Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 519.233.5

АНАЛИЗ АЛГОРИТМОВ ПРЕДСКАЗАНИЯ ОДНОМЕРНЫХ ВРЕМЕННЫХ РЯДОВ, ОСНОВАННЫХ НА МОДЕЛИ АВТОРЕГРЕССИИ

Федоров Виктор Олегович,
кандидат технических наук, доцент,
КФ МГТУ им. Баумана, кафедра ИУК-5, г. Калуга,
fedorov_vo@bmstu.ru

Кирюнин Артем Алексеевич,
студент, магистр, группа ИУК-2-31М,
КФ МГТУ им. Баумана, кафедра ИУК-2, г. Калуга,
artemkiryunin@mail.ru

Аннотация

В статье проводится исследование эффективности предсказания алгоритмов предсказания, основанных на модели авторегрессии, на временных рядах с различными характеристиками. Выявляются достоинства и недостатки алгоритмов AR, ARIMA и SARIMA, а также определяется их применимость к различным временным рядам. Даётся пояснение к результатам работы алгоритмов, а также приводятся таблицы со статистическими данными работы алгоритмов.

Ключевые слова: Авторегрессия, авторегрессионная модель, AR, ARIMA, SARIMA, временные ряды, методы предсказания временных рядов.

ANALYSIS OF ALGORITHMS FOR PREDICTING ONE-DIMENSIONAL TIME SERIES BASED ON THE AUTOREGRESSION MODEL

Viktor O. Fedorov,
Candidate of Technical Sciences, Associate Professor,
KB BMSTU, department of IUK-5, Kaluga,
fedorov_vo@bmstu.ru

Artem A. Kiryunin,
student, master's degree, IUK-2-31M group,
KB BMSTU, department of IUK-2, Kaluga,
artemkiryunin@mail.ru

ABSTRACT

The paper investigates the prediction performance of prediction algorithms based on the autoregressive model on time series with different characteristics. The advantages and disadvantages of AR, ARIMA and SARIMA algorithms are revealed, and their applicability to different time series is determined. An explanation of the results of the algorithms is given, and tables with statistical data of the algorithms are given.

Keywords: Autoregression, autoregressive model, AR, ARIMA, SARIMA, time series, time series prediction methods.

Введение

Авторегрессионная (AR-) модель – это модель временных рядов, в которой значения временного ряда в данный момент линейно зависят от предыдущих значений этого же ряда. Данная модель в своём общем виде давно известна и активно применяется для прогнозирования поведения временных рядов [1].

Слово «авторегрессия» означает зависимость последующего значения от предыдущего. Точно так же в модели авторегрессии прогнозирование использует предыдущие значения временного ряда. Зависимость в случае авторегрессии предполагается линейная, то есть прогноз представляет собой сумму членов временного ряда за предыдущие дни с некоторыми коэффициентами, которые являются постоянными и определяют параметры модели авторегрессии. Количество временных интервалов (периодов в общем случае), берущихся из предыдущих наблюдений, чтобы спрогнозировать последующие члены временного ряда называется порядком модели авторегрессии p [2].

На языке программирования Python данная модель реализована в библиотеке «statsmodels.tsa.ar_model» как «AR» [3]. Данная реализация позволяет автоматически подбирать параметр p наилучшим образом и, тем самым, значительно упростить процесс предсказания. Однако следует отметить, что предсказания алгоритмом AR осуществляются итеративно, т.е. по одному члену ряда за шаг, при этом постоянно опираясь на предыдущие члены временного ряда, зачастую, уже предсказанные ранее, что зачастую выливается в накопление ошибок.

В дальнейшем модель авторегрессии развилась в алгоритм ARMA (авторегрессия скользящих средних). Данный алгоритм не рассматривается в текущей статье, поскольку его современник, алгоритм ARIMA (авторегрессия интегрированных скользящих средних) обладает всеми плюсами своего предшественника и собственными наработками.

Данный алгоритм имеет три параметра, которые необходимо задать [4]:

p – порядок авторегрессии, то же самое, что для алгоритма AR;

q – порядок скользящего среднего. Используется для сглаживания временного ряда: чем параметр больше – тем более временной ряд приближен к среднему значению на конкретном промежутке;

d – количество разностей, необходимых для того, чтобы временной ряд стал стационарным, т.е. фактически, степень амплитудного сжатия данных.

На языке программирования Python данная модель реализована в библиотеке «statsmodels.tsa.arima_model» как «ARIMA» [5].

Главная сложность при использовании данного алгоритма – подбор параметров. Одним из популярных методов решения является полный перебор моделей со всеми комбинациями параметров вплоть до заданного значения и отбор лучшей из них по информационному критерию Акаике (AIC), который определяет наибольшую изменчивость данных за наименьшее количество параметров [6]. Чем AIC у модели меньше, тем лучше подходит данный набор параметров.

