

Федеральное государственное автономное образовательное учреждение
высшего образования «Московский политехнический университет»,
Московский Политех

ОТЧЕТ

по проектной деятельности
Проект по алгоритмической торговле “AlgoTrade”

Выполнили:
Студенты группы 233-321
Насыров Булат Ильшатovich
Зазуля Никита Владиславovich
Иванов Валерий Александрovich
Пухов Юрий Евгеньевич
Силина Анастасия Артемовна
Гафуров Раиль Ильдарovich
Баратов Зухриддин Фахриддин угли

Москва, 2024

Содержание

| | |
|--------------------------------------|----|
| 1 АКТУАЛЬНОСТЬ ПРОЕКТА | 2 |
| 2 ПРОБЛЕМАТИКА ПРОЕКТА | 5 |
| 3 ЦЕЛЬ И ЗАДАЧИ ПРОЕКТА | 6 |
| 4 ЦЕЛЕВАЯ АУДИТОРИЯ | 7 |
| 5 СТЕЙКХОЛДЕРЫ ПРОЕКТА | 8 |
| 6 АНАЛИЗ РЫНКА И КОНКУРЕНТОВ | 9 |
| 7 ПРОДУКТОВЫЙ РЕЗУЛЬТАТ | 15 |
| 8 КАЛЕНДАРНЫЙ ПЛАН ПРОЕКТА | 19 |
| 9 РЕСУРСНЫЙ ПЛАН ПРОЕКТА | 26 |
| 10 РИСКИ ПРОЕКТА | 30 |
| 11 SWOT - АНАЛИЗ ПРОЕКТА | 35 |
| 12 ПЕРСПЕКТИВЫ ПРОЕКТА | 37 |
| 13 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА | 40 |

Используемые источники

Приложения

1 Актуальность проекта

Проект по алгоритмической торговле на финансовых рынках остается актуальным и востребованным в современном мире. Алгоритмическая торговля представляет собой использование компьютерных программ для выполнения торговых операций на финансовых рынках с использованием предварительно заданных правил и стратегий.

- Алгоритмические торговые системы способны реагировать на изменения рыночных условий мгновенно и выполнить огромное количество сделок за короткое время. Это позволяет трейдерам получать прибыль даже при быстро изменяющихся рыночных условиях.
- Алгоритмическая торговля исключает влияние эмоций и субъективных решений трейдера, что может привести к более стабильным и предсказуемым результатам.
- Алгоритмические системы позволяют автоматизировать торговлю и оптимизировать стратегии на основе большого объема данных и аналитики.
- С постоянным развитием технологий и доступом к большим объемам данных алгоритмическая торговля становится все более точной и эффективной.

Однако стоит помнить, что алгоритмическая торговля также ставит перед собой вызовы, такие как необходимость постоянного мониторинга и обновления стратегий, а также риски, связанные с высокой волатильностью рынка и возможностью технических сбоев.

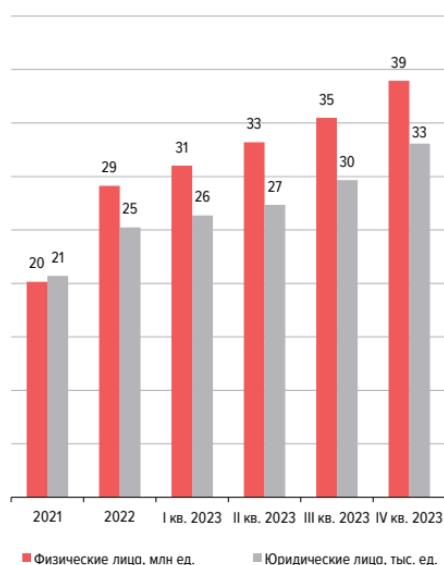
Согласно данным от центрального банка РФ на 2023 год:

“В 2023 г. количество клиентов брокеров продолжало расти благодаря высокому интересу населения к фондовому рынку и увеличению доходов

населения. Несмотря на постепенное насыщение клиентской базы брокеров, темпы роста ускорились в конце года в условиях повышенных инфляционных ожиданий, ослабления рубля и роста фондового индекса.

Суммарное количество клиентов (по всем участникам, включая повторные счета у разных брокеров) по итогам IV квартала 2023 г. выросло до 39 млн лиц (+10% к/к и +34% г/г). При этом, по данным Московской Биржи, число физических лиц, имеющих брокерские счета, с уникальными паспортными данными выросло до 29,7 млн лиц (+8% к/к и +29% г/г) и составило 39% экономически активного населения страны. При этом сделки на фондовом рынке Московской Биржи заключали в среднем 3 млн лиц ежемесячно, а сделки на валютном рынке – 0,4 млн лиц (годом ранее – 2,2 и 0,7 млн лиц соответственно), рис. 1”

РОСТ ЧИСЛА КЛИЕНТОВ БРОКЕРОВ УСКОРИЛСЯ
В КОНЦЕ ГОДА...



...А ОБЪЕМ АКТИВОВ УВЕЛИЧИВАЛСЯ НА ПРОТЯЖЕНИИ ГОДА
ЗА СЧЕТ ВЗНОСОВ И ПОЛОЖИТЕЛЬНОЙ ПЕРЕОЦЕНКИ
(ТРЛН РУБ.)

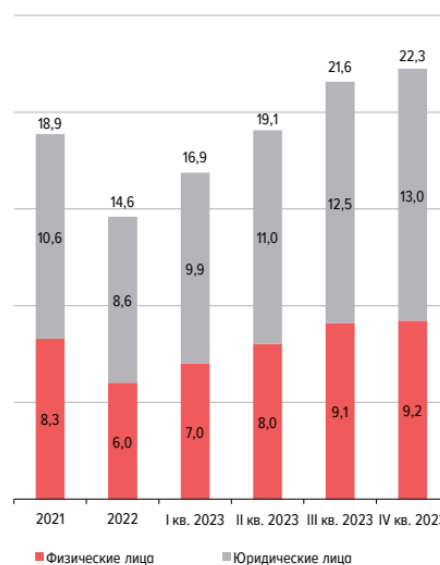


Рис. 1

Данные показатели позволяют убедиться в актуальности и необходимости проекта по алгоритмической торговле. В связи, с желанием людей вложить свои финансовые ресурсы в различные виды активов, не углубляясь в их особенностях.

2 Проблематика проекта

Проект по алгоритмической торговле может столкнуться с рядом проблем.

1. Алгоритмическая торговля требует высококвалифицированных специалистов для разработки и поддержки алгоритмов.
2. Алгоритмическая торговля может быть подвержена риску неисправностей и ошибок в алгоритмах, что может привести к значительным потерям.
3. Алгоритмическая торговля может плохо работать в неожиданных рыночных условиях, поскольку она в значительной степени зависит от исторических данных.
4. Алгоритмическая торговля может быть подвержена риску мошенничества и манипуляций на рынке.
5. Алгоритмическая торговля может быть подвержена риску регуляторных изменений, которые могут повлиять на ее эффективность.

3 Цель и задача проекта

Основные цели проекта:

- Решение проблем связанных с постоянным наблюдением за рынком;
- Универсальность использования, т.е. может быть использован для: фондового рынка для торговли ценными бумагами и их производными финансовыми инструментами (таких, как фьючерсы, опционы), криптовалютных рынках и сырьевых рынках;
- Увеличение капитала для крупных участников рынка, таких как хедж-фонды, банки, страховые компании и частные лица;
- Удобства применения программного обеспечения

На данный момент разрабатывается на языке программирования Python, но в дальнейшем планируется быть переписан на языке Си, для большей скорости работы. Используются основные модели прогнозирования изменчивости, такие как:

- Обобщенная авторегрессионная модель гетероскедастичности (Generalized Autoregressive Conditional Heteroscedastic model, GARCH) - Модель, используемая для прогнозирования ситуации на финансовых рынках в условиях нестабильности (волатильности).
- Авторегрессионная условная гетероскедастичность (AutoRegressive Conditional Heteroscedasticity, ARCH) - применяемая в эконометрике модель для анализа временных рядов (в первую очередь финансовых), у которых условная (по прошлым значениям ряда) дисперсия ряда зависит от прошлых значений ряда, прошлых значений этих дисперсий и иных факторов.

Сегодня алгоритмы, применяемые большинством участников торгов, решают основную задачу – анализ рынка, направленный на выявление его неэффективности. Именно такая ситуация (неэффективность рынка) считается лучшим моментом для входа и дает величину математического ожидания, позволяющую с высокой долей вероятности извлекать прибыль.

Эти задачи принято разделять на несколько классов, например:

- Технический анализ.
- Корреляционный (парный) трейдинг.
- Анализ объемов за счет работы со стаканом цен и лентой принтов.
- Арбитраж.
- Торговля волатильностью.

4 Целевая аудитория проекта

Современная биржевая торговля – высокоавтоматизированный процесс. Использование компьютерной техники для подачи и обработки заявок повысило скорость установки и отработки ордеров, что позволило участникам рынка более оперативно реагировать на изменение обстановки. При этом исчезла необходимость в присутствии человека в торговом зале - появилась возможность работать исключительно онлайн. В качестве основных интересантов алгоритмической торговли можно выделить:

- Крупных частных инвесторов (с капиталом более 100 млн. руб.)
- Финансовые институты с большим капиталом (хедж-фонды, пенсионные фонды и ПИФы)

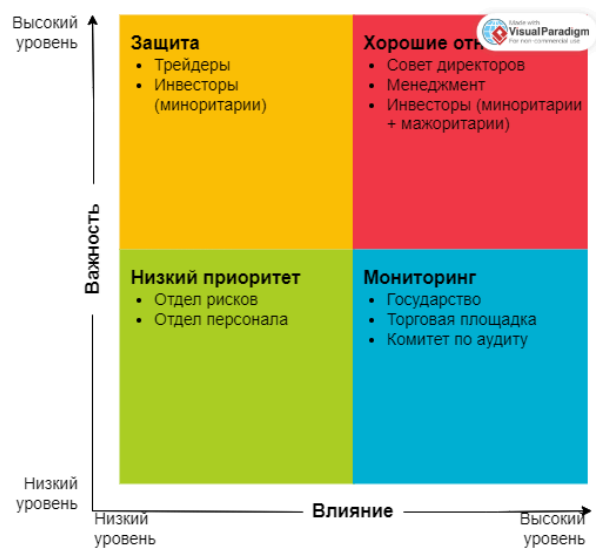
Также продукт по алгоритмической торговле могут использовать участники рынка с капиталом около 1 млн. руб. и капиталом более 100 тыс. руб., с другой ценовой сеткой (другими тарифами и более пассивных стратегий).

