**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ прикладной математики и информатики**

**Кафедра математического моделирования и анализа данных**

КАРПОВИЧ

Артём Дмитриевич

**ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ АВТОРЕГРЕССИОННЫХ МОДЕЛЕЙ**

Курсовая работа

Студента 4 курса 7 группы

|  |  |
| --- | --- |
|  | Научный руководитель:  кандидат физ.-мат. наук,  доцент В.И. Лобач |

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ прикладной математики и информатики**

**Кафедра математического моделирования и анализа данных**

**ЗАДАНИЕ**

**НА КУРСОВУЮ РАБОТУ**

студенту Карповичу А.Д.

1. Тема курсовой работы

«ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ АВТОРЕГРЕССИОННЫХ МОДЕЛЕЙ».

2. Исходные данные к курсовой работе

2.1. Кувайскова, Ю. Е Статистические методы прогнозирования / Ю. Е. Кувайскова, В. Н. Клячкин/ – Минск: УГТУ, 2019.

2.2. Ивченко, Г. И. Математическая статистика: Учеб. Пособие для втузов / Ивченко Г. И., Медведев Ю. И./ – Москва: МВТУ им. Н. Э. Баумана, 1984.

2.3. Грэйнджер, Клайв У. Дж. Эконометрический анализ временных рядов / Клайв У. Дж. Грэйнджер.

2.4. Четыркин, Е. М. Статистические методы прогнозирования / Четыркин Е. М. // издание «Статистика». – 1975.

3. Перечень подлежащих разработке вопросов:

3.1. Подготовить обзор методов прогнозирования временных рядов.

3.2. Подготовить математическое описание авторегрессионных моделей прогнозирования временных рядов.

3.3. Выполнить программную реализацию алгоритма.

3.4. Сравнить результаты прогнозирования временных рядов с помощью моделей одномерной авторегрессии и векторной авторегресии.

3.6. Подготовить отчет по курсовому проекту.

4. Дата выдачи задания: 18.09.2024.

5. Срок сдачи законченной курсовой работы: 20.12.2024.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. И. Лобач

Подпись обучающегося \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Д. Карпович

Дата 20.12.2024

Оглавление

[ВВЕДЕНИЕ 4](#_Toc185459862)

[1. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ 6](#_Toc185459863)

[1.1 Задача прогнозирования временного ряда 6](#_Toc185459864)

[1.2 Методы машинного обучения 6](#_Toc185459865)

[1.3 Методы решения переобучения 10](#_Toc185459866)

[1.3.1 Уменьшение сложности модели 10](#_Toc185459867)

[1.3.2 Регуляризация 10](#_Toc185459868)

[2. ПРОГНОЗИРОВАНИЕ ВРЕМЕННОГО РЯДА НА ОСНОВЕ МОЕДЕЛЕЙ АВТОРЕГРЕСИИ 12](#_Toc185459869)

[2.1. Общая информация об авторегрессионных моделях 12](#_Toc185459870)

[2.2. Прогнозирование по модели авторегресии 12](#_Toc185459871)

[2.2. Прогнозирование по модели скользящего среднего 14](#_Toc185459872)

[2.3. Модели авторегрессии – проинтегрированного скользящего среднего для нестационарных временных рядов 15](#_Toc185459873)

[2.4. Модели векторной авторегрессии 17](#_Toc185459874)

[3. ПРОГНОЗИРОВАНИЕ ВРЕМЕНОГО РЯДА НА ОСНОВЕ ОДНОМЕРНОЙ АВТОРЕГРЕССИОННОЙ МОДЕЛИ 19](#_Toc185459875)

[3.1. Подготовка данных 19](#_Toc185459876)

[3.2. Базовое решение 22](#_Toc185459877)

[3.3. Прогнозирование с помощью ARIMA 23](#_Toc185459878)

[3.4. Байесовский подход 27](#_Toc185459879)

[3.4. Вывод 31](#_Toc185459880)

[4. ПРОГНОЗИРОВАНИЕ ВРЕМЕНОГО РЯДА НА ОСНОВЕ ВЕКТОРНОЙ АВТОРЕГРЕССИОННОЙ МОДЕЛИ 32](#_Toc185459881)

[4.1. Отбор признаков 32](#_Toc185459882)

[4.2. Подготовка данных 34](#_Toc185459883)

[4.3. Построение прогноза 36](#_Toc185459884)

[4.4. Вывод 38](#_Toc185459885)

[ЗАКЛЮЧЕНИЕ 39](#_Toc185459886)

[СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ 40](#_Toc185459887)

[Приложение 42](#_Toc185459888)

# ВВЕДЕНИЕ

В современном мире большое количество явлений можно представить в виде последовательности наблюдений, измеренных в разные моменты времени и упорядоченных в хронологическом порядке, такие последовательности называются временными рядами. Прогнозирование временных рядов играет важную роль во многих областях, включая финансы, экономику, климатологию, медицину и другие. Так, можно представить, например, дорожный трафик в виде временного ряда, прогнозирование которого позволит решить многие задачи логистики для различных служб, помимо того, в виде временных рядов можно представить и метеоданные, которые серьезно влияют на эффективность как работы человека, так и на сокращение человеческих жертв до минимума в случае природных катаклизмов.

Анализ временных рядов включает в себя исследование статистических свойств ряда, выявление трендов, сезонности и других особенностей, а также построение моделей для прогнозирования будущих значений и выявления структуры временного ряда. Выявление структуры временного ряда необходимо для того, чтобы построить математическую модель того явления, которое является источником анализируемого временного ряда. Прогноз будущих значений временного ряда используется для эффективного принятия решений.

Временные ряды состоят из двух элементов [5]:

* периода времени, за который или по состоянию на который приводятся числовые значения;
* числовых значений того или иного показателя, называемых уровнями ряда.

Перед тем, как применять методы прогнозирования к временному ряду, необходимо провести анализ временного ряда для выявления тенденций и закономерностей, не заметных при поверхностном знакомстве с данными. Для анализа временных рядов применяются три основных метода: стационарность, автокорреляция и спектральный анализ.

Стационарный временной ряд – это ряд, чьи статистические свойства не меняются со временем. Формально, временной ряд считается стационарным, если выполнены следующие свойства [5]:

* стационарность по среднему: среднее значение ряда не зависит от времени и остается постоянным на протяжении всего ряда;
* стационарность по дисперсии: дисперсия ряда не зависит от времени и остается постоянной на протяжении всего ряда;
* стационарность по автоковариации: автоковариация между значениями ряда на разных временных отрезках зависит только от длины этих отрезках, но не от их положения во времени.

Стационарность является важным свойством при анализе и моделировании, так как она позволяет применять статистические методы и модели, которые предполагают постоянство статистических свойств данных. В стационарном ряде проще выявить закономерности, построить модели и делать прогнозы на основе его статистических свойств, чем в нестационарном ряде, где эти свойства могут меняться со временем. Для определения того, является ли временной ряд стационарным, самым простым способом является графическое представление данных, что позволят визуально определить наличие тренда, сезонности или цикличности данных. Помимо графического представления можно применять различные статистические критерии, например, расширенный критерий Дики-Фуллера.

Автокорреляция – это мера корреляции между значениями ряда в различные периоды времени, что позволит выявить наличие всё тех же тренда, сезонности или цикличности. Автокорреляция высчитывается с помощью функции корреляции Пирсона, и для определения наличия неблагоприятных для прогнозирования свойств, можно также отобразить изменение автокорреляции с течением времени.

Спектральный анализ позволяет исследовать частотную составляющую ряда. Он используется для выявления скрытых цикличных паттернов во временных рядах. Для этого временной ряд разбивается на сигналы разных частот, а затем анализируется спектр этих частот. Спектральный анализ часто используется для анализа финансовых рынков, в которых цены могут изменяться в зависимости от времени и частоты.

Выбирать метод анализа необходимо с учётом решаемой задачи, после проведения анализа, можно переходить непосредственно к прогнозированию временного ряда.

В данной работе рассматривается байесовский подход к прогнозированию параметрических временных рядов и подстановочный принцип к прогнозированию временных рядов, проводится их сравнительный анализ. Исследована также векторная авторегрессионная модель временного ряда.

# 1. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

## **Задача прогнозирования временного ряда**

Прежде, чем задаваться вопросом выбора модели для прогнозирования, необходимо определить, что есть прогнозирование временного ряда.

Предположим, что у нас есть наблюдений за какой-то промежуток времени и для каждого наблюдения у нас есть признаков, задача прогнозирования состоит в том, чтобы построить функцию такую, что величина   наилучшим образом приближает значение где – количество шагов для построения прогноза, формат которого определяет тем, как нам заданы данные, то есть, если мы имеем данные о, например, ценах акций какой-либо компании на бирже за год на каждый месяц этого года, то значением мы выбираем то количество месяцев, прогноз на которое нас интересует. В общем, мы строим прогноз значения ряда в момент времени [6], используя данные, которые нам были известны до момента времени .

На практике практически невозможно точно спрогнозировать значение временного ряда, поскольку просто невозможно учесть все факторы, которые так или иначе влияют на итоговый прогноз. Поэтому чаще всего строится доверительный интервал, то есть такой интервал , что [6], то есть мы определяем интервал, в который с вероятность попадёт наш прогноз, хоть этот вариант и не позволит точно определить значение временного ряда в интересующий нас период времени, но это позволяет лучше решить поставленную задачу прогнозирования.

Как уже выяснилось, задача прогнозирования значения временного ряда есть ни что иное, как определение значения на основании того, что у нас уже есть, а это уже является задачей регрессии. При этом мы можем использовать не все данные, которые у нас есть, а только последних, то есть модель примет вид

где – произвольная функция. Таким образом, для построения данной функции можно использовать методы машинного обучения [6].

## **Методы машинного обучения**

Машинное обучение – это раздел искусственного интеллекта, который позволяет найти невидимые человеку закономерности в данных с помощью различных алгоритмов, и с помощью выявленных закономерностей строить прогнозы для новых данных, по структуре схожих с имеющимися.

Основным классом задач, разрешимых с помощью методом машинного обучения является задача регрессии, которую решить для прогноза временного ряда. По сути, задачу прогнозирования временного ряда можно считать задачей обучения с учителем, поскольку у нас есть для каждого набора параметров расставлены свои метки, то есть в случае временного ряда, значения этого временного ряда.

В качестве исходных данных используется множество параметров - признаков объекта *Х* [1]:

или ,

где – результат -го наблюдения по *j*-му параметру-признаку, – количество строк – число наблюдений, – количество столбцов, или число признаков, например, параметров функционирования объекта; – вектор значений -го параметра-признака (значения признаков могут быть как количественными, так и бинарными, номинальными, порядковыми), и вектор-столбец ответов [1].

Каждой строке множества соответствует определенное значение вектора . Совокупность пар () образует выборку исходных данных – прецедентов [1].

Задача состоит в построении функции , которая предскажет ответ для любого заданного . Таким образом, функция (ее называют еще и алгоритмом или моделью алгоритма) должна не только приближать искомый результат на выборке исходных данных, но и «работать» на всей генеральной совокупности, из которой получено множество [1].

