

Применение временных рядов для анализа и предсказания курса валюты

Эрдман Александр Алексеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данном исследовании рассматривается применение трёх моделей временных рядов для анализа данных курса валют. Для реализации алгоритмов используется язык программирования Python. Результатом исследования является практическое применение реализуемых алгоритмов, а также сравнение их между собой с целью выявления более оптимального подхода для анализа поставленной задачи.

Ключевые слова: временные ряды, python, машинное обучение

The use of time series for currency exchange rate analysis and prediction

Erdman Alexander Alekseevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This study examines the application of three time series models to analyze currency exchange rate data. The Python programming language is used to implement the algorithms. The result of the research is the practical application of the implemented algorithms, as well as their comparison with each other in order to identify a more optimal approach for analyzing the task.

Keywords: time series, python, machine learning

1 Введение

1.1 Актуальность

Прогнозирование валютных курсов остается актуальной задачей в условиях нестабильности финансовых рынков. Точные прогнозы помогают минимизировать риски и оптимизировать управленческие решения. Современные методы анализа временных рядов, включая ARIMA, SARIMA и ARMA, позволяют учитывать характерные особенности финансовых данных - нестационарность, тренды и сезонность.

В данном исследовании анализируется эффективность различных подходов к прогнозированию на основе исторических данных курса валюты. Особое внимание уделяется сравнению моделей между собой в целях выявления оптимального алгоритма для поставленной задачи. Полученные

результаты могут быть применены для разработки систем финансового анализа и управления рисками.

1.2 Обзор исследований

С.Ж. Даулет в своем труде описал процесс извлечения данных из социальной сети с последующим анализом полученных данных [1]. А.О. Захаренков и Г.С. Евдокимова в совместном исследовании рассмотрели проблему выбора языка с реализацией компьютерной математики для задачи анализа больших наборов данных [2]. И.Х. Утакаева описала применение пакета статистического анализа Python для анализа данных рынка автомобилей [3]. М.Э. Чибирова в своей статье рассмотрел методику регрессионного анализа данных и разработки математических моделей, которые используются для прогнозирования процессов [4]. В.А. Мельникова и Д.А. Медведев посвятили исследование сравнительному анализу библиотек языка программирования Python для решения задач анализа данных [5].

1.3 Цель исследования

Целью исследования является практическая реализация моделей временных рядов для анализа и прогнозирования курса валют с помощью языка программирования Python.

2 Материалы и методы

Для решения поставленной задачи используется язык программирования Python, библиотеки Statsmodels, Matplotlib, NumPy и Pandas. В качестве среды разработки используется Google Colaboratory.

3 Результаты и обсуждения

Для анализа данных применяются различные модели, среди которых ARMA, ARIMA и SARIMA, каждая из которых имеет свои особенности и области применения.

Модель ARMA (Autoregressive Moving Average) является одной из базовых моделей для анализа стационарных временных рядов. Она сочетает в себе два компонента: авторегрессию (AR) и скользящее среднее (MA). Авторегрессионная часть модели (AR) описывает зависимость текущего значения ряда от его предыдущих значений, что позволяет учитывать влияние прошлых данных на текущее состояние. Скользящее среднее (MA) учитывает влияние прошлых ошибок прогноза, что помогает моделировать случайные колебания в данных. Модель ARMA эффективна для стационарных рядов, где статистические свойства, такие как среднее значение и дисперсия, остаются постоянными во времени. Однако, если ряд содержит тренд или другие нестационарные компоненты, требуется его предварительная обработка, например, дифференцирование.

Модель ARIMA (Autoregressive Integrated Moving Average) расширяет возможности ARMA, добавляя компонент интегрирования (I), который позволяет работать с нестационарными временными рядами. Интегрирование

заключается в дифференцировании ряда, то есть в вычислении разностей между последовательными наблюдениями, чтобы устранить тренд и сделать ряд стационарным. Модель ARIMA характеризуется тремя параметрами: p (порядок авторегрессии), d (порядок дифференцирования) и q (порядок скользящего среднего). Параметр d определяет, сколько раз нужно продифференцировать ряд для достижения стационарности. Модель ARIMA широко применяется для анализа финансовых данных, включая курсы валют, где часто присутствуют нестационарные компоненты, такие как тренды. Она позволяет учитывать как долгосрочные изменения, так и краткосрочные колебания, что делает её универсальным инструментом для прогнозирования.

Модель SARIMA (Seasonal Autoregressive Integrated Moving Average) является расширением ARIMA, которое учитывает сезонные компоненты временного ряда. Сезонность — это периодические колебания, которые повторяются через фиксированные интервалы времени, например, ежемесячно или ежеквартально. SARIMA включает дополнительные параметры для моделирования сезонных эффектов, что делает её особенно полезной для анализа данных с выраженной сезонностью, таких как курсы валют, которые могут зависеть от экономических циклов или других периодических факторов. Модель SARIMA позволяет более точно учитывать как долгосрочные тренды, так и сезонные колебания, что улучшает качество прогнозирования. Она характеризуется не только параметрами p , d и q , но и дополнительными сезонными параметрами (P , D , Q), которые описывают авторегрессию, дифференцирование и скользящее среднее для сезонной компоненты. Это делает SARIMA мощным инструментом для анализа сложных временных рядов, где присутствуют как тренды, так и сезонные закономерности. Модели ARMA, ARIMA и SARIMA позволяют анализировать и прогнозировать временные ряды с учётом трендов и сезонности. Их применение требует тестирования на реальных данных для выбора оптимальной модели.

