МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра теории вероятностей и математической статистики

ПОЛУЗЁРОВ Тимофей Дмитриевич

МОДЕЛИ ДОХОДНОСТЕЙ АКТИВОВ В СРЕДНЕ-ДИСПЕРСИОННОМ АНАЛИЗЕ МАРКОВИЦА НА КРИПТОВАЛЮТНЫХ РЫНКАХ

Магистерская диссертация
специальность 1-31 80 09 «Прикладная математика и информатика»

Научный руководитель
Харин Алексей Юрьевич
доктор физико-математических наук,
профессор

Допущена к защите
«___» _______________ 2025 г.
Зав. кафедрой теории вероятностей и математической статистики
____________ А. Ю. Харин
доктор физико-математических наук, профессор

Минск, 2025

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Ключевые слова: ПОРТФЕЛЬНАЯ ТЕОРИЯ, ИНВЕСТИЦИИ, АКТИВЫ, ВАЛЮТЫ, КРИПТОВАЛЮТЫ, СРЕДНЕ-ДИСПЕРИСИОННЫЙ АНАЛИЗ, ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ, ДОХОДНОСТЬ, АВТОРЕГРЕССИЯ, МАШИННОЕ ОБУЧНИЕ, БИРЖА.

Цель работы: исследовать на реальных данных эффективность методов оценки средней ожидаемой доходности в портфельной теории Марковица.

Объект исследования: методы прогнозирования средней доходности, портфельная теория.

Предмет исследования: эффективность методов оценки средней доходности и оценка доходностей соотвествующих портфелей.

Методы исследования: методы теории вероятностей, математической статистики и временных рядов, методы регрессионного анализа, методы машинного обучения.

Результаты работы: предложены методы оценки средних доходностей активов. На реальных данных исследованы доходности соотвествующих портфелей. Выполнена программная реализаци алгоритмов по определению оптимальных портфелей и оценка их доходностей.

Области применения: фондовые, валютные, криптовалютные биржи. Инвестиционные проекты, страхование.

Структура магистерской диссертации: работа изложена на 43 страницах, состоит из общей характеристики на 3 языках, введения, 3 глав, заключения, списка использованных источников и приложения. Содержит 10 рисунков, 5 таблиц и 1 приложение.

АГУЛЬНАЯ ХАРАКТЫРЫСТЫКА РАБОТЫ

Ключавыя словы: ПАРТФЕЛЬНАЯ ТЭОРЫЯ, ІНВЕСТЫЦЫІ, АКТЫВЫ, ВАЛЮТЫ, КРЫПТАВАЛЮТЫ, СЯРЭДНЕ-ДЫСПЕРСІЁННЫ АНАЛІЗ,ПРАГНАЗІРАВАННЕ ЧАСОВЫХ ШЭРАГАУ, ДАХОДНАСЦЬ, АУТАРЭГРЭСІЯ, МАШЫННАЕ НАВУЧАННЕ, БІРЖА.

Мэта работы: даследаваць на рэальных дадзеных эфектыўнасць метадаў ацэнкі сярэдняй чаканай даходнасці ў партфельнай тэорыі Маркавіца.

Аб'екта даследавання: метады прагназавання сярэдняй даходнасці, партфельная тэорыя.

Прадмет даследавання: эфектыўнасць метадаў ацэнкі сярэдняй даходнасці і ацэнкі даходнасці адпаведных партфеляў.

Метады даследавання: метады тэорыі верагоднасцей, матэматычнай статыстыкі і часовых шрагау, метады рэгрэсійнага аналізу, метады машыннага навучання.

Вынікі работы: прапанаваныя метады ацэнкі сярэдніх даходаў актываў. Па рэальных дадзеных даследаваны даходнасці адпаведных партфеляў. Выканана праграмная рэалізацыя алгарытмаў па вызначэнні аптымальных партфеляў і ацэнцы іх даходаў.

Вобласть ўжывання: фондавыя, валютныя, криптовалютные біржы. Інвестыцыйныя праекты, страхаванне.

Структура магістэрскай дысертацыі: праца выкладзена на 43 старонках, складаецца з агульных характарыстык на 3 мовах, увядзенні, 3 главы, заключэнні, спісы выкарыстаных крыніц і дадаткаў. Змяшчае 10 малюнкаў, 5 табліцу і 1 дадатак.

GENERAL DESCRIPTION OF WORK

Keywords: PORTFOLIO THEORY, INVESTMENTS, ASSETS, CURRENCIES, CRYPTOCURRENCIES, MEAN-VARIANCE ANALYSIS, TIME SERIES FORECASTING, RETURN, AUTOREGRESSION, MACHINE LEARNING, STOCK EXCHANGE.

The aim: to investigate the effectiveness of methods for estimating the average expected return in Markowitz's portfolio theory on real data.

The object: methods for forecasting average returns, portfolio theory.

Research methods: methods of probability theory, mathematical statistics and time series, methods of regression analysis, methods of machine learning.

The results: Methods for estimating average assets returns are proposed. The returns of the corresponding portfolios are studied using real data. A software implementation of algorithms for determining optimal portfolios and estimating their returns is completed.

Application: stock, currency, cryptocurrency exchanges. Investment projects, insurance.

Structure of a Master's Thesis: the work is presented on 43 pages, consists of a general description in 3 languages, an introduction, 3 chapters, a conclusion, a list of references and an appendix. Contains 10 figures, 5 tables and 1 appendix.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Портфельная теория Марковица дает точный ответ на вопрос выбора портфеля, если известны математическое ожидание и ковариации случайных величин, описывающих доходности ативов за будущий период инвестирования. На практике, конечно, эти значения неизвестны и их приходится их оценивать (прогнозировать). В финансовой математике уже были разработаны модели оценки характеристик будущих доходностей.

Модель CAPM (Capital Asset Pricing Model), разработанная У.Шарпом и Дж.Линтером [13], [9], которая базируется на концепции равновесного рынка, моделирует линейную зависимость между доходностями активов и «большим рынком», например, рыночным индексом.

Более современная теория — теория APT(Arbitrage Pricing Theory) С. Росса и Р.Ролла [12], [11], исходящая из многофакторной модели зависимости доходности активовот некоторых факторов (необязательно рыночных). Эта теория опирается на концепцию отсутсвия асимптотического арбитража.

В этой работе приводится альтернативный подход к оценке нужных характеристик. Имея в распоряжении историю наблюдений за ценами активов можно поставить задачу спрогнозировать будущие цены на основании предыдущей динамики. Для решения этой задачи можно применить современные методы прогнозирования временных рядов. К таким методам относятся: классические статистические модели, модели машинного обучения, модели временных рядов.

Идеи портфельной теории можно применить не только к выбору портфеля акций. Например, когда отдельный человек, желая сохранить свой капитал, принимает решение о том как разместить деньги по депозитам, купить валюту, страховку и так далее. А учитывася растущую популярность криптовалют в последнее время, особенный интерес представляет применение теории в этой области.

Криптовалюта — это альтернативный вид валюты в цифровой или виртуальной форме, для защиты транзакций используется криптография. В 2009 году был создан первый криптовалютный токен — Bitcoin. Принципы его работы были описаны в статье [10]. Далее выпускались и дргуе токены, напрмер Etherium, Litecoin. Торговля токенами ведется в интернете на специализированных биржах. Криптовалютные биржи устроены по тем же принципам что и фондовые и валютные биржри. Поведение цен на криптовалюты структурно отличается от поведения цен на обыкновеннные акции. Цены криптовалют склонны к очень резким скачкам и как правило в них отсутсвует долгосрочный тренд. Динамика цен более волатильная, причем волатильность не постонянна во времени. Цифровизация торговли позвляет иметь моентальный доступ к котировкам и вести активную торговлю. Это дает возможность трейдерам заниматься спекуляциями, а инвесторам — возможность среднесрочного и долгосрочного инвестирования.

В этой работе усовершествуется подхода Марковица путем использования продвинутых методов прогнозирования временных рядов для оценки будущих доходностей. Теоретические модели адаптируются для формирования однопериодных торговых стратегий на криптовалютном рынке. Доходность полученных стратегий проверяется на реальных данных. Также рассматривается вопрос диверсификации портфеля и контроля риска.

1. СРЕДНЕ-ДИСПЕРСИОННЫЙ АНАЛИЗ ПОРТФЕЛЯ

В этой главе определяеются основные понятия инвестирования и средне-дисперсионного анализа. Формулируется задача поиска оптимального портфеля.

1.1. Основные понятия

Портфельный анализ берет свое начало с выхода статьи Гарри Марковица в 1952 г [8]. Подход Марковица начинается с предположения что инвестор в настоящий момент времени имеет конкретную сумму денег для инвестирования. Эти деньги будут инвестированы на определенный промежуток времени, который назывется **периодом инвестирования**. В конце периода инвестор продает активы купленные в ранее. Набор приобретенных активов иначе называют **инвестиционным портфелем**. Поэтому проблема выбора и распределения средств по активам имеет название **проблемой выбора инвестиционного портфеля**.

Пусть цены актива на начало и конец периода инвестирования равны S^0 и S^1 соотвественно. Определим **доходность актива** (Return) r за период инвестирования как

$$r = \frac{S^1 - S^0}{S^0}$$

При формировании портфеля в начальный момент времени, инвестор должен иметь в виду, что доходность активов за будущий период владения заранее не известна. То есть, он вынужден принимает решение о выборе портфеля исходя из своих ожидаемых доходностей активов.

Если инвестор ставит задачей максимизировать доходность портфеля, то в этом случае его портфель должен состоять из единственного актива с наибольшей ожидаемой доходностью. Марковиц отмечает, что такой подход не является разумны, потому что типичный инвестор хоть и желает чтобы «доходность была высокой», но одновременно требует чтобы «доходность была настолько определенной насколько это возможно». Это означает, что инвестор, стремясь одновременно максимизировать доходность и минимизировать риск (неопределенность), имеет две противоречащие друг другу цели. Подход Марковиа к принятию решения дает возможность адекватно учесть обе эти цели.

Имея N доступных активов можно сформировать бесконечно много портфелей. Это множество называют **достижимым**. Как инвестору в этих условиях выбрать портфель? Логичными являются следующие принципы при формировании портфеля:

- Из двух портфелей с одинаковым риском, инвестор выберет порфтель с большей ожидаемой доходностью
- Из двух портфелей с одинаковой доходностью, инвестор выберет портфель с меньшим риском

Другими словами, из достижимого множества портфелей инвестор склонен выбирать парето-оптимальные портфели. Множество оптимальных портфелей иначе называют **эффективным множеством**. Достижимые портфели не из эффективного множества называют **неэффективными портфелями**.

Вопрос выбора конкретного портфеля из эффективного множества остается на стороне инвестора. Здесь он уже руководствуется своей внутренней толерантностю к риску. Обычно достаточно зафиксировать приемлимый уровень риска внутри достижимого множества и выбрать портфель с соответсвующей доходностью.