Алгоритм, способный разрешить эту задачу, называют моделью, то есть параметрическое семейство функций где – фиксированная функция, – множество допустимых значений .При числовых значениях признаков часто используют линейные модели с вектором параметров , чаще эти значения называют весами, то есть эти параметры определяют значимость того или иного признака:

На данном этапе можно сказать, что задачей машинного обучения является как раз-таки подбор данных параметров . Такой процесс называется обучение модели, который происходит на основании некоторой функции качества или функции потерь, который выступает в качестве метрики оценки точности построенного моделью прогноза относительно выставленной метки. Прогноз считается точным, если значение является нулем, однако на практике для этого требуются очень качественные данные, много времени или идеальная модель, что в реальности практически невозможно, поэтому при обучении модели обычно задают какое-либо пороговое значение или просто количество итераций обучения.

Функционал качества алгоритма на выборке объема (среднее

значение потерь) называют эмпирическим риском [1]:

Обучение модели заключается в минимизации данного функционала [1]:

Определим способ минимизации рассматриваемого функционала. Чаще всего на практике используется метод градиентного спуска. Основной идеей метода является корректировка параметров модели в сторону минимума функции потерь. То есть на каждой итерации обучения мы считаем градиент функции по интересующему нас параметру и отнимаем полученное значение от корректируемого параметра, во время обучения это происходит сразу для всех параметров:

где – значение на предыдущей итерации обучения, а – некоторый гиперпараметр, называемый скоростью обучения, слишком низкие значения которого негативно могут повлиять на скорость обучения, а также может привести к тому, что модель слишком хорошо подстроится под обучающие данные, что не позволит ей делать корректные прогнозы для новых данных, а слишком большие значения приведут к тому, что модель в принципе не обучится из-за того, что в какой-то момент просто начнёт уходить от минимума.

Реальные данные бывают довольно объемными, что может затруднить процесс обучения модели, поскольку чем выше объем выборки, тем больше времени и вычислительной мощности нужно для обработки всех данных, это следует из формулы (1.2.4). Поэтому можно определить репрезентативную выборку, которая способна отразить качества всей генеральной совокупности, на которой можно обучить модель и получить точный прогноз.

При обучении модели принято разбивать исходную выборку на три непересекающиеся части: тренировочную, тестовую и контрольную. Большую часть от общей выборки чаще всего составляет тренировочная выборка, на которой осуществляется обучение модели и подбор ее параметров, на контрольной выборке производится оценка производительности модели и окончательная настройка гиперпараметров. На тестовой выборке, как следует из названия, происходит окончательная оценка производительности модели.

Очевидно, что лишь по одной небольшой части выборки, обычно это 20 процентов от общего объема, неправильно определять качество модели, поэтому применяют метод, называемый кросс-валидацией. При использовании кросс-валидации выборка разбивается на *N* частей (на практике обычно принимают *N = 5* или *N = 10*). (*N – 1*) часть используется для обучения, одна – для контроля. Последовательно перебираются все варианты, например, при разбиении на пять частей вначале в качестве обучающей выборки используются части 1 – 4, а часть 5 – тестирования. На следующем шаге в качестве обучающей используются части 2 – 5, а часть 1 – для тестирования, и т.д. Для каждого разбиения решается задача обучения по выборке и вычисляется функция ошибок на контрольной выборке . Среднее значение этой функции по всем вариантам разбиения и характеризует обобщающую способность алгоритма.

Для построения качественных моделей необходим предварительный анализ исходных данных: проверка значимости признаков, исключение выбросов, иногда необходима нормировка(стандартизация). При большом числе признаков целесообразно сократить размерность, используя, например, метод главных компонент.

Таким образом, основные этапы машинного обучения таковы:

1. Постановка задачи.

2. Выделение признаков (параметров функционирования), оказывающих влияние на состояние объекта.

3. Формирование выборки исходных данных и способа ее разбиения на обучающую и контрольную части.

4. Выбор функционала качества.

5. Предварительная обработка данных – отбор значимых признаков, обработка выбросов и пропусков, проведение стандартизации.

6. Построение модели по обучающей выборке.

7. Оценка качества модели по тестовой выборке.

8. Использование модели для прогнозирования состояния объекта по известным параметрам функционирования.

Методы машинного обучения не лишены недостатков, главным из которых является проблема переобучения модели, то есть модель находит закономерности в данных там, где их нет, иными словами, она хорошо делает прогнозы для тренировочных данных, а на новых серьезно ошибается. Данная проблема зачастую вызвана тем, что модель получилась слишком сложной относительно данных.

## **Методы решения переобучения**

Проблема переобучения является достаточно существенной и для ее разрешения есть некоторые методы разрешения, однако мы рассмотрим лишь некоторые из них: уменьшение сложности модели, регуляризация.

### **Уменьшение сложности модели**

Как было сказано выше, наиболее распространенной причиной переобучения является то, что модель является слишком сложной, потому самым простым способом будет банальное использование более простой модели, например, для нашей задачи прогнозирования временных рядов, можно использовать не нейронные сети, а линейную регрессию. Если есть строгая необходимость использовать выбранную модель, то упростить модель можно путём изменения гиперпараметров.

### **Регуляризация**

Регуляризация представляет собой наложение своего рода штрафов на модель за ее излишнюю сложность. Регуляризация делится на два вида: L1-регуляризация или L2-регуляризация. По своей сути, каждая из этих регуляризаций является своим способом изменения функции потерь.

L1-регуляризация, или Lasso регуляризация, основана на добавлении штрафа, равного абсолютному значению коэффициентов модели, она выражается следующей формулой [7]:

где – реальное значение,   – прогноз модели, – коэффициент регуляризации. За счет наложения штрафа именно таким образом L1-регуляризация способна уменьшить размерность данных, поскольку она может снизить значимость признака до нуля.

L2-регуляризация, или Ridge регуляризация, действует аналогично Lasso, но добавляемый штраф является не абсолютным значением, а квадратом [7]:

где все параметры аналогичны (1.3.2.1). В отличие, от L1-регуляризации, L2-регуляризация не склонна к уменьшению размерности, а она штрафует слишком высокие значения параметров, приближая их к нулю, но не приравнивая.

Третьим видом регуляризации, о котором я не упоминал выше, является ElasticNet, которая является обобщением описанных способов, то есть формула ее:

таким образом она способна объединить преимущества Lasso и Ridge регуляризаций.

# 2. ПРОГНОЗИРОВАНИЕ ВРЕМЕННОГО РЯДА НА ОСНОВЕ МОЕДЕЛЕЙ АВТОРЕГРЕСИИ

## **2.1. Общая информация об авторегрессионных моделях**

Модели авторегрессии являются классом моделей машинного обучения, которые, как следует из названия, решают задачу регрессии, однако от регрессионных моделей они отличаются тем, что используются они именно под задачи прогнозирования временных рядов путём статистических вычислений. Простыми словами, модели авторегрессии представляют каждое значение временного ряда в виде функции от предыдущих значений временного ряда. В авторегрессионных моделях такие математические методы применяются для вычисления вероятностной корреляции между элементами последовательности. На основе полученных результатов они пытаются угадать следующий элемент в неизвестной последовательности.

При прогнозировании временных рядов, используя авторегрессионные модели, важно, чтобы временной ряд обладал свойством стационарности. В широком смысле, временной ряд называется стационарным, если его математическое ожидание и дисперсия постоянны (значение определяет постоянную, вокруг которой лежат значения временного ряда, а характеризует ширину полосы, в пределах которой рассеяны эти значения), а автоковариационная функция зависит лишь от разности аргументов[1]:

Стационарность процесса может быть оценена визуально по графику временного ряда или с помощью специальных тестов.

Как уже было сказано, временные ряды играют очень важную роль во многих сферах жизни, особенно в финансах и экономике. Таким образом, не сложно сделать вывод о том, что авторегрессионные модели являются мощным инструментом для формирования полезных финансовых стратегий.

## **2.2. Прогнозирование по модели авторегресии**

В предположении о нормальности распределения можно разбить ряд на две части, проверить гипотезы о равенстве дисперсий в каждой части по критерию Фишера, затем – о равенстве средних по критерию Стьюдента.

Иногда целесообразно представить значение ряда в момент в

виде авторегрессионной зависимости: линейной функции от предыдущего наблюдения плюс случайный компонент.

*Модель авторегрессии* (AutoRegressive) *первого порядка*

имеет вид [1]:

где – числовые коэффициенты – последовательность

величин, образующих *белый шум* (так называется последовательность

некоррелированных случайных величин с нулевым математическим

ожиданием и постоянной дисперсией) [1]. Из вида этой модели можно сделать вывод о том, что графический смысл модели авторегрессии первого порядка состоит в том, чтобы построить линию, наилучшим образом соответствующую среднему значению точек данных, распределенных на двухмерном графике. Прогноз по новым данным будет сделан исходя из этой прямой.

Свободный член часто приравнивается нулю, то есть

рассматриваются центрированные процессы, средний уровень

которых равен нулю – [1].

Прогноз [1]

Модель авторегрессии порядка определяется выражением [1]:

Прогноз по модели авторегрессии порядка [1]:

и т. п.

Коэффициенты модели авторегрессии определяются методом наименьших квадратов [1]. После определения коэффициентов авторегрессии необходимо проверить, являются ли эти коэффициенты статистически значимыми, то есть оценить то, насколько каждое из значений временного ряда влияет на текущее значение. Для выполнения этой задачи применяется t-критерий Стьюдента.

Суть критерия Стьюдента заключается в том, что мы выдвигаем гипотезы:

коэффициент то есть предыдущее значение соответсвующего коэффициента не влияет на значение текущего;

коэффициент что указывает на влияние.

Для проверки выдвинутых гипотез необходимо рассчитать -статистику, которая вычисляется по формуле:

где – оценка коэффициента, – стандартная ошибка этой оценки коэффициента.

После рассчета -статистики необходимо установить уровень значимости , обычно этот параметр выбирают равным 0.05, после чего выбирается критическое значение из таблицы распределения Стьюдента. Данное значение зависит от непосредственно уровня значимости , а также количества степеней свободы:

где – количество элементов выборки, – порядок модели.

Финальным этапом является сравнение значения -статистики, вычисленной по формуле (3.1.5) с критическим значением.

* Если , то нулевая гипотеза отвергается, то есть коэффициент является статистически значимым, и его нельзя исключить из нашей модели;
* Если , то нулевая гипотеза не отвергается, что говорит о том, что коэффициент не является статистически значимым, и его влияние на прогноз является незначительным, а значит для упрощения модели его можно исключить из нее.

## **2.2. Прогнозирование по модели скользящего среднего**

Случайную составляющую временного ряда иногда целесообразно представить и в виде взвешенной суммы настоящего и прошлых значений белого шума [1]:

Такой процесс называется *моделью скользящего среднего* (Moving Average) порядка *q* и обозначается *MA(q)* [1].

Как и в случае авторегрессионных моделей, наибольшее распространение на практике имеют модели первого и второго порядка [1]:

Часто полезно включить в модель и слагаемые, описывающие авторегрессию, и члены, соответствующие скользящим средним. *Модель авторегрессии – скользящего среднего ARMA(p,q)* имеет вид [1]

В частности, *ARMA(1,1)* [1]:

Прогноз выполняется так [1]:

где

и т.д.

Модель *ARMA*(2,2) [1]:

Прогноз [1]

и т.д.