Данные, представленные в CSV-файле, содержат информацию о курсе валюты за определённый период времени, начиная с 11 января 2022 года и заканчивая 31 августа 2022 года. Каждая строка в файле соответствует конкретной дате и содержит два основных столбца: дату в формате ДД.ММ.ГГГГ и значение курса валюты. Временной ряд является ежедневным, что позволяет проводить детальный анализ краткосрочных изменений. Однако стоит учитывать, что в данные могут быть включены пропуски, особенно в выходные и праздничные дни, когда финансовые рынки не работают. Курс валюты демонстрирует значительные колебания, варьируясь от минимального значения 51.1580 (зафиксированного 30 июня 2022 года) до максимального значения 120.3785 (зафиксированного 11 марта 2022 года). Такая волатильность может быть связана с внешними экономическими факторами, такими как изменения на мировых рынках, политические события или изменения в монетарной политике. Предварительный визуальный анализ позволяет предположить наличие как восходящих, так и нисходящих трендов, а также возможной сезонности, что требует дальнейшего изучения.

Основная цель анализа заключается в выявлении закономерностей в изменении курса валюты, построении модели, которая сможет объяснить его динамику, и прогнозировании будущих значений. Для этого будут использованы различные методы анализа временных рядов, такие как ARMA, ARIMA, SARIMA.

Первым применяемым методом для анализа и прогнозирования данных является ARMA. В начале его реализации происходит импорт необходимых библиотек и модулей (рис. 1). NumPy и Pandas используются для работы с массивами и таблицами данных, что позволяет эффективно обрабатывать временные ряды. Matplotlib применяется для визуализации данных, что помогает наглядно оценить структуру временного ряда. Библиотека statsmodels предоставляет инструменты для анализа временных рядов, включая функции для построения графиков автокорреляции (ACF и PACF), а также для создания и обучения моделей ARIMA. Эти инструменты необходимы для анализа и прогнозирования временных рядов.

```
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf, plot_predict
from statsmodels.tsa.stattools import adfuller, pacf, acf
from statsmodels.tsa.arima.model import ARIMA
```

Рисунок 1. Импорт модулей

Следующим шагом является загрузка данных (рис. 2).

```
[2] data = pd.read_csv("/content/drive/DataKurs.csv",
                    delimiter=";", header=0, names=["Дата", "Курс", "Ипотека"])
data = data[::-1].reset_index().drop(labels="index", axis=1)
data.head()
```

	Дата	Курс	Ипотека
0	11.01.2022	75.1315	9.84
1	12.01.2022	74.8355	9.84
2	13.01.2022	74.5277	9.84
3	14.01.2022	74.5686	9.84
4	15.01.2022	75.7668	9.84

Рисунок 2. Загрузка данных

Данные загружаются из CSV-файла, содержащего информацию о курсе валют и ипотеке. Данные о ипотеке являются лишними, так как данные были взяты из свободного источника. Файл загружается с помощью функции `pd.read_csv`, при этом данные переворачиваются (`[::-1]`), чтобы они шли в хронологическом порядке. Это важно для корректного анализа временного ряда. После переворота данных сбрасывается индекс с помощью `reset_index()`, чтобы избежать путаницы. Первые пять строк данных выводятся на экран с помощью `data.head()`, что позволяет быстро ознакомиться с их структурой.

Для анализа временного ряда строится график автокорреляции (ACF) (рис. 3).

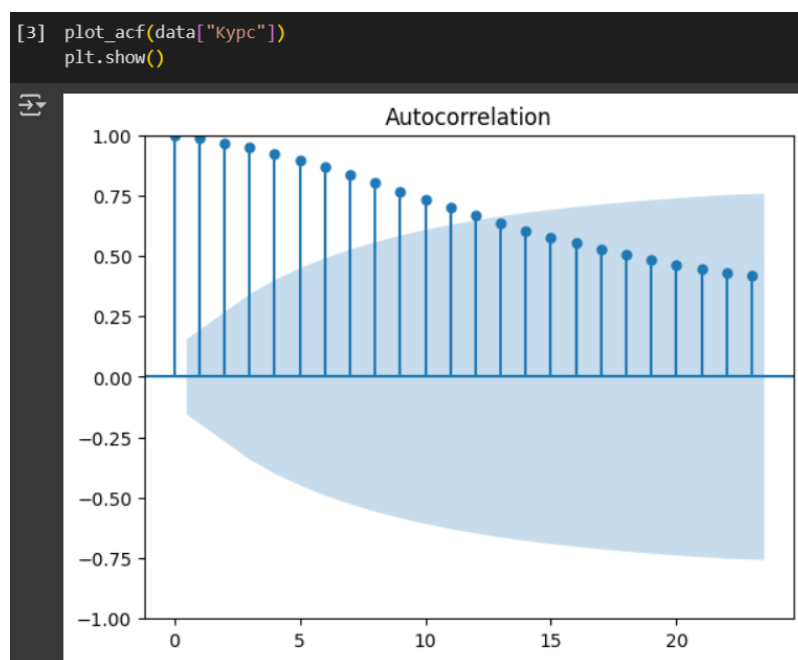


Рисунок 3. График автокорреляции

ACF показывает, насколько текущие значения ряда зависят от предыдущих. Это помогает выявить наличие сезонности, тренда или других закономерностей в данных. График строится с помощью функции `plot_acf` из библиотеки `statsmodels`. Визуализация ACF позволяет сделать первые выводы о структуре временного ряда и определить, какие параметры модели ARMA могут быть полезны.