1.2. Постановка задачи поиска оптимального портфеля

Будем рассматривать одношаговую задачу инвестирования. Инвестор собирает портфель по рыночным ценам активов S^0 стоимостью x в момент времени $n = 0$, а момент времени $n = 1$ этот портфель продается по рыночным ценам S^1.

Пусть инветору доступно инвестирование в N активов и начальный капитал x. Цены активов в начальный момент времени $n = 0$ равны $S_1^0, \ldots, S_N^0$

Обзначим

$$b = (b_1, \ldots, b_N), b_i \geq 0 \tag{1.1}$$

$i = \overline{1, N}$ число активов которые преобрел инвестор.

Тогда стоимость портфеля в начальный момент времени равна

$$X^0 = b_1 S_1^0 + \cdots + B_N S_N^0 \tag{1.2}$$

Иначе говоря, b есть портфель ценных бумаг, где b_i – число i-х акций, приобретенных по цене S_i^0.

Будущие цены акций в момент времени $n = 1$ равны $S_1^1, \ldots, S_N^1$. Их можно представить в терминах доходностей r_i используя начальные цены

$$S_i^1 = (1 + r_i) S_i^0, i = \overline{1, N} \tag{1.3}$$

Здесь r_i являются случайными величинами.

Если инвестор сформировал портфель $b = (b_1, \ldots, b_N)$, то его начальный капитал $X^0 = x$ превратится в

$$X^1 = b_1 S_1^1 + \cdots + b_N S_N^1, \tag{1.4}$$

Таким образом, стоимость портфеля на конец периода инвестирования является случайной величиной. Она определяется набором случайных величин — будущими доходностями активов, и тем как инвестор распределил капитал по доступным актвам. Если на первое повлиять невозможно, то второе полностью определяется инвестором. В его интересах собрать такой портфель, цена которого будет «побольше» и с высокой уверенностью. Это стремление максимизировать прибыть и минимизировать риск (неопределенность), Марковиц формулирует в терминах математического ожидания $\mathbb{E}\left[X^1\right]$ и дисперсии $\mathbb{D}\left[X^1\right]$ случайной величины X^1.

Имея эти две характеристики, можно по-разному формулировать оптимизационную задачу выбора наилучшего портфеля в зависимости от критерия оптимальности.

Можно, например, задаться вопросом о том, на каком портфеле b^* достигается максимум некоторой целевой функции $f = f(\mathbb{E}\left[X^1\right], \mathbb{D}\left[X^1\right])$ при «бюджетном ограничении» на класс допустимых портфелей:

$$B(x) = \{b = (b_1, \ldots, b_N) : b_i \geq 0, X_0(b) = x\}, x > 0 \tag{1.5}$$

Задача, сформулированная в этом разделе, допускает записи в более удобном виде. А именно, перейти от абсолютных цен к доходностям.

1.3. Сведение к доходностям

Сформулированная задача позволяет рабоать не с будущими ценами активов $S_1^1, \ldots, S_N^1$, а с доходностями $r_1, \ldots, r_N$.

Перейдем от величин $b = (b_1, \ldots, b_N)$ к величинам $\omega = (\omega_1, \ldots, \omega_N)$, которые определим как

$$\omega_i = \frac{b_i S_i^0}{x} \tag{1.6}$$

причем $\omega_i \geq 0, i = \overline{1, N}$ и

$$\sum_{i=1}^{N} \omega_i = \sum_{i=1}^{N} \frac{b_i S_i^0}{x} = \frac{1}{x} \sum_{i=1}^{N} b_i S_i^0 = \frac{X^0}{x} = 1 \tag{1.7}$$

Таким образом, ω есть не что иное как доли капитала, инвестируемые в соответсвующие активы.

Рассмотрим цену портфеля в момент времени $n = 1$. Доходность всего портфеля обозначим через R. Тогда

$$X^1 = (1 + R) X^0 \tag{1.8}$$

$$R = \frac{X^1}{X^0} - 1 = \frac{X^1}{x} - 1 = \left(\frac{\sum_{i=1}^{N} b_i S_i^0}{x} \right) - 1 = \left(\sum_{i=1}^{N} \omega_i \frac{S_i^1}{S_i^0} \right) - 1 = \tag{1.9}$$

$$= \left(\sum_{i=1}^{N} \omega_i \frac{S_i^1}{S_i^0} \right) - \sum_{i=1}^{N} \omega_i = \sum_{i=1}^{N} \omega_i \frac{S_i^1}{S_i^0} - \omega_i = \sum_{i=1}^{N} \omega_i \left(\frac{S_i^1}{S_i^0} - 1 \right) = \tag{1.10}$$

$$= \sum_{i=1}^{N} \omega_i r_i \tag{1.11}$$

Доходность всего портфеля определяется как смесь случайных величин — доходностей активов входящих в портфель.

$$\mathbb{E}\left[R\right] = \mathbb{E}\left[\sum_{i=1}^{N} \omega_i r_i\right] = \sum_{i=1}^{N} \omega_i \mathbb{E}\left[r_i\right] \tag{1.12}$$

$$\mathbb{D}\left[R\right] = \mathbb{D}\left[\sum_{i=1}^{N} \omega_i r_i\right] = \sum_{i=1}^{N} \omega_i^2 \mathbb{D}\left[r_i\right] + \sum_{i=1, j=1, i \neq j}^{N} \omega_i \omega_j \mathbf{Cov}\left(r_i, r_j\right) \tag{1.13}$$

Уравнения 1.12 и 1.13 удобно переписать в матричном виде.

Пусть

$$r = (r_1, \ldots, r_N) \in \mathbb{R}^N \tag{1.14}$$

— вектор-столбец доходностей активов (случайный вектор).

Математическое ожидание доходностей

$$\mu = \mathbb{E}\left[r\right] = \left(\mathbb{E}\left[r_1\right], \ldots, \mathbb{E}\left[r_N\right]\right) \tag{1.15}$$

и матрица ковариаций

$$\Sigma = \{\sigma_{ij} = \mathbf{Cov}\left(r_i, r_j\right)\}_{i=1, j=1}^{N, N} \in \mathbb{R}^{N \times N} \tag{1.16}$$

распределение капитала по активам

$$\omega = (\omega_1, \ldots, \omega_N) \in \mathbb{R}^N \tag{1.17}$$

Доходность портфеля (случайная величина) есть

$$R = \omega^T r \tag{1.18}$$

Математическое ожидание доходности портфеля обозначим μ_X

$$\mu_X = \mathbb{E}\left[R\right] = \mathbb{E}\left[\omega^T r\right] = \omega^T \mu \tag{1.19}$$

а дисперсию σ_X^2

$$\sigma_X^2 = \mathbb{D}\left[R\right] = \mathbb{D}\left[\omega^T r\right] = \omega^T \Sigma \omega \tag{1.20}$$

Полученные матричные обозначения позволяют удобно сформулировать задачу оптимизации. Причем эту задачу можно ставить в различных постановках.

1.4. Оптимизационная задача

Предположим, что нам известны математические ожидания μ и ковариации Σ доходностей активов за период инвестирования. На практике, конечно же, эти величины не известны и их приходится оценивать. Но в этом пункте сфокусируемся на постановке и решении оптимизационной задачи при известных входных данных.

Для удобства матричных записей используется обозначение $e = (1, \cdots, 1) \in \mathbb{R}^N$ — единичный вектор-столбец.

В качестве целевой функции возьмем линейную комбинацию доходности и риска портфеля. Для этого введем риск-параметр τ.

$$f\left(\mu_X, \sigma_X^2\right) = \tau \omega^T \Sigma \omega - \omega^T \mu \to \min_{\omega} \tag{1.21}$$

Используя риск-параметр $\tau \in [0, +\infty]$, оптимизационную задачу можно сформулировать в следующем виде

$$\begin{cases} \tau \omega^T \Sigma \omega - \omega^T \mu \to \min_{\omega} \\ \omega^T e = 1 \\ \omega \geq 0 \end{cases} \tag{1.22}$$

Множество решений при различных значениях τ образуют эффективное множество портфелей. При $\tau = 0$ имеем портфель минимального риска.

Альтернативно можно записать оптимизационную задачу когда риск портфеля нужно зафиксировать на определенном уровне

$$\begin{cases} -\omega^T \mu \to \min_\omega \\ \omega^T \Sigma \omega \leq \sigma_X^2 \\ \omega^T e = 1 \\ \omega \geq 0 \end{cases} \tag{1.23}$$

Аналогично, если требуется зафиксировать определенную доходность

$$\begin{cases} \omega^T \Sigma \omega \to \min_\omega \\ \omega^T \mu \geq \mu_X \\ \omega^T e = 1 \\ \omega \geq 0 \end{cases} \tag{1.24}$$

Полученные задачи решаются с помощью хорошо изученных методов квадратичного программирования.

Конечно, на практике распределения будущих доходностей, или хотябы их характрестики неизвестны. Поэтому для применения портфельной теории требуется оценить среднее и коварицию будущих доходностей. На основани истории наблюдений за доходностями активов можно построить прогноз необходимых характеристик и решать оптимизационную задачу.

Еще одним важным понятием в портфельной теории является диверсификация. Прежде чем переходить к рассмотрению методов оценки будущих доходностей, рассмотрим как с помощью диверсификации можно редуцировать риск портфеля.

1.5. Диверсификация портфеля

Обратимся теперь к вопросу о том, как диверсификацией можно добится снижения риска, измеряемого дисперсией или стандарным отклонением величины R.

С этой целью рассмотрим для начала пару случайных величин ξ_1 и ξ_2 с конечными вторыми моментами. Тогда если c_1 и c_2 – константы, $\sigma_i =$

$\sqrt{\mathbb{D}\left[\xi_i\right]}, i=1,2$, то

$$\mathbb{D}\left[c_1\xi_1+c_2\xi_2\right]=(c_1\sigma_1-c_2\sigma_2)^2+2c_1c_2\sigma_1\sigma_2(1+\sigma_{12}), \tag{1.25}$$

где $\sigma_{12}=\frac{\mathbf{Cov}(\xi_1,\xi_2)}{\sigma_1\sigma_2}$, $\mathbf{Cov}\left(\xi_1,\xi_2\right)=\mathbb{E}\left[\xi_1\xi_2\right]-\mathbb{E}\left[\xi_1\right]\cdot\mathbb{E}\left[\xi_2\right]$. Отсюда ясно, что если $c_1\sigma_1=c_2\sigma_2$ и $\sigma_{12}=-1$, то $\mathbb{D}\left[c_1\xi_1+c_2\xi_2\right]=0$.

Таким образом, если величины ξ_1 и ξ_2 отрицательно коррелированы с коэффициентом корреляции $\sigma_{12}=-1$, то таким подбором констант c_1 и c_2, что $c_1\sigma_1=c_2\sigma_2$, получаем комбинацию $c_1\xi_1+c_2\xi_2$ с нулевой дисперсией. Но, конечно, при этом среднее значение $\mathbb{E}\left[c_1\xi_1+c_2\xi_2\right]$ может оказаться достаточно малым. (Случай $c_1=c_2=0$ для задачи оптимизации не интересен в силу условия $b\in B(X)$).

