Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____Информатика и системы управления_____

КАФЕДРА _____Системы обработки информации и управления (ИУ5)_____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:
Прогнозирование цен акций S&P 500

Студент __ИУ5-65Б_____ _____ _____
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ _____
(Подпись, дата) (И.О.Фамилия)

Консультант _____ _____
(Подпись, дата) (И.О.Фамилия)

2022 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой _______
(Индекс)

_____ _____
(И.О.Фамилия)

« _____ » _____ 2022 г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме Прогнозирование цен акций S&P 500

Студент группы ___ИУ5-65Б

_____Богданова Валерия Вячеславовна_____
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

____ учебная _____

Источник тематики (кафедра, предприятие, НИР) ___ кафедра _____

График выполнения НИР: 25% к ___ нед., 50% к ___ нед., 75% к __ нед., 100% к ___ нед.

Техническое задание *Прогнозирование цен акций, обучение произвести на данных S&P 500 за 2020 год, прогноз сделать на 2021 год с помощью скользящего среднего. Рассчитать результаты с помощью метрик и сравнить с другой моделью*

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на ____ листах формата А4.

Дата выдачи задания « » _____ 2022 г.

Руководитель НИР _____ _____
(Подпись, дата) (И.О.Фамилия)

Студент _____ _____
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Оглавление

1. Введение

Данная программа, выполненная в рамках научно-исследовательской работы по предмету “Технологии машинного обучения”, предназначена для прогнозирования цен акций на данных S&P 500

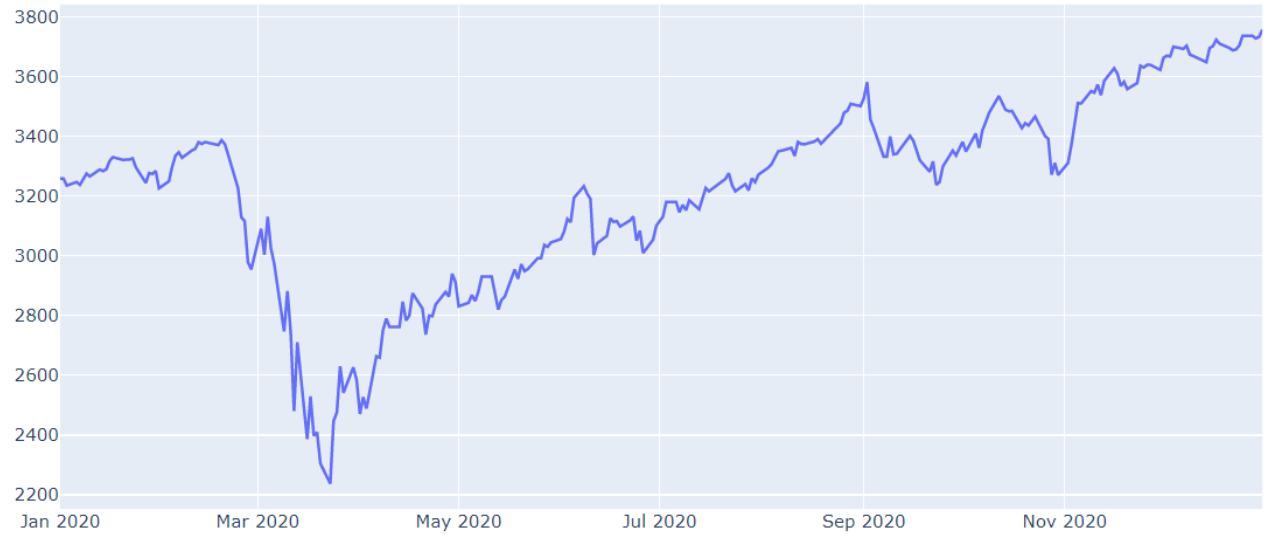
2. Основная часть

Сначала необходимо получить данные, сделать это можно с помощью web.DataReader, указав нужные даты.

```
sp500_data = pd.DataFrame(web.DataReader(['sp500'], data_source = "fred",
start=datetime.datetime(2020, 1, 1),
                          end =datetime.datetime(2020, 12, 31)))
test_data = pd.DataFrame(web.DataReader(['sp500'], data_source = "fred",
start=datetime.datetime(2021, 1, 1),
                          end =datetime.datetime(2021, 12, 31)))
```