Для более глубокого анализа временного ряда необходимо построить график первой производной (рис. 4).

Первая производная показывает изменения в тренде ряда, что помогает выявить моменты резких изменений. Также строится график автокорреляции для первой производной. Это позволяет понять, насколько сильно текущие изменения зависят от предыдущих. Графики реализуются с помощью `matplotlib`, что делает анализ наглядным и удобным для интерпретации.

Модель ARMA использует два целочисленных параметра: p и q . Порядок авторегрессии - p (AR). Его можно интерпретировать как выражение «элемент ряда будет близок к X , если предыдущие p элементов были близки к X » (рис. 5). Для этого вычисляется частичная автокорреляция (PACF) для первых двух производных временного ряда. На основе PACF выбирается оптимальный порядок авторегрессии.

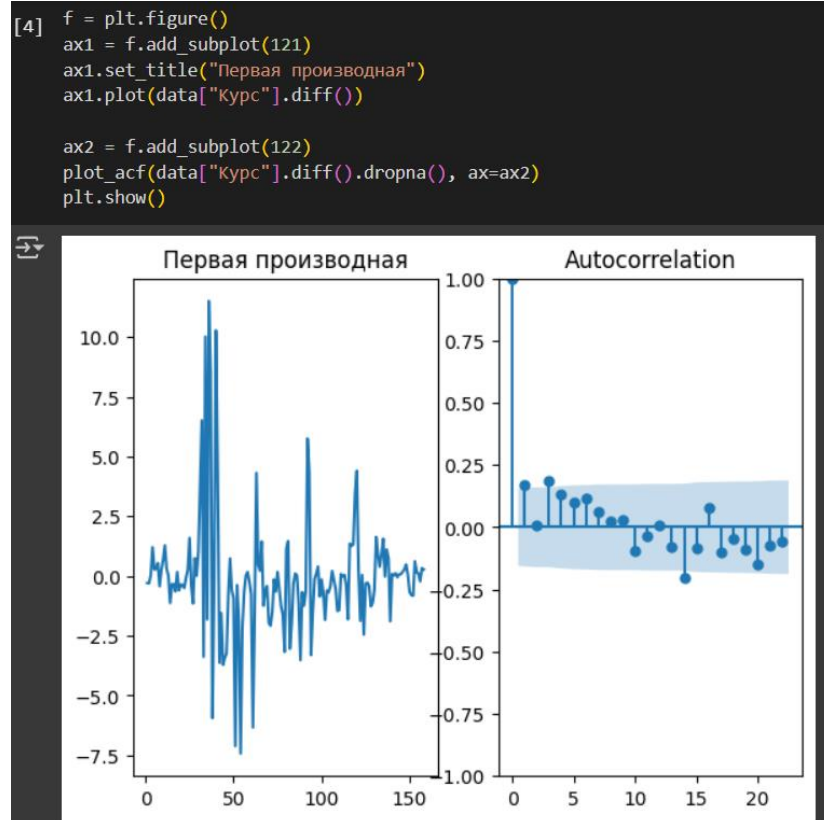


Рисунок 4. Построение графика первой производной

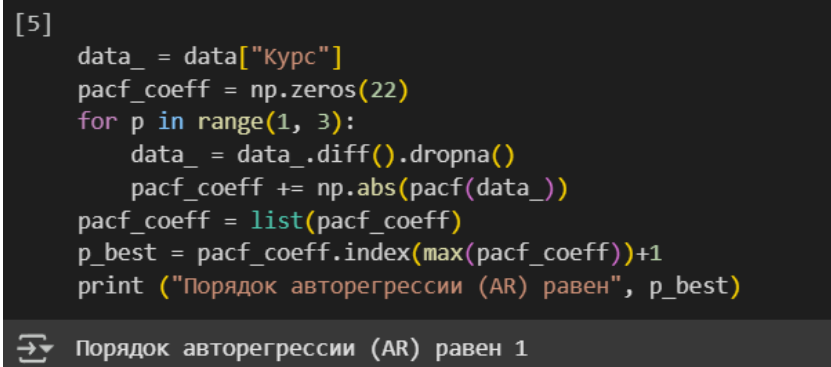


Рисунок 5. Определение порядка авторегрессии

Порядок скользящего среднего- q (МА), который позволяет установить погрешность модели как линейную комбинацию наблюдавшихся ранее значений ошибок (рис. 6).

После определения параметров p и q строится модель ARIMA (частный случай ARMA) (рис. 7). Модель обучается на данных с помощью функции `ARIMA.fit()`. После обучения выводится сводка по модели, которая включает статистические показатели, такие как коэффициенты модели и их значимость. Это позволяет оценить качество модели и её пригодность для прогнозирования.