Следующим этапом в развитии модели стало добавление сезонности, распознание которой в предыдущих алгоритмах было невозможно. Так появилась модель SARIMA (сезонная авторегрессия интегрированных скользящих средних). Данный алгоритм обладает теми же параметрами p, d, q, что и алгоритм ARIMA, но при этом также добавляются параметры p1, d1, q1 для сезонной функции, периодичность которой (в единицах временного ряда) задаётся параметром s. Таким образом, модель SARIMA обладает сразу 7 параметрами (p, d, q, s, p1, d1, q1), из которых очевидным является только один – параметр s. Данный параметр определяется периодом сезонной функции: для среднесуточной годовой температуры он будет равен 365 (т.к. 365 дней в году), если наблюдается циклическое изменение дневных данных в зависимости от дня недели, то 7 (т.к. дней в неделе 7) и т.д.

Остальные параметры так же, как и в случае с алгоритмом ARIMA, подбираются перебором параметров с последующей оценкой получившейся модели критерием AIC.

На языке программирования Python данная модель реализована в библиотеке «statsmodels.tsa.arima.model» как «SARIMA» [7].

Материалы и методы исследования

Зададим несколько функций, чтобы увидеть работу алгоритмов.

Сперва зададим функцию синуса:

$$x(i) = 2 * \sin\left(\frac{\pi * i}{8}\right) + 4 + rand \quad (1)$$

где i – номер члена в ряду;

rand – случайный шум в диапазоне от -0.5 до 0.5.

В интервале от 0 до 100 будет находиться тренировочный массив, а в интервале от 101 до 130 – проверочный, без добавления случайного шума.

Без добавления шумов, данный график будет выглядеть так (рис. 1):

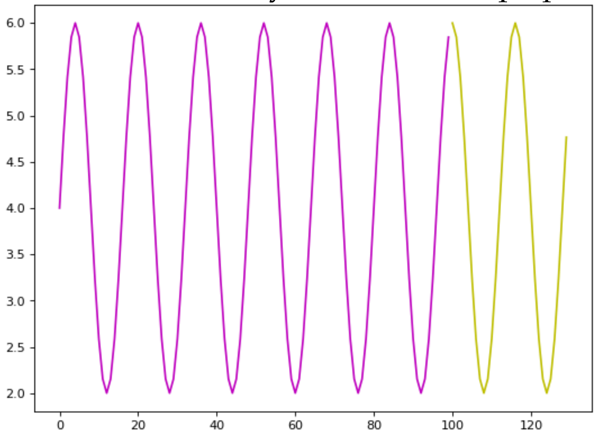


Рисунок 1. Идеальный график функции 1

На данном рисунке фиолетовым цветом изображён тренировочный массив, а жёлтым – проверочный. Тренд в данной функции стационарен, параметр сезонности «s» равен 16.

Теперь попробуем задать степенную функцию:

$$x(i) = i^5 - 20000 * i^3 + rand \quad (2)$$

где i – номер члена в ряду;

rand – случайный шум в диапазоне от -100000000 до 100000000.

Без добавления шумов, данный график будет выглядеть так (рис. 2):

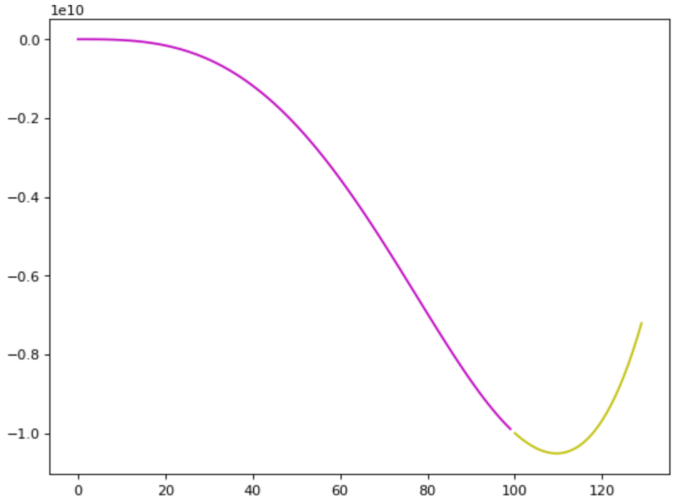


Рисунок 2. Идеальный график функции 2

На данном рисунке фиолетовым цветом изображён тренировочный массив, а жёлтым – проверочный. Сезонность в данной функции отсутствует, а тренд задан степенной функцией.