## **2.3. Модели авторегрессии – проинтегрированного скользящего среднего для нестационарных временных рядов**

Мы рассмотрели некоторые модели стационарного временного ряда. Реальные временные ряды обычно *нестационарны*. Однако на практике часто встречаются такие временные ряды, которые стационарны после вычитания из уровней ряда его неслучайной составляющей [1].

Назовем последовательной разностью первого порядка ряда новый ряд [1]:

последовательной разностью второго порядка – разность последовательных разностей первого порядка [1]:

*последовательной разностью d-го порядка* –

Предположим, что нестационарную составляющую можно представить в виде полинома степени *(d – 1).* Можно доказать, что переход к последовательным разностям *d*-го порядка исключает неслучайную составляющую. Будем предполагать, что оставшийся ряд, включающий только случайную составляющую, может быть представлен в виде модели *ARMA(p,q)* [1].

Тогда получим модель

называемую моделью *авторегрессии – проинтегрированного скользящего среднего* (Auto Regressive Integrated Moving Average), или сокращенно, *ARIMA(p,d,q)* [1].

Свободный член в модели (2.3.2) часто приравнивается нулю.

В общем случае сначала подбирают параметр *d* (обычно *d* ≤2), затем после перехода к стационарной модели проводится идентификация модели *ARMA(p,q)*, при этом, как правило, и *p* ≤ 2, и *q* ≤ 2 [1].

Прогноз, например, для модели *ARIMA*(1,1,1), можно найти так [1]:

тогда

и т.д.

Для модели *ARIMA*(2, 2, 1) по аналогии [1]:

тогда

и т.п.

## **2.4. Модели векторной авторегрессии**

Все описанные выше авторегрессионные модели применимы только для прогнозирования одномерных данных, то есть прогнозирование происходит только с учетом одного ряда признаков. На практике часто случается так, что помимо целевого значения в данных присутствуют и побочные параметры, которые будет полезно применить для построения более точного прогноза. Такие задачи носят название задач многомерной регрессии.

Для разрешения подобных задач был предложен новый вид моделей авторегрессии, а именно модели векторных авторегрессий или VAR-регрессии. Они позволяют строить прогнозы на основании взаимосвязи временных рядов между собой. Фактически, VAR является линейной комбинацией лагов прогнозируемого ряда и лагов соседних рядов. Предположим, что у нас есть временных рядов , и пусть является зависимым рядом, тогда получим следующую модель *VAR(p)* [8]:

Часто, для компактности, применяется матричная форма записи, для этого вводится вектор временных рядов [8]:

Построим прогноз, например, в случае , для модели *VAR(1).* Модель в этом случае, опираясь на (2.4.1), будет иметь следующий вид:

Сам прогноз строим следующим образом:

где

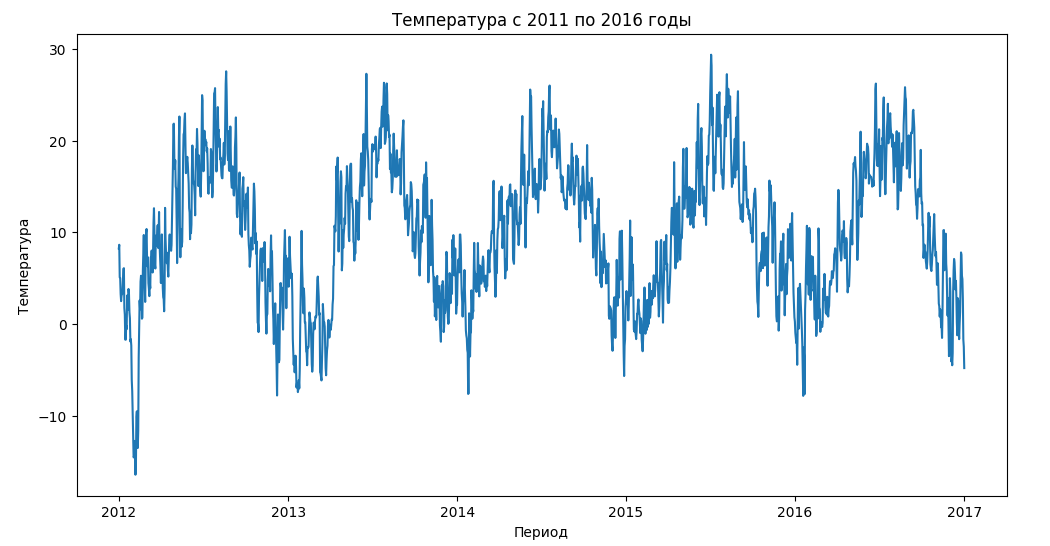
Важно отметить, что VAR-регрессии учитывают эндогенность переменных, что позволяет более точно анализировать взаимные влияния между ними. Это делает VAR более подходящим для изучения сложных систем в различных областях, включая экономику, финансы и социологию. Таким образом, VAR-регрессии представляют собой мощный инструмент для анализа и прогнозирования, что подчеркивает их значимость в современных исследованиях.

# 3. ПРОГНОЗИРОВАНИЕ ВРЕМЕНОГО РЯДА НА ОСНОВЕ ОДНОМЕРНОЙ АВТОРЕГРЕССИОННОЙ МОДЕЛИ

## **3.1. Подготовка данных**

Проведем анализ данных по с гидрометеоролической станции в Институте биогеохимии имени Макса Планка. Данный набор собирает 14 различных метеорологических показателей каждые 10 минут, начиная с 2003 года. Очевидно, что объем все выборки будет катастрофически огромным, поэтому будем работать с данными в промежутке между 2011 годом и 2016 [16]. Для начала, рассмотрим прогнозирование одномерных данных, то есть в модели будем учитывать только один показатель, целевой. Целевым выберем температуру.

Прежде, чем перейти к решению задачи прогнозирования, необходимо привести данные к более удобному виду, а именно сгруппируем данные по дням, данный прием позволит снизить объем выборки с, примерно, 260000 до 2000, что сделает вычисления значительно быстрее и проще. Первым шагом к изучению ряда будет его визуализация.



**Рисунок 3.1 –** Временной ряд

Рисунок 3.1 позволяет понять, что данные имеют форму синусоиды, из чего можно сделать вывод о том, что наши данные примерно повторяются каждый год, то есть присутствует ярко выраженная сезонность, что не позволяет нам назвать этот ряд стационарным. Для закрепления нашего предположения выполним расширенный тест Дики-Фуллера, который на основе регрессионной модели (3.1.1) выполняет проверку значимости коэффициентов этой модели

где первая разница временного ряда; постоянный член; коэффициент запаздывающего уровня временного ряда; коэффициент временного ряда; коэффициент запаздывающих первых разностей; остаточный член.

На основании следующей t-статистики

где – оценка параметра ; стандартная ошибка оценки.

Выполняется проверка гипотез

ряд не стационарен;

ряд стационарен.

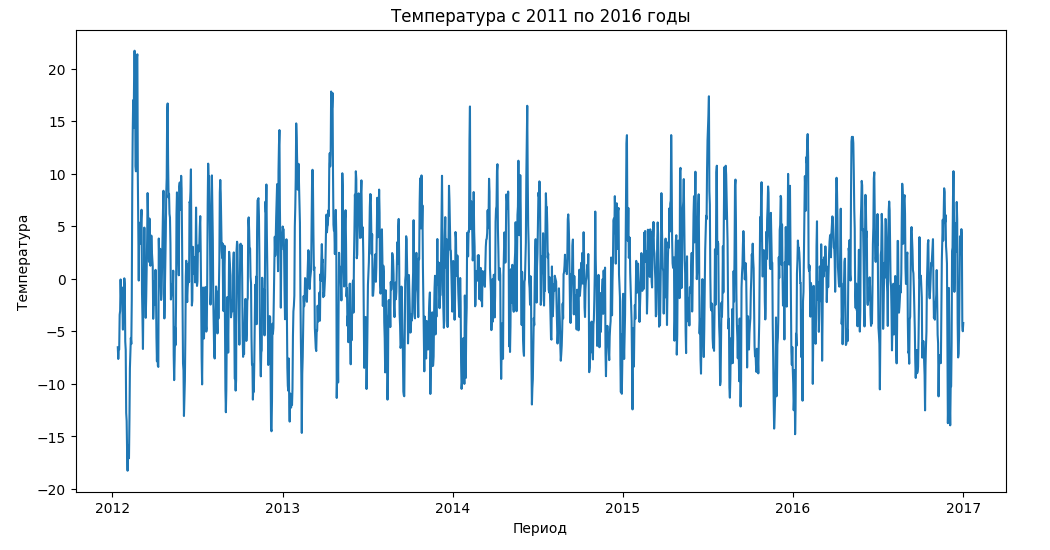
Статистика ADF рассчитывается на основе оценочного значения *b* и его стандартной ошибки. Его сравнивают с критическими значениями распределения Дики-Фуллера. Если статистика ADF более отрицательная, чем критическое значение на определенном уровне значимости, нулевая гипотеза единичного корня отклоняется. Это означает, что ряд стационарен. Рассмотрим ситуацию нашего ряда

**Таблица 3.1 –** Тест Дики-Фуллера

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADF-статистика | p-value | Критические значения | | |
| 1% | 5% | 10% |
| -3.08913 | 0.02736 | -3.43396 | -2.86313 | -2.56762 |

Как мы можем заметить в таблице 3.1, p-value у нас меньше 1, а следовательно, можно сделать вывод о стационарности ряда, хотя на рисунке 3.1 мы четко видим тренд. В этой ситуации стоит оговориться о том, что тест Дики-Фуллера в основном указывает на постоянность дисперсии или среднего, а это как раз на графике и отражено.

О сезонности избавимся, применив к нашему временному ряду операцию конечной разности, то есть просто вычтем из ряда этот же ряд, смещенный на длину периода, которая в нашем случае равна 365

**

**Рисунок 3.2 –**Продифференцированный временной ряд

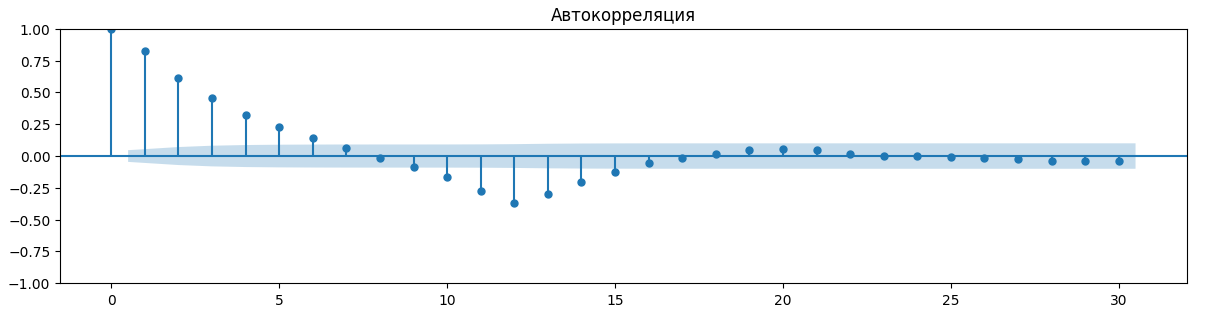
Дифференцирование приблизило нас к стационарному ряду, для подтверждения гипотезы о стационарности, выполним тест Дики-Фуллера.