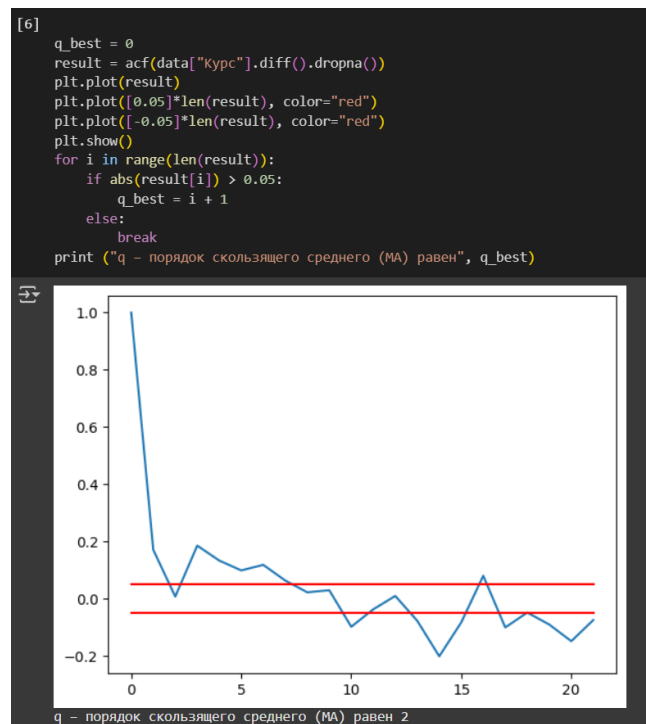


Рисунок 6. Определение порядка скользящего среднего

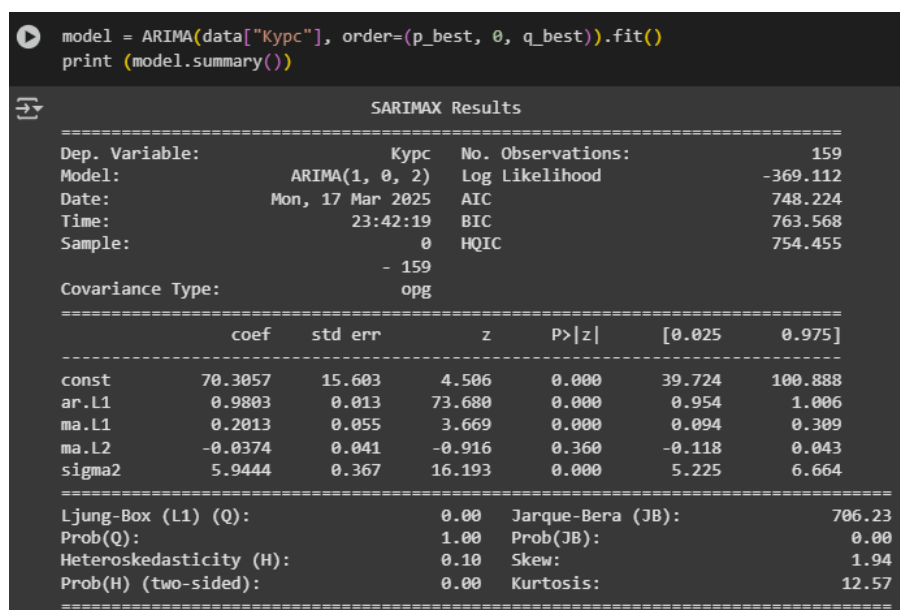


Рисунок 7. Построение частного случая ARMA

Для улучшения качества модели выполняется перебор различных комбинаций параметров p и q . Для каждой комбинации вычисляется значение AIC (Akaike Information Criterion), которое используется для выбора лучшей модели. Чем меньше значение AIC, тем лучше модель. Этот процесс позволяет найти оптимальные параметры модели, которые минимизируют ошибку прогнозирования (рис. 8).

```

8] aic_best = model.aic
   coeff_best = (p_best, 0, q_best)
   for p in range(1,5):
       for q in range(1,5):
           model_ = ARIMA(data["Курсы"], order=(p, 0, q)).fit()
           aic = model_.aic
           if aic < aic_best:
               coeff_best = (p, 0, q)
               aic_best = aic
   print ("Наилучшие коэффициенты:", coeff_best)

/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/statespace/sar:
warn('Non-stationary starting autoregressive parameters')
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/statespace/sar:
warn('Non-invertible starting MA parameters found.')
/usr/local/lib/python3.11/dist-packages/statsmodels/base/model.py:607:
warnings.warn("Maximum Likelihood optimization failed to ")
/usr/local/lib/python3.11/dist-packages/statsmodels/base/model.py:607:
warnings.warn("Maximum Likelihood optimization failed to ")
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/statespace/sar:
warn('Non-stationary starting autoregressive parameters')
/usr/local/lib/python3.11/dist-packages/statsmodels/base/model.py:607:
warnings.warn("Maximum Likelihood optimization failed to ")
/usr/local/lib/python3.11/dist-packages/statsmodels/base/model.py:607:
warnings.warn("Maximum Likelihood optimization failed to ")
/usr/local/lib/python3.11/dist-packages/statsmodels/base/model.py:607:
warnings.warn("Maximum Likelihood optimization failed to ")
Наилучшие коэффициенты: (2, 0, 3)
/usr/local/lib/python3.11/dist-packages/statsmodels/base/model.py:607:
warnings.warn("Maximum Likelihood optimization failed to ")

```

Рисунок 8. Перебор коэффициентов "по сетке"

После выбора оптимальных параметров модель ARIMA обучается на данных, за исключением последних 10 значений. Это позволяет оценить качество прогноза на реальных данных (рис. 9). Прогноз на 10 шагов вперед строится с помощью функции `get_forecast()`. Результаты прогноза визуализируются на графике вместе с исходными данными. Это позволяет наглядно оценить, насколько хорошо модель справляется с прогнозированием. В результате можно видеть, что прогнозируемая цена имеет отклонение от реальной цены, но не большое.

Для оценки точности прогноза используется метрика SMAPE (Symmetric Mean Absolute Percentage Error) (рис. 10). SMAPE измеряет относительную ошибку прогноза и выражается в процентах. Чем меньше значение SMAPE, тем точнее прогноз. Ошибка вычисляется между фактическими значениями и прогнозом для последних 10 точек данных. Это позволяет количественно оценить качество модели и её пригодность для практического использования. Ошибка получилась 37.9% на десять значений. Ошибка для такой простой модели как ARMA является нормальным показателем.