Следующей зададим функцию с явно выраженной сезонностью:

$$\begin{cases} x(i) = 10 + \frac{i}{5} + rand \quad \text{при } i\%20 < 5 \\ x(i) = 5 + \frac{i}{5} + rand \quad \text{при } i\%20 \geq 5 \end{cases} \qquad (3)$$

где i – номер члена в ряду;

rand – случайный шум в диапазоне от -1 до 1.

Без добавления шумов, данный график будет выглядеть так (рис. 3):

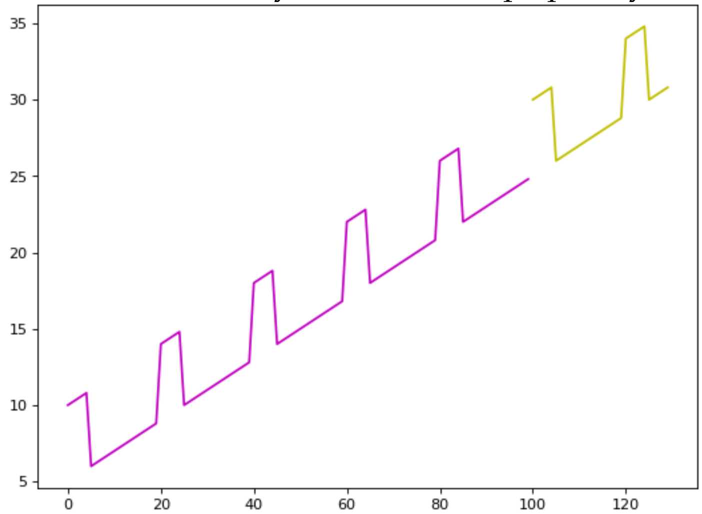


Рисунок 3. Идеальный график функции 3

На данном рисунке фиолетовым цветом изображён тренировочный массив, а жёлтым – проверочный. Тренд для данной функции линеен, а параметр сезонности «s» равен 20.

Последней создадим функцию с явно выраженной сезонностью и нелинейным трендом:

$$\begin{cases} x(i) = 100 + \frac{i}{5} - \frac{i^3}{30000} + rand \quad \text{при } i\%20 > 5 \\ x(i) = 85 + \frac{i}{5} - \frac{i^3}{30000} + rand \quad \text{при } i\%20 \leq 5 \end{cases} \qquad (4)$$

где i – номер члена в ряду;

rand – случайный шум в диапазоне от -1 до 1.

Без добавления шумов, данный график будет выглядеть так (рис. 4):

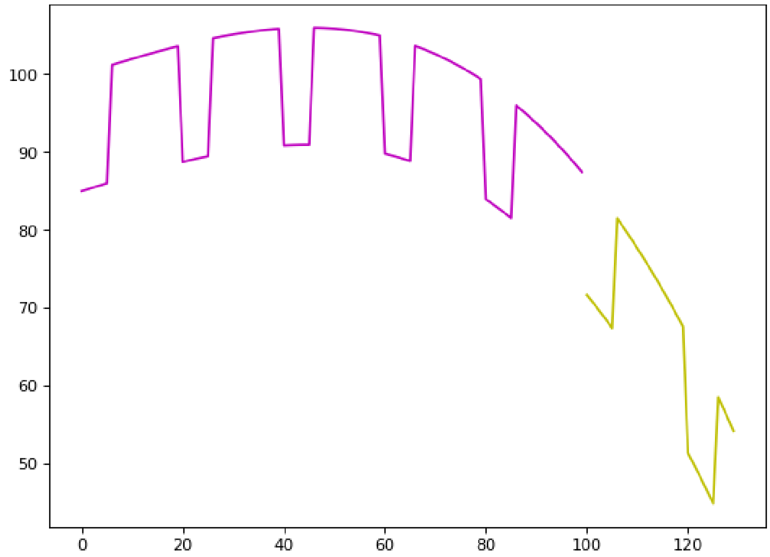


Рисунок 4. Идеальный график функции 4

На данном рисунке фиолетовым цветом изображён тренировочный массив, а жёлтым – проверочный. Параметр сезонности «s» в данной функции равен 20, а тренд задан степенной функцией.

Результаты исследования и их обсуждение

Теперь рассмотрим графики с шумом и предсказания, совершённые всеми тремя алгоритмами, заранее условившись, что каждый алгоритм будет предсказывать последующие 30 членов временного ряда из изначального ряда, длинной в 100.