5 Стейкхолдеры проекта

Стейкхолдеры в проекте по алгоритмической торговле могут быть разнообразными и включать в себя следующие группы:

1. Инвесторы, которые предоставляют капитал для торговли и ожидают получения прибыли от успешных операций.
2. Специалисты, занимающиеся разработкой и настройкой алгоритмов торговли, а также мониторингом и анализом рыночных условий.
3. Программисты, аналитики данных и инженеры, отвечающие за разработку и поддержку алгоритмических систем.
4. Лица, принимающие стратегические решения о распределении средств и выборе торговых стратегий.
5. Организации, отвечающие за контроль и регулирование деятельности на финансовых рынках, включая алгоритмическую торговлю.
6. Лица или организации, сотрудничающие с проектом по алгоритмической торговле и пользующиеся его услугами.
7. Люди, которые могут быть заинтересованы в деятельности проекта по алгоритмической торговле или влиянии его на финансовые рынки.

Успешная реализация проекта по алгоритмической торговле требует учета интересов и потребностей всех заинтересованных сторон, а также установления эффективного взаимодействия между ними для достижения общих целей и успеха проекта.



6 Анализ рынка и конкурентов

| Анализ конкурентов | | | | | Общая оценка | Комментарий |
|--------------------|---|--|--|--|--------------|-------------|
| Компания | Сайт | Что зацепило | Что отпугнуло | Цена | | |
| ITInvest | https://itinvest.ru/education/glossary/allgotrejdning/ | Простой и удобный сайт, лицензированный брокер | Мало информации на каких рынках доступна торговля, нет комментариев от других пользователей. | На фондовом рынке от 0,0087% от суммы сделки На срочном рынке от 20% от комиссии биржи На валютном рынке от 0,004% от суммы сделки | 4 | |

| | | | | | | |
|-------------------|---|---------------------------------|--|------------|--|------------------------------|
| IFC MARKETS | https://www.ifcmarkets.com/ru/about-forex/algorithms-trading | Много информации и удобный сайт | Несколько лицензий одна на Кипре, другая на Британских Виргинских островах, слишком много информации, а тарифов нет. | Не указано | | Сайт не внушает доверия 2 |
| Fast erCapital | https://www.ifcmarkets.com/ru/ifcm-invest-landing | Много информации | Нет информации о лицензии, доступна торговля только на Forex, плохой и неудобный сайт, часть сайта на английском | Не указано | | Сайт не внушает доверия 2 |

| | | | | | | | |
|----|-----|---|--|---|---|--------|--|
| ам | Фин | https://www.fina.m.ru/ | Мно го информац ии и знакомая компания, есть лицензия ЦБ РФ. | Тре буется время, чтобы разобрать ся на сайте. | Стр атег - 0,05% Инвестор - 0,035% и другие. | 1 0 | |
| | БКС | https://bcs.ru/ | Удо бный и понятный сайт, с большим количеством информац ии, есть лицензия ЦБ РФ и знакомая компания. | Нет информац ии о настройке алгоритми ческой торговли | Тре йдер - 0,01% и 300 руб./мес. | 1 0 | |

Размер рынка алгоритмической торговли

Рынок алгоритмической торговли в текущем году оценивается в 14,42 миллиарда долларов США, и ожидается, что среднегодовой темп роста составит 8,53%, достигнув 23,74 миллиарда долларов США за пять лет.



Трейдеры традиционно использовали технологии наблюдения за рынком, чтобы отслеживать свои торговые операции и инвестиционные портфели. Приложения со встроенным интеллектом, такие как алгоритмическая торговля, могут исследовать рынок в поисках различных возможностей на основе доходности и других параметров, указанных пользователем.

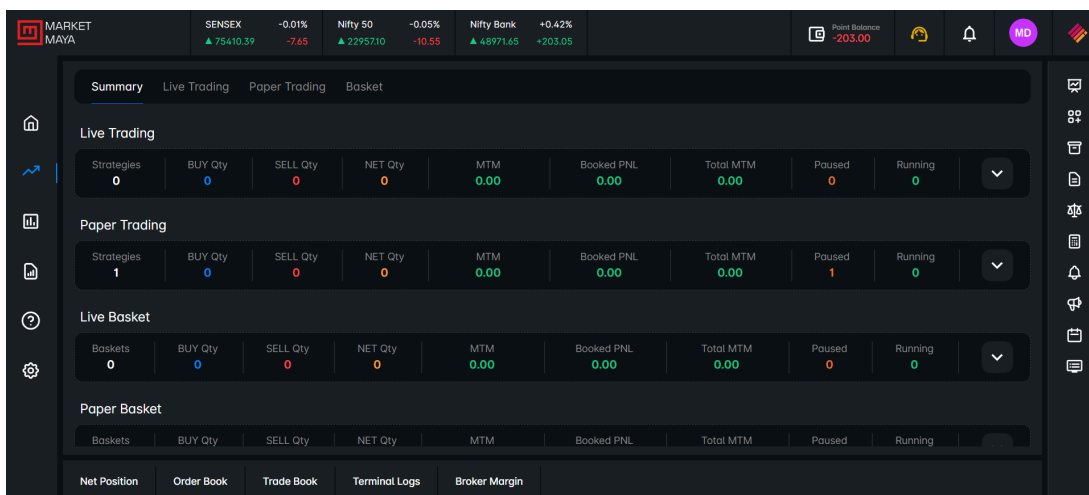
7 Продуктовый результат

На данный момент, готовый результат.

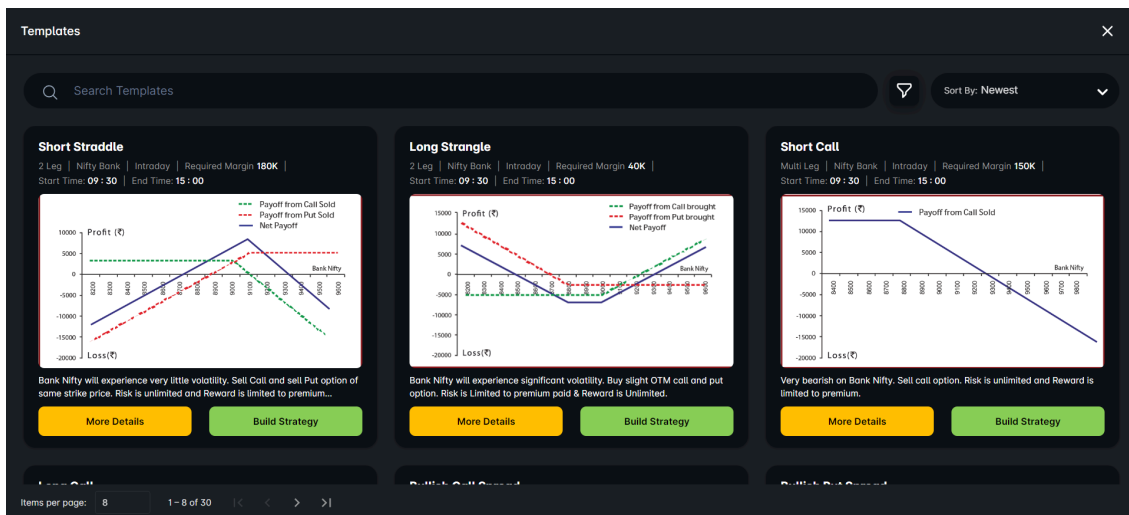
```
==== Constant Mean - GARCH Model Results =====
Dep. Variable:      returns      R-squared:      0.000
Mean Model:        Constant Mean  Adj. R-squared:  0.000
Vol Model:         GARCH          Log-Likelihood: -1417.76
Distribution:      Normal         AIC:            2843.51
Method:           Maximum Likelihood  BIC:            2863.15
                                     No. Observations: 1000
Date:             Sun, Dec 10 2023  Df Residuals:      999
Time:             13:26:10          Df Model:        1
                                     Mean Model
=====
              coef      std err      t      P>|t|      95.0% Conf. Int.
-----
mu           -0.0409   3.156e-02   -1.295   0.195 [ -0.103, 2.100e-02]
Volatility Model
=====
              coef      std err      t      P>|t|      95.0% Conf. Int.
-----
omega        0.0158   6.237e-03    2.536   1.120e-02 [ 3.596e-03, 2.805e-02]
alpha[1]     0.0000   1.021e-02    0.000   1.000 [-2.001e-02, 2.001e-02]
beta[1]      0.9837   1.357e-02   72.494   0.000 [ 0.957, 1.010]
=====

Covariance estimator: robust
count      1000.000000
mean        0.000636
std         1.002662
min        -3.212172
25%        -0.651059
50%        -0.000327
75%         0.703409
max         3.670723
Name: std_resid, dtype: float64
```

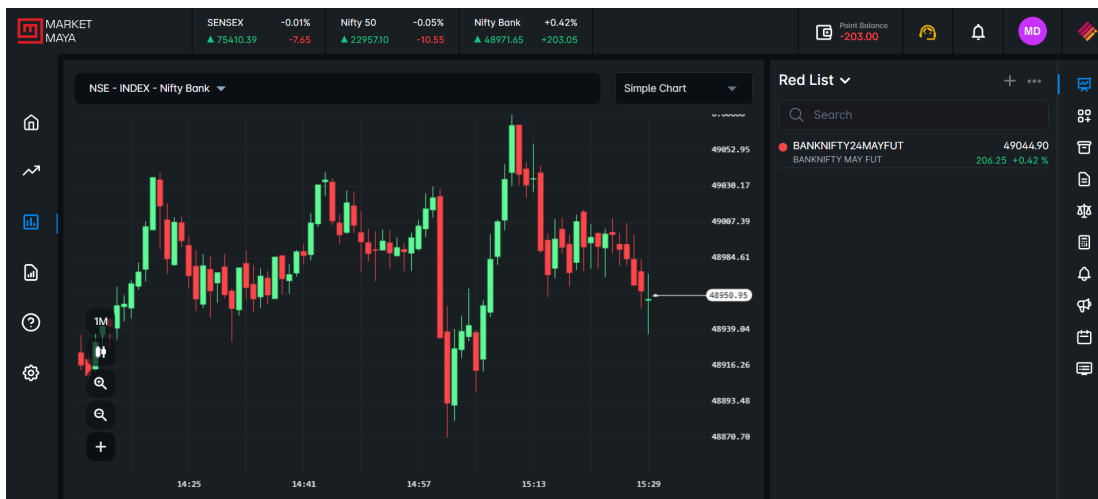
1. Главная страница с открытыми позициями по балансовому счету.



