Министерство науки и высшего образования
Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Южный федеральный университет»

ИНСТИТУТ ВЫСОКИХ ТЕХНОЛОГИЙ И ПЬЕЗОТЕХНИКИ

Кафедра информационных и измерительных технологий

Дранкин Евгений Вячеславович

ИССЛЕДОВАНИЕ ПОВЕДЕНИЯ ВРЕМЕННЫХ РЯДОВ ПРИ ПРОГНОЗИРОВАНИИ НА ОСНОВЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА МАГИСТРА
по направлению 09.04.03 — Прикладная информатика

Научный руководитель —
проф., д-р техн.наук Жмайлов Б.Б.

Рецензент —
проф., к.техн.наук Каныгин Г.И.

Ростов-на-Дону — 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ВЫСОКИХ ТЕХНОЛОГИЙ И ПЬЕЗОТЕХНИКИ

Кафедра информационных и

измерительных технологий

З А Д А Н И Е

на выпускную квалификационную работу магистра

Магистранту Дранкину Евгению Вячеславовичу

1. Тема: «Исследование поведения временных рядов при прогнозировании на основе методов интеллектуального анализа данных»

2. Срок сдачи законченной работы 10 июня 2020г.

3. Исходные данные: Нормативные документы Южного федерального университета, государственные образовательные стандарты, государственные стандарты в области информационных технологий.

4. Перечень вопросов, подлежащих разработке:

a. Изучение предметной области.
b. Анализ готовых решений.
c. Проектирование собственной системы.
d. Разработка собственной системы.
e. Тестирование разработанной системы.

5. Дата выдачи задания:

6. Руководитель _____ Жмайлов Б.Б.

Подпись *ФИО*

7. Задание принято к исполнению

_____	_____
Дата	*Подпись студента*

АННОТАЦИЯ

В работе представлены результаты исследования поведения временных рядов и возможность их прогнозирования на основе методов интеллектуального анализа данных. Также описана возможность прогнозирования движения тренда на основе индикативных показателей, при которой данные представлялись не в виде непрерывной последовательности, распределённой во времени, а как набор дискретных величин. Кроме того, проанализированы методы прогнозирования будущих цен при анализе временного ряда, с учётом индикативных показателей и без них.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В ходе развития информационных технологий и различных направлений деятельности широкое распространение получила всеобщая информатизация. В настоящий момент всецело, особенно в развитых странах, активно используются возможности, которые предоставили информационные технологии. И здесь не стали исключениями такие направления, как трейдинг и инвестиционная деятельность.

Трейдинг — работа, которой занимается трейдер: анализирует рыночную ситуацию и заключает торговые сделки.

Тре́йдер (от англ. Trader) — торговец, действующий по собственной инициативе и стремящийся извлечь прибыль непосредственно из процесса торговли. Обычно подразумевается торговля ценными бумагами (акциями, облигациями, фьючерсами, опционами) на фондовой бирже. Трейдерами также называют торговцев на валютном (форекс) и товарном рынках (например, «зернотрейдер»). Торговля осуществляется трейдером как на биржевом, так и на внебиржевом рынках [1].

Но, не стоит путать трейдера и других торговцев, которые занимаются торговлей не в своих собственных интересах лишь, а которые проводят сделки в интересах клиентов и по их требованиям (дилер, брокер, дистрибьютор).

Виды трейдеров:

1. По форме собственности:
    a. Профессиональные торговцы — работают в финасовых компаниях (банки, страховые компании, ПИФы, брокеры, дилеры). Порой имеют специальное образование и лицензию на проведение деятельности. Выполняют операции за деньги и в интересах компании или её клиентов. По российскому законодательству подобные торговцы должны иметь персональные аттестаты (ранее их выдавала ФСФР, сейчас этим занимается Банк России).
    b. Частные торговцы или независимые трейдеры — выполняют все операции полностью за свой счёт и в своих интересах (работают на

себя), для доступа к торговым системам пользуются услугами посредников (брокеров, дилеров). Проводимые ими операции, как правило не требуют лицензирования. Зачастую не имеют какого-либо специализированного образования, пользуются услугами консультантов и профессиональных торговцев.

2. По целям сделок:

a. Работа — обеспечение проведения операций или исполнение заявок клиентов (например, покупка на бирже для оплаты закупа оборудования или продажа выручки для выплаты заработной платы). Как правило, этим занимаются профессиональные торговцы.

b. Инвестор — долгосрочное инвестирование в сделку.

c. Спекулянт — торгует на разнице цен.

d. Арбитражёр — заключает противоположные сделки(одна на покупку, другая на продажу) со связанными инструментами для получения прибыли на движении цен одного актива относительно другого. В итоге общее движение цен актива нивелируется.

e. Хеджер — сделка заключается с целью уменьшения или фиксации уровня риска, например, риска изменения закупочных цен на сельскохозяйственную продукцию или котировок валют. Обычно применяется товаропроизводителями в виде опционов или фьючерсов для обеспечения возможности финансового планирования внутри производственного цикла.

3. По расположению рабочего места:

a. Трейдер на полу, трейдер в яме — обычно это внутридневные торговцы, работающие непосредственно в биржевом зале. Рабочее место располагается в самой низкой точке биржевого зала (в яме). Как правило они заключают сделки только по одному и тому же финансовому инструменту. До внедрения компьютеров в торговлю их плохо было видно, поэтому аренда места «на полу» стоила меньше, чем на ступеньках амфитеатра биржевой ямы. Трейдер на полу заключает сделку надеясь,

что через минуты или даже секунды сможет приобрести возмещающий контракт и получить с этого прибыль. Например, на рынках зерна трейдеры часто входят в сделку ради разницы в 0,0025 доллара за бушель.

b. Трейдер в зале — обычно это профессиональные торговцы, представляющие собой интересы большого числа клиентов или входящие в крупные сделки. Их рабочие места располагались выше уровня пола биржевой ямы, их лучше было видно и им гораздо лучше было видно не только других торговцев, но и информационные мониторы.

c. Трейдер у монитора торгует через специализированные торговые терминалы, которые позволяют видеть заявки других трейдеров и выставлять собственные, читать новости, просматривать историю котировок, производить её математический анализ и строить графики. Не нужно личного присутствия в биржевом зале. Разница между трейдерами на полу и в зале ликвидируется. В последнее время используется интернет, как канал связи торгового терминала с брокером или непосредственно с биржей. Сейчас именно Интернет-трейдинг сейчас является наиболее распространённой формой торговли.

4. По длительности:

a. Дневной трейдер (дейтрейдер) — заключает сделки внутри одного торгового дня (одной торговой сессии) и выходит из всех позиций перед закрытием операционного дня. Обычно имеет небольшой капитал. Закрытие позиций, как правило, мотивируется опасением гэпов («разрывов» между ценой закрытия предыдущего и ценой открытия следующего торгового дня) [2].

b. Скальпер (Пипсовщик) — совершает большое число сделок короткой продолжительности: от нескольких секунд до десятка минут (скальпинг). Как правило, результативность отдельной сделки мала, но число сделок велико(также Высокочастотный трейдинг).

c. Позиционный трейдер (краткосрочный) — заключает сделки, планируя закрытие позиций через несколько дней, закрывает все позиции

перед периодами уменьшения ликвидности (праздники, летние каникулы и т. п.)

d. Среднесрочный трейдер — заключает немного сделок за год и закрывает позиции при изменении недельных трендов.

e. Долгосрочный инвестор — позиции могут открываться на несколько лет, закрывает позиции только при изменении глобальных трендов.

Считается, что дневные и позиционные трейдеры больше работают с техническим анализом, а среднесрочные и долгосрочные инвесторы — с фундаментальным анализом рынков [1].

С развитием в настоящий момент информационных технологий стало появляться множество различных возможностей, которые многократно ускорили взаимодействие людей друг с другом. И это не обошло и биржевую сферу, где люди могут работать с финансами, совершая как долгосрочные сделки, которые будут действовать годы и десятилетия, но также имея возможность торговать необычайно быстрыми темпами, в пределах нескольких минут, секунд, а иногда и долей секунд. Стал занимать важную нишу так называемый краткосрочный трейдинг и скальпинг. Трейдеры стали работать прежде всего не на получение дивидендов или голосов в компаниях, а с целью заработать как можно больше и как можно скорее, но тут также и многократно возрастают риски.

Сейчас существуют 3 основных метода скальпинга [3]:

1. Стаканный (классический) скальпинг заключается в определении дисбаланса между объёмом спроса и предложения или определением спреда между ними, способного привести к направленному движению котировок, пусть даже незначительному. Распространен на разлчных инструментах, которые имеют конкретную базу (фьючерсы на акции).

2. Импульсный скальпинг представляет собой постоянную оценку внешних рынков и инструментов, способных вызвать импульсивное направленное движение финансового актива(при

торговле в России оценивается движение американских фьючерсов, европейских фьючерсов, нефти, доллара и т.д.). Широко распространён на фьючерсах на фондовые индексы.

3. Гибридный скальпинг сочетает черты предыдущих двух методов.

В с развитием этой сферы появилась необходимость в тех, кто умеет работать с рынками, а также в средствах для эффективной работы с ними. В настоящее время, в связи с развитием современных технологий, начали применяться новые методы обработки больших объёмов данных, в том числе и рынков, которые по своей сути представляют временной ряд. Сейчас всё чаще стали использоваться методы интеллектуального анализа данных.

ГЛАВА 1 МЕТОДЫ АНАЛИЗА ДАННЫХ

1.1 Интеллектуальный анализ данных

В последнее время огромное значение стало занимать развитие систем искусственного интеллекта, в частности интеллектуальный анализ данных. Интеллектуальный анализ данных (Data mining, добыча данных, глубинный анализ данных) [4] — это общее название, описывающее набор методов обнаружения в данных определённых шаблонов, знаний, которые помогут в принятии решений для различных задач. В 1989 года термин был введён Григорием Пятецким-Шапиро [5][6][7].

В настоящий момент английское словосочетание «data mining» не имеет устоявшегося перевода на русский язык. Вместо data mining в русском языке обычно используются другие словосочетания [8]: извлечение данных, добыча данных, просев информации, а также интеллектуальный анализ данных [9][10][11]. Одним из наиболее полных и точных является словосочетание «обнаружение знаний в базах данных» (knowledge discovery in databases, KDD).

Основой методов data mining различные методы классификации, моделирования и прогнозирования, которые, в свою очередь, основаны на применении искусственных нейронных сетей, деревьев решений, ассоциативной памяти, генетических алгоритмов, нечёткой логики, эволюционного программирования. К методам data mining порой относят статистические методы (дисперсионный анализ, дескриптивный анализ, факторный анализ, анализ временных рядов, корреляционный и регрессионный анализ, анализ связей, компонентный анализ, анализ выживаемости и дискриминантный анализ). Эти методы предполагают некоторые абстрактные представления об анализируемых данных, что расходится с целями data mining (обнаружение ранее неизвестных практически полезных и нетривиальных знаний).

Одним из самых важных назначений методологии интеллектуального анализа данных является визуализация результатов в виде различных графиков и диаграмм, что в свою очередь предоставит возможность всем, в том числе и

людям без математико-технического образования, использовать инструментарий data mining, потому как он предполагает знания определённые в статистической обработке данных и анализе, что в свою очередь предполагает знания не только математической статистикой, но и теорией вероятностей. Кроме того, инструментарий data mining может применяться как для обработки небольших объёмов данных, так и для работы с обширными наборами, например результаты деятельности компании или результаты эксперимента. Критерием достаточности набора данных являются не только область исследований, но и алгоритм, который будет применён по отношению к этим данным.

В ходе развития технологий и появления проблемы хранения знаний, стали появляться такие структуры, как базы данных и со временем для них был разработан специфический язык запросов для реляционных моделей баз данных — SQL. Он предоставляет достаточно широкие возможности по взаимодействию с хранимыми данными. Далее, в ходе широкого распространения этих структур данных, начала появляться необходимость в получении аналитических данных (к примеру, информация о расходах и доходах за некоторый промежуток времени), что, как оказалось, достаточно трудно организовать в пределах традиционных реляционных баз данных, которые в большей мере предназначаются для оперативного учёта, нежели для проведения анализа. В связи с этим были разработаны такие структуры, как “хранилища данных”, которые оказались гораздо более приспособленными для ведения математического анализа.

Знания, которые получаются методами интеллектуального анализа данных, как правило, представляются в виде различных закономерностей или шаблонов. Такими являются:

- деревья решений;
- ассоциативные правила;
- кластеры;
- математические функции.

Алгоритмы поиска этих закономерностей располагаются на пересечении областей: Математическое программирование, Визуализация, Искусственный интеллект, OLAP и Математическая статистика.

Чаще всего в интеллектуальном анализе данных применяются методы классификации и кластеризации, при работе с данными.

1.2 Классификационный и кластерный анализ

Кластерный анализ (cluster analysis) представляет собой многомерную статистическую процедуру, которая выполняет сбор данных, содержащих информацию о выборке объектов, и затем упорядочивает объекты в примерно однородные группы [12][13][14][15][16]. Задачу кластеризации относят к статистической обработке и к широкому классу задач обучения без учителя.

Многие исследователи склоняются к тому, что термин «кластерный анализ» (cluster — гроздь, сгусток, пучок) впервые был предложен математиком Р. Трионом [17]. После этого появился ряд терминов, которые в текущий момент считают синонимами термина «кластерный анализ»: автоматическая классификация, ботриология.

Применяемость кластерного анализа в настоящий момент времени необычайно широка: он используется в огромном множестве различных наук, таких как антропология, медицина, археология, химия, психология, государственное управление, биология, геология, маркетинг, филология, социология. Но в ходе активного развития и использования данного метода появился широкий спектр различных терминов, используемых при описании кластерного анализа.

Классификационный анализ или Обучение с учителем (Supervised learning) представляет собой метод машинного обучения, при котором обучение испытуемой системы происходит посредством использования различных примеров (стимул-реакий) [18]. В кибернетике такой метод называют кибернетическим экспериментом. Между входами и выходами (стимул-реакциями) возможно существование какой-то неизвестной зависимости. Только

лишь конечная совокупность прецедентов (пар «стимул-реакция») является известной и называется обучающей выборкой. Основываясь на такого рода данных, требуется восстановить зависимость (создать модель, способную прогнозировать достаточно точно и давать ответ). Чтобы измерить точность подобных ответов, вводится функционал качества. Все подобные эксперименты по сути своей являются частными случаями кибернетических экспериментов с обратной связью и, проводя такой эксперимент, предполагается наличие экспериментальной системы, метода обучения и испытания системы, либо метода измерения характеристик системы.

В свою очередь система экспериментальная, как правило, состоит из регулятора внутренних параметров (управление подкреплением), испытываемой системы и пространства стимулов, которые получаются из внешней среды. В виде системы управления подкреплением можно использовать автоматическое регулирующее устройство или человека-оператора, который будет способен реагировать на ответы испытываемой системы и стимулы внешней среды, используя определённые правила подкрепления, изменяющие состояние системной памяти. Всего различают несколько вариантов. Первый, когда реакция испытываемой системы меняет стимулы внешней среды. Второй, когда реакция испытываемой системы не меняет реакции внешней среды. Это как раз указывает на сходство подобных систем с биологическими нервными системами.

Типы входных данных:

- Признаковое описание — самый распространённый случай. Каждый объект описывается набором собственных характеристик, которые называются признаками. Признаки могут быть числовыми или нечисловыми.
- Матрица расстояний между объектами. Каждый объект описывается расстояниями до всех остальных объектов обучающей выборки. С таким типом входных данных работают немногие методы, например, метод k ближайших соседей, метод потенциальных функций и метод парзеновского окна.
- Изображение или видеоряд.

- Временной ряд — сигнал представляет собой последовательность измерений во времени. Каждое измерение может представляться числом, вектором, а в общем случае — признаковым описанием исследуемого объекта в текущий момент времени.
- Встречаются и более сложные случаи, когда входные данные представляются в виде графов, текстов, результатов запросов к базе данных, и т. д. Как правило, они приводятся к первому или второму случаю посредством предварительной обработки данных и извлечения признаков.

Типы откликов:

- Множество возможных ответов бесконечно (ответы являются векторами или действительными числами), говорят о задачах аппроксимации и регрессии;
- Множество возможных ответов конечно, говорят о задачах распознавания образов и классификации;
- Ответы характеризуют будущие поведения явления или процесса, говорят о задачах прогнозирования.

Вырожденные виды систем управления подкреплением:

- Система подкрепления с управлением по реакции (R — управляемая система) — характеризуется следующим: информационный канал от внешней среды к системе подкрепления не функционирует. Эта система, вопреки наличию системы управления, относится к спонтанному обучению, потому как испытуемая система обучается автономно, под действием только лишь своих выходных сигналов, независимо от их «действительности». При таком принципе обучения для управления изменением состояния памяти не требуется никакой внешней информации;
- Система подкрепления с управлением по стимулам (S — управляемая система) — характеризуется следующим: информационный канал от испытываемой системы к системе подкрепления не функционирует. Вопреки нефункционирующему каналу от выходов испытываемой системы, относится к обучению с учителем, так как в этом случае система подкрепления (учитель)

заставляет испытываемую систему вырабатывать реакции согласно заданному правилу, несмотря на то, что во внимание не принимается наличие истинных реакций испытываемой системы.

Это различие позволяет глубже взглянуть на различия между способами обучения, так как грань между обучением с учителем и обучением без учителя более тонкая. Кроме того, это различие позволило показать для искусственных нейронных сетей некоторые ограничения для S и R — управляемых систем, что следует из Теоремы сходимости перцептрона.

Теорема сходимости перцептрона — это теорема, описанная и доказанная Ф. Розенблаттом с участием Блока, Джозефа, Кестена и других исследователей, работавших вместе с ним. Она показывает, что элементарный перцептрон, обучаемый по методу коррекции ошибки (с квантованием или без него), независимо от последовательности появления стимулов и начального состояния весовых коэффициентов всегда приведёт к достижению решения за конечный промежуток времени [19].

Из чего следует, что возможно обучить модель, которая будет способна работать с любым набором данных, за конечное количество шагов, в том числе если эти данные будут представлять собой и временной ряд, прогнозирование которого является важной задачей в представленной работе.

Также, совместно с этими алгоритмами зачастую используются методы валидации и кросс-валидации.

1.3 Валидация и кросс-валидация

Валидация — проверка правильности работы (предсказательной способности) аналитической модели, построенной на основе машинного обучения, а также удостоверение, что она соответствует требованиям решаемой задачи [20].

Проводится на независимом, т.е. не использовавшемся при обучении и тестировании, валидационном множестве сразу после обучения и тестирования модели.

Кросс-валидация (Перекрёстная проверка) — представляет собой метод формирования обучающего и тестирующего наборов для обучения аналитической модели в условиях малого количества исходных данных или неравномерного представления классов [21][22].

Чтобы аналитическая модель успешно обучилась, ей требуется, чтобы классы были представлены в обучающем множестве примерно в одинаковой пропорции. Однако, если данных мало или процедура сэмплинга при формировании обучающего множества произведена неудачно, один из классов может оказаться доминирующим. Это может вызвать «перекос» в ходе обучения, и доминирующий класс будет рассматриваться как самый вероятный. Метод перекрестной проверки позволит избежать этого недочёта.

Основой данного метода является разделение исходного множества данных на k примерно равных блоков, например k=6. Затем на k−1, т.е. на 5-ти блоках, производится обучение данной модели, а 6-й блок используется, как тестовый. Процедура повторяется k раз, при этом на каждом из этих проходов для проверки подбирается новый блок, а обучение производится на оставшихся. В результате этого получается оценка эффективности модели с наиболее равномерным использованием имеющихся данных.

Кросс-валидация имеет следующие основные преимущества перед применением одного множества для обучения и одного для тестирования модели:

1. Распределение классов является более равномерным, что в свою очередь улучшает качество обучения.
2. Если при каждом проходе провести оценку выходной ошибки модели и усреднить ее по всем проходам, то полученная оценка будет являться более достоверной.

Совместно с методами валидации и кросс-валидации обычно используются различные методы классификации и кластеризации.

1.4 Методы классификации и кластеризации данных

В настоящий момент достаточно часто используются различные методы обработки данных, например, классификация или кластеризация.

В силу своей простоты, одним из самых популярных методов классификации данных в последнее время стал метод “Деревья решений”.

Дерево принятия решений (дерево классификации или регрессионное дерево) — это средство поддержки принятия решений, использующееся в анализе данных, машинном обучении и статистике. Структура дерева представляется в виде «листьев» и «веток». На рёбрах («ветках») дерева решения записываются атрибуты, от которых зависит целевая функция, в «листьях» записываются значения целевой функции, а в остальных узлах — атрибуты, по которым различаются случаи. Чтобы классифицировать новый случай, требуется спуститься по дереву до листа и выдать соответствующее значение. Такие деревья решений активно используются в интеллектуальном анализе данных. Цель состоит в следующем: создать модель, которая предскажет значение целевой переменной на основе нескольких переменных на входе [23].

Каждый лист является собой значением целевой переменной, которая изменяется в ходе движения от корня по листу. Каждый внутренний узел соответствует одной из входных переменных. Дерево можно также «изучить» разделением исходных наборов переменных на подмножества, которые основаны на тестировании значений атрибутов. Этот процесс повторяется на каждом из полученных подмножеств. Рекурсия завершится тогда, когда подмножество в узле будет иметь те же значения целевой переменной, таким образом, оно не добавляет ценности для предсказаний. Процесс, идущий «сверху вниз», индукция деревьев решений (TDIDT) [24], представляет собой пример поглощающего «жадного» алгоритма, и на текущий момент является самой распространённой стратегией деревьев решений для данных, но это не единственная возможная стратегия. В интеллектуальном анализе данных, деревья решений могут быть использованы в качестве вычислительных и

математических методов, чтобы помочь описать, классифицировать и обобщить набор данных.

Также, в настоящий момент не малую популярность имеет классификатор метода опорных векторов(SVC). Метод опорных векторов — набор похожих алгоритмов обучения с учителем, используемых для задач классификации и регрессии. Относится к семейству линейных классификаторов, а также может рассматриваться, как специальный случай регуляризации по Тихонову. Особым свойством метода опорных векторов является непрерывное уменьшение эмпирической ошибки классификации и увеличение зазора, поэтому метод также известен, как метод классификатора с максимальным зазором [25].

Основная идея метода — перевод исходных векторов в пространство большей размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам этой гиперплоскости, разделяющей классы. Разделяющей гиперплоскостью является гиперплоскость, которая максимизирует расстояние до двух параллельных гиперплоскостей. Алгоритм работает из предположения, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора.

Также, одним из наиболее распространённых методов классификации является MLP-классификация, представляющая собой многослойный персептрон. Многослойный перцептрон — это частный случай перцептрона Розенблатта, при котором один алгоритм обратного распространения ошибки обучает все слои. Название, по историческим причинам, не отражает особенности данного вида перцептрона, то есть не связано с тем, что в нём имеется несколько слоёв (так как несколько слоёв было и у перцептрона Розенблатта). Особенностью этой сети является наличие более, чем одного обучаемого слоя (как правило — два или три). Необходимость в большом количестве обучаемых слоёв пропадает, так как в теории единственного скрытого слоя достаточно, чтобы перекодировать входное представление таким образом, чтобы получить линейную разделимость для выходного представления. Также

порой предполагается, что, используя большее число слоёв, можно уменьшить число элементов в них, то есть суммарное число элементов в слоях будет меньше, чем если использовать один скрытый слой. Это предположение вполне успешно используется в технологиях глубокого обучения и имеет обоснование [26][27].

Эти все методы, как и многие другие, представлены в библиотеке Scikit-Learn. Она является одним из самых распространенных выборов для решения задач классического машинного обучения. Эта библиотека предоставляет широкий спектр алгоритмов обучения с учителем и без [28].

Также, для решения подобных задач может использоваться библиотека TensorFlow, которая имеет большое количество методов решения задач машинного обучения, направленных на работу с временными рядами. TensorFlow — это открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронных сетей с целью автоматического нахождения и классификации образов, достигая качества, близкого к человеческому восприятию [29][30]. В настоящий момент она применяется как для исследований, так и для разработки собственных продуктов Google. Эта библиотека является продолжением закрытого проекта DistBelief. Изначально TensorFlow была разработана командой Google Brain только с целью внутреннего использования в Google, однако в 2015 году система была переведена в свободный доступ с открытой лицензией Apache 2.0 [31][32].

1.5 Временные ряды

1.5.1 Автокорреляция временных рядов

Временной ряд (или ряд динамики) [33] — собранный в разные моменты времени статистический материал о значении каких-либо параметров (в простейшем случае одного) исследуемого процесса. Каждая единица статистического материала называется измерением или отсчётом, также допустимо называть его уровнем на указанный с ним момент времени. Во временном ряде для каждого отсчёта должно быть указано время измерения или

номер измерения по порядку. Временной ряд существенно отличается от простой выборки данных, так как при анализе учитывается взаимосвязь измерений со временем, а не только статистическое разнообразие и статистические характеристики выборки [34].

Временные ряды состоят из двух элементов:

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

Временные ряды классифицируются по следующим признакам:

- по форме представления уровней:
  - средних величин;
  - относительных показателей;
  - ряды абсолютных показателей.
- по характеру временного параметра: интервальные и моментные временные ряды. В моментных временных рядах уровни указывают значения показателя по состоянию на заданные моменты времени. В интервальных же уровни указывают значения показателей за определенные периоды времени. Важной особенностью интервальных временных рядов абсолютных величин является возможность суммирования их уровней. Некоторые уровни моментного ряда абсолютных величин содержат элементы повторного счёта. Из-за этого суммирование уровней моментных рядов становится бессмысленным;
- по числу показателей, для которых определяются уровни в каждый момент времени: много- и одно- мерные ряды;
- временные ряды разделяют на детерминированные и случайные: первые получают на основе значений некоторой неслучайной функции (ряд последовательных данных о количестве дней в месяцах); вторые есть результат реализации некоторой случайной величины.

- в зависимости от наличия основной тенденции выделяют стационарные ряды, в которых среднее значение и дисперсия постоянны, и нестационарные, содержащие основную тенденцию развития;
- по наличию пропущенных значений: полные и неполные временные ряды;
- по расстоянию между датами и интервалами времени выделяют равноотстоящие — даты регистрации или окончания их периодов следуют друг за другом с равными интервалами и неполные (неравноотстоящие) — когда принцип равных интервалов не соблюдается [34];

В свою очередь, временные ряды прогнозируются при помощи различных методов, при этом, предполагается, что существует зависимость между текущими значениями ряда и предыдущими (историческими), из-за чего предполагается, что будущие значения ряда можно спрогнозировать, отталкиваясь при этом от текущих. Предполагается, что между данными есть автокорреляция.

Автокорреляция — статистическая взаимосвязь между последовательностями различных величин одного ряда, которые были взяты со сдвигом, например, для случайного процесса — со сдвигом по времени.

Это понятие достаточно широко используется в эконометрике. Наличие автокорреляции случайных ошибок регрессионной модели приводит к ухудшению качества МНК-оценок параметров регрессии, а также к завышению тестовых статистик, по которым проверяется качество модели (то есть создается искусственное улучшение качества модели относительно её действительного уровня точности). Именно поэтому тестирование автокорреляции случайных ошибок является требуемой процедурой построения модели регрессии [35].

Автокорреляция ожидается во время “близости” наблюдений во времени или пространстве, наличии ненаблюдаемого фактора, действующего на “соседние” наблюдения. Но даже в простых примерах ситуации автокоррелированные ошибки оценки могут оказаться несостоятельными.