**Таблица 3.2 –** Тест Дики-Фуллера

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADF-статистика | p-value | Критические значения | | |
| 1% | 5% | 10% |
| -9.26942 |  | -3.43401 | -2.86315 | -2.56763 |

Сейчас, как и в прошлый раз, тест Дики-Фуллера позволяет нам принять гипотезу о стационарности временной ряда. Это позволяет нам точно понять, что применение дифференцирования никак не испортило постоянство дисперсии.

Построим график автокорреляции нашего временного ряда

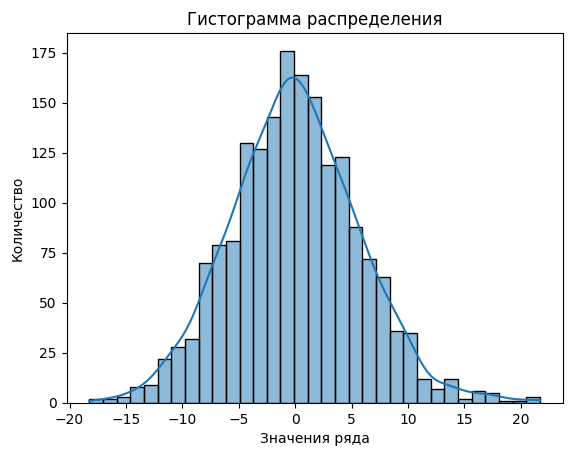


**Рисунок 3.3 –**График автокорреляции

Как можем заметить на рисунке 3.3, наш ряд с течением времени затухает, что указывает на отсутствие у него сезонности, то есть для автоковариационной функции выполняется условие (2.1.1).

Таким образом, мы смогли привести наш временной ряд к стационарному путем одной операции дифференцирования.

Определим, по какому закону распределены данные в нашем временно ряде, выполним это наиболее простым методом – построим гистограмму распределения



**Рисунок 3.4 –** Гистограмма распределения

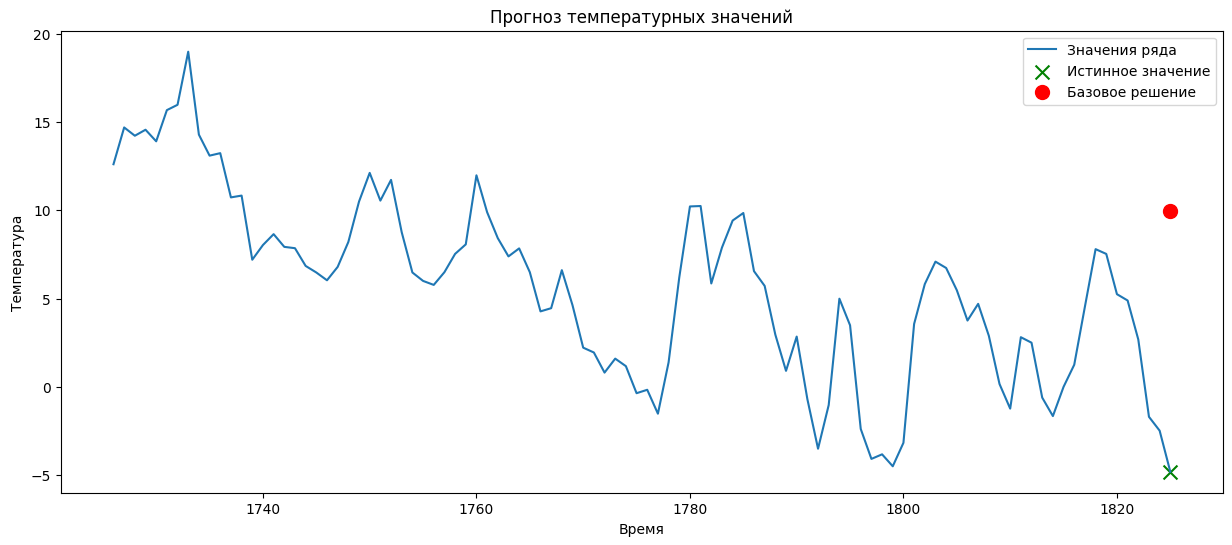
Как видно на рисунке 3.4 распределение нашего ряда напоминает нормальное распределение, соответственно, в дальнейшем и будем пользоваться этим распределением.

## **3.2. Базовое решение**

Прежде, чем использовать методы машинного обучение для прогнозирования временного ряда, зададим некую отправную точку, а именно построим базовое решение, то есть возьмем среднее значение температуры подвыборки данных за последний год, то есть прогноз будет строиться следующим образом:

где h – период прогнозирования, N – количество элементов.

Построим полученный результат на графике



**Рисунок 3.5 –**Базовое решение.

На рисунке 3.5 видно, то построенное таким образом никак не совпадает с истинным значением, поэтому перейдем к более серьезным подходам прогнозирования временных рядов.

## **3.3. Прогнозирование с помощью ARIMA**

Следующим этапом станет прогнозирование этого ряда, которое будем осуществлять с помощью модели (2.3.2) – модели авторегрессии проинтегрированного скользящего среднего.

Первой проблемой, с которой мы столкнемся при прогнозировании при использовании *ARIMA(p, d, q),* станет нахождение трех целочисленных параметров:

* p – порядок авторегрессии;
* d – порядок дифференцирования исходного временного ряда;
* q – порядок скользящего среднего.

Подбор этих параметров не является тривиальной задачей. Самым

простым методом будет банальный перебор различных наборов параметров, так называемый поиск по решетке. Такой перебор должен сопровождаться некоторой оценкой производительности, в нашем случае будем обращаться к информационным критериям, которые оценивают то, на сколько хорошо модель подходит под данные. Будем использовать два критерия [10]:

Формула (3.3.1) называется информационным критерием Акаика (AIC), в нем – значение логарифмической функции правдоподобия построенной модели, – количество использованных параметров, – объем рассматриваемой выборки. Формула (3.3.2) является Байесовским информационным критерием (BIC), который является модификацией AIC, разработанный исходя из байесовского подхода, в нем – значение логарифмической функции правдоподобия построенной модели, – количество использованных параметров, – объем выборки.

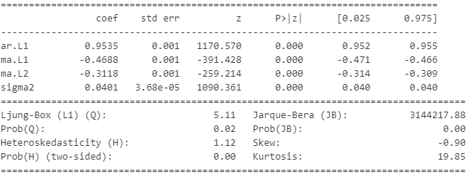
Рассмотрим критерий BIC и сделаем перебор для следующих диапазонов параметров

Выполним это программно и получим следующие показатели

Значение нашего байесовского информационного критерия следующее:

Для проверки оптимальности подобранных параметров выведем таблицу с результатами построенной модели.

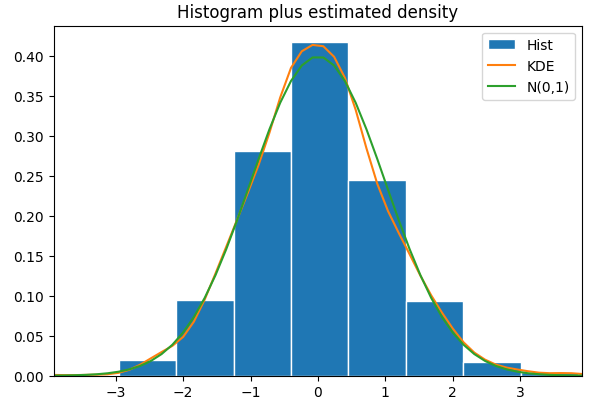
**Таблица 3.3 –** Результаты *ARIMA(1,1,2)*

**

Больше всего нас интересует столбец коэффициентов. Столбец *coef* показывает влияние каждого параметра на временной ряд, а – значимость. Чем ближе значение к нулю, тем выше значимость.

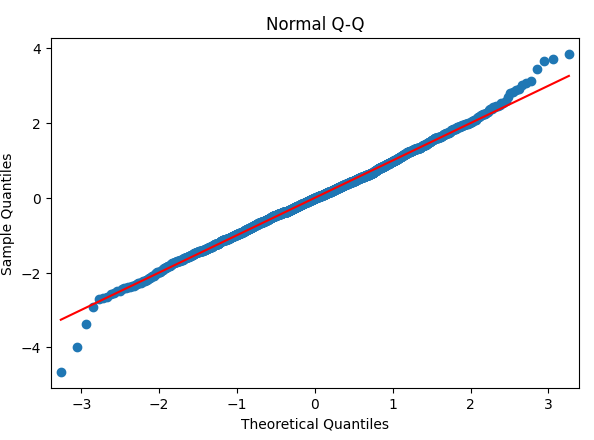
Для достоверности проверим гипотезу о нормальности распределения остатков для построенной модели, для этого построим несколько графиков.

Начнем с построения гистограммы распределения остатков.



**Рисунок 3.6 –** Гистограмма распределения остатков

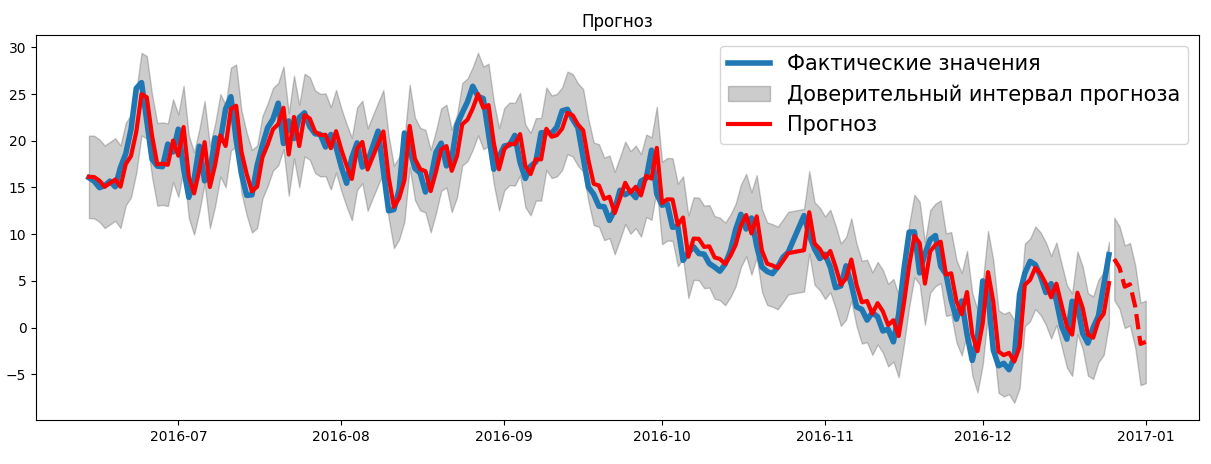
На рисунке 3.6 не наблюдается никаких аномалий, сама гистограмма практически идеально повторяет нормальное распределение. Для того, чтобы подтвердить этот факт, построим график квантиль-квантиль.



**Рисунок 3.7 –**График квантиль-квантиль для остатков.

Аналогично гистограмме, график квантиль-квантиль практически идеально повторяет график нормального распределения. Спорными являются лишь элементы, отклоняющиеся от нормального элемента на концах отрезка. Это не является критическим фактором для отклонения гипотезы о нормальности распределения остатков, что в совокупности с гистограммой позволяет нам сделать вывод о нормальности распределения остатков и сказать, что построенная модель является качественной.