Оказалось, что в полученных данных есть пропуски. Их можно заполнить с помощью bfill (according to the last observed value)

```
sp500_data = sp500_data.bfill()
test_data = test_data.bfill()
```



В pandas есть встроенная функция rolling, в которой можно указать окно. Выведем результаты для малых значений окна, они довльно хорошо повторяют данные, но слишком сильно переобучаются на них. На графике видно, что слишком маленькое окно переподгоняется к данным, а не к тенденции, а большое имеет некоторое смещение относительно реальных значений и общей тенденции.

```
rolling_mean2 = sp500_data.sp500.rolling(window=2).mean()
rolling_mean5 = sp500_data.sp500.rolling(window=5).mean()
rolling_mean8 = sp500_data.sp500.rolling(window=8).mean()
```

Попробуем для разных размеров окон подсчитать такие метрики качества прогноза, как mae и mape. Для этого используем "кросс валидацию" для временного ряда – TimeSeriesSplit.

```
tscv = TimeSeriesSplit()
mae_scores = [[],[],[],[],[],[],[],[],[],[],]
mape_scores = [[],[],[],[],[],[],[],[],[],[],]
for train_index, test_index in tscv.split(sp500_data):
    X_train, X_test = sp500_data.sp500.values[train_index],
sp500_data.sp500.values[test_index]
    for window in range(1,10):
        predictions = list()
        for t in range(len(X_test)):
            length = len(X_train)
            yhat = np.mean([X_train[i] for i in range(length-window,length)])
            predictions.append(yhat)
            X_train = np.append(X_train,X_test[t])
        error_mae = mean_absolute_error(X_test, predictions)
        error_mape = mean_absolute_percentage_error(X_test, predictions)

        mae_scores[window].append(error_mae)
        mape_scores[window].append(error_mape)

for window, scores in enumerate(mae_scores):
    if window == 0:
        continue
```

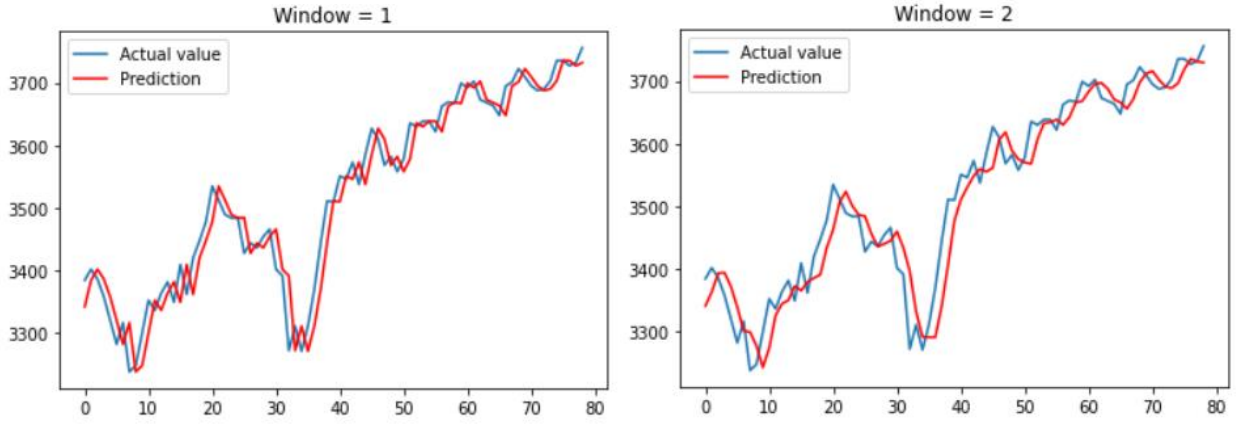
```
    print("Window = ", window, ", mae = ", np.mean(scores), ", mape = ",
          float('{:.3f}'.format(np.mean(mape_scores[window]))), '%')
```