Пример. Введём понятия:

E — математическое ожидание

Var — дисперсия

Cov — ковариация

Corr — корреляция

Отсутствие коррелируемости ошибок:

$\mathrm{Cov}(\varepsilon_i, \varepsilon_j|X)= \mathrm{E}(\varepsilon_i, \varepsilon_j|X)=0$

$i \neq j$— проверка независимости разных ошибок

Автокорреляцию изучают{Анализ временных рядов
Пространственная эконометрика

$$E(y_t) = \beta_1 + \beta_2 x_t$$

$y_t = \beta_1 + \beta_2 x_t + [\varepsilon_t = 0]$— если бы не было ошибок

Предположим что:

$$\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \ldots = \varepsilon_n$$

и все они одновременно принимают одни и теже значения:

ε_t	-1	1
Вероятность	1/2	1/2

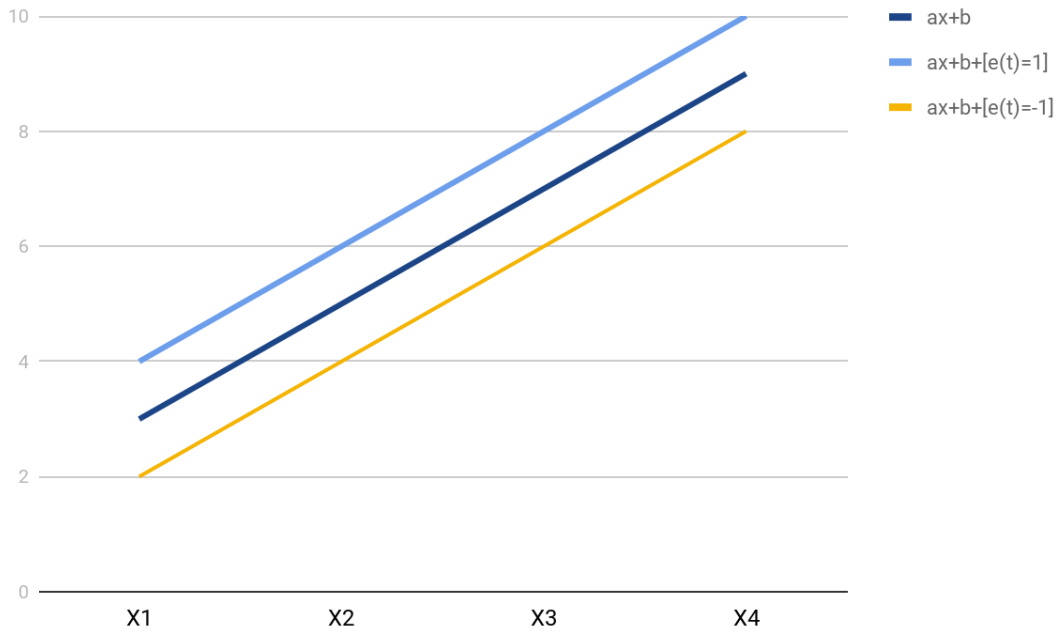

Рисунок 1. Несостоятельность автокоррелируемой ошибки от случайной величины

Соответственно, сколько бы ни было наблюдений, даже если $n \to \infty$, увидев одну из этих прямых(рис. 1), никогда не удастся понять где находилась настоящая, т.к. с ростом наблюдений по-прежнему нет информации о том чему были равны ε.

p=1 — автокорреляция 1 порядка.

$\varepsilon_t = \rho\varepsilon_{t-1} + U_t \quad -1 \leqslant \rho \leqslant 1$

U_t— независимы между собой, одинаково распределены, независимы от регрессоров.

$E(U_t) = 0 \; Var(U_t) = \delta_u^2$

Например, возьмём $\rho = \frac{1}{2}$

Но т.к. $\varepsilon_t = \rho\varepsilon_{t-1} + U_t$, а $\varepsilon_{t-1} = \rho\varepsilon_{t-2} + U_{t-1}$, то подставим в первое уравнение второе и получим: $\varepsilon_t = \frac{1}{2}\varepsilon_{t-1} + U_t = \frac{1}{2}\left(\frac{1}{2}\varepsilon_{t-2} + U_{t-1}\right) + U_t =$

$\frac{1}{2}\left(\frac{1}{2}\left(\frac{1}{2}\varepsilon_{t-3} + U_{t-2}\right) + U_{t-1}\right) + U_t =$

$= U_t + \frac{1}{2}U_{t-1} + \left(\frac{1}{2}\right)^2 U_{t-2} + \left(\frac{1}{2}\right)^3 \varepsilon_{t-3} = U_t + \frac{1}{2}U_{t-1} + \left(\frac{1}{2}\right)^2 U_{t-2} +$

$\left(\frac{1}{2}\right)^3 U_{t-3} + \ldots$

Вывод: Дисперсия постоянна и равна $Var(\varepsilon_t) = \delta_u^2$

Предполагаем, что U_t— независимы, одинаково распределены

$E(U_t) = 0 \; Var(U_t) = \delta_u^2$

$Cov(\varepsilon_t, \varepsilon_{t-1}) = \gamma_1 \qquad Cov(\varepsilon_t, \varepsilon_{t-2}) = \gamma_2$— от t(времени) не меняются. Найдём их.

$$Var(\varepsilon_t) = Var(\rho\varepsilon_t) + Var(U_t) + 2Cov(\rho\varepsilon_{t-1}, U_t)$$

Т.к. в ε_{t-1} U_t не входит(по пред уравнению), то $Cov(\rho\varepsilon_{t-1}, U_t) = 0$

$\delta_\varepsilon^2 = \rho^2\delta_\varepsilon^2 + \delta_U^2 + 0 \qquad \delta_\varepsilon^2 = \frac{\delta_U^2}{1-\rho^2} Cov(\varepsilon_t, \varepsilon_{t-1}) = Cov(\rho\varepsilon_{t-1} + U_t, \varepsilon_{t-1}) =$

$\rho Cov(\varepsilon_{t-1}, \varepsilon_{t-1}) + Cov(U_t, \varepsilon_{t-1}) = \rho\delta_\varepsilon^2$

δ_ε^2 — Дисперсия

$\varepsilon_{t-1} = U_{t-1} + \rho U_{t-2} \quad Cov(U_t, \varepsilon_{t-1})$=0

$Cov(\varepsilon_t, \varepsilon_{t-2}) = Cov(\rho\varepsilon_{t-1} + U_t, \varepsilon_{t-2}) = \rho Cov(\varepsilon_{t-1}, \varepsilon_{t-2}) + Cov(U_t, \varepsilon_{t-2}) =$

$$= \rho Cov(\varepsilon_{t-1}, \varepsilon_{t-1}) = \rho\rho\delta_\varepsilon^2 = \rho^2\delta_\varepsilon^2$$

Получаем, что ковариация между 2мя измерениями, отстоящая на k шагов:

$$\underline{Cov(\varepsilon_t, \varepsilon_{t-k}) = \rho^k\delta_\varepsilon^2}$$

Вычислим из этого корреляцию двух отстоящих друг от друга измерений на k шагов:

$$Corr(\varepsilon_t, \varepsilon_{t-k}) = \frac{Cov(\varepsilon_t, \varepsilon_{t-k})}{\sqrt{Var(\varepsilon_t)Var(\varepsilon_{t-k})}} = \frac{\rho^k\delta_\varepsilon^2}{\delta_\varepsilon^2} = \rho^k$$

Для корреляции 1-го порядка: $-1 < \rho < 1$

$Corr(\varepsilon_t, \varepsilon_{t-k}) = \rho^k$ ⇒Чем дальше 2 ошибки друг от друга по времени, тем меньше связь между ними.

$$\varepsilon_t = \varphi_1\varepsilon_{t-1} + \varphi_2\varepsilon_{t-2} + \ldots + \varphi_p\varepsilon_{t-p} + U_t$$

U_t— одинаково распределены, независимы между собой и от регрессоров

$E(U_t) = 0\ Var(U_t) = \delta^2$

В отличие от автокорреляции первого порядка, автокорреляция порядка p допускает более богатую(сложную) структуру $Corr(\varepsilon_i, \varepsilon_j)$

Если автокорреляция первого порядка убывала по модулю степени k, то автокорреляция порядка p начинает себя вести произвольно, хотя общая тенденция соблюдается: Наблюдения, далеко отстоящие по времени, независимы:

$$\lim_{k\to\infty} Corr(\varepsilon_t, \varepsilon_{t-k}) = 0$$

Условная автокорреляция и другие предпосылки

- автоматически нарушена предпосылка о независимости наблюдений (x_i, y_i)
- Во временных рядах обычно нарушена предпосылка $E(\varepsilon_t|X) = 0$.

Например, использование y_{t-1}в качестве регрессора нарушает $E(\varepsilon_t|X) = 0$(условие о строгой экзогенности).

Для возможности включения прошлого значения зависимой переменной в регрессоры есть 2 подхода: (1)ослабить предпосылки и (2) использование

принципиально другого метода — метод максимального правдоподобия и работать с ним, а не с методом наименьших квадратов и не с его свойствами. В основном большая часть временных рядов связана построены на этом методе, поэтому не будут разбираться варианты с ослаблением предпосылок, которые подходят, для того, чтобы принять метод наименьших квадратов.

Метод максимального правдоподобия:

Для оценки коэффициентов: $\hat{\beta} = (X'X)^{-1}X^{-1}y$

Для оценки ковариационной матрицы оценок коэффициентов: $\widehat{Var}(\hat{\beta}|X) = \frac{RSS}{n-k}(X'X)^{-1}$, где RSS — сумма квадратов остатков

В частности $\widehat{Var}(\hat{\beta}|X) = \frac{\hat{\delta}^2}{RSS_j}$ и $se(\hat{\beta}_j) = \sqrt{\widehat{Var}(\beta_j|X)}$

3 группы свойств:

- Конечная выборка без предположения о нормальности
- Конечная выборка с предположением о нормальности ε
- Асимптотические свойства без предположения о нормальности $\varepsilon(\varepsilon \to \infty)$

Конечная выборка без предположения о нормальности ε:

V — Линейность по y(сохраняется)

V — Несмещенность $E(\hat{\beta}|X) = \beta$, $E(\hat{\beta}) = \beta$

X — Оценка эффективности среди линейных несмещенностей(не сохраняется)

Конечная выборка о предположении о нормальности ε(теряем все свойства которые были при выполнении всех предпосылок классической линейной модели регрессии):

X — $\frac{\hat{\beta}_j - \beta_i}{se(\hat{\beta}_i)}|X \sim t_{n-k}$

X — $\frac{RSS}{\delta^2}|X \sim X^2_{n-k}$

X — $\frac{(RSS_R - RSS_{UD})/r}{RSS_{UR}/(n-k)} \sim F_{r,n-k}$

Асимптотические свойства:

V — $\hat{\beta} \to \beta$— по прежнему состоятельная оценка для β

V — $\frac{RSS}{n-k} \to \delta^2$

X — $\frac{\hat{\beta}_j - \beta_j}{se(\hat{\beta}_j)} \to N(0,1)$

X — $\frac{RSS_R - RSS_{UR}}{RSS_{UR}/(n-k)} \to X_r^2$

- Сами $\hat{\beta}$ можно интерпретировать и использовать
- Стандартные ошибки(считаемые по стандартным формулам) se($\hat{\beta}_j$) несостоятельны

Используя обычные se($\hat{\beta}_j$), нельзя строить доверительные интервалы или проверять гипотезы.

1.5.2 Робастные стандартные ошибки и тест Дарбина-Уотсона

Вместо обычной оценки ковариационной матрицы $\widehat{Var}(\hat{\beta}|X)$ используется матрица гетероскедастичности $\widehat{Var_{HAC}}(\hat{\beta}|X)$

$\widehat{Var}(\hat{\beta}|X)$— обычная $\widehat{Var_{HAC}}(\hat{\beta}|X)$— другая стандартная ошибка

$se_{HAC}(\hat{\beta}_j)$ — асимптотическое распределение

Робастная(устойчивая) к условной гетероскедастичности и автокорреляции оценка ковариационной матрицы

- Вместо $\widehat{Var}(\hat{\beta}|X) = \frac{RSS}{n-k}(X'X)^{-1}$используем $\widehat{Var_{HAC}}(\hat{\beta}|X)(X'X)^{-1} = (X'X)^{-1}\hat{\Phi}(X'X)^{-1}$
- Нью-Вест(Newey-West), 1987г.

$$\hat{\Phi} = \sum_{j=-k}^{k} \frac{k-|j|}{k}\left(\sum_{t} \hat{\varepsilon}_t\hat{\varepsilon}_t + jx'_t x_{t+j}\right)$$

Суть корректировки:

Меняем $se\ (\hat{\beta}_j)$на $se_{HAC}(\hat{\beta}_j)$

Какие проблемы решены?

- $\frac{\widehat{\beta}_j-\beta_j}{se_{HAC}(\widehat{\beta}_j)} \to N(0,1)$ — с ростом числа испытаний всё больше становится похоже на нормальное распределение
- Можем проверить гипотезу о β_j
- Можем строить доверительные интервалы для β_j

Проблемы:

X - Оценки β_j эффективны

X - $\frac{\widehat{\beta}_j-\beta_i}{se(\widehat{\beta}_i)}|X \sim t_{n-k}$

X - $\frac{RSS}{\delta^2}|X \sim X^2_{n-k}$

X - $\frac{(RSS_R-RSS_{UD})/r}{RSS_{UR}/(n-k)} \sim F_{r,n-k}$

На практике в R:

- Оценка модели методом наименьших квадратов model←lm(data=data,y~x+z)
- Считаем рабочую ковариационную матрицу пакет Sandwich VCOVHAC(model)
- Тестирование(пакет lmtest) cocftest(model,vcov=vcovHAC). Когда на практике нет — использовать робота стандартной ошибки;

Использовать требуется:

- Когда подозреваем наличие автокорреляции и не хотим заниматься её моделированием

Обнаружение автокорреляции:

- Оцениваем модель с помощью МНК(метод наименьших квадратов)
- Строим график остатков на осях $\hat{\varepsilon}_{t-1}, \hat{\varepsilon}_t$

Рисунок 2. График остатков.

Графики указывают(рис. 2) на прямую, обратную и отсутствие автокорреляции.

Формальные тесты на автокорреляцию:

- Тест Дарбина-Уотсона(только для тестирования корреляции р 1-го порядка)

$$\varepsilon_t = \rho\varepsilon_{t-1} + U_t$$

- Нормальность ошибок ε
- Сильная экзогенность $E(\varepsilon_t|X) = 0$
- Но об отсутствии автокорреляции, $\rho = 0$

1) Оценить основную регрессию, получить $\hat{\varepsilon}_t$
2) Посчитать статистику $DW = \frac{\sum_{i=2}^{n}(\hat{\varepsilon}_t - \hat{\varepsilon}_{t-1})^2}{\sum_{i=1}^{n}\hat{\varepsilon}_t^{\,2}}$

Если $\hat{\rho}$выборочная корреляция остатков, то

$DW = 2(1 - \hat{\rho})$, поэтому 0<DW<4

Если $DW \approx 0$ ($\hat{\rho} \approx 0$),то ∃сильная положительная автокорреляция

Если $DW \approx 2$ ($\hat{\rho} \approx 1$),то ∃отсутствие автокорреляции

Если $DW \approx 4$ ($\hat{\rho} \approx -1$),то ∃сильная отрицательная автокорреляция

С практической точки зрения:

- Если р-значение больше уровня значимости α(e.g. 5%), то H_0об отсутствии автокорреляции не отвергается
- Если р-значение меньше уровня значимости α(e.g. 5%), то H_0об отсутствии автокорреляции отвергается
- Существуют таблицы диапазонов критических значений

1.5.2 Стационарность временных рядов. Процесс скользящего среднего

Временные ряды бывают многомерные(год, население, ВВП) и одномерные(курс $)

Временной ряд — набор случайных чисел

Базовым предположением будет предположение о стационарности ряда(неизменность во времени, но y_tне являются константами, а имеют постоянные характеристики во времени)

Временной ряд называется стационарным, если

$$E(y_1) = E(y_2) = E(y_3) =..$$

$$Var(y_1) = Var(y_1) = Var(y_1) =... = \gamma_0$$

$$Cov(y_1, y_2) = Cov(y_2, y_3) = Cov(y_3, y_4) =... = \gamma_1$$

$$Cov(y_1, y_3) = Cov(y_2, y_4) = Cov(y_3, y_5) =... = \gamma_2$$

Значит, стационарный, если

$$E(y_t) = const$$

$Cov(y_t, y_{t-k}) = \gamma_k$— (авто)-ковариационная функция процесса

Самый простой пример стационарного процесса — это белый шум.

Ряд ε_t— белый шум, если

$E(y_t) = 0,\ Var(y_1) = \delta^2,\ Cov(y_t, y_{t-k}) = 0,\ y_t = \varepsilon_t \sim$ N(0,4) — белый шум(пример)

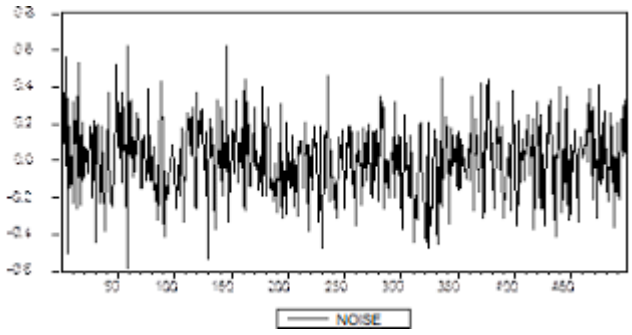

Рисунок 3. Белый шум $y_t = \varepsilon_t \sim$ N(0,1)

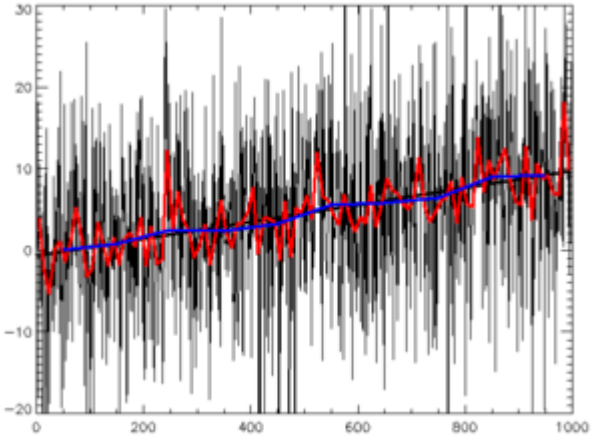

Рисунок 4. Трендовый временной ряд $y_t = 2 + 0.01t + \varepsilon_t \sim$ N(0,6^2)

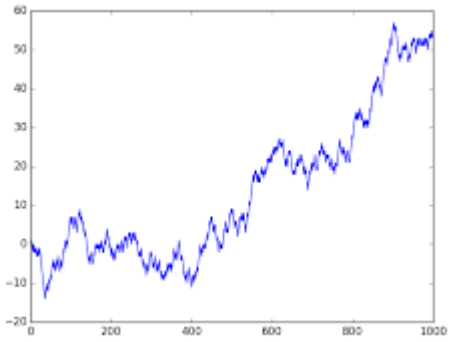

Рисунок 5. Случайное блуждание(нестационарный процесс)

Примеры нестационарных процессов:

- пример с детерминистическим трендом(рис. 3, рис. 4);
- случайное блуждание(рис. 5).

$$y_0 = 0$$

$$y_t = y_{t-1} + 2 + \varepsilon_t$$

$Var(y_t) = t\delta^2$— чем дальше отходит по времени, тем меньше вероятности, что он будет находиться около нуля.

Процессы скользящего среднего являются более сложными примерами стационарных процессов, по сравнению с белым шумом.

Процесс скользящего среднего можно представить в виде:

$$y_t = \mu + \varepsilon_t + a_1\varepsilon_{t-1} + \ldots + a_q\varepsilon_{t-q}$$

$$y_t = Ma(q) — \ Moving\ Average\ (q - \text{порядок})$$

Рассмотрим пример:

$y_t = 5 + \varepsilon_t + 3\varepsilon_{t-1} - 2\varepsilon_{t-2}$, $\varepsilon_t \sim N(0, \sigma^2)$

Вопросы:

a) $Ma(?) \rightarrow MA(2)$

b) $E(y_t)? \rightarrow E(y_t) = E(5 + \varepsilon_t + 3\varepsilon_{t-1} - 2\varepsilon_{t-2}) =$

$=E(5) + E(\varepsilon_t) + 3E(\varepsilon_{t-1}) - 2E(\varepsilon_{t-2}) = 5$, т.к. предполагаем, что ε_t— это белый шум

c) $Var(y_t)?$

d) $\gamma_k = Cov(y_t, y_{t-k})?$

c)$Var(y_t) = Var(5 + \varepsilon_t + 3\varepsilon_{t-1} - 2\varepsilon_{t-2}) =$

$= Var(\varepsilon_t) + Var(3\varepsilon_{t-1}) + Var(-2\varepsilon_{t-2}) = Var(\varepsilon_t) + 9Var(\varepsilon_{t-1}) + 4Var(\varepsilon_{t-2})$

$= 14\sigma^2$

т.к. предполагаем, что ε_t— это белый шум и белый шум в разный момент времени взаимо независим, поэтому дисперсия суммы равна сумме дисперсий.

d) $\gamma_0 = Cov(y_t, y_{t-0}) = Var(y_t) = 14\sigma^2$

$\gamma_1 = Cov(y_t, y_{t-1}) = Cov(5 + \varepsilon_t + 3\varepsilon_{t-1} - 2\varepsilon_{t-2}, 5 + \varepsilon_{t-1} + 3\varepsilon_{t-2} - 2\varepsilon_{t-3}) =$

$= Cov(3\varepsilon_{t-1}, \varepsilon_{t-1}) + Cov(-2\varepsilon_{t-2}, 3\varepsilon_{t-2}) = 3Var(\varepsilon_{t-1}) - 6Var(\varepsilon_{t-2}) = -3\sigma^2$

$\gamma_2 = Cov(5 + \varepsilon_t + 3\varepsilon_{t-1} - 2\varepsilon_{t-2}, 5 + \varepsilon_{t-2} + 3\varepsilon_{t-3} - 2\varepsilon_{t-4}) = -2\sigma^2$

$\gamma_3 = \gamma_5 = \ldots = \gamma_k = 0$

$|14\sigma^2, k = 0$

$\gamma_k = |-3\sigma^2, k = 1$

$|-2\sigma^2, k = 2$

$|0, k \geq 3$

Уравнения во временных рядах удобно записывать при помощь оператора Лага.

L — оператор Лага: $Ly_t = y_{t-1}$

$LLy_t = L^2 y_t = y_{t-2}$

$Ma(2): y_t = 2 + \varepsilon_t + 3\varepsilon_{t-1} - 2\varepsilon_{t-2} \Rightarrow y_t = 2 + (1 + 3L - 2L^2)\varepsilon_t$

1.5.3 Автокорреляционная и частотная автокорреляционная функция

Коэффициенты процесса скользящего среднего плохо интерпретируемые и для этого используют так называемую автокорреляционную функцию с процесса(у стационарного):

$\rho_k = Corr(y_t, y_{t-k})$ — (авто−)корреляционнная функция процесса

По смыслу для стационарного процесса с нормально-распределенными y_t:

Если y_t — стационарный процесс и $y_t \sim N(\mu_y, \sigma_y^2)$, то ρ_k — на сколько изменится y_t при росте y_{t-k} на единицу.

Пример:

$$y_t = 5 + \varepsilon_t + 3\varepsilon_{t-1} - 2\varepsilon_{t-2}$$

$\rho_k = Corr(y_t, y_{t-k})$ — ?

$Corr(y_t, y_{t-k}) = \frac{Cov(y_t, y_{t-k})}{\sqrt{Var(y_t), Var(y_{t-k})}} = \frac{Cov(y_t, y_{t-k})}{Var(y_t)} = \frac{\gamma_k}{\gamma_0} = \rho_k$(Дисперсии равны, т.к. процесс стационарный $Var(y_t) = Var(y_{t-k})$)

$$\gamma_k = \begin{cases} 14\sigma^2, k = 0 \\ -3\sigma^2, k = 1 \\ -2\sigma^2, k = 2 \\ 0, k \geq 3 \end{cases}$$

$\rho_0 = \frac{\gamma_0}{\gamma_0} = 1$; $\rho_1 = \frac{\gamma_1}{\gamma_0} =$

$-\frac{3}{14}$ — если y_{t-1} выпал на 1, то y_t упал на $\frac{3}{14}$

$$\rho_2 = \frac{\gamma_2}{\gamma_0} = -\frac{2}{14};$$

$$\rho_3 = \rho_4 = \ldots = 0$$

Помимо автокорреляционной функции у каждого стационарного процесса есть частная автокорреляционная функция

Рассмотрим 4 случайных величины, следующих подряд

$y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_4$. Мы хотим измерить прямой эффект y_1 на y_4. Существует, конечно, совокупный цепочный эффект(изменяется 1я, из-за этого 2я, затем 3я и 4я), а частная автокорреляция показывает прямой эффект y_1 на y_4, при неизменности остальных величин.

Формальное определение частной автокорреляции:

$\phi_k = Corr(y_t - P(y_t), y_{t-k} - P(y_{t-k}))$, где $P(y_t)$ — проекция случайной величины y_t на линейную оболочку величин $y_{t-1}, y_{t-2}, \ldots, y_{t-k+1}$(очищаем зависимость от влияния случайных промежуточных величин)

Способ подсчёта: решение линейных уравнений.

$$\gamma_0\phi_1 = \gamma_1$$

$|\gamma_0 *_1 + \gamma_1\phi_2 = \gamma_1$

$|\ \gamma_1 *_1 + \gamma_0\phi_2 = \gamma_2$

$*_1$,$*_2$— вспомогательные, ненужные переменные

$|\gamma_0 *_1 + \gamma_1 *_2 + \gamma_2\phi_3 = \gamma_1$

$|\gamma_1 *_1 + \gamma_0 *_2 + \gamma_1\phi_3 = \gamma_2$

$|\gamma_2 *_1 + \gamma_1 *_2 + \gamma_0\phi_3 = \gamma_3$

Пример:

$y_t = 5 + \varepsilon_t + 3\varepsilon_{t-1} - 2\varepsilon_{t-2},\ \varepsilon_t$— белый шум $Var(\varepsilon_t) = \sigma^2$

$$\varphi_1, \varphi_2, \varphi_3, \ldots$$

$$\gamma_k = Cov(y_t, y_{t-k}) = \begin{cases} 14\sigma^2, k = 0 \\ -3\sigma^2, k = 1 \\ -2\sigma^2, k = 2 \\ 0, k \geq 3 \end{cases}$$

$\gamma_0\varphi_1 = \gamma_1\ \ \varphi_1 = \frac{\gamma_1}{\gamma_0} = -\frac{3}{14}$

$\varphi_1 = \rho_1 \quad y_{t-1} \to y_t$

$$\varphi_2 = \begin{cases} \gamma_0 *_1 + \gamma_1\varphi_2 = \gamma_1 \\ \gamma_1 *_1 + \gamma_0\varphi_2 = \gamma_2 \end{cases} \qquad \begin{matrix} 14 *_1 - 3\varphi_2 = -3 \\ -3 *_1 + 14\varphi_2 = 14 \end{matrix}$$

$$\Rightarrow 187\varphi_2 = -37$$

$\varphi_2 = -\frac{37}{187}\rho_2 = -\frac{2}{14}$

φ_2— прямой эффект ρ_2— совокупный эффект

φ_3: $\begin{cases} \gamma_0 *_1 + \gamma_1 *_2 + \gamma_2\varphi_3 = \gamma_1 \\ \gamma_1 *_1 + \gamma_0 *_2 + \gamma_1\varphi_3 = \gamma_2 \\ \gamma_2 *_1 + \gamma_1 *_2 + \gamma_0\varphi_3 = \gamma_3 \end{cases}$

$|14 *_1 - 3 *_2 - 2\varphi_3 = -3$

$|-3 *_1 + 14 *_2 - 3\varphi_3 = -2$

$|-2 *_1 - 3 *_2 + 14\varphi_3 = 0$

$\Rightarrow 2762\varphi_3 = -111$

$$\varphi_3 = -\frac{111}{2762}$$

1.5.4 Процесс авторегрессии

Процесс авторегрессии — это стационарный процесс вида:

$$y_t = c + b_1 y_{t-1} + b_2 y_{t-2} + \ldots + b_p y_{t-p} + \varepsilon_t$$

y_t~AR(p) — AutoRegression ε_t— белый шум

Пример:

AR(1) — $y_t = 2 + 0.5y_{t-1} + \varepsilon_t$ $Var(\varepsilon_t) = \sigma^2 \rightarrow$ стационарный процесс

E(y_t)? ρ_k? φ_k?