Перейдем к построению прогноза нашего временного ряда. Поскольку мы занимаемся прогнозом погоды, то самым оптимальным периодом для прогноза будет 7 дней.



**Рисунок 3.8** *–* Прогноз с помощью *ARIMA(1, 1, 2)*

На рисунке 3.8 мы видим, что реальные значения ряда в большинстве случаев попадают в пределы доверительного интервала прогноза ARIMA. У дальнейшего прогноза точность будет уменьшаться с увеличением дальности. Согласно нашему прогнозу, на следующей неделе ожидается похолодание.

Оценим построенную модель с помощью такой метрики, как средний квадрат ошибки *(MSE)*

где – количество наблюдений, – фактическое значение временного ряда, – предсказанные значения моделью ARIMA(1, 1, 2). Рассматривать будем саму изначальную выборку, для которой данная метрика будет следующей

В целом, можно сказать, что данный показатель является очень хорошим для задачи прогнозирования временного ряда. Рассмотрим, можно ли его улучшить.

## **3.4. Байесовский подход**

Рассмотрим оценивание параметров нашей модели *ARIMA* с точки зрения байесовского подхода. В отличие от рассмотренного в предыдущем пункте поиска по сетке, данный метод для подбора оптимальных параметров отслеживает результаты прошлых оценок, которые используются для подбора гиперпараметров модели.

Байесовская оптимизация может использоваться для оптимизации каждого отдельного параметра модели *ARIMA*. Для каждого параметра, процесс байесовской оптимизации может быть следующим:

1. *Определение априорной модели.* Вначале необходимо определить априорную вероятностную модель для каждого параметра. Априорная модель может быть задана в виде распределения параметра. Например, можно предположить, что параметр следует равномерному распределению на заданном диапазоне значений, то есть его априорное распределение

где – диапазон параметра, – случайная величина, представляющая параметр .

1. *Определение целевой функции*. Далее, необходимо определить целевую функцию, которую мы хотим оптимизировать. В случае оптимизации параметра , целевая функция может основываться на точности прогнозирования модели ARIMA с заданным значением . Будем рассматривать средний квадрат ошибки

где – количество наблюдений, – фактическое значение временного ряда, – предсказанные значения моделью ARIMA с параметром равным .

1. *Апостериорная модель*. Апостериорная модель вычисляется с использованием байесовского правила и обновляет вероятностные представления параметра на основе имеющихся данных. По формуле Байеса [11]:

где – апостериорное распределение параметра после учета данных , – функция правдоподобия, – нормализующий множитель (вероятность данных ).

Функция правдоподобия вычисляется следующим образом

где – *i-*й элемент временного ряда, – дисперсия временного ряда, который в случае стационарности имеет нормальное распределение.

Подставим в формулу (4.3.3)

Математическое ожидание и дисперсию посчитаем программно

Подставим эти значения в (4.3.4)

Дальнейшую подстановку выполним после определения промежутка для оценки параметров.

1. *Выбор следующего значения параметра.* Выбор следующего значения параметра p может быть выполнен с использованием различных методов выбора, таких как выбор значения с наибольшей апостериорной вероятностью или методы, основанные на оценке риска и исследовании пространства параметров. Например, можно использовать метод максимальной вероятности (*MAP*), который выбирает значение параметра , соответствующее наибольшей апостериорной вероятности:

где обозначает оптимальное значение параметра .

1. *Обновление и повторение.* Выбранное значение параметра используется для обновления модели *ARIMA* и вычисления новой апостериорной модели. Затем процесс повторяется для следующего параметра (например, *d* и *q*), чтобы оптимизировать их значения.

Весь процесс байесовской оптимизации выполняется итеративно, где

на каждой итерации обновляются апостериорные модели и выбираются новые значения параметров, пока не будет достигнуто условие остановки или найдено оптимальное значение параметров *p, d, q,* соответствующее оптимальной модели *ARIMA.*

Как уже было сказано, главным преимуществом байесовской оптимизации над условным поиском по сетке является скорость работы. Однако это действительно ощущается случаев, когда мы работаем с большим объемом количества параметров, так, если, например, предположить, что параметры нашей модели находятся в следующих промежутках

то алгоритм поиска по сетке должен перебрать комбинаций, что, если считать, что на одну комбинацию программа потратит 1 секунду, займет больше 2-ух часов.

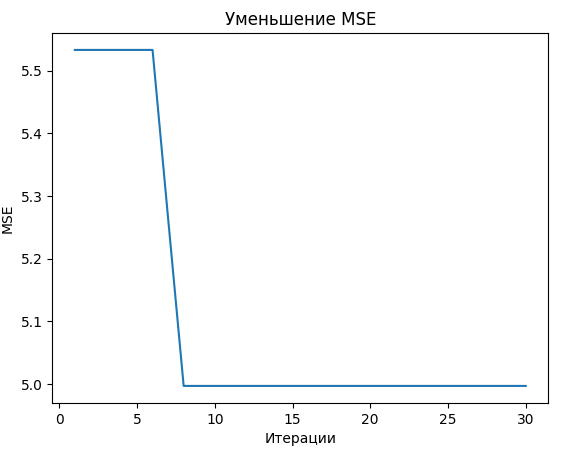
В случае Байесовской оптимизации вернемся к формуле (3.4.5) и выполним подстановку значений

Дальнейший этап по оценке параметров методом MAP выполним программно и получим

При этом лучшее значение целевой функции было

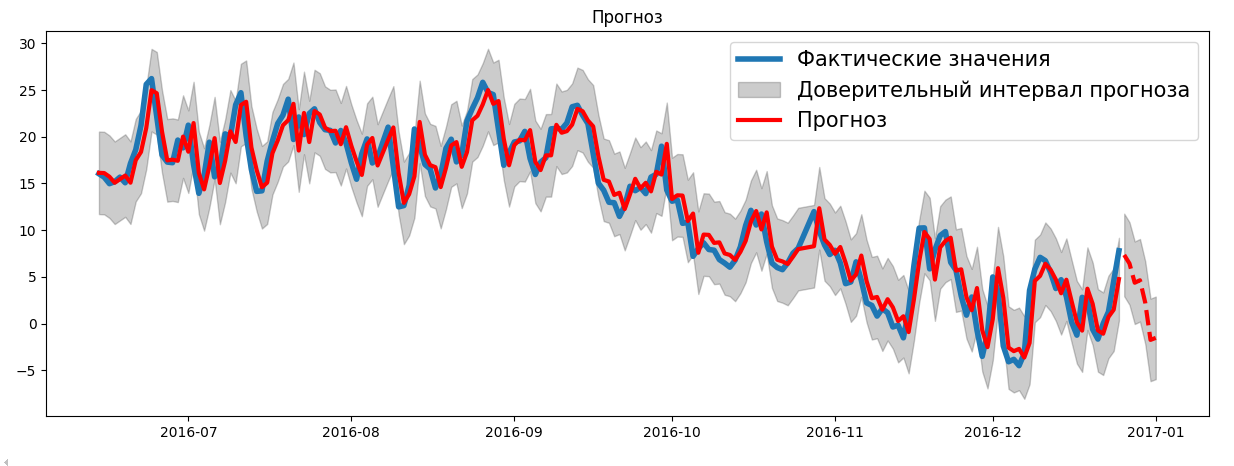
Как мы видим, результат в сравнении с моделью ARIMA(12, 0, 19) уменьшился не значительно.

При этом график уменьшения целевой функции имеет следующий вид



**Рисунок 3.9 –** График уменьшения MSE

Попробуем построить прогноз с помощью модели *ARIMA(12, 0, 19).*



**Рисунок 3.10 –** Прогноз с помощью *ARIMA(12, 0, 19)*

Если мы вспомни рисунок 3.8 и сравним полученные результаты, то можно сказать, что на конкретных данных Байесовский подход не сильно улучшил прогноз, поскольку даже до применения этого подхода мы смогли получить достаточно хорошие результаты.

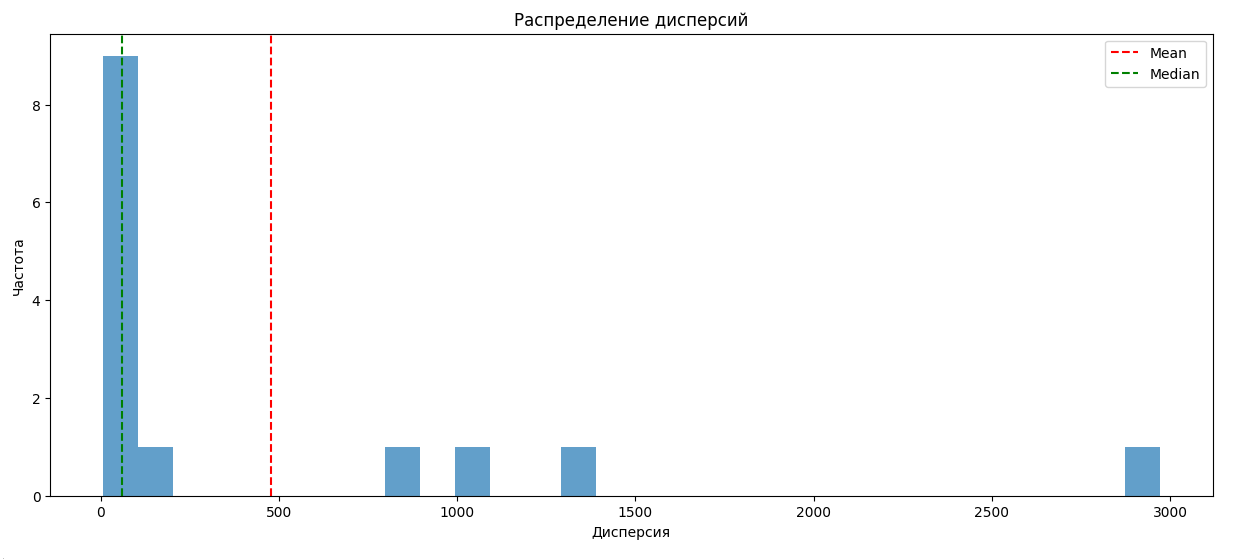
## **3.4. Вывод**

По итогам проведенной работы было выяснено, что модель одномерной авторегресии неплохо справилась с прогнозирование выбранного временного ряда. Однако ключевой проблемой при построении модели авторегресии стала задача подбора параметров, ведь они требуют больших вычислительных мощностей и времени. Нам удалось облегчить эту задачу путём использования байесовского подхода.

