Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Томский политехнический Университет»

****

самостоятельная работа № 7

курс:

# «Линейная регрессия методом МНК »

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Исполнитель:** |  | | | | |
| студент группы | 0В21ДзебанА.А |  | ФИО студента |  | 26.12.24 |
|  |  |  |  |  |  |
| **Руководитель:** |  | | | | |
| преподаватель |  |  | Полищук В.Ю. |  |  |
|  |  |  |  |  |  |

Томск – 2024

**Цель работы:**

Обрести навыки реализации алгоритмов машинного обучения.

**Задание:**

1. Импорт необходимых библиотек:

import numpy as np

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

import matplotlib.pyplot as plt

import yfinance as yf

from sklearn.model\_selection import train\_test\_split

1. Подготовим данные (нормализация, приведение данных к необходимому виду):

#Загрузка данных

data = yf.download('NVDA', start='2020-01-01', end='2024-01-01')

data['SMA\_10'] = data['Close'].rolling(window=10).mean()

data['SMA\_20'] = data['Close'].rolling(window=20).mean()

data['EMA\_10'] = data['Close'].ewm(span=10, adjust=False).mean()

data['Volatility'] = data['High'] - data['Low']

data['Price\_Change'] = data['Close'].diff()

data.fillna(data.bfill(),inplace=True)

data.fillna(data.ffill(), inplace=True)

#Нормировка данных

features = ['SMA\_10', 'SMA\_20', 'EMA\_10', 'Volatility', 'Price\_Change', 'Close']

scaler = MinMaxScaler(feature\_range=(0, 1))

scaled\_data = scaler.fit\_transform(data[features])

X = scaled\_data[:,:-1]

y = scaled\_data[:,-1]

Составим различные графики:

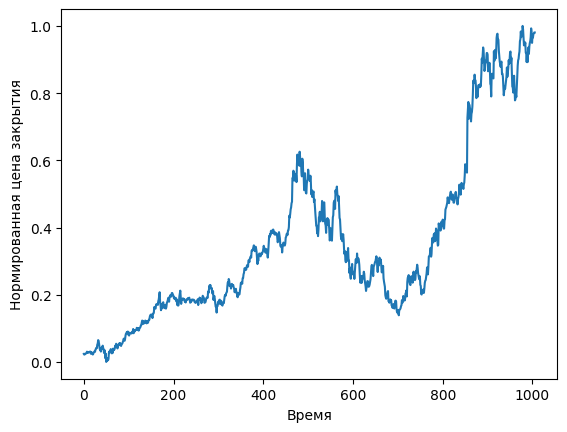
1. Нормированная цена закрытия
2. Столбчатая диаграмма изменения цен
3. График скользящего среднего

#График цен

plt.plot(y)

plt.ylabel("Нормированная цена закрытия")

plt.xlabel("Время")



#Гистограмма изменения цен

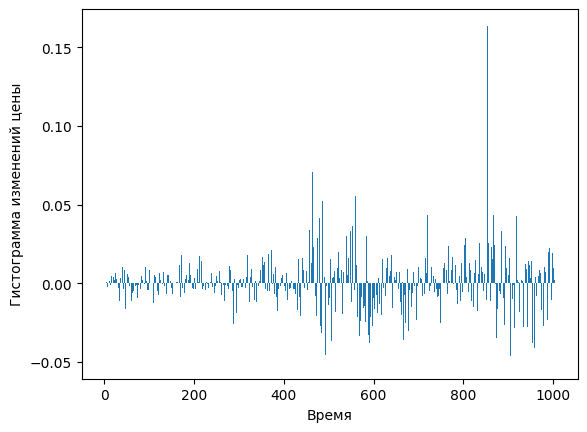
diffs = [y[i] - y[i-1] for i in range(1,len(y))]

time = [i for i in range(len(diffs))]

plt.bar(time,diffs)

plt.xlabel("Время")

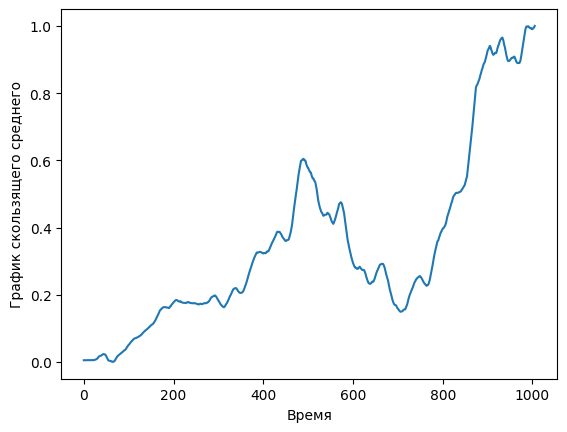
plt.ylabel("Гистограмма изменений цены")



plt.plot(X[:,1])

plt.xlabel("Время")

plt.ylabel("График скользящего среднего")



#Добавим столбец единиц для правильной формы входных данных

X = np.concatenate([np.ones((X.shape[0], 1)), X], axis=1)

**Определим линейную модель как вычисление НК матричным методом:**

def linear\_model(X,y):

    beta = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)

    return beta

beta = linear\_model(X,y)

#Для дополнительной проверки разделим выборку в соотношении 60/40 и проверим данные

X\_train, X\_test, y\_train,y\_test = train\_test\_split(X,y, test\_size=0.4, shuffle=False)

beta2 = linear\_model(X\_train, y\_train)

y\_pred = X\_test.dot(beta2)

**Построим график фактических и предсказанных данных:**

# График фактических и предсказанных значений

plt.plot(y\_test, label='Фактические значения')

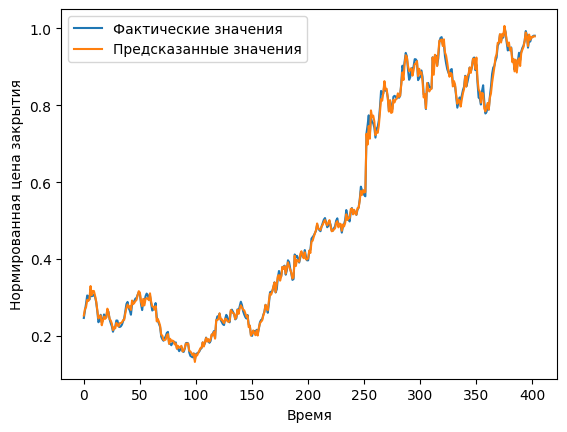
plt.plot(y\_pred, label='Предсказанные значения')

plt.ylabel("Нормированная цена закрытия")

plt.xlabel("Время")

plt.legend()

plt.show()

****

**Вывод: была реализована линейная регрессия с помощью МНК.**