Оглавление

[ВВЕДЕНИЕ 2](#_Toc103427870)

[1. КЛАСТЕРИЗАЦИЯ ФИНАНСОВЫХ АКТИВОВ КАК ВРЕМЕННЫХ РЯДОВ 5](#_Toc103427871)

[1.1. Классическая постановка задачи кластеризации 5](#_Toc103427872)

[1.2. Постановка и особенности задачи кластеризации финансовых активов временных рядов 6](#_Toc103427873)

[1.3. Алгоритм динамической трансформации временной шкалы 8](#_Toc103427874)

[1.4. Кластеризация финансовых активов по описательным статистикам 11](#_Toc103427875)

[1.5. Кластеризация финансовых активов по скрытому векторному представлению 13](#_Toc103427876)

[1.6. Алгоритмы решения задачи кластеризации 14](#_Toc103427877)

[1.7. Сбор и обработка данных финансовых активов индекса S&P 500 20](#_Toc103427878)

[1.8. Кластеризация финансовых активов индекса S&P 500 23](#_Toc103427879)

[2. ПРОГНОЗИРОВАНИЕ ДОХОДНОСТИ ФИНАНОСОВЫХ АКТИВОВ 30](#_Toc103427880)

[2.1. Постановка задачи прогнозирования доходности финансовых активов как временных рядов 30](#_Toc103427881)

[2.2. Постановка задачи классификации динамики доходности финансовых активов 33](#_Toc103427882)

[2.3. Алгоритмы прогнозировании временных рядов и классификации 35](#_Toc103427883)

[2.4. Регрессионный подход к прогнозированию динамики индекса S&P 500 39](#_Toc103427884)

[2.5. Классификационный подход к прогнозированию динамики индекса S&P 500 39](#_Toc103427885)

[3. ФОРМИРОВАНИЕ ОПТИМАЛЬНОЙ СТРУКТУРЫ ИНВЕСТИЦИОННОГО ПОРТФЕЛЯ 41](#_Toc103427886)

[ЗАКЛЮЧЕНИЕ 42](#_Toc103427887)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 43](#_Toc103427888)

# **ВВЕДЕНИЕ**

В современном мире финансовые рынки представляют все больший и больший интерес как крупным, так и средним инвесторам, которые заинтересованы в страховании своих денежных средств и накоплений от инфляции, будь то долгосрочные инвестиции с целью получения дивидендов и получения прибыли от роста доходности финансового актива или краткосрочные инвестиции с целью спекуляции на ценах финансовых активов.

Именно поэтому интерес к финансовым рынкам растет не только у инвесторов, но и аналитиков, брокеров, банкиров, страховых и инвестиционных компаний, которые рассматривают финансовый рынок как объект бизнеса и инструмент дохода.

Проблемы анализа, прогнозирования и кластеризации финансовых активов являются очень сложными в силу целого ряда причин, – высокая волатильность рынка, скрытые рыночные зависимости и закономерности, высокая зависимость динамики от новостного фона и спроса инвесторов на финансовый актив на финансовом рынке. Поэтому, динамику цен финансовых активов можно рассматривать как временные ряды и использовать соответствующие модели машинного обучения с целью решения описанных выше проблем.

В настоящее время, в связи с увеличением интереса бизнеса к популярным методам машинного обучения, анализа данных и большим данным, инструментарий анализа, прогнозирования и кластеризации финансовых активов регулярно пополняется новыми методами машинного и глубинного обучения, которые рассматриваются в современных зарубежных исследованиях по данной тематике.

**Актуальность данной выпускной квалификационной работы** объясняется важностью и востребованностью в современном мире анализа и моделирования процессов и динамики развития рынка ценных бумаг, развитием машинного обучения как современного универсального инструмента решения поставленных бизнес-задач, доступностью огромных объемов соответствующих статистических данных о динамике ценных бумаг, а также относительной научной редкостью использования методов машинного обучения в научных работах на исследуемую тематику.