# 4. ПРОГНОЗИРОВАНИЕ ВРЕМЕНОГО РЯДА НА ОСНОВЕ ВЕКТОРНОЙ АВТОРЕГРЕССИОННОЙ МОДЕЛИ

## **4.1. Отбор признаков**

Продолжим идею главы 3, а именно спрогнозируем значения температуры, учитывая в прогнозе больше факторов, в нашем датасете изначально есть 14 признаков, что являются слишком высоким показателем для построения модели, поэтому нам необходимо снизить размерность данных. Поэтому при прогнозировании будем использовать только параметры с высокой дисперсией, для определения порогового значения, по которому будем отсеивать параметры, построим гистограмму распределения дисперсий



**Рисунок 4.1 –** Распределение дисперсий параметров

На рисунке 4.1 видно, что медиана находится примерно на значении 50, это и выберем в качестве порогового значения. Выведем дисперсии каждого параметра

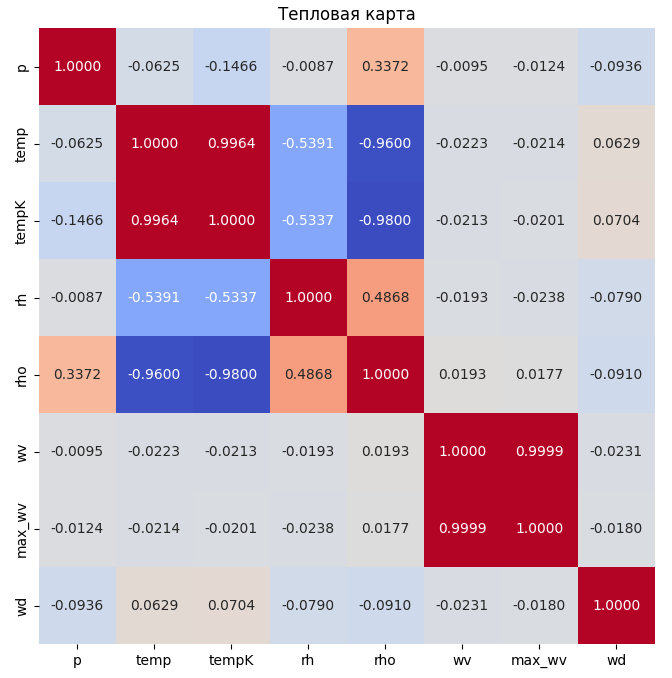
**Таблица 4.1 –** Дисперсии параметров

|  |  |  |  |
| --- | --- | --- | --- |
| *Параметр* | *Дисперсия* | *Параметр* | *Дисперсия* |
| p | 64.7 | tempK | 59.37 |
| tdew | 40.1 | rh | 129.2 |
| vpmax | 49.2 | vpact | 16.7 |
| vpdef | 13.6 | sh | 6.7 |
| h2oc | 17.1 | rho | 1342.1 |
| wv | 857.11 | max\_wv | 1060.0 |
| wd | 2971.3 |

Отсюда следует, что под наши критерии не подходят переменные tdew, vpmax, vpact, vpdef, sh, h2oc, соответственно, при дальнейшем исследовании их учитывать не будем.

Несмотря на то, что мы исключили 6 признаков, у нас остается есть 7 признаков, которые могут не нести никакой информации о целевой переменной, то есть выберем те признаки, коэффициент корреляции Пирсона с целевой переменной будет больше 0.5. Для этого необходимо построить матрицу корреляции, элементами которой являются коэффициенты корреляции Пирсона, которые вычисляются по следующей формуле:

Для наглядности построим тепловую карту.



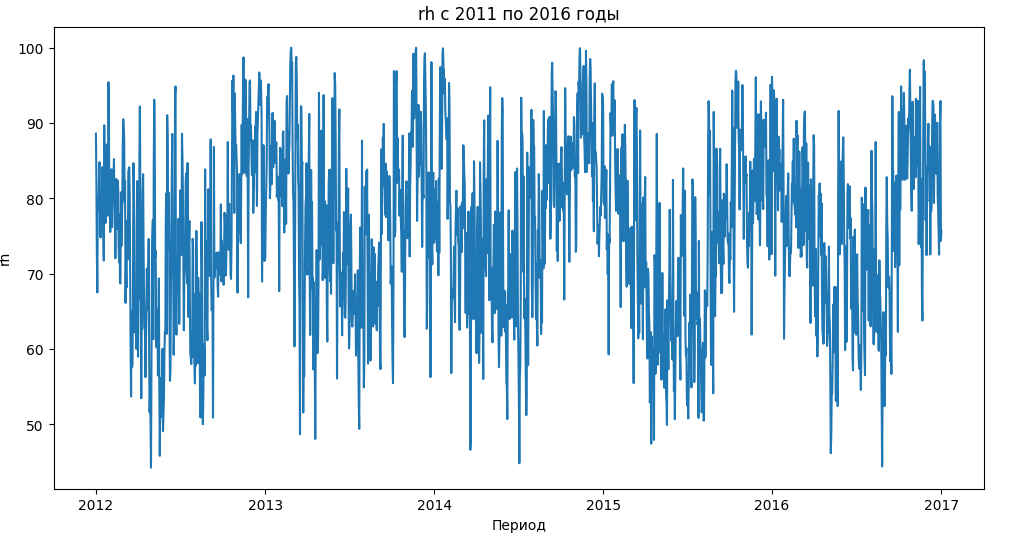
**Рисунок 4.2 –** Тепловая карта

Проанализировав рисунок 4.2, видно, что сильно коррелируют с целевой переменной, temp, переменные tempK, rh, rho. Помимо этого, на тепловой карте мы можем увидеть, что сильной корреляцией обладают переменные tempK и rho, что может сильно повлиять на результаты прогноза, поэтому исключим одну из них, tempK, поскольку она отражает температуру в Кельвинах.

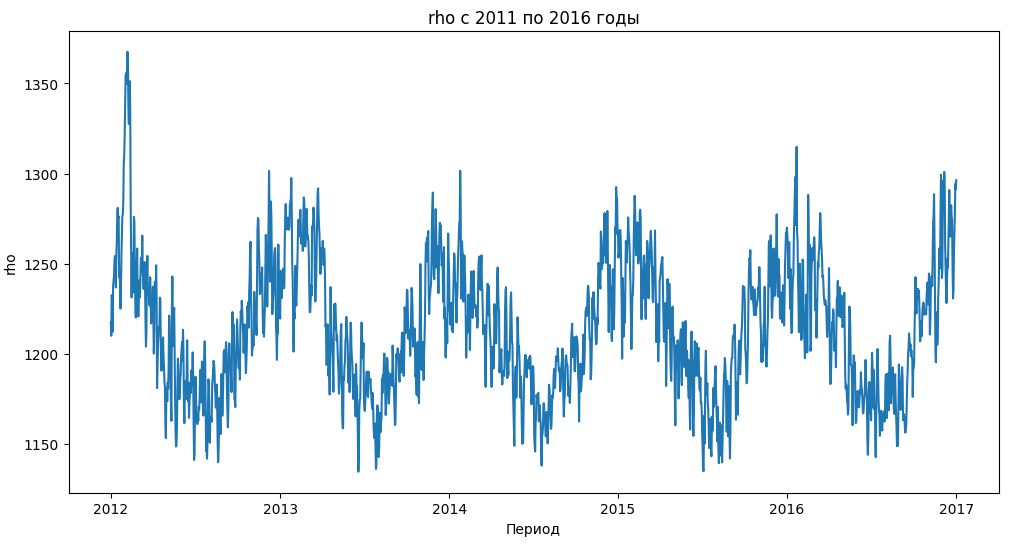
Таким образом, в прогнозе будут участвовать переменные rh, rho.

## **4.2. Подготовка данных**

Возвращаясь к пункту 3.1, для прогнозирования временного ряда с помощью одномерной авторегрессионной модели нам необходимо привести ряд к стационарному, с целевым рядом мы это уже проделывали. Теперь же в прогнозе участвуют еще 2 ряда, для которых необходимо добиться условия стационарности. Проделаем аналогичные действия для rh и rho: построим графики каждого из них:



**Рисунок 4.3 –** График временного ряда rh

**

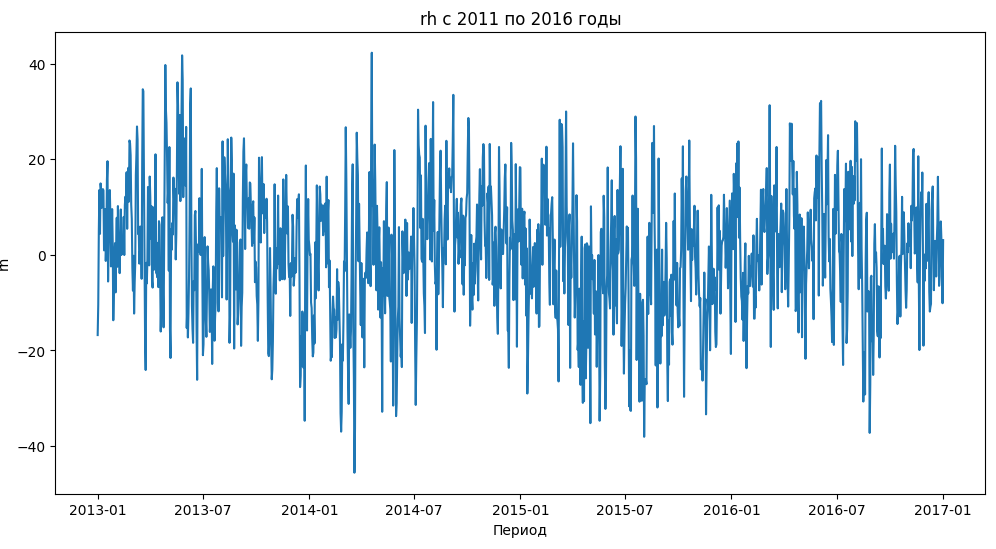
**Рисунок 4.4 –** График временного ряда rho

На обоих рисунках видно, что временные ряды обладают постоянной дисперсией, но при этом наблюдается и наличие сезонности. Выполним тесты Дики-Фуллера:

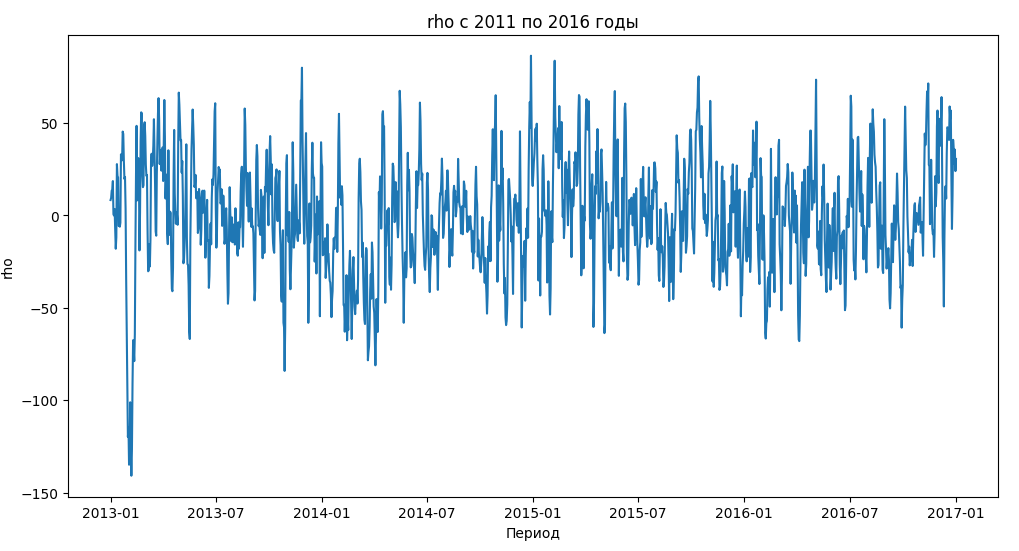
**Таблица 4.2 –** Тесты Дики-Фуллера

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Временной ряд | ADF-статистика | p-value | Критические значения | | |
| 1% | 5% | 10% |
| rh | -4.55573 | 0.00015 | -3.43397 | -2.86314 | -2.56762 |
| rho | -3.00072 | 0.03482 | -3.43397 | -2.86314 | -2.56762 |

Из теста Дики-Фуллера можно сделать вывод о том, что наши временные ряды стационарны по дисперсии, то есть дисперсии постоянна. Избавимся от тренда, применив преобразование (3.1.3), поскольку данные повторяют свою форму с периодичностью в 1 год или 365 дней, тогда получим:



**Рисунок 4.5 –** График продифференцированного временного ряда rh

**

**Рисунок 4.6 – График продифференцированного временного ряда rho**

Таким образом, для стационарности обоих рядов необходимо провести по одной операции дифференцирования.

