

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
МОСКОВСИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
(Государственный университет)  
ФАКУЛЬТЕТ УПРАВЛЕНИЯ И ПРИКЛАДНОЙ МАТЕМАТИКИ  
Кафедра «Системные исследования»

Выпускная квалификационная работа по направлению 03.03.01  
«Прикладные математика и физика»

**Анализ и применение алгоритмов и  
методов машинного обучения в  
трейдинге**

Работу выполнил студент 905-ой группы

Емцев Илья

Научный руководитель

д.э.н. Орлова Елена Роальдовна



## АННОТАЦИЯ

Данный дипломный проект представляет собой комплексный анализ трейдинга и исследования алгоритмов и методов машинного обучения и их применение в трейдинге. В данной работе:

- Рассматриваются необходимые сведения по трейдингу (типы и виды трейдинга, финансовые инструменты и т.п.). Обсуждаются различные показатели оценки прибыли и потерь (*Profit&Loss*). Отдельное внимание уделено трейдингу с плечом (*trading with leverage*) и рассмотрены его основные плюсы и минусы
- Проводится детальная работа с данными (где брать данные, как формируются цены, что такое скорректированная цена закрытия). Также подробно рассмотрена проблема пропущенных значений в данных
- Рассмотрена основная метрика в трейдинге - доходность (*return*). Очень детально рассмотрена проблема "не нормальности" распределения доходности и приведены различные методы аппроксимации и сравнения её распределения
- Подробно рассмотрены различные стратегии, которые применяются в трейдинге (*buy and hold, SMA crossover, simple contrarian/momentum, mean reversion*). Определены методы оценки качества той или иной стратегии (*backtesting, forward testing*)
- Рассмотрены основные задачи машинного обучения (кластеризация, регрессия, классификация). Рассмотрены подробно алгоритмы, как линейная/логистическая регрессия, *Gaussian Mixture Models (GMM)*, *EM*-алгоритм. Рассматривается полносвязная нейронная сеть (*DNN*). Затем все эти модели применяются для анализа финансовых данных и прогнозирования трендов на рынке

# Оглавление

<b>Введение</b>	<b>4</b>
<b>1 Трейдинг - основные понятия и сведения</b>	<b>6</b>
1.1 Вступление . . . . .	6
1.2 Виды трейдинга: особенности и отличия . . . . .	7
1.3 Ключевые термины и показатели <i>Profit &amp; Loss</i> . . . . .	10
1.4 Трейдинг с плечом : оценка перформанса, плюсы и минусы . . . . .	12
1.5 Вознаграждение( <i>Reward</i> )/Риск( <i>Risk</i> ) . . . . .	15
1.5.1 Оценка вознаграждения ( <i>reward</i> ) . . . . .	15
1.5.2 Виды доходности( <i>return</i> ) . . . . .	16
<b>2 Анализ данных</b>	<b>19</b>
2.1 Получение данных для анализа . . . . .	19
2.2 Цена акции . . . . .	20
2.3 Что делать с пропущенными данными . . . . .	22
2.4 Скорректированная цена закрытия . . . . .	24
2.5 Анализ распределения доходности . . . . .	26
2.5.1 <i>QQ – plot (quantile – quantile plot)</i> . . . . .	27
2.5.2 Коэффициенты ассиметрии и эксцесса . . . . .	29
2.5.3 Доверительный интервал и корреляция . . . . .	30
<b>3 Применение машинного обучения в трейдинге</b>	<b>33</b>
3.1 Постановка задачи машинного обучения . . . . .	33
3.2 Стратегии в трейдинге . . . . .	34

3.2.1	<i>Backtesting</i> и <i>Forward Testing</i>	35
3.2.2	Стратегия: <i>Buy and Hold</i>	35
3.2.3	Стратегия: <i>SMA Crossover</i>	36
3.2.4	Стратегия: <i>Simple Contrarian/Momentum</i>	39
3.2.5	Стратегия: <i>Mean – Reversion</i>	41
3.3	<i>Gaussian Mixture Models (GMM)</i> и <i>EM</i> -алгоритм	42
3.3.1	<i>Gaussian Mixture Models (GMM)</i>	42
3.3.2	<i>EM</i> - алгоритм	44
3.3.3	Поиск распределения доходности ( <i>return</i> )	45
3.4	Линейная регрессия	47
3.4.1	Постановка задачи	47
3.4.2	Необходимые предположения	48
3.4.3	Метрики качества	49
3.4.4	Дополнение	49
3.4.5	Показатели альфа и бетта	51
3.5	Логистическая регрессия	53
3.5.1	Постановка задачи	53
3.5.2	Метрики качества	55
3.6	Полносвязная нейронная сеть ( <i>DNN</i> )	57
3.6.1	<i>DNN</i> и работа нейрона	57
3.6.2	Обучение нейронной сети ( <i>DNN</i> )	59
3.6.3	<i>Dropout/Batch Normalization</i>	61
3.7	Предсказание доходности с помощью <i>ML</i> и <i>DL</i>	62
3.7.1	Линейная/Логистическая регрессия	62
3.7.2	Полносвязная нейронная сеть ( <i>DNN</i> )	67
<b>Заключение</b>		<b>71</b>
<b>Приложения</b>		<b>73</b>
Биржевые настройки ( <i>Orders</i> ) и их типы		73
Netting и Hedging		74
<b>Литература</b>		<b>76</b>

## ВВЕДЕНИЕ

Финансовые данные включают в себя информацию о ценах акций, объемах торговли, финансовых отчетах компаний, новостях и других факторах, которые могут влиять на их развитие. Анализ этих данных может помочь нам понять прошлые тренды и выявить закономерности, которые могут повлиять на будущие изменения их цен. Однако стоит помнить, что финансовые рынки сложны и подвержены множеству факторов, которые могут влиять на их поведение. Даже с использованием машинного обучения и статистических моделей невозможно предсказать с абсолютной точностью, как будут развиваться тренды рынка, поскольку рынки могут быть подвержены волатильности, макроэкономическим факторам, политическим событиям и другим переменным, которые трудно учесть в моделях. Поэтому важно понимать, что инвестирование в акции всегда связано с риском, и нет гарантии получения прибыли. Хотя анализ финансовых данных и использование статистических моделей могут помочь нам принимать более обоснованные инвестиционные решения, всегда следует оценивать риски и диверсифицировать портфель, чтобы снизить потенциальные потери. В итоге, использование машинного обучения и статистических методов в анализе финансовых данных может быть полезным инструментом для предсказания трендов акций, но они не являются магическим решением и требуют осторожного и осознанного подхода при принятии инвестиционных решений.

Изначально планировалось сделать что-то менее теоретическое и более практическое, но с большим погружением в тему я стал понимать, что без хорошего теоретического введения не обойтись. Более того, я планировал закончить данный диплом построением бота, который бы мог по командам алгоритма самостоятельно трейдить (алготрейдинг). В итоге я рассматривал брокеров *FXCM* и *OANDA*, поскольку они предоставляли *API* подключение напрямую к платформе. Поэтому мною был создан демо-аккаунт и были проведены

детальные исследования первого брокера, однако в начале апреля он решил убрать такую функцию и соответственно трейдить уже не получилось. Что касается второго брокера, то он не доступен на территории Российской Федерации. Не смотря на это, большинство терминов и понятий, затрагивающих алготрейдинг были приведены на примере брокера *OANDA*. Также к концу мая появился способ создать демо-аккаунт и на брокере *OANDA*, что позволит в будущем развить эту тему до алготрейдинга. Поэтому в данной работе присутствуют не только методы машинного обучения, но также детально рассмотрены различные понятия, которые помогают понять, что трейдинг из себя представляет и какую прибыль или убытки он может приносить.

Также важно отметить, что различные биржи и торговые платформы могут иметь некоторые отличия в терминологии и спецификации. Поэтому перед использованием любой из них в реальной торговле, рекомендуется ознакомиться с правилами и функционалом конкретной платформы или биржи.

Приложу ссылку на *github* со всем кодом, разбитый по главам : [yemtssev](#)

# Глава 1

## ТРЕЙДИНГ - ОСНОВНЫЕ ПОНЯТИЯ И СВЕДЕНИЯ

### 1.1 Вступление

Начнём с определения трейдинга и базовых терминов, которые часто используются и без которых дальнейшее повествование будет бессмысленным

**Определение 1.1.1.** Трейдинг (*Trading*) - это процесс покупки и продажи финансовых инструментов на финансовых рынках с целью получения прибыли от разницы в ценах.

**Определение 1.1.2.** Финансовые инструменты (*Financial Instruments*) - включают в себя разнообразные активы, такие как акции, облигации, товары, валюты, инвестиционные фонды, ценные бумаги. Каждый финансовый инструмент имеет свои уникальные характеристики, права и обязательства, которые определяются его конкретными условиями и правилами торговли.

Также введу ещё три важных понятия для трейдинга:

**Определение 1.1.3.** Ликвидность (*Liquidity*) - это мера способности актива или рынка быть быстро и эффективно купленным или проданным без значительного влияния на его цену. Она описывает наличие достаточного объема покупателей и продавцов на рынке, готовых совершать сделки.

Ликвидность является важным аспектом для трейдеров и инвесторов, поскольку она влияет на возможность эффективно осуществлять сделки, минимизировать потери при входе и выходе с рынка, а также обеспечивать возможность быстрой конвертации активов в

наличные средства.

**Определение 1.1.4.** Взаимозаменяемость (*Fungibility*) - это свойство, которое описывает возможность заменить одну единицу актива или товара другой единицей того же типа без изменения его стоимости или характеристик. Это означает, что каждая единица актива является взаимозаменяемой с другой единицей того же актива.

Взаимозаменяемость играет важную роль в обеспечении ликвидности рынка и облегчении торговли. Если активы или товары не являются взаимозаменяемыми, это может привести к сложностям при определении их стоимости или при проведении сделок.

**Определение 1.1.5.** Волатильность (*Volatility*) - это мера степени изменчивости цен на финансовые инструменты в определенный период времени. Она отражает колебания и количественно измеряет разброс ценовых значений вокруг их среднего значения.

Высокая волатильность означает, что цены на активы изменяются сильно и быстро, как вверх, так и вниз. Низкая волатильность, наоборот, указывает на более стабильные и малые колебания цен. Волатильность может быть измерена различными статистическими показателями, такими как стандартное отклонение и другие.

## 1.2 Виды трейдинга: особенности и отличия

На данный момент существуют два основных вида трейдинга: дневная торговля и долгосрочное инвестирование. Разберём что это и в чём их основные отличия. Параллельно будем вводить новые понятия и сразу их применять.

**Определение 1.2.1.** Дневной трейдинг (*Day Trading*) - это стратегия торговли на финансовых рынках, при которой трейдер открывает и закрывает позиции внутри одного торгового дня.

**Определение 1.2.2.** Долгосрочное инвестирование (*Long – Term Investing*) - это стратегия инвестирования, которая предполагает приобретение активов на долгий период времени, обычно на несколько лет или более.

Основная цель дневного трейдинга - получение прибыли на основе краткосрочных ценовых колебаний, в то время, как основная цель долгосрочных инвестиций - рост капитала на протяжении длительного времени

**Определение 1.2.3.** Диверсификация портфеля (*Portfolio Diversification*) — это распределение капитала по разным активам. Если по одним доходность упадёт, то по другим вырастет, и шанс получить прибыль будет выше

Долгосрочные инвестиционные портфели очень диверсифицированы, включая акции, облигации, сырьевые товары и многое другое. Решения принимаются на уровне портфеля. В дневной торговле диверсификация не является обязательным требованием, и трейдеры могут сосредоточиться на одном или нескольких инструментах. Цель состоит в том, чтобы получать прибыль на уровне инструмента.

**Определение 1.2.4.** Длинная позиция (*Long Position*) - означает покупку актива с целью его дальнейшей продажи в надежде на рост его стоимости

**Определение 1.2.5.** Короткая позиция (*Short Position*) - означает продажу актива, которым трейдер не владеет, с целью его обратной покупки в будущем по более низкой цене

Дневная торговля предполагает открытие как длинных, так и коротких позиций, чтобы извлечь выгоду из роста и падения цен. Цель состоит в том, чтобы получать прибыль от краткосрочных колебаний цен. С другой стороны, долгосрочное инвестирование в первую очередь выигрывает от долгосрочного роста рынка, дивидендов и повышения цен.

**Определение 1.2.6.** Деривативы (*Derivatives*) - это контракты с ограниченным сроком действия, которые представляют собой финансовые инструменты, чья стоимость зависит от базового актива.

Торговля деривативами позволяет трейдерам спекулировать на изменение цены базового актива без необходимости владения самим активом.

**Определение 1.2.7.** Контракт на разницу цен (*CDF*) - это популярный тип дериватива, который используется для торговли на бирже.

**Определение 1.2.8.** Кредитное плечо (*Financial Leverage*) - это соотношение денег трейдера к общему объему средств, которыми он торгует. По правовой сущности это услуга брокера, предоставляющая средства, превышающие собственные в несколько раз.

Когда дело доходит до инструментов, дневная торговля часто включает деривативные финансовые инструменты (*CFD*), которые позволяют торговать с кредитным плечом. С другой стороны, долгосрочные инвестиции обычно предполагают физическое владение акциями и не полагаются на кредитное плечо. Риск, как правило, от низкого до умеренного

из-за диверсификации портфеля.

**Определение 1.2.9.** Технический анализ (*Technical Analysis*) - это метод анализа финансовых рынков, который основывается на исследовании и интерпретации исторических ценовых данных и объемов торговли активов.

**Определение 1.2.10.** Фундаментальный анализ (*Fundamental Analysis*) - это метод анализа финансовых рынков, который фокусируется на изучении фундаментальных факторов, влияющих на стоимость активов, таких как экономические показатели, финансовые отчеты, конкурентная среда, политические события и т.д.

Дневная торговля подразумевает использование технического анализа - использование графиков, индикаторов и других технических инструментов для выявления повторяющихся паттернов, трендов и сигналов, которые могут помочь прогнозировать будущую ценовую динамику активов. С другой стороны, долгосрочное инвестирование подразумевает использование фундаментального анализа - анализ финансовых показателей компании, оценку ее бизнес-модели, рыночную позицию и другие факторы, которые могут влиять на ее будущую прибыльность и рост.

Подводя итог, можно сказать, что основные различия между дневной торговлей и долгосрочным инвестированием заключаются в их целях, временных горизонтах, подходах к диверсификации, источниках доходности и используемых инструментах.

	Дневная торговля	Долгосрочные инвестиции
Временной горизонт	Короткий	Долгий
Цель	Краткосрочная прибыль	Долгосрочный рост капитала
Частота торговли	Высокая	Низкая
Время уделяемое торговле	Целый день	Ограниченнное время
Анализ	Технический	Фундаментальный
Стратегия	Использование краткосрочных трендов	Покупка и удержание акций
Риск	Высокий	Умеренный
Навыки	Анализ графиков и быстрые реакции	Анализ финансовой отчётности и долгосрочное планирование

Также ещё стоит упомянуть, что существует два типа трейдинга:

**Определение 1.2.11.** Spot-трейдинг (*Spot Trading*) - это форма торговли финансовыми инструментами, при которой сделки заключаются немедленно или в краткосрочной перспективе с непосредственной покупкой или продажой физических финансовых инструментов.

**Определение 1.2.12.** Торговля деривативами (*Derivative Trading*) - это форма торговли

финансовыми инструментами, основанная на контрактах, известных как деривативы.

Spot-трейдинг и торговля деривативами представляют разные подходы к торговле финансовыми инструментами. Spot-трейдинг основан на непосредственной покупке или продаже активов по текущей рыночной цене, в то время как торговля деривативами позволяет трейдерам спекулировать на изменение цены базового актива без необходимости физического владения активом. Оба этих подхода имеют свои особенности и риски, и трейдеры могут выбирать между ними в зависимости от своих целей и предпочтений.

### 1.3 Ключевые термины и показатели *Profit & Loss*

В торговле на финансовых рынках, таких как валютный рынок *FOREX*, часто используются некоторые ключевые термины (рис. 1.1).

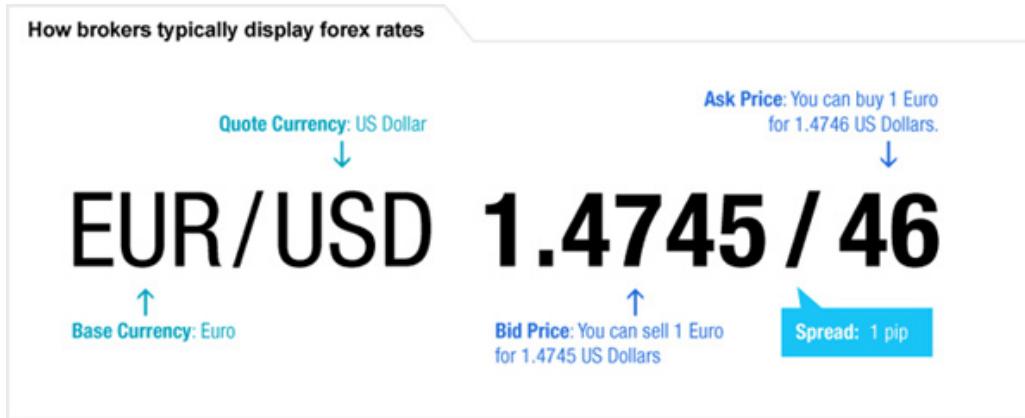


Рис. 1.1: Ключевые термины на финансовых рынках

**Определение 1.3.1.** Базовая валюта (*Base Currency*) - это валюта, которая является основным объектом покупки или продажи валютной пары.

**Определение 1.3.2.** Котируемая валюта (*Quote Currency*) - это валюта, в которой выражается цена базовой валюты.

Например, в паре *EUR/USD* базовой валютой является евро, а котируемой доллар

**Определение 1.3.3.** Цена продажи (*Ask Price*) - в контексте валютной пары, это цена, по которой трейдер может купить котируемую валюту, платя базовой валютой.

**Определение 1.3.4.** Цена покупки (*Bid Price*) - в контексте валютной пары, это цена,

по которой трейдер может продать котируемую валюту, получая базовую валюту

**Определение 1.3.5.** Спред (*Spread*) - это разница между ask price и bid price. Она представляет собой комиссионные затраты на торговлю и является прибылью для брокера. Чем меньше спред, тем более ликвидная торговая пара.

**Определение 1.3.6.** Пункт (*Pip*) - это наименьшее изменение цены, которое может произойти на рынке.

Обычно он представляет собой четвертый десятичный знак в цене. Например, если цена *EUR/USD* изменяется с 1.4745 до 1.4746, это означает, что произошло изменение на 1 пункт (pip) = 0.0001

Теперь посмотрим, как посчитать прибыль от торговли. Представим, что мы совершили покупку 10<sup>5</sup> единиц инструмента *EUR/USD* - открыли длинную позицию (*buy market*) с уверенностью, что цена увеличится и затем продали их - открыли короткую позицию (*close market*) (рис. 1.2)

Trades	Orders	Positions	Activity	+						
Ticket	Type	Market	Units	Price	Half Spread Cost	Profit (USD)	Commission	Balance	Date / Time	
20	Close Trade	EUR/USD	100,000	1.17791	-9.5000	16.00	0.00	100016.00	11.8.2020, 15:48:49	
19	Market Order	EUR/USD	100,000	-	-	-	-	-	11.8.2020, 15:48:49	
18	Buy Market	EUR/USD	100,000	1.17775	-8.0000	0.00	0.00	100000.00	11.8.2020, 15:46:10	
17	Market Order	EUR/USD	100,000	-	-	-	-	-	11.8.2020, 15:46:10	

Рис. 1.2: Транзакции (*Order Book*)

Построим следующую таблицу, которая отражает основные показатели для оценки прибыли и потерь (*Profit & Loss*) (рис. 1.3):

**Определение 1.3.7.** Стоимость сделки (*Trade Value*) - это общая стоимость сделки, которая рассчитывается путем умножения объема торговли на цену сделки.

$$Trade Value = Units * Price$$

Например, если вы покупаете 1000 единиц базовой валюты по цене 1.25, то общая стоимость сделки составит  $1000 \cdot 1.25 = 1250$  единиц котируемой валюты

**Определение 1.3.8.** Прибыль (*Profit*) - это разница между стоимостью сделки при открытии и закрытии.

$$Profit = Trade Value_{close} - Trade Value_{open}$$

Time	Type	Instrument	Units	Price (per Unit)	Trade Value (Units * Price)	Profit (USD)	Profit (Pips)	Pip Value (USD)	Balance	Spread (Pips)	Half Spread Costs (USD)	Mid Price (hypothetical)
15:48:49	Sell/Close	EUR/USD	100 000	\$1,17791 Bid	\$117 791,00	\$16,00	1,60	\$10,00	\$100 016,00	1,90	-\$9,50	\$1,17801
15:46:10	Buy	EUR/USD	100 000	\$1,17775 Ask	\$117 775,00	-	-	\$10,00	\$100 000,00	1,60	-\$8,00	\$1,17767

Рис. 1.3: Основные показатели для оценки (*Profit & Loss*)

**Определение 1.3.9.** Половина стоимости спреда (*Half Spread Costs*) - это половина разницы между *ask price* и *bid price*. Половина стоимости спреда представляет собой приблизительную комиссию, которую трейдер платит за открытие и закрытие позиции.

$$\text{Half Spread Costs} = -\frac{\text{Spread} \cdot \text{Pip Value}}{2}$$

**Определение 1.3.10.** Средняя цена (*Mid Price*) - это среднее значение между *ask price* и *bid price*. Она используется для определения центральной цены, которая может быть использована в расчетах или для отображения текущего рыночного значения.

$$\begin{cases} \text{Mid Price} = \text{Bid Price} + \frac{\text{Spread} \cdot 0.0001}{2} \\ \text{Mid Price} = \text{Ask Price} - \frac{\text{Spread} \cdot 0.0001}{2} \end{cases}$$

## 1.4 Трейдинг с плечом : оценка перформанса, плюсы и минусы

Рассмотрим трейдинг с плечом и без плеча, оценим их перформанс, введём основные показатели и термины. Построим таблицу, которая иллюстрирует такие понятия, как компенсационный взнос и плечо (рис. 1.4):

Margin & Leverage		
Leverage	30	: 1
Margin		3,33%
Margin Required		\$3 925,83
Margin Available		\$100 000,00

Рис. 1.4: Компенсационный взнос и плечо

**Определение 1.4.1.** Компенсационный взнос (*Margin*) - это сумма денежных средств или активов, которую трейдер должен предоставить в качестве обеспечения для открытия и поддержания позиции на рынке. Компенсационный взнос позволяет трейдерам торговать

на рынке с использованием заемных средств, увеличивая свой потенциальный доход и риск.

$$Margin = \frac{100}{Leverage}\%$$

**Определение 1.4.2.** Требуемый компенсационный взнос (*Margin Required*) - это минимальная сумма компенсационного взноса, которую трейдер должен иметь на своем торговом счете для открытия и поддержания определенной позиции.

$$Margin Required = Margin \cdot Trade Value_{open}$$

Брокеры устанавливают требуемый компенсационный взнос в процентном отношении от полной стоимости позиции. Например, если требуемый компенсационный взнос составляет 5% и стоимость позиции равна 10,000\$, то трейдер должен иметь на счету минимум 500\$ в качестве требуемого компенсационного взноса.