2. Страница выбора стратегии/алгоритма для торговли



3. Страница с графиком инструмента, для удобства будут отображаться сделки открытия и закрытия позиций.



4. Страница история открытий и закрытий позиций по всем, либо выбранным инструментам.

Reports

Ledger Point Usage Strategy PNL Basket PNL Daily PNL Monthly PNL Order Book Trade Book Activity Logs Login Logs

Search by keyword Today

| Date | Particular | Credit | Debit | Balance |
|----------------------|-----------------|--------|---------|---------|
| 26 May 2024 00:00:00 | Opening Balance | - | -203.00 | -203.00 |
| 26 May 2024 23:59:59 | Closing Balance | - | -203.00 | -203.00 |
| | | 0.00 | 0.00 | -203.00 |

Items per page: 25 1 - 2 of 2

В приложение 1 показан класс ARCHModel, который представляет собой модель авторегрессивной условно гетероскедастической модели (ARCH), которая используется для моделирования изменяющейся дисперсии временного ряда.

Класс содержит следующие атрибуты:

- data: входные данные
- n_obs: количество наблюдений
- p: порядок авторегрессии
- o: порядок скользящей средней
- q: порядок авторегрессии в квадрате остатков
- dist: тип распределения ошибок (например, нормальное)
- params: оценки параметров модели
- arch_err: остатки модели ARCH
- sigma2: дисперсия модели ARCH

Методы класса:

- __init__: инициализация класса
- initialize: инициализация атрибутов arch_err и sigma2
- residuals: вычисление остатков модели ARCH
- sigma2_update: обновление дисперсии модели ARCH
- fit: оценка параметров модели ARCH
- forecast: прогноз дисперсии модели ARCH

GARCH (Generalized Autoregressive Conditional Heteroskedasticity) - это модель, используемая в финансовой эконометрике для описания волатильности финансовых инструментов. Код, который вы предоставили, реализует GARCH-модель с помощью Python и библиотеки arch.

В приложение 2 показан класс GARCHModel, который наследуется от ARCHModel и добавляет функциональность для работы с GARCH-моделью. Модель имеет следующие параметры:

- p: количество лагов в автокорреляционной части модели
- o: количество лагов в moving average части модели
- q: количество лагов в GARCH-части модели
- dist: тип распределения ошибок (по умолчанию - нормальное распределение)

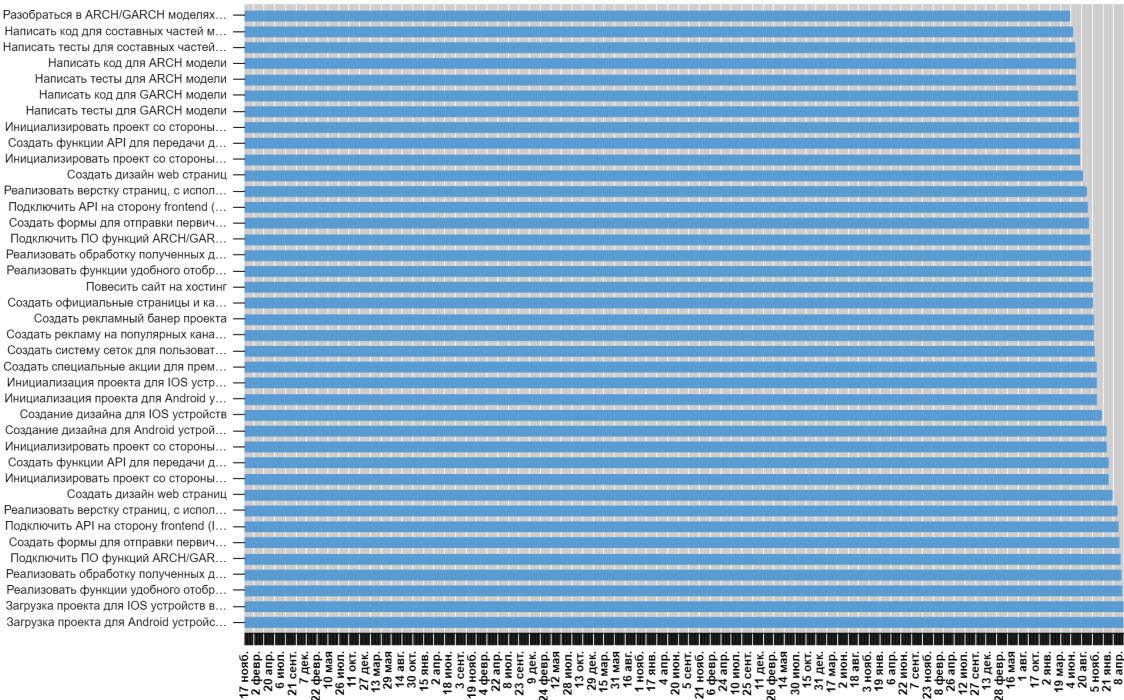
Методы класса:

- `__init__`: инициализирует модель с данными, параметрами p , o , q и типом распределения ошибок
- `initialize`: инициализирует внутренние переменные модели
- `sigma2_update`: обновляет значение волатильности на основе предыдущих ошибок и параметров модели
- `fit`: обучает модель на данных, используя метод максимизации правдоподобия
- `forecast`: производит прогноз волатильности на основе обученной модели

В методе `fit` происходит инициализация параметров модели, а затем происходит итеративное обновление параметров на основе ошибок и волатильности. Метод `forecast` использует обученную модель для прогноза волатильности на заданное количество шагов вперед.

8 Календарный план

диаграммы Ганта



| | График АлгоТрейд | | | | |
|---|---|----------------|------------------|--------------|----------------|
| | | | | | |
| | Этап проекта | На чало | Длит ельность | За держка | Кон ец |
| . | Разобраться в ARCH/GARCH моделях и их основных частях | 15. 05.2024 | 15 | 0 | 29. 05.2024 |
| | Написать код для составных частей моделей | 30. 05.2024 | 20 | 0 | 18. 06.2024 |
| | Написать тесты для составных частей моделей | 22. 06.2024 | 5 | 3 | 26. 06.2024 |
| | Написать код для ARCH модели | 27. 06.2024 | 10 | 0 | 06. 07.2024 |
| | Написать тесты для ARCH | 02. | 5 | -5 | 06. |

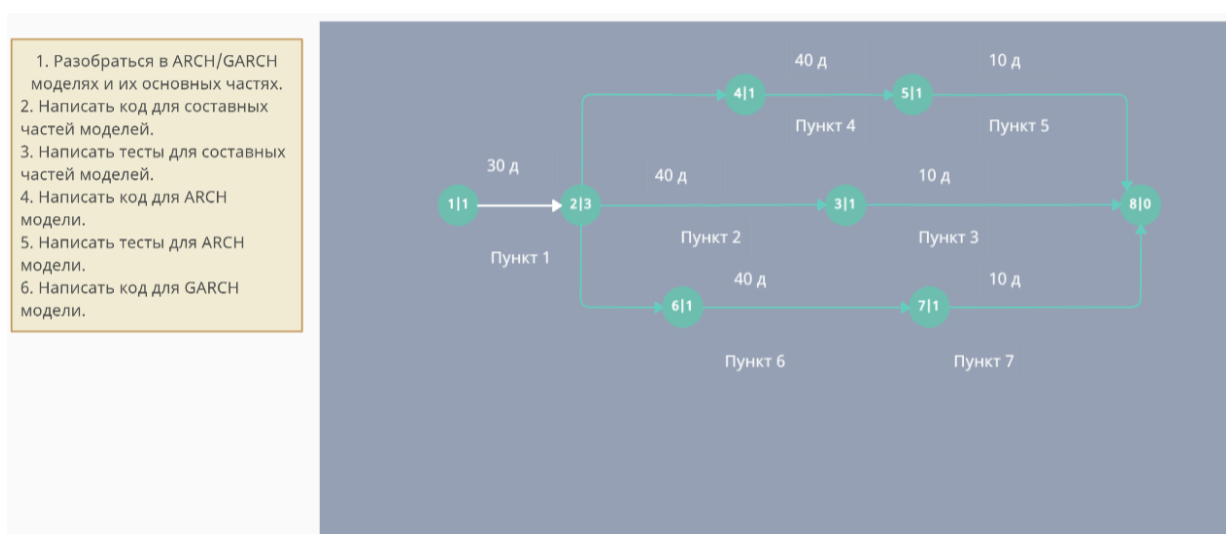
| | | | | | |
|---|---|----------------|----|----|----------------|
| | модели | 07.2024 | | | 07.2024 |
| | Написать код для GARCH модели | 11. 07.2024 | 10 | 4 | 20. 07.2024 |
| | Написать тесты для GARCH модели | 18. 07.2024 | 5 | -3 | 22. 07.2024 |
| . | Инициализировать проект со стороны backend (сервера) | 23. 07.2024 | 1 | 0 | 23. 07.2024 |
| | Создать функции API для передачи данных на сторону frontend (пользователя) | 24. 07.2024 | 10 | 0 | 02. 08.2024 |
| | Инициализировать проект со стороны frontend (пользователя) | 01. 08.2024 | 1 | -2 | 01. 08.2024 |
| | Создать дизайн web страниц | 02. 08.2024 | 15 | 0 | 16. 08.2024 |
| | Реализовать верстку страниц, с использованием HTML, CSS и JS | 17. 08.2024 | 30 | 0 | 15. 09.2024 |
| | Подключить API на сторону frontend (пользователя) | 16. 09.2024 | 2 | 0 | 17. 09.2024 |
| | Создать формы для отправки первичных данных на сторону backend, со стороны frontend | 18. 09.2024 | 10 | 0 | 27. 09.2024 |
| | Подключить ПО функций ARCH/GARCH моделей, на сторону backend | 28. 09.2024 | 5 | 0 | 02. 10.2024 |
| | Реализовать обработку полученных данных и передача их по API | 03. 10.2024 | 3 | 0 | 05. 10.2024 |
| | Реализовать функции удобного отображения данных на стороне frontend (пользователя) | 06. 10.2024 | 10 | 0 | 15. 10.2024 |
| | Повесить сайт на хостинг | 16. 10.2024 | 5 | 0 | 20. 10.2024 |