E(y_t)=$2 + 0.5E(y_{t-1}) + 0$

E(y_t) — 0.5E(y_{t-1})=2

E(y_t)=4

Вычтем из уравнения математическое ожидание и получим:

$2 + 0.5y_{t-1} + \varepsilon_t \equiv (y_t - 4) = 0.5(y_{t-1} - 4) + \varepsilon_t$

$\gamma_k = Cov(y_t, y_{t-k})$ $\gamma_0 = Cov(y_t, y_t)$ $\gamma_1 = Cov(y_t, y_{t-1})$

$\gamma_0 = Cov(2 + 0.5y_{t-1} + \varepsilon_t\ , 2 + 0.5y_{t-1} + \varepsilon_t) = 0.25\gamma_0 + \sigma^2$($\gamma_0$— это ковариация между y_{t-1}и y_{t-1}, σ^2— это ковариация между ε_tи ε_t, а ковариации между ε_tи y_{t-1}нет, т.к. y_{t-1}— это текущий у, а ε_t— это будущий шум и на сегодняшний у он не влияет)

$$\gamma_1 = Cov(2 + 0.5y_{t-1} + \varepsilon_t\ , y_{t-1}) = 0.5\gamma_0$$
$$\gamma_2 = Cov(2 + 0.5y_{t-1} + \varepsilon_t\ , y_{t-2}) = 0.5\gamma_1$$
$$\gamma_3 = Cov(2 + 0.5y_{t-1} + \varepsilon_t\ , y_{t-3}) = 0.5\gamma_2$$

$0.75\gamma_0 = \sigma^2$ $\Rightarrow$ $\gamma_0 = \frac{4}{3}\sigma^2$

$\gamma_1 = 0.5\,\gamma_0$ $\gamma_2 = 0.25\,\gamma_0$ $\gamma_3 = 0.5^3\gamma_0$

$$\rho_1 = \frac{\gamma_1}{\gamma_0} = 0.5 \qquad \rho_2 = \frac{\gamma_2}{\gamma_0} = 0.25 \qquad \rho_k = 0.5^k$$

1.5.5 Расчёт частной автокорреляционной функции AR(1) процесса и множественность их решений

$$y_t = 2 + 0.5y_{t-1} + \varepsilon_t \qquad \varphi_1 = 0.5;\ \varphi_2 = 0;\ \varphi_3 = 0$$

$$\gamma_1 = 0.5\gamma_0$$

$$\gamma_2 = 0.5^2\gamma_0$$

…

$$\gamma_k = 0.5^k\gamma_0$$

$$\varphi_1 = \gamma_0 \quad \varphi_1 = \gamma_2 \quad \varphi_1 = \frac{\gamma_1}{\gamma_0} = \rho_1 = 0.5$$

$$\varphi_2 = |\gamma_0 *_1 + \gamma_1\varphi_2 = \gamma_1 \qquad |\gamma_0 *_1 + \gamma_0 0.5\varphi_2 = \gamma_0 0.5$$

$$|\gamma_1 *_1 + \gamma_0\varphi_2 = \gamma_2 \qquad |\gamma_0 0.5 *_1 + \gamma_0\varphi_2 = \gamma_0 0.5^2$$

$$|*_1 + 0.5\varphi_2 = 0.5$$

$$|0.5 *_1 + \varphi_2 = 0.5^2$$

$$\Rightarrow 0.5 *_1 + 0.25\varphi_2 = 0.5^2$$

$$0.75\varphi_2 = 0 \qquad \varphi_2 = 0 \qquad \varphi_3 = 0$$

$$|*_1 + 0.5 *_2 + 0.5^2\varphi_3 = 0.5$$

$$\varphi_3 = \ |0.5 *_1 + *_2 + 0.5\varphi_3 \ = 0.5^2$$

$$|0.5*_1 + 0.5 *_2 + \varphi_3 \ = 0.5^3$$

$$\begin{array}{ccccccccccc} & & \varepsilon_1 & & \varepsilon_2 & & \varepsilon_3 & & \varepsilon_4 & & \varepsilon_5 \\ & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \rightarrow & & y_1 & \rightarrow & y_2 & \rightarrow & y_3 & \rightarrow & y_4 & \rightarrow & y_5 \\ & & & & & & \uparrow & & & & \end{array}$$

Фиксирован, когда считаем φ_2

На y_tвлияет y_{t-1}(вчерашний) и ε_t(сегодняшний)

Возьмём y_2 и y_4. $\rho_2 > 0$— что он меряет? Если y_2был много выше среднего, то это приведёт к тому, что y_3с большей долей вероятности будет выше среднего и y_4также предполагается, что будет выше среднего уровня. Естественно данный эффект ослабевает, чем длиннее расстояние между элементами. ρ убывает, т.к. случайные составляющие ε снижают важность этого эффекта. ρ_2измеряет совокупный эффект y_2 на y_4.

Что измеряет φ_2? Измеряет влияние y_2 на y_4, если y_3был зафиксирован, поэтому воздействия y_2на y_4, т.к. по формуле y_4 зависит только от y_3и ε_4. Поэтому у AR(1) процесса$\varphi_2 = \varphi_3 = \varphi_4 = \ldots = 0$.

Из одного только уравнения $y_t = 2 + 0.5y_{t-1} + \varepsilon_t$ стационарность автоматически никак не следует и не выводится(!)

Определим на примере, что уравнение имеет бесконечное количество решений.

$y_t = 2 + 0.5y_{t-1} + \varepsilon_t \quad \forall t \quad \varepsilon_t -$ белый шум. $\varepsilon_t \sim N(0,1)$

Возьмём $y_0 = 0$

$y_1 = 2 + \varepsilon_t \qquad y_2 = 2 + (1 + 0.5\varepsilon_t) + \varepsilon_2 = \ldots = 1 + 0.25$

$E(y_0) = 0 \qquad y_1 \sim N(2,1) \;\; y_2 \sim N(3; 1.25) \ldots \;\; Var(y_0) = 0$

И хотя y_0, y_1, y_2, y_3 удовлетворяют этому уравнению, однако явно этот процесс нестационарный, т.к. у него меняется и математическое ожидание(сначала 0, потом 2, затем 3), и дисперсия(сначала 0, потом 1, затем 1.25), поэтому данное решение этого разностного уравнения не является стационарным. Однако, у этого уравнения есть и стационарное решение:

$E(y_t) = 4 \quad Var(y_t) = \frac{4}{3}\sigma^2 = \frac{4}{3} * 1$

Предположим, что $y_0 \sim N(4; \frac{4}{3})$ и не зависит от $\varepsilon_1, \varepsilon_2, \ldots$

$y_1 = 2 + 0.5y_0 + \varepsilon_1 \qquad E(y_1) = 2 + 0.5E(y_0) = 2 + 0.5 * 4 = 4$

$$Var(y_1) = 0.25Var(y_0) + Var(\varepsilon_1) = \frac{1}{4} * \frac{4}{3} + 1 = \frac{4}{3}$$

$y_1 \sim N(4; \frac{4}{3}) \qquad y_2 = 2 + 0.5y_2 + \varepsilon_2 \Rightarrow y_2 \sim N(4; \frac{4}{3})$

Вывод: Это разностное уравнение имеет множество решений, среди которых ∃ и стационарное.И зависит от начального условия.

1.5.6 Стационарность через характеристический многочлен

Если записываем уравнение, то если оно имеет хотя бы 1 стационарное решение, то предполагаем именно его. Если уравнение не имеет стационарных решений, то это нестационарный процесс, а стационарный процесс подразумевает, что есть хотя бы 1 стационарное решение уравнения.

AR можно записать с помощью Лага

$$y_t = 2 + 0.5y_{t-1} - 0.006y_{t-2} + \varepsilon_t$$

$$(1 - 0.5L + 0.06L^2)y_t = 2 + \varepsilon_t$$

Стационарность процесса AR можно определить только исходя из сомножителя с Лагом, находящийся перед y_t, поэтому это выражение называется характеристический многочлен:

$f(L)y_t = c + \varepsilon_t$ $f(L)$ − характеристический многочлен

И стационарность процесса определяется корнями этого характеристического многочлена.

Если корни характеристического уравнения AR процесса по модулю больше 1, то существует единственное стационарное решение, в котором y_t выражается через прошлые шумы ε_t , ε_{t-1} , …

$$y_t = 7 + 0.5y_{t-1} - 0.006y_{t-2} + \varepsilon_t$$

$$y_t = 7 + 0.5Ly_t - 0.006L^2y_t + \varepsilon_t$$

$(1 - 0.5L + 0.06L^2)y_t = 7 + \varepsilon_t$ $f(L) = 1 - 0.5L + 0.06L^2$

$1 - 0.5x + 0.06x^2 = 0$ $x_1 = \frac{0.6}{0.12}$ $x_2 = \frac{0.4}{0.12}$

$|x_1| > 1$ $|x_2| > 1$ $\Rightarrow$ есть стационарное решение

Пример 2:

$y_t = -3 + 1.2y_{t-1} + 0.2y_{t-2} + \varepsilon_t$

$$(1 - 1.2L + 0.2L^2)y_t = 3 + \varepsilon_t$$

$f(L) = 1 - 1.2L + 0.2L^2$ $x_1 = 1$ $x_2 = \frac{1}{0.2}$

$|x_1| = 1 \quad |x_2| > 1 \quad \Rightarrow$ нет стационарного решения

Пример 3:

$$y_t = 2 + 2y_{t-1} - 2y_{t-2} + \varepsilon_t$$

$$(1 - 2L + 2L^2)y_t = 2 + \varepsilon_t$$

$$f(L) = 1 - 2L + 2L^2$$

$|x_1| = \left|\frac{1+i}{2}\right| = \left|\frac{1}{2} + \frac{1}{2}i\right| < 1 \qquad |x_2| = \left|\frac{i-1}{2}\right| = \left|\frac{1}{2}i - \frac{1}{2}\right| < 1$

$\Rightarrow$ нет стационарных решений

Существует две трактовки нахождения стационарности уравнения:

1. $f(L)$ – хар мн. $f(L)y_t = C + \varepsilon_t \qquad |x| > 1 \ \forall x_i$
2. $f(L)y_t = C + \varepsilon_t * f * x_1$

$\lambda_1 = \frac{1}{x_1} \quad \lambda_2 = \frac{1}{x_2} \ldots \quad \lambda_p = \frac{1}{x_p}$

$|\lambda_i| < 1 \quad \forall \lambda_i$ – условие стационарности

1.5.7 Прогнозирование процессов авторегрессии. Модель авторегрессии и скользящего среднего(ARMA) и его оценка

Прогноз на h шагов вперёд: $E(y_{t+h}|y_t, y_{t-1}, y_{t-2}, \ldots) \equiv \widehat{y_{t+h}}$

На следующих примерах построим прогноз на 1 и 2 шага вперёд, построим точечный и интервальный прогноз(предиктивный интервал)

$y_t = 2 + 0.5y_{t-1} - 0.006y_{t-2} + \varepsilon_t, \quad \varepsilon_t \sim N(0; 4)$

Имеем данные по прошлым наблюдениям $y_1, \ldots, y_{100}$

$y_{99} = 3 \; y_{100} = 4$

a) $\widehat{y_{100+1}}, \widehat{y_{100+2}}$ — построить точечный прогноз

b) $Var(y_{101} - \widehat{y_{100+1}}|y_1 \ldots y_{100})$

$Var(y_{102} - \widehat{y_{100+2}}|y_1 \ldots y_{100})$ — посчитать дисперсию ошибки прогноза

c) Построить 95% предиктивный(доверительный) интервал для будущих y_{101} и y_{102}

a) $\widehat{y_{100+1}} = E(y_{101}|y_{100,}y_{99},\dots) = E(2 + 0.5y_{100} - 0.06y_{99} + \varepsilon_{101}|y_{100}, y_{99},\dots) =$

$=2 + 0.5 * 4 - 0.06 * 3 + 0 = 2 + 2 - 0.18 = 3.82$ (ε_{101} и $y_{100}, y_{99},\dots$ — не зависят)

$\widehat{y_{100+2}} = E(y_{102}|y_{100,}y_{99},\dots) = E(2 + 0.5y_{101} - 0.06y_{100} + \varepsilon_{102}|y_{100}, y_{99},\dots) =$

$=2 + 0.5 * 3.82 - 0.06 * 4 + 0 = 2 + 1.91 - 0.24 = 3.67$

(ε_{102} и $y_{100}, y_{99},\dots$ — не зависят)

b) $Var(y_{101} - \widehat{y_{100+1}}|y_1 \dots y_{100}) =$(т.к. прогноз строится только на основе того, что известно и $\widehat{y_{100+1}}$ также строится на основании того, что известно, поэтому с нашей точки зрения это известная величина и она на условную дисперсию никак не влияет. Поэтому упрощаем)

$=Var(y_{101}|y_1 \dots y_{100}) = Var(2 + 0.5y_{100} - 0.06y_{99} + \varepsilon_{101}|y_1 \dots y_{100}) =$

$= Var(\varepsilon_{101}|y_1 \dots y_{100})$=($\varepsilon_{101}$— будущий шум никак не зависит от текущих y, поэтому условная дисперсия становится безусловной)

$=Var(\varepsilon_{101}) = 4$

$$Var(y_{102} - \widehat{y_{100+2}}|y_1 \dots y_{100}) = Var(y_{102}|y_1 \dots y_{100}) =$$

$$= Var(2 + 0.5y_{101} - 0.06y_{100} + \varepsilon_{102}|y_1 \dots y_{100}) = Var(0.5y_{101} + \varepsilon_{102}|y_1 \dots y_{100})$$

$=$

$=Var(0.5(2 + 0.5y_{100} - 0.06y_{99} + \varepsilon_{101}) + \varepsilon_{102}|y_1 \dots y_{100}) =$

$=Var(0.5\varepsilon_{101} + \varepsilon_{102}|y_1 \dots y_{100}) = Var(0.5\varepsilon_{101} + \varepsilon_{102}) = 0.5^2 * 4 + 4 = 5,$

$\varepsilon_t \sim N(0,4)$

c) 95% предиктивный интервал для нормального распределения(рис. 6)

$$y_{101}: \left[\widehat{y_{100+1}} - 1.96\sqrt{4}; y_{100+1} + 1.96\sqrt{4}\right] = [-0.1; 7.74]$$

$$y_{102}: \left[\widehat{y_{100+2}} - 1.96\sqrt{5}; y_{100+2} + 1.96\sqrt{5}\right] = [-0.713; 8.05]$$

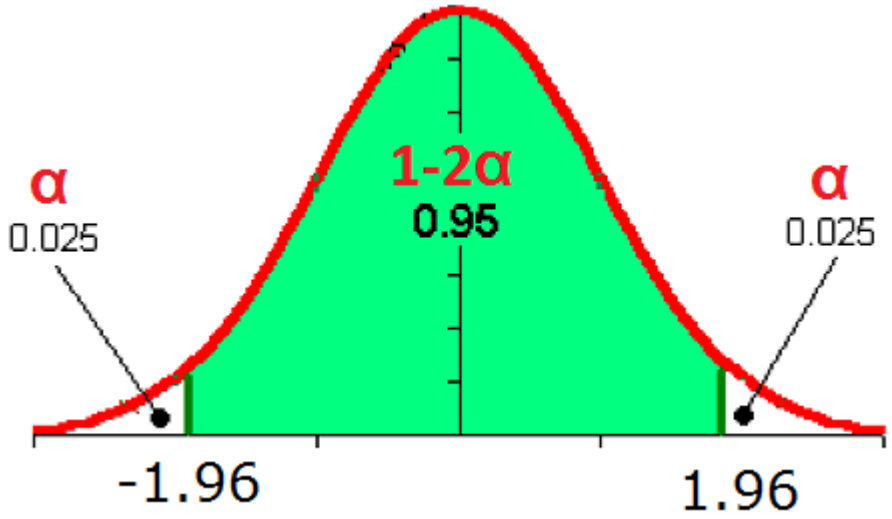

Рисунок 6. 95% предиктивный интервал

- Стационарный процесс вида:

$y_t = C + b_1 y_{t-1} + b_2 y_{t-2} + \ldots + b_p y_{t-p} + \varepsilon_t + a_1 \varepsilon_{t-1} + \ldots + a_q \varepsilon_{t-q}$, где сумма p+q минимально возможная

$y_t \sim ARMA(p, q)$, где p — количество Лагов по части авторегрессии, а q — количество Лагов по части скользящего среднего

- $y_t = \varepsilon_t \equiv y_t - y_{t-1} = \varepsilon_t - \varepsilon_{t-1}$ — выбираем кратчайший процесс

$y_t \sim ARMA(0,0)$

ARMA — это в каком-то смысле всё, что необходимо знать про стационарные процессы

Теорема: Любой стационарный процесс можно представить в виде $AR(\infty)$

Вывод: С помощью ARMA(p,q) можно компактно и очень точно описать любой стационарный процесс

Итого про ARMA(p,q)

- коэффициенты не интерпретируемые
- используются для прогнозирования

Оценивание коэффициентов:

$\exists T$ наблюдений: $y_1, y_2, y_3, \ldots, y_T$. Чаще всего используется метод максимального правдоподобия

- Предполагается независимость и нормальность $\varepsilon_t \sim N(0, \sigma^2)$
- Стационарность y_t

Результат метода максимального правдоподобия:

оценки: $\hat{\theta} = (\hat{c}, \widehat{a_1}, \ldots, \widehat{a_q}, \widehat{b_1}, \ldots, \widehat{b_q}, \widehat{\sigma^2})$

И оценка их ковариационной матрицы: $\widehat{Var}(\hat{\theta})$

Проверка гипотез и доверительных интервалов(только асимптотически — только при большом числе выборок). Здесь результатов для малых выборок нет даже в предположении нормальности

$$\frac{\widehat{a_j} - a_j}{se(\widehat{a_j})} \to N(0,1)$$

Выборочная автокорреляционная функция:

ACF — autocorrelation function

По имеющимся реальным данным $y_1, \ldots, y_t$ можно оценить неизвестную автокорреляционную функцию, вычислив каждую выборочную автокорреляцию следующим методом:

$$\widehat{\rho_k} = \frac{\sum_{t=k+1}^{T} \left(y_t - \underline{y}\right)\left(y_{t-k} - \underline{y}\right)}{\sum_{t=1}^{T} \left(y_t - \underline{y}\right)^2}$$

Выборочная частная автокорреляционная функция:

PACF — partial autocorrelation function

Аналогичным образом можно оценить частную автокорреляционную функцию

Получаем $\widehat{\phi_k}$ из оценки регрессии

$$\widehat{y_t} = * + * y_{t-1} + * y_{t-2} + \ldots + * y_{t-k+1} + \phi_k y_{t-k} + \upsilon_t$$

Примечания к расчёту частной автокорреляционной функции:

- Для оценки каждого $\widehat{\phi_k}$ строится отдельная регрессия
- Из каждой регрессии нужен только последний коэффициент

Алгоритм на практике:

- Строим графики ряда ACF, PACF и определяем характеристики ряда
- Если ряд нестационарный, то преобразуем
- Выбираем p и q
- Оцениваем ARMA(p,q)

Основное преобразование нестационарного вида к стационарному: взятие разности: переход от y_tк Δy_t. Мы моделируем не денежную массу, а её изменение

Обозначение:

- $y_t \sim ARIMA(p,1,q)$ равносильно $\Delta y_t \sim ARMA(p,q)$
- $y_t \sim ARIMA(p,0,q)$ равносильно $y_t \sim ARMA(p,q)$

Выбор p и q по графикам

Теоретическая автокорреляционная функция ρ_kи частотная автокорреляционная функция ϕ_k, определены для стационарных процессов!

Выборочная автокорреляционная функция $\widehat{\rho_k}$и выборочная частотная автокорреляционная функция $\widehat{\phi_k}$ существуют всегда! Даже у нестационарного процесса, но у него эта оценка является бессмысленной, т.к. она позволяет узнать процесс, но настоящая автокорреляционная функция не существует и оценка, соответственно, не несёт смысл оценки. Примеры:

Белый шум $y_t = \varepsilon_t$(рис. 7) $\forall t$ независимы $\Rightarrow$ нулевые автокорреляции ACF(рис. 8) и нулевые частные автокорреляции PACF (рис. 9)

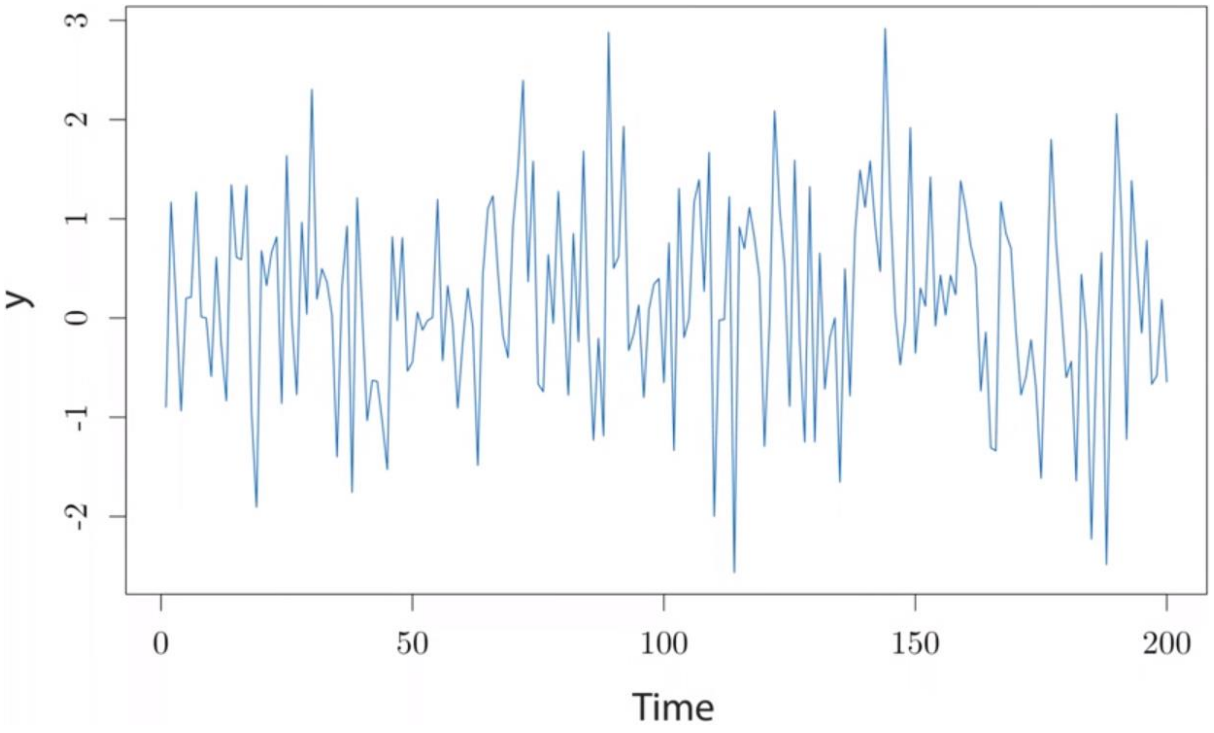

Рисунок 7. График белого шума, $y_t = \varepsilon_t$

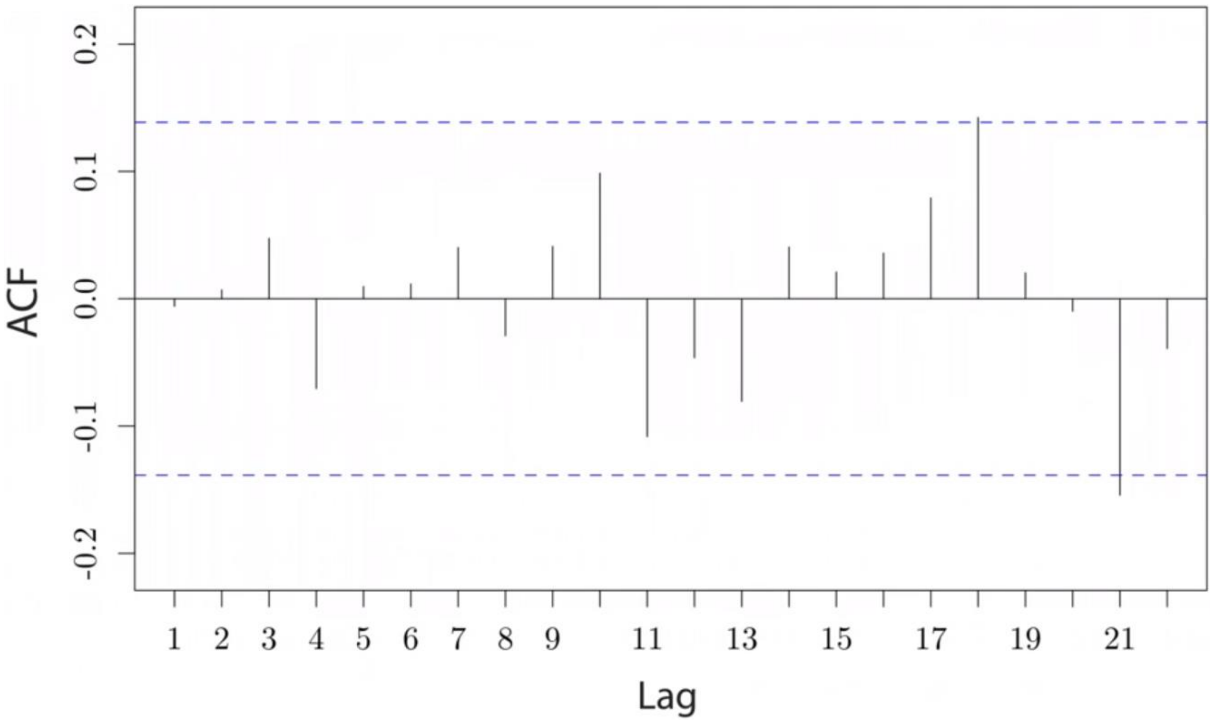

Рисунок 8. ACF, $y_t = \varepsilon_t$

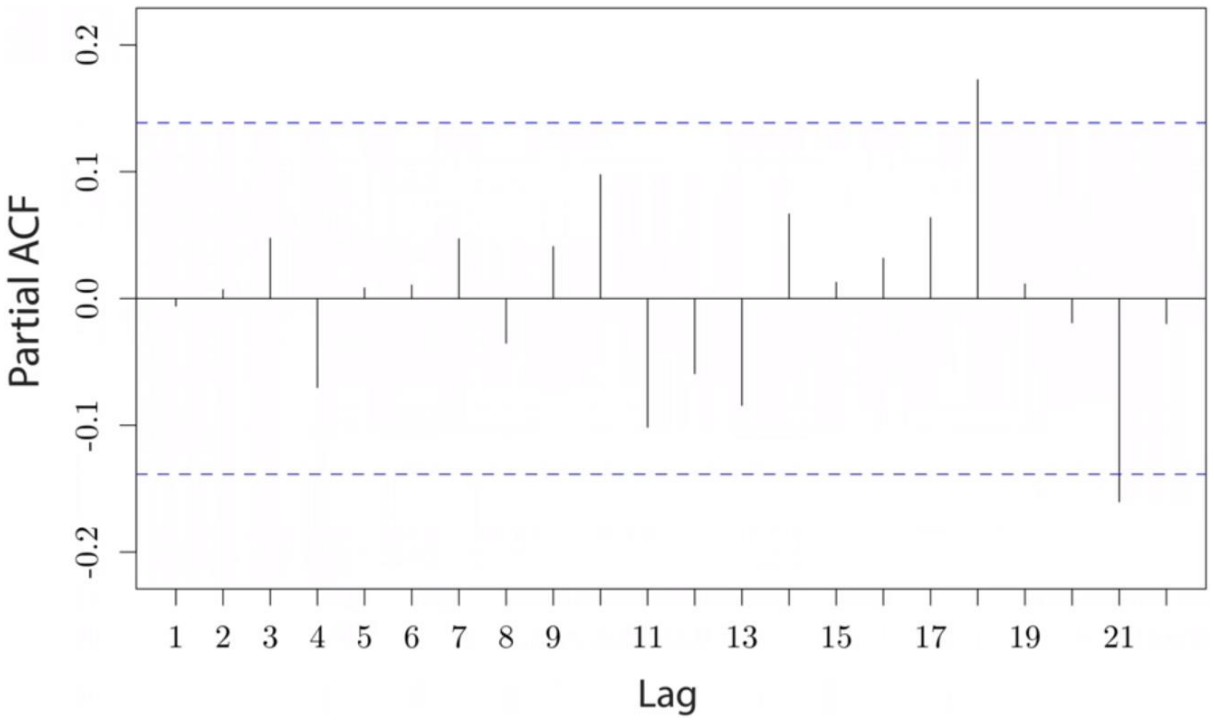

Рисунок 9. PACF, $y_t = \varepsilon_t$

При случайном блуждании(рис. 10) функция может сильно отходить от нуля и ACF(рис. 11) и PACF(рис. 12) медленно убывают — это серьёзный признак случайного блуждания. $y_t = y_{t-1} + \varepsilon_t$— это нестационарный процесс.