**Определение 1.4.3.** Доступный компенсационный взнос (*Margin Available*) - это компенсационный взнос, которая остается у трейдера после открытия позиции. Доступный компенсационный взнос позволяет трейдеру открывать дополнительные позиции или поддерживать уже открытые позиции

$$Margin Available = All Margin - Margin Required$$

Трейдер должен следить за доступным компенсационным взносом, так как если она становится недостаточной для поддержания открытых позиций, брокер может потребовать дополнительного пополнения (*margin call*) или автоматически закрыть часть или все позиции трейдера (*close out*) для предотвращения отрицательного баланса на счете.

Теперь построим таблицу, которая отражает основные показатели для оценки перформанса (*Performance Attribution*) (рис. 1.5):

<b>Performance Attribution</b>				
	<u>Profit (USD)</u>	<u>Profit (pips)</u>	<u>Return (unlevered)</u>	<u>Return (levered)</u>
Gross Profit	\$33,50	3,35	0,0284%	0,8533%
Trading Costs	-\$17,50	-1,75	-0,0149%	-0,4458%
<b>Net Profit</b>	<b>\$16,00</b>	<b>1,60</b>	<b>0,0136%</b>	<b>0,4076%</b>
				30

Рис. 1.5: Оценка перформанса

**Определение 1.4.4.** Валовая прибыль (*Gross Profit*) - это общая прибыль от торговли, рассчитанная как разница между выручкой от продажи активов или закрытия позиций и затратами на открытие или покупку этих активов. Она не учитывает дополнительные издержки, такие как комиссии, налоги или прочие торговые расходы.

$$Gross Profit = (Mid Price_{close} - Mid Price_{open}) \cdot Units$$

**Определение 1.4.5.** Торговые расходы (*Trading Costs*) - это дополнительные издержки, связанные с проведением торговых операций. Они могут включать комиссии брокера, спреды, налоги на сделки и другие платежи, связанные с торговлей на рынках.

$$Trading Costs = Half Spread Costs_{open} + Half Spread Costs_{close}$$

**Определение 1.4.6.** Чистая Прибыль (*Net Profit*) - это прибыль, полученная после вычета всех торговых расходов (комиссий, налогов и прочих издержек) из валовой прибыли. Чистая прибыль отображает реальную финансовую выгоду от торговых операций после учета всех расходов.

$$Net Profit = Gross Profit + Trading Costs$$

**Определение 1.4.7.** Доходность (*Return*) - показывает, какую прибыль или убыток в процентном отношении получил трейдер или инвестор от своего начального капитала.

$$\begin{cases} Return(unlevered) = \frac{Profit/Loss}{Trade Value_{open}} \cdot 100\% \\ Return(levered) = \frac{Profit/Loss}{Margin Required} \cdot 100\% \end{cases}$$

Как мы видим, торговля с плечом (*Trading with Leverage*) увеличивают не только нашу потенциальную валовую прибыль (*Gross Profit*), но также увеличивают и торговые расходы (*Trading Costs*). Поэтому мы можем не только сильно увеличить нашу чистую прибыль (*Net Profit*), но также сильно увеличить риски. В нашем примере, наша доходность потенциально возрастает, насколько нам позволяет плечо (увеличивается в 30 раз), но и возрастают потенциальные торговые расходы (увеличивается также в 30 раз).

Теперь поговорим ещё о некоторых показателях оценки и контроля риска (рис. 1.6)

**Определение 1.4.8.** Нереализованная прибыль/убыток (*Unrealized P/L*) - это прибыль или убыток, который трейдер получит, если он закроет свою позицию в данный момент.

$$\begin{cases} Unrealized P/L = (Current Bid Price - Open Ask Price) \cdot Units \\ Unrealized P/L = (Open Bid Price - Current Ask Price) \cdot Units \end{cases}$$

Open Position											
Time	Type	Instrument	Units	Price (per Unit)	Trade Value (Units * Price)	Balance	Margin Used	Current Bid Price	Unrealized P/L	Net Asset Value/ Margin Closeout Value	Margin Available
15:46:10	Buy	EUR/USD	100 000	\$1,17775 Ask	\$117 775,00	\$5 000,00	\$3 925,83	\$1,14700	-\$3 075,00	\$1 925,00	\$0,00

Рис. 1.6: Margin Closeout

**Определение 1.4.9.** Закрытие компенсационного взноса (*Margin Closeout*) - это уровень стоимости активов на торговом счете, при котором брокер автоматически закрывает открытые позиции трейдера из-за недостаточного компенсационного взноса для поддержания этих позиций. Значение закрытия рассчитывается с учетом требуемого компенсационного взноса и размера открытых позиций. Например, если  $Margin \ Closeout \leq 0.5$ , то происходит автоматическое закрытие

$$Margin \ Closeout \ Value = Margin \ Used + Unrealized \ P/L$$

$$Margin \ Closeout = \frac{Margin \ Closeout \ Value}{Margin \ Used}$$

## 1.5 Вознаграждение(*Reward*)/Риск(*Risk*)

В трейдинге вознаграждение (*reward*) и риск (*risk*) являются двумя важными понятиями, связанными с оценкой потенциальной прибыли и потерь при проведении сделок на финансовых рынках

**Определение 1.5.1.** Вознаграждение (*Reward*) - представляет собой потенциальную прибыль, которую трейдер может получить при успешной сделке (положительная средняя доходность)

**Определение 1.5.2.** Риск (*Risk*) - представляет собой потенциальные потери, которые трейдер может понести, если сделка оказывается неудачной (волатильность доходности)

В финансах/инвестициях действует закон: "Большой риск должен вознаграждаться большой доходностью"

### 1.5.1 Оценка вознаграждения (*reward*)

Рассмотрим две метрики оценки вознаграждения, которые в отличии от простого среднего арифметического более интуитивные и легко интерпретируемые.

**Определение 1.5.3.** *Investment Multiple* - это показатель, который определяет, во сколько раз увеличилась или уменьшилась исходная инвестиция по истечении определенного периода времени.

$$\text{Multiple} = \frac{\text{Last Price}_t}{\text{Initial Price}_t}, \quad t - \text{заданный промежуток времени}$$

*Investment Multiple* позволяет оценить вознаграждение инвестиции в абсолютных значениях. Минус этой метрики в том, что без указания определённого периода метрика становится бесмысленной

**Определение 1.5.4.** *CAGR (Compound Annual Growth Rate)* - это годовая средняя ставка роста инвестиции за определенный период времени.

$$\text{CAGR} = \text{multiple}^{\frac{1}{n}} - 1, \quad n - \text{количество лет инвестирования(не обязательно целое)}$$

*CAGR* позволяет сравнить доходность инвестиций на основе их годовой ставки роста и является полезным инструментом для сравнения инвестиций с разными временными горизонтами и помогает оценить долгосрочную доходность.

## 1.5.2 Виды доходности(*return*)

Доходность (*Return*) - является важным показателем для оценки эффективности трейдинговых стратегий и инвестиций. Он позволяет трейдерам и инвесторам измерять процентную доходность от своих операций и сравнивать результаты с другими альтернативами инвестирования или торговли. Поэтому рассмотрим разные виды доходности:

**Определение 1.5.5.** (Обычная) доходность (*Simple Return*) - показывает, какую прибыль или убыток в процентном отношении получил трейдер или инвестор за промежуток времени  $[t - 1, t]$ .

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

где  $P_t$  - цена в момент времени  $t$ ,  $P_{t-1}$  - цена в момент времени  $t - 1$

**Определение 1.5.6.** *Geometric mean return* - представляет собой метод оценки средней годовой доходности инвестиций на основе геометрического среднего

$$\text{geometric mean} = \text{multiple}^{\frac{1}{n}} - 1, \quad n - \text{количество периодов времени (дней/часов/минут)}$$

как мы видим, геометрическое среднее очень схоже с понятием *CAGR*. Важное замечание: геометрическое среднее всегда меньше или равно арифметическому среднему, поэтому является более полезным показателем, чем среднее арифметическое. Также используя

обычное среднее арифметическое мы не сможем получить последнюю цену в отличии от геометрического среднего.

Для дальнейшего повествования введём несколько экономических терминов:

**Определение 1.5.7.** Present Value ( $PV$ ) - представляет собой текущую стоимость или цену будущих денежных потоков с учетом ставки дисконтирования.

$PV$  позволяет сравнивать различные денежные потоки, возникающие в разные периоды времени, приводя их к общей базе - текущему моменту времени.

**Определение 1.5.8.** Future Value ( $FV$ ) - представляет собой ожидаемую стоимость или сумму денежных потоков в будущем на основе текущих вложений или инвестиций.

$FV$  позволяет определить, сколько денег будет иметься в будущем на основе текущих инвестиций или денежных потоков.

**Определение 1.5.9.** Капитализация (*Compounding*) - частота, с которой проценты начисляются или добавляются к начальной сумме вклада или займа.

$$\begin{cases} FV = PV \cdot \left(1 + \frac{r}{m}\right)^{n \cdot m}, & \text{дискретная капитализация} \\ FV = PV \cdot e^{r \cdot n}, & \text{непрерывная капитализация} \end{cases}$$

тут  $r$  – годовая ставка дисконтирования (процентная ставка),  $m$  – количество периодов капитализации,  $n$  – количество лет

**Определение 1.5.10.** Эффективная процентная ставка (*Effective Interest Rate*) - представляет собой процентную ставку, которая учитывает капитализацию и приводит будущие денежные потоки ( $FV$ ) к их текущей стоимости ( $PV$ )

$$\begin{cases} r_{\text{эфф}} = \left(\frac{FV}{PV}\right)^{\frac{1}{n}} - 1, & \text{дискретная капитализация} \\ r_{\text{эфф}} = e^r - 1, & \text{непрерывная капитализация} \end{cases}$$

Цены же финансовых инструментов меняются (примерно) непрерывно, поэтому скорее логичнее использовать логарифмическую доходность (*log return*)

**Определение 1.5.11.** Логарифмическая доходность (*log return*) - является больше приближенной к реальности типом доходности, которая обладает свойством аддитивности, большей устойчивости к выбросам, большей склонности к нормальному распределению.

$$\text{Log } R_t = \log\left(\frac{P_t}{P_{t-1}}\right) = \log(1 + R_t)$$

где  $P_t$  - цена в момент времени  $t$ ,  $P_{t-1}$  - цена в момент времени  $t - 1$ ,  $R_t$  - доходность (обычная) в момент времени  $t$

Также, используя логарифмическую доходность, мы можем использовать простое среднее арифметическое, которое будет работать как средне егеометрическое для обычной доходности, т.е теперь средняя доходность стала информативной. Также стоит заметить, что т.к доходность имеет очень маленькую величину, то  $\log(1 + R_t) \approx R_t$ . В дальнейшем буду работать с логарифмической доходностью и буду считать её основной.

## Глава 2

### АНАЛИЗ ДАННЫХ

#### 2.1 Получение данных для анализа

Теперь возникает вопрос, откуда брать актуальные финансовые данные, поскольку они отличаются от наборов данных, используемых в обычном машинном обучении. В отличие от доступного множества данных для задач, таких как классификация изображений на датасетах *MNIST* или *ImageNet*, финансовые данные требуют специального подхода. Исторические данные, собранные 10 лет назад, маловероятно помогут предсказать события, происходящие в ближайшие недели или месяцы. Поэтому для достоверных финансовых прогнозов всегда требуются недавние и обновляемые данные.

Я выделил несколько источников, где можно взять финансовые данные:

- *Quandl* - это открытая платформа, которая предлагает широкий спектр финансовых и экономических наборов данных. Платформа предоставляет доступ к историческим ценам на акции, экономическим показателям, данным по сырьевым товарам и многому другому. В основном, доступ к наиболее интересным наборам данным - платный.
- *pandas – datareader* - это библиотека *python*, с помощью которой пользователи могут извлекать данные из широкого спектра источников, включая поставщиков финансовых данных, таких как *Yahoo Finance*, *Google Finance* и *Quandl* и т.п.
- *Yahoo Finance (yfinance)* - это пакет на *python*, который предоставляет удобный способ доступа не только к историческим рыночным данным *Yahoo Finance*, но и к данным в режиме реального времени. Он позволяет пользователям извлекать такие данные, как историческая информация о ценах, данные о дивидендах, финансовые отчеты, данные об

опционах и индексные данные.

- Скомпилированные данные с интернета - это могут быть данные с платформы *Kaggle*, курсов *Udemy*, *Coursera*, *Udacity* и другие.

Именно последнюю библиотеку я выбрал в качестве основной для получения данных в этой главе, в связи с простотой использования и хорошей интеграцией с *pandas*. Однако в других главах будут использоваться и другие наборы данных, о чём будет написано.

Для наглядного анализа в некоторых последующих секциях в этой главе - загрузим библиотеку *yfinance* и выгрузим данные акций "Google" за последние три года (с мая 2020-го года по май 2023-го года).

## 2.2 Цена акции

Теперь мы рассмотрим полученные финансовые данные, уделяя особое внимание данным о ценах на акции и информации об объемах продаж.

**Определение 2.2.1.** Цена акции - это числовая стоимость одной единицы акции, выраженная в определенной валюте.

Цена акции может изменяться в течение дня в зависимости от множества факторов, таких как новости о компании, экономические условия, политические события, изменения в отрасли и другие влияющие факторы. Сама цена акции формируется в результате встречи спроса и предложения. Когда покупатели заинтересованы в приобретении акций данной компании и готовы платить определенную сумму за них (*ask price*), а продавцы хотят продать акции по определенной цене (*bid price*), то происходит сделка и цена акции обновляется. Точка, в которой продавец и покупатель договорились о цене - точка равновесия (*equilibrium*)

**Определение 2.2.2.** Равновесие (*Equilibrium*) - это состояние, при котором рыночное предложение и спрос уравновешивают друг друга, и в результате цены становятся стабильными.

Если же людей, желающих купить акцию, больше, чем тех, кто готов ее продать (покупателей больше, чем продавцов) - цена акции вырастет из-за возросшего спроса. С другой стороны, если больше людей продают данную акцию, чем покупают ее, то ее цена снижается. Также, цена акции отражает ожидания и восприятие рынком стоимости компании. Высокая цена акции может указывать на то, что инвесторы считают компанию успешной

и перспективной, а низкая цена акции может свидетельствовать о негативном отношении рынка к компании.

Поэтому, при выгрузке данных из-за изменчивости цены, мы увидим табличное представление, где каждая строка соответствует определенному периоду времени, который обычно составляет день, но также может быть еженедельным, ежемесячным или даже внутри-дневным (1 минута, 1 час и другие) (рис. 2.1).

	Open	High	Low	Close	Adj Close	Volume	Name
Date							
2020-05-01	66.425003	67.603500	65.550003	66.030502	66.030502	41450000	GOOG
2020-05-04	65.411499	66.383003	64.949997	66.339996	66.339996	30080000	GOOG
2020-05-05	66.896004	68.696999	66.873001	67.555496	67.555496	33030000	GOOG
2020-05-06	68.084503	68.556000	67.364502	67.364998	67.364998	24308000	GOOG
2020-05-07	68.296997	68.879997	67.763496	68.627998	68.627998	27952000	GOOG

Рис. 2.1: Обзор данных *Google* за 3 года

Табличное представление состоит из таких столбцов, как

- *OPEN* - Цена открытия (относится к цене акций на начало периода)
- *CLOSE* - Цена закрытия (представляет собой цену акций на конец периода)
- *HIGH* - Наиболее высокая цена (указывает на максимальную цену, достигнутую за период)
- *LOW* - Наиболее низкая цена (указывает на минимальную цену за период)
- *VOLUME* - Объем акция (относится к общему количеству акций, которые были проданы за указанный период времени. Это означает, что некоторые акции были проданы одной стороной и куплены другой)

Кроме того, может существовать дополнительный столбец под названием *AdjustedClose*, который мы обсудим позже. Одним из преимуществ такого обобщения является то, что когда мы упорядочиваем цены акций по дням, мы получаем временной ряд, который служит фундаментальной структурой данных.

Также в трейдинге часто используют представление данной таблицы в виде свечи - один из основных типов графиков, используемых в техническом анализе на финансовых рынках, включая трейдинг. Он получил свое название из-за внешнего вида свечей, которыми представляются данные. Каждая свеча на графике представляет определенный период времени

(например, 1 минута, 1 час, 1 день и т.д.) и содержит информацию о цене открытия, закрытия, наивысшей и наименьшей цене за этот период. Форма и цвет свечи указывают на то, как цена изменилась за данный период (рис. 2.2).

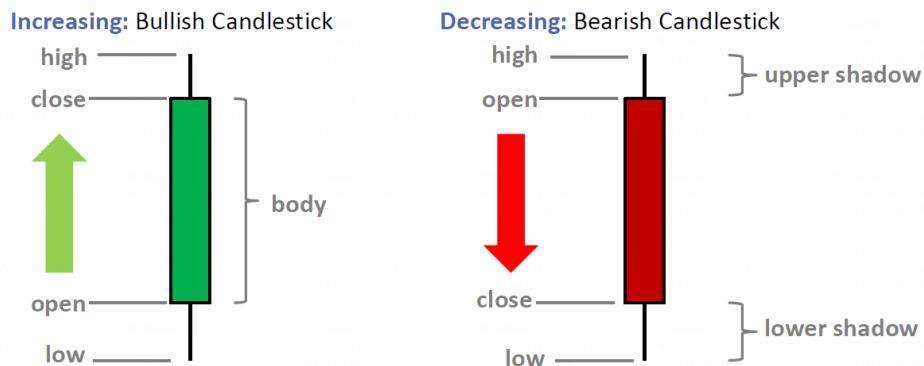


Рис. 2.2: Представление данных в виде свечи

Основные элементы свечи включают:

**Тело свечи** - отображает диапазон между ценами открытия и закрытия. Если цена закрытия выше цены открытия, тело свечи обычно заполнено или закрашено цветом (обычно белым или зеленым). Если цена закрытия ниже цены открытия, тело свечи обычно пустое или закрашено другим цветом (обычно черным или красным).

**Верхняя тень** - эта линия представляет наивысшую цену за данный период. Она расширяется вверх от тела свечи.

**Нижняя тень** - эта линия представляет наименьшую цену за данный период. Она расширяется вниз от тела свечи.

Свечные графики могут использоваться для анализа и прогнозирования движений цен на рынке с помощью технических индикаторов (о части из них пойдёт речь в следующей главе). Трейдеры и аналитики могут использовать различные модели и комбинации свечей для выявления трендов, сигналов покупки и продажи, а также других паттернов, которые могут указывать на будущее направление цен.

## 2.3 Что делать с пропущенными данными

Сначала ответим на вопрос, почему вообще появляются пропущенные данные при выгрузке финансовых данных. Есть несколько причин, почему так может происходить:

1. Данные внезапно исчезают. Например, когда одна компания покупает другую. Например, если взять акции "Delivery Club" то поскольку данная компания была куплена "Yandex" в сентябре 2022 года, то мы увидим, что данные резко пропадут (рис. 2.3.a)
  2. Данные внезапно появляются. Может быть такое, что мы по каким-то причинам хотим добавить несколько компаний на один график, однако одна из компаний начала своё существование позже начала временного промежутка. Поэтому на графике мы увидим, что данных будет недостаточно (рис. 2.3.b)
  3. Данные иногда пропадают, потом опять появляются. Это в основном происходит с маленькими компаниями с низкой ликвидностью, т.е у них часто бывают дни, когда не проходит трейда и соответственно нет никакой цены (рис. 2.3.v)
  4. Могут быть ошибки при выгрузке данных, либо сами данные очень плохие (такое часто происходит в индустрии, при работе с большим набором данных)
- Отдельно поговорим, какие методы заполнения пропущенных финансовых данных существуют и какие методы использовать не целесобразно.