Предсказания для функции номер 1 будут выглядеть так (рис. 5):

а *б*

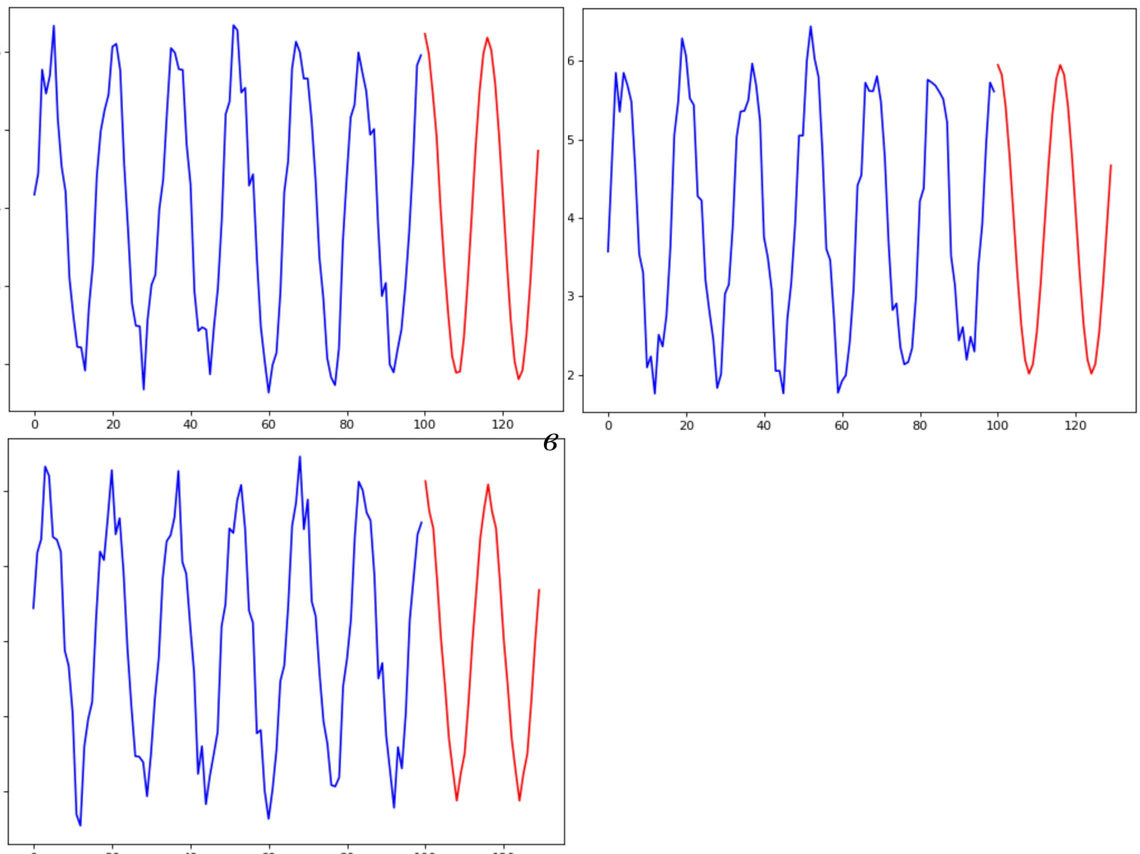


Рисунок 5. Результаты работы алгоритмов с функцией 1: а – алгоритм AR, б – алгоритм ARIMA, в – алгоритм SARIMA

Здесь синим цветом изображён тренировочный массив, а красным – предсказанный.

Как можно заметить визуально, все алгоритмы справились со своей задачей. Однако в единичном случае картину могут слишком сильно исказить случайные шумы, поэтому для оценки точности работы алгоритма стоит повторять предсказание некоторое ограниченное число раз, чтобы получить более объективные результаты.

Теперь определим критерии, которыми будем оценивать точность и время работы алгоритмов:

Среднее время работы на итерацию (t_sr), в секундах;

Минимальное время работы (t_min), в секундах;

Максимальное время работы (t_max), в секундах;

Средний арифметический коэффициент R-square за все итерации (r2_sr);

Минимальный коэффициент R-square (r2_min);

Максимальный коэффициент R-square (r2_max);

Среднее арифметическое от средних процентных ошибок за каждую итерацию (error_sr), в процентах. В каждой итерации определяется процентная ошибка на каждом шаге, из них считается средняя арифметическая для данной единичной итерации, а затем общая по алгоритму;

Максимальная средняя процентная ошибка алгоритма (error_max), в процентах;

Минимальная средняя процентная ошибка алгоритма (error_min), в процентах.

Для алгоритмов AR и ARIMA количество итеративных предсказаний одной и той же функции с различными (случайными) шумами равно 100. Для алгоритма SARIMA, как самого время- и ресурсозатратного, равно 5. Статистические данные по 1 функции представлены в таблице 1.