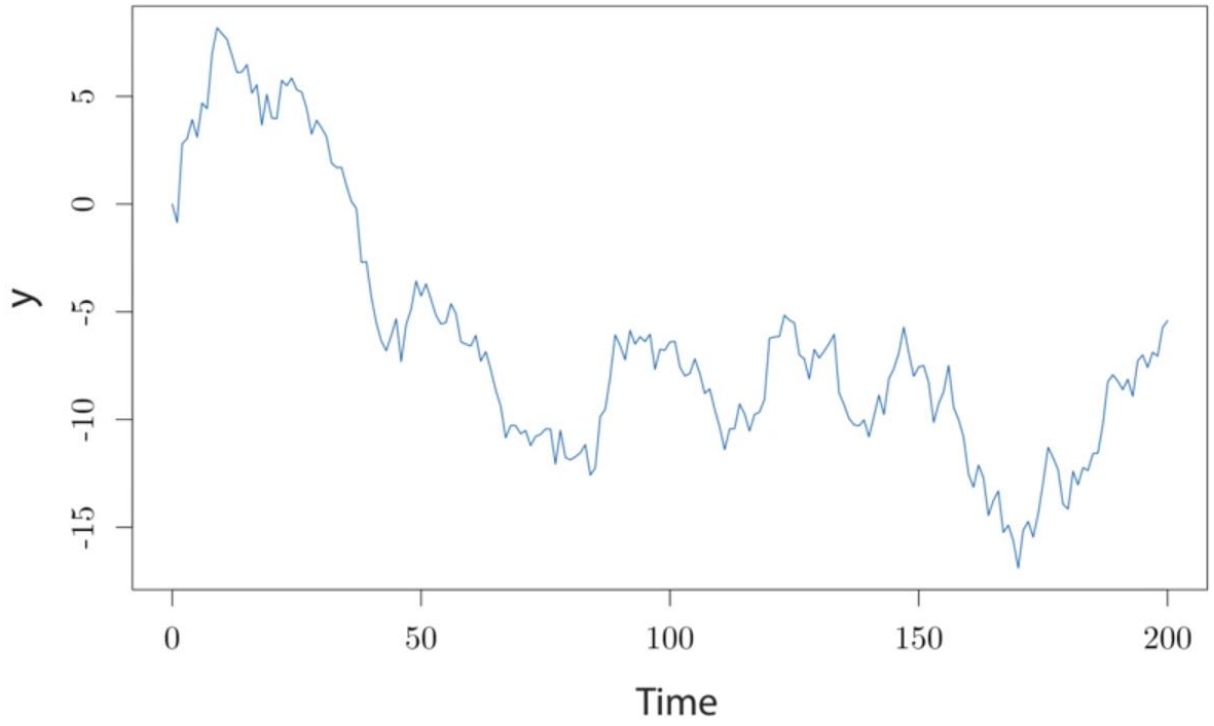

Рисунок 10. Случайное блуждание, $y_t = y_{t-1} + \varepsilon_t$

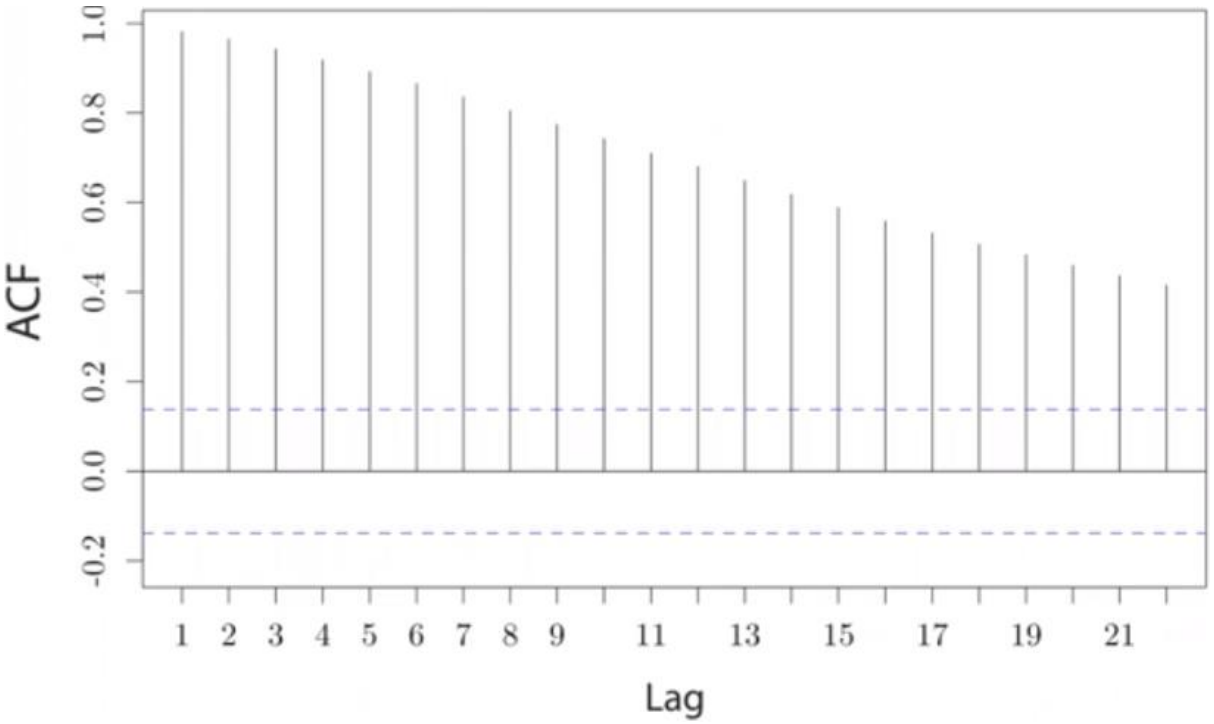

Рисунок 11. AFC, $y_t = y_{t-1} + \varepsilon_t$

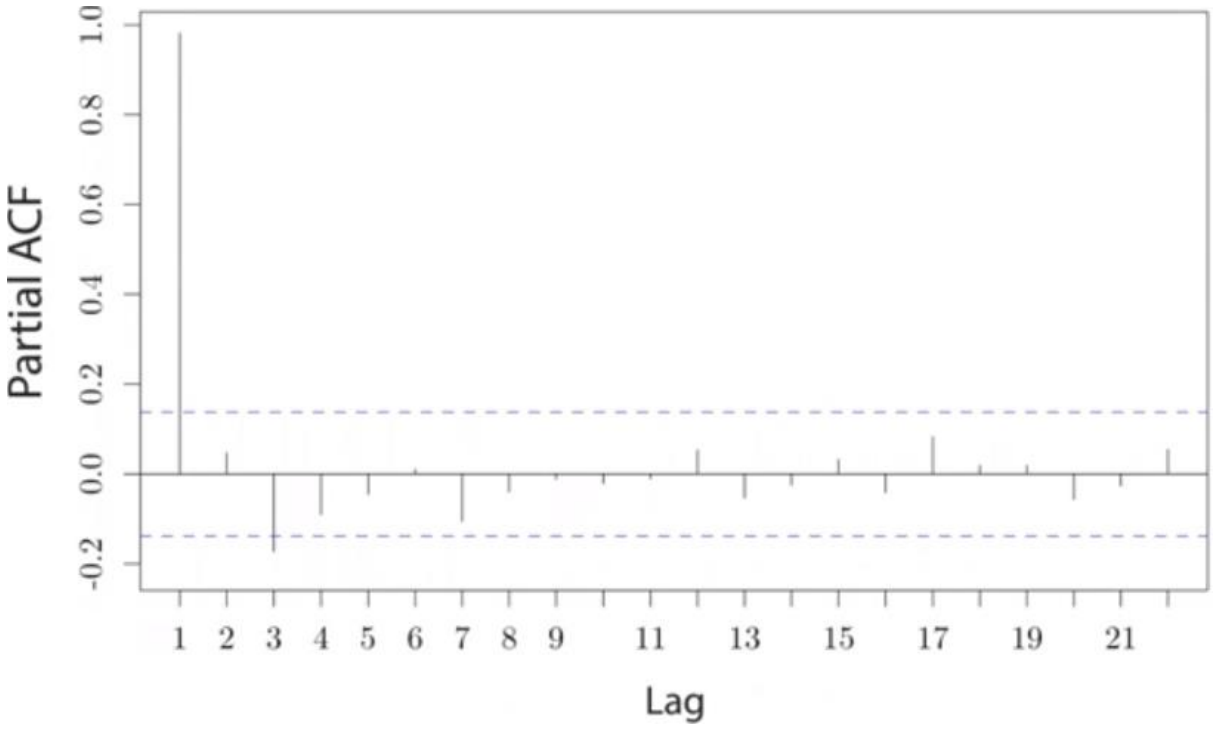

Рисунок 12. PAFC, $y_t = y_{t-1} + \varepsilon_t$

Аналогичный процесс с трендом. Это нестационарный процесс, у него не определена автокорреляционная функция, однако есть оценка(рис. 14) и она плавно убывает. Частная автокорреляционная функция убывает довольно резко(рис. 15) и ряд на самом графике для исходного ряда колеблется вокруг тренда(рис. 13)

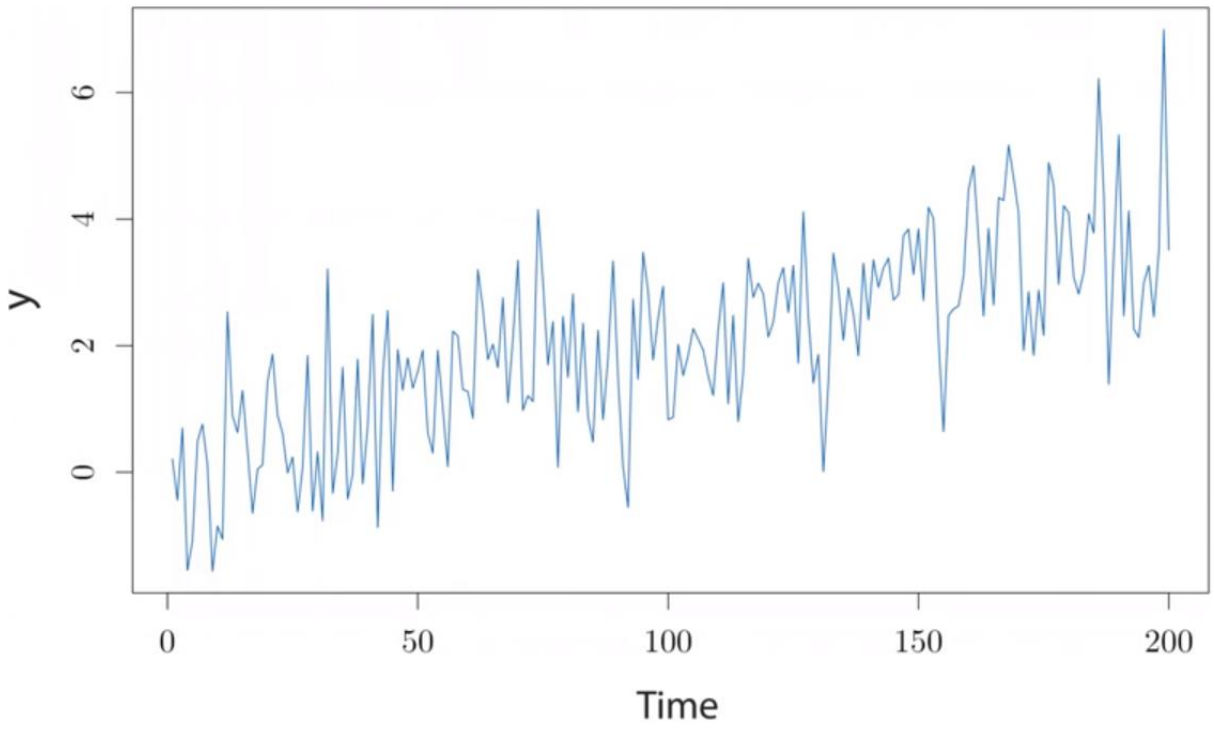

Рисунок 13. Процесс с трендом, $y_t = 0.02t + \varepsilon_t$

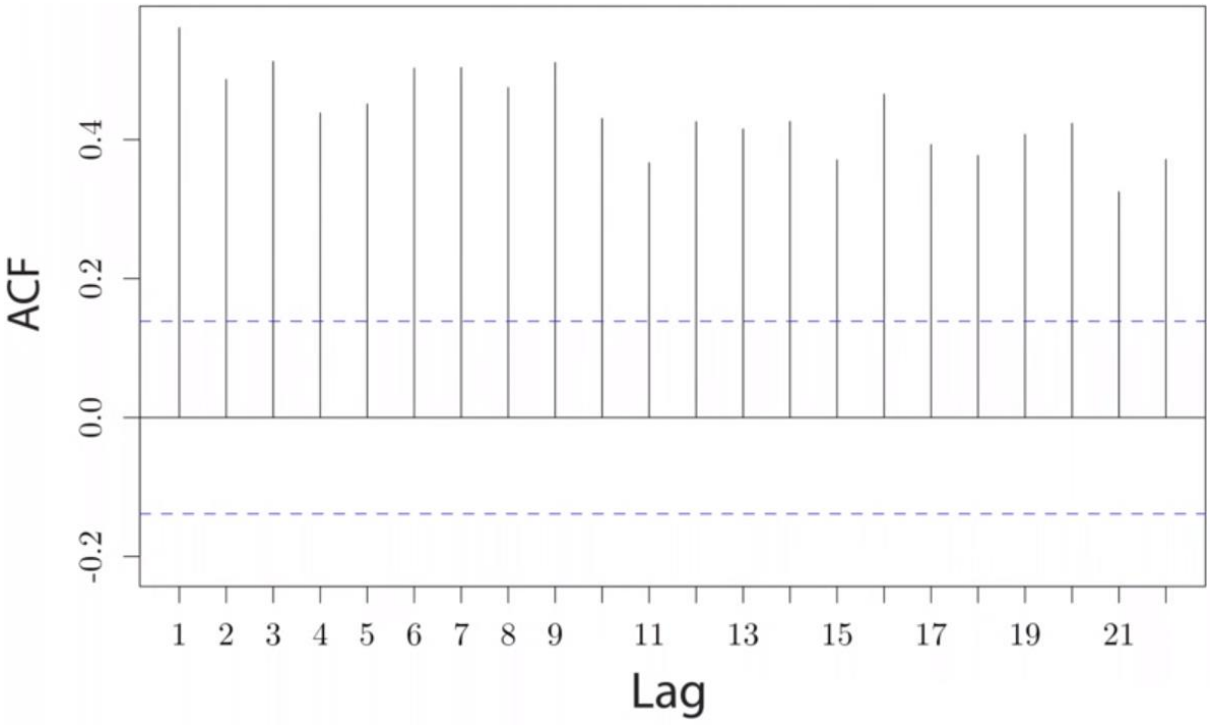

Рисунок 14. ACF, $y_t = 0.02t + \varepsilon_t$

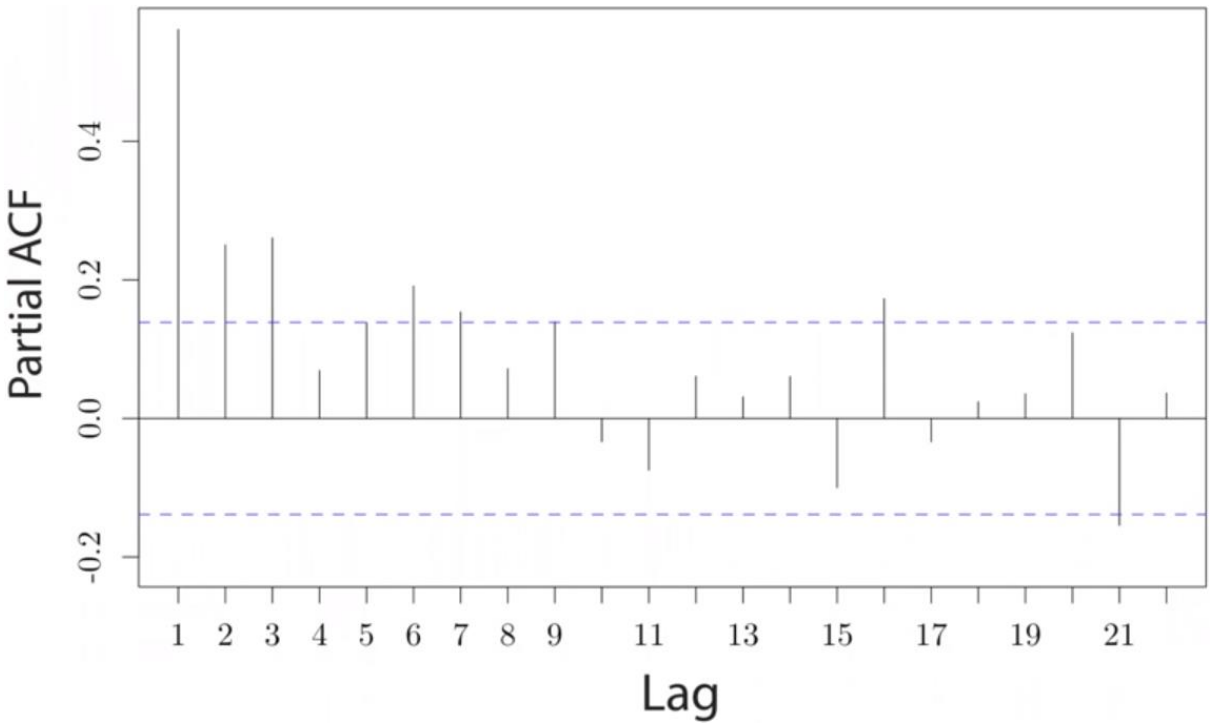

Рисунок 15. PACF, $y_t = 0.02t + \varepsilon_t$

Движение с трендом. График вокруг линии тренда движется, есть оценки ACF и PACF-процессов.

- Временные ряды: стационарные и нестационарные
- Стационарные моделируются с помощью ARMA

- Нестационарные приводятся к стационарным

Аналогичным образом выглядят графики для AR(1) (рис. 16) и AR(2) (рис. 17) процессов. По графику самого ряда его практически не отличить от белого шума(рис. 7), однако, если посмотреть на графики автокорреляционной функции(рис. 18, рис. 19) и частной автокорреляционной функции(рис. 20, рис. 21), то можно увидеть закономерность. График автокорреляционной функции довольно резко убывает к нулю, но тем не менее не ноль некоторое время, а график частной автокорреляционной для AR(1) процесса (рис. 20), только первое значение не равняется нулю, когда остальные практически все нулевые, а для AR(2) процесса (рис. 21) первые 2 значения частной автокорреляционной функции не нули, тогда как остальные нулевые. На графиках границы доверительного интервала указаны пунктирной линией.