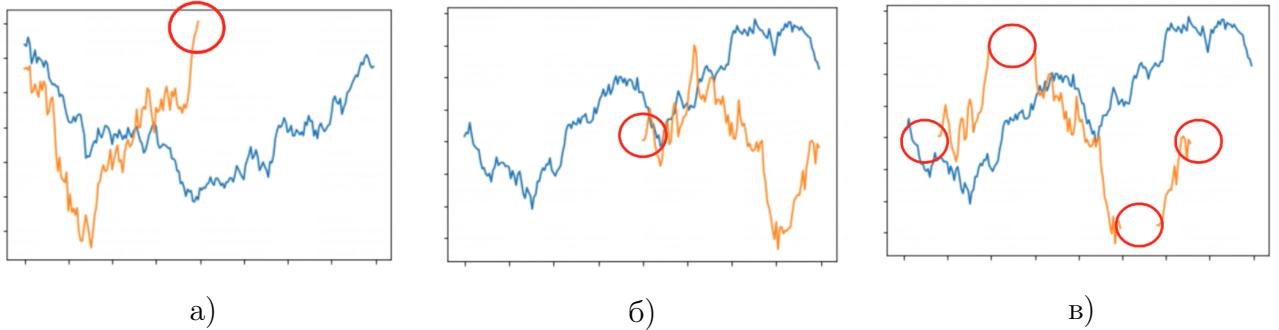


Рис. 2.3: Разные виды пропущенных данных

Многие впервые очередь попробуют использовать линейную интерполяцию (соединяют левую точку с правой линейно), однако такой подход неверный, поскольку тут мы заглядываем как бы в будущее, о котором в моменте нахождения в левой точке мы не знаем. Вместо линейной интерполяции можно использовать следующее:

1. *Forward Filling* - метод продолжения данных самим последним известным значением до пропажи данных. Используется для продолжения данных от старых к новым (рис. 2.4)
2. *Backward Filling* - аналогично Forward Filling, однако используется для продолжения данных от новых к старым (рис. 2.4)

Мы в основном будем использовать именно *Forward Filling* и часто в стратегиях (о которых будет рассказано в следующей главе)

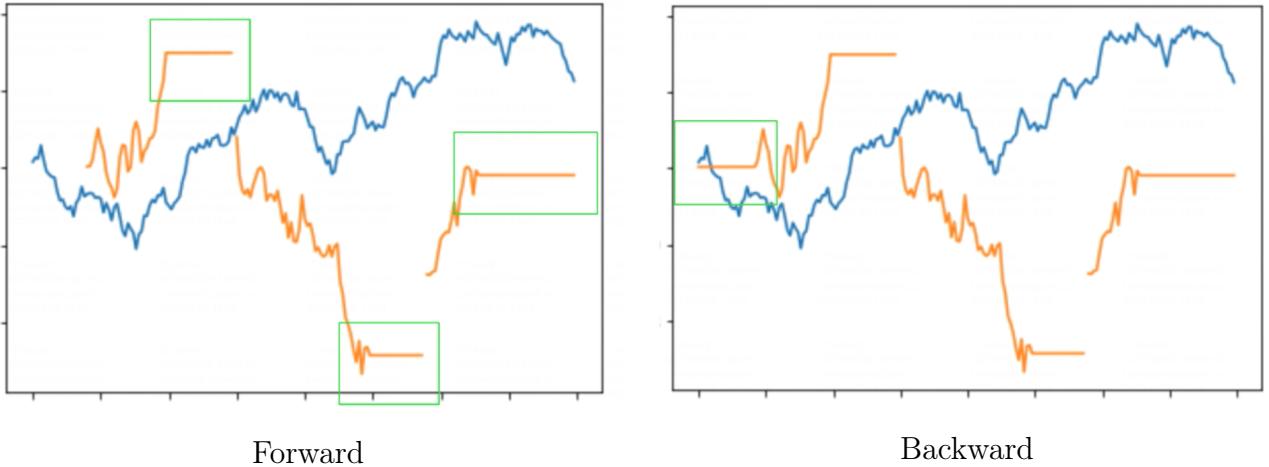


Рис. 2.4: Виды заполнения пропущенных данных

Отдельно отмечу дни как выходные и праздники. Они не считаются пропущенными данными и если по каким-то причинам в данных они просутствуют, то их стоит просто удалить. Также, стоит заметить, что в трейдинге разница между пятницей и понедельником всего 1 день, а не 2 дня, как в реальной жизни.

## 2.4 Скорректированная цена закрытия

Теперь разберём дополнительный столбец *Adjusted Close Price*, но сперва введём некоторые новые понятия.

**Определение 2.4.1.** Разделение акций (*Stock Split*) - это процесс, при котором компания увеличивает общее количество своих акций, путём деления одной акции на несколько меньших.

Например, в случае 2-за-1 ( $2 - to - 1$ ) разделения, каждая акция компании делится на две новых акции, и общее количество акций удваивается. При этом цена акций соответственно снижается в два раза. Разделение акций обычно проводится с целью повышения привлекательности для инвесторов и увеличения ликвидности, т.к проведение разделения позволяет снизить цену, делая компанию более привлекательной для широкого круга инвесторов. В случае разделения акций, цена закрытия резко упадёт, поэтому на графике мы увидим резкий спад. В итоге, формула для доходности (*return*) даст неправильный результат, если использовать просто цену закрытия. Обычно при выгрузке данных, *Adjusted Close Price* присваивает значение цены закрытия, которая была у последних строк (самые недавние

цены закрытия). Рассмотрим (рис. 2.5), на котором дважды происходило разделение акций. По *Adjusted Close Price* построен синий график, в то время как желтый график иллюстрирует просто *Close Price*. В этом случае, нужно считать доходность (*return*) по синему графику, чтобы посчитать верно (в зелёных квадратиках показано, насколько цена желтого графика отличается от синего в определённые промежутки времени).

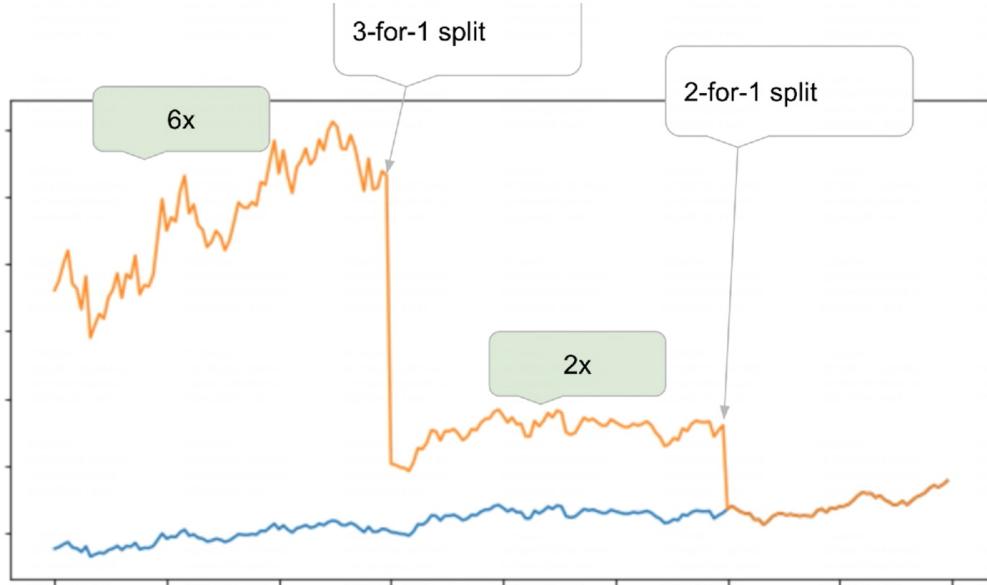


Рис. 2.5: Пример *Stock Split* и *Adjusted Close Price*

Однако не только разделение акций влияет на *Adjusted Close Price*. Введём понятие дивидендов:

**Определение 2.4.2.** Дивиденды (*Dividends*) - представляют собой денежные выплаты, которые компания делает своим акционерам из своей прибыли.

Это часто происходит ежеквартально или ежегодно. Дивиденды выплачиваются акционерам в зависимости от количества акций, которыми они владеют. Обычно дивиденды являются процентным отношением к текущей цене акций. Например, если компания объявляет дивиденды в размере 0.50\$ на акцию, и у вас есть 100 акций, то вы получите 50\$ в качестве дивидендов. При наличии дивидендов, формула для доходности (*return*) будет следующей:

$$R_t = \frac{P_t - P_{t-1} + D_t}{P_{t-1}}, \text{ где } P_t - \text{ цена акции в момент времени } t, D_t - \text{ выплаченные дивиденды в момент времени } t$$

При наличии дивидендов, *Adjusted Close Price* высчитывается как разница между ценой закрытия и выплаченными дивидендами в момент времени *t*:

$$Adjusted Close Price_t = P_t - D_t$$

Теперь можно дать определение скорректированной цене закрытия (*Adjusted Close Price*):

**Определение 2.4.3.** Скорректированная цена закрытия (*Adjusted Close Price*) - это цена акции, учтенная после прохождения корпоративных действий, таких как разделение акций или выплата дивидендов.

Получается, что скорректированная цена закрытия помогает обеспечить сравнимость цен акций на протяжении времени. Стоит заметить, что т.к мы в дальнейшем будем использовать библиотеку *yfinance*, то в ней цена закрытия (*Close Price*) уже работает как *Adjusted Close Price* при наличии разделения акций (*Stock Split*), т.е единственная разница между ними - это дивиденды. Поэтому в дальнейшем предлагаю игнорировать дивиденды и смотреть на просто цену закрытия.

## 2.5 Анализ распределения доходности

Далее, мы будем подробно смотреть на показатель доходности (*return*) (логарифмической), т.к она является одним из важнейших показателей оценки эффективности торговли. Более того, поскольку одной из важных задач в финансах является задача понять, какое распределение у доходности, я хотел бы попытаться максимально приблизить её распределение. Более того, зная распределение, мы можем посчитать такие важные параметры как ожидаемое значение (*EV*) и волатильность.

**Определение 2.5.1.** Ожидаемое значение (*EV*) - это ожидаемое среднее значение для инвестиций в какой-то момент в будущем. Инвесторы используют *EV* для оценки целесообразности инвестиций, часто в зависимости от их относительной рискованности. Вычисляется оно как обычное матожидание (в нашем случае  $X_i = Return_i$ ):

$$EV = \sum X_i \cdot P(X_i)$$

Теперь, на основе данных, мы постараемся понять, как выглядит распределение. Визуализируем данные, которые мы выгрузили ранее (рис. 2.6):

Как мы видим, цена закрытия и скорректированная цена закрытия совпадают, поскольку "Google" не выплачивает дивиденды. Также в данных отсутствуют пропущенные данные. Теперь посчитаем доходность (*return*) и посмотрим на её распределение (рис. 2.7.а). Видим, что доходность выглядит примерно, как нормальное распределение, поэтому постараемся им его приблизить (рис. 2.7.б).



Рис. 2.6: Визуализация цены закрытия акций "Google" за последние 3 года

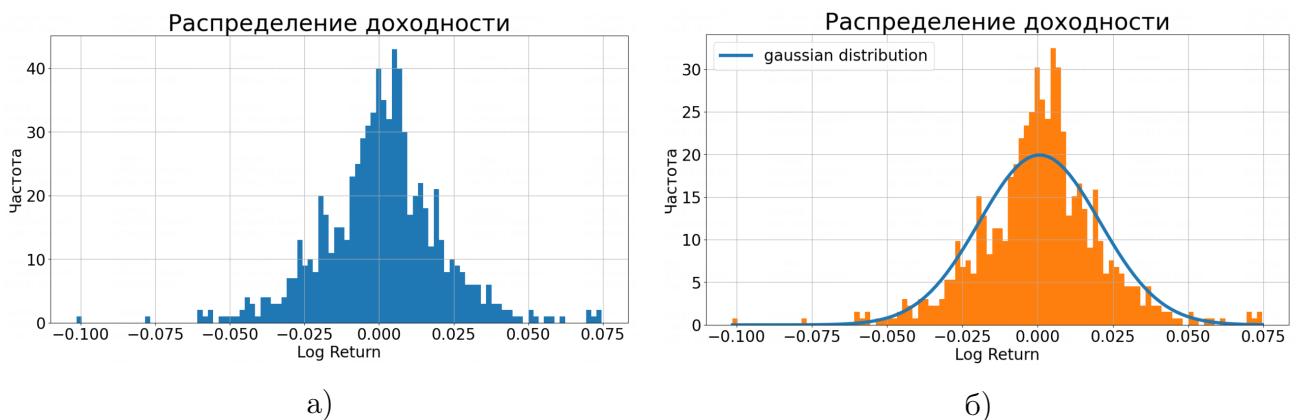


Рис. 2.7: Распределение доходности

Видно, что нормальное распределение достаточно плохо приближает доходность, но в чём основное отличие и есть ли распределение, которое приблизит доходность лучше? Для ответа на этот вопрос, рассмотрим некоторые методы из математической статистики.

### 2.5.1 $QQ - plot$ ( $quantile - quantile plot$ )

Одним из способов проверки нормальности распределения является  $QQ - plot$  ( $probability plot$  или  $normal probability plot$ , если распределение, с которым мы сравниваем - нормальное)

**Определение 2.5.2.**  $QQ - plot$  ( $quantile - quantile plot$ ) - является графическим инструментом для визуальной оценки соответствия распределения данных теоретическому рас-

пределению. Он используется для проверки, насколько хорошо данные соответствуют теоретическому распределению, такому как нормальное распределение.

*QQ*-график строится следующим образом: на оси  $X$  отображаются квантили теоретического распределения, а на оси  $Y$  - квантили данных. Квантиль - это значение, ниже которого попадает определенная доля данных. Анализ *QQ*-графика позволяет выявить отклонения данных от теоретического распределения. Если точки на графике отклоняются от диагонали, это указывает на то, что данные не соответствуют теоретическому распределению. Например, если точки сгруппированы вокруг прямой, но имеют изгибы или выпадающие значения, это может указывать на асимметрию данных или наличие выбросов (рис. 2.8)

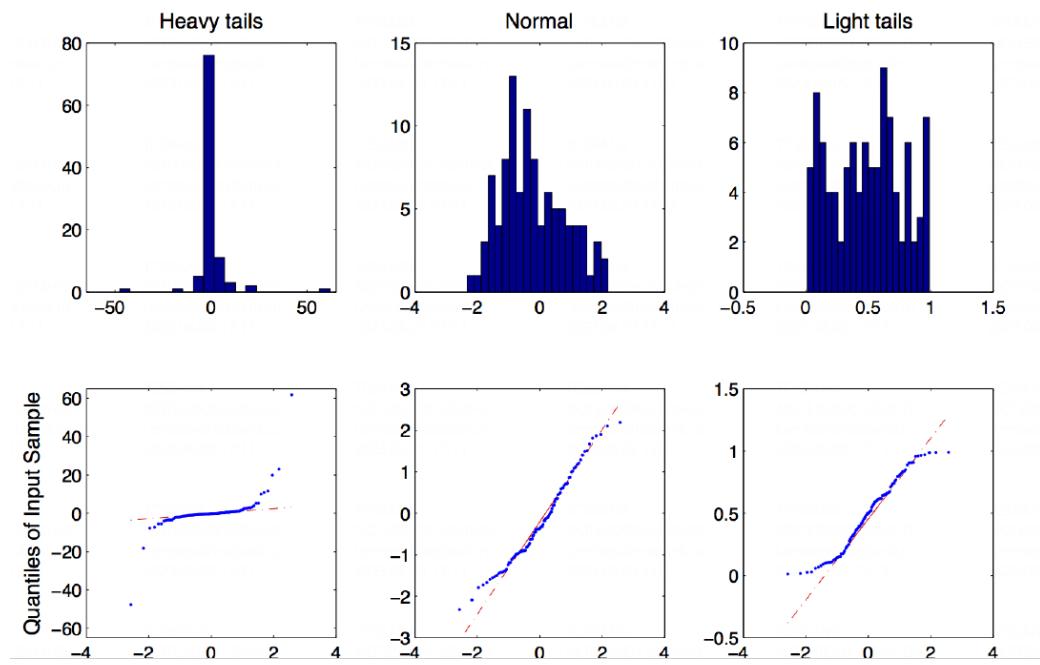


Рис. 2.8: Визуализация различных отклонений

Теперь построим *QQ* – *plot* для наших данных (рис. 2.9.a). Как мы видим, распределение доходности акций "Google" не похоже на нормальное, т.к имеет более тяжёлые хвосты. Поэтому, раз мы не смогли приблизить доходность нормальным распределением, попробуем приблизить распределением Стьюдента (*t-distribution*), которое выглядит как нормальное, но как раз имеет более тяжёлые хвосты (рис. 2.9.б и рис. 2.9.в). В целом, распределение Стьюдента достаточно не плохо смогло приблизить наше распределение доходности. Поэтому можем считать, что доходность имеет *t*-распределение.

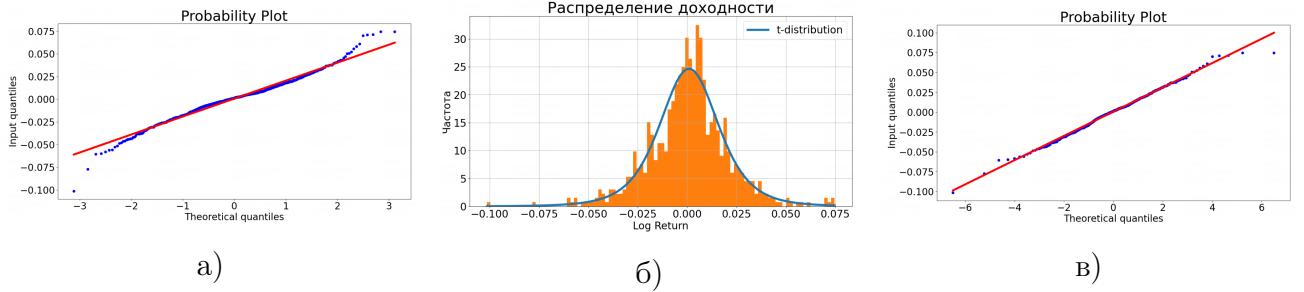


Рис. 2.9: Разное приближение

### 2.5.2 Коэффициенты асимметрии и эксцесса

Введём ещё два понятия, которые помогут нам описать форму распределение доходности.

**Определение 2.5.3.** Асимметрия (*Skewness*) - измеряет степень и направление отклонения распределения от симметрии. Она показывает, насколько данные смещены влево или вправо относительно центрального значения. Асимметрия может быть положительной, отрицательной или нулевой (рис. 2.9.а).

$$Skewness = \frac{E[(X-EX)^3]}{DX^3}$$

1. Положительная асимметрия - распределение имеет длинный хвост справа. Графически это может выглядеть как смещение распределения влево.
2. Отрицательная асимметрия - распределение имеет длинный хвост слева. Графически это может выглядеть как смещение распределения вправо.
3. Нулевая асимметрия - распределение симметрично относительно среднего значения.

**Определение 2.5.4.** Эксцесс (*Kurtosis*) - измеряет степень остроты или плоскости пика распределения по сравнению с нормальным распределением. Он указывает на то, насколько данные имеют выбросы или очень крутые/плоские хвосты (рис. 2.9.б).

$$Kurtosis = \frac{E[(X-EX)^4]}{DX^4} - 3$$

1. Положительный эксцесс - распределение имеет более крутые пики и тяжелые хвосты, чем нормальное распределение. Такие данные имеют большую концентрацию значений вокруг среднего и большую вероятность появления выбросов.
2. Отрицательный эксцесс - распределение имеет менее крутые пики и более легкие хвосты, чем нормальное распределение. Такие данные имеют менее выраженный пик и меньшую вероятность появления выбросов.

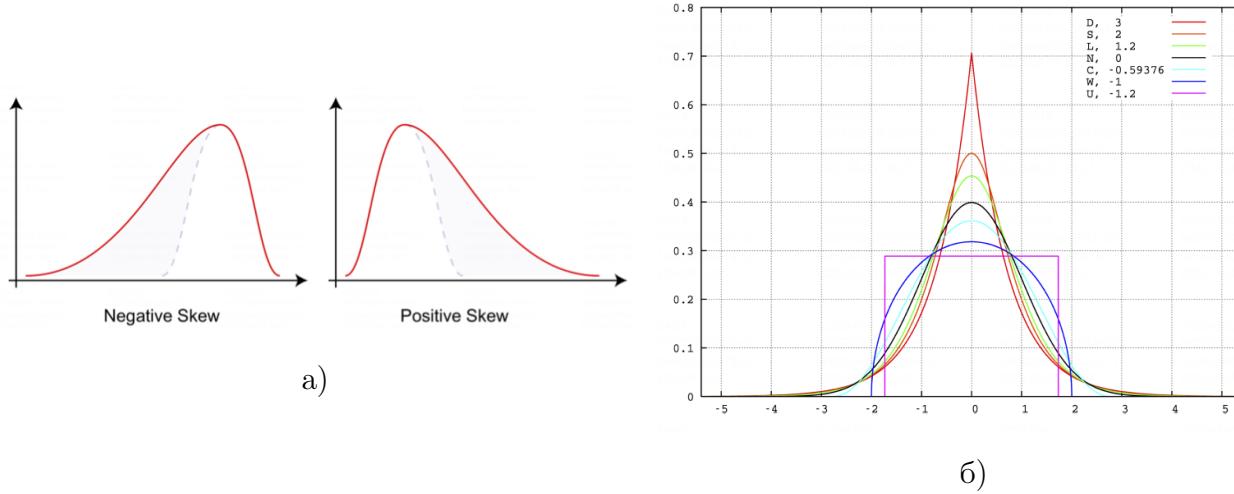


Рис. 2.10: Виды коэффициентов

**3.** Нулевой эксцесс - распределение имеет пики и хвосты, сравнимые с нормальным распределением.

Наша же цель - избежать отрицательного коэффициента асимметрии, поскольку если он отрицательный, то у нас получается данные смещены влево (т.е туда, где доходность отрицательна), т.к в таком случае могут быть потенциально очень большие потери. Аналогично, нам нужно, чтобы коэффициент эксцесса не был большим, т.к тогда мы опять же имеем потенциал получить очень большие потери (т.к будут очень тяжёлые хвосты у распределения). Для наших данных получились следующие коэффициенты:

$$Skewness = -0.115$$

$$Kurtosis = 2.083$$

Как мы видим, наши данные смещены влево и имеют достаточно тяжёлые хвосты, что достаточно плохо и говорит нам, что при торговле акциями "Google" нужно быть аккуратными и много не рисковать.

### 2.5.3 Доверительный интервал и корреляция

Посмотрим на ещё два важных понятия, как доверительный интервал и коэффициент корреляции.

**Определение 2.5.5.** Доверительный интервал:

$$P(\theta \in [C_L, C_U]) \geq 1 - \alpha,$$

$1 - \alpha$  — уровень доверия,

$C_L, C_U$  — нижний и верхний доверительные пределы

**Неверная интерпретация:** неизвестный параметр лежит в пределах построенного доверительного интервала с вероятностью  $1 - \alpha$ .

**Верная интерпретация:** при бесконечном повторении процедуры построения доверительного интервала на аналогичных выборках в  $100 \cdot (1 - \alpha)\%$  случаев он будет содержать истинное значение  $\theta$ .