| | | | | | |
|---|--|----------------|----|---|----------------|
| . | Создать официальные страницы и каналы на популярных мессенджерах | 21. 10.2024 | 2 | 0 | 22. 10.2024 |
| | Создать рекламный банер проекта | 23. 10.2024 | 2 | 0 | 24. 10.2024 |
| | Создать рекламу на популярных каналах и других источниках | 25. 10.2024 | 2 | 0 | 26. 10.2024 |
| | Создать систему сеток для пользователей, в зависимости от стоимости используемых продуктов | 27. 10.2024 | 10 | 0 | 05. 11.2024 |
| | Создать специальные акции для премиальных пользователей | 06. 11.2024 | 10 | 0 | 15. 11.2024 |
| . | Инициализация проекта для IOS устройств | 16. 11.2024 | 1 | 0 | 16. 11.2024 |
| | Инициализация проекта для Android устройств | 17. 11.2024 | 1 | 0 | 17. 11.2024 |
| | Создание дизайна для IOS устройств | 18. 11.2024 | 30 | 0 | 17. 12.2024 |
| | Создание дизайна для Android устройств | 18. 12.2024 | 30 | 0 | 16. 01.2025 |
| | Инициализировать проект со стороны backend (сервера) | 17. 01.2025 | 1 | 0 | 17. 01.2025 |
| | Создать функции API для передачи данных на сторону frontend (IOS/Android) | 18. 01.2025 | 10 | 0 | 27. 01.2025 |
| | Инициализировать проект со стороны frontend (IOS/Android) | 28. 01.2025 | 1 | 0 | 28. 01.2025 |
| | Создать дизайн web страниц | 29. 01.2025 | 30 | 0 | 27. 02.2025 |
| | Реализовать верстку страниц, с использованием HTML, CSS и JS | 28. 02.2025 | 30 | 0 | 29. 03.2025 |

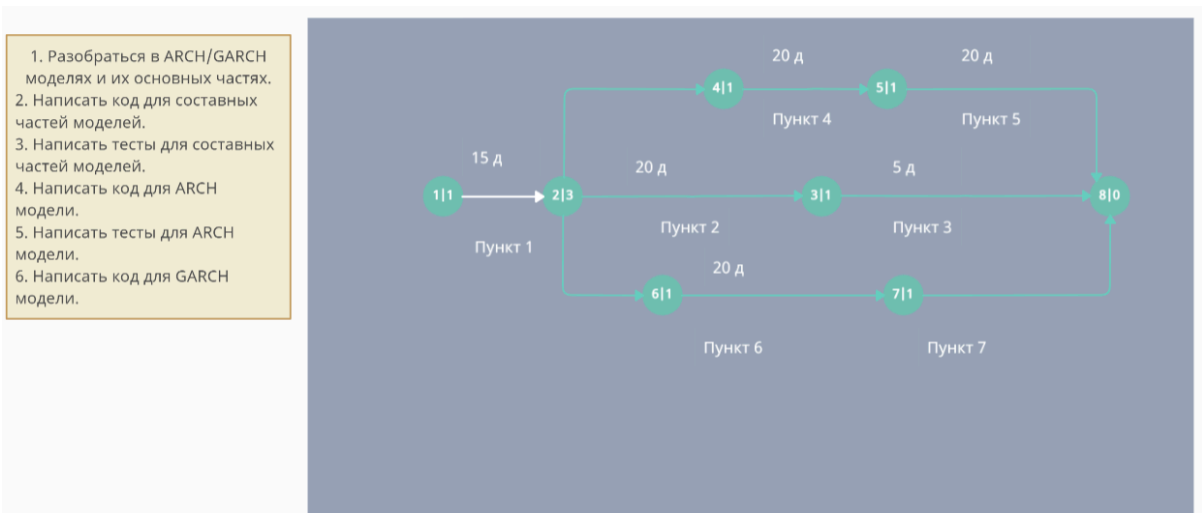
| | | | | | |
|--|---|----------------|----|---|----------------|
| | Подключить API на сторону frontend (IOS/Android) | 30. 03.2025 | 4 | 0 | 02. 04.2025 |
| | Создать формы для отправки первичных данных на сторону backend, со стороны frontend | 03. 04.2025 | 5 | 0 | 07. 04.2025 |
| | Подключить ПО функций ARCH/GARCH моделей, на сторону backend | 08. 04.2025 | 10 | 0 | 17. 04.2025 |
| | Реализовать обработку полученных данных и передача их по API | 18. 04.2025 | 3 | 0 | 20. 04.2025 |
| | Реализовать функции удобного отображения данных на стороне frontend (IOS/Android) | 21. 04.2025 | 10 | 0 | 30. 04.2025 |
| | Загрузка проекта для IOS устройств в App Store | 01. 05.2025 | 4 | 0 | 04. 05.2025 |
| | Загрузка проекта для Android устройств в Google Play | 05. 05.2025 | 4 | 0 | 08. 05.2025 |

PERT - анализ

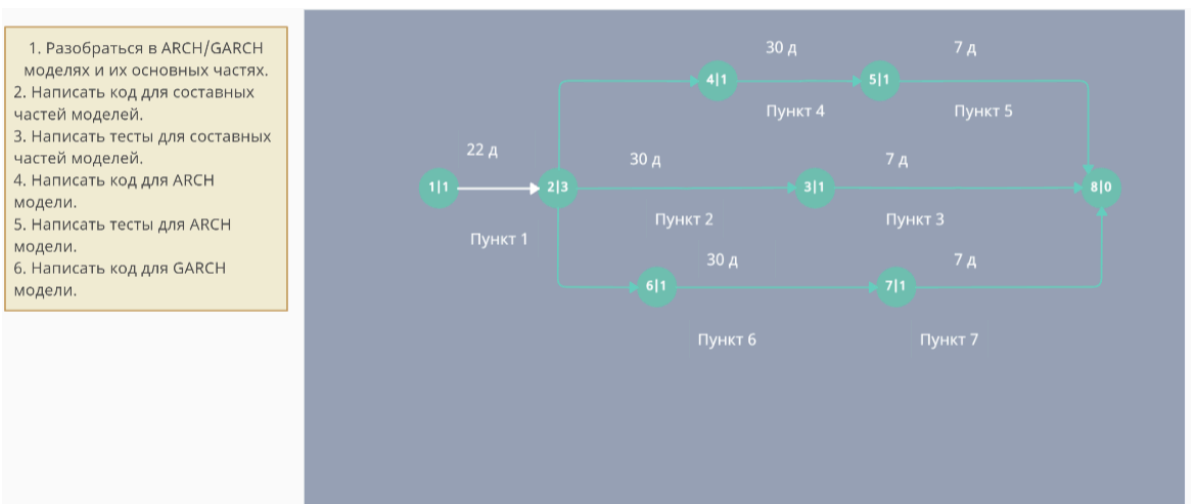
Худший вариант выполнения разработки проекта.



Наилучший вариант разработки проекта.