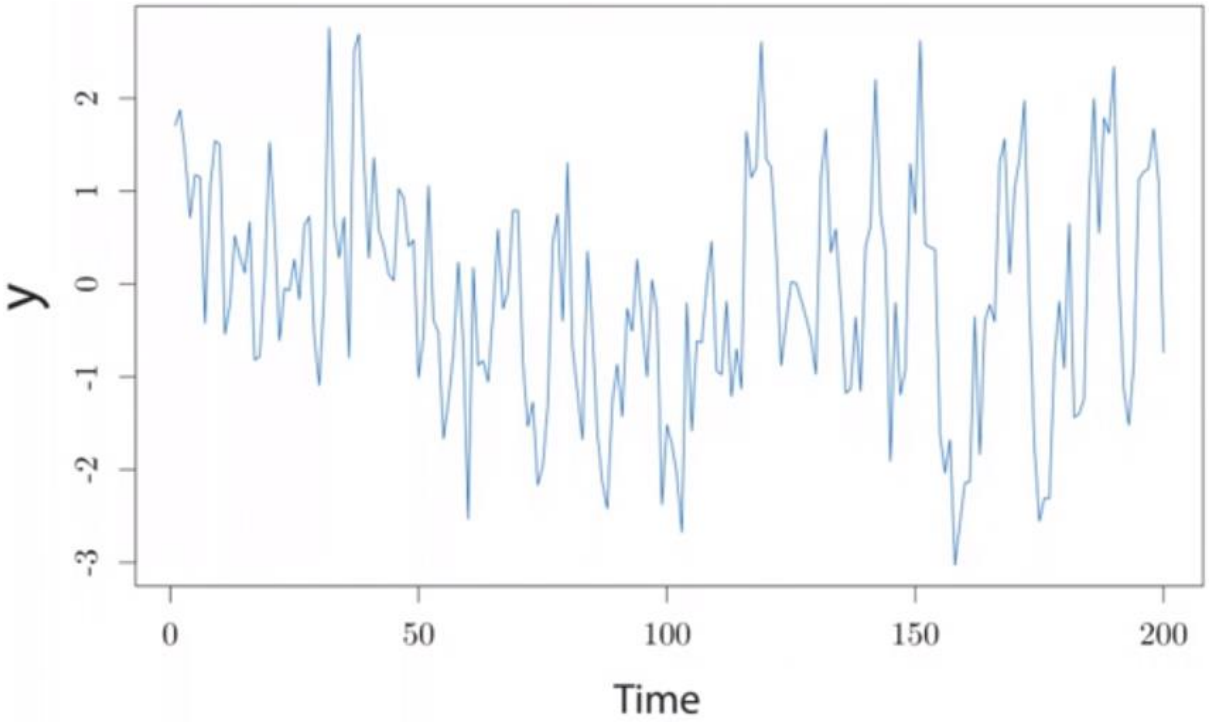

Рисунок 16. AR(1), $y_t = 0.7y_{t-1} + \varepsilon_t$

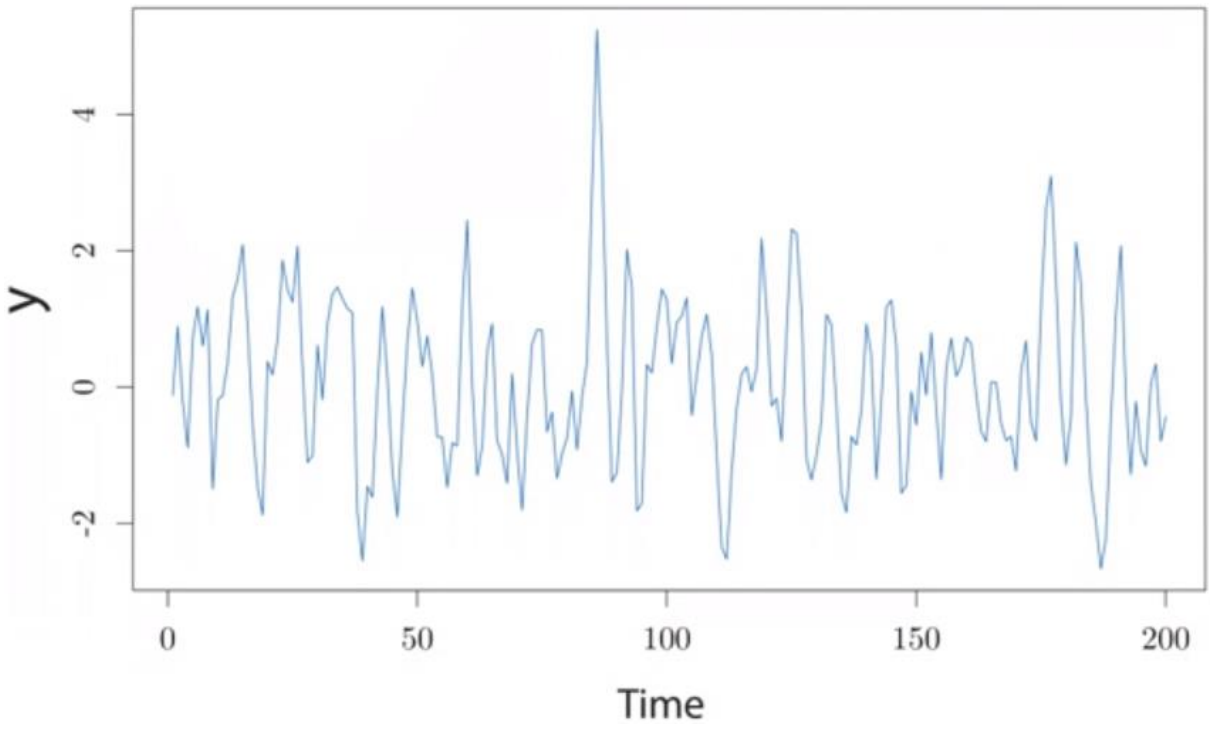

Рисунок 17. AR(2), $y_t = 0.9y_{t-1} - 0.5y_{t-2} + \varepsilon_t$

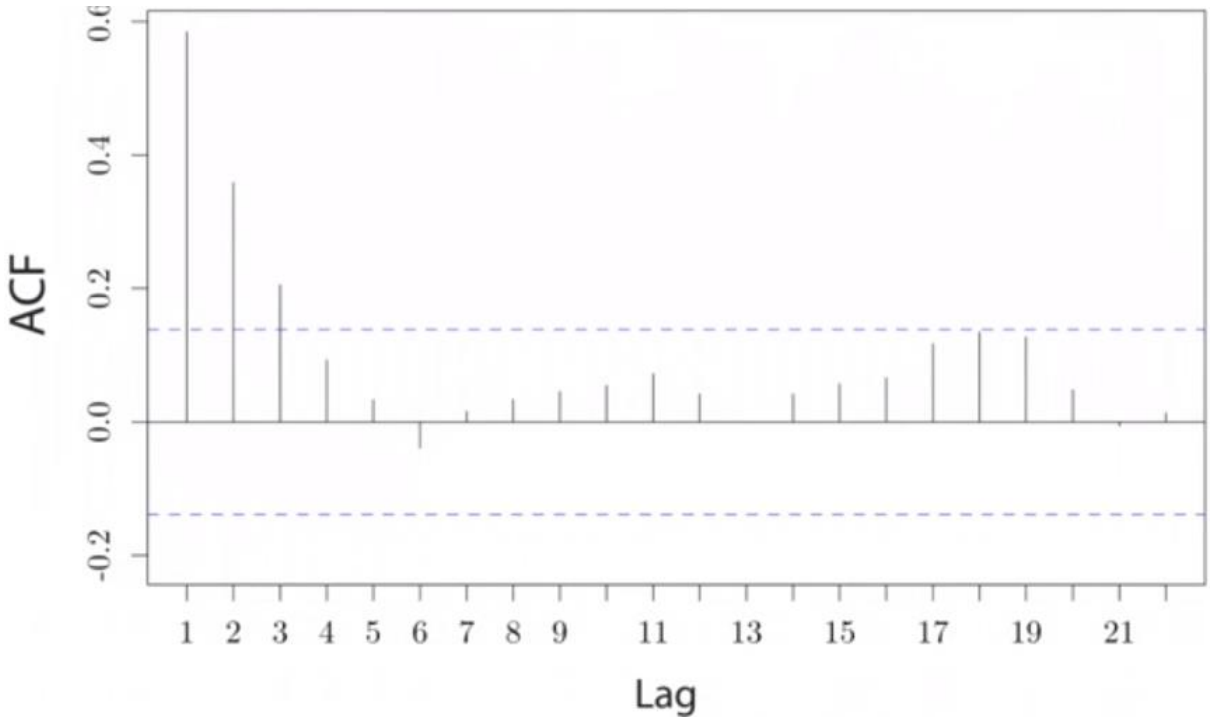

Рисунок 18. ACF, AR(1), $y_t = 0.7y_{t-1} + \varepsilon_t$

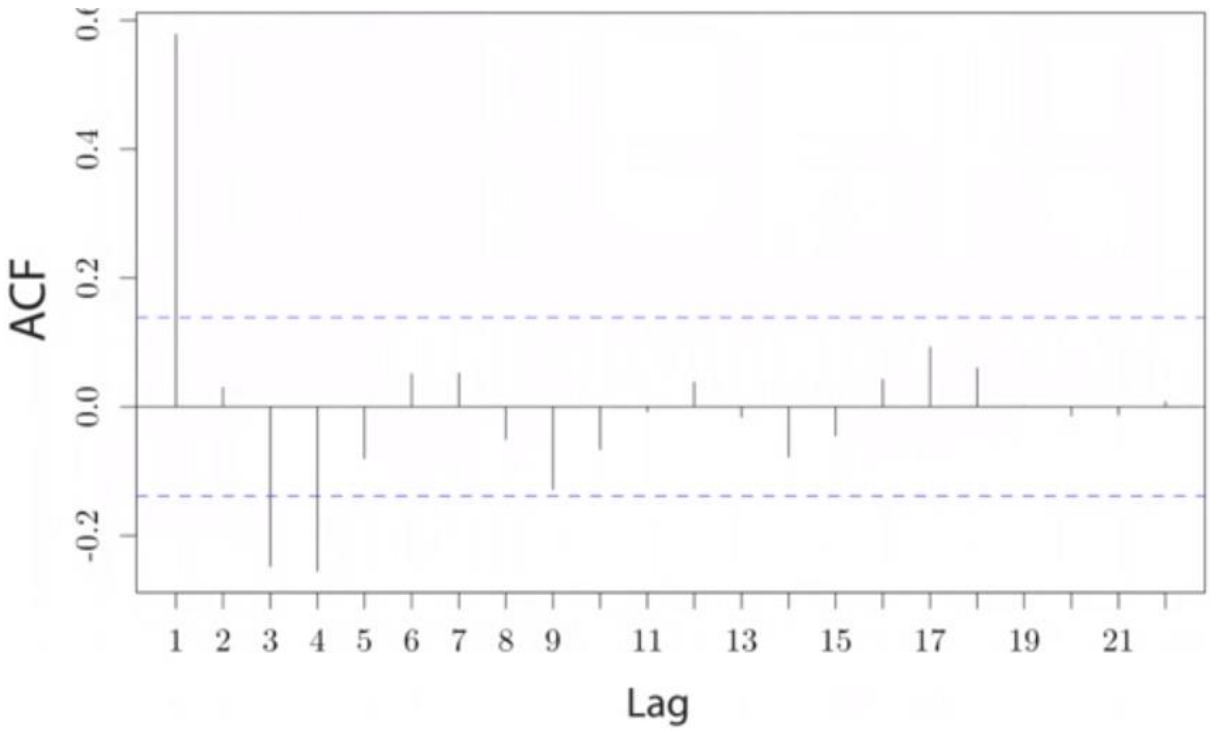

Рисунок 19. ACF, AR(2), $y_t = 0.9y_{t-1} - 0.5y_{t-2} + \varepsilon_t$

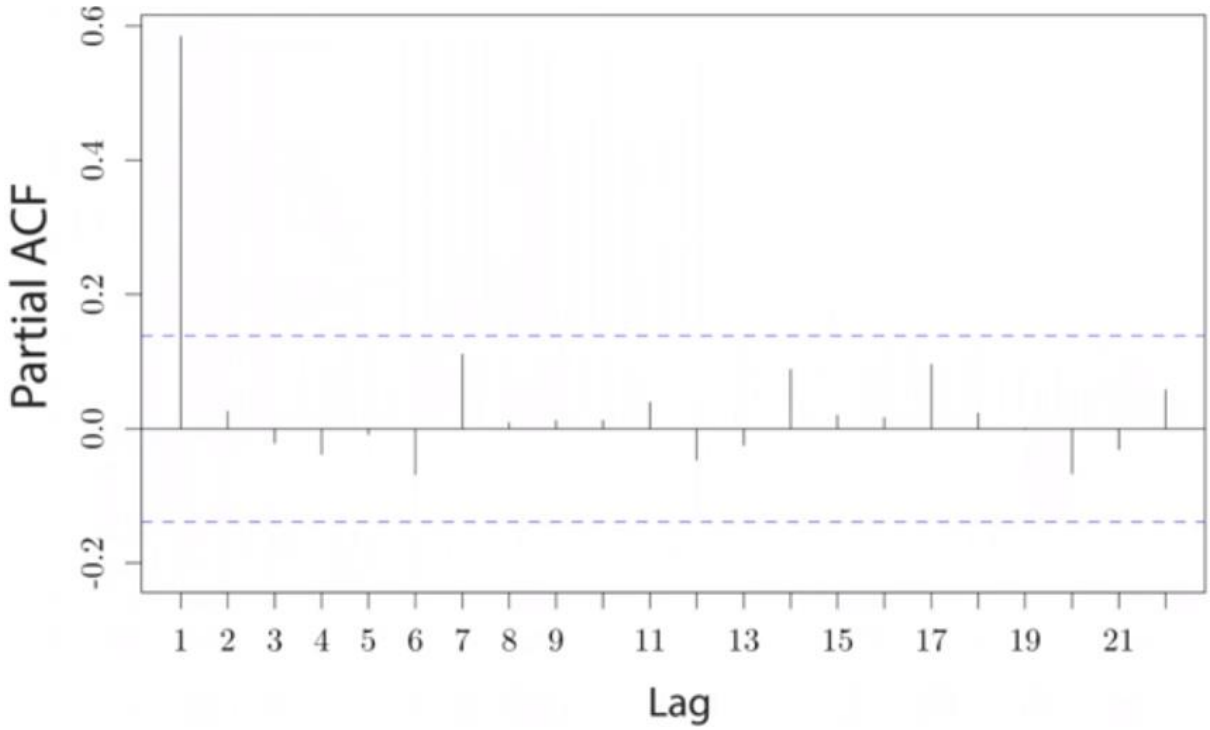

Рисунок 20. AR(1), $y_t = 0.7y_{t-1} + \varepsilon_t$

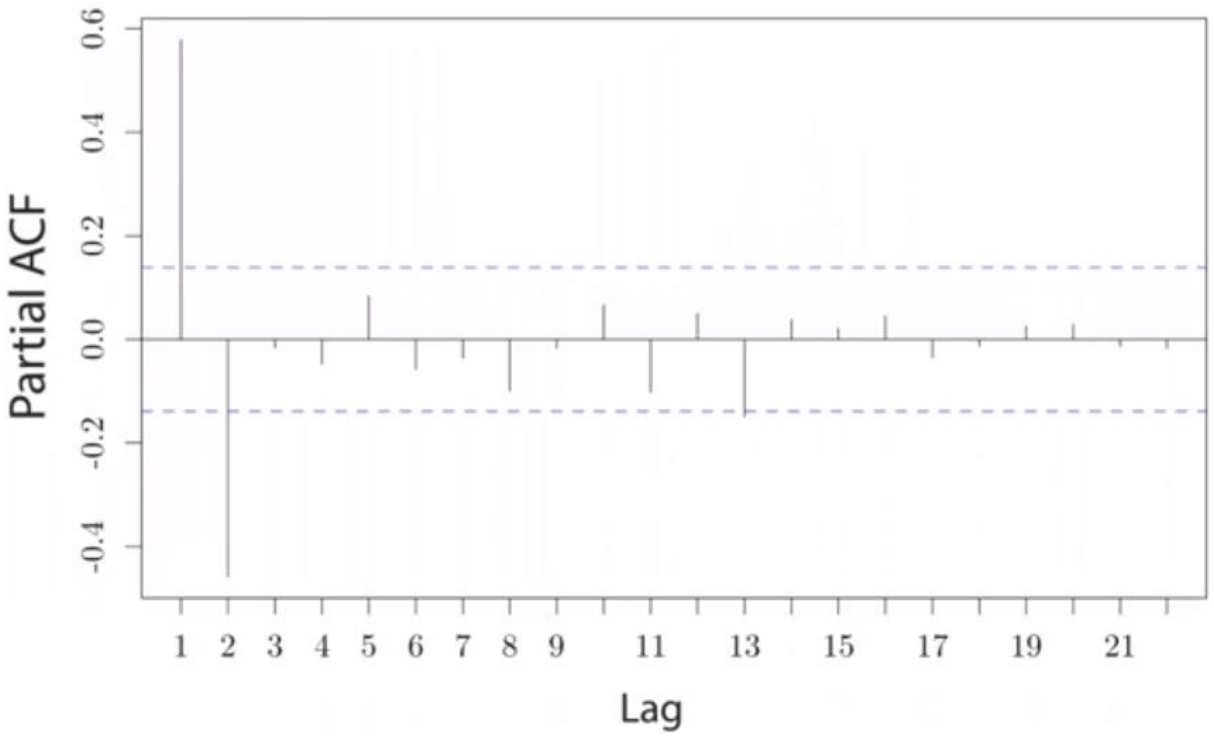

Рисунок 21. AR(2), $y_t = 0.9y_{t-1} - 0.5y_{t-2} + \varepsilon_t$

Аналогичные графики для процессов скользящего среднего MA(1) (рис. 22) и MA(2) (рис. 23) выглядят в некотором смысле зеркально графикам для AR процесса, а именно, автокорреляционная функция первые 1(рис. 24) или 2(рис. 25) значения не равны нулю, а соответственно частная автокорреляционная функция убывает довольно быстро к нулю(рис. 26, рис. 27). Соответственно, по этим признакам возможно по графикам выбрать параметры p и q, определить стационарность процесса.

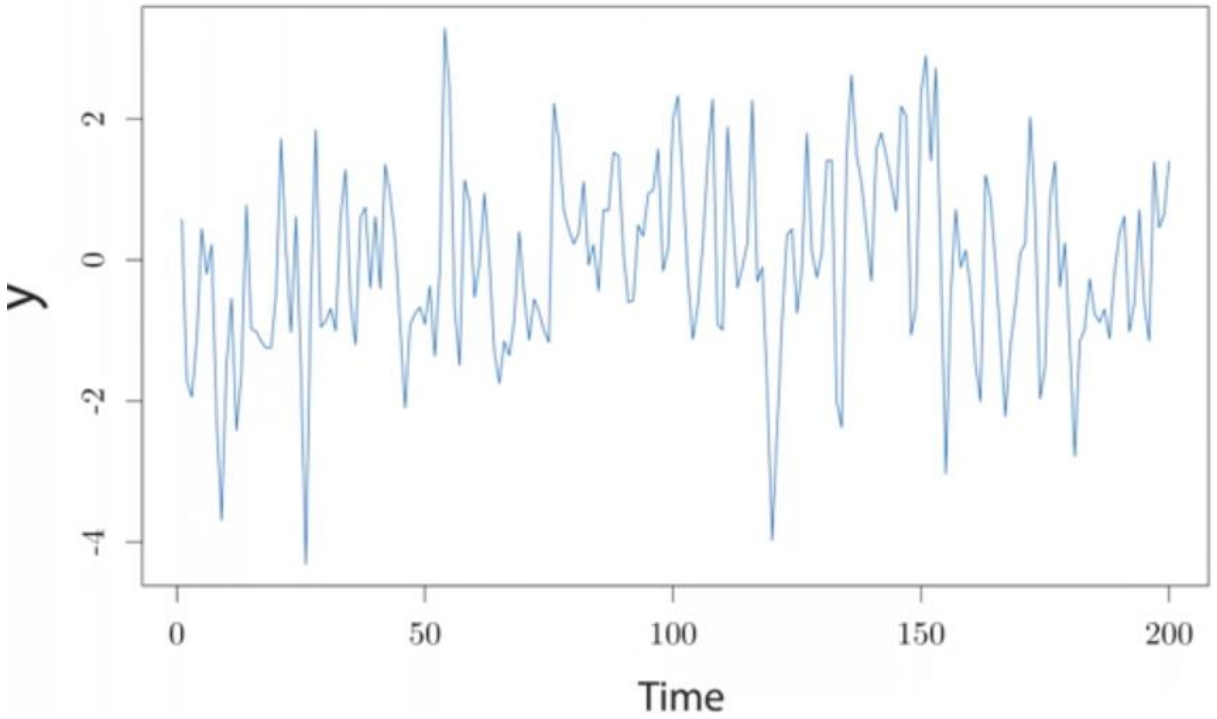

Рисунок 22. MA(1), $y_t = 0.7y_{t-1} + \varepsilon_t$

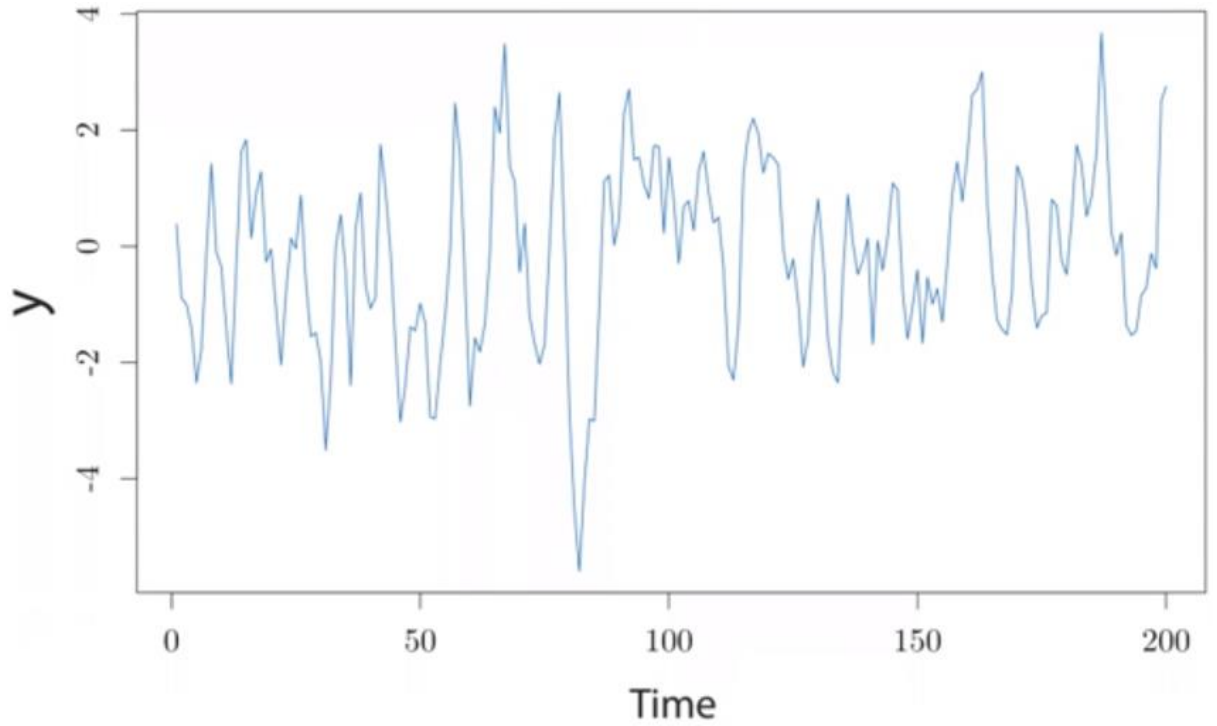

Рисунок 23. MA(2), $y_t = 0.9y_{t-1} + 0.5y_{t-2} + \varepsilon_t$

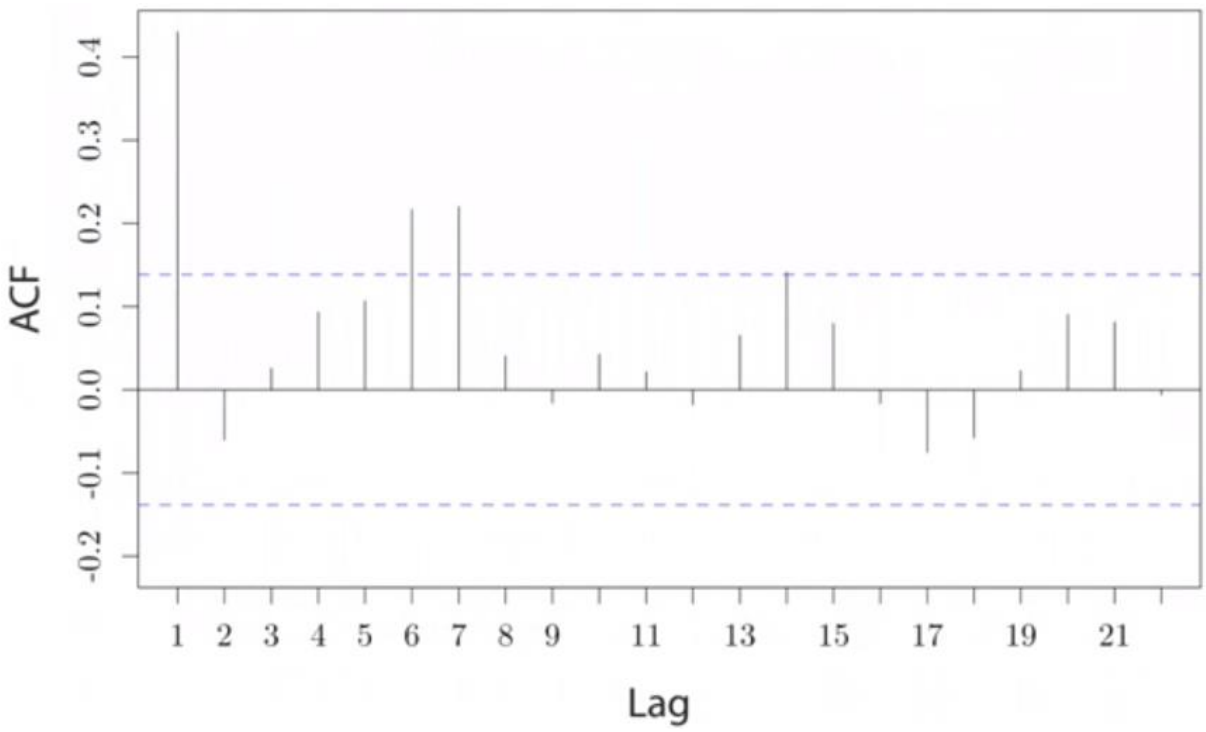

Рисунок 24. AFC, MA(1), $y_t = 0.7y_{t-1} + \varepsilon_t$

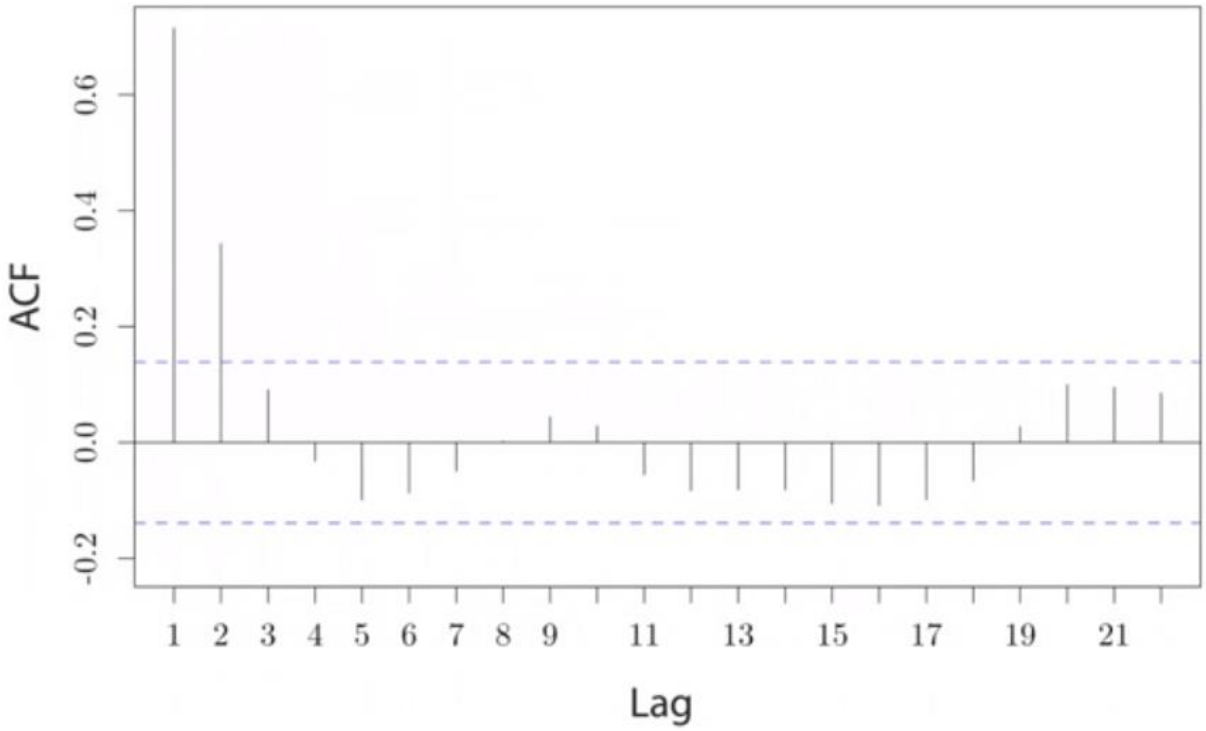

Рисунок 25. AFC, MA(2), $y_t = 0.9y_{t-1} + 0.5y_{t-2} + \varepsilon_t$

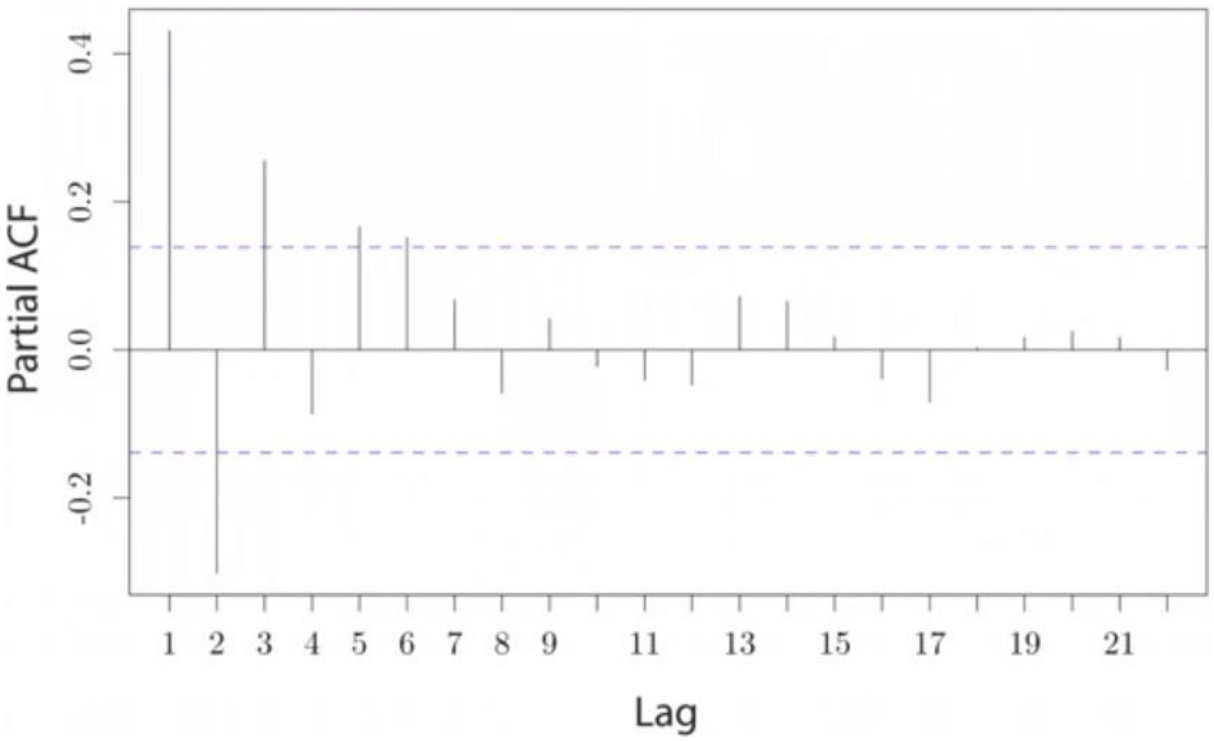

Рисунок 26. PAFC, MA(1), $y_t = 0.7y_{t-1} + \varepsilon_t$

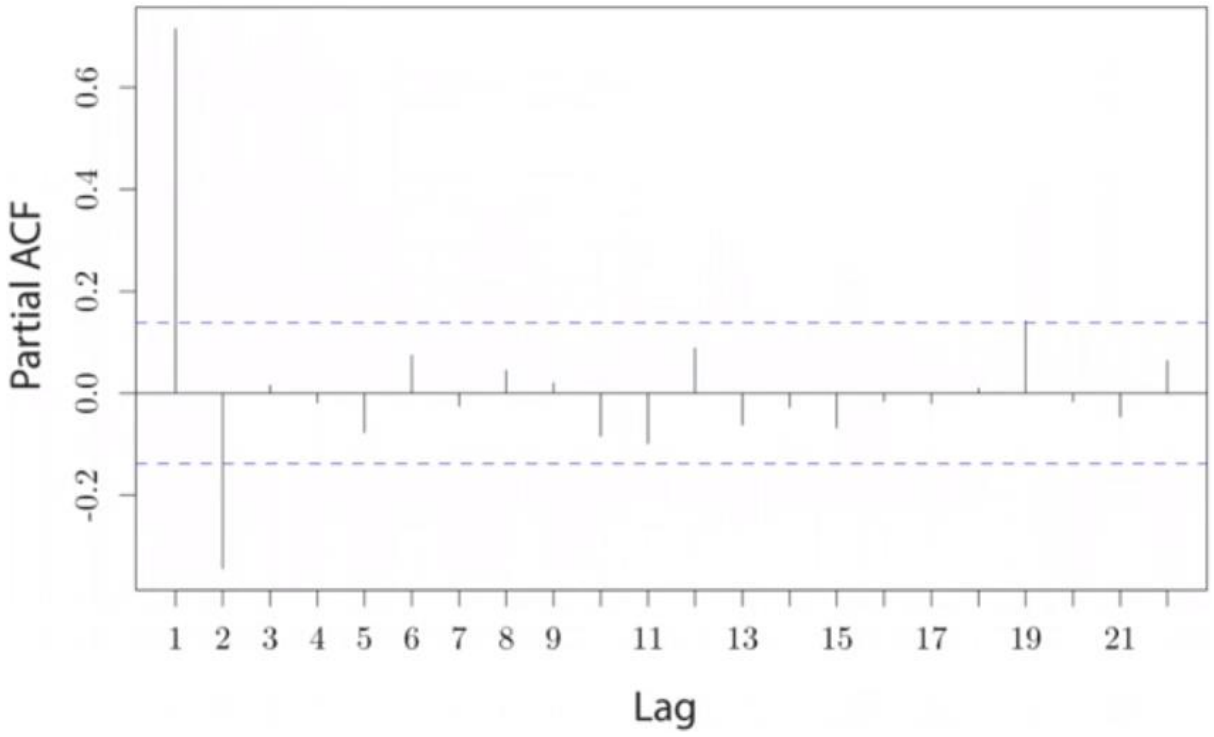

Рисунок 27. PAFC, MA(2), $y_t = 0.9y_{t-1} + 0.5y_{t-2} + \varepsilon_t$

Аналогичный график для ARMA(1,1) процесса (рис. 28). Сам процесс практически не отличим по графику ни от AR, ни от MA, ни от белого шума, однако, на графиках частной автокорреляционной функции(рис. 30) и автокорреляционной функции(рис. 29) видно довольно быстрое убывание коэффициентов к нулю.