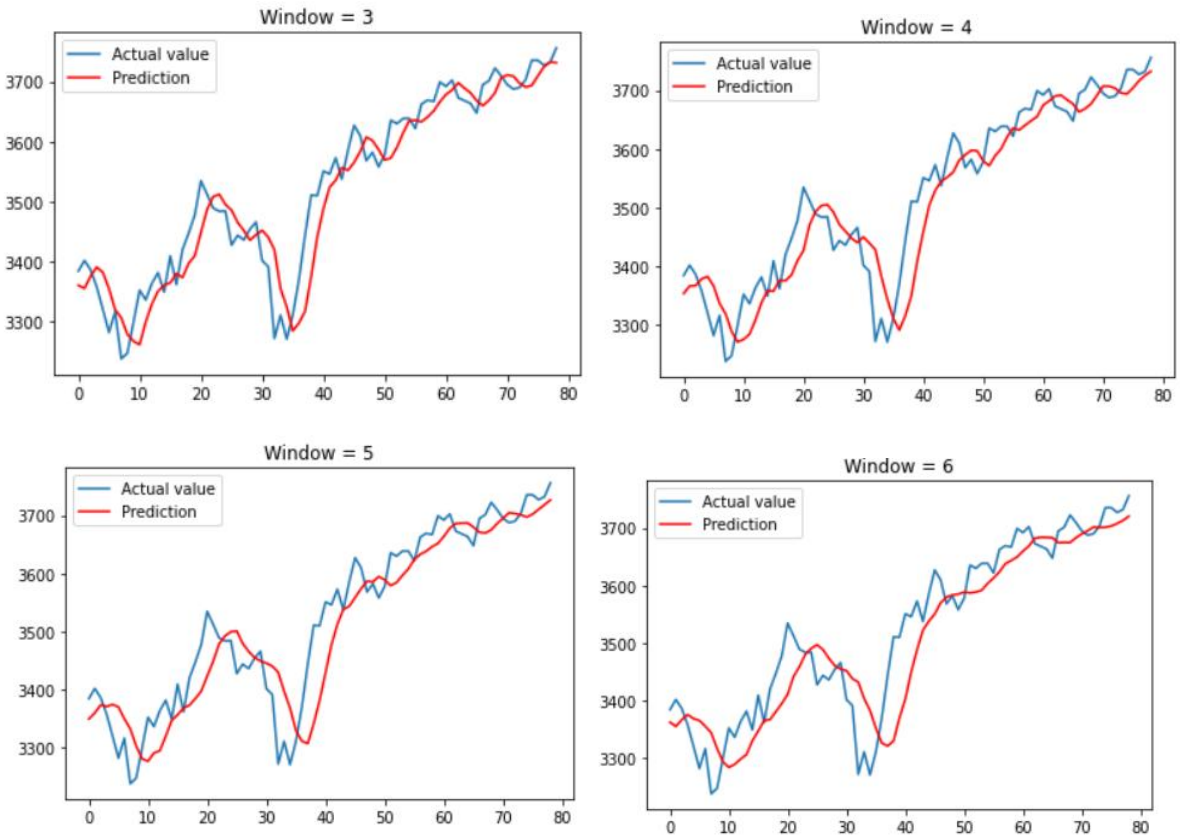
Получаем результаты:

```
Window =  1 , mae =  40.77372093023255 , mape =  0.014 %
Window =  2 , mae =  46.42106976744185 , mape =  0.015 %
Window =  3 , mae =  52.01931782945735 , mape =  0.017 %
Window =  4 , mae =  57.803651162790686 , mape =  0.019 %
Window =  5 , mae =  62.853990697674384 , mape =  0.021 %
Window =  6 , mae =  67.20640310077518 , mape =  0.022 %
Window =  7 , mae =  70.09994684385381 , mape =  0.023 %
Window =  8 , mae =  73.23298837209299 , mape =  0.024 %
Window =  9 , mae =  76.12607751937983 , mape =  0.025 %
```