Таблица 1. Статистические данные по работе алгоритмов с функцией 1

Алгоритм	AR	ARIMA	SARIMA
Всего предсказаний	100	100	5
t_sr (сек)	0.22027860	3.11642737	65.39935479
t_max (сек)	0.40051556	12.80044198	77.02871537
t_min (сек)	0.17994404	0.76911712	57.46478176
r2_sr	0.98222739	-0.85392778	0.99165769
r2_min	0.90928004	-83.75445051	0.98962245
r2_max	0.99799784	0.99985268	0.99383161
error_sr (%)	4.48179555	25.36998262	3.04170699
error_max (%)	11.19494299	367.50673178	3.30353191
error_min (%)	1.21948126	0.42090374	2.77099197

Исходя из данных, представленных в таблице 1, можно заключить, что наибольшая точность достигнута алгоритмом SARIMA, однако и время его работы значительно выше, чем у конкурентов.

Совершим предсказания для функции номер 2 (рис. 6):

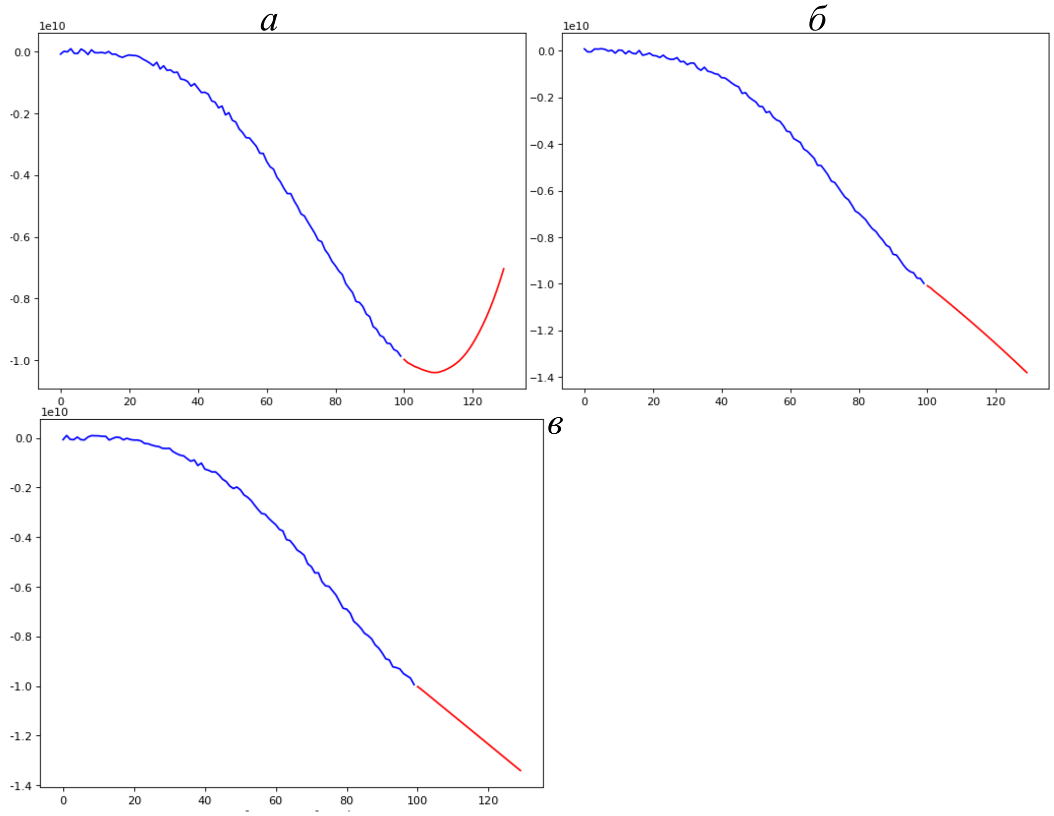


Рисунок 6. Результаты работы алгоритмов с функцией 1: а – алгоритм AR, б – алгоритм ARIMA, в – алгоритм SARIMA

Здесь синим цветом изображён тренировочный массив, а красным – предсказанный.

Как можно заметить визуально, только алгоритм AR справился с поставленной задачей. Это объясняется тем, что только алгоритм AR смог подобрать достаточно высокий коэффициент p, т.к. в реализации модели ARIMA и SARIMA максимальное значение любого из коэффициентов равняется 2.

Если брать верхнюю планку параметров выше, значительно возрастает время перебора, в модели ARIMA равное $O(n^3)$, а в модели SARIMA равное $O(n^6)$, где n – максимальное значение коэффициента, что заметно влияет на время работы. Следует отметить, что выше была приведена только сложность перебора вариантов для поиска лучшего коэффициента AIC, а никак не сложность самих алгоритмов.

Статистические данные по 2 функции представлены в таблице 2.