Средний вариант разработки проекта.



| Описание вехи | Кат егория | Кому назначено | Ход выполнен ия | Н ачало | Д ни |
|---------------|---------------|-------------------|-----------------------|------------|---------|
|---------------|---------------|-------------------|-----------------------|------------|---------|

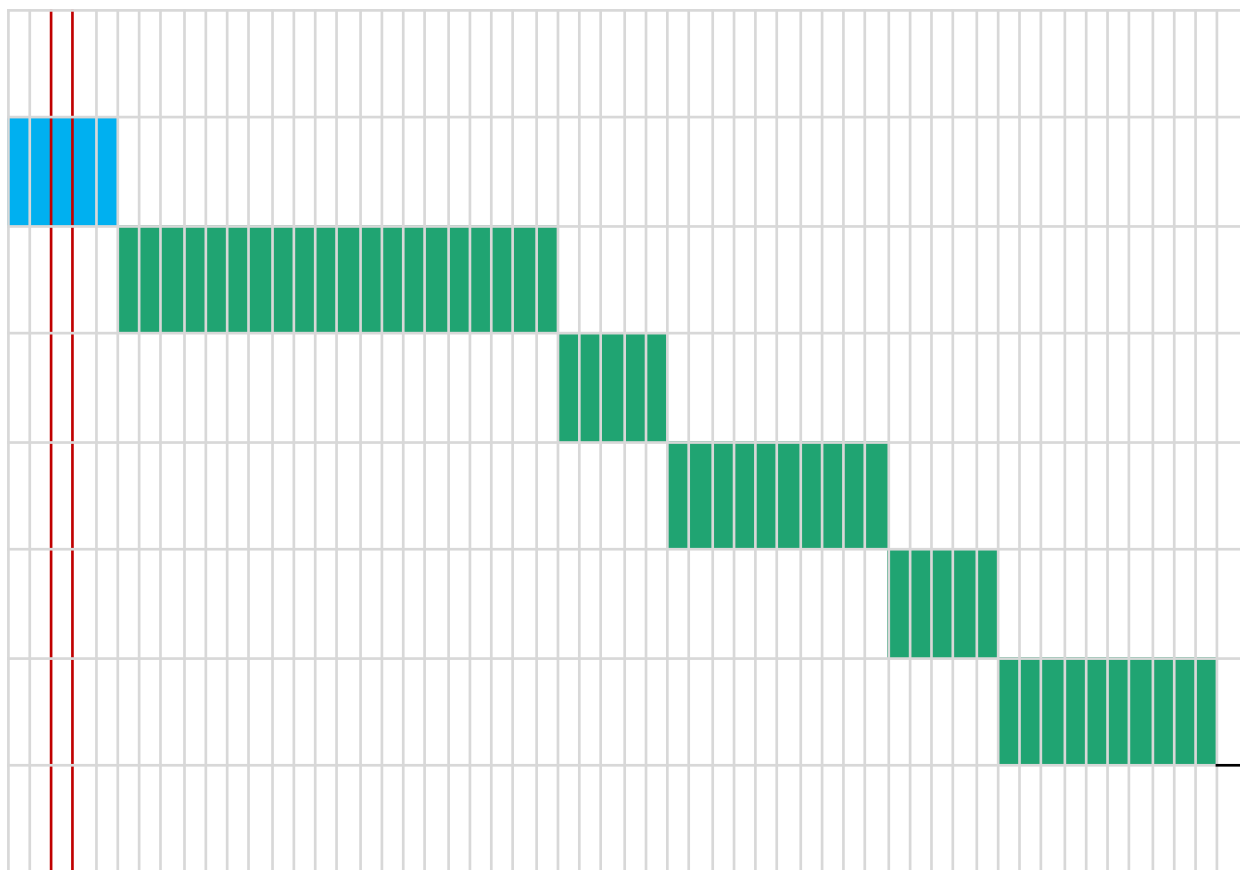
Разработка функций

ARCH/GARCH моделей

| | | | | | |
|---|-----------------|--|---------|--------------------|--------|
| Разобраться в ARCH/GARCH моделях и их основных частях | Низ кий риск | | 90 % | 1 5.05.202 4 | 1 5 |
| Написать код для составных частей моделей | По плану | | 90 % | 3 0.05.202 4 | 2 0 |

| | | | | |
|---|----------|-----|------------|----|
| Написать тесты для составных частей моделей | По плану | 90% | 19.06.2024 | 5 |
| Написать код для ARCH модели | По плану | 50% | 24.06.2024 | 10 |
| Написать тесты для ARCH модели | По плану | 10% | 4.07.2024 | 5 |
| Написать код для GARCH модели | По плану | 0% | 9.07.2024 | 10 |
| Написать тесты для GARCH модели | По плану | 0% | 19.07.2024 | 5 |

| | | | | |
|---------------------------|------------------------|----------------------|----------------------|-------------------------|
| С огласн о плану | И изки й риск | С редни й риск | В ысоки й риск | И е назна чено |
|---------------------------|------------------------|----------------------|----------------------|-------------------------|



9 Ресурсный план проекта

| | | | | | |
|---|-----------|-----------|----------|----------|--|
| Человек в команде | 7 | | | | |
| Обсуждения и встречи | 0,1 5 | | | | |
| Поддержка/баги | 0,2 5 | | | | |
| | | | | | |
| | | | | | |
| Разработка ARCH/GARCH моделей для ПО | | | | | |
| Часы | Ма й | Ию нь | Ию ль | | |
| Спринты | 1. 2/3 | 1/3 | 1/3 | | |
| Чистое время | 672 | 100 8 | 112 0 | | |
| Common задачи | 100 ,8 | 151 ,2 | 168 | | |
| Поддержка/баги | 168 | 252 | 280 | | |
| Задачи | 403 ,2 | 604 ,8 | 672 | 168 0 | |
| | | | | | |
| Задачи | | | | | |
| Разработка основных компонентов | 134 ,4 | 201 ,6 | 224 | | |
| Разработка ARCH модели | 134 ,4 | 201 ,6 | 224 | | |
| Разработка GARCH модели | 134 ,4 | 201 ,6 | 224 | | |
| Итого в спринте | 403 ,2 | 604 ,8 | 672 | 168 0 | |
| | | | | | |

| | | | | | |
|--------------------------------------|-----------|--------|----------|---------|--------|
| | | | | | |
| Разработка WEB-приложения | | | | | |
| Часы | Июль | Август | Сентябрь | Октябрь | |
| Спринты | 1. 2/3 | 1/3 | 1/3 | 1/3 | |
| Чистое время | 448 | 1120 | 1176 | 1288 | |
| Common задачи | 67,2 | 168 | 176,4 | 193,2 | |
| Поддержка/баги | 112 | 280 | 294 | 322 | |
| Задачи | 268,8 | 672 | 705,6 | 772,8 | 1646,4 |
| | | | | | |
| Задачи | | | | | |
| Разработка Backend | 89,6 | 224 | 235,2 | 257,6 | |
| Разработка Frontend | 89,6 | 224 | 235,2 | 257,6 | |
| Разработка связки ПО с сервером | 89,6 | 224 | 235,2 | 257,6 | |
| Итого в спринте | 268,8 | 672 | 705,6 | 772,8 | 1646,4 |
| | | | | | |
| | | | | | |
| Маркетинг | | | | | |
| Часы | Октябрь | Ноябрь | | | |
| Спринты | 1. 2/3 | 1/3 | | | |
| Чистое время | 224 | 616 | | | |
| Common задачи | 33,6 | 92,4 | | | |

| | | | | | |
|------------------------|-------|-------|-----|--|--|
| Поддержка/баги | 56 | 154 | | | |
| Задачи | 134,4 | 369,6 | 504 | | |
| | | | | | |
| Задачи | | | | | |
| Реклама | 44,8 | 123,2 | | | |
| Создание системы сеток | 44,8 | 123,2 | | | |
| Ведение соц. сетей | 44,8 | 123,2 | | | |
| Итого в спринте | 134,4 | 369,6 | 504 | | |

Ресурсный план проекта по алгоритмической торговле включает в себя определение необходимых ресурсов для эффективной реализации проекта.

- Необходимо определить бюджет проекта, который будет включать расходы на разработку и настройку алгоритмов, техническую поддержку, аналитику данных, обучение персонала и другие операционные расходы.
- Для проекта по алгоритмической торговле потребуются специалисты по программированию, аналитикам данных, трейдеры, технические специалисты и управляющие фондом. Важно правильно распределить задачи и обязанности между членами команды.
- Для разработки и поддержки алгоритмических систем может потребоваться специализированное оборудование, программное обеспечение и доступ к данным рынка.
- Определение временных рамок проекта, разработка графика работ и установление сроков для достижения ключевых этапов проекта.

- Доступ к актуальной информации о финансовых рынках, аналитические данные, новости и обзоры рынка, которые могут использоваться для принятия решений.

10 Риски проекта

Многие профессиональные участники фондового рынка отмечают проявления рисков, порождаемых алготорговлей, с которыми им приходится сталкиваться все чаще и чаще. Всего можно выделить несколько групп такого рода рисков.

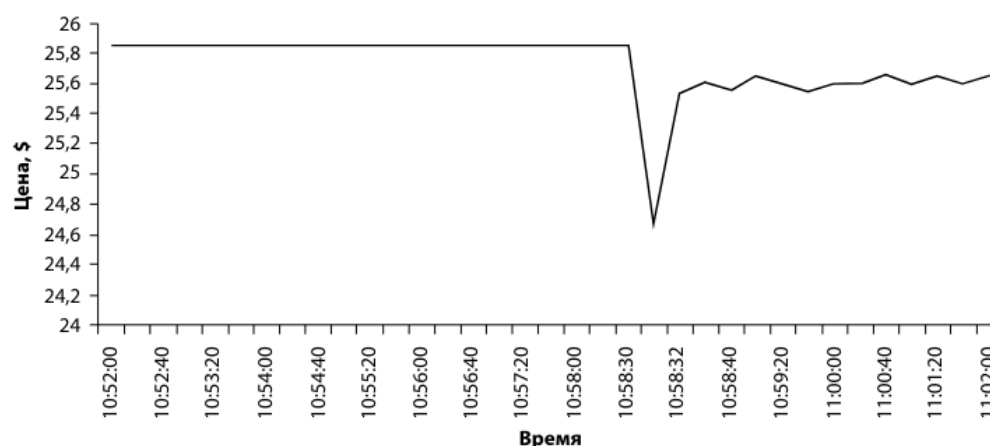
Операционные риски

На одно из первых мест по частоте проявления выходят случаи технологических сбоев. Алготрейдеры могут резко увеличивать поток заявок, и торговые системы не выдерживают чрезмерной нагрузки, что ведет к приостановке торгов и потерям для их участников. Так произошло во время IPO компании Facebook: компьютерный софт оказался фактически неспособен контролировать высокий темп подачи и снятия заявок алготорговцев. Как итог инвесторы получили убытки, т.к. их ордера не были исполнены своевременно.