**Целью выпускной квалификационной работы является** проведение анализа и моделирования рынка ценных бумаг с целью выявления рыночных зависимостей и закономерностей с применением современных методов машинного и глубинного обучения.

**Для достижения данной цели были поставлены следующие задачи:**

1. Провести обзор и анализ существующих научных исследований на данную тему.
2. Изучить и рассмотреть различные источники статистических данных о динамике ценных бумаг, собрать и обработать соответствующие статистические данные.
3. Изучить, описать и применить различные подходы к кластеризации финансовых активов как временных рядов, сравнить результаты и оценить качество кластеризации.
4. На основе результатов кластеризации выделить «устойчивые» кластеры финансовых активов с целью дальнейшей диверсификации портфеля ценных бумаг.
5. Изучить, описать и применить различные подходы к прогнозированию динамики финансовых активов как временных рядов, сравнить результаты и оценить качество прогнозирования.
6. Изучить, описать и применить различные подходы к решению задачи распределения активов в инвестиционном портфеле, используя предыдущие результаты решения задач кластеризации финансовых активов и прогнозирования динамики финансовых активов.
7. Сформировать инвестиционный портфель, используя результаты решения задачи распределения активов в инвестиционном портфеле, спрогнозировать его доходность, оценить на тестовых данных доходность и риск сформированного инвестиционного портфеля, оценить точность прогноза.

**Объектом работы** является рынок ценных бумаг, как один из элементов мирового финансового рынка.

**Предметом работы** являются процессы, зависимости и закономерности, определяющие динамику развития рынка ценных бумаг, совместное взаимодействие ценных бумаг и их влияние друг на друга.

Структурно работа состоит из трёх разделов. **В первом разделе** производится постановка и решение задачи кластеризации финансовых активов как временных рядов с целью последующей диверсификации портфеля, сравнение различных подходов к решению задачи кластеризации, оценка качества кластеризации. **Во втором разделе** производится постановка и решение задачи прогнозирования динамики финансовых активов, сравнение различных подходов к решению задачи прогнозирования, оценка качества прогнозирования. **В третьем разделе** производится постановка и решение задачи распределения активов в инвестиционном портфеле.

# **1. КЛАСТЕРИЗАЦИЯ ФИНАНСОВЫХ АКТИВОВ КАК ВРЕМЕННЫХ РЯДОВ**

## **1.1. Классическая постановка задачи кластеризации**

Прежде, чем рассматривать задачу кластеризации временных рядов, рассмотрим классическую задачу кластеризации в общем виде.

Кластеризация – задача объединения множества объектов в однородные группы (их называют кластера) так, чтобы объекты по своим признакам из одной группы были более похожи друг на друга, чем на объекты из других кластеров.

Задача кластеризации – относится к ряду классических задач машинного обучения без учителя. Обучение без учителя – такой способ обучения, при котором у каждого объекта нет метки класса, и требуется обнаружить в множестве объектов внутренние зависимости, взаимосвязи и зависимости.

Приведем математическую постановку задачи кластеризации. Дано множество – множество объектов обучающей выборки, – -ый объект обучающей выборки, представляющий собой вектор , где – количество признаков в обучающей выборки, дано множество меток кластеров . Для любой пары объектов обучающей выборки задан критерий схожести (функция расстояния): .

Необходимо разбить обучающую выборку на кластеры таким образом, чтобы объекты внутри одного кластера были близки относительно функции расстояния .

Существует много различных мер близости объектов внутри одного кластера, и, как правило, для каждого алгоритма кластеризации, существует своя классическая мера близости, обозначим такую меру близости , где – вектор меток кластеров для каждого объекта обучающей выборки.

Более формально, необходимо найти такую функцию (алгоритм кластеризации) , которая определяет взаимно-однозначное соответствие между множеством меток кластеров и множеством объектов обучающей выборки, и, которая минимизирует меру близости объектов внутри одного кластера .