Если считать, что доходность (*return*) распределена нормально с известной дисперсией, то 95% доверительный интервал для оценки средней доходности ( $R$ ) будет высчитываться по следующей формуле:

$$R = \bar{x} \pm 1.96 \cdot \frac{\sigma}{\sqrt{n}}$$

где  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$

Если считать, что доходность (*return*) распределена нормально с неизвестной дисперсией, то 95% доверительный интервал для оценки средней доходности ( $R$ ) будет высчитываться по следующей формуле:

$$R = \bar{x} \pm 1.96 \cdot \frac{s}{\sqrt{n}}$$

где  $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$  - выборочная несмешённая дисперсия,  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$

В нашем случае подходит вторая формула (если считать, что всё-таки распределение доходности примерно нормальное). Тогда доверительный интервал для наших данных будет выглядеть следующим образом (рис. 2.11)

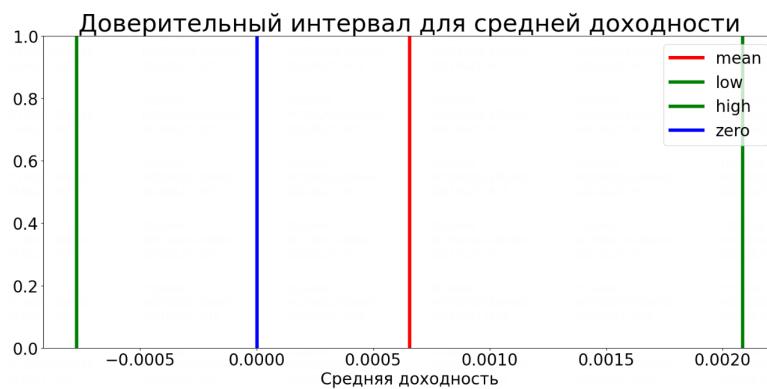


Рис. 2.11: Доверительный интервал для средней доходности наших данных

Как мы видим, потенциально в среднем мы можем иметь отрицательную доходность при торговле акциями "Google" что подтверждает, что доходность не может быть распределена нормально и смешена влево.

**Определение 2.5.6.** Корреляция (*Correlation*) - статистическая взаимосвязь между случайными величинами. Корреляция не является достаточным условием для причинно-следственной связи.

Есть разные коэффициенты корреляции (Пирсона, Спирмена, Кендалла). В данной дипломной работе мы будем рассматривать коэффициент корреляции Пирсона.

**Определение 2.5.7.** Коэффициент корреляции Пирсона ( $\rho_{XY}$ ) случайных величин  $X$  и  $Y$  - мера силы **линейной** корреляции между ними:

$$\rho_{XY} = \frac{(X - EX)(Y - EY)}{\sqrt{DX} \cdot \sqrt{DY}}$$

Пусть имеется простая выборка пар  $(X_i, Y_i)$ , где  $i = 1, \dots, n$ . Тогда выборочный коэффициент корреляции Пирсона:

$$\rho = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \cdot \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

Корреляция в финансовом анализе помогает понять, как связаны акции разных инструментов/компаний между собой. Например, когда цены на золото увеличиваются, доллар падает  $\Rightarrow$  у них отрицательная корреляция. Данный термин нам понадобится в дальнейшем.

## Глава 3

---

### ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ В ТРЕЙДИНГЕ

#### 3.1 Постановка задачи машинного обучения

Теперь перейдём к алгоритмам и методам машинного обучения и рассмотрим различные стратегии, которые используются в трейдинге (которые также используются в виде признаков). Перед этим, дадим классическое, общее определение машинного обучения:

**Определение 3.1.1.** Говорят, что компьютерная программа обучается при решении какой-то задачи из класса  $T$ , если ее производительность, согласно метрике  $P$ , улучшается при накоплении опыта  $E$ .

Далее в разных сценариях под  $T$ ,  $P$ , и  $E$  подразумеваются совершенно разные вещи. Под опытом  $E$  понимаются данные, и в зависимости от этого алгоритмы машинного обучения могут быть поделены. Основные подходы в машинном обучении включают в себя:

- Обучение с учителем (*Supervised Learning*): В этом подходе модель обучается на основе помеченных данных, где каждому входному примеру соответствует правильный выходной ответ. Алгоритм стремится научиться предсказывать правильные ответы на новых данных, основываясь на обучающих данных.
- Обучение без учителя (*Unsupervised Learning*): В этом случае модель обучается на непомеченных данных, без прямой обратной связи. Алгоритм ищет скрытые структуры или закономерности в данных, выделяет группы или кластеры объектов, основываясь на их сходстве.
- Обучение с подкреплением (*Reinforcement Learning*): В этом подходе модель обучается

взаимодействуя с окружающей средой и получая обратную связь в виде награды или штрафа. Алгоритм стремится научиться принимать оптимальные действия в заданной среде с целью максимизации общей награды.

Среди самых популярных задач  $T$  в машинном обучении:

- Классификация: Эта задача заключается в присвоении объектам определенных меток классов на основе их характеристик. Например, классификация электронных писем на "спам" и "не спам" или классификация изображений на различные категории.
- Регрессия: В регрессии модель предсказывает числовую величину на основе входных данных. Например, предсказание цены недвижимости на основе ее характеристик или прогнозирование спроса на товары.
- Кластеризация: В задаче кластеризации алгоритм группирует объекты в подмножества (кластеры) на основе их сходства. Например, кластеризация клиентов в маркетинговых исследованиях для выявления сегментов рынка.
- Обнаружение аномалий: Здесь модель ищет объекты или события, выделяющиеся из нормальных паттернов. Это может быть полезно, например, для обнаружения мошеннических операций на кредитных картах или выявления необычного поведения в системе безопасности.
- Рекомендательные системы: Эти системы используют алгоритмы машинного обучения для предоставления рекомендаций пользователю на основе его предпочтений и истории. Примером может быть рекомендация фильмов или товаров в интернет-магазинах. Наконец, третья абстракция в определении машинного обучения - это метрика оценки производительности алгоритма  $P$ . Такие метрики различаются для разных задач и алгоритмов и будут рассмотрены позже.

## 3.2 Стратегии в трейдинге

Рассмотрим основные стратегии в трейдинге и способы их тестирования (данная секция относится к разделу машинного обучения, т.к в этой секции рассматриваются скользящие статистики, методы тестирования, т.е всё, что будет использоваться в стратегиях, которые подкреплены машинным обучением):

### 3.2.1 Backtesting и Forward Testing

**Определение 3.2.1.** *Backtesting* - это процесс проверки торговой стратегии на исторических данных для определения ее прибыльности и эффективности

В *backtesting* используются исторические данные, чтобы смоделировать прошлое движение рынка и применить стратегию к этим данным. Таким образом, можно оценить, какая прибыль могла бы быть получена при использовании данной стратегии в прошлом. *Backtesting* позволяет трейдерам проверить и уточнить свои торговые идеи, а также определить статистические показатели.

**Определение 3.2.2.** *Forward Testing*, также известное как *out – of – sample testing* - является продолжением *backtesting* и в отличие от него использует реальные данные, которые становятся доступными после проведения *backtesting*.

*Forward Testing* позволяет оценить эффективность стратегии на новых, не использовавшихся в *backtesting* данных. *Forward Testing* помогает оценить устойчивость и пригодность стратегии для текущих рыночных условий и проверить, сохраняется ли эффективность стратегии в реальном времени.

Комбинирование обоих методов может помочь трейдерам принимать более оптимальные решения и повысить вероятность успеха при применении торговых стратегий.

В основном мы будем использовать абсолютную метрику оценки перформанса : *cumulative log return*

**Определение 3.2.3.** *Cumulative Log Return* - метрика абсолютного перформанса, показывающая доходность за какой-то промежуток времени

$$\text{cumulative log return} = \sum_t \log \text{return}_t$$

Также для сравнения нескольких финансовых инструментов друг с другом нужно будет приводить их к одинаковой частотности(т.к иначе в среднем доходность та же, но чем больше частота, тем больше волатильность). Часто приводят к годовым показателям, т.е умножают среднее число на 252 дня (примерное количество трейдинговых дней в году).

### 3.2.2 Стратегия: Buy and Hold

**Определение 3.2.4.** *Buy and Hold* (покупка и удержание) - это простая и популярная стратегия в трейдинге, которая предполагает покупку активов и долгосрочное их удержание без активных операций покупки и продажи в течение длительного времени.

Основная идея стратегии *Buy and Hold* заключается в том, чтобы верить в долгосрочный рост рынка и сохранять позицию на протяжении продолжительного периода времени, независимо от краткосрочных колебаний рынка.

Преимущества стратегии *Buy and Hold*:

- Простота: стратегия не требует активного участия трейдера и постоянного мониторинга рынка. Это может быть привлекательным для инвесторов, которые не хотят тратить много времени и энергии на торговлю.
- Долгосрочный потенциал роста: исторически рынки акций обычно имели тенденцию к росту на длинных временных интервалах, поэтому владение акциями на протяжении длительного периода может привести к значительному росту капитала.
- Уменьшение транзакционных издержек: Поскольку стратегия не требует активного торгового вмешательства, это позволяет снизить транзакционные издержки, связанные с частыми операциями покупки и продажи активов.

Однако, стоит учитывать и некоторые ограничения и риски стратегии *Buy and Hold*:

- Неэффективное использование капитала: поскольку средства остаются вложенными в одни активы на протяжении длительного времени, может возникнуть потеря возможности использовать капитал для других инвестиций, которые могли бы быть более прибыльными (альтернативные издержки).
- Необходимость выбора правильных активов: выбор правильных активов для покупки и удержания важен для успешной реализации стратегии. Неверный выбор может привести к низкой доходности или потере капитала.

В дальнейшем мы будем сравнивать перформанс любой стратегии с этой стратегией, чтобы понимать, прибыльна ли новая стратегия или лучше ничего не внедрять.

### 3.2.3 Стратегия: *SMA Crossover*

**Определение 3.2.5.** *SMA (Simple Moving Average)* - представляет собой простое арифметическое среднее значение цен за указанный период.

**Определение 3.2.6.** *SMA Crossover Strategy* - это торговая стратегия, основанная на сигналах, которые возникают при пересечении двух или более простых скользящих средних (*SMA*)

Эта стратегия основывается на предположении, что пересечение двух *SMA* может указывать на изменение тренда на рынке. Обычно используются две *SMA* с разными пери-

одами - более короткий и более длинный. Например, часто используются 50-дневная и 200-дневная.

Когда более короткий *SMA* пересекает более длинный *SMA* сверху вниз, это сигнализирует о возможном начале нисходящего тренда и может быть сигналом для продажи актива. Наоборот, когда более короткий *SMA* пересекает более длинный *SMA* снизу вверх, это может указывать на возможное начало восходящего тренда и служить сигналом для покупки актива (рис. 3.1.а). Обозначим покупку как 1, а продажу как  $-1$  и изобразим как мы будем действовать в нашем случае (рис. 3.1.б). Данная стратегия даёт нам понимание как действовать в следующий день/час/минуту в зависимости от частотности.

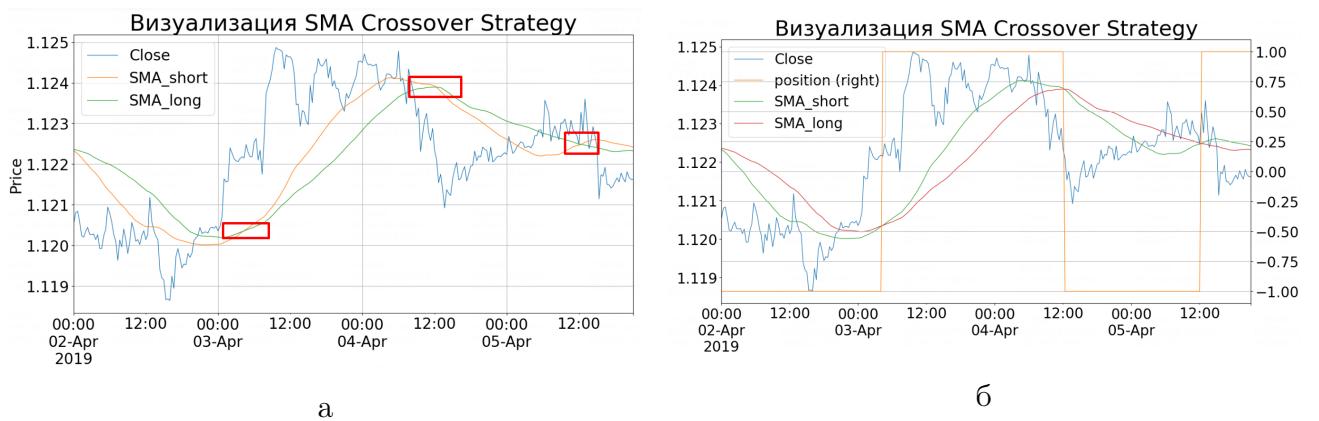


Рис. 3.1: Визуализация стратегии *SMA Crossover*

Будем использовать данные финансового инструмента *EUR/USD* с частотой в 20 минут с 2017-го года по 2020-ый год и проведём *backtesting* с коротким *sma* равный 50 и длинным *sma* равный 150 (рис. 3.2.а). Видно, что наша стратегия работает хуже обычной стратегии (купить в 2017 и продать в 2020). Построим функцию, которая будет подбирать подходящие короткую и длинную *sma* с помощью фреймворка *optuna*. Я буду ориентироваться на следующую метрику - доходность от стратегии за всё время (т.е просто сумма всех доходностей от стратегии). Функция за 1000 итераций выдала следующие значения:

$$short = 61, \ long = 88$$

Таким образом, подставив данные значения, мы получим, что с такими значениями скользящих средних наша стратегия начинает с мая 2018 года работать не хуже обычной стратегии, а в конце временного промежутка даже лучше (рис. 3.2.б).

Теперь попробуем учесть затраты на транзакции *Trading Costs*. Будем считать, что средний спред  $spread = 1.5 \text{ pip}$ , тогда  $half \ spread = \frac{1.5}{2} \text{ pip}$ . Найдём среднее отношение цены

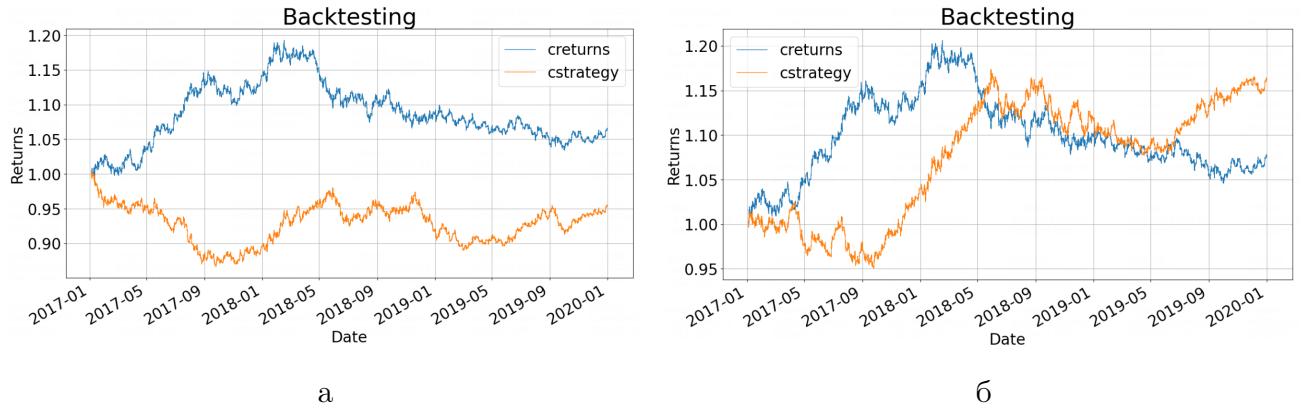


Рис. 3.2: *Backtesting* стратегии *SMA Crossover*

одной транзакции к средней цене  $proportion = \frac{half\ spread}{price_{mean}}$ . Затем найдём сколько всего трейдов было сделано (около 1%), посмотрим на частоту принятия решений за месяц (рис. 3.3.а) и посчитаем нашу чистую прибыль (доходность) (рис. 3.3.б), то мы увидим, что стратегия всё-таки работает чуть хуже, чем обычная стратегия из-за комиссии за транзакции, не смотря на не высокочастотный трейдинг

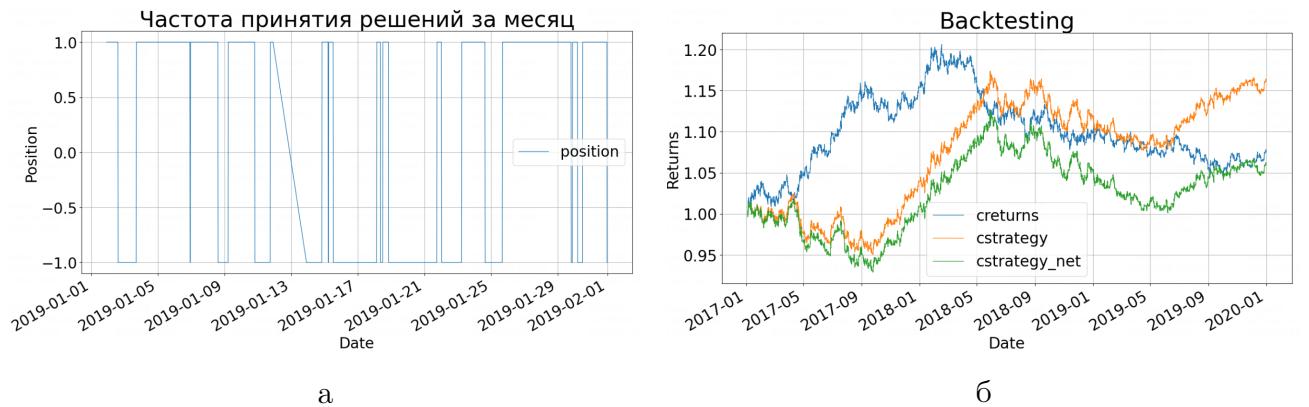


Рис. 3.3: *SMA Crossover* с учётом стоимости транзакций и частота принятия решений

Перечислим некоторые плюсы этой стратегии:

- Устранение шума: Использование скользящих средних позволяет сгладить краткосрочные колебания цен, что может помочь выявить более значимые тренды и уменьшить влияние шума на принятие решений.
- Гибкость: Стратегия может быть применена к различным временным рамкам и рынкам, что позволяет трейдерам адаптировать ее к своим индивидуальным потребностям и стилю торговли.

**Определение 3.2.7.** Боковое движение рынка - означает, что цены финансового инструмента колеблются в узком диапазоне без явно выраженного направления вверх или вниз.

Перечислим некоторые минусы этой стратегии:

- Ложные сигналы: Стратегия может порождать ложные сигналы пересечения, особенно в периодах бокового движения рынка. Это может приводить к ненужным сделкам и потере капитала.
- Задержка: Поскольку скользящие средние основаны на прошлых данных, они имеют некоторую задержку по отношению к текущим ценам. Это может означать, что трейдеры входят в сделки с некоторым опозданием, пропуская часть движения цены.
- Подверженность выбору параметров: Выбор периодов для скользящих средних может влиять на эффективность стратегии.

### 3.2.4 Стратегия: *Simple Contrarian/Momentum*

**Определение 3.2.8.** *Simple Momentum Strategy* - это торговая стратегия, основанная на принципе, что активы, которые двигаются в одном направлении, будут продолжать двигаться в этом направлении в ближайшем будущем.

Эта стратегия основана на предположении, что активы, которые показывают сильный рост или снижение цен, могут продолжить свое движение на определенный период времени.

**Определение 3.2.9.** *Simple Contrarian Strategy* - это торговая стратегия, которая основывается на принципе противоположного направления движения цен и противоречит текущему тренду на рынке.

Принципом стратегии *Simple Contrarian* является идея, что рынок периодически переоценивается, и когда цены поднимаются или падают слишком быстро и достигают экстремальных уровней, они могут отклониться и начать движение в противоположном направлении. Для *backtesting Simple Contrarian* стратегии воспользуемся теми же данными, что и в 3.2.3. Рассмотрим разные окна, на которых мы будем смотреть направление. Например, если окно 5, то мы посмотрим на 5 предыдущих свечей, посчитаем среднюю доходность и в зависимости от знака доходности будем двигаться в противоположном направлении. Как мы видим на (рис. 3.4.а) лучше всего, когда мы действуем против двух самых недавних трендов (смотрим на окно размера 2). Также, если посмотреть на ту же самую картинку, но действовать по тренду, то все наши стратегии работают хуже обычной (рис. 3.4.б).

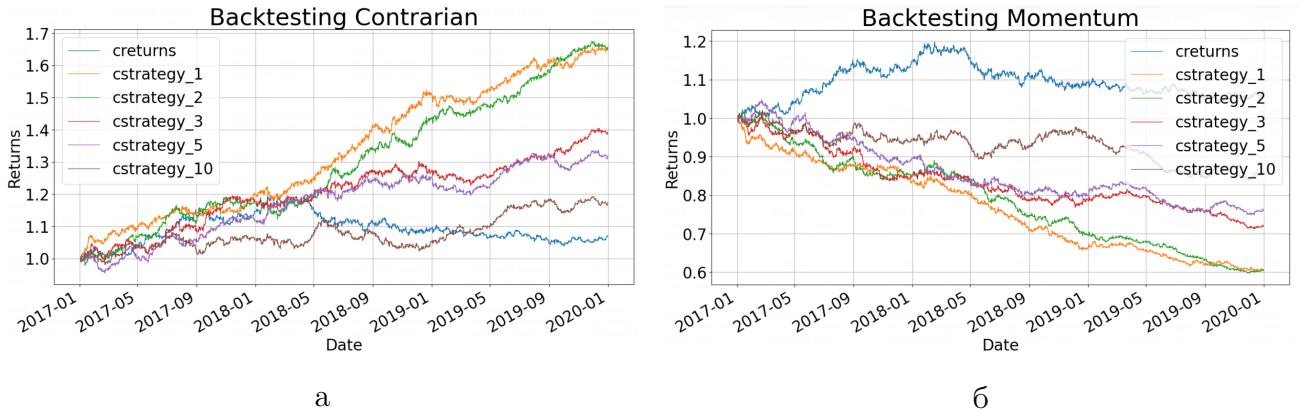


Рис. 3.4: *Backtesting Simple Contrarian* и *Simple Momentum*

Более того, т.к мы смотрим только на недавние, то мы очень часто меняем наше решение (рис. 3.5.а), получается мы занимаемся высокочастотным трейдингом. Если учесть затраты на транзакции, то мы получим печальную картину (рис. 3.5.б) однако мы выигрываем значительно без учёта *trading costs*, но если посмотреть например сентябрь 2019-го года как часто мы меняли позиции (покупали и продавали) (рис. 3.4.б), то мы видим, что мы занимаемся высокочастотным трейдингом и поэтому нам стоит оценить сколько мы потеряем из-за комиссии брокера.