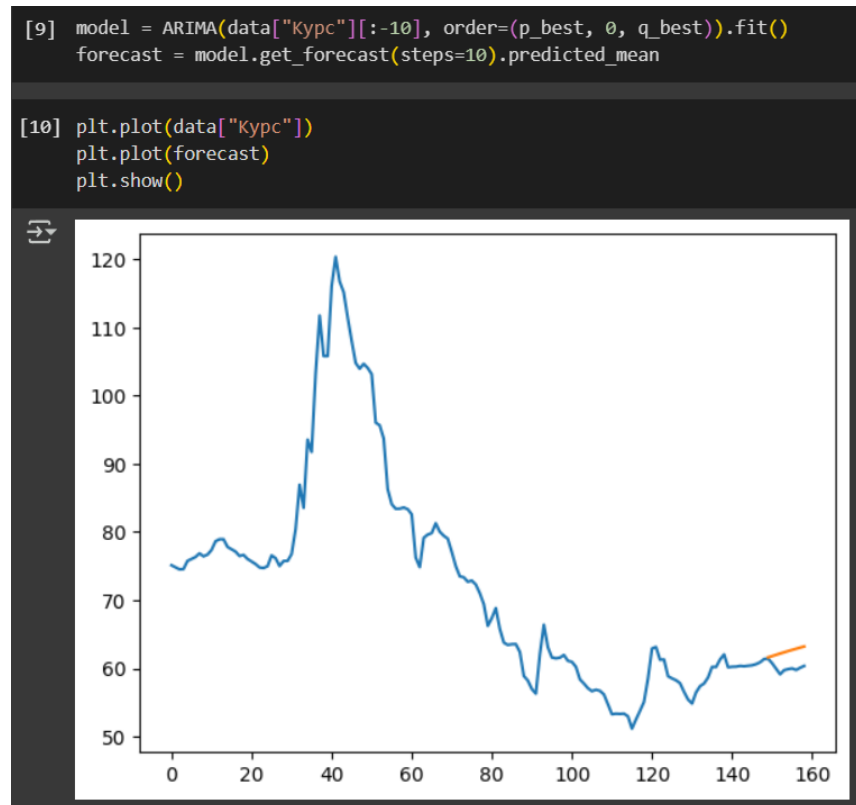


Рисунок 9. Расчёт и визуализация результата

```
[11] def smape(A, F):
    return 100*np.sum(2 * np.abs(F - A) / (np.abs(A) + np.abs(F)))
```

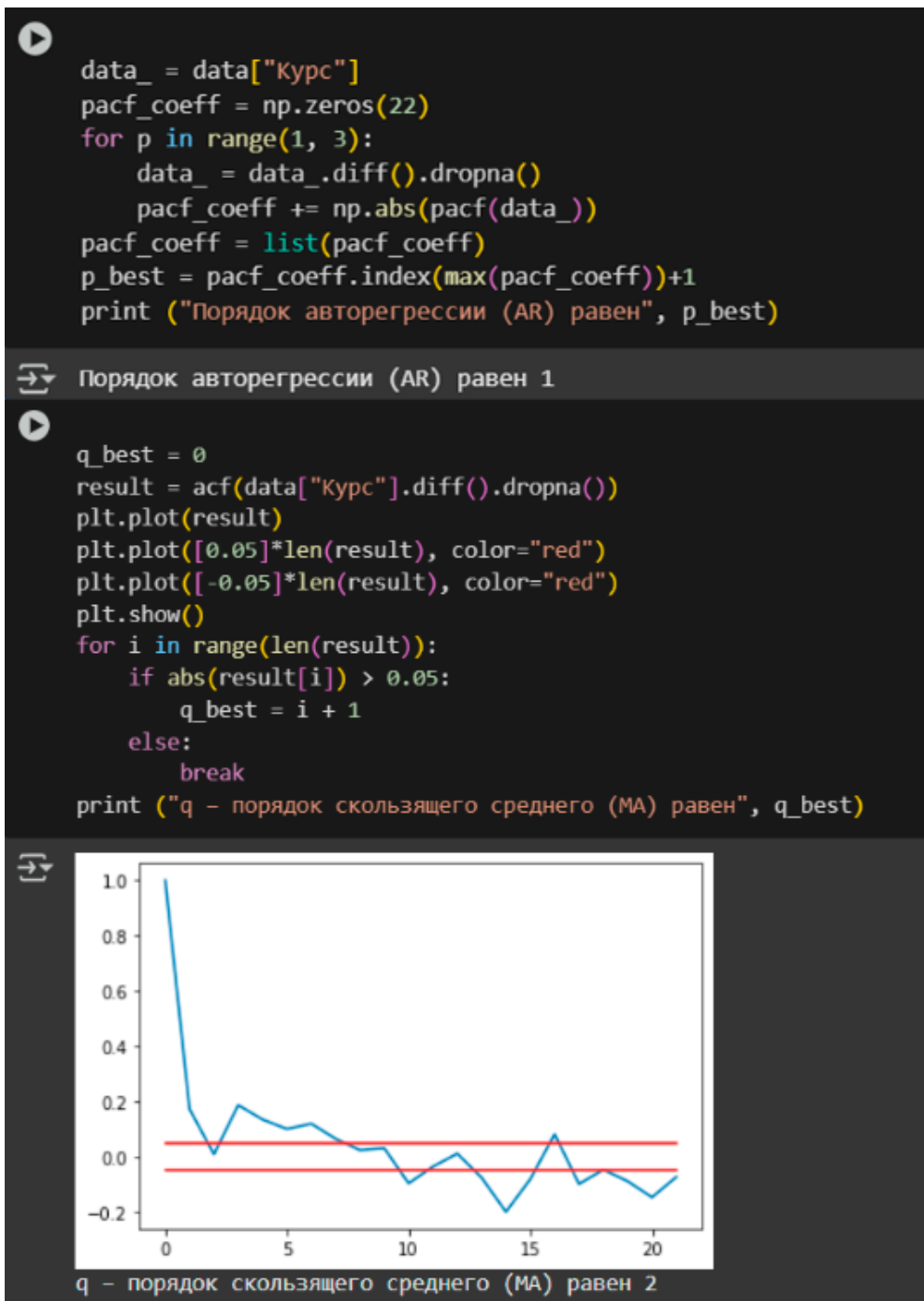
```
[12] print ("Ошибка модели:", smape(data["Курс"][-10:], forecast))
```