Видим, что ошибка лишь увеличивается при увеличении окна, но также мы уже видели переподгонку меньшего окна к данным, поэтому попробуем изобразить разные окна на графиках.

```
window = 3
data_rate = 0.7
for window in range(1,7):
    X_train, X_test = sp500_data.sp500.values[:int(len(sp500_data)*data_rate)],
\
    sp500_data.sp500.values[int(len(sp500_data)*data_rate):]
    predictions = list()
    for t in range(len(X_test)):
     #print(X_test[t])
        length = len(X_train)
        yhat = np.mean([X_train[i] for i in range(length-window,length)])
        #print([X_train[i] for i in range(length-window,length)])
        predictions.append(yhat)
        X_train = np.append(X_train,X_test[t])
    pyplot.plot(X_test)
    pyplot.plot(predictions, color='red')
    pyplot.title("Window = "+ str(window))
    pyplot.legend(['Actual value', 'Prediction'])
    pyplot.show()
```



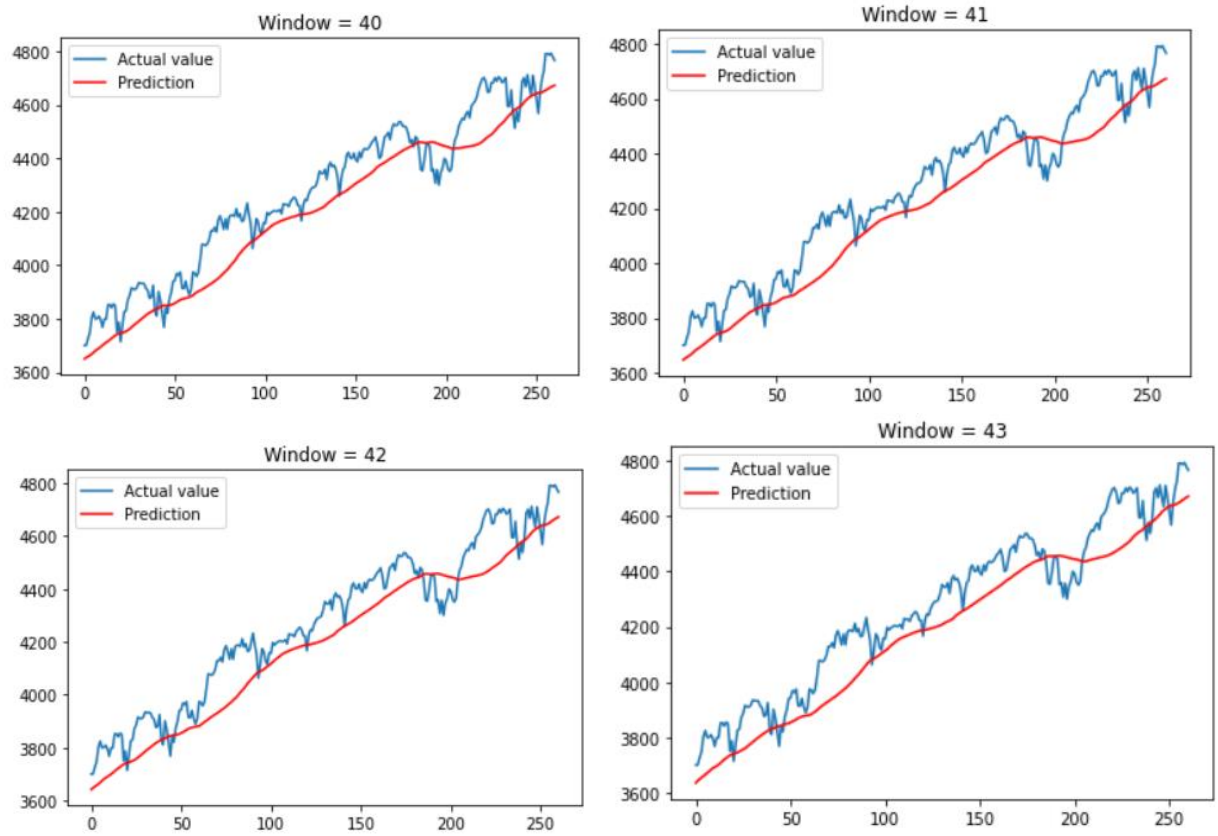


Но на графиках видно, что малое значение окна переподгоняется к данным и по сути мало полезно, нам нужна сама тенденция роста или спада

Ориентируясь на краткосрочные тенденции для "быстрых" продаж и покупок, возьмем размер окна = 5.