Из этих элементарных рассуждений ясно, что при заданных ограничениях на (c_1,c_2) и класс величин (ξ_1,ξ_2) при решении задачи о том, чтобы сделать $\mathbb{E}\left[c_1\xi_1+c_2\xi_2\right]$ «побольше», а $\mathbb{D}\left[c_1\xi_1+c_2\xi_2\right]$ «поменьше», надо стремиться к выбору таких пар (ξ_1,ξ_2), для которых их ковариация была бы как можно ближе к минус единице.

Изложенный эффект отрицательной коррелированности, называемый эффектом Марковица, является одной из основных идей диверсификации при инвестировании — при составлении портфеля ценных бумаг надо стремиться к тому, чтобы вложения делались в бумаги, среди которых по возможности много отрицательно коррелированных.

Другая идея, лежащая в основе диверсификации, основана на следующем соображении.

Пусть $\xi_1,\ldots,\xi_N$ — последоватльность некоррелированных случайных величин с дисперсиями $\mathbb{D}\left[\xi_i\right]\le C, i=1,\ldots,N$, где C — некоторая константа. Тогда

$$\mathbb{D}\left[\omega_1\xi_1+\cdots+\omega_N\xi_N\right]=\sum_{i=1}^{N}\omega_i^2\mathbb{D}\left[\xi_i\right]\le C\sum_{i=1}^{N}\omega_i^2. \tag{1.26}$$

Поэтому, взяв, например, $\omega_i=\frac{1}{N}$, находим, что

$$\mathbb{D}\left[\omega_1\xi_1+\cdots+\omega_N\xi_N\right]\le\frac{C}{N}\to 0, N\to\infty \tag{1.27}$$

Этот эффект некоррелиованности говорит о том, что если инвестирование производится в некоррелированные ценные бумаги, то для уменьшения риска $\mathbb{D}\left[\omega_1\xi_1+\cdots+\omega_N\xi_N\right]$, надо по возможности брать их число N как можно большим.

Вернемся к вопросу о дисперсии $\mathbb{D}\left[R\right]$ величины

$$R=\omega_1 r_1+\cdots+\omega_N r_N \tag{1.28}$$

Имеем

$$\mathbb{D}\left[R\right]=\sum_{i=1}^{N}\omega_i^2\mathbb{D}\left[r_i\right]+\sum_{i,j=1,i\neq j}^{N}\omega_i\omega_j\mathbf{Cov}\left(r_i,r_j\right) \tag{1.29}$$

Возьмем здесь $\omega_i=\frac{1}{N}$. Тогда

$$\sum_{i=1}^{N}\omega_i^2\mathbb{D}\left[r_i\right]=\sum_{i=1}^{N}\frac{1}{N^2}\mathbb{D}\left[r_i\right]=\frac{1}{N^2}\sum_{i=1}^{N}\mathbb{D}\left[r_i\right]=\frac{1}{N}\cdot\frac{1}{N}\sum_{i=1}^{N}\mathbb{D}\left[r_i\right]= \tag{1.30}$$

$$=\frac{1}{N}\cdot\overline{\sigma}_N^2, \tag{1.31}$$

где $\bar{\sigma}_N^2=\frac{1}{N}\sum_{i=1}^{N}\mathbb{D}\left[r_i\right]$ — средняя дисперсия.

Далее,

$$\sum_{i,j=1,i\neq j}^{N}\omega_i\omega_j\mathbf{Cov}\left(r_i,r_j\right)=\frac{1}{N^2}\cdot\sum_{i,j=1,i\neq j}^{N}\mathbf{Cov}\left(r_i,r_j\right)= \tag{1.32}$$

$$=\frac{1}{N^2}\cdot N(N-1)\cdot\frac{1}{N(N-1)}\sum_{i,j=1,i\neq j}^{N}\mathbf{Cov}\left(r_i,r_j\right)= \tag{1.33}$$

$$=\frac{N-1}{N}\cdot\overline{\mathbf{Cov}}_N \tag{1.34}$$

где $\overline{\mathbf{Cov}}_N$ есть средняя ковариация

$$\overline{\mathbf{Cov}}_N=\frac{1}{N(N-1)}\sum_{i,j=1,i\neq j}^{N}\mathbf{Cov}\left(r_i,r_j\right). \tag{1.35}$$

Таким образом,

$$\mathbb{D}\left[R\right] = \frac{1}{N} \cdot \overline{\sigma}_N^2 + \left(1 - \frac{1}{N}\right) \cdot \overline{\mathbf{Cov}}_N, \tag{1.36}$$

и ясно, что если $\overline{\sigma}_N^2 \leq C$ и $\overline{\mathbf{Cov}}_N \to \overline{\mathbf{Cov}}$ при $N \to \infty$, то

$$\mathbb{D}\left[R\right] \to \overline{\mathbf{Cov}}, N \to \infty. \tag{1.37}$$

Из этой формулы мы видим, что если $\overline{\mathbf{Cov}}$ равна нулю, то диверсификацией с достаточно большим N риск инвестирования $\mathbb{D}\left[R\right]$, может быть сделан сколь угодно малым. К сожалению, на практике, как правило, имеется положительная корреляция в ценах (они движутся довольно-таки согласованно в одном направлении), что приводит к тому, что $\overline{\mathbf{Cov}}_N$ не стремится к нулю при $N \to \infty$. Предельное значение $\overline{\mathbf{Cov}}$ и есть тот систематический, иначе — рыночный — риск, который присущ рассматриваемому рынку и диверсификацией не может быть редуцирован. Первый же член в формуле 1.36 определяет несистематический риск, который может быть редуцирован, как мы видели, выбором большого числа акций.

2. МОДЕЛИ ВРЕМЕННЫХ РЯДОВ

В этой главе будут рассмотрены некоторые модели временных рядов с помощью которых можно решать задачу прогнозирования ожидаемой доходности активов. Проверка на реальных данных качества прогнозирования и доходностей портфелей, построенных с импользованием рассматриваемых моделей, будет приведена в главе 3.

2.1. Общий подход к прогнозированию рядов

Процесс прогнозирования заключается в оценке будущего значения временного ряда либо путем моделирования ряда исключительно на основе его прошлого поведения (авторегрессия), либо путем включения других внешних переменных.

Чтобы применить модели машинного обучения к задачам прогнозирования, временной ряд необходимо преобразовать в матрицу, где каждое значение связано с определенным прерыдущим значением ряда (лагом). В контексте временного ряда, лаг относительно момента времени t определяется как значение ряда на предыдущих временных шагах. Например, лаг 1 представляет значение на временном шаге $t - 1$, тогда как лаг m представляет значение на временном шаге $t - m$.

Это преобразование необходимо для моделей машинного обучения для захвата зависимостей и закономерностей, которые существуют между прошлыми и будущими значениями во временном ряду. Используя лаги в качестве входных признаков, модели машинного обучения могут учиться на прошлом и делать прогнозы относительно будущих значений. Количество лагов, используемых в качестве входных признаков в матрице, является важным гиперпараметром, который необходимо тщательно настраивать для получения наилучшей производительности модели.

Модели машинного обучения в основном заточены на решение табличных задач. Однако, они несложным образом адаптируются для пронозирования временных рядов. Признаки формируются как лаги временного ряда. Процесс формирования матрицы объекты-признаки схематично показан на рисунке 2.1.

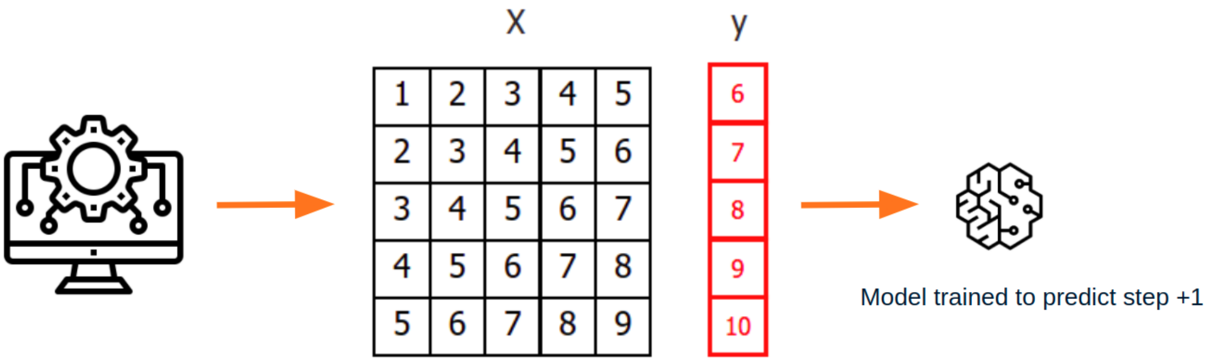

Рис. 2.1. Матрица объекты-признаки

После того, как данные были перестроены в новую форму, любая регрессионная модель может быть обучена прогнозировать следующее значение (шаг) ряда. Во время обучения модели каждая строка считается отдельным экземпляром данных, где значения на лагах $1, 2, \ldots p$ считаются предикторами для целевого количества временного ряда на временном шаге $p + 1$.

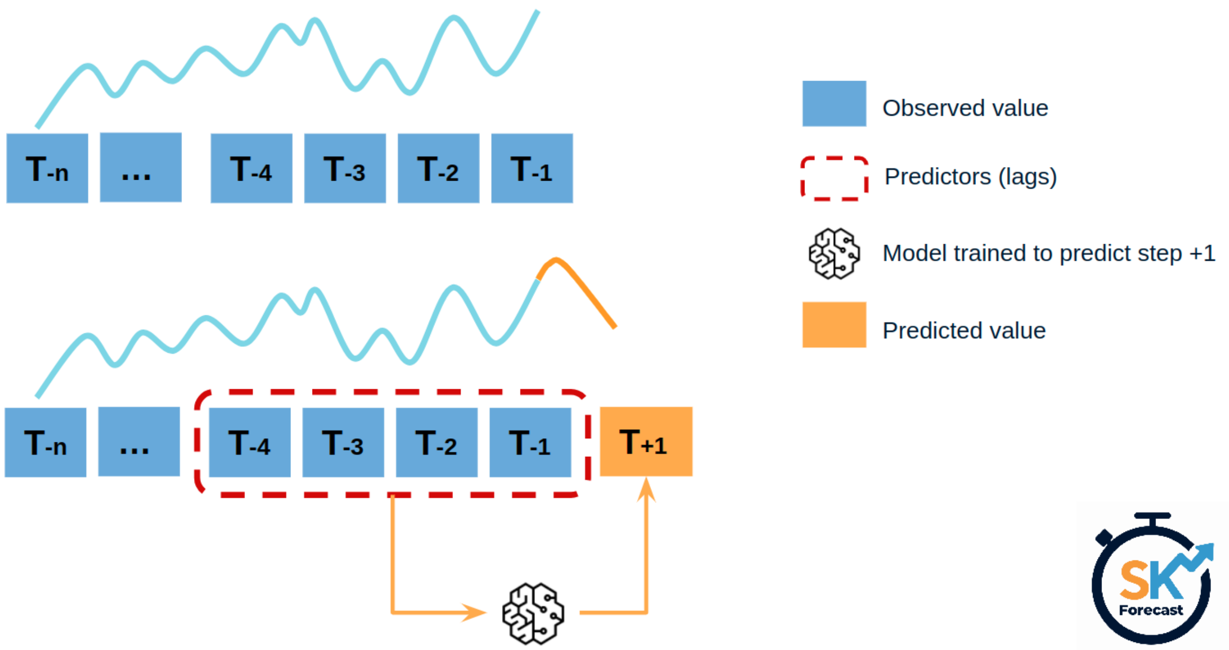

Рис. 2.2. Формирование признаков как временных лагов ряда

В прогнозировании временных рядов процесс бэктестинга заключается в оценке производительности предсказательной модели путем ее ретроспективного применения к историческим данным. Таким образом, это особый тип перекрестной проверки, применяемый к предыдущим периодам.