Ошибка модели: 37.91886763247961

Рисунок 10. Оценка ошибки

Следующим рассматриваемым методом анализа и прогнозирования является ARIMA. Аналогично вышеописанному методу импортируются необходимые модули и загружаются данные.

Модель ARIMA отличается от ARMA в первую очередь большим количеством параметров p , d и q . Здесь аналогично модели ARMA имеются параметры p и q -порядок авторегрессии (AR) и порядок скользящего среднего (MA), расчёт которых отличается от предыдущего метода (рис.11).

Рисунок 11. Вычисление параметров p и q ARIMA модели

В отличие от предыдущего метода, здесь используется PACF для определения p и ACF для определения q .

Параметр d - порядка интегрирования (I) вычисляется следующим образом (рис. 12).

```

data_ = data["Курс"]
d_best = 0
for d in range(1, 5):
    data_ = data_.diff().dropna()
    result = adfuller(data_)
    print ("p-значение для " + str(d) + "-производной ряда", result[1])
    if d_best == 0 and result[1] < 0.05:
        d_best = d
print ("d - порядок интегрирования (I) равен", d_best)

```

p-значение для 1-производной ряда 1.6475432053954194e-06
 p-значение для 2-производной ряда 1.0121605641452575e-15
 p-значение для 3-производной ряда 7.120990020397908e-11
 p-значение для 4-производной ряда 2.3156793235712853e-16
 d - порядок интегрирования (I) равен 1

Рисунок 12. Вычисление порядка интегрирования

Для определения порядка интегрирования (I) используется тест Дики-Фуллера. Этот тест проверяет стационарность ряда. Если p-значение меньше 0.05, ряд считается стационарным. Порядок интегрирования (d) выбирается как минимальное количество разностей, необходимых для достижения стационарности. Этот блок кода добавляет новый шаг в анализ, который отсутствовал в предыдущем коде.

После определения параметров p, d и q строится модель ARIMA, обучение которой происходит на данных с помощью вышеупомянутой функции ARIMA.fit(). После данной функции следует перебор коэффициентов "по сетке", который расширяется за счёт добавление нового параметра d.

После проделанных действий можно видеть результат вычисления модели ARIMA (рис. 13). На данных результатах видно, что модель стала точнее предсказывать курс валют, а также более чем в 2 раза уменьшилась ошибка модели.

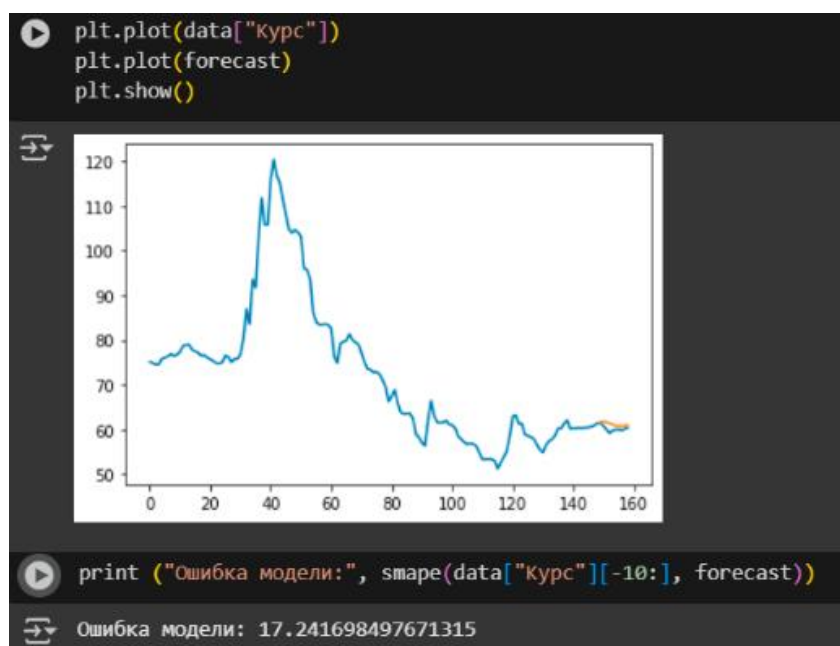


Рисунок 13. Результат работы модели ARIMA

Следующая модель SARIMA должна показать ещё более точные результаты, так как сама модель является более сложной и рассчитывается с использованием ещё большего числа параметров. Для её реализации подключается модуль SARIMAX из библиотеки statsmodels, который расширяет ARIMA за счет учета сезонности. Общий принцип программирования модели схож с предыдущими двумя. Особенность и различие проявляется в учёте сезонности (период сезонности). Помимо периода сезонности используются также следующие параметры для улучшения точности: P – порядок сезонной авторегрессии; D – порядок сезонного интегрирования; Q – порядок сезонного скользящего среднего.