Риск резкого увеличения рыночной волатильности

Сегодня при анализе рисков алгоритмической торговли особое внимание уделяется аномальному поведению рынка — так называемым флешкрэшам, или резким скачкам цен на финансовые инструменты, не обоснованным никакими экономическими факторами. Свое название они получили от широко известного события, произошедшего 6 мая 2010 г., когда цены многих акций, торгуемых на рынке США, продемонстрировали необыкновенно быстрое падение и восстановление. Другой пример — ситуация, произошедшая 16 июня того же года, когда цена на акции американской корпорации The Charles Schwab внезапно упала на 4,6% (рис. 2).

Рис. 2. Резкое изменение цены акций компании The Charles Schwab Corporation 16 июня 2014 г.



Риски, связанные с рыночной ликвидностью

В то же время в условиях рыночной турбулентности, которая может возникать как из-за алготорговцев, так и без их участия, создается дополнительная опасность резкого ухода ликвидности. При стрессовом движении рынка алгоритмические трейдеры могут прекратить осуществление транзакций, что приведет к масштабному оттоку ликвидности. Как показывает статистика, доля операций, особенно торговых заявок роботов, крайне высока, поэтому их уход с рынка может мгновенно обрушить котировки. Так, например, согласно статистике, публикуемой Московской биржей, доля роботов в заявках на основном сегменте в 2013 г. составляла около 97%. При этом они формировали и немалую часть рыночных оборотов — около 40%.

Риск роста издержек

Рост числа участников торгов, использующих алгоритмические системы при осуществлении транзакций, вместе с увеличением сложности и быстродействия программируемых алгоритмов повышает издержки для торговых площадок и регулирующих органов.

Риски намеренного использования роботов для неправомерных действий и манипулирования ценами

Еще одним риском применения алгоритмических систем является то, что их могут намеренно использовать для воздействия на рыночные цены по конкретным активам, причем как для получения выгод самими алготорговцами, так и по каким-то другим причинам

Матрица рисков.

| Матрица вероятности и последствий | | | | | | | | | | |
|-----------------------------------|--------|------|------|------|------|---------------------------|------|------|------|------|
| Вероятность | Угрозы | | | | | Благоприятные возможности | | | | |
| 0,90 | 0,05 | 0,09 | 0,18 | 0,36 | 0,72 | 0,72 | 0,36 | 0,18 | 0,09 | 0,05 |
| 0,70 | 0,04 | 0,07 | 0,14 | 0,28 | 0,56 | 0,56 | 0,28 | 0,14 | 0,07 | 0,04 |
| 0,50 | 0,03 | 0,05 | 0,10 | 0,20 | 0,40 | 0,40 | 0,20 | 0,10 | 0,05 | 0,03 |
| 0,30 | 0,02 | 0,03 | 0,06 | 0,12 | 0,24 | 0,24 | 0,12 | 0,06 | 0,03 | 0,02 |
| 0,10 | 0,01 | 0,01 | 0,02 | 0,04 | 0,08 | 0,08 | 0,04 | 0,02 | 0,01 | 0,01 |
| | 0,05 | 0,10 | 0,20 | 0,40 | 0,80 | 0,80 | 0,40 | 0,20 | 0,10 | 0,05 |

Воздействие (по относительной шкале) на цель (например, стоимость, сроки, содержание или качество)

Каждому риску присваивается показатель (ранг) на основании вероятности его появления и воздействия на цель проекта в случае его возникновения. На матрице показаны принятые в организации пороги для низких, умеренных и высоких рисков, которые определяют, будет ли риск считаться высоким, умеренным или низким для данной цели.

Качественный анализ рисков

1 - Высокий

2 - Средний

3 - Низкий

| Риск | Вероятность | Влияние | Значимость |
|---------------------|-------------|---------|------------|
| Технический сбой ПО | 3 | 1 | 1 |
| Аномальное | 1 | 2 | 1 |

| | | | |
|---|---|---|---|
| поведение на рынке ценных бумаг | | | |
| Нехватка ликвидности на рынке | 2 | 1 | 2 |
| Рост издержек для биржи | 2 | 3 | 2 |
| Использование для незаконных действий | 3 | 3 | 1 |
| Изменение правил использования ПО | 3 | 1 | 1 |



На рисунке представлена модель управления рисками, состоящая из следующих этапов:

1. Обнаружение рисков:

Первый этап заключается в идентификации и определении потенциальных рисков, которые могут возникнуть в деятельности организации.

2. Анализ рисков:

После обнаружения рисков, необходимо провести их анализ. Это включает в себя оценку вероятности возникновения каждого риска и его потенциальных последствий.

3. Планирование ответных действий: На этом этапе разрабатываются стратегии и действия для минимизации или предотвращения рисков.

Это может включать в себя разработку планов по управлению рисками, создание резервных копий данных, повышение осведомленности о рисках и т.д.

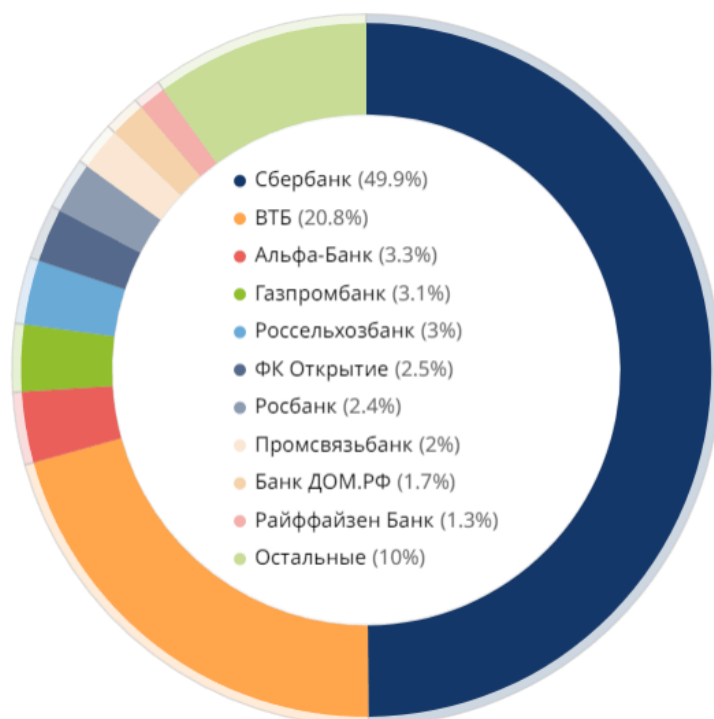
4. **Внедрение ответных действий:** После планирования, необходимо реализовать разработанные меры.
5. **Наблюдение, измерение и контроль:** На этом этапе следует отслеживать эффективность внедренных мер. Важно мониторить ситуацию и при необходимости корректировать планы и действия.

11 SWOT - анализ проекта

| | |
|---|--|
| <p>Сильные стороны (Strengths):</p> <p>Автоматизированность процесса принятия решений</p> <p>Более быстрая скорость операций</p> <p>Обработка большого объема данных</p> <p>Более высокая точность и объективность, чем у обычного трейдера</p> | <p>Слабые стороны (Weaknesses):</p> <p>Зависимость от качества данных и алгоритмов</p> <p>Риск неисправностей и ошибок в алгоритмах</p> <p>Низкая гибкость и сложность адаптации к неожиданным изменениям на рынке</p> <p>Требование к высококвалифицированным специалистам для разработки и поддержки, и настройки алгоритмов</p> |
| <p>Возможности (Opportunities):</p> <p>Возможность разработки более совершенных алгоритмов и технологий</p> <p>Возможность внедрение в разные виды рынков</p> <p>Возможность использования алгоритмов в других странах</p> | <p>Угрозы (Threats):</p> <p>Возможность мошенничества и манипуляций на рынке</p> <p>Риск регуляторных изменений, которые могут повлиять на эффективность</p> <p>Возрастающая конкуренция со стороны других организаций, использующих алгоритмическую торговлю</p> |

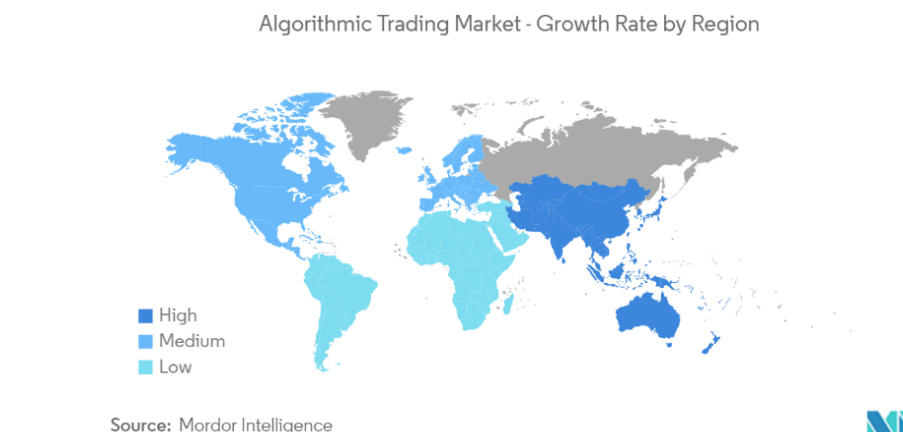
| | |
|--|--|
| | Возможность плохой работы в неожиданных рыночных условиях. |
|--|--|

Доли основных банков России на российском рынке.



12 Перспективы проекта

Количество используемых алгоритмического торгов по всему миру



Согласно рисунку видно, что в РФ широко не используется алгоритмическая торговля. Что дает большую возможность для реализации алгоритмической торговли.

Также в дальнейшей реализации проекта, можно будет использовать и на других финансовых рынках, таких как Соединенных Штатов Америки.