Цель бэктестинга — оценить точность и эффективность модели . Проверяя модель на исторических данных, можно оценить насколько хорошо она работает на данных, которые она ранее не видела. Это важный шаг в процессе моделирования, поскольку он помогает гарантировать, что модель является надежной и устойчивой.

Бэктестинг можно проводить с использованием различных методов, таких как простые разделения обучения и тестирования или более сложные методы, такие как скользящие окна или расширяющиеся окна. Выбор метода зависит от конкретных потребностей анализа и характеристик данных временных рядов.

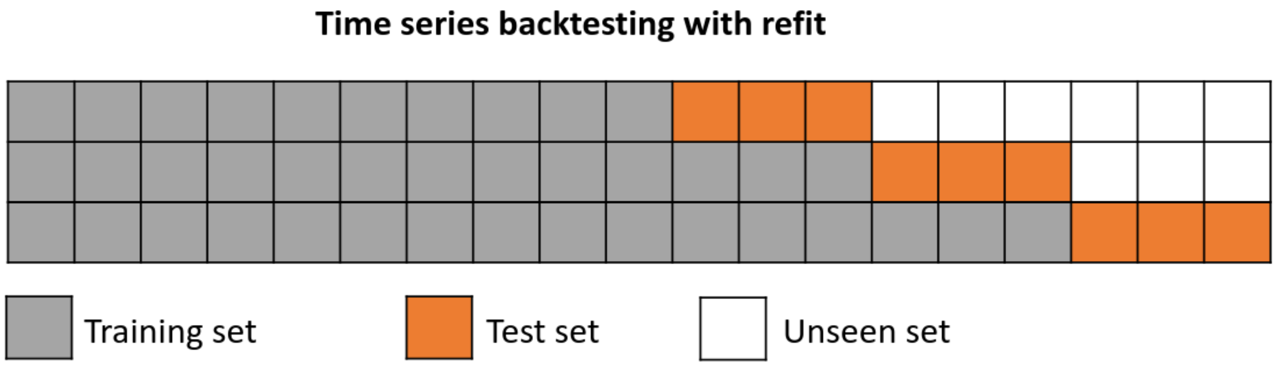

Рис. 2.3. Формирование выборок при бектестинге

Будем использовать подход с расширением обучающего множества и переобучением на каждом шаге. При таком подходе модель обучается перед каждым прогнозированием, и все доступные данные на тот момент используются в процессе обучения. Это отличается от стандартной перекрестной проверки, где данные случайным образом распределяются между обучающими и проверочными наборами.

Вместо рандомизации данных этот бэктестинг последовательно увеличивает размер обучающего набора, сохраняя временной порядок данных. Благодаря этому модель можно тестировать на все больших объемах исторических данных, что обеспечивает более точную оценку ее качества прогнозирования.

2.2. Линейная регрессия

Пусть X и Y матрица обектов-признаков и вектор целевых значений, построенные по историческим данным.

Линейная регрессия есть линейная комбинация признаков (лагов) x и весов w.

$$a(x) = w_0 + w_1 x_1 + \dots + w_n x_n = \langle x, w \rangle \tag{2.1}$$

Решение w^* находится методом наименьших квадратов. Определим функционал потель как средний квадрат ошибки модели на всех элементах выборки.

$$Q(a) = \frac{1}{\ell} \sum_{i=1}^{\ell} (y_i - a(x_i))^2 = ||Xw - y||^2 \tag{2.2}$$

Необходимое условие минимума

$$\frac{\partial Q}{\partial w} = 2X^T(Xw - y) = 0 \tag{2.3}$$

Решая систему уравнений получим аналитический вид решения оптимизационной задачи

$$w^* = \left(X^T X\right)^{-1} X^T y \tag{2.4}$$

Огромным преимуществом данной модели является её простота и скорость обучения. Также веса модели можно интерпретировать. Большие абслолютные значения весов означают сильный вклад соответсвующих лагов в целевую переменную. Далее перейдем к рассмотрению более сложных моделей.

2.3. Случайный лес

Алгоритм случайного леса есть простое голосование над решающими деревьями. Он является одним из сильнейших алгоритмов машинного обучения. Описывается основная идея работы алгоритма, а более подробно об этом алгоритме написано в статье [7].

Дерево решений есть бинарное дерво. Определены вершины двух типов:

- внутренние — содержит предикат $b_v : \mathbb{X} \to \{0, 1\}$
- листовые — хранит выходное значение $c_v \in \mathbb{Y}$

Этапы обработки деревом входящего объекта x

1. Стартуем из корня
2. Вычисляем текущий предикат $b_v(x)$
3. Если $b_v(x) = 0$ то делаем шаг в левое поддерево, иначе — в правое
4. Пока не дошли до листовой вершины, повторяем шаги 2 и 3
5. Возвращаем значение c_v в листе

В качестве базовых алгоритмов выберем набор решающих деревьев $b_1, \ldots, b_k$. Объеденим результаты работы базовых алгоритмов с помощью простого голосования

$$a(x) = \frac{1}{k} \sum_{i=1}^{k} b_i(x) \tag{2.5}$$

Ошибку работы ансамбля можно разложить на 3 компоненты

$$Q(a) = bias(a) + variance(a) + noie \tag{2.6}$$

где

$$bias(a) = f(x) - \mathbb{E}_X \left[a(x, X)\right] \tag{2.7}$$

$$\tag{2.8}$$

— смещение алгоритма,

$$variance(a) = \mathbb{E}_X \left[a(x, X)\right] - \mathbb{E}_X \left[a(x, X)\right]^2 \tag{2.9}$$

$$\tag{2.10}$$

— разброс алгоритма,

$$noie = \mathbb{E}_X \left[\mathbb{E}_\varepsilon \left[(y(x, \varepsilon) - f(x))^2\right]\right] \tag{2.11}$$

— неустранимый шум.

Смещение ансамбля определяется смещением базового алгоритма. Поэтому разумно строить неглубокие деревья.

$$bias(a) = f(x) - \mathbb{E}_X\left[a(x,X)\right] = \tag{2.12}$$

$$= f(x) - \mathbb{E}_X\left[\frac{1}{k}\sum_{i=1}^{k} b(x, X^i)\right] = \tag{2.13}$$

$$= f(x) - \mathbb{E}_X\left[b(x,X)\right] = \tag{2.14}$$

$$= bias_X(b) \tag{2.15}$$

Разброс ансамбля определяется числом базовых алгоритмов в нем и корреляциями между ними. Постараемся добиться некоррелированности, или, по крайней мере, непохожести базовых алгоритмов за счет обучения каждого из нах на разных данных. С этим помогает идея бутстрапирования.

$$variance(a) = \mathbb{E}_X\left[a(x,X) - \mathbb{E}_x\left[a(x,X)\right]\right]^2 = \tag{2.16}$$

$$= \mathbb{E}_X\left[\frac{1}{k}\sum_{i=1}^{k} b_i - \mathbb{E}_X\left[\frac{1}{k}\sum_{i=1}^{k}\right]\right]^2 = \tag{2.17}$$

$$= \frac{1}{k^2}\mathbb{E}_X\left[\sum_{i=1}^{k}\left(b_i - \mathbb{E}_X\left[b_i\right]\right)\right]^2 = \tag{2.18}$$

$$= \frac{1}{k^2}\sum_{i=1}^{k} variance_X(b_i) + \frac{1}{k^2}\sum_{i\neq j}\mathbf{Cov}\left(b_i, b_j\right) \tag{2.19}$$

Помимо алгоритмов машинного обучения для прогнозирования временных рядом можно воспользоваться классическими моделями временных рядов. Одной из них является модель ARIMA.

2.4. ARIMA

Модель ARIMA обобщает модель ARMA. Подробно работа модели описывается здесь [5].

Модель $ARMA(p,q)$ сочетает в себе моедли авторегрессии $AR(p)$ и скользящего среднего $MA(q)$.

Пусть задано фильтрованное вероятностное пространство $(\Omega, \mathscr{F}, (\mathscr{F}_n), P)$

Будем считать что $\mathscr{F}_n = \sigma(\dots, \varepsilon_{-1}, \varepsilon_0, \varepsilon_1, \dots, \varepsilon_n)$ с белым шумом $\varepsilon = (\varepsilon_n)$.

По определению, последовательность $x = (x_n)$ является ARMA-моделью, если

$$x_n = \mu_n + \sigma\varepsilon_n \tag{2.20}$$

где

$$\mu_n = (a_0 + a_1x_{n-1} + \dots + a_ph_{n-p}) + (b_1\varepsilon_{n-1} + b_2\varepsilon_{n-2} + \dots + b_q\varepsilon n - q) \tag{2.21}$$

Эта модель допускает обобщение на случай когда исходный ряд не стационарен. А именно, расммотрим разности процесса x

$$\Delta x_n = x_n - x_{n-1} \tag{2.22}$$

Повторяя операцию взятия разности d раз, получим процесс $\Delta^d x = (\Delta^d x_n)$. Если полученный процесс «более стационарный» чем исходный, то построим уже для него $ARMA(p, q)$ модель.

Описанная модель есть трехпараметрическая модель временного ряда $ARIMA(p, d, q)$. Символически это можно записать

$$\Delta^d ARIMA(p, d, q) = ARMA(p, q) \tag{2.23}$$

В этой главе были рассмотрены некоторые алгоритмы построения моделей временных рядов позволяющие строить прогноз будущих значений на основании истории ряда. Применение эти модели для оценки будущих средних доходностей в теории Марковица, позволяет строить множества оптимальных портфелей. Перейдем к построению и оценке инвесиционных стратегий, основанных на рассмотренных моделях.

3. ПРОВЕРКА СТРАТЕГИЙ НА РЫНОЧНЫХ ДАННЫХ

В этой главе оценивается качество моделей для прогнозирования ожидаемых средних доходностей и оцениваются доходности соотвествующих торговых стратегий. Под стратегией понимается построенный на определенный момент времени и с определенным сроком инвестирования портфель. Рассматриваются стратегии двух типов. Тривиальные, когда капитал инвестируется в один отдельный актив или в весь рынок одновременно, и стратегии, основанные на теории Марковица, где для оценки средней ожидаемой доходности используется некоторая модель из главы 2.