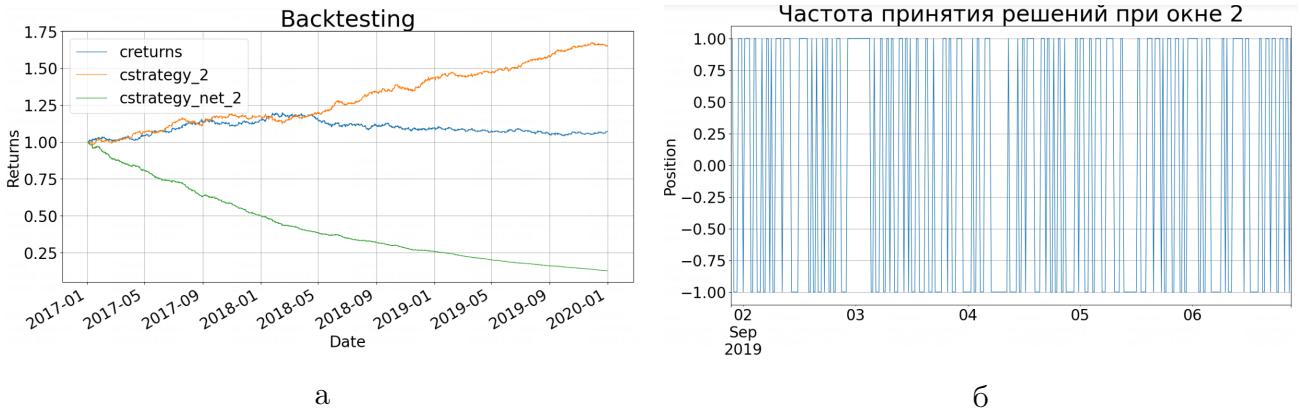


Рис. 3.5: Чистая прибыль и частота стратегии *Simple Contrarian*

Более того важно отметить, что эта стратегия требует внимательного анализа и опыта. Это связано с высоким уровнем риска, поскольку цены могут продолжать двигаться в основном направлении тренда, игнорируя антитрендовые сигналы.

### 3.2.5 Стратегия: *Mean – Reversion*

**Определение 3.2.10.** *Mean – Reversion* - это торговая стратегия, основанная на предположении, что цены активов имеют тенденцию колебаться вокруг своего среднего значения и временно отклоняться от этого значения.

Принцип стратегии заключается в том, чтобы входить в позиции, когда цена актива отклоняется от своего среднего значения в направлении, обратном отклонению. Например, если цена актива сильно возрастает и отклоняется от своего среднего значения, стратегия может предполагать продажу актива с ожиданием, что цена вернется к своему среднему значению. В основном для принятия решения используется индикатор *bollinger bands*:

**Определение 3.2.11.** Индикатор *Bollinger* (*Bollinger Bands*) - это технический аналитический индикатор, который используется для измерения волатильности цены актива и определения границы, в пределах которой цена вероятно будет колебаться.

Индикатор *Bollinger* состоит из трех компонентов:

- Средняя полоса (*SMA*): Обычное *SMA* за какой-то период.
- Верхняя полоса (*Upper Band*): Это линия, которая находится на фиксированное количество стандартных отклонений выше средней полосы. Обычно используется значение 2 стандартных отклонения, но также может быть настроено по желанию трейдера.
- Нижняя полоса (*Lower Band*): Это линия, которая находится на фиксированное количество стандартных отклонений ниже средней полосы. Опять же, обычно используется значение 2 стандартных отклонения.

Идея использования индикатора *Bollinger* заключается в следующем:

1. Когда цена актива приближается или касается верхней полосы, это может указывать на перекупленность актива, и цена может начать снижаться. Тут необходимо продавать актив.
2. Когда цена актива приближается или касается нижней полосы, это может указывать на перепроданность актива, и цена может начать расти. Тут необходимо покупать актив. Для визуализации стратегии воспользуемся теми же данными, что и в 3.2.3. Возьмём (рис. 3.6.а). проставим соответствующие позиции, где 1 - покупка, -1 - продажа и 0 - пересечение со средним. Все позиции, которые пустые (не пересекаются со средним и лежат между верхней и нижней границ) заполняем как *forward filling*, а те, для кого не было предыдущих значений заполняем нулём (рис. 3.6.б)

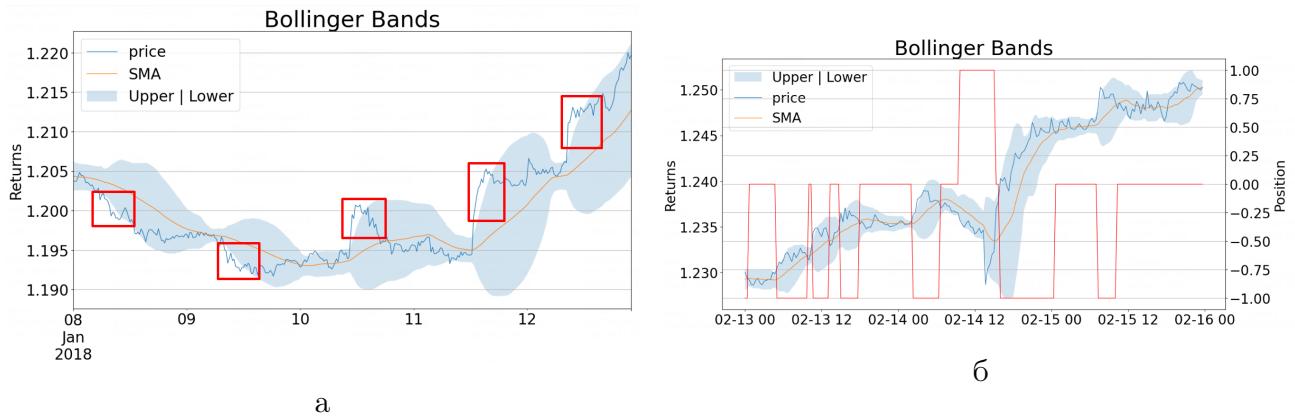


Рис. 3.6: Визуализация стратегии *Mean – Reversion*

Если мы сделаем *backtesting*, то обнаружим, что наша стратегия работает хорошо (рис. 3.7.а). Однако даже учитывая маленькое количество транзакций (около 4%) в чистой прибыли мы проигрываем (рис. 3.7.б). Тут я взял  $sma = 18$  подобрав аналогично 3.2.3 через фреймворк *optuna* используя ту же метрику. Возможно, для лучшего качества работы данной стратегии нужно уменьшить частоту данных, для уменьшения волатильности.

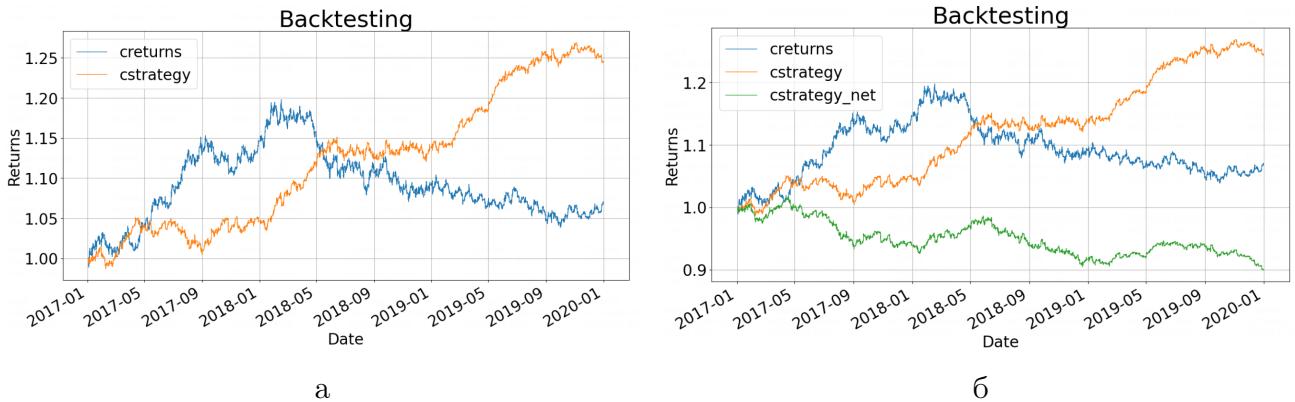


Рис. 3.7: *Backtesting* стратегии *Mean – Reversion*

### 3.3 Gaussian Mixture Models (GMM) и EM-алгоритм

#### 3.3.1 Gaussian Mixture Models (GMM)

Рассмотрим алгоритм обучения без учителя - кластеризации, для оценки распределения доходности (*return*). Мы заметили, что распределение доходности (*return*) имеет более тяжёлые хвосты, чем нормальное распределение и что распределение Стьюдента

достаточно неплохо описывает его. Однако, хочется ещё лучше его приблизить, поскольку это один из важнейших показателей в торговле. Обратимся к методу кластеризации *Gaussian Mixture Models (GMM)*:

**Определение 3.3.1.** *GMM (Gaussian Mixture Models)* или модель смеси гауссиан - это вероятностная модель, которая используется для описания сложных распределений данных. Она представляет собой комбинацию нескольких нормальных (гауссовых) распределений, каждое из которых моделирует подмножество данных (рис. 3.8). Каждое гауссово значение в смеси состоит из следующих параметров:

- Среднее значение  $\mu$ , определяющее его центр
- Ковариация  $\Sigma$ , определяющая ее ширину. Это было бы эквивалентно размерам эллипса в многомерном сценарии.
- Вероятность смешения  $\pi$ , которая представляет собой вероятность того, что конкретный образец данных принадлежит к определенной компоненте смеси гауссиан.

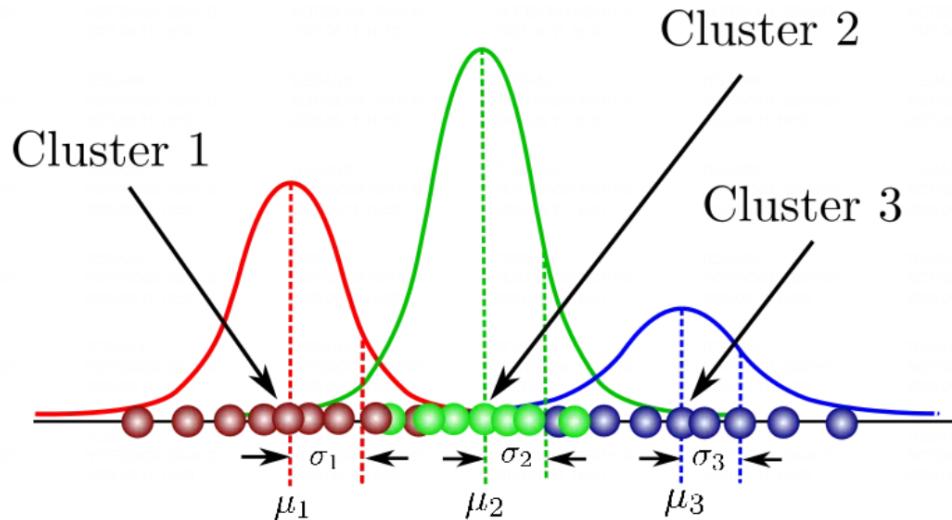


Рис. 3.8: Иллюстрация *GMM* с тремя кластерами

*GMM* может быть использована для различных задач, таких как кластеризация данных, моделирование плотности вероятности, сжатие данных и генерация новых данных. Как и для всех вероятностей, для вероятностей смешения выполняется следующее равенство:

$$\sum_{i=1}^K \pi_i = 1$$

Мы можем записать плотность распределения для каждого кластера:

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^k \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

Затем мы можем применить метод максимального правдоподобия и найти оптимальные параметры. Однако, поскольку мы имеем дело не с одним, а со многими гауссианами, все немного усложнится, когда придет время найти параметры для всей смеси.

Запишем плотность распределения всей смеси (для наблюдения  $\mathbf{x}$ ):

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \cdot N(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Мы хотим максимизировать функцию правдоподобия ( $N$  - количество наблюдений):

$$P(\mathbf{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^N \left[ \sum_{k=1}^K \pi_k \cdot N(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right]$$

### 3.3.2 EM - алгоритм

Определим следующие переменные:

$$z_{nk} = \begin{cases} 1, & \text{если } x_n \text{ в кластере } k \\ 0, & \text{иначе} \end{cases}$$

$$\gamma(z_{nk}) = P(z_{nk} = 1 | \mathbf{x}_n)$$

$$\gamma(z_{nk}) = \frac{P(z_{nk} = 1) \cdot P(\mathbf{x}_n | z_{nk} = 1)}{\sum_{j=1}^K [P(z_{nj} = 1) \cdot P(x_n | z_{nj} = 1)]} = \frac{\pi_k \cdot N(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K [\pi_j \cdot N(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)]} \quad (3.1)$$

$$N_k = \sum_{n=1}^N \gamma_{nk}$$

После максимизации мы получим следующие выражения:

$$\begin{cases} \boldsymbol{\mu}_k = \frac{1}{N_k} \cdot \sum_{n=1}^N [\gamma_{nk} \cdot \mathbf{x}_n] \\ \boldsymbol{\Sigma}_k = \frac{1}{N_k} \cdot \sum_{n=1}^N [\gamma_{nk} \cdot (\mathbf{x}_n - \boldsymbol{\mu}_k) \cdot (\mathbf{x}_n - \boldsymbol{\mu}_k)^T] \\ \pi_k = \frac{N_k}{N} \end{cases} \quad (3.2)$$

Однако мы видим, что наши переменные  $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)$  зависят от  $\gamma$ , а переменная  $\gamma$  зависит от наших переменных  $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)$ . Тут на помощь приходит EM-алгоритм:

**Определение 3.3.2.** *EM*-алгоритм (*Expectation – Maximization*) - это итеративный статистический метод, используемый для оценки параметров статистической модели в присутствии скрытых (латентных) переменных.

**Определение 3.3.3.** Функция неполного правдоподобия - функция правдоподобия, учитывающая скрытые (латентные) переменные

Основная идея *EM*-алгоритма состоит в чередовании двух шагов: *E*-шаг (*Expectation*) и *M*-шаг (*Maximization*). На *E*-шаге вычисляются ожидания скрытых переменных, используя текущие значения параметров модели. На *M*-шаге выполняется максимизация неполного правдоподобия по параметрам модели, основываясь на полученных ожиданиях скрытых переменных. После *M*-шага обновляются значения параметров. Затем процесс *E*-шага и *M*-шага повторяется до достижения сходимости. *EM*-алгоритм гарантирует монотонное увеличение логарифма неполного правдоподобия на каждой итерации, что приводит к приближенной оценке параметров модели. Однако, сходимость к глобальному оптимуму не всегда гарантируется, и результаты могут зависеть от начальных приближений и выбора модели. Критерии остановки могут быть разные: максимальное количество итераций, сходимость параметров (с заданной  $\epsilon$ ), сходимость неполного правдоподобия (с заданной  $\epsilon$ ). Запишем псевдокод:

```

1 Инициализируем  $\mu_k$ ,  $\Sigma_k$ ,  $\pi_k$ 
2 Высчитываем  $\gamma_{nk}$  - E-шаг (3.1)
3 Пересчитываем  $\mu_k$ ,  $\Sigma_k$ ,  $\pi_k$  - M-шаг (3.2)
4 if условие остановки then
5   | return  $\mu_k$ ,  $\Sigma_k$ ,  $\pi_k$ 
6 else
7   | возвращаемся к шагу 2 (E-шаг)
8 end

```

Algorithm 1: *EM*-алгоритм

### 3.3.3 Поиск распределения доходности (*return*)

*GMM* поможет нам в предсказании распределения доходности, т.к оно в отличии от обычного нормального распределения имеет более тяжёлые хвосты (т.к является комбинацией нормальных распределений с разными параметрами  $\mu$  и  $\Sigma$ ) и поэтому может очень хорошо приблизить наше исходное распределение. Нам дано некоторое распределение со средним  $\mu$  и с дисперсией  $\sigma^2$ . Возьмём два кластера с одинаковыми мат ожиданиями. Тогда плот-

нность распределения для наблюдения  $\mathbf{x}$  будет выглядеть как:

$$p(\mathbf{x}) = \pi \cdot N(x | \mu, \sigma_1^2) + (1 - \pi) \cdot N(x | \mu, \sigma_2^2)$$

Теперь мы можем подобрать  $\sigma_1^2$  и  $\sigma_2^2$  таким образом, чтобы получилась итоговая  $\sigma^2$ , но итоговая комбинация двух кластеров будет иметь более тяжёлые хвосты, чем просто  $N(\mu, \sigma^2)$ . Наглядная демонстрация на (рис. 3.9). Тут я взял два распределения с  $\mu = 0$  и  $\sigma_1^2 = 0.02$  и  $\sigma_2^2 = 0.004$ . Как мы видим, получившаяся смесь очень похожа на наше распределение доходности.

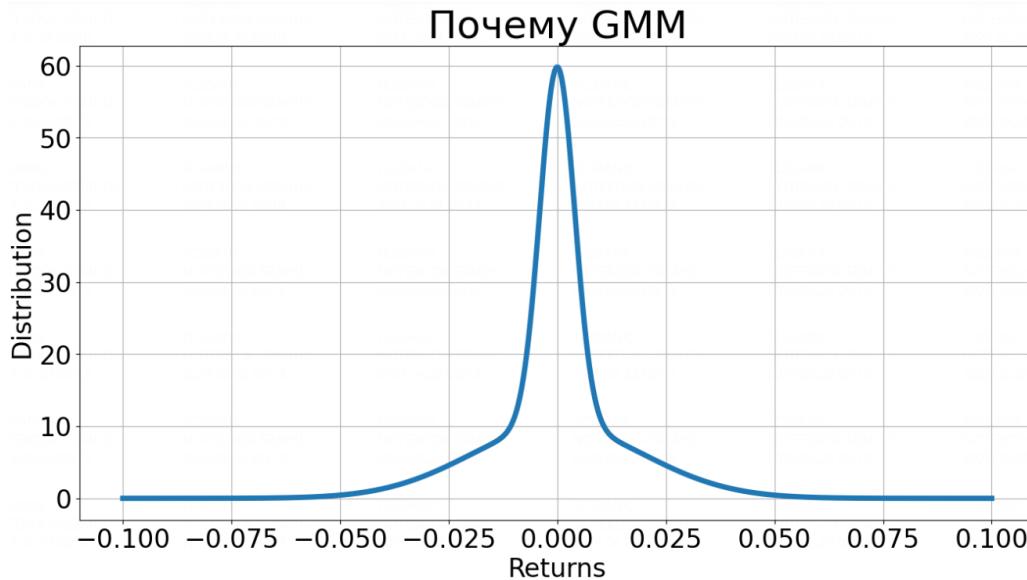


Рис. 3.9: Смесь из двух распределений с одинаковым  $\mu$ , но разными  $\sigma$

Возьмём данные акций компании "Google" с 2014-го года по 2023-ий год с частотой в один день и построим с помощью *GMM* приближение нашего распределения. Получившееся распределение сравним с исходным распределением (рис. 3.10.а) и с распределением Стьюдента (рис. 3.10.б).

Также посчитаем коэффициен эксцесса для исходного распределения, полученного и  $t$ -распределения:

$$t\text{-distribution kurtosis} = 5.481$$

$$GMM \text{ kurtosis} = 5.500$$

$$Real \text{ kurtosis} = 6.422$$

Получается, что наша интуиция нас не подвела и вправду существует лучшее (хоть и на чутко-чуть) распределение, которое приближает наше исходное распределение доходности

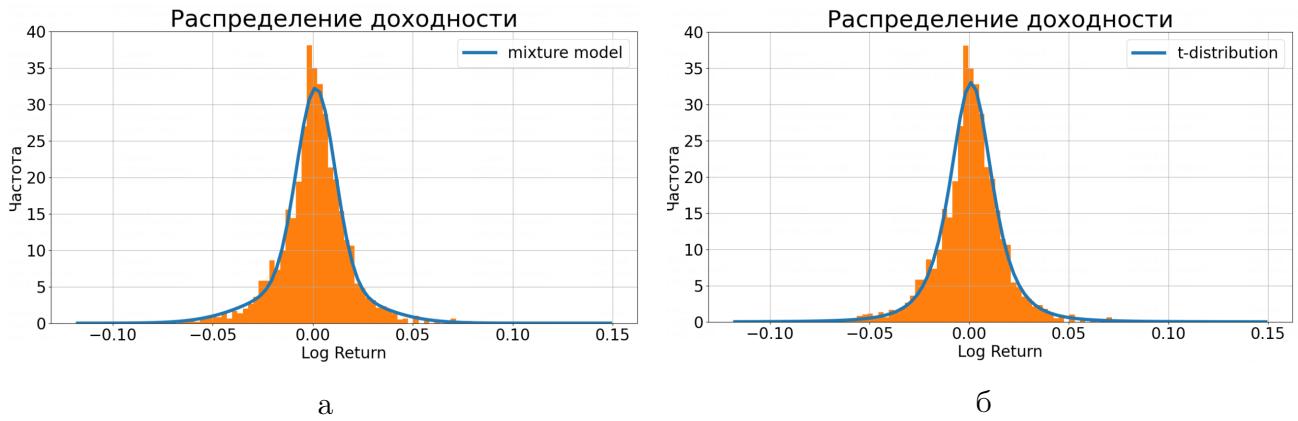


Рис. 3.10: Сравнение  $GMM$  и  $t$ -распределения(Стьюдента)

(*return*), как минимум для данного случая. Однако из-за достаточно маленького прироста, проще использовать распределение Стьюдента и не заморачиваться с  $GMM$ .

## 3.4 Линейная регрессия

Разберём один из основных методов машинного обучения в классе задач регрессии и обучения с учителем - линейную регрессию

### 3.4.1 Постановка задачи

Предположим, что у нас есть матрица независимых переменных  $\mathbf{X}$  размерности  $N \times (m + 1)$ , где каждая строка представляет собой вектор независимых переменных  $\mathbf{x}_i$ , а первый столбец состоит из единиц (для учета свободного члена в модели). Также у нас есть вектор зависимой переменной  $\mathbf{y}$  размерности  $N \times 1$ , состоящий из фактических значений для каждого наблюдения. Мы хотим найти линейную модель, которая связывает независимые переменные с зависимой переменной.

Модель линейной регрессии можно представить в матричной форме следующим образом:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \varepsilon,$$

где  $\mathbf{b}$  - вектор параметров размерности  $(m + 1) \times 1$ , содержащий параметры модели, а  $\varepsilon$  - вектор ошибок размерности  $N \times 1$ .