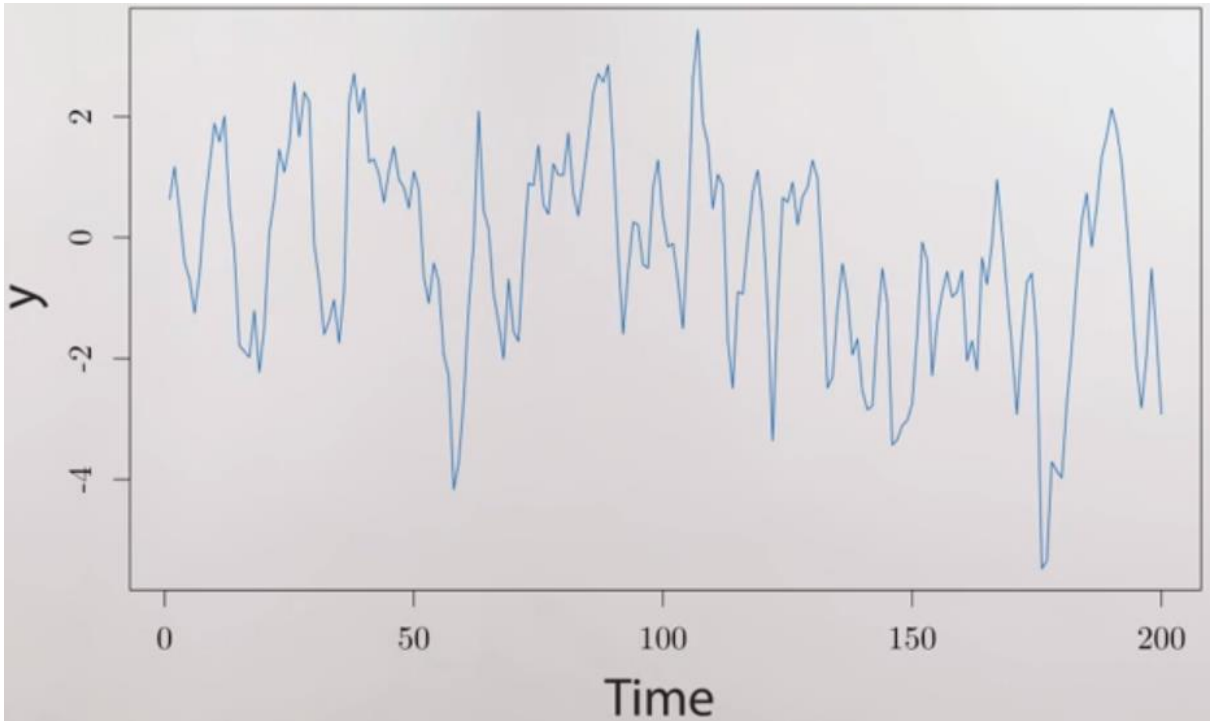

Рисунок 28. ARMA(1,1), $y_t = 0.7y_{t-1} + 0.5\varepsilon_{t-1} + \varepsilon_t$

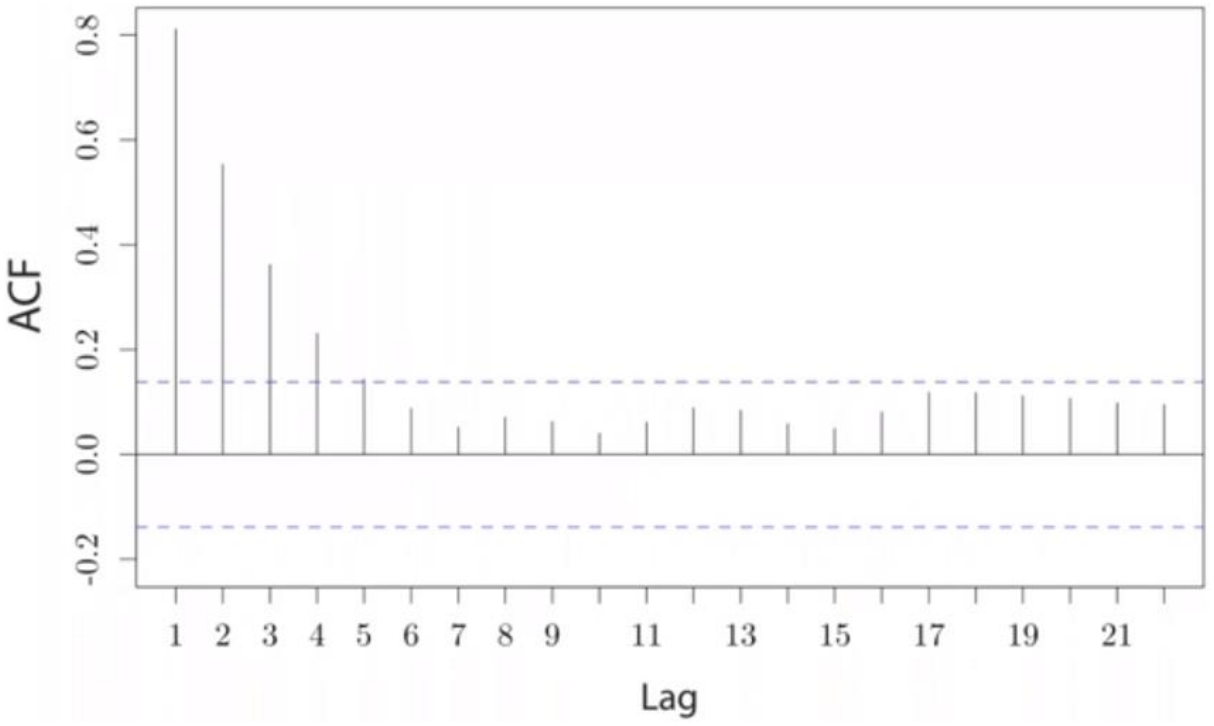

Рисунок 29. ACF, ARMA(1,1), $y_t = 0.7y_{t-1} + 0.5\varepsilon_{t-1} + \varepsilon_t$

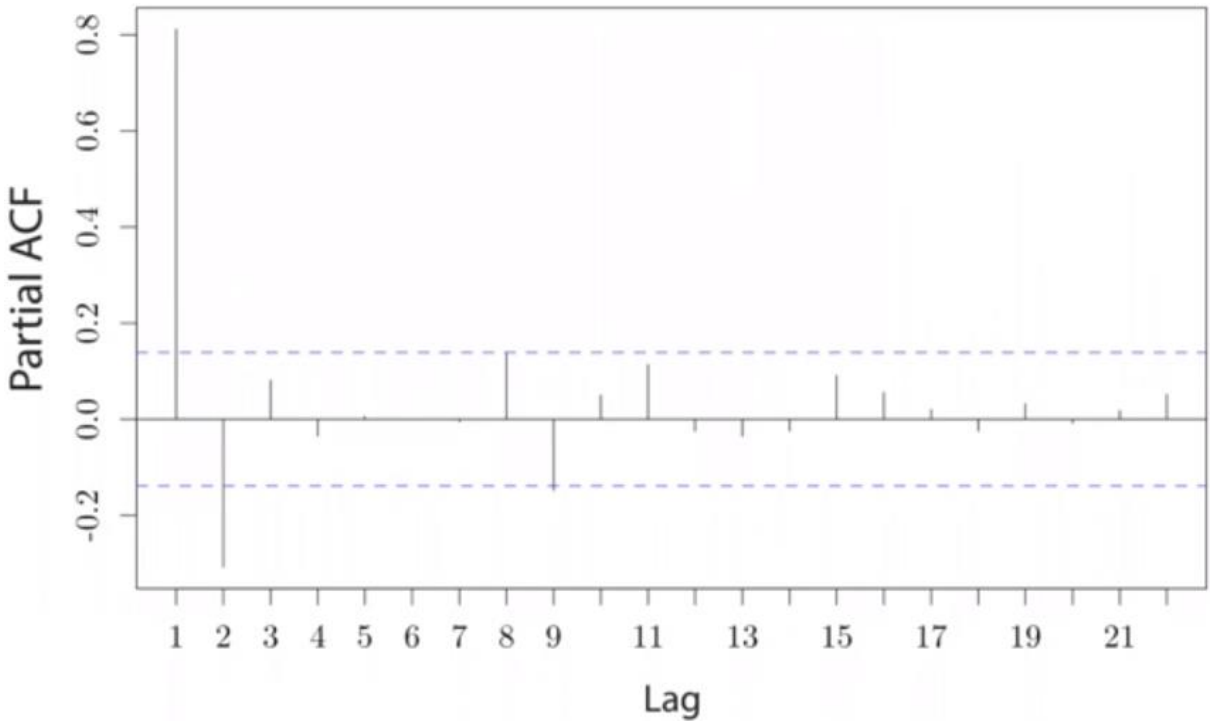

Рисунок 30. PACF, ARMA(1,1), $y_t = 0.7y_{t-1} + 0.5\varepsilon_{t-1} + \varepsilon_t$

ГЛАВА 2 ПОСТАНОВКА ЗАДАЧИ

2.1 Описательная постановка задачи

Необходимо провести анализ биржевых данных и на их основе построить модель, способную прогнозировать дальнейшее движение цен.

Данный алгоритм будет направлен на создание возможности прогнозирования цен рынка в режиме реального времени, что позволит трейдерам проводить более качественный технический анализ.

Технический анализ — это метод прогнозирования движения цен на рынке, при помощи использования специальных инструментов прогнозирования, основывающихся на закономерностях изменений цен исторических в аналогичных ситуациях [36]. Основополагающим же фактором является анализ графиков цен — «чартов» (chart — график, диаграмма), а также биржевого стакана. В теории, технической анализ возможно применить на любом рынке, однако наиболее используемым технический анализ стал на высоколиквидных свободных финансовых рынках, таких как биржи.

Так как предполагается прежде всего работа с техническим анализом, то было принято решение составлять прогноз не только основываясь на показаниях цен и объёмов, но и учитывать ряд индикаторов, описанных в приложении В.

2.2 Формальная постановка задачи

Данная задача предполагает построение различных классификационных моделей и изучение результатов, представляемых ими.

Предполагается использовать такие модели, как RandomForestClassifier, LogisticRegression, Kneighbors, SVC, MLPClassifier и Sequential.

2.3 Декомпозиция задачи

Все вышеизложенные задачи будут разрабатываться по следующему примерному плану:

1) Изучение предметной области.

2) Выбор наиболее приемлемых данных для прогнозирования.

3) Выбор наиболее приемлемых и широко используемых индикаторов.

4) Создания скрипта для сбора данных и показателей индикаторов, выгрузка этих данных в csv-файл.

5) Создание метода для чтения csv-файла и подготовки данных для дальнейшей работы с ними.

6) Создание “эталонных” значений прогнозной модели, которые будут являться показателями качества работы классификационных моделей, в случае преодоления “эталонного порога” прогноза.

7) Написание программ-классификаторов данных и получение вероятностных значений.

8) Подбор наилучших параметров классификаторов.

9) Анализ полученных результатов.

2.4 Аналитический обзор существующих методом решения данной проблемы

В настоящий момент существует множество подобных методов решений, основанных либо на техническом анализе, где берётся значение нескольких индикаторов и выбирается среднее между ними, либо на фундаментальном анализе. В обоих случаях, как правило, даются рекомендации на покупку актива, либо на продажу. Фундаментальный анализ предполагает долгосрочную работу с активами, а технический краткосрочную.

Фундаментальный анализ (англ. Fundamental analysis) — это термин , которым обозначается метод прогнозирования финансовых рынков, в частности определённых активов и компаний, основываясь на финансовых и производственных показателях их деятельности, а также на новостных сводках [37].

Фундаментальный анализ обычно используется инвесторами для долгосрочной оценки стоимости компании или другого финансового

инструмента, которая отражает общее состояние дел и рентабельность деятельности. Если брать компанию, то, помимо новостных сводок, также анализу подвергаются следующие финансовые показатели компании: выручка, чистая прибыль, EBITDA (Earnings Before Interests, Taxes, Depreciation and Amortization), обязательства, чистая стоимость компании, её денежный поток, производственные показатели компании и величина выплачиваемых ею дивидендов.

Также существуют модели, которые строят прогнозы не дальнейшего движения тренда на рост или падение, а предсказывающие именно дальнейшую цену актива. Такие модели также, как правило строятся либо на техническом, либо на фундаментальном анализе.

Одними из самых популярных аналитических систем в настоящий момент, предоставляемых в свободном доступе, основанных на техническом и/или фундаментальном анализе, являются усреднённый технический анализ по ряду индикаторов от TradingView, ежедневный фундаментальный анализ и обзор от ITICapital и Инвестник от Tinkoff.

2.5 Функциональные свойства приложения

Приложение представляет собой набор скриптов для сбора данных из рынка и набор программных средств, использующих классификационные модели для прогнозирования дальнейшего движения тренда, либо цен актива.

ГЛАВА 3 ПРОЕКТИРОВАНИЕ

3.1 Основания для разработки технического задания

Предполагалось, что конечный продукт представит не единое приложение с развёрнутым функционалом, а будет прежде всего направлено на исследование биржевых данных, представляющих собой временные ряды и прогнозирование дальнейшее движение тренда на их основе.

3.2 Оценка и выбор перспективных направлений разработки

Решение использовать готовые библиотеки, написанные для Python, такие как Scikit-Learn и TensorFlow, было принято с предположением того, что они имеют широкий функционал, множество проработанных моделей и дополнительных возможностей, что заведомо позволит сократить расходы на время разработки и фиксирования ошибок с этими моделями, если бы их пришлось разрабатывать самостоятельно с нуля. Также, так как эти библиотеки в настоящий момент поддерживаются разработчиками, то не придётся тратить дополнительно время на дальнейшее их развитие, что также сократит временные расходы и позволит гораздо быстрее перейти к шагу анализа моделей и результативности прогнозов.

3.3 Обоснование выбора инструментальных средств

Язык для написания скриптов для сбора информации, был выбран MQL5, который является составной частью торговой платформы MetaTraider5, где можно получить показания индикаторов, цен выбранных активов и объёмов за необходимый период времени.

При прогнозировании использовался ряд классификационных моделей, представленных в Scikit-Learn, такие как SVC, KNeighbors, MLPClassifier, RandomForestClassifier, LogisticRegression и модель, входящую в состав Keras, предствляющего собой надстройку над TensorFlow, — Sequential.

Наибольший интерес при разработке моделей и анализе представляли MLPClassifier и Sequential, которые являются нейросетями и которые, как предполагается, дадут один из самых лучших результатов, при прогнозировании движения тренда.

MLPClassifier представляет собой модель, основанную на многослойном персептроне(MLP). Многослойный персептрон (MLP) - это класс искусственной нейронной сети с прямой связью (ANN). Термин MLP используется неоднозначно, иногда свободно для обозначения любой прямой связи ANN, иногда строго для обозначения сетей, состоящих из нескольких слоев персептронов (с пороговой активацией) [38][39].

Модель Sequential представляет собой модель библиотеки Keras, которая работает с входной картиной векторов и в которой возможно задать различные методы решения задачи, в том числе и при помощи рекуррентных нейросетей.

Рекуррентные нейронные сети (PHC, Recurrent neural network, RNN) — это такой вид нейронных сетей, где связи между элементами образуют направленную последовательность [40]. В связи с этим появляется возможность обрабатывать наборы событий, находящиеся во времени или последовательные пространственные цепочки. Большим плюсом рекуррентных нейронных сетей, в отличие от многослойных перцептронов, является то, что они могут использовать свою внутреннюю память для обработки последовательностей произвольной длины. Именно поэтому сети RNN применяются в таких задачах, где нечто целостное разбито на части, например: распознавание рукописного текста [41] или распознавание речи [42][43]. Множество различных архитектурных решений было предложено для такого типа сетей от самых простых до сложных. В последнее время наибольшее распространение получили сеть с долговременной и кратковременной памятью (LSTM) и управляемый рекуррентный блок (GRU).

ГЛАВА 4 РЕАЛИЗАЦИЯ

4.1 Реализация скрипта для MQL5

Процесс реализации скрипта состоял из следующих этапов:

1. Установка MetaTraider5.
2. Изучение основных функций языка MQL5.
3. Создание документа скрипта в редакторе MQL5.
4. Реализация скрипта MQL5 для записи данных в csv-файл(прил. С).

Для получения данных и записи дальнейшей их в файл использовалась встроенная функция Copy, например CopyClose(sym,tf,start,end,c), которая позволяет копировать цены закрытия для актива “sym”, по заданному таймфрейму “tf” от даты начала “start” до даты окончания “end”. Копируется всё в массив “c”.

Если нет специализированной функции копирования для каких-либо данных, то копирование производится при помощи функции CopyBuffer. Например: CopyBuffer(iMA(sym,tf,14,0,0,PRICE_CLOSE),0,start,end,ma), где первый аргумент — это необходимые данные для копирования. В текущем случае это индикатор Moving Average. Далее передаётся число, указывающее на поток, который будет копироваться. Т.к. Moving Average передаёт возвращает только одно значение, то и копировать необходимо лишь нулевой поток. После задаётся время старта и окончания копирования и переменная, куда записываются данные из потока.

Подключение к csv-файлу происходит при помощи функции FileOpen(FileName, FILE_WRITE|FILE_CSV, ','), где FileName — это имя файла, FILE_WRITE|FILE_CSV — метод работы с файлом, а ',' — разделитель.

Запись в csv-файл производилась при помощи функции FileWrite(file,arg1,arg2,...), где file — это указатель на поток, который записывается в файл, а далее следуют аргументы, которые необходимо записать.

4.2 Изучение данных

Прежде всего была намечена задача прогнозирования движения тренда и для её изначального упрощения был выбран метод прогнозирования либо восходящего, либо нисходящего движения. Спред не учитывался, чтобы можно было более качественный прогноз на первых шагах построить.

В работе использовались классификационные и кластеризационные методы, представленные в библиотеках Scikit-Learn и TensorFlow.

Работа производилась с данными без изменений, преобразованными при помощи PCA и стандартизованными.

При разработке использовался пакет программ Anaconda, Jupyter Lab, TensorFlow(рис. 31).

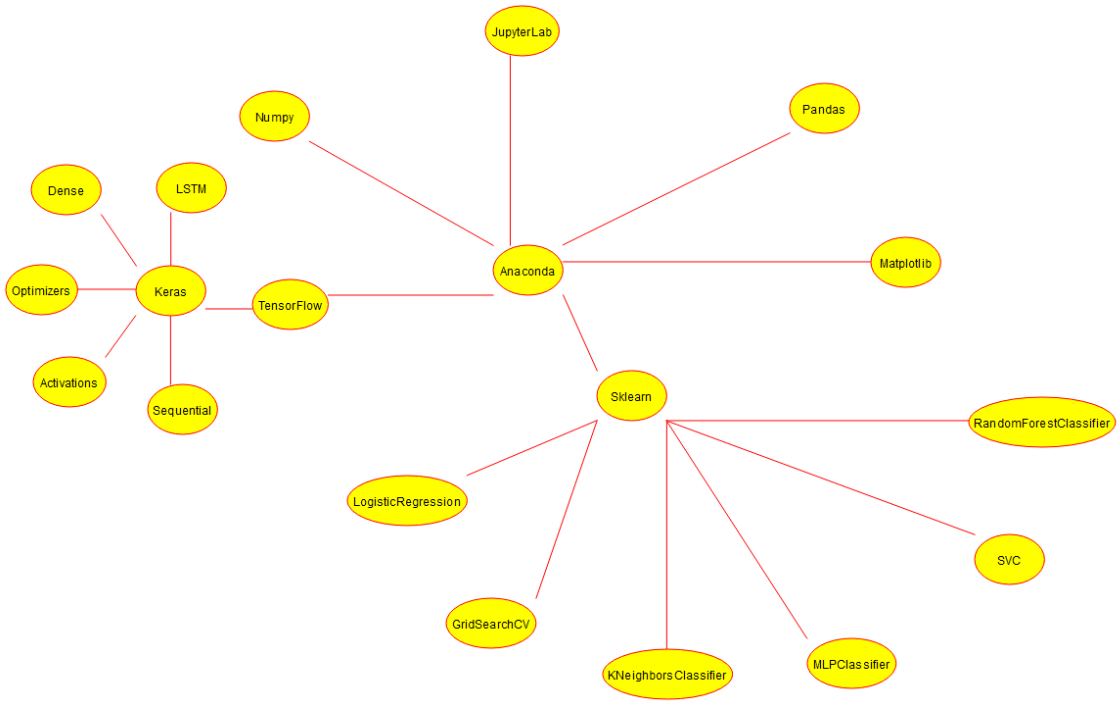

Рисунок 31. Пакет использованных программных средств

Прежде всего была построена диаграмма рассеяния данных начальных(рис. 32) и преобразованных через PCA(рис. 33), чтобы посмотреть, являются ли данные линейно разделимыми или нет.

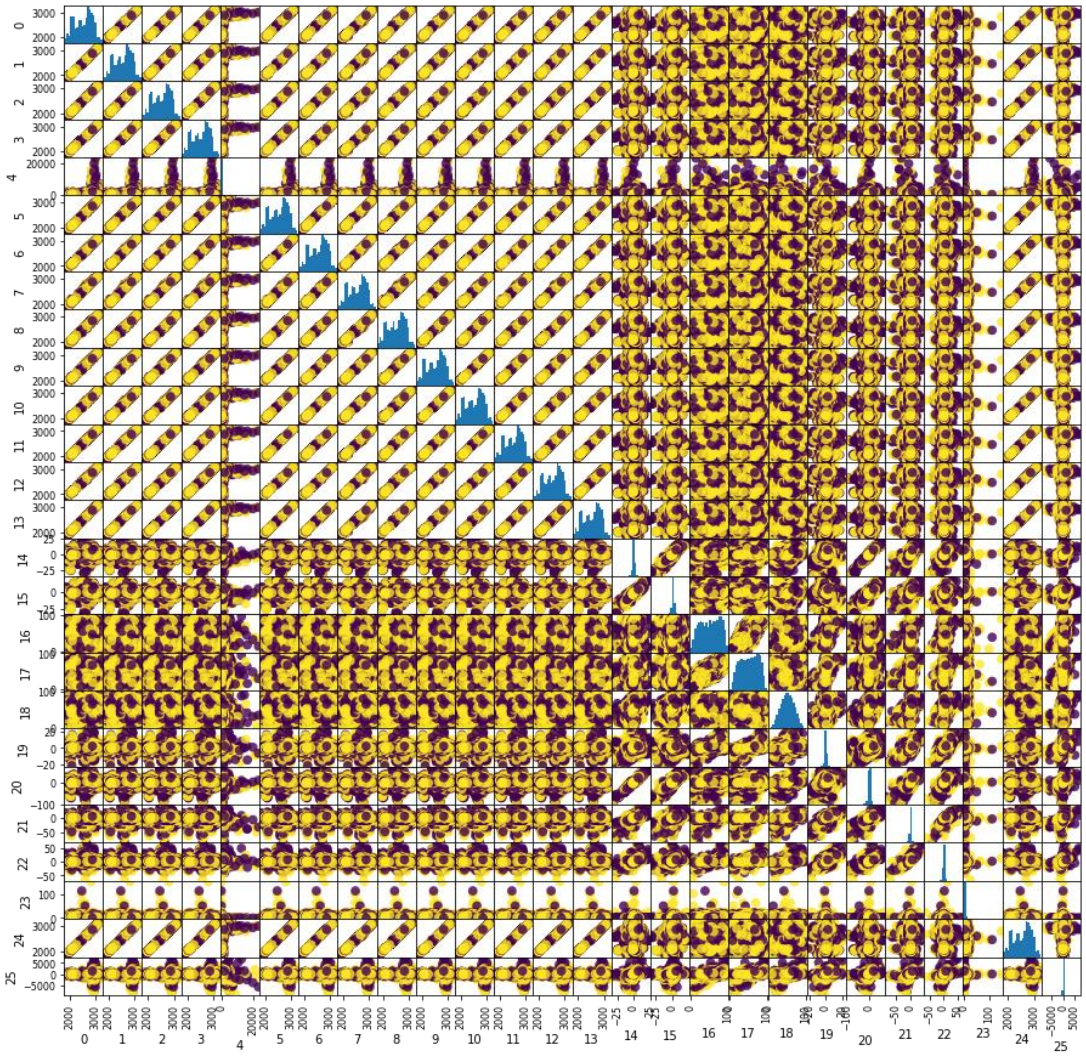

Рисунок 32. Диаграмма рассеяния данных S&P500 с индикаторами.