3.1. Подготовка данных

Рыночные данные были выгружены с помощью API с криптовалютной биржи OKX [3]. В качестве доступных для торговли активов рассматриваются 8 наиболее популярных крюптовалют. Временной период с 1 января 2022 по 1 января 2025. Был выбран дневной таймфрейм. Период инвестирования 1 неделя.

На графике 3.1 изображены динамики цен активов.

Перейдем от цен к недельным доходностям. Временные ряды, соответсвующие доходностям, представлены на графике 3.2, а некоторые статистики относительно распределений доходностей в таблице 3.1

Таблица 3.1. Доходности активов

	BTC	ETH	DOT	OKB	XRP	SOL	TRX	LTC
mean	0.0073	0.0038	-0.0031	0.0077	0.0132	0.0114	0.0105	0.0025
std	0.0779	0.0961	0.1110	0.0936	0.1328	0.1482	0.0800	0.1001
min	-0.3328	-0.3830	-0.3925	-0.3591	-0.3596	-0.6018	-0.3162	-0.3392
25%	-0.0358	-0.0477	-0.0724	-0.0401	-0.0506	-0.0765	-0.0236	-0.0513
50%	0.0026	-0.0013	-0.0084	-0.0016	-0.0003	-0.0034	0.0106	0.0001
75%	0.0446	0.0555	0.0573	0.0494	0.0430	0.0906	0.0373	0.0546
max	0.3566	0.5056	0.6188	0.4003	1.0235	0.7409	0.7407	0.5294

Рис. 3.1. Цены активов

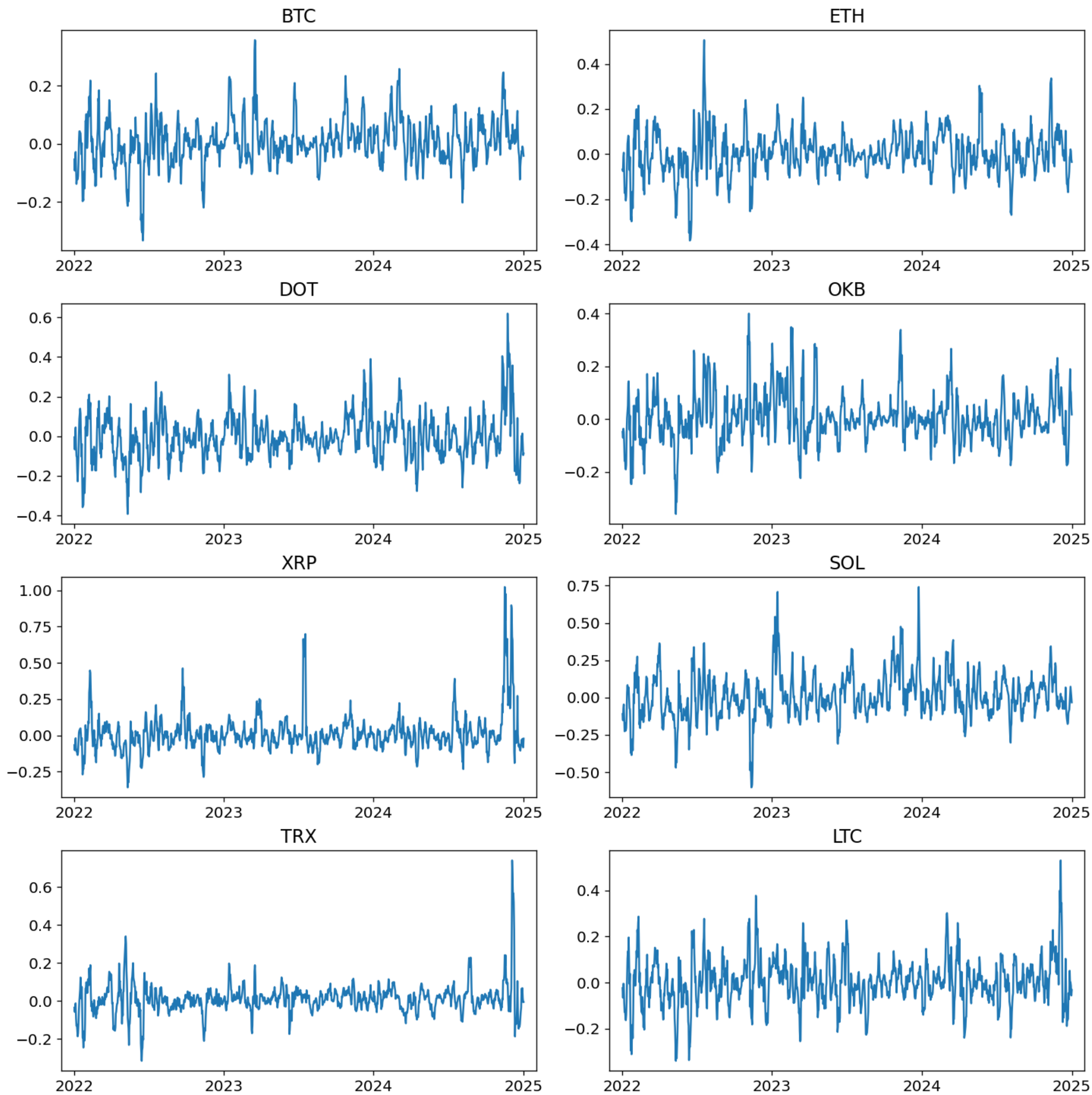

Рис. 3.2. Доходности активов

Можно видеть редкие но достаточно сильные скачки.

Распределение доходностей активов представлено на гистограммах на рисунке 3.3

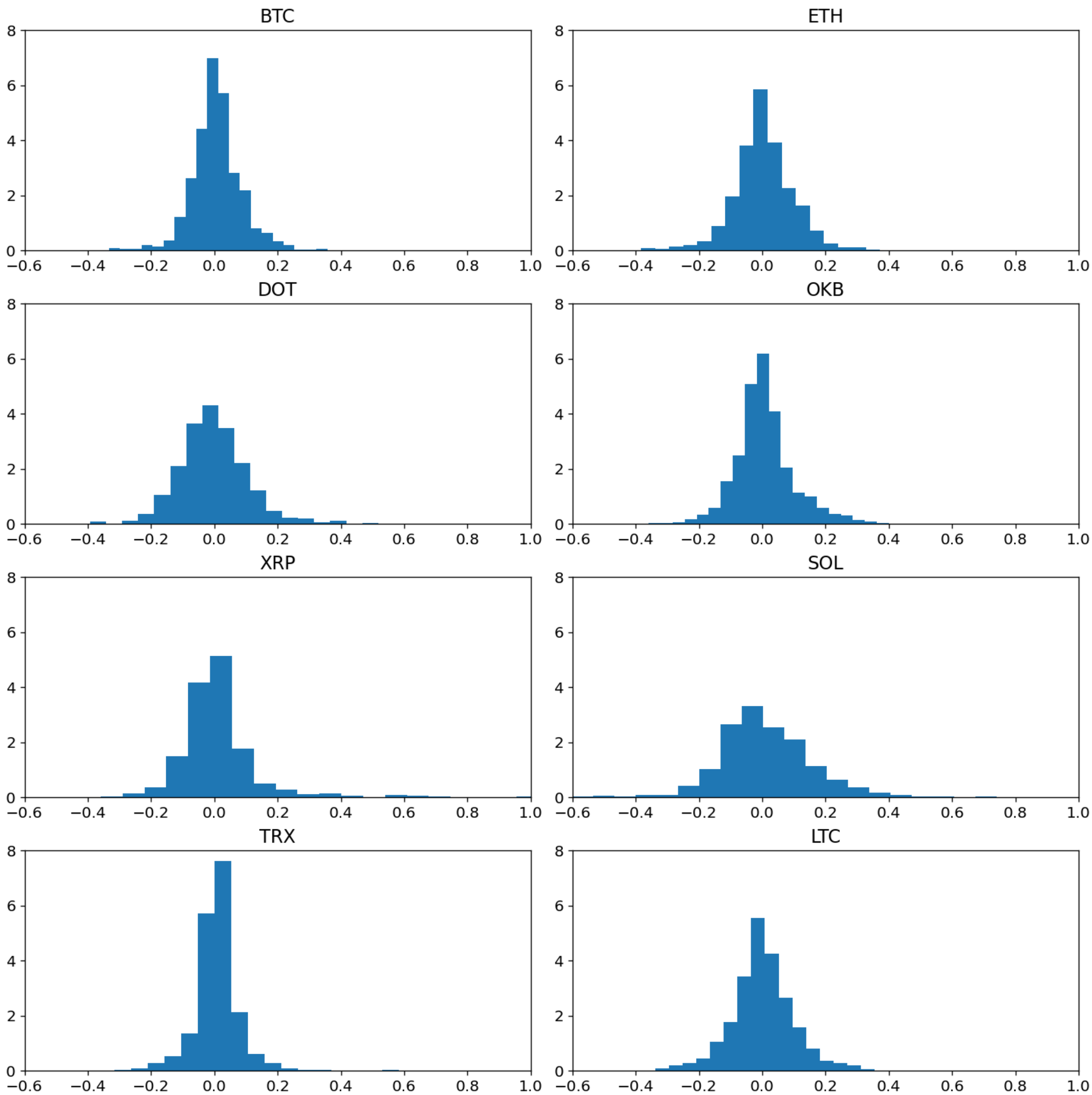

Рис. 3.3. Гистограммы доходностей активов

Из гистограмм видно, что распределение доходностей унимодально и имеет тяжелые хвосты.

На графике 3.4 сравниваются активы с точки зрения среднего и стандартного отклонения доходности.