Таблица 2. Статистические данные по работе алгоритмов с функцией 2

Алгоритм	AR	ARIMA	SARIMA
Число итераций	100	100	5
t_sr (сек)	0.22374382	1.46801339	1.53091269
t_max (сек)	0.42654061	2.51700306	2.06208968
t_min (сек)	0.18153596	0.79759145	1.13660264
r2_sr	0.10173173	-9.31593106	-9.77157725
r2_min	-2.98998815	-16.66635001	-11.28300646
r2_max	0.99851780	-3.08891401	-8.66152646
error_sr (%)	6.71826448	25.17518669	26.49739998
error_max (%)	16.31536877	34.50479815	28.57494007
error_min (%)	0.34357648	15.13674995	24.76276281

Исходя из данных, представленных в таблице 2, можно заключить, что наибольшая точность достигнута алгоритмом AR, однако лишь при удачном стечении шумов. В противном случае – ни один из алгоритмов не способен точно предсказать данную функцию.

Совершим предсказания для функции номер 3 (рис. 7):

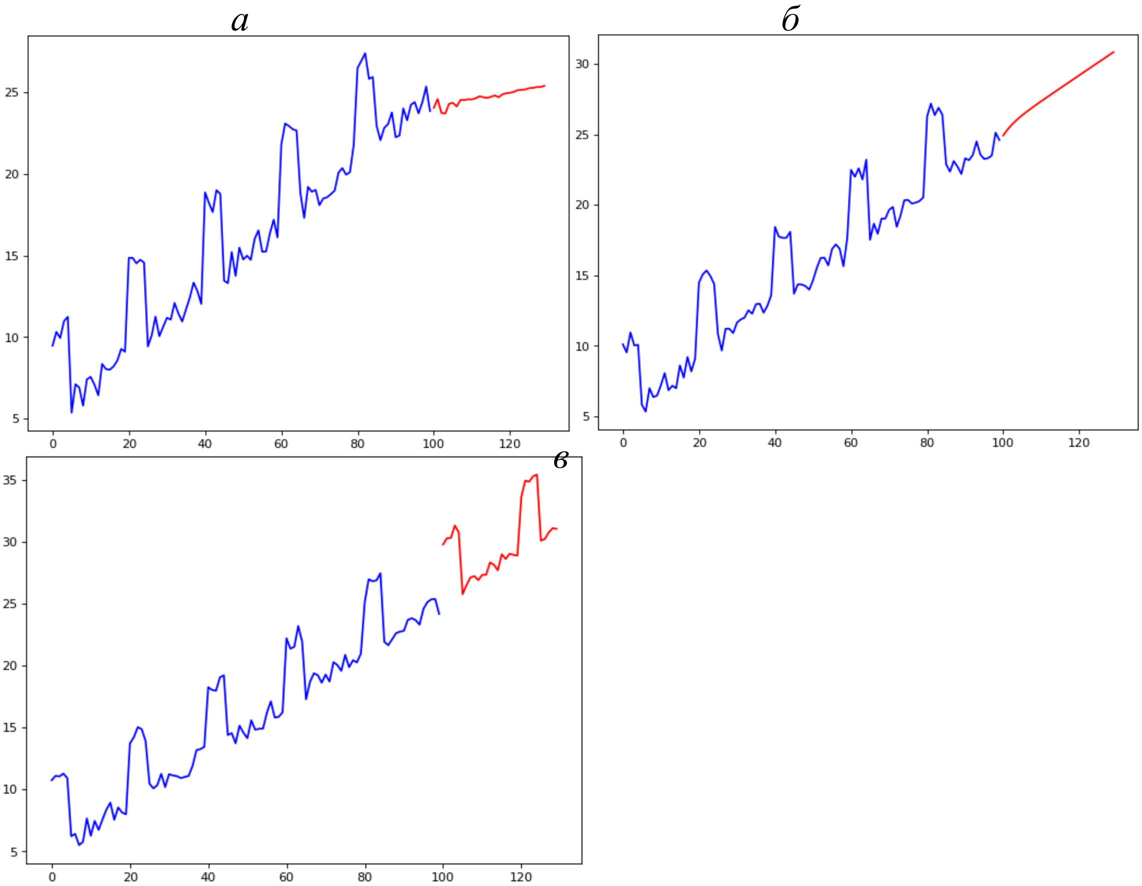


Рисунок 7. Результаты работы алгоритмов с функцией 1: а – алгоритм AR, б – алгоритм ARIMA, в – алгоритм SARIMA

Здесь синим цветом изображён тренировочный массив, а красным – предсказанный.

Как можно заметить визуально, для функции с явно выраженной сезонностью только алгоритм SARIMA справился с поставленной задачей. Это объясняется фундаментальными принципами алгоритмов: не существует такой единой функции, описывающей поведение данного графика, однако если разбить его на тренд и сезонность, и каждый из них задать функцией, а затем сложить на определённых участках, определённых сезонностью, то можно получить приближенный к реальному результат.