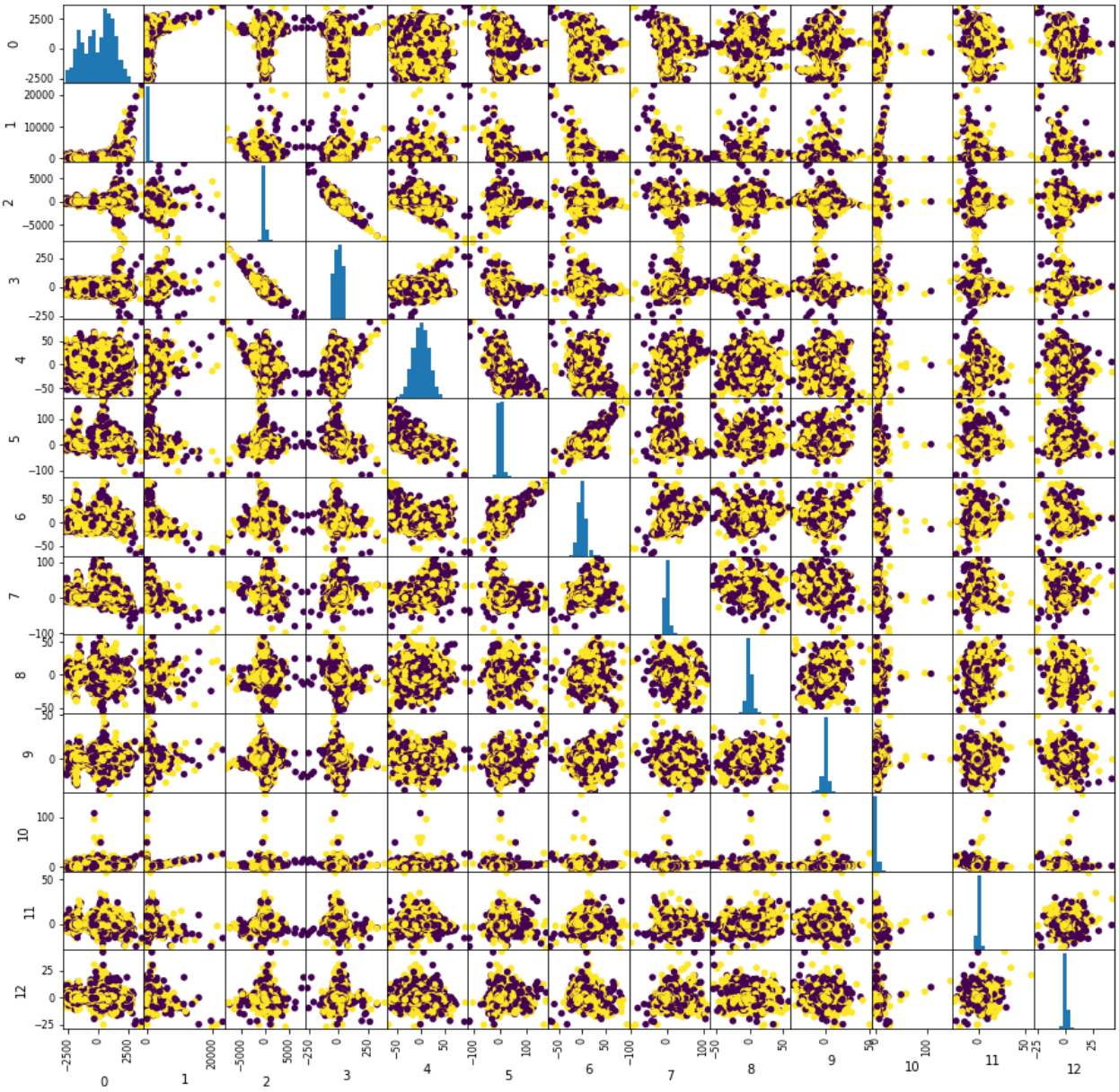

Рисунок 33. Диаграмма рассеяния данных S&P500 с индикаторами, преобразованная с помощью PCA.

После построения диаграмм рассеяния и их визуального анализа, стало ясно, что данные не являются линейно разделимыми, а, следовательно, классификаторы, которые работают, как линейные разделители, вероятнее всего, не дадут хорошую результативность, однако, их исключать не стоит, чтобы можно было сверить с ними работоспособность остальных классификаторов.

Также построим диаграмму авторегрессии и частотной авторегрессии(рис. 34), чтобы убедиться в том, что ряд является стационарным или нет.

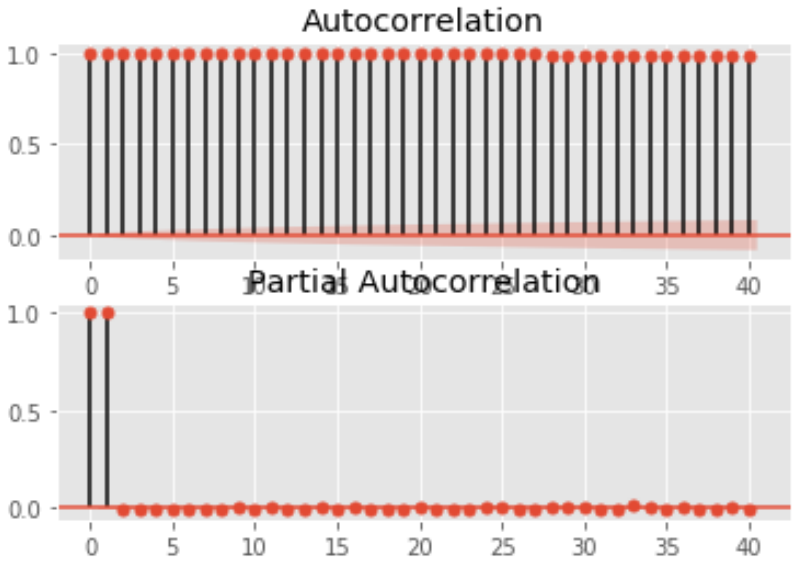

Рисунок 34. Диаграмма авторегрессии и частотной авторегрессии

Исходя из результатов данного графика(рис. 34), можно сказать, что данные представляют собой нестационарный процесс, являющийся трендом. Видно, что существует довольно сильная корреляция между соседними измерениями данных, однако, оценка резко падает и находится около нуля практически всё время.

Стационарность или постоянство — это такое свойство процесса, которое позволяет ему не менять свои характеристики с течением времени. Понятие используется в нескольких разделах науки [44].

Стационарный процесс — это стохастический процесс, у которого с течением времени не изменяется вероятностное распределение, т.е. математическое ожидание и дисперсия остаются постоянными. Потому как это свойство является основополагающим для многих статистических процедур, в том числе и при анализе временных рядов, то нестационарные процессы, как правило, преобразуются к стационарному виду. Чаще всего стационарность нарушается из-за тенденции к среднему значению ряда, которое, в свою очередь может обуславливаться либо дерминированностью тренда, либо наличием единого корня. В первом случае детерминированного тренда процесс называется стационарным процессом тренда, а стохастические шоки имеют только временные эффекты, после которых переменная стремится к детерминистически развивающемуся (непостоянному) среднему значению. В крайнем случае

единичного корня стохастические удары имеют постоянные эффекты, и процесс не является средним возвратом. Тенденционный стационарный процесс не является строго стационарным, но может легко трансформироваться в стационарный процесс, если устранить лежащий в его основе тренд, являющийся толко функцией времени. Также и другие процессы с одним и более единичиными корнями можно преобразовать к стационарному виду при помощи различий. Важным видом нестационарного процесса, который не включает трендоподобное поведение, является циклостационарный процесс, который является стохастическим процессом, который циклически изменяется со временем.

4.3 Вычисление “эталонных значений”

Для того, чтобы дальше приступить к работе с данными, необходимо вычислить значения, которые будут являться “эталонными” и с которыми будет производиться сравнение результатов, полученных при вычислении с помощью классификационных моделей.

Изначально предполагаются 2 набора данных. Первоначальный набор, который будет разделён на обучающий и тестовый, собранный за 2 года и контрольный тестовый набор данных, собранный за 2020 год, результирующее значение которого и будет являться самым важным.

Для качественного прогнозирования дальнейшего тренда было выдвинуты следующие предположения: (1) изменение тренда следующего бара будет равен предыдущему, (2) изменение тренда следующего бара будет равен среднему значению по 100 предыдущим.

Были получены следующие результаты, представленные в таблице 1.

Таблица 1. “Эталонные значения”

Data\Model	**previous=this**	**MA(100)**
Test	50,56%	55,37%
Train	51,28%	56,04%

New	50,93%	52,67%

Сравнение с предыдущим значением(листинг 1) и сравнение со средним значением(листинг 2) проводились в цикле.

Листинг 1. Сравнение данных с предыдущим значением

```
#счётчик верно спрогнозированных данных
count=0
#цикл проверки
for i in range(y_train.shape[0]-1):
    #сравнение предыдущего значения с текущим
    if(math.floor(y_train[i])==math.floor(y_train[i+1])):
        count+=1
#вывод результата
print("Train accuracy: {}".format(count/(y_train.shape[0]-1)*100))
```

Листинг 2. Сравнение данных с усреднённым значением предыдущих показателей

```
#счётчик верно спрогнозированных данных
count=0
#цикл проверки
for i in range(y_train.shape[0]-100):
    #сравнение предыдущего значения с текущим

if(math.floor(np.mean(y_train[i:i+99]))!=math.floor(y_train[i+100])
):
        count+=1
#вывод результата
print("Train accuracy: {}".format(count/(y_train.shape[0]-
100)*100))
```

Также, было вычислено “эталонное значение” ошибки при прогнозировании цен временного ряда. Код вычисления представлен в Листинге 3. Размер “эталонной” ошибки получился равным 5,98%.

Листинг 3. Расчёт “эталонной” ошибки при прогнозировании цены у временного ряда.

```
#размер ошибки
deviation=0.0
s=0.0
#отступ к свече для сравнения(предыдущая в данном случае)
st=1
#подсчёт ошибки
for i in range(0,len(ser_g)-st):
    s=0.0
    for j in range(0,st):
        s+=ser_g[field1][i+j]
    deviation+=(fabs(ser_g[field1][i+st]-
s/(st))/ser_g[field1][i+st])
#вывод размера ошибки
print("\nMAPE: %.2f%%" % (deviation))
```

4.4 Реализация классификационных и кластеризационных методов

При построении прогнозов использовались следующие модели: KNeighborsClassifier, SVC, LogisticRegression, RandomForestClassifier, MLPClassifier, Sequential.

Был выбран набор из 44000 примеров с 31 параметром, который взят в пределах от 2018 по 2019 год. Для новой выборки был собран набор данных из 8000 примеров за 2020 год с 31 параметром также.

Были получены следующие наилучшие результаты для разных типов данных спрогнозированные разными моделями, представленные в таблице 2.

Таблица 2. Наилучшие прогнозные результаты

Data/ Model	**Kneighbors**	**SVC**	**LogisticRegression**	**RandomForestClassifier**	**MLPClassifier**
Train	56,83	56,53	56,64	82,27	55,42

Test	56,58	56,85	56,76	51,10	56,01
New	52,09	52,65	52,09	—	53,00

Результаты показали, что одним из наиболее перспективных моделей классификации является MLP-классификация. Модель SVC также показала неплохие результаты, однако, время обучения этой модели оказалось на порядок выше, чем обучение MLP-классификатора, поэтому, было принято решение отказаться от её дальнейшего изучения.

Классификатор случайных лесов показал достаточно неплохие результаты на этапе обучения, однако, на этапе тестирования результативность оказалась близкой к случайности. Это показывает, что произошло переобучение модели, при чём переобучение наступает очень быстро с дальнейшим падением результативности при тестировании.

KNeighbors при обучении показало сходимость обучающего и тестового результата с увеличением числа соседей, однако, после того, как число соседей стало более 200, было принято решение также отказаться от данной модели в связи с увеличением требуемого времени для обучения данной модели.

Результативность логистической регрессии практически не изменялась в ходе изменения параметров.

Для удобного подбора параметров использовался метод, представленный в Scikit-Learn, GridSearchCV. Пример использования представлен в листинге 4.

Листинг 4. Применение метода GridSearchCV.

```
#подгрузка метода GridSearchCV
from sklearn.model_selection import GridSearchCV
#задание перебираемых параметров
parameters={'activation':['identity','logistic','tanh','relu'],
            'solver':['lbfgs','sgd','adam']
           }
#объявление модели, для которой будут перебираться установленные параметры
clf=GridSearchCV(MLPClassifier(),parameters, n_jobs=-1)
```

```
#передача данных для обучения модели
clf.fit(X_train, y_train)
#вывод наилучших подобранных параметров
print(clf.best_params_)
#Прогон данных обучающих, тестовых и новых на обученной модели
print(clf.score(X_train,
y_train),clf.score(X_test,y_test),clf.score(X_new,Y_new))
```

Кроме того, после подбора параметров также остались осцилляции, при изменении числа итераций в MLP-классификаторе, но, т.к. все осцилляции постоянно находились в заданном диапазоне(рис. 35) и в основном были равно одному и тому же значению, можно утверждать, что сеть обучена верно.

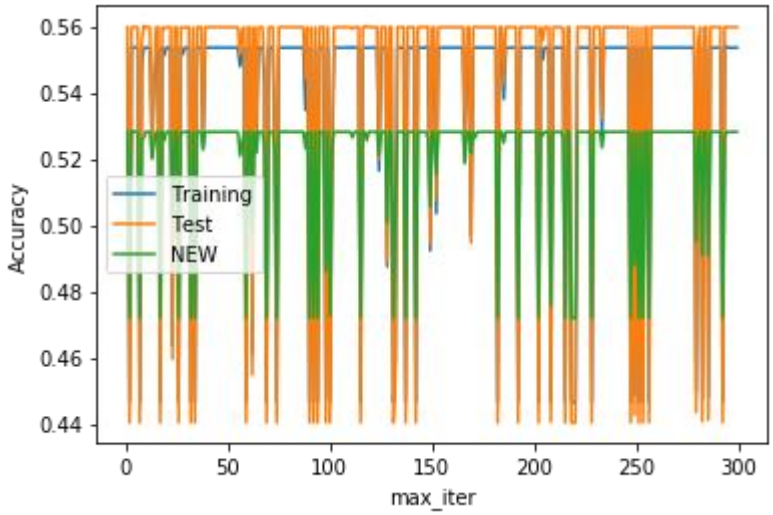

Рисунок 35. Осцилляции при изменении параметра в MLP-классификаторе

Все вышеописанные методы классификации и кластеризации работали с данными, представленными дискретным видом, хоть и находящихся в хронологическом порядке. Возможность работы с данными, как с временным рядом, обеспечивает библиотека TensorFlow, в частности его надстройка Keras, которая содержит модель Sequential и позволяет работать с временными рядами, используя рекуррентные нейронные сети.

Прогнозирование происходило в нескольких форматах.

Сначала производилось прогнозирование будущих цен актива, но прежде цены были преобразованы. Прогноз строился по ценам, полученным через

обработку цен по десятичному логарифму(рис. 36), по отношению предыдущей цены к текущей(рис. 37) и по натуральному логарифму отношения предыдущей цены к текущей(рис. 38).

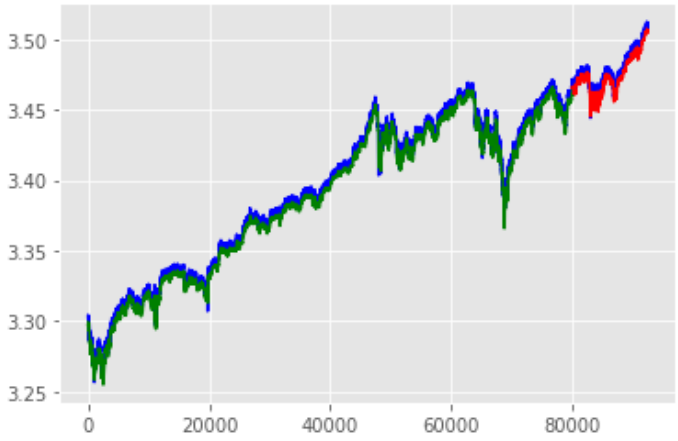

Рисунок 36. Прогноз log10(S&P500)

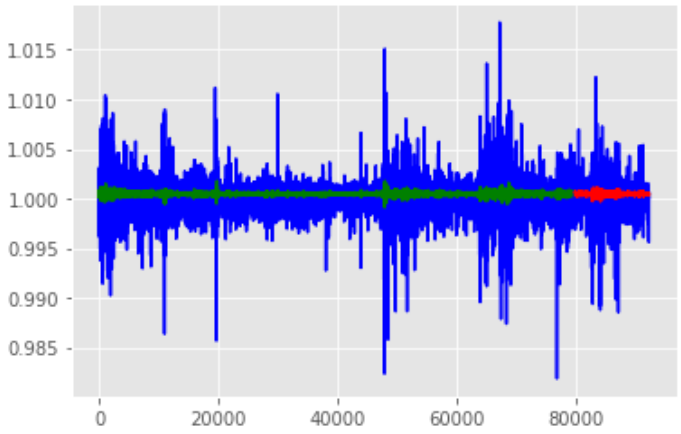

Рисунок 37. Прогноз S&P(i)/S&P(i-1)

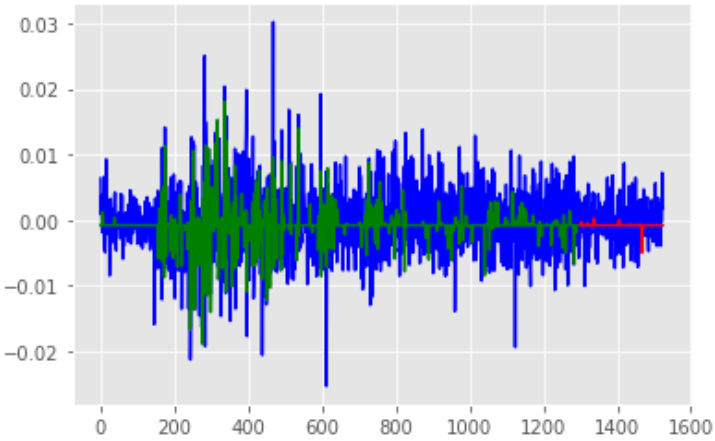

Рисунок 38. Прогноз log(S&P(i)/S&P(i-1))

Напомню, что эталонное значение ошибки, вычисленное для прогнозирования цен, равнялось 5,98%. В ходе эксперимента один из наилучших

результатов равнялся 0,17%(рис. 36), где модель достаточно корректно смогла спрогнозировать цену. Лучше результат ошибки показали цены, преобразованные по модели отношения текущей цены к предыдущей(рис. 37). Но по графику видно, что качество такой модели плохое, т.к. прогноз был построен около одной линии. Кроме того, было преобразование логарифма отношения(рис. 38), но ошибка оказалось чрезвычайно высокой, а это говорит о том, что прогноз не является корректным.

Пример объявления модели для прогнозирования цен временного ряда представлен в листинге 5.

Листинг 5. Объявление модели Sequential

```
#  Обучение нейронной сети

# Создание модели
model = Sequential()
# Добавление скрытого слоя в модель, состоящего из 10 нейронов и
функцией активации relu
model.add(Dense(8, input_dim=91, activation='relu'))
model.add(Dense(1, activation='linear'))
# Задание метрики, функции оптимизации и функции потерь
model.compile(loss='mean_squared_error', optimizer='adam',
metrics=['mean_absolute_percentage_error'])
# обучение модели
model.fit(X_train, y_train, epochs=300, batch_size=None)
```

Кроме того, была построена модель, на основе данных временного ряда, которая прогнозировала дальнейшее движения тренда, а не цену. В ней также были использованы рекуррентные нейронные сети. Пример объявления такой модели представлен в листинге 6.

Листинг 6. Рекуррентная нейронная сеть для временного ряда

```
#Объявление модели
single_step_model = tf.keras.models.Sequential()
#Добавление рекуррентной нейронной сети с 32 внутренними узлами
single_step_model.add(tf.keras.layers.LSTM(32,
```

```
input_shape=x_train_single.shape[-2:]))
single_step_model.add(tf.keras.layers.Dense(1))
#объявление функции оптимизации и функции потерь
single_step_model.compile(optimizer=tf.keras.optimizers.RMSprop(),
loss='mae')
#Обучение модели
single_step_history = single_step_model.fit(train_data_single,
epochs=EPOCHS,
steps_per_epoch=EVALUATION_INTERVAL,
validation_data=val_data_single,
validation_steps=50)
#получение значения ошибок
scores = single_step_model.evaluate(x_train_single, y_train_single)
#вывод значения ошибок
print("\nMAPE: %.2f%%" % (scores))
#построение прогноза движения тренда
scores = single_step_model.predict(x_train_single)
pl=0
mn=0
#расчёт качества прогноза
for i in range(0,scores.shape[0]):
    if(scores[i][0]>=0.5 and x_train_single[i]==1):
        pl+=1
    else:
        if(scores[i][0]<0.5 and x_train_single[i]==0):
            mn+=1
#вывод результатов прогноза
print("pluses:{}\nminuses:{}\nscore:{}".format(pl,mn,(pl+mn)/scores
.shape[0]))
```

В ходе работы алгоритма были получены результаты, которые можно увидеть в таблице 3.

Таблица 3. Результат предсказания тренда с использованием модели Sequential

Data\Model	Sequential
Train	55,73%
Test	55,25%
New	52,69%

По результатам видно, что прогнозная способность нейронной сети, работающей с временным рядом, сопоставима по результатам с прогнозной способностью нейронный сети, работающей с дискретным набором величин.

4.5 Дополнительные разработки

В ходе взаимодействия с трейдерами было выдвинуто предположение о том, что первый младший бар, входящий в состав старшего бара, будет определять направление закрытия старшего бара.

Были собраны и подготовлены данные. Предполагалось, что наибольшая корреляция наблюдается между часовым и суточным баром.

Бар — это элемент графика котировок, который отображает движение цены на определенном временном промежутке (таймфреме) [45]. Бар показывает такую информацию:

- цена открытия позиции;
- цена закрытия позиции;
- максимальная цена за период;
- минимальная цена за период.

Было принято решение собрать данные по биткоину(так как он торгуется круглосуточно и ранее трейдер на нём проверял данное предположение). После обучения нейросети и проведения тестирования, результаты, представленные в таблице 4, показались многообещающими.

Таблица 4. Результат предсказания движения тренда старшего бара по младшему

Data(BTC)/Model	MLPClassifier
Train	52,22%
Test	51,89%
New	55,56%

На наборе данных 2020 года результативность оказалась 55,56%, что на 2,5 процентных пункта выше, чем это было при прогнозировании с использованием индикаторов. Однако, результаты, построенные на обучающей и тестовой выборках оказались много ниже, чем на новом наборе данных, что является плохим показателем. После этого было принято решение выявить наименьшее, среднее и наибольшее значение прогноза модели, сохранив все параметры и изменяя только значение максимального числа итераций. Результаты представлены в таблице 5.

Таблица 5. Минимальное, среднее и максимальное значение прогноза модели при изменении числа итераций

Data\Type	Минимум	Среднее	Максимум
Train	43,67%	56,54%	61,08%
Test	41,51%	56,23%	66,04%
New	43,14%	57,70%	65,36%

Из данных таблицы 5 видно, что разброс между минимальным, средним и максимальным значением достаточно велик. Также, при изменении максимального числа итераций, на графике(рис. 39) видны значительные осцилляции, что также ставит под сомнение состоятельность данного предположения.

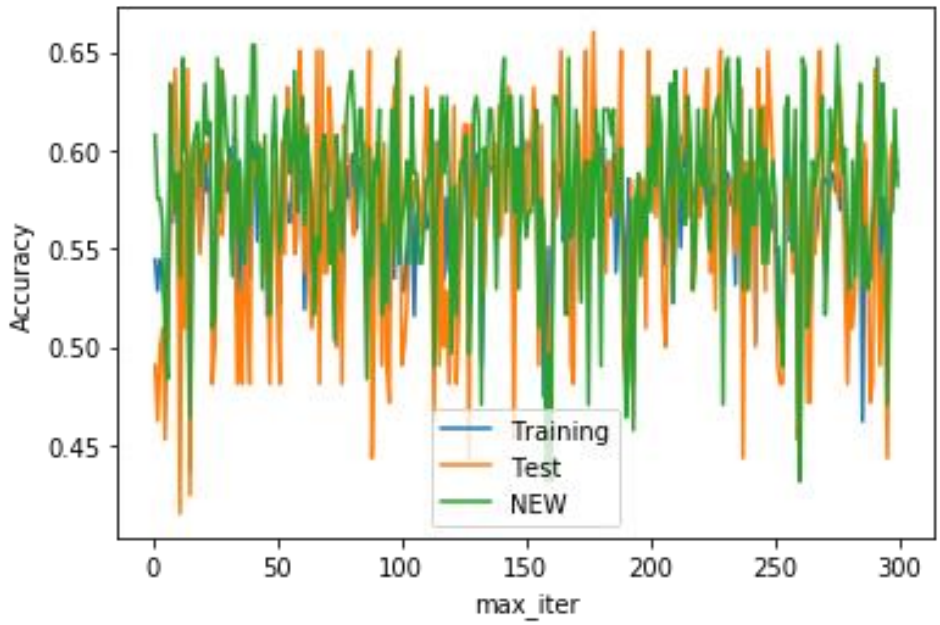

Рисунок 39. Зависимость точности предсказываемой модели от числа итераций

Также, после получения данных результатов эта стратегия была перепроверена непосредственно на рынке и было принято решение объявить её частично несостоятельной. Потребуется найти дополнительные признаки, которые будут также влиять на точность данной модели.

Кроме того, в ходе работы в данном направлении был создан индикатор для TradingView(рис. 40) на внутреннем языке программирования Pine4, который указывает возможный диапозон ценовой, в котором окажется в дальнейшем цена тренда и от которого можно торговать.

Цена строится от средневзвешенной цены тренда, значения одинарной, двойной или тройной экспоненциальной скользящей средней относительно текущей цены или с отступом. Все параметры возможно задать вручную(рис. 41). Описание параметров представлено в Приложении D.