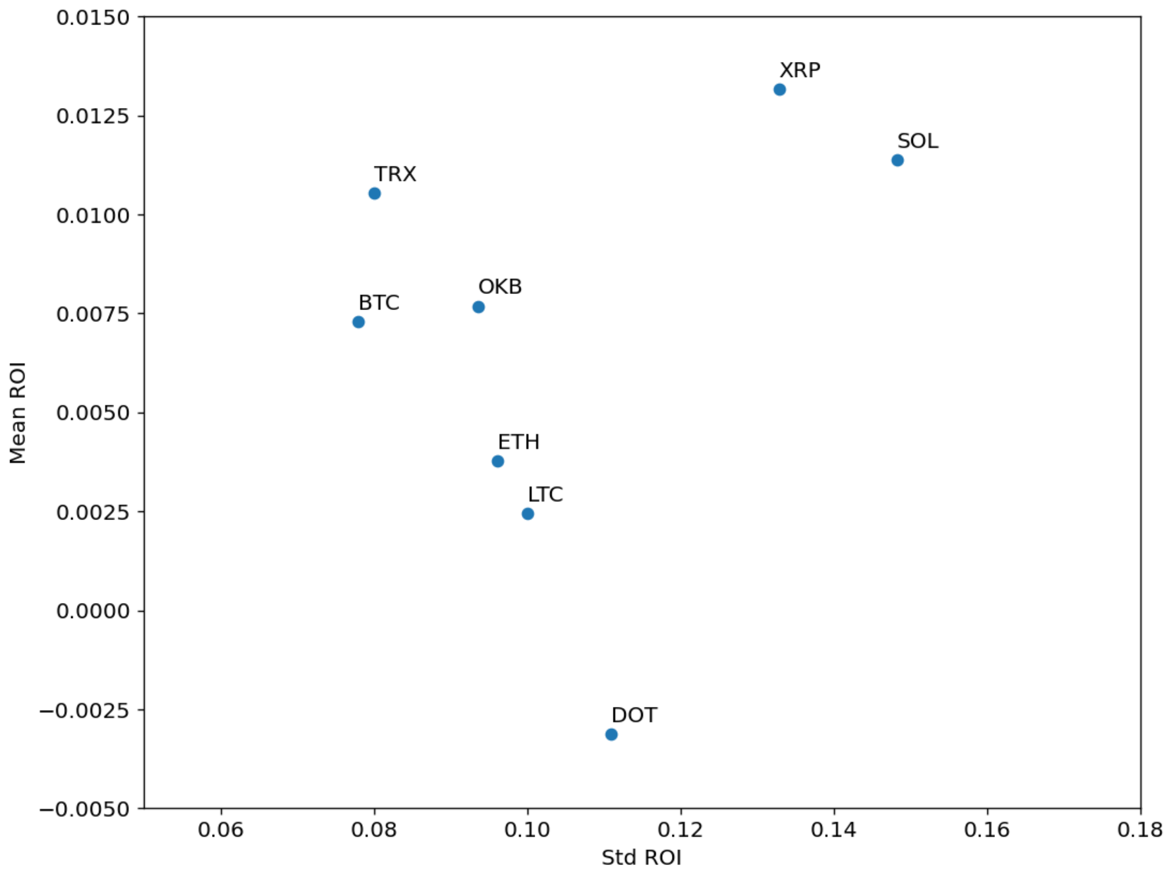

Рис. 3.4. Среднее и стандартное отклонение доходностей

Разделим имеющиеся данные на валидационную и тестовую выборки. В качестве тестовых данных возьмем 2024 год. По валидационной выборке подберем оптимальное число лагов ряда и индивидуальные гиперпараметры алгоритмов.

В дальнейшем тестовые данные будут использоваться для:

1. оценки качества прогнозирования средней ожидаемой доходности
2. тестирования портфельных стратегий

Из тестовых днных формируется набор тест-кейсов на которых и оценивается качество. Процесс формирования тест-кейсов схематично проилюстрирован на рисунке 3.5.

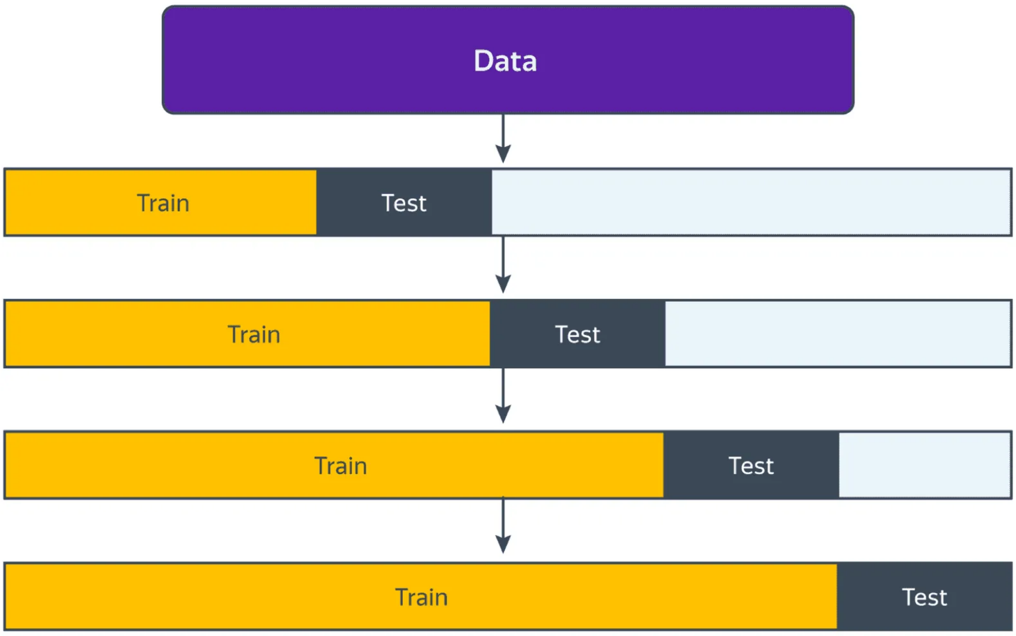

Рис. 3.5. Формирование тест-кейсов из тестовых данных

Следующим этапом идет расчет необходимых параметров для оптимизации портфеля — оценка ковариаций и прогноз средних значений доходности.

3.2. Оценка ковариации между активами

Особую сложность представляет задача прогноза будущей ковариации временных рядов. Вполне естественным вляется предположение стационарности ковариации во времени. Поэтому воспользуемся выборочной оценкой ковариации по историческим данным.

Имея r_t - вектор-столбец доходностей в момент времени t, по истории наблюдений $r_1, \cdots r_n$ выборочная ковариация $\hat{\Sigma}$ рассчитывается как

$$\hat{\Sigma} = \frac{1}{n} \sum_{t=1}^{n} (r_t - \overline{r}) \cdot (r_t - \overline{r})^T \tag{3.1}$$

где $\overline{r} = \frac{1}{n} \sum_{t=1}^{n} r_t$.

Корреляция Пирсона между доступными активами представлена на рисунке 3.6.

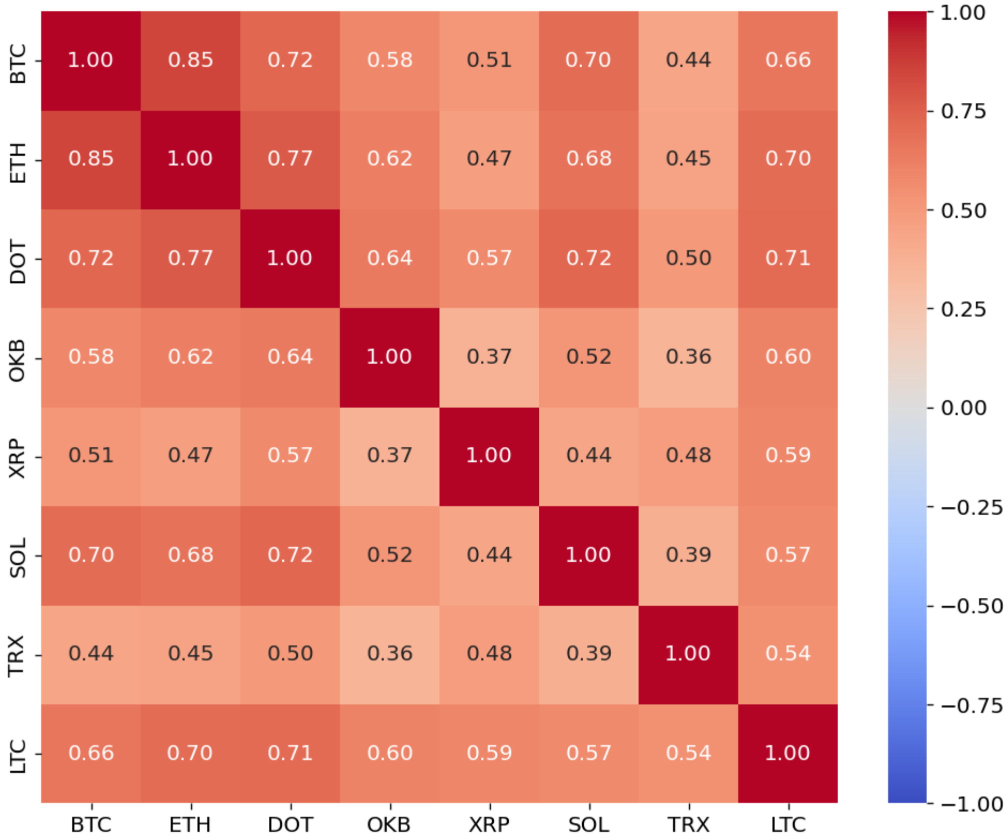

Рис. 3.6. Корреляции доходностей активов

Активы имею сильную положительную корреляцию. Согласно теории, описанной в главе 1, желательным является возможность инвестирования в активы с отрицательной корреляцией. Однако, это не помешает редуцировать риски портфеля.

Для формирования портфеля остается оценить средние ожидаемые доходности. Перейдем к рассмотрению моделей для прогнозирования этих значений.

3.3. Модели оценки средней доходности

Задача оценки средней ожидаемой доходности сводиться к умению прогнозировать значение основываясь на стории наблюдений. Для этого подходят классические статистические модели, модел имашинного обучения и нейросети, адаптированные для прогнозирования временных рядов. Реалиации моделей взяты из программных библиотек для Python [1] и [2].

Ограничимся рассмотрением следующих моделей:

1. NAIVE - прогнозирование историческим средним
2. MARTINGAL - прогноз последним наблюдаемым значением
3. ARIMA - модель авторегрессии и скользящего среднего (см. 2.4)
4. LR - линейная регрессия (см. 2.2)
5. RF - случайный лес (см. 2.3)

Для каждого актива будем строить отдельную модель не принимающую в расчет историю других активов. Таким образом, для прогноза будующих доходностей активов необходимо построить моделей по числу активов.

Некоторые модели (ARIMA, RF) — допускают свободу в выборе гиперпараметров. Подбор гиперпараметров моделей осуществлялся по тренировочной выборке.

Качество прогнозирования моделей оценивается с помощью среднеквадратичной ошибки MSE (Mean Squared Error):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2 \tag{3.2}$$

где r_i - истинное значение доходности, а $\hat{r}_i$ - прогнозное значение модели на i-м объекте тестовой выборки.

Результаты оценки качества прогнозирования на тестовых данных представлены в таблице 3.2

Таблица 3.2. Качество прогнозирования MSE$\cdot 10^4$

	NAIVE	MARTINGAL	LR	ARIMA	RF
BTC	5.63	1.20	1.58	1.62	2.07
ETH	8.00	1.99	4.47	3.70	5.05
DOT	16.51	3.89	4.09	3.98	5.28
OKB	6.06	1.56	1.77	1.95	2.05
XRP	24.33	5.04	6.98	5.71	6.53
SOL	21.19	4.47	11.86	6.16	5.31
TRX	7.96	1.85	4.19	5.30	6.04
LTC	8.53	2.66	2.24	3.22	5.11

Наихудшее значениие показывает подход NAIVE. Это обусловлено резким ростом цен в 2024 году после относительно спокойной динамики. Метод

MARTINGAL показывает хорошие результаты в случае рядов с затяжным трендов. Остальные модели показывают сопоставимое качество.

