-------|-----------|-------|
| 0 | 20.92 | 20.92 | 20.60 | 21.850597 | 20.64 |
| 1 | 21.00 | 21.15 | 20.72 | 26.910139 | 20.94 |
| 2 | 20.70 | 21.57 | 20.70 | 64.585536 | 21.02 |
| 3 | 20.60 | 20.70 | 20.20 | 45.886018 | 20.70 |
| 4 | 20.72 | 20.75 | 20.51 | 27.459940 | 20.60 |

Результат

- Рассмотрим рекуррентную двухслойную нейронную сеть, исключая точки, дающие шумы в прогнозе
- могут использовать свою внутреннюю память для обработки последовательностей произвольной длины:
- учитываем сезонность: наиболее простой способ называется классическим разложением и заключается в том, чтобы оценить тренд $T(t)$ через скользящее среднее, посчитать $S(t)$, как среднее без тренда $Y(t) - T(t)$ для каждого сезона

$$Y(t) = S(t) + T(t) + R(t)$$

$S(t)$ — сезонный компонент.
 $T(t)$ — компонент трендового цикла.
 $R(t)$ — остаток.

```
def build_model(layers):  
    model = Sequential()  
    model.add(LSTM(  
        input_dim=layers[0],  
        output_dim=layers[1],  
        return_sequences=True))  
    model.add(Dropout(0.4))  
  
    model.add(LSTM(  
        layers[2],  
        return_sequences=False))  
    model.add(Dropout(0.3))  
  
    model.add(Dense(  
        output_dim=layers[3]))  
    model.add(Activation("linear"))  
  
    start = time.time()  
    model.compile(loss="mse", optimizer="rmsprop", metrics=['accuracy'])  
    return model
```

```
model = build_model([X_train.shape[2], window, 100, 1])
```


Пример работы прогноза

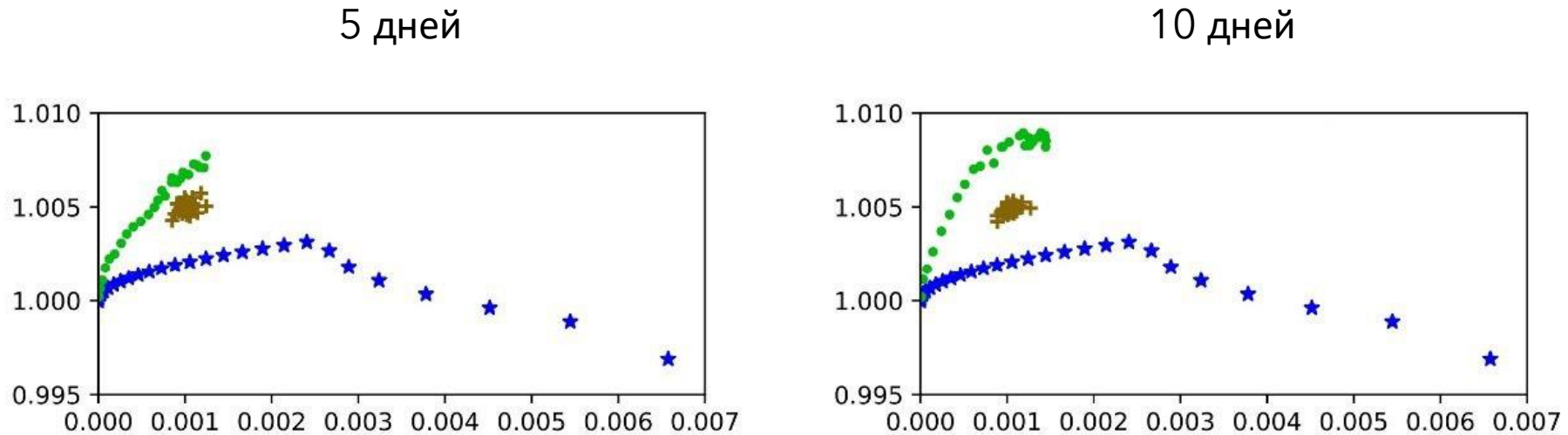


Рис.2 Сравнение исходных данных, модели Марковица, улучшенной модели (оси: риск-доход)

Анализ работы

Поведение Марковица, проявляющееся в увеличении риска и снижении доходности, обусловлено изменением вероятностного распределения доходности с течением времени, что может привести к моделям, recommending приобретение высокорискованных активов, теряющих ценность в период тестирования.

Выводы

Улучшенная модель же строит более точный прогноз, поскольку смотрит на распределение вероятностей прошлого показательной выборки, то есть, отбрасывая девиантные точки, базируется на апостериорной вероятности.

Однако погрешности в прогнозе всё же есть, равно как и длительная обработка кода

Библиография

(Р. А. Жиллов) Применение нейронных сетей
при кластеризации данных DOI:10.35330

(Daniel Oeltz, Jan Hamaekers, Kay F. Pilz) Parameterized Neural Networks for
Finance arXiv:2304.08883

(Булинская Е.В.) Теория рисков и перестрахование. Часть 1. Организация рисков
<https://arxiv.org/pdf/1909.10578>

(Лю Ю.-Д.) Методы и алгоритмы финансовой математики
<https://habr.com/ru/articles/516236/>