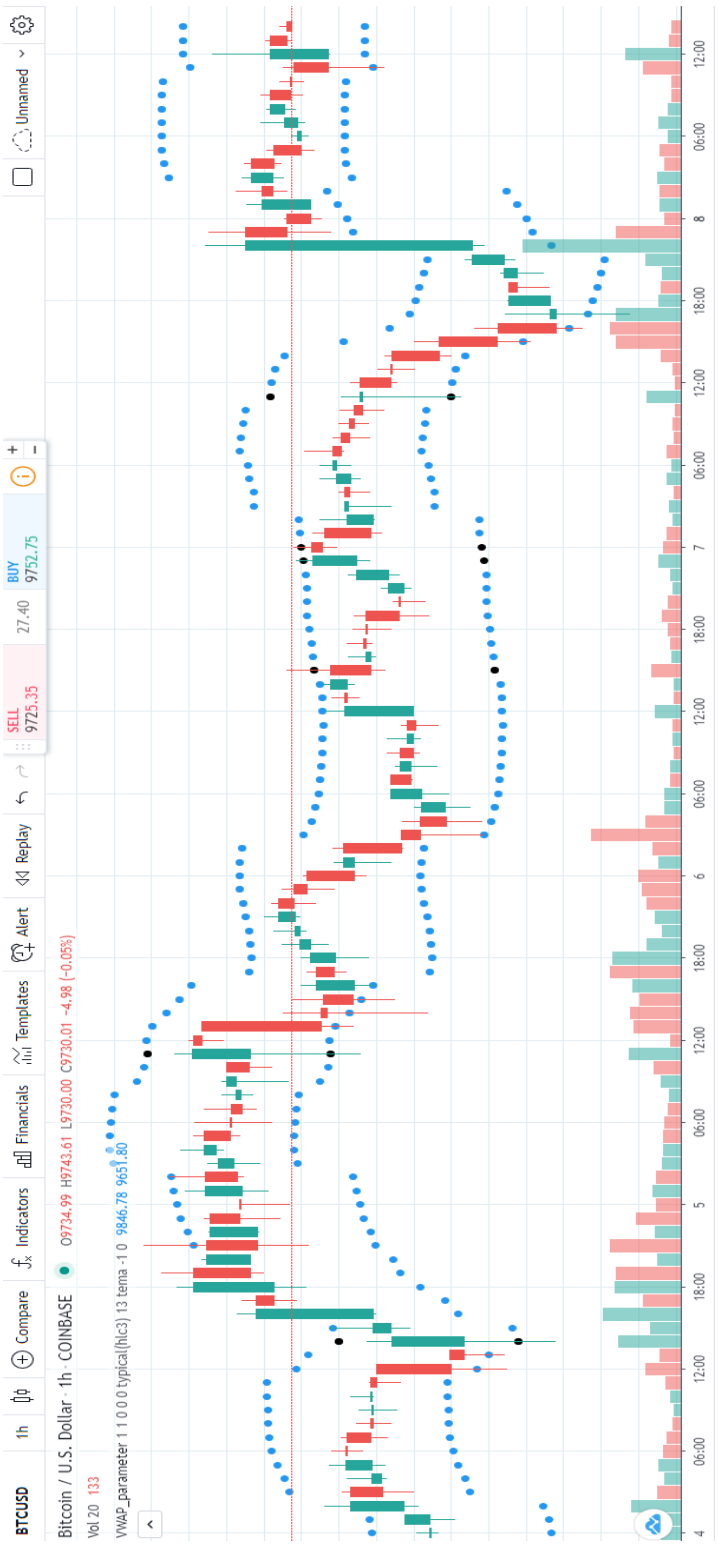

Рисунок 40. Индикатор для TradingView

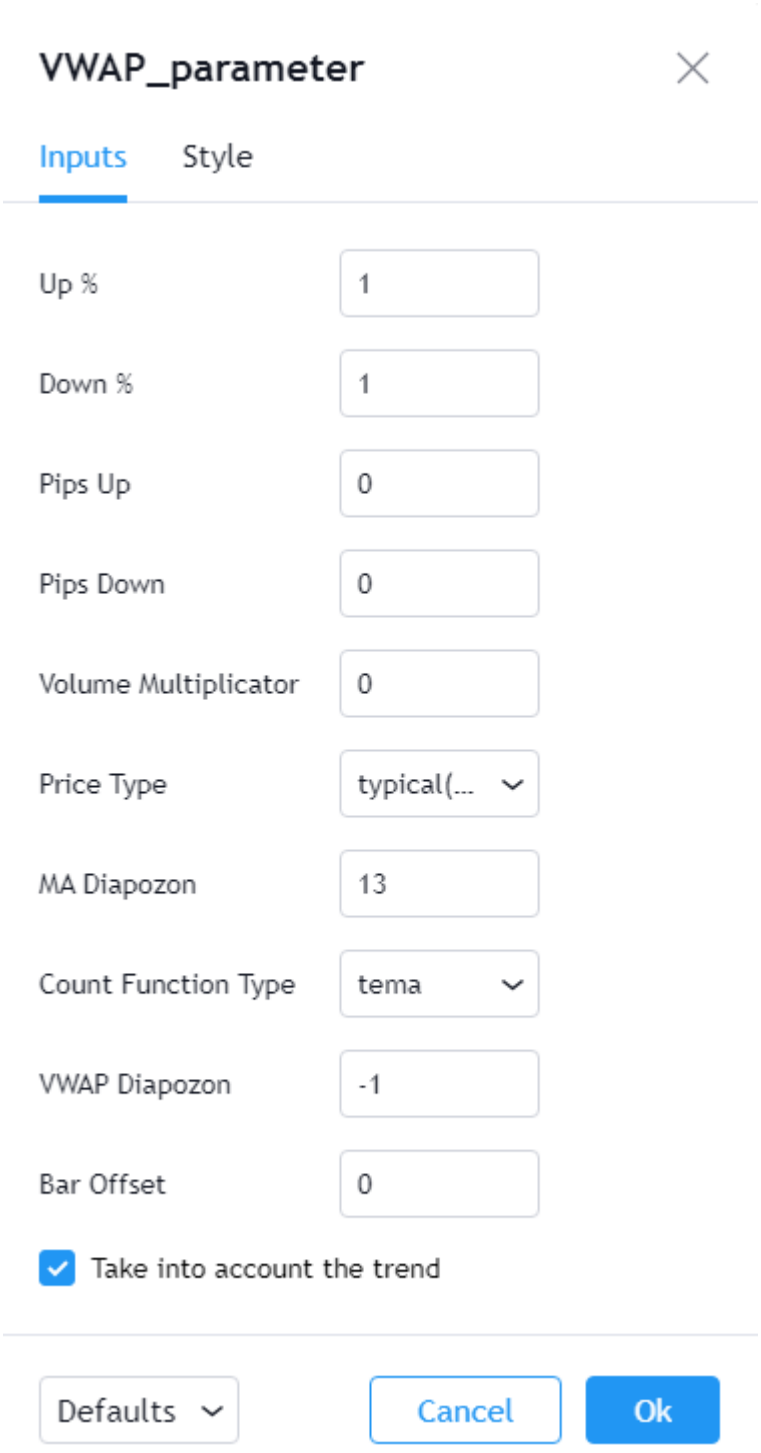

Рисунок 41. Параметры индикатора

ЗАКЛЮЧЕНИЕ

Исследование успешно проведено и были протестированы несколько предположений. Кроме того, в ходе работы также был разработан торговый индикатор, который был протестирован на бирже неоднократно.

Выполненное исследование позволило усовершенствовать навыки программирования на языке Python, изучить различные методы построения прогнозных значений, а также более точно понять суть как простых, так и рекуррентных нейронных сетей. Помимо этого были усовершенствованы навыки работы в программной среде JupyterLab, подробно изучены библиотеки, предназначенные для разработки систем машинного обучения, такие как Scikit-Learn и TensorFlow.

В ходе работе удалось выявить недостатки предположенных моделей, а также возможности проведения дальнейших исследования в выбранной предметной области.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Трейдер. URL: https://ru.wikipedia.org/wiki/Трейдер (Дата обращения: 05.05.2020г).
2. Марк Фридфертиг, Джордж Уэст — «Электронная внутридневная торговля».
3. Скальпинг. URL: https://ru.wikipedia.org/wiki/Скальпинг (Дата обращения: 05.05.2020г).
4. Data Mining. URL: https://ru.wikipedia.org/wiki/Data_mining (Дата обращения: 05.05.2020г).
5. Великие раскопки и великие вызовы. Интервью Григория Пятецкого-Шапиро, данное журналу «Компьютерра» в 2007 году(стр 48—51).
6. В. А. Дюк, А. В. Флегонтов, И. К. Фомина, Применение технологий интеллектуального анализа данных в естественнонаучных, технических и гуманитарных областях.
7. О. С. Коваленко, Обзор проблем и перспектив анализа данных.
8. А. А. Ежов, С. А. Шумский, Лекция: Извлечение знаний с помощью нейронных сетей.
9. Microsoft SQL Server 2008 R2: новый подход к управлению информацией.
10. Data Mining от Oracle: настоящее и будущее.
11. Степанов Р. Г. Технология Data Mining: Интеллектуальный Анализ Данных.
12. Кластерный анализ. URL: https://ru.wikipedia.org/wiki/Кластерный_анализ (Дата обращения: 05.05.2020г).
13. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: Классификация и снижение размерности. — М.: Финансы и статистика, 1989. — стр. 607.
14. Мандель И. Д. Кластерный анализ. — М.: Финансы и статистика, 1988. — стр. 176.

15. Хайдуков Д. С. Применение кластерного анализа в государственном управлении// Философия математики: актуальные проблемы. — М.: МАКС Пресс, 2009. — стр. 287.

16. Классификация и кластер. Под ред. Дж. Вэн Райзина. М.: Мир, 1980. стр. 390.

17. Tryon R.C. Cluster analysis. — London: Ann Arbor Edwards Bros, 1939. — стр. 139.

18. Обучение с учителем. URL: https://ru.wikipedia.org/wiki/Обучение_с_учителем (Дата обращения: 05.05.2020г).

19. Теорема сходимости перцептрона. URL: https://ru.wikipedia.org/wiki/Теорема_сходимости_перцептрона (Дата обращения: 05.05.2020г).

20. Валидация моделей. URL: https://help.loginom.ru/userguide/processors/validation.html (Дата обращения: 07.06.2020г).

21. Кросс-валидация. URL: https://wiki.loginom.ru/articles/cross-validation.html (Дата обращения: 07.06.2020г).

22. Перекрёстная проверка. URL: https://ru.wikipedia.org/wiki/Перекрёстная_проверка (Дата обращения: 07.06.2020г).

23. Дерево решений. URL: https://ru.wikipedia.org/wiki/Дерево_решений (Дата обращения: 07.06.2020г).

24. Quinlan, J. R., (1986). Induction of Decision Trees. Machine Learning 1: 81-106, Kluwer Academic Publishers.

25. Метод опорных векторов. URL: https://ru.wikipedia.org/wiki/Метод_опорных_векторов (Дата обращения: 07.06.2020г).

26. Многослойный персептрон Румельхарта. URL: https://ru.wikipedia.org/wiki/Многослойный_перцептрон_Румельхарта (Дата обращения: 07.06.2020г).

27. Yoshua Bengio, Aaron Courville, Pascal Vincent Representation Learning: A Review and New Perspectives, 2014.

28. Введение в Scikit-Learn. URL: https://neurohive.io/ru/osnovy-data-science/vvedenie-v-scikit-learn/ (Дата обращения: 07.06.2020г).

29. TensorFlow. URL: https://ru.wikipedia.org/wiki/TensorFlow (Дата обращения: 07.06.2020г).

30. «TensorFlow: Open source machine learning» «It is machine learning software being used for various kinds of perceptual and language understanding tasks» — Jeffrey Dean, отрезок 0:47—2:17.

31. Why TensorFlow. URL: https://www.tensorflow.org/about (Дата обращения: 07.06.2020г).

32. Google Just Open Sourced TensorFlow, Its Artificial Intelligence Engine. URL: https://www.wired.com/2015/11/google-open-sources-its-artificial-intelligence-engine/ (Дата обращения: 07.06.2020г).

33. Временной ряд. URL: https://ru.wikipedia.org/wiki/Временной_ряд (Дата обращения: 05.05.2020г).

34. Шмойлова Р. А. Общая теория статистики: Учебник. — М.: Финансы и статистика, 2002.

35. Б.Б. Демешев, Лекция: Основы анализа данных.

36. Технический анализ. URL: https://ru.wikipedia.org/wiki/Технический_анализ (Дата обращения: 05.06.2020г).

37. Фундаментальный анализ. URL: https://ru.wikipedia.org/wiki/Фундаментальный_анализ (Дата обращения: 05.06.2020г).

38. Multilayer perceptron. URL: https://en.wikipedia.org/wiki/Multilayer_perceptron (Дата обращения: 06.06.2020г).

39. Hastie, Trevor. Tibshirani, Robert. Friedman, Jerome. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY, 2009.

40. Рекуррентные нейронные сети. URL: https://ru.wikipedia.org/wiki/Рекуррентная_нейронная_сеть (Дата обращения: 06.06.2020г).

41. Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence: journal. — 2009. — Vol. 31, no. 5.

42. Sak, Hasim Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling.

43. Li, Xiangang & Wu, Xihong (2014-10-15), Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition.

44. Стационарность. URL: https://ru.wikipedia.org/wiki/Стационарность (Дата обращения: 07.06.2020г).

45. Что такое бар в трейдинге. URL: https://blog.purnov.com/chto-takoe-bar-v-trejdinge/ (Дата обращения: 08.06.2020г).

ПРИЛОЖЕНИЯ

Приложение А. Техническое задание

Требуется разработать модели прогнозирования дальнейшего движения тренда и скрипты, которые позволят биржевые данные выгрузить в csv-файл.

Далее необходимо разработать модели расчёта “эталонных значений” прогноза, с которыми будут сравниваться результаты, полученные из классификационных моделей.

Кроме того, необходимо сравнить эффективность прогнозных возможностей нейросетей и других классификаторов.

Приложение В. Индикаторы

В работе использовались следующие индикаторы:

1. Moving Average — общее название для семейства функций, значения которых в каждой точке определения равны среднему значению исходной функции за предыдущий период.

2. Bollinger Bands — инструмент технического анализа финансовых рынков, отражающий текущие отклонения цены акции, товара или валюты. Индикатор рассчитывается на основе стандартного отклонения от простой скользящей средней. Обычно отображается поверх графика цены. Параметрами для расчета служит тип стандартного отклонения (обычно двойное) и период скользящей средней.

3. Alligator — индикатор технического анализа, состоящий из трех (в оригинальной версии – сглаженных). Эти MA имеют разный период, а также сдвинуты вперед на графике. Быстрая MA (зеленая, или «губы») в оригинальной версии, разработанной самим Вильямсом, имеет период 5, и сдвинута вперед на 3 единичных отсечки таймфрейма. Средняя MA (красная, или «зубы»), имеет период 8 и сдвинута на 5. Медленная MA (синяя, или «челюсть»), имеет период 13 и сдвинута на 8.

4. Parabolic SAR — технический индикатор, разработанный Уэллсом Уайлдером и представленный в июне 1978 года в его книге «Новые концепции в технических торговых системах». Цель параболической системы — определить допуск в рамках которого возможно движение цены, для того, чтобы оставаться в текущем тренде. Он рисует снизу графика точки, если тренд восходящий и сверху, если тренд нисходящий.

5. Triple Exponential Moving Average — индикатор был представлен в январе 1994 года Патриком Маллой в статье в журнале Технический анализ акций и товаров : "Сглаживание данных с помощью более быстрых скользящих средних". Он пытается убрать внутреннюю задержку, связанную с скользящими средними, увеличивая вес последних значений. Название предполагает, что это достигается путем применения тройного экспоненциального сглаживания.

6. MACD — технический индикатор, разработанный Джеральдом Аппелем, используемый в техническом анализе для оценки и прогнозирования колебаний цен на фондовой и валютной биржах. Индикатор используют для проверки силы и направления тренда, а также определения разворотных точек. Строится на основе скользящих средних. Существует две модификации индикатора MACD: линейный MACD и MACD-гистограмма.

7. Stochastic — индикатор технического анализа, который показывает положение текущей цены относительно диапазона цен за определенный период в прошлом. Измеряется в процентах. Согласно толкованию автора индикатора Джорджа Лэйна, основная идея состоит в том, что при тенденции роста цены (возрастающий тренд) цена закрытия очередного таймфрейма имеет тенденцию останавливаться вблизи предыдущих максимумов. При тенденции снижения цены (падающий тренд) цена закрытия очередного таймфрейма имеет тенденцию останавливаться вблизи предыдущих минимумов. Фактически, индикатор демонстрирует расхождение цены закрытия текущего периода относительно цен предыдущих периодов в рамках заданного временного промежутка.

8. Money Flow Index — технический индикатор, призванный показать интенсивность, с которой деньги вкладываются в ценную бумагу и выводятся из неё, анализируя объёмы торгов и соотношения типичных цен периодов.

9. Accelerator Oscillator — помогает трейдерам определить ускорение и замедление движущей силы рынка. Accelerator Oscillator дает ранние сигналы об изменении силы рынка, тем самым информируя трейдера о смене торговой ситуации на рынке. Индикатор AC рассчитывается на основании индикатора Awesome Oscillator.

10. Awesome Oscillator — индикатор, показывающий разницу между двумя простыми скользящими средними, что позволяет определить движущую силу рынка. Автор разработал данный индикатор на основе уже существующего индикатора MACD, внеся несколько изменений.

11. Bears Power — классический осциллятор, определяющий силу продавцов рынка в конкретный временной период. Этот простой удобный

инструмент был разработан известным «советским» американцем Александром Элдером, как один из компонентов его же комплексного индикатора Лучи.

12. Bulls Power — это авторская разработка врача-психиатра Александра Элдера, лежащая в основе стратегии «Рентген рынка», описанной в книге «Trading for a Living». Став трейдером, Элдер сумел применить профессиональные медицинские навыки в рыночной торговле, сконцентрировавшись на поиске методов анализа «поведения толпы». Определяет силу покупателей рынка в конкретный временной период.

13. BW-ZoneTrade — технический инструмент определяет на графике наличие зеленых, красных и серых участков, которые были описаны в книге Билла Вильямса. Зеленые участки характеризуются покупками call, красные - продажами put, и серые обозначают рыночную консолидацию – флет. Индикатор BW-zone в сочетании с торговой системой, на основе японских свечей обеспечивает максимальный результат.

14. Custom Moving Average — индикатор, сглаживающий ценовые колебания путем преобразования этих колебаний в средние значения за выбранный период времени.

15. Chaikin Oscillator — средство технического анализа, отображающее особенности ускорения ценового движения. Будучи построенным на основе индикатора Ускорения/Замедления (Accumulation/Distribution, A/D), он являет собой разницу десятидневной и трехдневной экспоненциальных средних скользящих последнего.

```
#property copyright "Copyright 2020, MetaQuotes Software Corp."
#property link      "https://www.mql5.com"
#property version   "1.00"
void OnStart()
  {
   string sym="ES",year="2020",sym1="XAUUSD",sym2="BRN";
   ENUM_TIMEFRAMES tf=PERIOD_M15;
   datetime start=StringToTime("2020.12.31"),
end=StringToTime("2020.01.01");
   double spread=0.00000;
   Print("StartProgram");
   int
file=FileOpen(sym+"_"+sym1+"_"+sym2+"_"+year+"_"+getTimeFrame(tf)+"
.csv",FILE_WRITE|FILE_CSV,',');

FileWrite(file,"time","RESULT","hight","low","open","close","volume
","Moving Average","Bollinger Bands","Bollinger Bands1","Bollinger
Bands2","Alligator","Alligator1","Alligator2","Parabolic
SAR","Tripple Exponential Moving Average"
   ,"MACD","MACD1","Stochastic","Stochastic1","Money Flow
Index","Accelerator Oscillator","Awesome Oscillator","Bears
Power","Bulls Power"
   ,"BW-ZoneTrade","Custom Moving Average","Chaikin
Oscillator","high_oil","low_oil","open_oil","close_oil","volume_oil
","high_brn","low_brn","open_brn","close_brn","volume_brn");
   datetime date_arr[];
   double
h[],l[],o[],c[],ma[],bb[],bb1[],bb2[],alligator[],alligator1[],alli
gator2[],sar[],tema[],macd[],macd1[],stoch[],stoch1[],mfi[]

,ac[],ao[],bears[],bulls[],bw[],cma[],zigzag[],h_oil[],l_oil[],o_oi
l[],c_oil[],h_brn[],l_brn[],o_brn[],c_brn[];
   long volume[],volume_oil[],volume_brn[],result;
   CopyTime(sym,tf,start,end,date_arr);
```

```
   CopyHigh(sym,tf,start,end,h);
   CopyLow(sym,tf,start,end,l);
   CopyOpen(sym,tf,start,end,o);
   CopyClose(sym,tf,start,end,c);
   CopyTickVolume(sym,tf,start,end,volume);
   CopyBuffer(iMA(sym,tf,14,0,0,PRICE_CLOSE),0,start,end,ma);
   CopyBuffer(iBands(sym,tf,20,0,2,PRICE_CLOSE),0,start,end,bb);
   CopyBuffer(iBands(sym,tf,20,0,2,PRICE_CLOSE),1,start,end,bb1);
   CopyBuffer(iBands(sym,tf,20,0,2,PRICE_CLOSE),2,start,end,bb2);
CopyBuffer(iAlligator(sym,tf,13,8,5,8,5,3,MODE_EMA,PRICE_CLOSE),0,s
tart,end,alligator);
CopyBuffer(iAlligator(sym,tf,13,8,5,8,5,3,MODE_EMA,PRICE_CLOSE),1,s
tart,end,alligator1);
CopyBuffer(iAlligator(sym,tf,13,8,5,8,5,3,MODE_EMA,PRICE_CLOSE),2,s
tart,end,alligator2);
   CopyBuffer(iSAR(sym,tf,0.02,0.2),0,start,end,sar);
   CopyBuffer(iTEMA(sym,tf,14,0,PRICE_CLOSE),0,start,end,tema);
   CopyBuffer(iMACD(sym,tf,12,26,9,PRICE_CLOSE),0,start,end,macd);
   CopyBuffer(iMACD(sym,tf,12,26,9,PRICE_CLOSE),1,start,end,macd1);
CopyBuffer(iStochastic(sym,tf,5,3,3,MODE_SMA,STO_LOWHIGH),0,start,e
nd,stoch);
CopyBuffer(iStochastic(sym,tf,5,3,3,MODE_SMA,STO_LOWHIGH),1,start,e
nd,stoch1);
   CopyBuffer(iMFI(sym,tf,14,VOLUME_TICK),0,start,end,mfi);
   CopyBuffer(iAC(sym,tf),0,start,end,ac);
   CopyBuffer(iAO(sym,tf),0,start,end,ao);
   CopyBuffer(iBearsPower(sym,tf,13),0,start,end,bears);
   CopyBuffer(iBullsPower(sym,tf,13),0,start,end,bulls);
   CopyBuffer(iBWMFI(sym,tf,VOLUME_TICK),0,start,end,bw);
   CopyBuffer(iCustom(sym,tf,"Examples\\Custom Moving
Average",13,0,MODE_SMA,PRICE_CLOSE),0,start,end,cma);
CopyBuffer(iChaikin(sym,tf,3,10,MODE_EMA,VOLUME_TICK),0,start,end,z
igzag);
   CopyHigh(sym1,tf,start,end,h_oil);
   CopyLow(sym1,tf,start,end,l_oil);
   CopyOpen(sym1,tf,start,end,o_oil);
```

```
   CopyClose(sym1,tf,start,end,c_oil);
   CopyTickVolume(sym1,tf,start,end,volume_oil);
   CopyHigh(sym2,tf,start,end,h_brn);
   CopyLow(sym2,tf,start,end,l_brn);
   CopyOpen(sym2,tf,start,end,o_brn);
   CopyClose(sym2,tf,start,end,c_brn);
   CopyTickVolume(sym2,tf,start,end,volume_brn);

   for(int i=0;i<ArraySize(date_arr);i++){
      result=0;
      if(i+1<ArraySize(date_arr))
         if(c[i]+spread<=c[i+1]){
            result=1;
         }else if(c[i]-spread>=c[i+1]){
            result=0;
         }
FileWrite(file,date_arr[i],result,h[i],l[i],o[i],c[i],volume[i],ma[
i],bb[i],bb1[i],bb2[i],alligator[i],alligator1[i],alligator2[i],sar
[i],tema[i],macd[i],macd1[i], stoch[i],
stoch1[i],mfi[i],ac[i],ao[i],bears[i],bulls[i],bw[i],cma[i],zigzag[
i],h_oil[i],l_oil[i],o_oil[i],c_oil[i],volume_oil[i],h_brn[i],l_brn
[i],o_brn[i],c_brn[i],volume_brn[i]);
   }}
  string getTimeFrame(int lPeriod)
   {switch(lPeriod)
      {
      case PERIOD_M1:  return("M1");
      case PERIOD_M2:  return("M2");
      case PERIOD_M3:  return("M3");
      case PERIOD_M4:  return("M4");
      case PERIOD_M5:  return("M5");
      case PERIOD_M6:  return("M6");
      case PERIOD_M10: return("M10");
      case PERIOD_M12: return("M12");
      case PERIOD_M15: return("M15");
      case PERIOD_M20: return("M20");
```

```
   case PERIOD_M30: return("M30");
   case PERIOD_H1:  return("H1");
   case PERIOD_H2:  return("H2");
   case PERIOD_H3:  return("H3");
   case PERIOD_H4:  return("H4");
   case PERIOD_H6:  return("H6");
   case PERIOD_H8:  return("H8");
   case PERIOD_H12: return("H12");
   case PERIOD_D1:  return("D1");
   case PERIOD_W1:  return("W1");
   case PERIOD_MN1: return("MN1");
   }
return IntegerToString(lPeriod);
}
```

Приложение D. Описание индикатора

Отступ в % и пунктах от средневзвешенной цены VWAP по заданному диапазону по заданной цене, показывающий сигнал только при касании и запускающий alert, учитывая разницу в текущем и предыдущем объёме, учитывающий текущий тренд по сравнительному анализу средневзвешенной цены VWAP на open и close

Параметры:

- Up % — отступ в % вверх от рассчитанной цены индикатором(по-умолчанию 1)
- Down % — отступ в % вниз от рассчитанной цены индикатором(по-умолчанию 1)
- Pips Up — отступ в пунктах вверх от рассчитанной цены индикатором(по-умолчанию 0)
- Pips Down — отступ в пунктах вниз от рассчитанной цены индикатором(по-умолчанию 0)
- Volume Multiplicator — число, обозначающее во сколько раз прошедший объём текущего бара должен быть больше предыдущего, чтобы индикатор указал вход в позицию(по-умолчанию 0)
- Price Type — тип цены, относительно которого будут производиться все расчёты цены индикатора(по-умолчанию open)
- MA diapozon — учитывается, если выбран какой-то тип скользящей средней. Указывает на диапазон, по которому будет производиться расчёт средней(по-умолчанию 5)
- Count Function Type — тип функции расчёта. Если "single", то берётся значение цены *Price Type*, иначе рассчитывается цена индикатора по указанной функции
- VWAP Diapozon — диапазон, указывающий на число баров, по которым рассчитывается VWAP (по-умолчанию 0)
- Bar offset — число, которое указывает отступ в барах от последнего

- Take into account the trend — флажок-указатель на необходимость учитывания тренда при подаче сигналов от индикатора

Примечание:

- Если указать up %, down %, pips up и pips down равными 0(нулю), то индикатор будет показывать собственную рассчитанную цену БЕЗ отступов
- Если указать volume multiplicator равным 0(нулю), то НЕ будет учитываться разница в объёмах текущего и предыдущего баров при срабатывании сигнала индикатора на вход в сделку
- Если указать в *Count Function Type* параметр "single", то рассчитанная цена, по которой будет рассчитываться VWAP будет равняться *Price Type*
- Если указать в *VWAP Diapozon* 0, то цена будет рассчитываться по стандартной функции vwap , которая начинается с началом торговой сессии. Если указать 1, то цена после преобразования VWAP будет равна полученной цене(Пример: vwap (open)=open). В иных случаях цена vwap будет рассчитываться числу баров, указанных в диапазоне
- Если указать Bar offset равным 0(нулю), то будет строиться цена индикатора относительно последнего бара на графике. Если 1, то относительно предпоследнего и тд…
- Take into account the trend — если выбрана галочка, то при показе сигналов индикаторов учитывается тренд. Если не выбрана, то тренд не учитывается.
- VWAP diapozon — добавлена новая возможность! Если значение выбрано равное -1!!!, то будет vwap обновляться в начале каждого дня и при получении большего объёма за день, чем предыдущий!