## **4.3. Построение прогноза**

Для прогнозирования температуры с использование дополнительных параметров rh, rho будем пользоваться моделью VAR-регресии (2.4.1). Аналогично ARIMA-модели, важным для качества прогноза шагом является подбор гиперпараметров, а именно количество лагов или временных отрезков, используемых при прогнозировании. Подбор данного значения будет осуществляться с помощью описанных ранее критериев AIC (3.3.1) и BIC (3.3.2) из диапазона Подбор осуществим программно и получим следующие показатели:

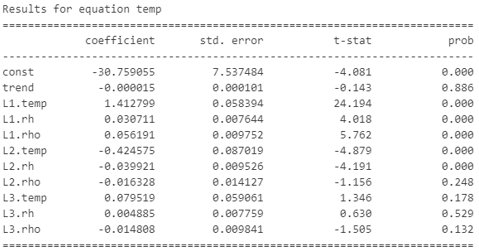
Мы получили расхождение результатов, потому прибегнем к третьему критерию HQIC или критерию Ханнана-Куинна, который работает по следующей формуле:

где максимальное значение логарифмической функции правдоподобия, число параметров модели, объем выборки. Наилучшим значением данного критерия является

*.*

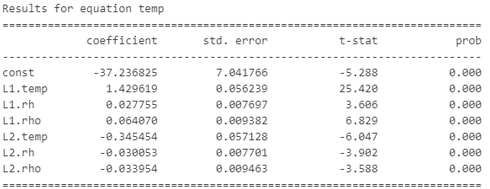
Таким образом, мы определили оптимальное число лагов для построения модели. Построим модель и выведем результаты полученной модели

**Таблица 4.3** – Коэффициенты модели VAR(3)



Заметим, что как минимум 4 элемента построенной модели не являются статистически значимыми, что говорит о том, что их можно не учитывать. Попробуем построить модель

**Таблица 4.4** – Коэффициенты модели VAR(2)



В полученной модели мы можем видеть, что все элементы модели является значимыми. Проверим качество модели, используя критерий Дарбина-Уотсона, который применяется для проверки остатков на автокорреляцию. Статистика Дарвина-Уотсона вычисляется следующим образом:

где остатки или невязки регрессионной модели, объем выборки.

На основании значения статистики Дарвина-Уотсона (4.3.2) делаются следующие выводы:

* Значения около 2 означают отсутствие автокорреляции;
* Значения значительно меньше 2 означают наличие положительной автокорреляции;
* Значения значительно больше 2 означают наличие негативной автокорреляции.

Вычислим значения статистики

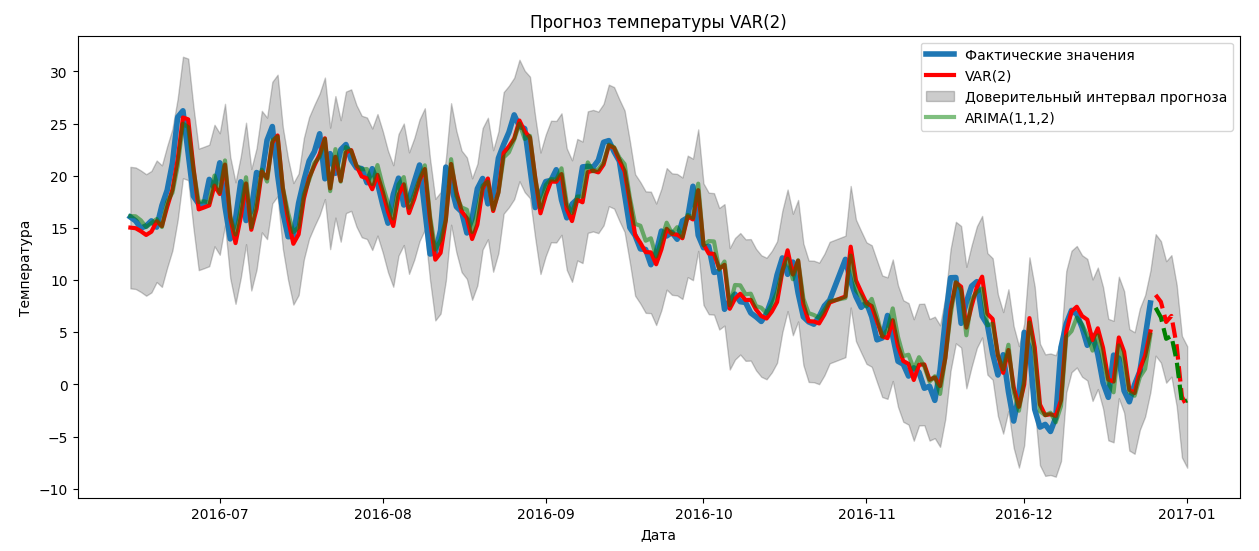
**Таблица 4.4 –** Статистика Дарвина-Уотсона

|  |  |  |  |
| --- | --- | --- | --- |
|  | temp | rh | rho |
| Значение | 1.95934 | 2.01195 | 1.94696 |

Все значения находятся примерно на уровне 2, следовательно, можно

сделать вывод об отсутствии автокорреляции остатков, что говорит о качестве построенной модели.

Всё, что остается сделать на данном этапе – построить прогноз модели. Так же, как и в главе 3, прогноз будем строить на последнюю неделю 2017 года. Для наглядности, добавим на график прогноз, полученный с помощью модели ARIMA(1,1,2).



**Рисунок 4.7 –** Результаты VAR(2) для temp

На рисунке 4.7 видно, что если различия между двумя моделями и есть, то они незначительные. Для того, чтобы определить, какая из моделей справилась лучше, необходимо прибегнуть к метрикам. Как и в прошлый раз, возьмем MSE:

Возвращаясь к результатам прогноза ARIMA(1,1,2), тогда значение MSE было приблизительно равно 5.11, что отличается от полученного значения всего на 0.04, учитывая объем выборки, это можно посчитать как погрешность, то есть обе модели справились одинаково хорошо.

## **4.4. Вывод**

В ходе проведенных исследований выяснилось, что результаты, полученный с помощью модели векторной авторегресии не стали лучше, чем результаты, полученные при работе с моделью ARIMA, что означает то, что не всегда использование более сложных моделей является необходимостью.

При использовании более сложных данных модели векторной авторегресии должны составить более точный прогноз.

# ЗАКЛЮЧЕНИЕ

В ходе работы были получены следующие результаты

1. Построены прогнозирующие статистики для параметрических ARIMA- временных рядов на основе байесовского подхода
2. Построен алгоритм прогнозирования векторного AR-временного ряда
3. Проведен сравнительный анализ байесовского и классического подхода к прогнозированию параметрических временных рядов
4. Разработано программное приложение на языке Python, реализующее анализ временного ряда, прогнозирование его с помощью модели ARIMA и байесовский подход к модели ARIMA. А также проведено сравнение прогнозирования с помощью модели одномерной авторегресии и векторной авторегресии.

# СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ

1. Кувайскова, Ю. Е Статистические методы прогнозирования / Ю. Е. Кувайскова, В. Н. Клячкин/ – Минск: УГТУ, 2019.
2. Ивченко, Г. И. Математическая статистика: Учеб. Пособие для втузов / Ивченко Г. И., Медведев Ю. И./ – Москва: МВТУ им. Н. Э. Баумана, 1984.
3. Грэйнджер, Клайв У. Дж. Эконометрический анализ временных рядов / Клайв У. Дж. Грэйнджер.
4. Абрамова, А. Е. Метод Монте Карло по схеме марковской цепи для оценки вероятности редких событий в задачах биоинформатики / Абрамова А. Н./ – Санкт-Петербург: СПБГУ, 2017.
5. Временной ряд – Википедия: [Электронный ресурс]. URL: https://ru.wikipedia.org/ (Дата обращения: 26.11.2023).
6. Временной ряд: [Электронный ресурс]. URL: https://education.yandex.ru/handbook/ml/article/vremennye-ryady (Дата обращения: 16.10.2024).
7. Регуляризация (математика) – Википедия: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki (Дата обращения: 16.10.2024).
8. Векторная авторегрессия – Википедия: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/ (Дата обращения: 18.11.2024)
9. Градиентный спуск: всё, что нужно знать: [Электронный ресурс]. URL: https://neurohive.io/ (Дата обращения: 10.12.2023).
10. Comparing Clustering Methods: Using AIC and BIC for Model Selection | by Kevin Menear | Medium: [Электронный ресурс]. URL: https://medium.com/@kevin.menear/comparing-clustering-methods-using-aic-and-bic-for-model-selection-bf80d0d37ec2 (Дата обращения 13.12.2023).
11. Теорема Байеса [3Blue1Brown] – YouTube: [Электронный ресурс]. URL: https://www.youtube.com/ (Дата обращения: 02.12.2023)
12. Hamilton, J. D. Time Series Analysis / James D. Hamilton/ – Princeton, New Jersey: Princeton University, 1994.
13. A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning: [Электронный ресурс]. URL: https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f (Дата обращения: 08.05.2024).
14. Tuning ARIMA for Forecasting: An Easy Approach in Python: [Электронный ресурс]. URL: https://medium.com/@sandha.iitr/tuning-arima-for-forecasting-an-easy-approach-in-python-5f40d55184c4 (Дата обращения: 08.05.2024).
15. Оптимизация гиперпараметров — Википедия: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki (Дата обращения: 08.05.2024).
16. Подбор гиперпараметров: [Электронный ресурс]. URL: https://education.yandex.ru/handbook/ml/article/podbor-giperparametrov (Дата обращения: 09.05.2024).
17. Прогнозирование временных рядов с помощью рекуррентных нейронных сетей / Хабр: [Электронный ресурс]. URL: https://habr.com/ru/articles/495884/ (Дата обращения: 10.12.2024).
18. Информационный критерий Ханнана-Куина (Hannan-Quinn criterion): [Электронный ресурс]. URL: https://wiki.loginom.ru/articles/hq.html (Дата обращения: 11.12.2024).