Для предсказания долгосрочной тенденции лучше подходит большое значение окна, например, 40 или 50. Посмотрим на них визуально и оценим их ошибку предсказания.

```
window = 50
for window in range(40,45):
    X_train, X_test = sp500_data.sp500, \
    test_data.sp500
#print(X_test)
    predictions = list()
    for t in range(len(X_test)):
     #print(X_test[t])
        length = len(X_train)
        yhat = np.mean([X_train[i] for i in range(length-window,length)])
        #print([X_train[i] for i in range(length-window,length)])
        predictions.append(yhat)
        X_train = np.append(X_train,X_test[t])
    pyplot.plot(X_test)
    pyplot.plot(predictions, color='red')
    pyplot.title("Window = "+ str(window))
    pyplot.legend(['Actual value', 'Prediction'])
    pyplot.show()
```




```
window_start = 45
window_end = 65

for train_index, test_index in tscv.split(sp500_data):
    X_train, X_test = sp500_data.sp500.values[train_index],
sp500_data.sp500.values[test_index]
    id_wind=0
    for window in range(window_start,window_end):
        predictions = list()
        for t in range(len(X_test)):
            length = len(X_train)
            yhat = np.mean([X_train[i] for i in range(length-window,length)])
            predictions.append(yhat)
            X_train = np.append(X_train,X_test[t])
        error_mae = mean_absolute_error(X_test, predictions)
        error_mape = mean_absolute_percentage_error(X_test, predictions)

        mae_scores[id_wind].append(error_mae)
        mape_scores[id_wind].append(error_mape)
        id_wind+=1

for ind in range(0,window_end-window_start):
    print("Window = ", window_start, ", mae = ", np.mean(mae_scores[ind]), ",
mape = ",
          float('{:.3f}'.format(np.mean(mape_scores[ind]))), '%')
    window_start+=1
```

Получаем результаты:

```
Window =  45 , mae =  175.85048578811364 , mape =  0.059 %
Window =  46 , mae =  88.5444994944388 , mape =  0.029 %
Window =  47 , mae =  88.4193330034636 , mape =  0.029 %
Window =  48 , mae =  87.53309302325583 , mape =  0.029 %
Window =  49 , mae =  86.85519886093974 , mape =  0.029 %
Window =  50 , mae =  86.31141767441864 , mape =  0.028 %
Window =  51 , mae =  86.03530323757414 , mape =  0.028 %
```

```
Window =  52 , mae =  85.9794436493739 , mape =  0.028 %
Window =  53 , mae =  86.14382799473456 , mape =  0.028 %
Window =  54 , mae =  86.56512144702847 , mape =  0.029 %
Window =  55 , mae =  87.1854156448203 , mape =  0.029 %
Window =  56 , mae =  87.93011295681062 , mape =  0.029 %
Window =  57 , mae =  88.74183761729905 , mape =  0.029 %
Window =  58 , mae =  89.69120769847635 , mape =  0.03 %
Window =  59 , mae =  90.60845092629093 , mape =  0.03 %
Window =  60 , mae =  91.57478759689926 , mape =  0.03 %
Window =  61 , mae =  92.56334578726651 , mape =  0.031 %
Window =  62 , mae =  93.52175543885974 , mape =  0.031 %
Window =  63 , mae =  94.50367515688446 , mape =  0.031 %
Window =  64 , mae =  95.42395930232556 , mape =  0.032 %
```

Видно, что для размера окна в таком диапазоне оптимально значение 52, оно дает наименьшую ошибку

3. Заключение

В ходе исследования было выявлено, что для прогнозирования на короткий срок, например, 5 или 10 минут, лучше использовать размер окна = 5, а на более длительный срок размер окна должен быть равен 52

4. Список использованных источников информации

1. Курс машинного обучения 3 курса ИУ5 МГТУ им. Баумана
2. Статья по основам скользящего среднего https://www.machinelearningmastery.ru/implementing-moving-averages-in-python-1ad28e636f9d/