Алгоритм AR полностью не способен выполнить поставленную задачу, когда появляется сезонный элемент. Алгоритм ARIMA приводит значения временного ряда к трендовому значению, определяемому скользящей средней, и только алгоритм SARIMA правильно определяет и тренд, и сезонность.

Статистические данные по 3 функции представлены в таблице 3.

Таблица 3. Статистические данные по работе алгоритмов с функцией 3

Алгоритм	AR	ARIMA	SARIMA
Число итераций	100	100	5
t_sr (сек)	0.22413357	4.28670014	137.61584482
t_max (сек)	0.82197976	8.85632181	141.30327511
t_min (сек)	0.18328738	2.64672399	133.38742280

r2_sr	-2.26116077	-0.27244635	ош
r2_min	-3.80425760	-2.53849091	ош
r2_max	-1.23193524	0.36620451	0.98541255
error_sr (%)	13.09535186	6.40399434	ош
error_max (%)	16.98719683	14.76345878	ош
error_min (%)	10.02758848	4.81674540	0.89799236

К сожалению, где-то в используемой библиотеке для алгоритма SARIMA допущена ошибка, из-за которой иногда модель предсказывает принципиально неверные данные, «ломая» статистику. Например, для данной функции это могут быть случайные значения более $2.5 * 10^{67}$ (пример – рисунок 8). Для строк, которые зависят от этих данных, в таблице 3 прописано «ош». Если же их игнорировать, то алгоритм SARIMA показывает большую точность в предсказании временного ряда с периодической компонентой.

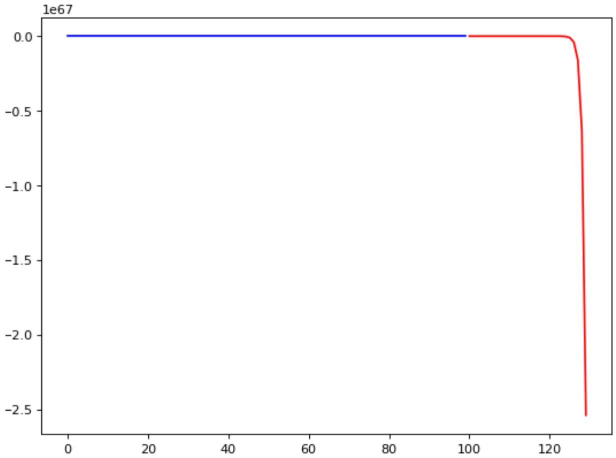
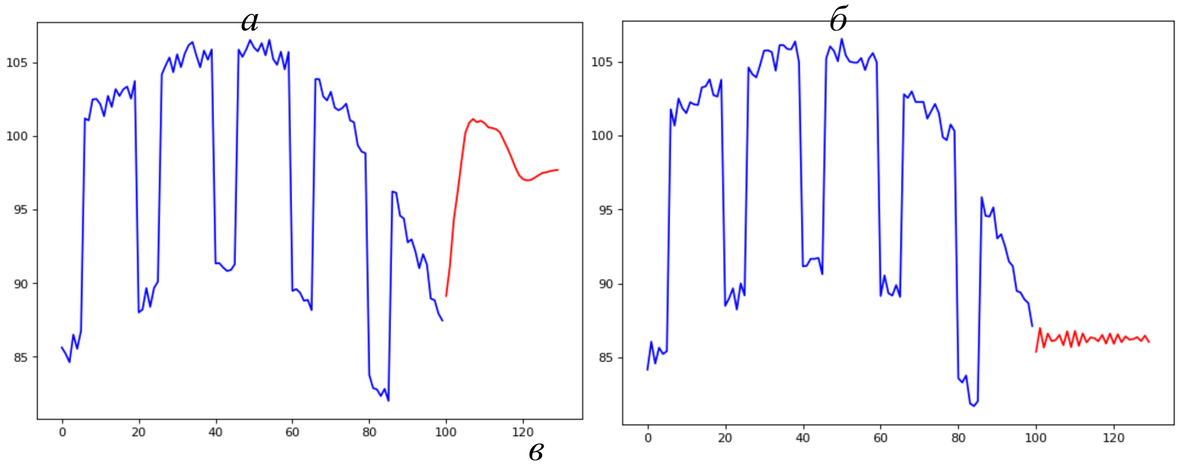


Рисунок 8. Критическая ошибка предсказания алгоритма SARIMA

Совершим предсказания для функции номер 4 (рис. 9):