При построении модели используются следующие параметры $p=2$, $d=1$, $q=3$ для несезонной части и $P=1$, $D=1$, $Q=1$, $T=7$ для сезонной части (рис. 14).

```
model = SARIMAX(data["Kypc"], order=(2, 1, 3), seasonal_order=(1,1,1,7)).fit()
print(model.summary())
```

SARIMAX Results

Dep. Variable:		Kypc		No. Observations:		159	
Model:		SARIMAX(2, 1, 3)x(1, 1, [1], 7)		Log Likelihood		-354.272	
Date:		Mon, 17 Mar 2025		AIC		724.544	
Time:		23:36:24		BIC		748.682	
Sample:		0		HQIC		734.350	
Covariance Type:		opg					
	coef	std err	z	P> z	[0.025	0.975]	
ar.L1	1.4686	0.070	20.923	0.000	1.331	1.606	
ar.L2	-0.7487	0.076	-9.903	0.000	-0.897	-0.601	
ma.L1	-1.3282	0.091	-14.623	0.000	-1.506	-1.150	
ma.L2	0.4566	0.109	4.179	0.000	0.242	0.671	
ma.L3	0.3045	0.052	5.865	0.000	0.203	0.406	
ar.S.L7	0.0085	0.089	0.096	0.923	-0.165	0.182	
ma.S.L7	-0.9999	161.770	-0.006	0.995	-318.064	316.064	
sigma2	5.4245	877.350	0.006	0.995	-1714.149	1724.999	

Ljung-Box (L1) (Q):	0.08	Jarque-Bera (JB):	212.32
Prob(Q):	0.77	Prob(JB):	0.00
Heteroskedasticity (H):	0.10	Skew:	0.89
Prob(H) (two-sided):	0.00	Kurtosis:	8.53

Рисунок 14. Результат построения модели SARIMA

Для улучшения качества модели выполняется перебор различных комбинаций параметров P, D и Q для сезонной части модели. Для каждой комбинации вычисляется значение AIC (Akaike Information Criterion), которое используется для выбора лучшей модели. Данный блок кода аналогичен схожему предыдущему, но здесь перебираются параметры сезонной части модели SARIMA.

После выявления лучших комбинаций параметров происходит обучение на данных и соответственно предсказание последних 10 значений (рис. 15). На графике явно видно, что результат предсказания стал ещё более точным. Сразу после этого происходит вычисление ошибки, результат которой равен 7.05, что существенно отличается от предыдущих рассматриваемых моделей.

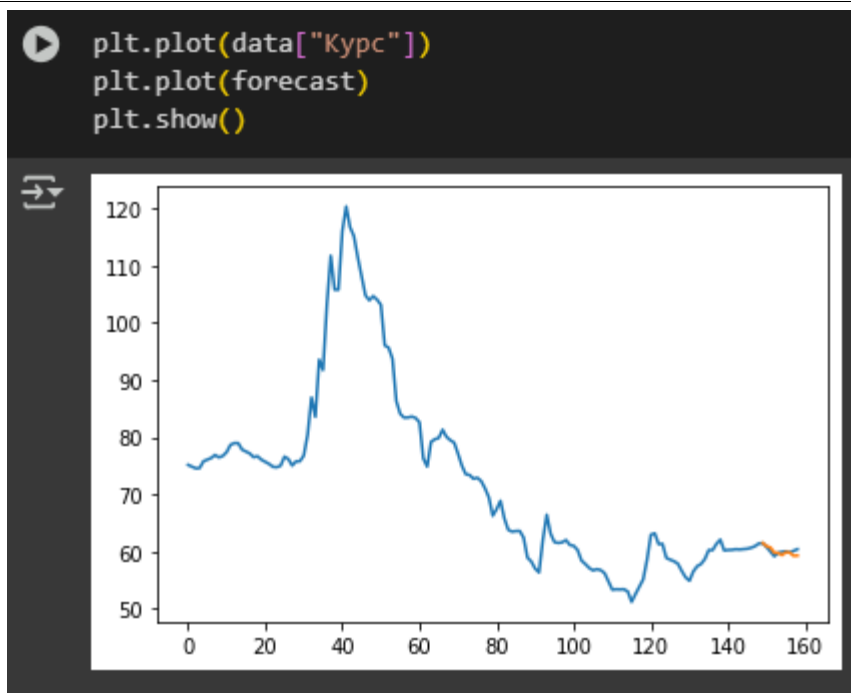


Рисунок 15. Наглядный результат предсказания модели SARIMA

Модели ARMA, ARIMA и SARIMA были применены для анализа временного ряда курса валют, и каждая из них продемонстрировала различные результаты, что позволяет сделать выводы о применимости этих моделей в зависимости от характеристик данных. Начнём с модели ARMA (Autoregressive Moving Average), которая сочетает в себе авторегрессионную компоненту (AR) и компоненту скользящего среднего (MA). Эта модель предназначена для работы со стационарными временными рядами, где отсутствуют тренд и сезонность. Однако в данном случае ARMA показала наихудшие результаты: высокие значения информационных критериев AIC (748.224) и BIC (763.568), а также значительную ошибку ($\sigma^2 = 5.9444$). Это свидетельствует о том, что модель недостаточно хорошо описывает данные. Тест Ljung-Box (Q) показал, что остатки модели не коррелированы ($\text{Prob}(Q) = 1.00$), что является положительным моментом, однако тест Jarque-Bera ($JB = 706.23$, $\text{Prob}(JB) = 0.00$) выявил ненормальность распределения остатков. Кроме того, наличие гетероскедастичности ($H = 0.10$, $\text{Prob}(H) = 0.00$) указывает на нестабильность модели. Таким образом, ARMA не подходит для анализа данного временного ряда, что, связано с наличием тренда и сезонности в данных.