Расчет экономических показателей для проекта по алгоритмической торговле зависит от конкретных целей и задач проекта. В качестве экономических показателей можно использовать такие показатели, как:

- Возврат на инвестиции (ROI) = (Доход - Инвестиции) / Инвестиции
- Доходность (Return) = (Доход / Инвестиции) * 100%
- Вариация дохода (Volatility) = Стандартное отклонение доходов
- Коэффициент Шарпа (Sharpe Ratio) = (Средний доход - Средняя ставка безрискового вложения) / Стандартное отклонение доходов
- Коэффициент Sortino = (Средний доход - Минимально допустимый доход) / Стандартное отклонение доходов ниже минимально допустимого уровня
- Доля риска (Risk/Reward Ratio) = Риск / Награда

- Среднеквадратичное отклонение (Standard Deviation) = Квадратный корень из среднего арифметического квадратов отклонений значений показателя от его среднего значения
- Коэффициент drawdown = Максимальное значение drawdown за период наблюдения

| Показатель | Расчет | Значение |
|--|---|----------|
| Возврат на инвестиции (ROI) | $(76480000 - 100000000) / 100000000$ | 665% |
| Доходность (Return) | $(76480000 / 100000000) * 100\%$ | 764,8% |
| Вариация дохода (Volatility) | Стандартное отклонение доходов | 475,4% |
| Коэффициент Шарпа (Sharpe Ratio) | $((76480000 / 3) / 100000000 - 1\%) / 475,4\%$ | 0,053% |
| Коэффициент Sortino | $((76480000 / 3) / 100000000 - 0\%) / 475,4\%$ | 0,054% |
| Доля риска (Risk/Reward Ratio) | $-2660000 / 73820000$ | 0,373 |
| Среднеквадратичное отклонение (Standard Deviation) | Квадратный корень из среднего арифметического квадратов отклонений значений показателя от его среднего значения | 475,4% |
| Коэффициент drawdown | Максимальное значение drawdown за период наблюдения | -100% |

Для расчета этих показателей необходимо собирать данные о доходах и издержках проекта, а также о рыночных колебаниях, которые могут влиять на результаты проекта. Затем, используя соответствующие формулы, можно вычислить значения этих показателей.

Значения этих показателей можно использовать для оценки эффективности проекта по алгоритмической торговле, сравнения его результатов с другими проектами, а также для принятия решений о необходимости изменений в стратегии проекта.

13 Экономические показатели проекта

| Стоимость 1 стратегии | р.2 500,00 | | | | | | | | |
|--|--------------------|--------------------|---------------------|--------------------|--------------------|--------------------|------------------|-------------|-------------------|
| Показатели | 2024 | 2025 | 2026 | Отклонение, +,- | | | Темп роста, % | | |
| | | | | 2025-2024 | 2026-2025 | 2026-2024 | 2025-2024 | 2026-2025 | 2026-2024 |
| Выручка, руб. | р.500 000,00 | р.25 000 000,00 | р.100 000 000,00 | р.24 500 000,00 | р.75 000 000,00 | р.24 000 000,00 | 5 000,00 % | 4 00,00% | 2 0000,00 % |
| Себестоимость продаж, руб. | р.3 150000 | р.1 5120000 | р.2 3520000 | 1 197000 0 | 8 400000 | 8 820000 | 4 80,00% | 1 55,56% | 7 46,67% |
| Валовая прибыль, руб. | р.-2 650 000,00 | р.9 880 000,00 | р.76 480 000,00 | р.12 530 000,00 | р.66 600 000,00 | р.15 180 000,00 | - 372,83 % | 7 74,09% | - 2886,04 % |
| Уровень валовой прибыли к выручке, % | - 530,00 % | 3 9,52% | 7 6,48% | 5 69,52% | 3 6,96% | 1 099,52 % | - 7,46% | 1 93,52% | - 14,43% |
| Среднегодовая стоимость основных средств, руб. | р.- 220 833,33 | р.823 333,33 | р.6 373 333,33 | р.1 044 166,67 | р.5 550 000,00 | р.1 265 000,00 | - 372,83 % | 7 74,09% | - 2886,04 % |
| Фондоотдача, руб./руб. | 0 ,083 | 0 ,083 | 0 ,083 | 0 ,000 | 0 ,000 | - 0,083 | 1 00,00% | 1 00,00% | 1 00,00% |
| Рентабельность основных фондов, % | - 84,13% | 6 5,34% | 3 25,17% | 1 49,47% | 2 59,83% | 2 33,60% | - 77,67% | 4 97,63% | - 386,52 % |

| | | | | | | | | | |
|--|------------------|----------------|-----------------|----------------|-----------------|----------------|------------|------------|-------------|
| Среднегодовая стоимость оборотных средств, руб. | - р.2 650 000,00 | р.3 615 000,00 | р.43 180 000,00 | р.6 265 000,00 | р.39 565 000,00 | р.8 915 000,00 | - 136,42 % | 1 194,47 % | - 1629,43 % |
| Время обращения оборотных средств, дни | 219 | 365 | 365 | 146 | 0 | - 73 | 1 66,67% | 1 00,00% | 1 66,67% |
| Скорость обращения оборотных средств, число оборотов | - 0,19 | 2,53 | 1,31 | 2,72 | - 1,22 | 2,91 | 1341,09 % | 5 1,67% | 692,99 % |
| Среднесписочная численность работников, чел. | 7 | 10 | 20 | 33 | 10 | - 4 | 1 42,86% | 2 00,00% | 2 85,71% |
| Расходы на оплату труда, тыс.руб. | р.1 050 000,00 | р.1 260 000,00 | р.1 960 000,00 | р.210 000,00 | р.700 000,00 | р.840 000,00 | 1 20,00% | 1 55,56% | 1 86,67% |
| Среднемесячная заработная плата 1 работника, руб. | р.150 000,00 | р.126 000,00 | р.98 000,00 | - р.24 000,00 | - р.28 000,00 | - р.174 000,00 | 8 4,00% | 7 7,78% | 6 5,33% |
| Производительность труда, руб./чел. | р.852,27 | р.715,91 | р.556,82 | - р.136,36 | - р.159,09 | - р.988,64 | 8 4,00% | 7 7,78% | 6 5,33% |
| Издержки | р.10 | р.500 | р.2 000 | р.490 | р.1 500 | р.480 | 5 000,00 | 4 00,00% | 2 0000,00 |

| | | | | | | | | | |
|--|------------------------|-----------------------|------------------------|------------------------|------------------------|------------------------|------------------|--------------------------|-------------------|
| обращения, тыс.руб. | 000,00 | 000,00 | 000,00 | 000,00 | 000,00 | 000,00 | % | | % |
| Уров ень издержек обращения, % | 2 ,00% | 2 ,00% | 2 ,00% | 0 ,00% | 0 ,00% | - 2,00% | 1 00,00% | 1 00,00% | 1 00,00% |
| Приб ыль от продаж, тыс.руб. | - р.2 660 000,00 | р .9 380 000,00 | р .74 480 000,00 | р .12 040 000,00 | р .65 100 000,00 | р .14 700 000,00 | - 352,63 % | - 7 2800,00 94,03% | - 2800,00 % |
| Рент абельность продаж, % | - 84,44% | 6 2,04% | 3 16,67% | 1 46,48% | 2 54,63% | 2 30,93% | - 73,46% | 5 10,45% | - 375,00 % |
| Проч ие доходы, тыс.руб. | 0 | 0 | 0 | 0 | 0 | 0 | 0 ,00% | 0 ,00% | 0 ,00% |
| Проч ие расходы, тыс.руб. | 0 | 0 | 0 | 0 | 0 | 0 | 0 ,00% | 0 ,00% | 0 ,00% |
| Приб ыль до налогооблож ения, тыс.руб. | - р.2 660 000,00 | р .9 380 000,00 | р .74 480 000,00 | р .12 040 000,00 | р .65 100 000,00 | р .14 700 000,00 | - 352,63 % | - 7 2800,00 94,03% | - 2800,00 % |
| Теку щий налог на прибыль, тыс.руб. | 4 7000 | р .1 354 000,00 | р .11 119 000,00 | р .1 307 000,00 | р .9 765 000,00 | р .1 260 000,00 | 0 ,00% | 8 21,20% | 0 ,00% |
| Чист ая прибыль, тыс.руб. | - р.2 707 000,00 | р .8 026 000,00 | р .63 361 000,00 | р .10 733 000,00 | р .55 335 000,00 | р .13 440 000,00 | - 296,49 % | - 7 2340,64 89,45% | - 2340,64 % |
| Рент абельность деятельности , % | 1 01,77% | 8 5,57% | 8 5,07% | - 16,20% | - 0,49% | - 117,97 % | - 8 4,08% | - 9 9,42% | - 8 3,59% |

1. Выручка, руб. - суммарная стоимость продаж продуктов или услуг за определенный период времени.

2. Себестоимость продаж, руб. - суммарные затраты на производство и реализацию продуктов или услуг за определенный период времени.
3. Валовая прибыль, руб. - разница между выручкой и себестоимостью продаж.
4. Уровень валовой прибыли к выручке, % - отношение валовой прибыли к выручке, выраженное в процентах.
5. Среднегодовая стоимость основных средств, руб. - средняя стоимость всех недвижимых и движимых активов, используемых в бизнесе, за год.
6. Фондоотдача, руб./руб. - отношение чистой прибыли к среднегодовой стоимости основных средств, выраженное в рублях за рубль.
7. Рентабельность основных фондов, % - отношение чистой прибыли к среднегодовой стоимости основных средств, выраженное в процентах.
8. Среднегодовая стоимость оборотных средств, руб. - средняя стоимость всех оборотных активов, таких как запасы и дебиторская задолженность, за год.
9. Время обращения оборотных средств, дни - среднее время, за которое оборотные средства полностью оборачиваются.
10. Скорость обращения оборотных средств, число оборотов - количество полных оборотов оборотных средств за определенный период времени.
11. Среднесписочная численность работников, чел. - среднее количество сотрудников, работающих в компании за определенный период времени.
12. Расходы на оплату труда, руб. - суммарные затраты на заработную плату сотрудников за определенный период времени.