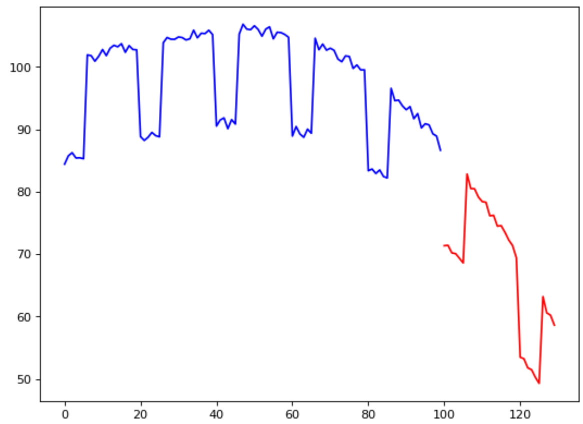


Рисунок 9. Результаты работы алгоритмов с функцией 1: а – алгоритм AR, б – алгоритм ARIMA, в – алгоритм SARIMA,

где синим цветом изображён тренировочный массив, а красным – предсказанный.

Очевидно, что только алгоритм SARIMA смог правильно предсказать поведение функции по всё тем же причинам, что и в функции 3. Однако, если степень в формуле (4) была бы больше 3, для достижения схожей точности было бы необходимо увеличить максимальные параметры для модуля подбора, что привело бы к ещё большему увеличению времени работы алгоритма.

Статистические данные по 4 функции представлены в таблице 4.

Таблица 4. Статистические данные по работе алгоритмов с функцией 4

Алгоритм	AR	ARIMA	SARIMA
Число итераций	100	100	5
t_sr (сек)	0.19996991	2.98662501	140.75375896
t_max (сек)	0.25020719	6.80050826	145.56701469
t_min (сек)	0.18257737	2.19863653	135.14111614
r2_sr	-8.07234198	-3.91056350	ош
r2_min	-8.21800155	-6.86239036	ош
r2_max	-7.93465113	-0.05477482	0.99506499
error_sr (%)	53.23552722	37.22239714	ош
error_max (%)	53.72036392	48.94741915	ош
error_min (%)	52.76484844	15.45873623	1.03436477

Если отбросить вновь возникшие ошибки в библиотеке SARIMA, вновь наиболее точным стал алгоритм SARIMA в связи с наличием малой степенной сезонности во временном ряде.

Заключение

В заключении хотелось бы отметить, что алгоритм AR быстро и весьма точно справляется с предсказанием стационарных временных рядов без элементов сезонности, тогда как алгоритм SARIMA работает гораздо дольше, но способен с большой точностью определять поведение временных рядов с элементами сезонности, если заранее известны параметры для алгоритма, или, если значения данных параметров достаточно малы, чтобы имелась возможность их предсказания перебором.

Список литературы:

1. Миллс, Т. К. Методы временных рядов для экономистов. Издательство Кембриджского университета – 1990г. – С. 78-94
2. Радченко, С. Г. Основные концепции множественного регрессионного анализа / С. Г. Радченко // Математические машины и системы. – 2013г. – №1 – С. 150-155
3. Модель авторегрессии. Python. URL: https://www.statsmodels.org/stable/_modules/statsmodels/tsa/ar_model.html (Дата обращения 30.11.2022)
4. Суворов, Н.В. Метод построения регрессионных моделей с динамическими структурными параметрами / Н. В. Суворов // Проблемы прогнозирования. – 2005г. – С. 143-155
5. Модель авторегрессии интегрированного скользящего среднего. Python. URL: https://tedboy.github.io/statsmodels_doc/_modules/statsmodels/tsa/arima_model.html (Дата обращения 30.11.2022)
6. AIC. Python. URL: https://pypi.org/project/RegscorePy/ (Дата обращения 30.11.2022)
7. Модель сезонной авторегрессии интегрированного скользящего среднего. Python. URL: https://www.statsmodels.org/stable/generated/statsmodels.tsa. arima. model.ARIMA.html (Дата обращения 30.11.2022)

References:

1. Mills, T. K. Time series methods for economists. Cambridge University Press – 1990. – P. 78-94
2. Radchenko, S. G. Basic concepts of multiple regression analysis / S. G. Radchenko // Mathematical machines and systems. – 2013 – №1 – P. 150-155
3. Autoregression model. Python. URL: https://www.statsmodels.org/stable/_modules/statsmodels/tsa/ar_model.html (date of access: 30.11.2022)
4. Suvorov, N.V. Method of constructing regression models with dynamic structural parameters / N. V. Suvorov // Problems of forecasting. – 2005 – P. 143-155
5. Autoregressive integrated moving average. Python. URL: https://tedboy.github .io/statsmodels_doc/_modules/statsmodels/tsa/ arima_model.html (date of access 30.11.2022)
6. AIC. Python. URL: https://pypi.org/project/RegscorePy/ (date of access 30.11.2022)
7. Seasonal autoregressive integrated moving average. Python. URL: https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html (date of access 30.11.2022)