Цель состоит в нахождении оптимальных значений параметров  $\mathbf{b}$ , минимизирующих сумму квадратов остатков:

$$\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2,$$

Аналитическое решение для оптимальных значений параметров  $\mathbf{b}$  может быть выражено следующим образом:

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y},$$

Для применения линейной регрессии необходимо использовать следующие предположения, описанные теоремой Гаусса-Маркова:

### 3.4.2 Необходимые предположения

**Теорема 3.4.1** (Гаусс-Марков). При соблюдении следующих условий:

1. Линейность: Модель линейной регрессии должна быть линейной по параметрам

$$y = \mathbf{X}\mathbf{b} + \varepsilon$$

2. Случайность выборки и полнота ранга: Наблюдения  $(x_i, y_i)$  - независимы  $\forall i$  и ни один из признаков не является константой или линейной комбинацией других признаков в выборке

$$(rank(\mathbf{X}) = m + 1)$$

3. Случайность ошибок: Ошибки модели должны быть случайны

$$E(\varepsilon | X) = 0$$

4. Гомоскедастичность: Ошибки должны иметь постоянную дисперсию, то есть быть одинаково изменчивыми по всем значениям независимых переменных

$$D[\varepsilon | X] = \sigma^2$$

5. Некоррелированность ошибок: Ошибки модели должны быть нескоррелированы друг с другом

$$cov(\varepsilon_i, \varepsilon_j) = 0, \quad i \neq j$$

оценки параметров, полученные методом наименьших квадратов (МНК), являются наилучшими линейными несмещёнными оценками. Они имеют наименьшую дисперсию среди всех линейных несмещённых оценок параметров.

При добавлении условия нормальности ошибок ( $\varepsilon \sim N(0, \sigma^2)$ ), МНК оценка совпадает с оценкой методом максимального правдоподобия (ММП)

**Определение 3.4.1.** Метод максимального правдоподобия (*MLE*) - это статистический метод, при котором значения параметров модели выбираются таким образом, чтобы вероятность (правдоподобие) получения  $y$ , при наблюдаемых данных  $X$  была максимальна.

### 3.4.3 Метрики качества

Рассмотрим основные метрики качества (функции ошибок) для регрессии:

- Среднеквадратичная ошибка (*Mean Squared Error (MSE)*)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

где  $y_i$  - фактическое значение,  $\hat{y}_i$  - предсказанное значение,  $n$  - общее количество наблюдений

- Средняя абсолютная ошибка (*Mean Absolute Error (MAE)*)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

где  $y_i$  - фактическое значение,  $\hat{y}_i$  - предсказанное значение,  $n$  - общее количество наблюдений

Основная разница между *MAE* и *MSE* в их отношении к выбросам. *MAE* является более устойчивой к выбросам, в отличии от *MSE*, но при этом *MAE* является не всюду дифференцируемой функцией. Поэтому выбор между *MAE* и *MSE* зависит от особенностей задачи, типа данных и требований к модели.

И введём ещё одну метрику качества:

- Коэффициент детерминации (*Coefficient of Determination ( $R^2$ )*) - показывает, какую часть дисперсии зависимой переменной объясняет модель. Значение ( $R^2$ ) находится в диапазоне от 0 до 1, где 0 означает, что модель не объясняет вариабельность данных, а 1 - что модель идеально объясняет данные.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

где  $y_i$  - фактическое значение,  $\hat{y}_i$  - предсказанное значение,  $n$  - общее количество наблюдений,  $\bar{y}$  - среднее фактическое значение

### 3.4.4 Дополнение

#### Разложение ошибки на смещение и разброс

Пусть истинное значение целевой переменной представлено в следующем виде:

$$\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\varepsilon}, \quad (\boldsymbol{\varepsilon} \sim \mathbf{N}(0, \sigma^2))$$

Мы пытаемся приблизить функцию  $f(\mathbf{x})$  линейной функцией от параметров  $f(\hat{\mathbf{x}})$ . Тогда ошибку запишем в виде:

$$Error(\mathbf{x}) = E[(y - f(\hat{\mathbf{x}}))^2]$$

Которую можно записать в виде:

$$Error(\mathbf{x}) = (f(\mathbf{x}) - E[\hat{f}(\mathbf{x})]^2) + D[\hat{f}(\mathbf{x})] + \sigma^2 = \text{Bias}(\hat{f}(\mathbf{x}))^2 + D[\hat{f}(\mathbf{x})] + \sigma^2$$

- Квадрат смещения ( $\text{Bias}(\hat{f}(\mathbf{x}))$ ) - средняя ошибка по всевозможным наборам данных
- Дисперсия ( $D[\hat{f}(\mathbf{x})]$ ) - вариативность ошибки, то, на сколько ошибка будет отличаться, если обучать модель на разных наборах данных
- Неустранимая ошибка ( $\sigma^2$ )

На практике часто приходится балансировать между смещенными ( $\text{Bias}(\hat{f}(\mathbf{x})) \neq 0$ ) и нестабильными оценками (высокая  $D[\hat{f}(\mathbf{x})]$ ) (рис. 3.11)

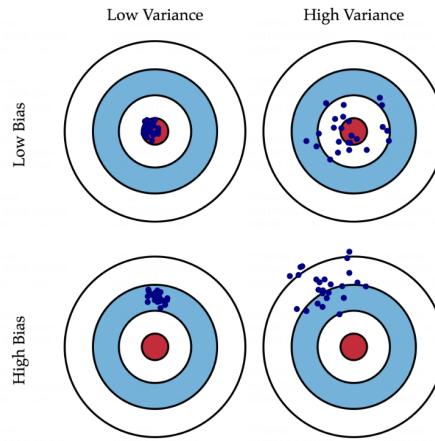


Рис. 3.11: Визуализация ошибки

## Регуляризация

**Определение 3.4.2.** Мультиколлинеарность - приближённая линейная зависимость

$$x_i \approx x_j + \dots + x_k$$

Очень часто в данных присутствует мультиколлинеарность и поэтому веса становятся огромными, поскольку детерминант матрицы X стремится к нулю  $\Rightarrow (X^T X)^{-1}$  стремится к бесконечности. Одним из методов борьбы с этим является регуляризация (основная идея - ограничение весов по норме) (рис. 3.12)

**Определение 3.4.3.** Регуляризация в линейной регрессии - это метод, используемый для контроля переобучения модели путем добавления штрафа к функции потерь или к самой модели.

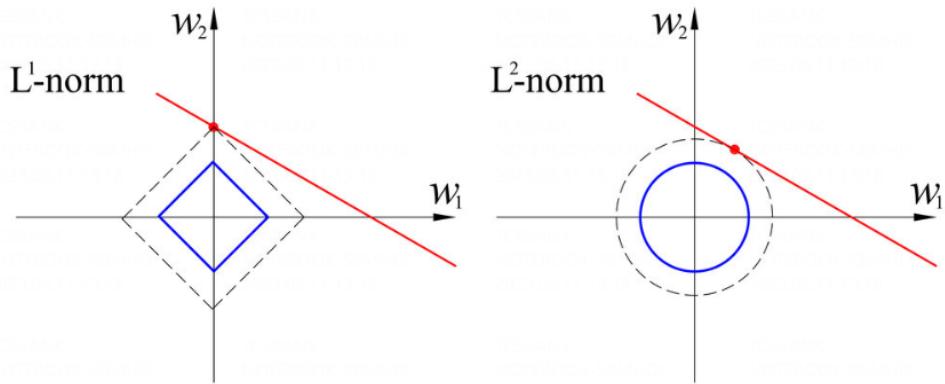


Рис. 3.12:  $L_1/L_2$  регуляризации

Регуляризация помогает уменьшить влияние шума в данных и улучшить обобщающую способность модели. В линейной регрессии существуют два распространенных метода регуляризации:  $L_1$ -регуляризация (Лассо) и  $L_2$ -регуляризация (Гребневая регрессия).

**Определение 3.4.4.**  $L_1$ -регуляризация (Лассо):

$$L(\mathbf{b}) = \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{b}^T \mathbf{x}_i)^2 + \lambda \sum_{j=1}^m |b_j|,$$

$L_1$ -регуляризация способствует разреженности весов, что может быть полезно для отбора признаков, исключая несущественные признаки из модели.

**Определение 3.4.5.**  $L_2$ -регуляризация (Гребневая регрессия):

$$L(\mathbf{b}) = \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{b}^T \mathbf{x}_i)^2 + \lambda \sum_{j=1}^m b_j^2.$$

$L_2$ -регуляризация способствует сжатию значений весов, уменьшая их абсолютные значения, и тем самым уменьшая влияние отдельных признаков (борется с мультиколлинеарностью).

Коэффициент регуляризации  $\lambda$  контролирует величину штрафа. Большее значение  $\lambda$  приводит к более сжатым (ближе к нулю) весам и более сильной регуляризации, тогда как меньшее значение  $\lambda$  уменьшает штраф и позволяет весам принимать большие значения.

### 3.4.5 Показатели альфа и бетта

Поговорим о важных показателях в финансах, которые используются для измерения риска и доходности инвестиций относительно рыночного индекса (обычно используется бирже-

вый индекс, такой как  $S&P 500$ ) - альфа и бетта. Оба показателя - исторические  $\Rightarrow$  они зависят от выбранного временного отрезка и не гарантируют результат в будущем. Рассмотрим их подробнее:

**Определение 3.4.6.** Альфа ( $\alpha$ ) - является мерой "избыточной" доходности инвестиции или портфеля относительно доходности рынка.

Альфа показывает, насколько инвестиция превышает (положительная альфа) или отстает (отрицательная альфа) от ожидаемой доходности на основе риска.

**Определение 3.4.7.** Бета ( $\beta$ ) - измеряет чувствительность инвестиции или портфеля к изменениям в рыночном индексе.

Бета показывает, насколько инвестиция движется в согласованности с рыночными движениями.

Теперь проинтерпретируем их с помощью линейной регрессии, т.к на самом деле эти показатели относятся к корреляции. Посмотрим на (рис. 3.13). По оси  $x$  отложена доходность рынка ( $S&P 500$ ), который мы назвали просто  $SPY$ , а по оси  $y$  отложена доходность любой другой компании (в нашем примере *Apple*). В таком случае используя линейную регрессию можно получить, что

$$Return = \alpha + \beta \cdot Return_{market}$$

Получается, что бета ( $\beta$ ) интерпретируется, как наклон графика (вес), а альфа ( $\alpha$ ), как свободный член. Получается, что бета показывает, как будет вести себя доходность другой компании, по отношению к рынку. Например, пусть  $\beta = 2$ , то если доходность рынка увеличится на 2%, то доходность другой компании увеличится на 4% и аналогично, если доходность рынка будет падать, то доходность другой компании упадёт в два раза. Тут мы учитываем, что показатель  $\alpha = 0$ . Показатель альфа же показывает, насколько доходность другой компании лучше, чем доходность рынка, поскольку если в примере выше  $\alpha > 0$  и равен например  $\alpha = 0.5$ , то мы получим, что при увеличении доходности рынка на 1%, другая компания при  $\beta = 2$  увеличит доход не на 2%, а на 2.5% из-за избыточной доходности.

Вот как интерпретируют значения беты в общем случае:

- $\beta < 1 \Rightarrow$  корреляция актива и рынка обратная, при этом актив более волатилен.
- $-1 < \beta < 0 \Rightarrow$  корреляция по-прежнему обратная, но актив ведет себя стабильнее рынка.
- $0 < \beta < 1 \Rightarrow$  актив движется однородно с рынком, риск меньше рыночного.

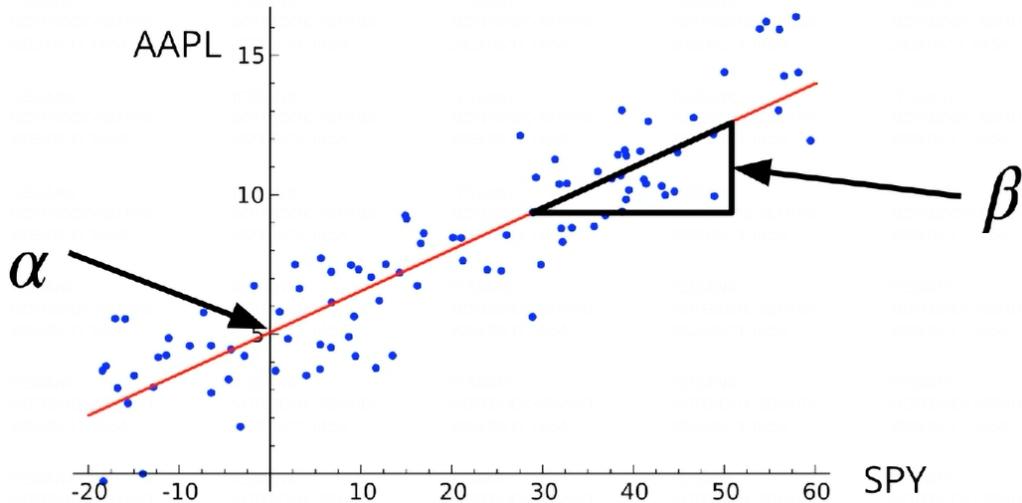


Рис. 3.13: Интерпретация альфа и бета

- $\beta > 1 \Rightarrow$  актив коррелирует с индексом и более волатилен, то есть он очень рисковый.  
При этом отрицательная бета на практике встречается крайне редко.

## 3.5 Логистическая регрессия

Логистическая регрессия является одним из основных методов машинного обучения, применяемых для решения задач классификации.

### 3.5.1 Постановка задачи

Логистическая регрессия очень похожа на линейную регрессию (3.4), однако работает как линейный классификатор, основная идея которого заключается в том, что признаковое пространство может быть разделено гиперплоскостью на два полупространства, в каждом из которых прогнозируется одно из двух значений целевого класса. Если это можно сделать без ошибок, то обучающая выборка называется линейно разделимой. Один из самых простых линейных классификаторов получается на основе регрессии вот таким образом:

$$a(\vec{x}) = \text{sign}(\vec{w}^T \vec{x})$$

, где  $\vec{x}$  - вектор признаков примера (вместе с единицей),  $\vec{w}$  - веса в линейной модели (вместе со смещением  $w_0$ ),  $a(\vec{x})$  - ответ классификатора на примере  $\vec{x}$

В отличии от линейной регрессии, логистическая регрессия позволяет моделировать вероятность принадлежности объекта к определенному классу на основе входных признаков.

При этом вероятность отнесения примера к классу "+"(при условии, что мы знаем его признаки и веса модели) равна сигмоид-преобразованию линейной комбинации вектора весов модели и вектора признаков примера:

$$p_+(x_i) = P(y_i = 1 \mid \vec{x}_i, \vec{w}) = \frac{1}{1 + \exp^{-\vec{w}^T \vec{x}_i}} = \sigma(\vec{w}^T \vec{x}_i)$$

Тогда для класса "-"аналогична вероятность:

$$p_-(x_i) = P(y_i = -1 \mid \vec{x}_i, \vec{w}) = 1 - \sigma(\vec{w}^T \vec{x}_i) = \sigma(-\vec{w}^T \vec{x}_i)$$

Оба этих выражения можно ловко объединить в одно:

$$P(y = y_i \mid \vec{x}_i, \vec{w}) = \sigma(y_i \vec{w}^T \vec{x}_i)$$

**Определение 3.5.1.**  $M(\vec{x}_i) = y_i \vec{w}^T \vec{x}_i$  - называется отступом (*margin*) классификации на объекте  $\vec{x}_i$

Если отступ неотрицателен, модель не ошибается на объекте  $\vec{x}_i$ , если же отрицателен, то это значит, что класс для  $\vec{x}_i$  спрогнозирован неправильно. Замечу, что отступ определен для объектов именно обучающей выборки, для которых известны реальные метки целевого класса  $y_i$ .

Модель линейной регрессии решает задачу метода максимального правдоподобия, которая приводит к минимизации выражения:

$$\mathcal{L}_{log}(X, \vec{y}, \vec{w}) = \sum_{i=1}^{\ell} \log(1 + \exp^{-y_i \vec{w}^T \vec{x}_i})$$

Эта функция является логистической функцией потерь, просуммированная по всем объектам обучающей выборки. Посмотрим на новую функцию как на функцию от отступа:  $L(M) = \log(1 + \exp^{-M})$ . Нарисуем ее график, а также график 1/0 функций потерь (*zero – one loss*), которая просто штрафует модель на 1 за ошибку на каждом объекте (отступ отрицательный):  $L_{1/0}(M) = [M < 0]$  (рис. 3.14)

Картинка отражает общую идею, что в задаче классификации, не умея напрямую минимизировать число ошибок (по крайней мере, градиентными методами это не сделать - производная 1/0 функций потерь в нуле обращается в бесконечность), мы минимизируем некоторую ее верхнюю оценку. В данном случае это логистическая функция потерь, и справедливо

$$\begin{aligned} \mathcal{L}_{1/0}(X, \vec{y}, \vec{w}) &= \sum_{i=1}^{\ell} [M(\vec{x}_i) < 0] \\ &\leq \sum_{i=1}^{\ell} \log(1 + \exp^{-y_i \vec{w}^T \vec{x}_i}) \\ &= \mathcal{L}_{log}(X, \vec{y}, \vec{w}) \end{aligned}$$

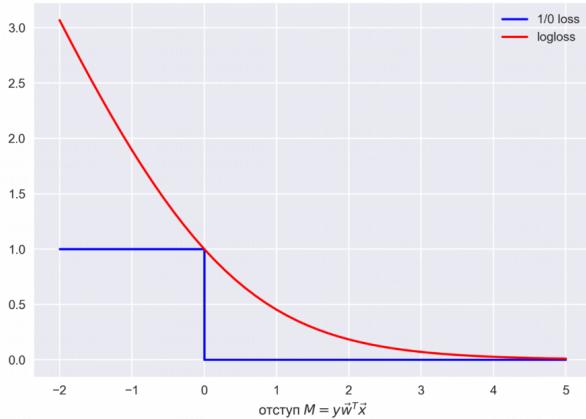


Рис. 3.14: Logloss и настоящая функция ошибки

где  $\mathcal{L}_{1/0}(X, \vec{y}, \vec{w})$  - попросту число ошибок логистической регрессии с весами  $\vec{w}$  на выборке  $(X, \vec{y})$ .

То есть уменьшая верхнюю оценку  $\mathcal{L}_{log}$  на число ошибок классификации, мы таким образом надеемся уменьшить и само число ошибок. Также замечу, что для логистической регрессии справедливы те же методы борьбы с переобучением, что и для линейной регрессии.

### 3.5.2 Метрики качества

Теперь кратко разберём различные метрики качества, которые применяются в задачах классификации, но сначала введём очень важное понятие - матрица ошибок, которая используется для оценки точности моделей в задачах классификации.

**Определение 3.5.2.** Матрица ошибок (*Confusion Matrix*) - таблица с 4 различными комбинациями прогнозируемых и фактических значений. Прогнозируемые значения описываются как положительные и отрицательные, а фактические - как истинные и ложные (рис. 3.15)

В матрице ошибок присутствуют следующие элементы:

*True Positive (TP)* - число объектов, которые действительно принадлежат положительному классу и были верно предсказаны как положительные.

*True Negative (TN)* - число объектов, которые действительно принадлежат отрицательному классу и были верно предсказаны как отрицательные.

*False Positive (FP)* - число объектов, которые на самом деле принадлежат отрицательному классу, но были неправильно предсказаны как положительные.

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Рис. 3.15: Визуализация матрицы ошибок (*Confusion Matrix*)

*False Negative (FN)* - число объектов, которые на самом деле принадлежат положительному классу, но были неправильно предсказаны как отрицательные.

Затем из этой таблицы вытекают следующие метрики оценки качества:

**Точность (Accuracy)** - доля правильных ответов алгоритма. Эта метрика бесполезна в задачах с неравными классами

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

**Полнота (Recall)** - показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

$$Recall = \frac{TP}{TP + FN}$$

**Точность (Precision)** - можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными

$$Precision = \frac{TP}{TP + FP}$$

*Recall* демонстрирует способность алгоритма обнаруживать данный класс вообще, а *precision* способность отличать этот класс от других классов.

**F-мера ( $F_\beta$  – score)** - среднее гармоническое *precision* и *recall* :

$$F_\beta - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$F_\beta$  – score - является способом объединить *precision* и *recall* в агрегированный критерий качества.

И теперь поговорим коротко про пороги, которые выставляются при конвертации вещественного ответа алгоритма (как правило, вероятности принадлежности к классу) в бинарную метку. Естественным и близким кажется порог, равный 0.5, но он не всегда оказывается оптимальным. Поэтому используют следующую метрику:

**Площадь под ROC кривой (ROC AUC)** - способ оценить модель в целом, не привязываясь к конкретному порогу. Данная кривая представляет из себя линию от  $(0, 0)$  до  $(1, 1)$  в координатах *True Positive Rate (TPR)* и *False Positive Rate (FPR)*:

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}$$

## 3.6 Полносвязная нейронная сеть (*DNN*)

### 3.6.1 *DNN* и работа нейрона

Теперь поговорим об очень мощном инструменте - нейронных сетях. Нейронные сети являются мощным инструментом в области искусственного интеллекта и машинного обучения. Они основаны на идеях, вдохновленных работой нервной системы живых организмов, и стремятся симулировать способность мозга к обработке информации. Нейронные сети состоят из сети взаимосвязанных искусственных нейронов, которые обмениваются данными и принимают решения на основе определенных правил. Одним из ключевых преимуществ нейронных сетей является их способность извлекать сложные зависимости и обнаруживать скрытые паттерны в больших объемах данных. Посмотрим, как выглядит самая простая нейронная сеть - она же полносвязная нейронная сеть (рис. 3.16.а).

**Определение 3.6.1.** Полносвязная нейронная сеть (*DNN*) - нейронная сеть, которая состоит из нескольких слоев нейронов, где каждый нейрон предыдущего слоя связан с каждым нейроном следующего слоя.

Она состоит из следующих компонентов: входного слоя, скрытых слоев и выходного слоя:

- **Входной слой** - входные данные подаются на входной слой нейронной сети.
- **Скрытые слои** - находятся между входным слоем и выходным слоем нейронной сети. Они выполняют вычисления и обработку данных.
- **Выходной слой** - получает данные из последнего скрытого слоя и генерирует окончательный выход или прогноз.