13. Среднемесячная заработная плата 1 работника, руб. - средняя заработная плата одного сотрудника за месяц.
14. Производительность труда, руб./чел. - отношение выручки к среднесписочной численности работников, выраженное в рублях за одного сотрудника.
15. Издержки обращения, руб. - суммарные затраты на продажи и маркетинг за определенный период времени.
16. Уровень издержек обращения, % - отношение издержек обращения к выручке, выраженное в процентах.
17. Прибыль от продаж, руб. - разница между выручкой и издержками обращения.
18. Рентабельность продаж, % - отношение прибыли от продаж к выручке, выраженное в процентах.
19. Прочие доходы, руб. - любые другие доходы, полученные компанией за определенный период времени.
20. Прочие расходы, руб. - любые другие расходы, понесенные компанией за определенный период времени.
21. Прибыль до налогообложения, руб. - разница между прибылью от продаж и прочими доходами.
22. Текущий налог на прибыль, руб. - сумма налога, взимаемого на прибыль от продаж за определенный период времени.
23. Чистая прибыль, руб. - разница между прибылью до налогообложения и текущим налогом на прибыль.

Используемые источники

- Обзор ключевых показателей брокеров ЦБ РФ
https://cbr.ru/Collection/Collection/File/48976/review_broker_Q4_2023.pdf
- Обзор ключевых показателей на мировом рынке
<https://www.mordorintelligence.com/ru/industry-reports/algorithmic-trading-market>
- Информация о рисках - <https://infostart.ru/pm/1100990/>

Приложения

Приложение 1:

```
import numpy as np

class ARCHModel():
    """
    Модель ARCH (авторегрессивная условно гетероскедастическая модель)
    представляет собой модель дисперсии временного ряда.
    Модели ARCH используются для описания изменяющейся, возможно,
    изменчивой дисперсии. Хотя модель ARCH можно использовать для
    описания постепенного увеличения дисперсии с течением времени,
    чаще всего она используется в ситуациях, когда могут быть короткие
    периоды увеличения дисперсии. (Постепенно увеличивающуюся дисперсию,
    связанную с постепенно увеличивающимся средним уровнем,
    лучше обрабатывать путем преобразования переменной.)
    """
    def __init__(self, data, p, o, q, dist='normal'):
        # data - входные данные
        self.data = data
        # n_obs - количество наблюдений
        self.n_obs = len(data)
        # p - порядок авторегрессии
        self.p = p
        # o - порядок скользящей средней
        self.o = o
        # q - порядок авторегрессии в квадрате остатков
        self.q = q
        # dist - тип распределения ошибок (например, нормальное)
        self.dist = dist
        # params - оценки параметров модели
        self.params = None
        # arch_err - остатки модели ARCH
        self.arch_err = None
        # sigma2 - дисперсия модели ARCH
        self.sigma2 = None
        # forecast - прогнозы модели ARCH
        self.forecast = None

    def initialize(self):
        self.arch_err = np.zeros(self.n_obs)
        self.sigma2 = np.zeros(self.n_obs)
        self.sigma2[0] = self.data.var()

    def residuals(self):
        """Метод для расчета остатков модели ARCH"""
```

```

self.initialize()
self.arch_err[0] = self.data[0]

for t in range(1, self.n_obs):
    data_slice = self.data[t - np.arange(1, self.p + 2)]
    params_slice = self.params[:self.p + 1]
    self.arch_err[t] = self.data[t] - np.sum(params_slice *
data_slice)

def sigma2_update(self, t):
    """Метод для обновления дисперсии модели ARCH,
    t - текущий момент времени"""
    params_slice = self.params[self.p + 1:self.p + 1 + self.q]
    err_slice = self.arch_err[t - np.arange(1, self.q + 1)]
    self.sigma2[t] = np.sum(params_slice * err_slice) +
self.params[-1].item()

def fit(self, update_freq=1, disp='off'):
    """Метод для оценки параметров модели ARCH,
    update_freq - частота обновления параметров
    disp - флаг для вывода информации о процессе оценки"""
    self.initialize()
    self.params = np.random.rand(self.p + 1 + self.q + 1).reshape(-1, 1)

    for t in range(self.p + 1, self.n_obs):
        if t % update_freq == 0:
            print(f'Iteration {t}')

            self.sigma2_update(t)

            if self.dist == 'normal':
                self.params[0] += (
                    self.data[t - 1] * self.arch_err[t - 1] / self.sigma2[t]
                )
                self.params[1:self.p + 1] += (
                    self.data[t - np.arange(1, self.p + 1)] *
                    self.arch_err[t - np.arange(1, self.p + 1)] /
self.sigma2[t]
                )
                self.params[self.p + 1:self.p + 1 + self.q] += (
                    self.arch_err[t - np.arange(1, self.q + 1)] *
                    self.arch_err[t - np.arange(1, self.q + 1)] /
self.sigma2[t]
                )
                self.params[-1] += self.sigma2[t] / self.n_obs
            else:
                raise NotImplementedError

```



```

def forecast(self, h):
    """Метод для прогнозирования дисперсии модели ARCH,
    h - горизонт прогнозирования"""
    if not isinstance(h, int) or h <= 0:
        raise ValueError("h должен быть положительным целочисленным
числом!")

    forecast_values = np.zeros(h)
    forecast_values[0] = self.sigma2[-1]

    for t in range(1, h):
        if t <= self.q:
            params_slice = self.params[self.p + 1:self.p + 1 + self.q]
            err_slice = self.arch_err[-np.arange(1, t + 1)].reshape(-1,
1)

            forecast_values[t] = (
                np.sum(params_slice * err_slice, axis=0)[0] +
self.params[-1][0]
            )
        else:
            params_slice = self.params[self.p + 1:self.p + 1 + self.q]
            forecast_slice = forecast_values[
                t - np.arange(1, self.q + 1)].reshape(-1, 1)
            forecast_values[t] = (
                np.sum(params_slice * forecast_slice, axis=0)[0] +
self.params[-1][0]
            )

    return forecast_values

```

Приложение 2:

```

import numpy as np
try:
    from .arch import ARCHModel
except ImportError:
    from arch import ARCHModel

class GARCHModel(ARCHModel):
    def __init__(self, data, p, o, q, dist='normal'):
        super().__init__(data, p, o, q, dist)

    def initialize(self):
        self.arch_err = np.zeros(self.n_obs)
        self.sigma2 = np.zeros(self.n_obs, dtype=np.float64)

```

```

self.sigma2[0] = self.data.var()

def sigma2_update(self, t):
    if not isinstance(t, (int, float)):
        raise ValueError("epsilon must be a number")
    if np.isnan(t):
        raise ValueError("epsilon cannot be NaN")
    params_slice = self.params[self.p + 1:self.p + 1 + self.q]
    err_slice = self.arch_err[t - np.arange(1, self.q + 1)]
    sigma2_slice = np.clip(
        self.sigma2[t - np.arange(1, self.p + 1)], 1e-10, 1e10)
    self.sigma2[t] = (
        np.sum(params_slice * err_slice ** 2) +
        np.sum(self.params[self.p + 1 + self.q:] * sigma2_slice) +
        self.params[-1][0]
    )

def fit(self, update_freq=1, disp='off'):
    if disp not in ['off', 'on']:
        raise ValueError("Invalid disp value. Must be 'off' or 'on'.")
    self.initialize()
    self.params = np.random.rand(self.p + 1 + self.q + self.p +
1).reshape(-1, 1)

    for t in range(self.p + 1, self.n_obs):
        if t % update_freq == 0:
            print(f'Iteration {t}')

        self.sigma2_update(t)

        if self.dist == 'normal':
            self.params[0] += (
                self.data[t - 1] * self.arch_err[t - 1] / self.sigma2[t]
            )
            self.params[1:self.p + 1] += (
                self.data[t - np.arange(1, self.p + 1)] *
                self.arch_err[t - np.arange(1, self.p + 1)] /
self.sigma2[t]
            )
            self.params[self.p + 1:self.p + 1 + self.q] += (
                self.arch_err[t - np.arange(1, self.q + 1)] *
                self.arch_err[t - np.arange(1, self.q + 1)] /
self.sigma2[t]
            )
            self.params[self.p + 1 + self.q:] += (
                self.sigma2[t - np.arange(1, self.p + 1)] *
                self.sigma2[t - np.arange(1, self.p + 1)] /
self.sigma2[t]

```

```

        )
        self.params[-1] += self.sigma2[t] / self.n_obs
    else:
        raise NotImplementedError

    def forecast(self, h):
        if not isinstance(h, int) or h <= 0:
            raise ValueError("h должен быть положительным целочисленным
числом!")

        forecast_values = np.zeros(h)
        forecast_values[0] = self.sigma2[-1]

        for t in range(1, h):
            if t <= self.q:
                params_slice = self.params[self.p + 1:self.p + 1 + self.q]
                err_slice = self.arch_err[-np.arange(1, t + 1)].reshape(-1,
1)

                sigma2_slice = self.sigma2[-np.arange(1, t + 1)].reshape(-1,
1)

                forecast_values[t] = (
                    np.sum(params_slice * err_slice ** 2, axis=0)[0] +
                    np.sum(self.params[self.p + 1 + self.q:] * sigma2_slice,
axis=0)[0] +
                    self.params[-1][0]
                )
            else:
                params_slice = self.params[self.p + 1:self.p + 1 + self.q]
                forecast_slice = forecast_values[
                    t - np.arange(1, self.q + 1)].reshape(-1, 1)
                sigma2_slice = forecast_values[
                    t - np.arange(1, self.p + 1)].reshape(-1, 1)
                forecast_values[t] = (
                    np.sum(params_slice * np.clip(forecast_slice ** 2, 0,
1e10), axis=0)[0] +
                    np.sum(
                        self.params[self.p + 1 + self.q:] *
np.clip(sigma2_slice, 0, 1e10),
axis=0)[0] +
                    self.params[-1][0]
                )

        return forecast_values

```