Перейдём к модели ARIMA (Autoregressive Integrated Moving Average), которая расширяет ARMA за счёт добавления интегрирования (I) для работы с нестационарными рядами. Эта модель способна учитывать тренд в данных, что делает её более гибкой по сравнению с ARMA. Результаты ARIMA оказались лучше, чем у ARMA: значения AIC (737.987) и BIC (750.237) снизились, а ошибка ($\sigma^2 = 5.9406$) немного уменьшилась. Тест Ljung-Box (Q) также показал отсутствие автокорреляции в остатках ($\text{Prob}(Q) = 0.91$), что является хорошим признаком. Однако тест Jarque-Bera ($JB = 296.55$, $\text{Prob}(JB) = 0.00$) по-прежнему указывает на ненормальность распределения остатков, а

гетероскедастичность ($H = 0.11$, $\text{Prob}(H) = 0.00$) остаётся проблемой. ARIMA лучше подходит для нестационарных рядов с трендом, но без сезонности. В данном случае её результаты улучшились по сравнению с ARMA, но всё ещё не являются идеальными, что свидетельствует о наличии сезонных компонентов в данных, которые ARIMA не учитывает.

Наиболее успешной моделью оказалась SARIMA (Seasonal Autoregressive Integrated Moving Average), которая расширяет ARIMA за счёт добавления сезонных компонентов. Эта модель способна учитывать как тренд, так и сезонность, что делает её наиболее подходящей для сложных временных рядов. SARIMA показала наилучшие результаты: значения AIC (724.544) и BIC (748.682) значительно ниже, чем у ARMA и ARIMA, а ошибка ($\sigma^2 = 5.4245$) минимальна. Тест Ljung-Box (Q) подтвердил отсутствие автокорреляции в остатках ($\text{Prob}(Q) = 0.77$), что является важным критерием качества модели. Однако тест Jarque-Bera ($JB = 212.32$, $\text{Prob}(JB) = 0.00$) по-прежнему указывает на ненормальность распределения остатков, а гетероскедастичность ($H = 0.10$, $\text{Prob}(H) = 0.00$) остаётся проблемой, хотя и в меньшей степени, чем у ARMA и ARIMA. Успех SARIMA обусловлен наличием сезонных колебаний в данных, которые эта модель смогла учесть.

Сравнивая модели, можно прийти к следующим выводам об описанных моделях.

ARMA подходит только для стационарных временных рядов без тренда и сезонности. В данном случае она показала наихудшие результаты, что указывает на нестационарность данных и наличия сезонных компонентов.

ARIMA лучше подходит для нестационарных рядов с трендом, но без сезонности. Её результаты улучшились по сравнению с ARMA, но всё ещё не являются идеальными, что может свидетельствовать о наличии сезонности.

SARIMA является наиболее подходящей моделью для анализа данного временного ряда, так как она учитывает как тренд, так и сезонность. Её успех подтверждается минимальными значениями AIC и BIC, а также наименьшей ошибкой.

4 Выводы

В рамках исследования был проведён анализ временного ряда курса валют с использованием моделей ARMA, ARIMA и SARIMA. Модель ARMA, предназначенная для стационарных рядов без тренда и сезонности, показала наихудшие результаты. ARIMA, учитывающая тренд, продемонстрировала улучшенные результаты, но её точность оставалась ограниченной из-за сезонности. Наилучшие результаты показала модель SARIMA, которая учитывает как тренд, так и сезонность, что подтверждается минимальными значениями информационных критериев и наименьшей ошибкой. Таким образом, SARIMA является наиболее подходящей моделью для анализа данного временного ряда, подчеркивая важность учёта сезонных колебаний. Этот опыт подтвердил, что выбор модели должен основываться на характеристиках данных, а SARIMA — эффективный инструмент для анализа сложных временных рядов с трендом и сезонностью.

Библиографический список

1. Даулет С.Ж. Извлечение данных в Twitter, предварительная обработка и анализ настроек с использованием Python 3.0 // Молодой ученый. 2020. № 13 (303). С. 260-264.
2. Захаренков А.О., Евдокимова Г.С. Сравнительный анализ языков Ruby, R и Python в вопросах анализа данных на примере задачи кластеризации // Системы компьютерной математики и их приложения. 2020. № 21. С. 38-44.
3. Утакаева И.Х. Применение пакета статистического анализа Python для анализа данных автомобильного рынка // Вестник Алтайской академии экономики и права. 2019. № 2-2. С. 346-351.
4. Чибирова М.Э. Анализ данных и регрессионное моделирование с применением языков программирования Python и R // Научные записки молодых исследователей. 2019. Т. 7. № 2. С. 37-45.
5. Мельникова В.А., Медведев Д.А. Анализ больших данных с использованием Python // Труды Братского государственного университета. Серия: Естественные и инженерные науки. 2019. Т. 1. С. 46-49.