Теперь разберём работу нейрона (рис. 3.16.б). Нейрон в нейронной сети выполняет обработку входных данных (не только от входного слоя) и передает результаты следующему слою. Процесс работы нейрона можно разделить на следующие шаги:

- **Взвешенная сумма** - нейрон умножает каждое входное значение на соответствующий вес связи и суммирует результаты. Это выражается как:  $\tilde{x} = \sum_{j=1}^n x_j \cdot \omega_j$

**Определение 3.6.2.** Затухание градиента - это проблема, возникающая в глубоких нейронных сетях при обратном распространении ошибки (будет дальше), когда градиенты, передаваемые от выходного слоя к входному, с течением времени становятся все меньше и меньше. В результате этого обновления весов остаются незначительными, и обучение модели замедляется или прекращается.

- **Функция активации** - на полученную взвешенную сумму нейрон применяет функцию активации. Она добавляет нелинейность в нейронную сеть, что позволяет ей моделировать более сложные зависимости. Некоторые из популярных функций активации:

**Сигмоидная функция (Sigmoid)** - широко применяется в задачах бинарной классификации, где требуется получить вероятности в диапазоне  $(0, 1)$ . Однако сигмоидная функция имеет проблему затухания градиента при глубоких нейронных сетях.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

**Гиперболический тангенс (Tanh)** - преобразует значения в интервал  $(-1, 1)$ , что позволяет моделировать как положительные, так и отрицательные зависимости. Гиперболический тангенс также страдает от проблемы затухания градиента, особенно при больших значениях входа.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

**Rectified Linear Unit (ReLU)** - очень проста и эффективна в вычислительном плане. *ReLU* помогает сети избежать проблемы затухания градиента при положительных значениях.

$$ReLU(x) = \max(0, x)$$

**Leaky ReLU** - такая же, как и *ReLU*, но помогает сети избежать проблемы затухания градиента для всех значений.

$$Leaky ReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{otherwise} \end{cases}$$

где  $\alpha$  - это маленькое положительное число (обычно около 0.01), называемое параметром утечки.

**Softmax:** Для каждого элемента  $x_i$  в векторе  $x$  размерности  $N$ , *softmax* вычисляет вероятность  $p_i$  по формуле:

$$p_i = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}$$

Выбор функции активации зависит от конкретной задачи и архитектуры сети. Обычно *ReLU* является хорошим выбором для скрытых слоев нейронных сетей, поскольку он способствует более быстрой сходимости обучения и может предотвратить затухание градиента. Сигмоидная функция и гиперболический тангенс обычно используются в последнем слое для задач классификации или регрессии, где требуется получить вероятности или ограниченные значения.

- **Выход** - результат функции активации становится выходным значением нейрона. Это значение передается следующему слою нейронной сети и используется в дальнейших вычислениях.

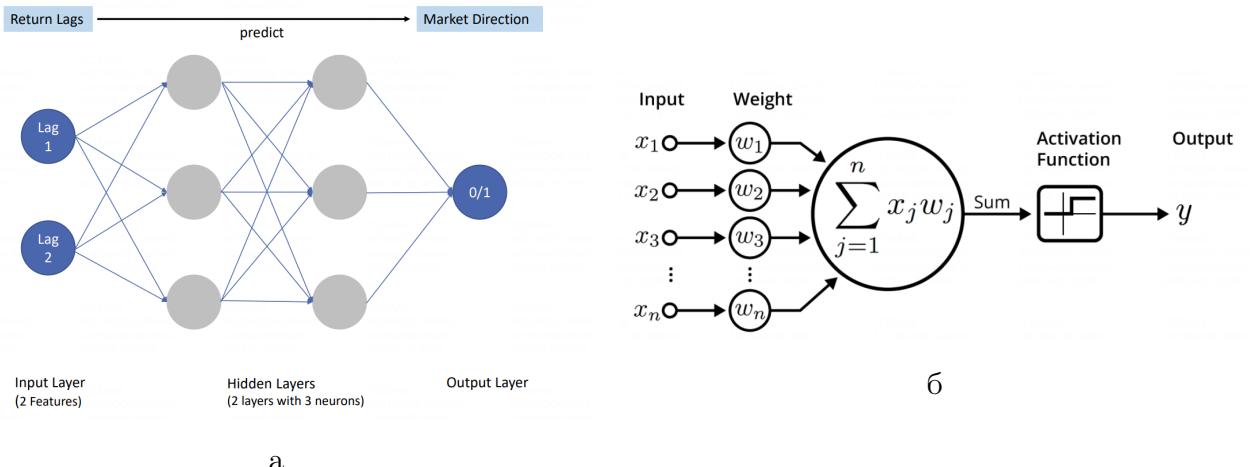


Рис. 3.16: Визуализация *DNN* и работы нейрона

### 3.6.2 Обучение нейронной сети (*DNN*)

Обучение нейронной сети заключается в настройке весов связей между нейронами для минимизации ошибки на тренировочных данных. Вот некоторые распространенные функции ошибки для различных типов задач:

**Среднеквадратичная ошибка (*MSE*):**

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

где  $N$  - количество примеров в наборе данных,  $y_i$  - фактическое значение,  $\hat{y}_i$  - предсказанное значение. Используется в задачах регрессии.

### Категориальная перекрестная энтропия (*Categorical Cross – Entropy*)

$$CCE = - \sum_{i=1}^N y_i \log(\hat{y}_i)$$

где  $N$  - количество классов,  $y_i$  - фактическое значение (*one – hot* кодирование),  $\hat{y}_i$  - предсказанная вероятность класса.  $CCE$  обычно применяется в задачах многоклассовой классификации.

### Бинарная перекрестная энтропия (*Binary Cross – Entropy*)

$$BCE = - \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

где  $N$  - количество классов,  $y_i$  - фактическое значение,  $\hat{y}_i$  - предсказанная вероятность класса.  $BCE$  обычно используется в задачах бинарной классификации.

Затем рассмотрим обучение нейронных сетей:

**Определение 3.6.3.** *Backpropagation* (обратное распространение ошибки) - ключевой алгоритм, используемый для обучения нейронных сетей. Он позволяет вычислить градиент функции потерь по весам сети, что позволяет оптимизировать эти веса с использованием алгоритмов градиентного спуска.

Процесс обучения нейронной сети с использованием *backpropagation* включает следующие шаги:

1. Инициализация - веса заполняются случайными значениями (или из какого-то распределения).
2. Прямое распространение - входные данные передаются через сеть от входного слоя до выходного слоя.
3. Расчет функции потерь - сравнивается выход модели с ожидаемыми выходами, и вычисляется значение функции потерь.
4. Обратное распространение - с помощью алгоритма *backpropagation* вычисляется градиент функции потерь по каждому весу в сети.
5. Обновление весов - с использованием оптимизационного алгоритма, например, градиентного спуска, веса нейронной сети корректируются в направлении, противоположном градиенту функции потерь.

**6.** Повторение процесса: Процессы прямого и обратного распространения повторяются для каждой эпохи.

Обучение нейронной сети продолжается до достижения заданного критерия останова, например, заданного числа эпох или достижения определенной точности предсказаний.

**Определение 3.6.4.** Произошла одна эпоха (*Epoch*) - весь датасет прошел через нейронную сеть в прямом и обратном направлении только один раз.

**Определение 3.6.5.** Батч (*Batch*) - маленькая партия данных при их делении. Данное деление необходимо, потому что нельзя пропустить через нейронную сеть разом весь датасет из-за вычислительных мощностей.

**Определение 3.6.6.** Итерации - число батчей, необходимых для завершения одной эпохи.

Данные параметры являются гиперпараметрами, которые могут приводить к переобучению или недообучению. Также, помимо  $L1/L2$  регуляризации есть метод *dropout* для борьбы с переобучением.

### 3.6.3 *Dropout/Batch Normalization*

**Определение 3.6.7.** *Dropout* - это метод регуляризации, идея которого заключается во временном и случайном отключении (обнулении) некоторых нейронов во время процесса обучения.

Во время применения *dropout*, каждый нейрон входного или скрытого слоя с вероятностью  $p$  может быть временно удален из сети на каждой итерации обучения. Это означает, что его выходные значения игнорируются, и связи, входящие и исходящие из этого нейрона, не участвуют в *backpropagation*. Основная идея состоит в том, что отключение случайных нейронов во время обучения предотвращает сильную взаимозависимость между нейронами и принуждает сеть к более устойчивому обобщению.

Отмечу ещё один важный этап в нейронных сетях - нормализацию:

**Определение 3.6.8.** *Batch Normalization (BN)* - это техника, применяемая в глубоких нейронных сетях для нормализации входов каждого слоя путем их корректировки и масштабирования. Она направлена на решение проблемы внутреннего ковариационного сдвига, который представляет собой изменение распределения входов слоев в процессе обучения (разница статистических характеристик).

Основная идея *batch normalization* состоит в нормализации входов слоя путем вычитания среднего значения и деления на стандартное отклонение батча данных. Этот этап нормализации помогает стабилизировать процесс обучения, уменьшая зависимость сети от масштаба входных данных. Данный процесс также помогает в борьбе с переобучением, т.к приводит каждый слой к одному масштабу.

*Batch Normalization* обычно применяется после линейного преобразования и перед функцией активации в каждом слое сети. Во время обучения *BN* вычисляет среднее значение и стандартное отклонение входов внутри каждого батча и использует их для нормализации входов. Нормализованные входы затем масштабируются и смещаются с помощью обучаемых параметров, называемых  $\gamma$  и  $\beta$ , которые позволяют сети научиться оптимальному масштабу и смещению для каждого слоя.

$$\begin{aligned}\mu_B &= \frac{1}{m} \sum_{i=1}^m x_i \\ \sigma_B^2 &= \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \\ \hat{x}_i &= \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \\ y_i &= \gamma \hat{x}_i + \beta\end{aligned}$$

где  $m$  - размер батча (*batchsize*),  $x_i$  - вход,  $\mu_B$  - среднее значение батча,  $\sigma_B^2$  - дисперсия батча,  $\hat{x}_i$  - нормализованный вход,  $\gamma$  - масштабный параметр (*scaling parameter*),  $\beta$  - смещение (*shift parameter*),  $\epsilon$  - небольшая константа для численной стабильности.

## 3.7 Предсказание доходности с помощью *ML* и *DL*

### 3.7.1 Линейная/Логистическая регрессия

Сейчас мы создадим простую модель линейной регрессии для предсказания доходности (*return*), поскольку предсказание доходности (*return*) в трейдинге считается более предпочтительным, чем предсказание цен, поскольку основные решения принимаются на основе изменений в доходности актива, а не в его цене.

Однако стоит признать, что предсказание цен более весёлая затея, т.к создаёт иллюзию высокой точности и надёжности, но тем не менее является абсолютно бесполезным в трейдинге и инвестировании.

В этот раз возьмём данные инструмента *EUR/USD* за 2019 год с интервалом в 5 минут. Визуализируем данные (рис. 3.17)



Рис. 3.17: Визуализация данных

Теперь посчитаем доходность (*return*) (логарифмическую) и для построения простой линейной регрессии мы будем использовать один признак - смещение доходности на 1 вперёд, т.е этот признак показывает например в строке со временем 12 : 00 доходность за время 11 : 55. Назовём этот признак *return\_lag\_1*. Построим линейную регрессию и зависимость признака от таргета. Сразу видно, что между ними нет никакой зависимости и поэтому линейная регрессия тут сработает плохо (рис. 3.18.а). Более того, линейная регрессия нам говорит, что если в предыдущий раз у нас была положительная доходность, то в следующий раз она будет отрицательной (судя по наклону линии предсказания). Также построим на одном графике истинную доходность и предсказанную (рис. 3.18.б). Видно, что предсказание стабильно, в то время как доходность достаточно волатильна.

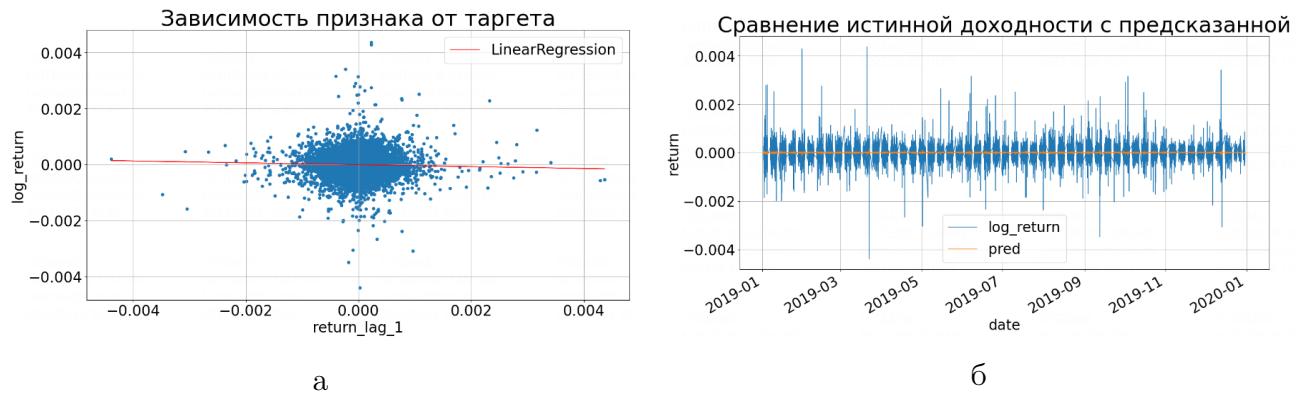


Рис. 3.18: Иллюстрация предсказания линейной регрессии

Понятно, что сама линейная регрессия предсказывает значения доходности плохо, од-

нако вдруг она хорошо показывает направление движения цен. Пусть  $-1$  означает, что цена падает, а  $+1$  что цена растёт. Обозначим поле, показывающее направление, как *movement\_sign*. Получается, что если направление верное, то

$$\text{sign}(\log_{\text{return}} \cdot \text{movement\_sign}) \geq 0$$

Введём понятие *hit ratio* - наша основная метрика качества.

**Определение 3.7.1.** *Hit Ratio (HT)* - это количество раз, когда был сделан правильный прогноз, по отношению к общему количеству предсказаний. В задаче классификации данная метрика аналогична *accuracy*.

Посчитаем *HT*, и мы получим:

$$\text{Hit Ratio (HT)} = 50.803\%$$

что чуть лучше обычного случайного предсказания (подбрасывания симметричной монетки)

Конечно, такое значение метрики нас не особо устраивает, поэтому предлагаю построить модель посложнее, добавив не только смещение на 1, но и смещение от 2-ух до 5-ти и посчитать *HT*. В этом случае оно получится:

$$\text{Hit Ratio (HT)} = 50.865\%$$

что чуть больше, чем простая модель линейной регрессии, однако всё равно не особо лучше обычного случайного предсказания.

Теперь проведём *backtesting* нашей стратегии на этих же данных (это называется *In – Sample Backtesting*), чтобы понять, на сколько увеличение *hit ratio* на 0.8% даёт нам увеличения в прибыли. Для этого посмотрим на фактическую кумулятивную доходность и на кумулятивную стратегическую доходность, как в 3.2. Получается, что когда предсказание нам будет показывать, что цена будет падать, то мы будем продавать и когда предсказание будет показывать, что цена будет расти, мы будем продавать. Таким образом, если мы угадали направление правильно, то мы будем иметь положительную доходность, а иначе отрицательную. Как мы видим на (рис. 3.19.a) такой маленький прирост в нашей метрике качества даёт прирост в доходности примерно на 50%. Однако это *gross return* и поэтому, посчитав количество транзакций (их около 50%) мы получим, что наша стратегия слишком высокочастотна  $\Rightarrow$  такого прироста в доходности не хватит, чтобы погасить комиссионные затраты *trading costs* (посчитаные, как в 3.2.3), что мы и получаем на (рис. 3.19.b)

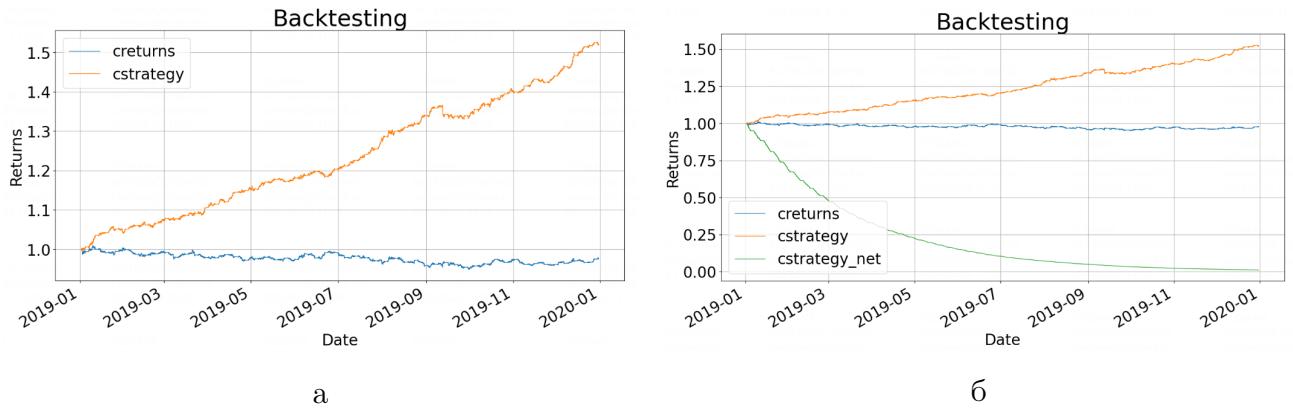


Рис. 3.19: *Backtesting*

Получается, что наша стратегия будет приносить нам большие убытки, судя из *In-Sample Backtesting*. Однако, когда мы делаем *In-Sample Backtesting* мы сталкиваемся ещё с одной проблемой, которая очень распространена в финансах - *Look Ahead Bias*

**Определение 3.7.2.** *Look Ahead Bias* (предвзятость прогноза) - ошибка, которая возникает, когда информация, которая стала доступной только после определенного момента времени, используется для создания модели или стратегии, которая должна принимать решения на основе информации, доступной только до этого момента времени.

Например, в нашем примере, когда мы делаем предсказание для тех же данных, на которых обучались, мы для любой точки временного отрезка используем информацию, которая стала доступна после этой точки временного отрезка (т.к коэффициенты модель сгенерировала после просмотра всех данных на данном временном отрезке), т.к например для предсказания направления тренда в начале 2019 года, мы не должны знать информацию о конце 2019 года и т.п. По итогу, это может привести к искажению результатов и созданию ложных представлений о возможностях прибыльности стратегии.

Поэтому мы не можем полагаться на *In-Sample Backtesting* и нам нужно провести *Forward Testing* для того, чтобы убедиться в том, что наша стратегия работает (хотя бы в плане *gross return*). Возьмём тестовую выборку с начала 2020-го года по сентябрь 2020-го года и проведём *Forward Testing* и посчитаем *hit ratio*. В итоге получим, что

$$Hit\ Ratio\ (HT) = 50.745\%$$

Также сама стратегия также не плохо даёт нам в *gross return* на новых данных (примерно на 30% больше в доходности (рис. 3.20.а), но если также посчитать количество транзакций (опять около 50%) и посмотреть на *net return* (рис. 3.20.б), то мы увидим, что всё также

плохо из-за очень высокой частоты трейдинга (похоже очень на 3.2.4)

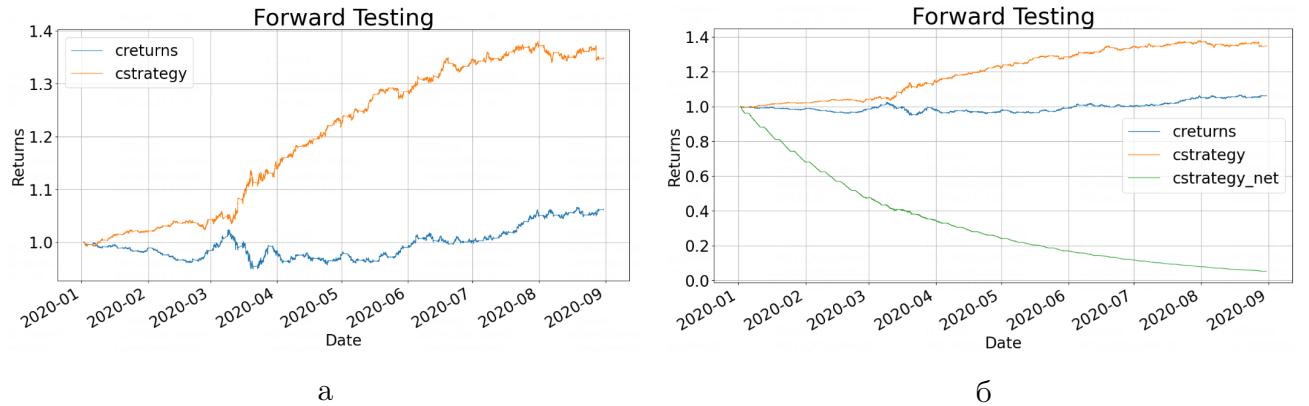


Рис. 3.20: *Forward Testing*

Попробуем теперь предсказывать сразу направление движения рынка, поскольку задача регрессии не дала особого результата из-за волатильности доходности. Для обучения модели логистической регрессии мы будем использовать коэффициент регуляризации ( $\lambda$ ) очень маленьким, т.к признаки имеют очень маленький коэффициент корреляции Пирсона между друг другом (рис. 3.21.а), поэтому тут нам регуляризация не нужна. Также я буду использовать не бинарную классификацию, а классификацию на классы  $-1, 0, 1$ , поскольку в фактической переменной бывает и нулевая доходность. Как итог, я буду использовать тип классификации - один против всех (*One vs Rest*). Однако, посмотрев на *In – Sample Backtesting* мы увидим ту же самую картину, что на (рис. 3.19.а), но количество транзакций сократилось на 12%, что всё-равно считается высокочастотным и картина с чистой прибылью повторяет (рис. 3.19.б). Что касается *Forward Testing*, то он уже выглядит похуже, чем то, что давала нам линейная регрессия, поскольку доходность на новых данных увеличивается на 26% (рис. 3.21). Что касается *hit ratio*, то на *backtesting* она лучше чем у линейной регрессии, а вот на *ForwardTesting* чуть хуже чем у линейной регрессии

$$\text{Hit Ratio Backtesting (HT)} = 51.00\%$$

$$\text{Hit Ratio Forward Testing (HT)} = 50.88\%$$

Как мы видим, особой разницы в предсказании нет, поэтому в дальнейшем я предлагаю предсказывать именно направление движения рынка.