# Приложение

Листинг кода приложения на языке Python:

import warnings

warnings.filterwarnings("ignore")

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from statsmodels.graphics.tsaplots import plot\_acf

from statsmodels.tsa.stattools import adfuller

from sklearn.metrics import mean\_squared\_error

from statsmodels.tsa.arima.model import ARIMA

from statsmodels.tsa.api import VAR

import tensorflow as tf

save\_dir = 'Datasets'

csv\_filename = 'jena\_climate\_2009\_2016.csv.zip'

zip\_path = tf.keras.utils.get\_file(

fname=csv\_filename,

origin='https://storage.googleapis.com/tensorflow/tf-keras-datasets/jena\_climate\_2009\_2016.csv.zip',

cache\_dir=save\_dir,

extract=True

)

df = pd.read\_csv(r'Datasets\datasets\jena\_climate\_2009\_2016.csv.zip')

df = df.rename(columns={'Date Time': 'date', 'p (mbar)': 'p', 'T (degC)': 'temp'})

df.head(7)

df['date'] = pd.to\_datetime(df['date'], format='%d.%m.%Y %H:%M:%S')

max\_date = df['date'].max()

two\_years = max\_date.replace(year=max\_date.year - 5)

df = df[df['date'] >= two\_years]

df.head(7)

df['day'] = df['date'].dt.date

daily\_means = df.groupby('day').mean().reset\_index()

daily\_means['day'] = pd.to\_datetime(daily\_means['day'], format='%d.%m.%Y %H:%M:%S')

daily\_means.set\_index('day', inplace=True)

daily\_means = daily\_means.drop(columns='date')

daily\_means.head(7)

df\_temp = daily\_means['temp']

df\_temp.head(7)

def plot(data, name):

plt.figure(figsize=(12, 6))

plt.plot(data.index, data)

plt.title(f'{name} с 2011 по 2016 годы')

plt.xlabel('Период')

plt.ylabel(f'{name}')

plt.show()

plot(df\_temp, 'Температура')

def fuller(data):

result = adfuller(data)

print('ADF статистика:', result[0])

print('p-значение:', result[1])

print('Критические значения:')

for key, value in result[4].items():

print(f' {key}: {value}')

fuller(df\_temp)

diff = df\_temp - df\_temp.shift(365)

diff.dropna(inplace=True)

plot(diff, 'Температура')

fuller(diff)

plt.figure(figsize=(12, 6))

plt.subplot(211)

plot\_acf(diff, lags=30, ax=plt.gca())

plt.title('Автокорреляция')

plt.tight\_layout()

plt.show()

sns.histplot(diff, kde='norm')

plt.xlabel('Значения ряда')

plt.ylabel('Количество')

plt.title('Гистограмма распределения')

plt.show()

one\_year = max\_date.replace(year=max\_date.year - 1)

mean\_value = df\_temp[-365:].mean()

plt.figure(figsize=(15, 6))

last\_100\_values = df\_temp[-365:]

plt.plot(last\_100\_values.index, last\_100\_values, label='Значения ряда')

plt.scatter(df\_temp.index[-1], df\_temp[-1:], marker='x', label='Истинное значение', color='g', s=100)

plt.scatter(last\_100\_values.index[-1], mean\_value, color='r', label='Базовое решение', s=100)

plt.legend()

plt.title('Прогноз температурных значений')

plt.xlabel('Индекс')

plt.ylabel('Температура')

plt.show()

p = range(0,10)

d = range(0,3)

q = range(0,3)

pdq = list(itertools.product(p, d, q))

best\_pdq = (0,0,0)

best\_bic = np.inf

for params in pdq:

model\_test = ARIMA(df\_temp, order = params)

result\_test = model\_test.fit()

if result\_test.bic < best\_bic:

best\_pdq = params

best\_bic = result\_test.bic

p, d, q = best\_pdq

print(f'Порядок авторегресии: {p}\nПорядок дифференцирования: {d}\nПорядок скользящего среднего: {q}')

print(f'Наилучшее значение BIC: {best\_bic}')

model = ARIMA(df\_temp, order=(p, d, q))

model\_fit = model.fit()

print(model\_fit.summary())

model\_fit.plot\_diagnostics(figsize=(15, 10))

plt.show()

def plot\_forecast(data, model, order, threshold):

model = model(data, order=order)

model\_fit = model.fit()

forecast = model\_fit.get\_prediction(start=0, end=len(df\_temp)-1)

forecast\_ci = forecast.conf\_int()

threshold = threshold

mean\_value = forecast\_ci[['lower temp', 'upper temp']].mean().mean()

def replace\_anomalies\_with\_mean(row):

if abs(row['lower temp']) > threshold:

row['lower temp'] = mean\_value

if abs(row['upper temp']) > threshold:

row['upper temp'] = mean\_value

return row

forecast\_ci = forecast\_ci.apply(replace\_anomalies\_with\_mean, axis=1)

plt.figure(figsize=(15,5))

plt.plot(data[-200:-7], label="Фактические значения", linewidth=4)

plt.fill\_between(forecast\_ci.index[-200:-7],

forecast\_ci.iloc[:, 0][-200:-7],

forecast\_ci.iloc[:, 1][-200:-7], color='k', alpha=.2, label='Доверительный интервал прогноза')

plt.plot(forecast.predicted\_mean[-200:-7], color='red', label="Прогноз", linewidth=3)

plt.plot(forecast.predicted\_mean[-7:], color='red', linestyle='dashed', linewidth=3)

plt.fill\_between(forecast\_ci.index[-7:],

forecast\_ci.iloc[:, 0][-7:],

forecast\_ci.iloc[:, 1][-7:], color='k', alpha=.2)

plt.legend(fontsize=15)

plt.title('Прогноз')

plt.show()

plot\_forecast(df\_temp, ARIMA, (p, d, q), 100)

forecast = model\_fit.predict()

mse = mean\_squared\_error(df\_temp, forecast)

print(f'MSE: {mse}')

def arima\_objective\_function(params):

global data\_values

p, d, q = params

try:

model = ARIMA(data\_values, order=(p, d, q))

predictions = model.fit()

y\_pred = predictions.predict()

mse = mean\_squared\_error(data\_values, y\_pred)

except:

mse = float('inf')

return mse

param\_space = [hp.choice('p', range(0, 20)), hp.choice('d', range(0, 20)), hp.choice('q', range(0, 20))]

data\_values = df\_temp

trials = Trials()

best = fmin(fn=arima\_objective\_function, space=param\_space, algo=tpe.suggest, max\_evals=30, trials=trials)

plot\_forecast(df\_temp, ARIMA, (p, d, q), 100)

losses = [trial['result']['loss'] for trial in trials.trials]

best\_losses = [min(losses[:i+1]) for i in range(len(losses))]

plt.plot(range(1, len(losses)+1), best\_losses)

plt.xlabel('Итерации')

plt.ylabel('MSE')

plt.title('Уменьшение MSE')

plt.show()

data\_values.mean()

data\_values.var()

plt.figure(figsize=(15,6))

variances = daily\_means.var()

plt.hist(variances, bins=30, alpha=0.7)

plt.axvline(x=variances.mean(), color='r', linestyle='--', label='Mean')

plt.axvline(x=variances.median(), color='g', linestyle='--', label='Median')

plt.title('Распределение дисперсий')

plt.xlabel('Дисперсия')

plt.ylabel('Частота')

plt.legend()

plt.show()

var\_dict = {col: daily\_means[col].var() for col in daily\_means.columns}

var\_df = pd.DataFrame.from\_dict(var\_dict, orient='index', columns=['Variance'])

var\_df

selected\_features = var\_df[var\_df['Variance'] > 50].index

filtered\_data = daily\_means[selected\_features]

cov\_matrix = filtered\_data.corr()

plt.figure(figsize=(12, 8))

sns.heatmap(cov\_matrix, annot=True, fmt='.4f', cmap='coolwarm', square=True,

cbar\_kws={"shrink": .8}, xticklabels=cov\_matrix.columns, yticklabels=cov\_matrix.columns)

plt.title('Тепловая карта')

plt.show()

for col in ['rh', 'rho']:

plot(daily\_means[col], col)

fuller(daily\_means['rh'])

fuller(daily\_means['rho'])

for col in ['rh', 'rho']:

diff = daily\_means[col] - daily\_means[col].shift(365)

diff.dropna(inplace=True)

daily\_means[col+'\_diff'] = diff

plot(diff, col)

model = VAR(daily\_means[['temp', 'rh', 'rho']])

lag\_order = model.select\_order(maxlags=10)

print(lag\_order.summary())

model\_fit = model.fit(3, ic='hqic')

model\_fit.summary()

model\_fit = model.fit(2, ic='hqic')

model\_fit.summary()

from statsmodels.stats.stattools import durbin\_watson

dw\_stats = durbin\_watson(model\_fit.resid)

for col, stat in zip(daily\_means[['temp', 'rh', 'rho']].columns, dw\_stats):

print(f'{col}: {stat}')

forecast\_index = daily\_means.index

all\_forecasts = []

std\_errors = []

for i in range(len(forecast\_index)):

if i >= model\_fit.k\_ar:

last\_values = daily\_means[['temp', 'rh', 'rho']].iloc[i - model\_fit.k\_ar:i].values

forecast = model\_fit.forecast(last\_values, steps=1)

all\_forecasts.append(forecast[0])

cov\_matrix = model\_fit.sigma\_u

std\_error = np.sqrt(np.diag(cov\_matrix))

std\_errors.append(std\_error[0])

else:

all\_forecasts.append([None, None, None])

std\_errors.append(None)

forecast\_df = pd.DataFrame(all\_forecasts, index=forecast\_index, columns=['temp', 'rh', 'rho'])

std\_errors = np.array(std\_errors, dtype=object)

forecast\_df = forecast\_df.fillna(forecast\_df.mean())

confidence\_level = 2.56

lower\_bound = forecast\_df['temp'].copy()

upper\_bound = forecast\_df['temp'].copy()

for i in range(len(std\_errors)):

if std\_errors[i] is not None:

lower\_bound[i] -= confidence\_level \* std\_errors[i]

upper\_bound[i] += confidence\_level \* std\_errors[i]

else:

lower\_bound[i] = None

upper\_bound[i] = None

plt.figure(figsize=(15, 6))

plt.plot(daily\_means.index[-200:-7], daily\_means['temp'][-200:-7], label="Фактические значения", linewidth=4)

plt.plot(forecast\_df.index[-200:-7], forecast\_df['temp'][-200:-7], color='red', label="VAR(2)", linewidth=3)

plt.fill\_between(forecast\_df.index[-200:], lower\_bound[-200:], upper\_bound[-200:], alpha=0.2, color='k', label='Доверительный интервал прогноза')

plt.plot(forecast\_df.index[-7:], forecast\_df['temp'][-7:], color='red', linestyle='--', linewidth=3)

model\_arima = ARIMA(df\_temp, order=(1,1,2))

model\_arima\_fit = model\_arima.fit()

forecast\_arima = model\_arima\_fit.get\_prediction(start=0, end=len(df\_temp)-1)

plt.plot(forecast\_arima.predicted\_mean[-200:-7], color='g', alpha=0.5, label='ARIMA(1,1,2)', linewidth=3)

plt.plot(forecast\_arima.predicted\_mean[-7:], color='g', linestyle='dashed', linewidth=3)

plt.title('Прогноз температуры VAR(2)')

plt.xlabel('Дата')

plt.ylabel('Температура')

plt.legend()

plt.show()

print(mean\_squared\_error(daily\_means['temp'], forecast\_df['temp']))