3.4. Оценка доходностей стратегий

Будем рассматривать стратегии двух видов:

- тривиальные
- основанные на идеи Марковица

Среди тривиальных стратегий выберем следующие:

1. UNIFORM - равномерное инвестированиие во все доступные активы
2. MOST RISKY - актив с наибольней дисперсией доходности
3. LESS RISKY - актив с наименьшей дисперсией доходности
4. BEST RETURN - актив с наибольшей средней доходностью
5. WORST RETURN - актив с наименьшей средней доходностью

Стратегии Марковица определяются риск-параметром и моделью оценки средней ожидаемой доходностью. Риск-параметр будем воспринимать как параметризацию класса стратегий с определенной моделью оценки средней ожидаемой доходности. Таким образом, одной стратегии Марковица соответсвует множество стратегий с разным риск-параметром. Это множество стратегий будет называть фронтирой.

На каждом тест-кейсе с помощью стратегии формируется инвестиционный портфель в расчете на единичную сумму инвестирования и оценивается доходность ROI (Return On Investment) полученной стратегии. Метрика ROI определяется для стратегии аналогично доходности портфеля 1.1.

На графике 3.7 представлены фронтиры соответсвующие торговым стратегиям. Серым цветом отмечены тривиальные портфели. По оси абсцисс отложены стандартые отклонения ROI, а по оси ординат — средние значение ROI.

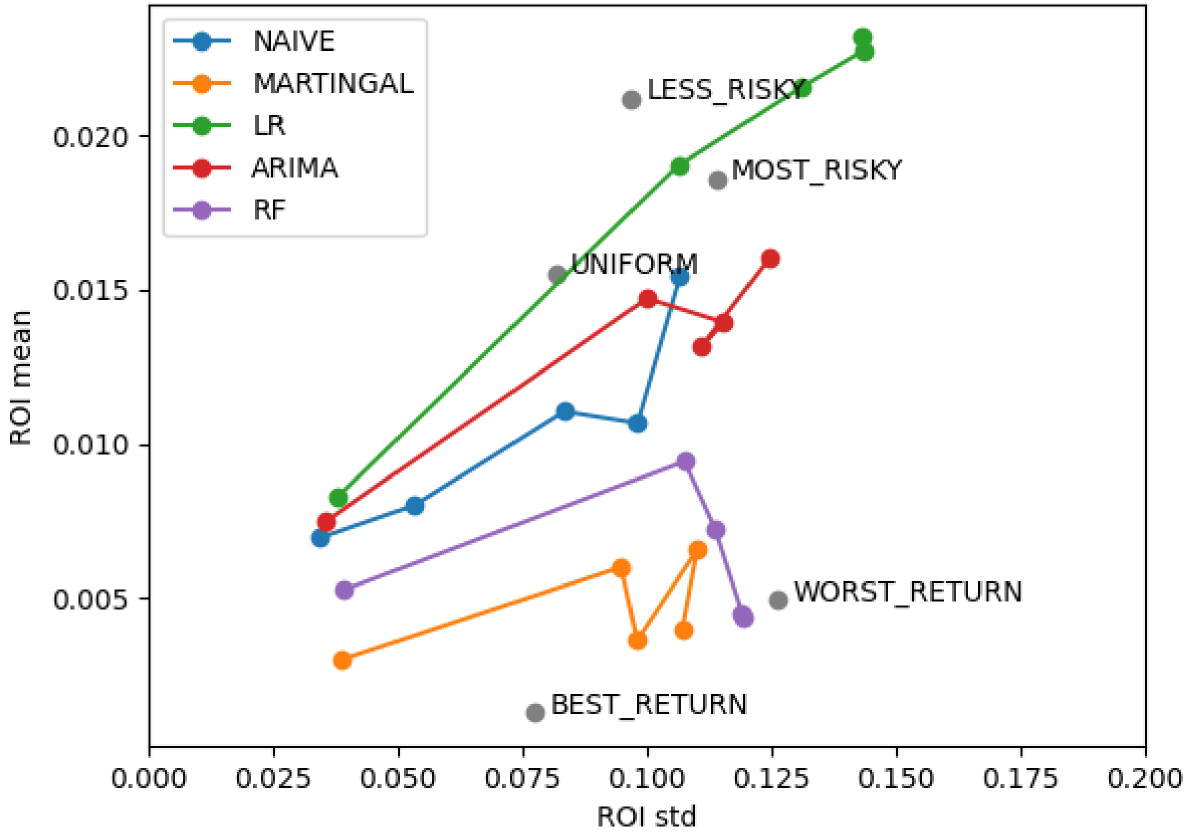

Рис. 3.7. Результаты тестирования стратегий

Более детально средние значения и стандартные отклонения ROI стратегий представлены в таблицах 3.3 и 3.4 соответственно.

Метрики тривиальных портфелей представлены в таблице 3.5

Таблица 3.3. Средние ROI $\cdot 10^3$

	0.01	0.25	0.50	0.75	1.00
NAIVE	6.9451	8.0025	11.0462	10.6657	15.4227
MARTINGAL	2.9859	6.0019	3.6178	6.5656	3.9942
LR	8.2600	19.0185	21.5537	22.7442	23.1668
ARIMA	7.4648	14.7066	13.9296	13.1520	15.9971
RF	5.2633	9.4236	7.2398	4.3777	4.5050

Таблица 3.4. Стандартное отклонение ROI $\cdot 10^2$

	0.01	0.25	0.50	0.75	1.00
NAIVE	3.4328	5.3375	8.3427	9.8131	10.6650
MARTINGAL	3.8577	9.4886	9.8051	11.0111	10.7128
LR	3.7892	10.6314	13.1000	14.3570	14.3071
ARIMA	3.5454	10.0037	11.5506	11.1001	12.4518
RF	3.9215	10.7564	11.3710	11.9397	11.9118

Таблица 3.5. Тривиальные портфели

	mean ROI $\cdot 10^3$	std ROI $\cdot 10^2$
UNIFORM	15.5372	8.1775
MOST RISKY	18.5904	11.3970
LESS RISKY	21.1635	9.6881
BEST RETURN	1.2790	7.7329
WORST RETURN	4.9217	12.6324

На тестовых данных метод оценки средней доходности с помощью модели линейной регрессии строго доминирует над всеми остальными методами при любом значении риск-параметра.

Модели ARIMA и NAIVE позволяют контролировать риски портфеля за счет изменения риск-параметра. При формировании портфеля минимального риска эти модели сопоставимы между собой.

Модели RF и MARTINGAL показали неудовлетворительное качество. При увеличении толерантности к риску, доходность портфеля не возрастает, что противоречит теории и здравому смыслу.

Инвестирование равных долей во все доступные активы является приемлимым, однако такой подход не позволяет контролировать риски.

Формирование портфеля состоящего только из одного актива сильно непредсказуемо и, следовательно, рисковано из-за сильных колебаний цен (специфика криптовалютных рынков).

ЗАКЛЮЧЕНИЕ

В работе была рассмотрена проблема формирования оптимального портфеля с точки зрения ожидаемой доходности и принимаемого риска. Был предложен метод оценки средних ожидаемых доходностей, необходимых для построения портфеля оптимального по Марковицу. В рамках этого метода рассматривались некоторые модели прогнозирования временных рядов. В практической части были проверены на реальных данных доходности торговых стратетегий, построенных согласно теории Марковица, где для прогнозирования неизвестных характеристик использовались рассмотренные модели.

По имеющимся эмпирическим результатам можно сделать следующие выводы:

- активы имеют сильную положительную корреляцию
- стремление сформировать портфель с большей доходностью влечет большие риски
- диверсификация действительно позволяет снижать риск портфеля
- как правило, формирование портфеля доминирует над инвестированием в отдельные активы
- линейная модель авторегрессии показала лучшее качество для оценки средней доходности

Полученные теоретические выкладки и программную реализацию формирования портфелей можно применять при выборе своей инвестиционной стратегии.

Дальнейшие шаги по исследованию данной темы могут быть следующими:

- рассмотреть другие классы методов прогнозирования временных рядов
- помимо авторегрессионных признаков, учесть влияние внешних факторов на формирование цен
- расширить рассматриваемый набор активов
- исследовать другие таймфреймы и периоды инвестирования

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Библиотека Python для прогнозирования временных рядов с использованием моделей машинного обучения [Электронный ресурс]. — Режим доступа: https://skforecast.org/0.15.1/. — Дата доступа 25.05.2025
2. Библиотека машинного обучения с открытым исходным кодом. [Электронный ресурс]. — Режим доступа: https://scikit-learn.org/stable/. — Дата доступа 25.05.2025
3. Криптовалютная биржа с расширенными финансовыми предложениями [Электронный ресурс]. — Режим доступа: https://www.okx.com/. — Дата доступа: 25.05.2025
4. Шарп, У.Ф. Инвестиции : учебник : пер. с англ. / У.Ф. Шарп, Г.Д. Александер, Д.В. Бэйли. – Москва : ИНФРА-М,, 2022. – 1028 с.
5. Ширяев, А. Н. Основы стохастической финансовой математики: Т.1: Факты, модели / А. Н. Ширяев – МЦНМО, 2016. – 440 с.
6. Ширяев, А. Н. Основы стохастической финансовой математики: Т.2: Теория / А. Н. Ширяев – МЦНМО, 2016. – 464 с.
7. Breiman, L. Random Forests. Machine Learning / Leo Breiman — Statistics Department University of California Berkele, CA, 2001, 33 p.
8. Markowitz, H. Portfolio selection / H. Markowitz — The Journal of Finance, March 1952, 77-91 p.
9. Linter J. The valuation of risky assets and the selection of risky investments on stock portfolios and capital budgets / John Linter — Review of Economics and Statistics, February 1965, 13-34 p.
10. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System / Satoshi Nakamoto — Japan, 2008, 9 p.
11. Roll R., Ross S. A. An emperical investigation of the arbitrage pricing theory / Richard Roll, Stephen Ross — Journal of Finance, 1980, 1073-1103 p.
12. Ross S. A. The arbitrage theory of capital asset pricing / Stephen A. Ross — Journal of Finance, 1989, 1-18 p.
13. Sharpe W. F. Capital asset prices: A theory of market equilibrium under conditions of risk / William F. Sharp — Journal of Finance, September 1964, 425-442 p.