Итого, в основном основная проблема в применении линейной и логистической регрессий в задаче предсказания доходности заключается не только в том, что признаки не линейной связаны с таргетом, но и в том, что невязка не распределена нормально, т.к доходность

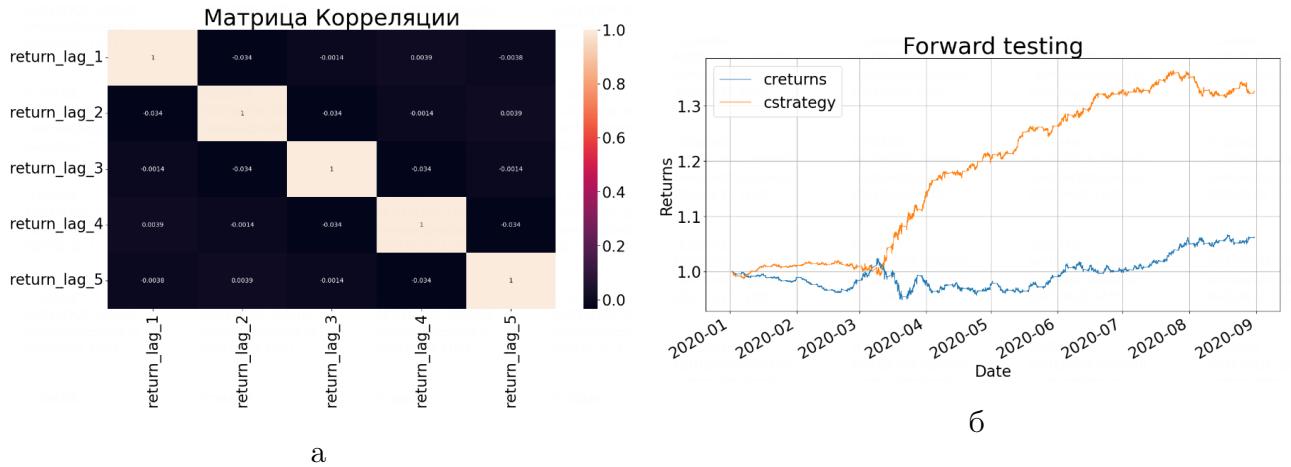


Рис. 3.21: Матрица корреляции и *Forward Testing*

имеет распределение близкое к распределению Стьюдента. Более того, разница между предсказанием и фактом не является ошибкой, а тоже хранит информацию (распределение ошибки выглядит примерно, как (рис. 3.18.б)). Именно по этой причине я даже не стал добавлять никаких признаков из стратегий из 3.2, т.к знал, что данные типы моделей не могут отлавливать такие сложные зависимости. Более того, с этими проблемами достаточно хорошо могут справиться временные ряды или нейронные сети (о которых пойдёт речь дальше).

### 3.7.2 Полносвязная нейронная сеть (*DNN*)

Теперь перейдём к построению своей собственной полносвязной нейронной сети с помощью фреймворка TensorFlow. Я буду использовать данные финансового инструмента *EUR/USD*, которые я использовал в 3.2, поскольку для них мы знаем оптимальные параметры для каждой рассмотренной стратегии. В качестве обучения я возьму 70% данных, а в качестве теста 30% (возьмём по времени, т.е. наиболее недавние данные будут в тесте). Теперь приступим к генерации признаков для нейронной сети с помощью стратегий из 3.2:

- **Direction** - направление движения рынка (наш таргет), где мы ставим единицу, если доходность больше нуля и нуль иначе.
- **Simple SMA** - возьмём короткую *SMA* с окном 61 и длинную *SMA* с окном 88 и посчитаем стратегию, как в 3.2.3.
- **Bollinger** - возьмём скользящую среднюю с окном 14 и два стандартных отклонения с таким же окном и посчитаем стратегию, как в 3.2.5.

- **Momentum** - будем брать отрицательный *momentum (contrarian)* с окном 2 (как в 3.2.4).

И к генерации ещё дополнительных признаков (отсечки выбраны наугад):

- **Min** - расстояние между минимальной среди 50-ти предыдущих цен и текущей ценой в процентах.
- **Max** - расстояние между максимальной среди 50-ти предыдущих цен и текущей ценой в процентах.
- **Volatility** - разброс (стандартное отклонение) предыдущих 10-ти значений.

Теперь превратим эти признаки в признаки со смещением. Сделаем 5 смещений по каждому признаку (как в 3.4), в итоге получим  $7 \cdot 5 = 35$  признаков и один таргет (*direction*). Затем сделаем нормализацию признаков (в тестовой выборке матожидание и дисперсию будем использовать, полученные на обучающей выборке):

$$\frac{X - \mu}{\sigma}$$

Теперь можем переходить к построению полносвязной нейронной сети (*DNN*), которая состоит из следующих слоёв:

- **Input Layer** - состоит из 35 нейронов (наших признаков)
- **4 Hidden Layer** - состоят из 50 нейронов с функцией активации *ReLU* (4 слоя и 50 нейронов были подобраны фреймворком *optuna*)
- **Output Layer** - состоит из одного нейрона с функцией активации *Softmax*

Также стоит отметить, что между каждым слоем используется *dropout* для того, чтобы избежать переобучения. Теперь определим метрики качества, функцию ошибки и основные параметры:

- **Функция ошибки** - бинарная кросс-энтропия (*BCE*)
- **Метрика оценки качества** - *accuracy* (она же *hit ratio*)

Количество эпох равно 50, размер батча равен 32, валидационный *split* равен 20%. Также я передал начальные веса классов, для того, чтобы избежать их дисбаланса. Теперь визуализируем функцию ошибки и метрику качества (рис.3.22).

Теперь проводим нормализацию тестовой выборки с теми же параметрами, что получили на обучающей выборке. По итогу на последнем слое получаем вероятности на выходе со следующим распределением (рис.3.23) на обучающей выборке и на teste.

Выберем отсечки для вероятностей следующим образом: если вероятность меньше 47%, то тогда мы говорим, что рынок точно движется в отрицательном направлении. Если больше

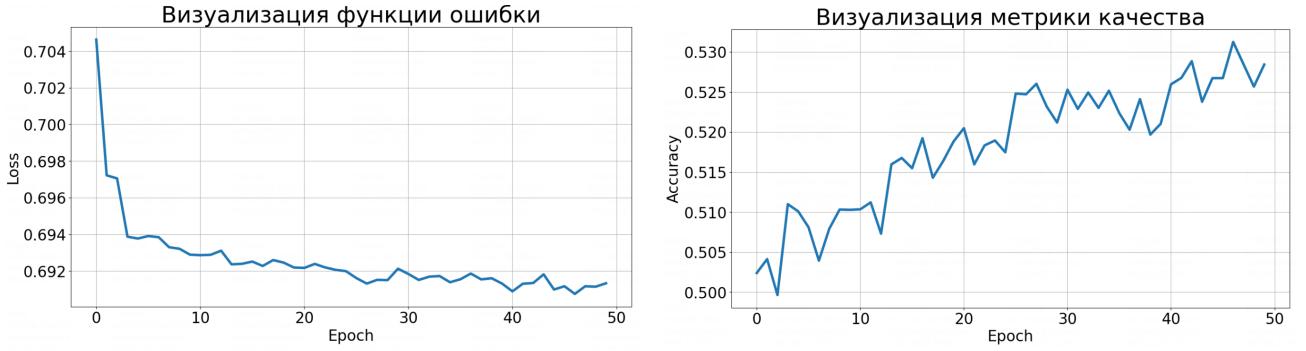


Рис. 3.22: Визуализация функции ошибки и метрики качества

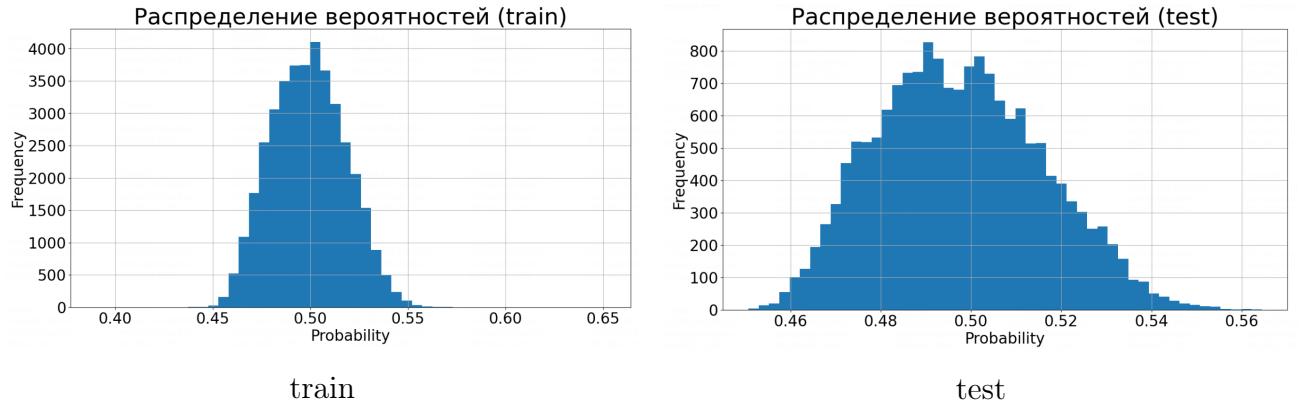


Рис. 3.23: Распределение вероятностей

52%, то точно в положительном направлении. Всё, что между заполняем *forward filling*, т.е предыдущими значениями. Мы это делаем, потому что основное скопление вероятности лежит между 47% и 52% (примерно 82%) и поэтому не понятно, какое решение принимать наверняка, однако можно определить какие-то точные отсечки для принятия решения, которые не будут лежать в этом скоплении и затем придерживаться направлений, которые задали эти отсечки.

Посчитаем значение метрики качества на тестовой выборке:

$$\text{Accuracy} (\text{Test}) = 52.11\%, \text{ посчитано моделью}$$

$$\text{Accuracy} (\text{Test}) = 50.67\%, \text{ посчитано мной с моей отсечкой}$$

Как мы видим, качество с отсечкой упало, однако всё-равно лучше случайного предсказания. Более того, нам на самом деле не нужно знать направление рынка каждые 20 минут, т.к если мы будем покупать и продавать как только будет изменяться направление рынка, то мы конечно же получим хороший *gross profit*, но такой трейдинг будет крайне высокочастотным, что приведёт к огромным потерям, как мы видели в 3.2. Поэтому, хоть качество

упало, оно не полностью отражает хорошо ли работает наша стратегия, полученная нейронной сетью и нашей отсечкой. Поэтому посмотрим на *backtesting* нашей стратегии для определения сначала *gross profit* (рис.3.24.а). Как и ожидалось, судя по метрике качества у нас получился достаточно не плохой прирост на примерно 10%. Теперь посмотрим на *backtesting* учитывая *trading costs*, который посчитаем также, как и в 3.2.3. Получим, что и в *net profit* мы в плюсе примерно на 3% (рис.3.24.б) и всё это из-за достаточно низкой частоты совершения операций (примерно 3.3%). Получается, что стратегия подкреплён-

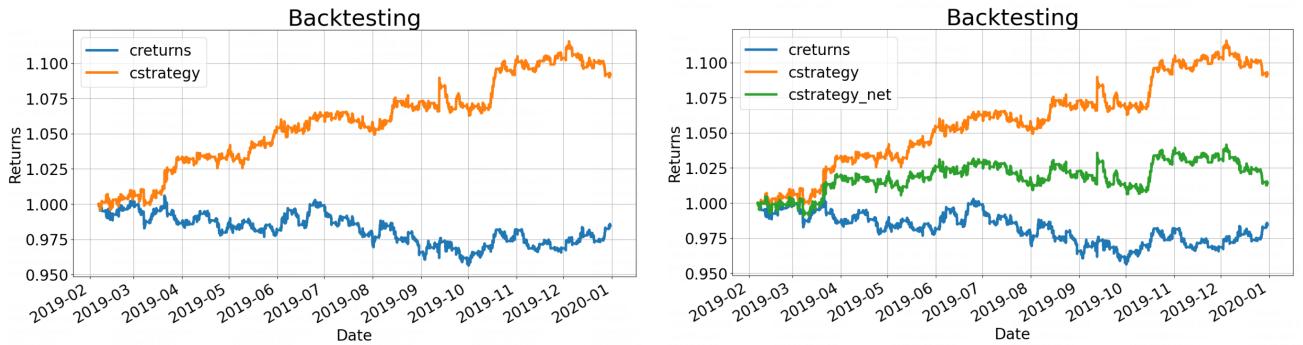


Рис. 3.24: *DNN Backtesting*

ная машинным обучением и вправду позволила нам получить прибыль, хоть и не такую большую.

## ЗАКЛЮЧЕНИЕ

Данную дипломную работу можно поделить на три полноценных направления:

- Трейдинг - рассмотрены основные понятия, которые используются в трейдинге. Также было уделено особое внимание различным метрикам оценки перформанса. Были рассмотрены различные типы и виды трейдинга и их отличия. Отдельно был рассмотрен трейдинг с плечом и были подчёркнуты его основные преимущества и недостатки.
- Доходность - рассмотрено понятие доходности, различные ее виды и особенности. Было выявлено, что распределение доходности является не совсем нормальным и поэтому были предприняты попытки разобраться в чём их отличия. В итоге, были применены различные понятия из прикладной статистики. Исходя из  $qq-plot$  было выявлено, что распределение точно не является нормальным из-за более тяжёлых хвостов. Коэффициент асимметрии и доверительные интервалы показали, что распределение доходности смещено влево, что в свою очередь говорит нам о потенциально большем числе убытков, нежели прибыли. Коэффициент эксцесса же дал численное значение "тяжести" хвостов и показал, что эти убытки могут ещё быть и потенциально огромными. Для приближения распределения было предложено два метода. Один из них заключался в выборе распределения Стьюдента, для которого также был подсчитан коэффициент эксцесса, который оказался достаточно близко к реальному коэффициенту. Также был построен  $qq-plot$  для сравнения распределения доходности и распределения Стьюдента. Результат показал, что они достаточно близки. Второй метод заключался в использовании алгоритма кластеризации  $GMM$ . Были взяты два нормально распределённых кластера, которые в сумме должны дать нормальное распределение, но с более тяжёлыми хвостами. По итогу, коэффициент эксцесса у такого распределения оказался чуть ближе к распределению доходности, чем у распределения Стьюдента, хотя визуально картинки не особо отличаются.

- Стратегии и машинное обучение - были рассмотрены основные задачи машинного обучения (классификация, регрессия, кластеризация). Были рассмотрены одни из основных стратегий в трейдинге (*buy and hold*, *SMA crossover*, *contrarian*, *momentum* и *mean – reversion*) и методы их тестирования (*backtesting* и *forward testing*). Подробно были разобраны проблемы высокочастотного трейдинга и оценено его влияние в стратегиях. Были построены стратегии с помощью признаков смещения в линейной и логистической регрессии и было показано, что они не умеют отлавливать такие сложные зависимости, подверженные волатильности. Более того, невязка в этих методах не была ошибкой, а тоже несла в себе информацию. По итогу был сделан вывод, что все эти стратегии и признаки в отдельности и не в очень сложных моделях не могут принести прибыль, а могут наоборот нести убытки. Было предложено два решения этой проблемы: временные ряды и нейронные сети. Разобрано было только одно решение - последнее. По итогу, была построена полносвязная нейронная сеть (*DNN*) с 4-мя скрытыми слоями и 50-ю нейронами в каждом. В качестве признаков были добавлены все стратегии, волатильность, максимальное и минимальное изменение цен за какой-то промежуток. Были выбраны отсечки для вероятности движения рынка и посчитаны метрики качества и разница в *gross* прибыли. Была оценена общая цена транзакций и посчитана чистая прибыль. В итоге, нейронная сеть показала свою эффективность и позволила получить чистую прибыль.

В дальнейшем хотелось бы рассмотреть следующие темы для более точного моделирования и предсказания результатов трейдинговых стратегий:

- Временные ряды и их преимущества над обычной линейной регрессией.
- Оптимизация портфеля (нейронная сеть даёт нам выигрыш, но не большой. Поэтому появляется идея, а что если использовать несколько активов одновременно).
- Реализация алгоритмического бота, применение нейронной сети непосредственно на бирже.
- Обучение с подкреплением (ещё один способ предсказывать тренды на рынке).
- Рассмотреть методы оптимизации в трейдинге (*ADMM*, проксимальный градиентный спуск и т.д.).

Такие исследования будут способствовать более эффективному и успешному трейдингу.

## ПРИЛОЖЕНИЯ

### Биржевые настройки (*Orders*) и их типы

**Определение 3.7.3.** Ордер (Order) - термин, относящийся к инструкции или запросу, направленному трейдером брокеру или торговой платформе для исполнения определенной операции с финансовым инструментом. Ордер указывает на желаемое действие трейдера, такое как покупка или продажа акций, валюты или других финансовых инструментов.

Перечислим типы ордеров (orders):

**Определение 3.7.4.** Market Order - это тип ордера, при котором трейдер запрашивает покупку или продажу актива по наилучшей текущей рыночной цене. Он выполняется немедленно по текущей цене, и гарантирует исполнение сделки, но не гарантирует конкретную цену исполнения. Этот тип ордера особенно полезен, когда трейдеру необходимо быстро войти или выйти из позиции.

**Определение 3.7.5.** Limit Order - это тип ордера, который трейдер использует для указания конкретной цены, по которой он хочет купить или продать актив (открыть позицию), при движении цены актива в правильном направлении (в том, в котором мы были уверены изначально). Трейдер указывает свою цену (лимит) и объем актива. Limit Order остается в системе ордеров до тех пор, пока не будет достигнута указанная цена или пока не закончится время его действия, которое указывает трейдер. Такой ордер позволяет трейдеру контролировать цену исполнения, но не гарантирует его исполнение.

**Определение 3.7.6.** Stop Order - это тип ордера, который трейдер использует для указания конкретной цены, по которой он хочет купить или продать актив (открыть позицию),

при движении цены актива в неправильном направлении (в том, которое противоречило нашим ожиданиям). Stop Order, как и Limit Order не гарантирует исполнение. Stop Order используется для активации сделок в направлении движения цены, чтобы защитить трейдера от потерь.

**Определение 3.7.7.** Take-Profit Order - это тип ордера, который используется для фиксации прибыли с открытой позиции. Трейдер указывает цену прибыли, на которой он хотел бы закрыть позицию. Когда цена актива достигает или превышает указанную цель прибыли, ордер активируется и позиция закрывается автоматически. В отличии от limit order, который может никогда не выполнится, take-order profit открывает позицию, но не факт, что её закроет, т.к цена может пойти в противоположном направлении

**Определение 3.7.8.** Stop-Loss Order - это тип ордера, который используется для ограничения потерь на открытой позиции. Трейдер указывает цену, при достижении или превышении которой позиция должна быть автоматически закрыта. Ордер стоп-лосс помогает трейдеру ограничить свои потери и защитить капитал

## Netting и Hedging

В трейдинге есть два разных подхода, которые используются для управления рисками и защиты от потерь: "netting" и "hedging".

**Определение 3.7.9.** Сетирование (Netting) - это процесс сокращения или сведения нескольких позиций или обязательств в одну позицию или обязательство

В контексте трейдинга, сетирование может использоваться для свертывания открытых позиций или сделок в одну консолидированную позицию. Например, если трейдер имеет несколько открытых позиций на покупку и продажу одного актива, он может использовать сетирование, чтобы свернуть все эти позиции в одну позицию с общим размером и направлением.

**Определение 3.7.10.** Хеджирование (Hedging) - это стратегия, при которой создаётся позиция, которая будет компенсировать потенциальные потери от другой позиции или актива. Если одна позиция или актив подвергается неблагоприятным изменениям цены, хеджирование может помочь смягчить эти потери, поскольку компенсирующая позиция или актив изменяется в противоположном направлении. Хеджирование позволяет трейдерам снизить риск, связанный с неопределенностью рынка.

В случае с трейдингом у брокера, netting - является стратегией по умолчанию. При использовании стратегии hedging трейдер будет платить комиссию (Trading Costs) дважды. Например, пусть мы открыли длинную позицию на 100000 юнитов и через некоторое время мы стали меньше уверены в этой позиции и не хотим терять много денег и поэтому решаемся продать половину. В случае netting при открытии короткой позиции на 50000 юнитов мы получим одну длинную позицию на 50000 юнитов и потом ещё через некоторое время мы решили закрыть позицию. В общей сумме мы три раза платим Trading Costs. В случае hedging, на этапе открытия короткой позиции в 50000 юнитов мы получим две позиции, которые работают одновременно. Потом мы открываем ещё одну короткую позицию в 50000 юнитов и получаем две позиции - короткую и длинную в 100000 юнитов каждая. Затем мы закрываем каждую. По итогу мы 5 раз платим Trading Costs

## ЛИТЕРАТУРА

1. Научные статьи и исследования с образовательных, ИТ и бизнес порталов: Medium, Vc.ru, Financial-hacker, ITInvestCapital, journal.tinkoff, investopedia
2. Habr: Метрики в задачах машинного обучения; Habr: Открытый курс машинного обучения; Habr: Нейронные сети
3. Stepik: Deep Learning School
4. Курс по прикладной статистике от МФТИ : PSAD
5. Coursera: "Машинное обучение и анализ данных" от МФТИ и Yandex
6. Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python, 2nd Edition 2nd Edition by Stefan Jansen, 2020
7. Udemy: Financial Engineering and Artificial Intelligence in Python; Udemy: Algorithmic Trading A-Z with Python, Machine Learning AWS
8. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 3rd Edition, 2022
9. The Machine Learning Bible: [4 in 1] From Scikit-Learn to Pytorch and Everything in between to Build Smart Systems – Top Secret Tips and Tricks to Break the System Design Interview Paperback, 2023
10. Probabilistic Machine Learning: An Introduction (Adaptive Computation and Machine Learning series) by Kevin P. Murphy, 2022

- 11.** Trading Strategies: Day Trading + Swing Trading. A Beginner's Guide to Trading with Easy and Replicable Strategies to Maximize Your Profit. How to Use Tools, Techniques, Risk Management, and Mindset Paperback, 2022
- 12.** Mean Reversion Trading: Using Options Spreads and Technical Analysis Kindle Edition by Nishant Pant, 2022