ПРИЛОЖЕНИЕ А

```python

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import json
import tqdm
from scipy import optimize as opt
from sklearn import metrics as skmetrics
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostRegressor, RandomForestRegressor
from skforecast.direct import ForecasterDirect, ForecasterDirectMultiVariate
from skforecast.recursive import ForecasterRecursive, ForecasterSarimax
from skforecast.sarimax import Sarimax
from skforecast.model_selection import TimeSeriesFold
from skforecast.model_selection import backtesting_forecaster, \
    backtesting_sarimax, grid_search_forecaster, grid_search_sarimax


df_prices = pd.read_csv(
    '../code/data/crypto.csv',
    index_col='dt',
    parse_dates=['dt'])
df_prices.drop('TON-USDT', axis=1, inplace=True)
df_prices.columns = [c.split('-')[0] for c in df_prices.columns]
print(df_prices.head())

days_shift = 7
df_returns = df_prices.diff(days_shift) / df_prices.shift(days_shift)
df_returns = df_returns[df_returns.isna().sum(axis=1) == 0]
df_returns = df_returns[df_returns.index >= '2022-01-01']

n_observations, n_assets = df_returns.shape
print(n_observations, n_assets)
print(df_returns.head())

threshold_date = '2023-10-01'
df_returns_test = df_returns[df_returns.index >= threshold_date]
df_returns_train = df_returns[df_returns.index < threshold_date]
print(df_returns_train.shape, df_returns_test.shape)

def mse_last_value(y_true, y_pred):
    idxs = range(0, len(y_true), days_shift)
    return skmetrics.mean_squared_error(
        y_true.iloc[idxs], y_pred.iloc[idxs])

# HP optimization
cv = TimeSeriesFold(
    steps=days_shift,
    initial_train_size=50,
    refit=False,
)

lags_grid = {
    '0': 1,
    '1': range(1, 4),
    '2': range(1, 8),
    '3': range(1, 15),
}

arima_params = {
    'order': [
        (1, 0, 0),
        (0, 0, 1),
        (1, 0, 1),
        (1, 1, 1),
        (2, 0, 2),
```

```
        (2, 1, 2),
    ]
}

model_grid = [
    (LinearRegression, 'LR', {}),
    (RandomForestRegressor, 'RF', {
        'n_estimators': [10, 50, 100],
        'random_state': [27]
        }),
]

best_models = {}
for c in df_returns.columns:
    # ARMA
    train_data = df_returns_train[c].reset_index(drop=True)
    forecaster = ForecasterSarimax(
                    regressor=Sarimax(),
                    forecaster_id=f'ARIMA_{c}'
                    )
    metric = grid_search_sarimax(
                    forecaster=forecaster,
                    y=train_data,
                    param_grid=arima_params,
                    cv=cv,
                    metric=mse_last_value,
                    return_best=True,
                    n_jobs='auto',
                    verbose=False,
                    show_progress=True
                )
    if best_models.get('ARIMA') is None:
        best_models['ARIMA'] = []
    best_models['ARIMA'].append(forecaster)

    # ML
    for model_builder, model_name, params in model_grid:
        train_data = df_returns_train[c].reset_index(drop=True)
        forecaster = ForecasterRecursive(
                        regressor=model_builder(),
                        lags=range(1, 8),
                        forecaster_id=f'{model_name}_{c}'
                    )
        metric = grid_search_forecaster(
                        forecaster=forecaster,
                        y=train_data,
                        param_grid=params,
                        lags_grid=lags_grid,
                        cv=cv,
                        metric=mse_last_value,
                        return_best=True,
                        n_jobs='auto',
                        verbose=False,
                        show_progress=True
                    )
        if best_models.get(model_name) is None:
            best_models[model_name] = []
        best_models[model_name].append(forecaster)


print(best_models.keys())

# evaluate models on test data
cv = TimeSeriesFold(
    steps=days_shift,
    initial_train_size=50,
    refit=True,
)
backtest_metrics = {}

for model_type, models in best_models.items():
    if model_type == 'ARIMA':
        for c, forecaster in zip(df_returns_test.columns, models):
```

```
                data_test = df_returns_test[c].reset_index(drop=True)
                metric, predictions = backtesting_sarimax(
                              forecaster=forecaster,
                              y=data_test,
                              cv=cv,
                              metric=mse_last_value,
                              n_jobs=-1,
                          )
                if backtest_metrics.get(model_type) is None:
                    backtest_metrics[model_type] = []
                backtest_metrics[model_type].append(metric.values.item())
        else:
            for c, forecaster in zip(df_returns_test.columns, models):
                data_test = df_returns_test[c].reset_index(drop=True)
                metric, pred = backtesting_forecaster(
                    forecaster=forecaster,
                    y=data_test,
                    cv=cv,
                    metric=mse_last_value,
                    n_jobs=-1,
                )
                if backtest_metrics.get(model_type) is None:
                    backtest_metrics[model_type] = []
                backtest_metrics[model_type].append(metric.values.item())


# martingal mse
backtest_metrics['MARTINGAL'] = (
    (df_returns_test - df_returns_test.shift())**2
    ).mean(axis=0).to_list()
backtest_metrics['NAIVE'] = (
    (df_returns_train.mean(axis=0) - df_returns_test)**2
    ).mean(axis=0).to_list()

# mse on test data
(pd.DataFrame(
    {m: backtest_metrics[m] for m in [
        'NAIVE', 'MARTINGAL', 'LR', 'ARIMA', 'RF'
        ]},
    index=df_returns.columns) * 1000
    ).to_latex('../tables/ml_eval_metrics.tex',
               caption='Качество прогнозирования',
               float_format='%.2f',
               position='h',
               label='tab:ml_eval_metrics'
               )

def portfolio_optimizer(mu_hat, cov_hat, tau):
    def objective(w):
        w = w.reshape((-1, 1))
        return (w.T @ cov_hat @ w - tau * w.T @ mu_hat).item()

    def unit_portfolio(w):
        return np.abs(w).sum() - 1

    eq_cons = {
        'type': 'eq',
        'fun': unit_portfolio,
    }
    bounds = [(-1, 1) for i in range(n_assets)]
    x0 = np.ones(n_assets) / n_assets
    sol = opt.minimize(
        fun=objective,
        x0=x0,
        method='SLSQP',
        bounds=bounds,
        constraints=[eq_cons]
    )
    if sol.success:
        return sol.x

# %%
```

```python
def frontier_evaluator(mu_hat, cov_hat, ret_true, frontier_tau):
    frontier = np.full_like(frontier_tau, np.nan)
    for i in range(len(frontier_tau)):
        tau = frontier_tau[i]
        w = portfolio_optimizer(mu_hat, cov_hat, tau)
        if w is None:
            print('not converged')
            continue
        roi = np.dot(w, ret_true)
        frontier[i] = roi
    return frontier

# mu estimators
def naive_estimator(df_hist):
    return df_hist.mean(axis=0)

def martingal_estimator(df_hist):
    return df_hist.iloc[-1]

def ml_estimator_builder(models):
    def func(df_hist):
        mu_hat = []
        for c, forecaster in zip(df_hist.columns, models):
            y = df_hist[c].reset_index(drop=True)
            forecaster.fit(y)
            mu_hat.append(forecaster.predict(days_shift).iloc[-1])
        return np.array(mu_hat)
    return func

n_assets = df_returns.shape[1]

idx_most_risky = np.argmax(df_returns_train.describe().T['std'])
idx_less_risky = np.argmin(df_returns_train.describe().T['std'])
idx_best_return = np.argmax(df_returns_train.describe().T['mean'])
idx_worst_return = np.argmin(df_returns_train.describe().T['mean'])
print(idx_most_risky,
      idx_less_risky,
      idx_best_return,
      idx_worst_return)

def single_asset_portfolio_builder(idx):
    w =  np.zeros(n_assets)
    w[idx] = 1
    return w

def uniform_portfolio_builder():
    return np.full(n_assets, 1 / n_assets)

print(best_models.keys())

results_frontier = []
results_trivial = []

min_history_leng = 91
total_runs = df_returns_test.shape[0] - days_shift - min_history_leng
print(total_runs)
frontier_tau = np.linspace(0.01, 1, 5)

mu_estimators = [
    ('NAIVE', naive_estimator),
    ('MARTINGAL', martingal_estimator),
    ('LR', ml_estimator_builder(best_models['LR'])),
    ('ARIMA', ml_estimator_builder(best_models['ARIMA'])),
    ('RF', ml_estimator_builder(best_models['RF'])),
]

trivial_portfolios = [
    ('UNIFORM', uniform_portfolio_builder()),
    ('MOST_RISKY', single_asset_portfolio_builder(idx_most_risky)),
    ('LESS_RISKY', single_asset_portfolio_builder(idx_less_risky)),
    ('BEST_RETURN', single_asset_portfolio_builder(idx_best_return)),
    ('WORST_RETURN', single_asset_portfolio_builder(idx_worst_return)),
]
```

```
for t in tqdm.trange(total_runs):
    # prepare data
    idx_history = min_history_leng + t
    idx_future = idx_history + days_shift - 1
    df_history = df_returns_test.iloc[:idx_history]
    df_future = df_returns_test.iloc[idx_future]

    # estimate cov, common for all models
    cov_hat = df_history.cov().values

    # estimate mu using list of models
    frontiers = []
    for name, mu_estimator in mu_estimators:
        mu_hat = mu_estimator(df_history)

        # evaluate each portfolio in frontier for currnet model
        roi_frontier = frontier_evaluator(
            mu_hat, cov_hat, df_future.values, frontier_tau)
        frontiers.append(roi_frontier)
    results_frontier.append(frontiers)

    # evaluate trivial strategies
    trivials = []
    for name, w in trivial_portfolios:
        roi = np.dot(w, df_future.values).item()
        trivials.append(roi)
    results_trivial.append(trivials)

frontier_means = np.nanmean(results_frontier, axis=0)
frontier_stds = np.nanstd(results_frontier, axis=0)

trivial_means = np.mean(results_trivial, axis=0)
trivial_stds = np.std(results_trivial, axis=0)

print(frontier_means, frontier_stds)

fig, ax = plt.subplots()
for m, s, (label, _) in zip(
    frontier_means, frontier_stds, mu_estimators):
    ax.plot(s, m, marker='o', label=label)

ax.scatter(trivial_stds, trivial_means, color='grey')
for s, m, (name, _) in zip(
    trivial_stds, trivial_means, trivial_portfolios):
    ax.text(s + 0.003, m, name)

ax.set_xlabel('ROI std')
ax.set_ylabel('ROI mean')
ax.set_xlim(0, 0.2)
ax.legend()
fig.savefig('../images/result_frontiers.png')

print(np.isnan(results_frontier).mean(axis=0))

# mean ROI
(pd.DataFrame(
    frontier_means, columns=[f'{t: .2f}' for t in frontier_tau],
    index=[n for n, _ in mu_estimators]) * 1000
    ).to_latex('../tables/roi_mean.tex',
            caption='Средние ROI $\cdot 10^3$',
            float_format='%.4f',
            position='h',
            label='tab:roi_mean',
            )

# std ROI
(pd.DataFrame(
    frontier_stds, columns=[f'{t: .2f}' for t in frontier_tau],
    index=[n for n, _ in mu_estimators]) * 100
    ).to_latex('../tables/roi_std.tex',
            caption='Стандартное отклонение ROI $\cdot 10^2$',
            float_format='%.4f',
            position='h',
```

```
            label='tab:roi_std'
            )

(pd.DataFrame({
    'mean ROI $\cdot 10^3$': trivial_means * 1000,
    'std ROI $\cdot 10^2$': trivial_stds * 100,
    },
    index=[n.replace('_', ' ') for n, _ in trivial_portfolios])
).to_latex('../tables/trivial_rois.tex',
            caption='Тривиальные портфели',
            float_format='%.4f',
            position='h',
            label='tab:trivial_rois',
            )
```