Очевидно, что не существует оптимального решения задачи кластеризации, поскольку, во-первых, результат решения задачи кластеризации зависит от функции расстояния между объектами , во-вторых, результат решения задачи кластеризации зависит от количества кластеров , в-третьих, нет объективных критериев качества кластеризации, как это, например, есть в задачах обучения с учителем.

## **1.2. Постановка и особенности задачи кластеризации финансовых активов временных рядов**

Кластеризация временных рядов решает часть таких практических задач, как:

1. Нахождение финансовых активов, обладающих схожей динамикой;
2. Объединение пациентов в группы по результатам их кардиограммы;
3. Поиск схожих аудиозаписей на основе звуковой спектрограммы.

В рамках данной выпускной квалификационной работы будет рассмотрена первая практическая задача – выделение финансовых активов, обладающих схожей динамикой. Выделение групп финансовых активов со схожей динамикой позволяет, во-первых, оценить степень диверсификации портфеля, диверсифицировать инвестиционный портфель, а, во-вторых, определять динамику финансовых активов, которые с некоторым лагом зависят от других финансовых активов – другими словами, это позволяет понять, вырастет или упадет стоимость финансового актива, исходя из того, вырос или упал актив из его кластера в последнюю единицу времени.

Как и в кластерном анализе объектов, качество результатов кластеризации временных рядов определяется алгоритмом кластеризации и функцией расстояния двух временных рядов.

Временные ряды задаются векторами , такая многоуровневая индексация объясняется тем, что временные ряды не обязательно имеют одинаковое количество наблюдений.

Что касается алгоритмов кластеризации – все алгоритмы имеют место быть, все зависит от специфики конкретной задачи. Однако, с расстоянием двух временных рядов не все так однозначно.

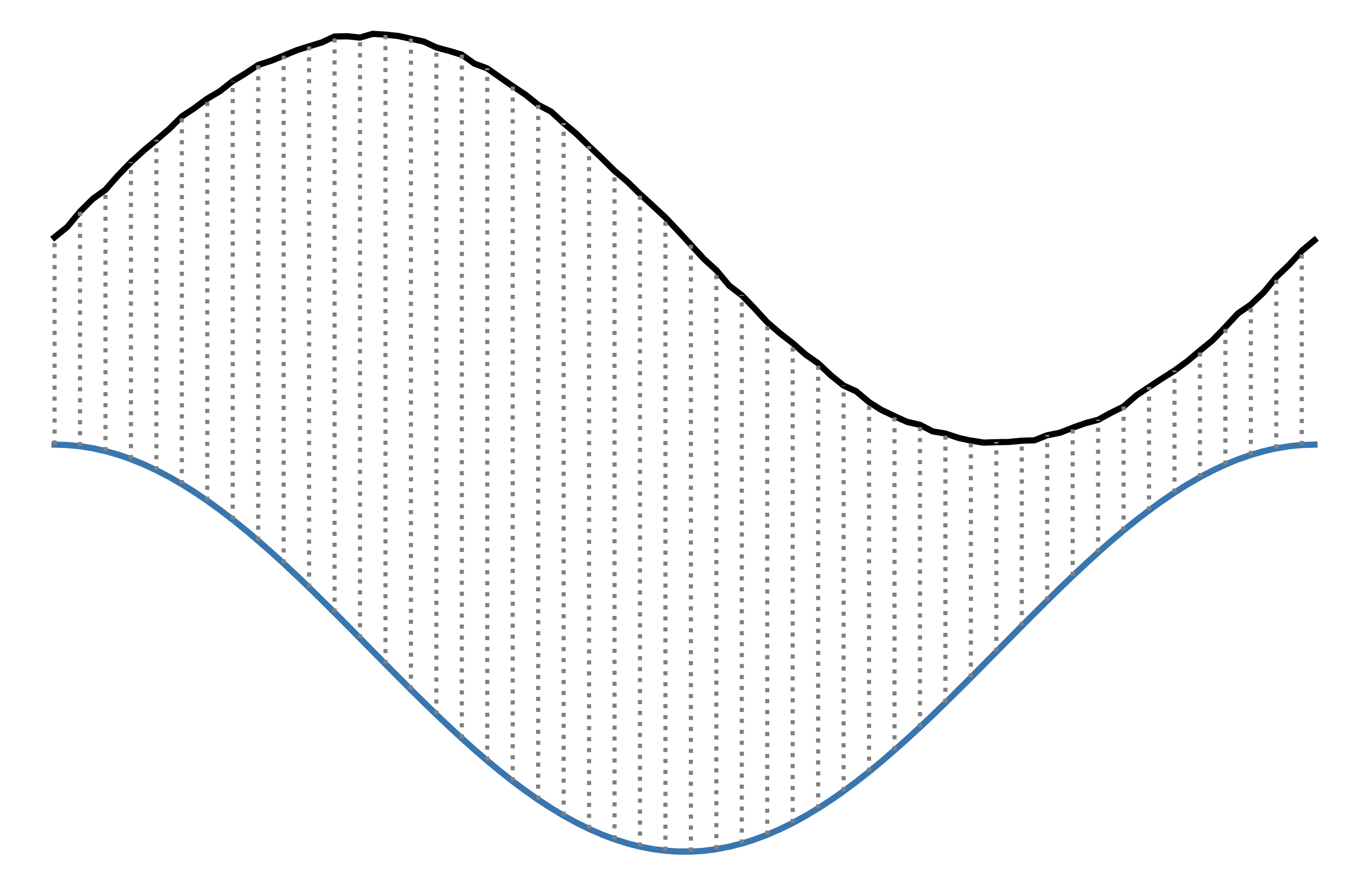
В случае, если временные ряды и имеют одинаковое количество наблюдений, то для них можно вычислить такие классические расстояния, как:

1. Евклидово расстояние
2. Манхэттенское расстояние
3. Расстояние Чебышева

В случае, если временные ряды и имеют различное количество наблюдений, то можно предпринять некоторые меры, чтобы привести ряды к одинаковому количеству наблюдений, например:

1. Обрезать слева либо справа временной ряд, в котором больше наблюдений – это называется каттинг (англ. cutting)
2. Заполнить слева либо справа временной ряд, в котором меньше наблюдений, нулями – это называется паддинг (англ. padding)

После этого, для полученных модификаций временных рядов можно вычислить рассмотренные выше классические расстояния. Фактически, такой подход вычисления расстояний между временными рядами можно визуализировать образом, показанным на рисунке ниже.



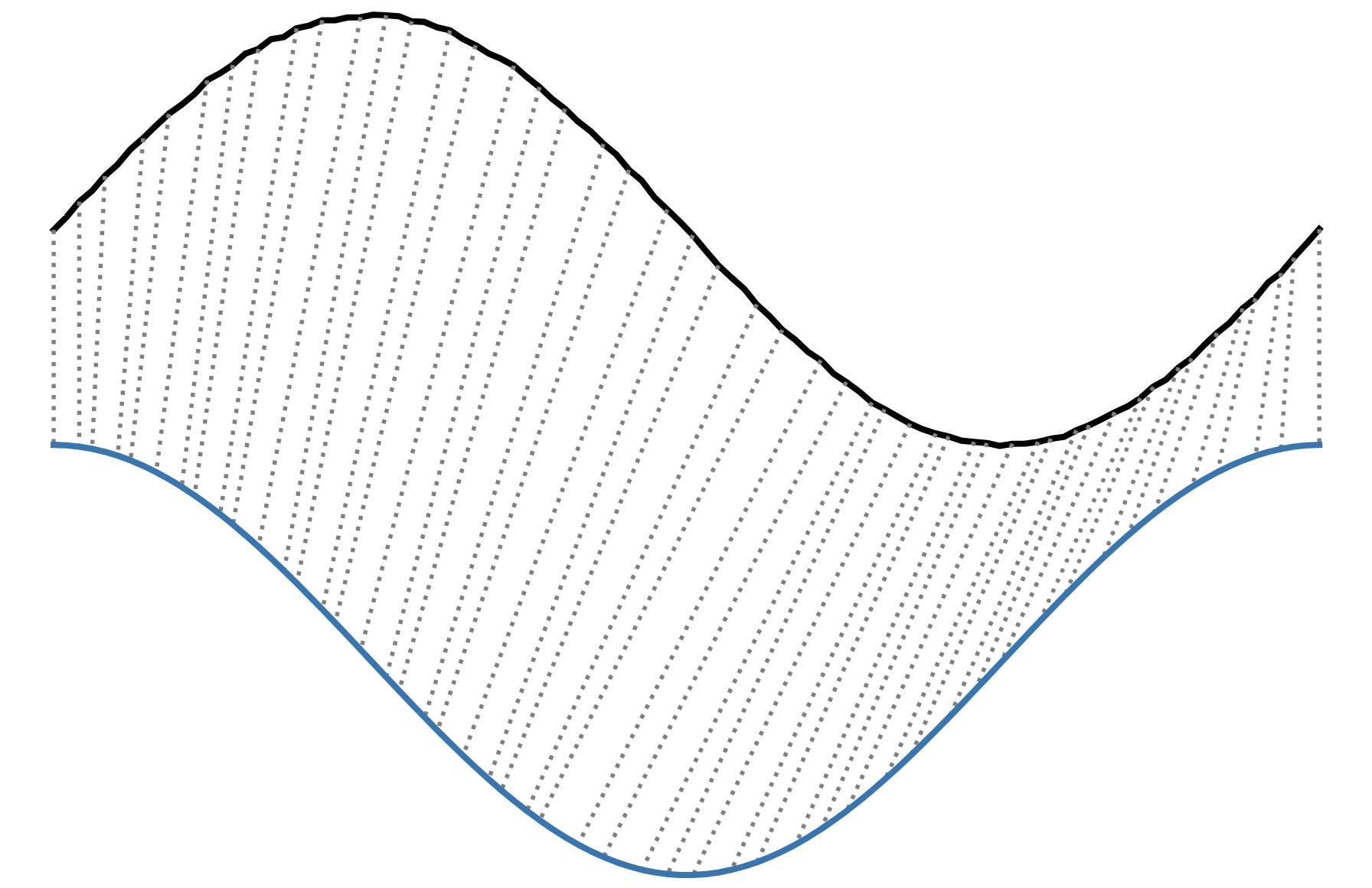
Однако, к несчастью, такой подход не совсем в полной мере отражает фактическую дистанцию, фактическое расстояние между временными рядами. Для примера, рассмотрим конкретный временной ряд и его копию, которая сдвинута по оси абсцисс влево – в таком случае, рассчитав любое классическое расстояние между рядами расстояние будет существенно отличаться от нуля несмотря на то, что временные ряды, фактически, идентичные.

Для решения данной проблемы существуют специальные меры расстояния между временными рядами, однако принято выделять, как правило, одно конкретное расстояние – DTW-расстояние, которое вычисляется по алгоритму динамической трансформации временной шкалы.

**1.3. Алгоритм динамической трансформации временной шкалы**

DTW-расстояние впервые было применено в задачи распознавания речи – для двух звуковых дорожек необходимо было определить, произнесены ли на них одинаковые смысловые фразы, учитывая разную скорость произнесения одинаковых фраз разными людьми.

Смысл алгоритма динамической трансформации временной шкалы, исходя из названия, состоит в том, что алгоритм трансформирует временной ряд (то есть, сжимает либо растягивает) для того, чтобы достичь оптимального сопоставления элементов двух последовательностей. Визуализация DTW-расстояния представлена на рисунке ниже.

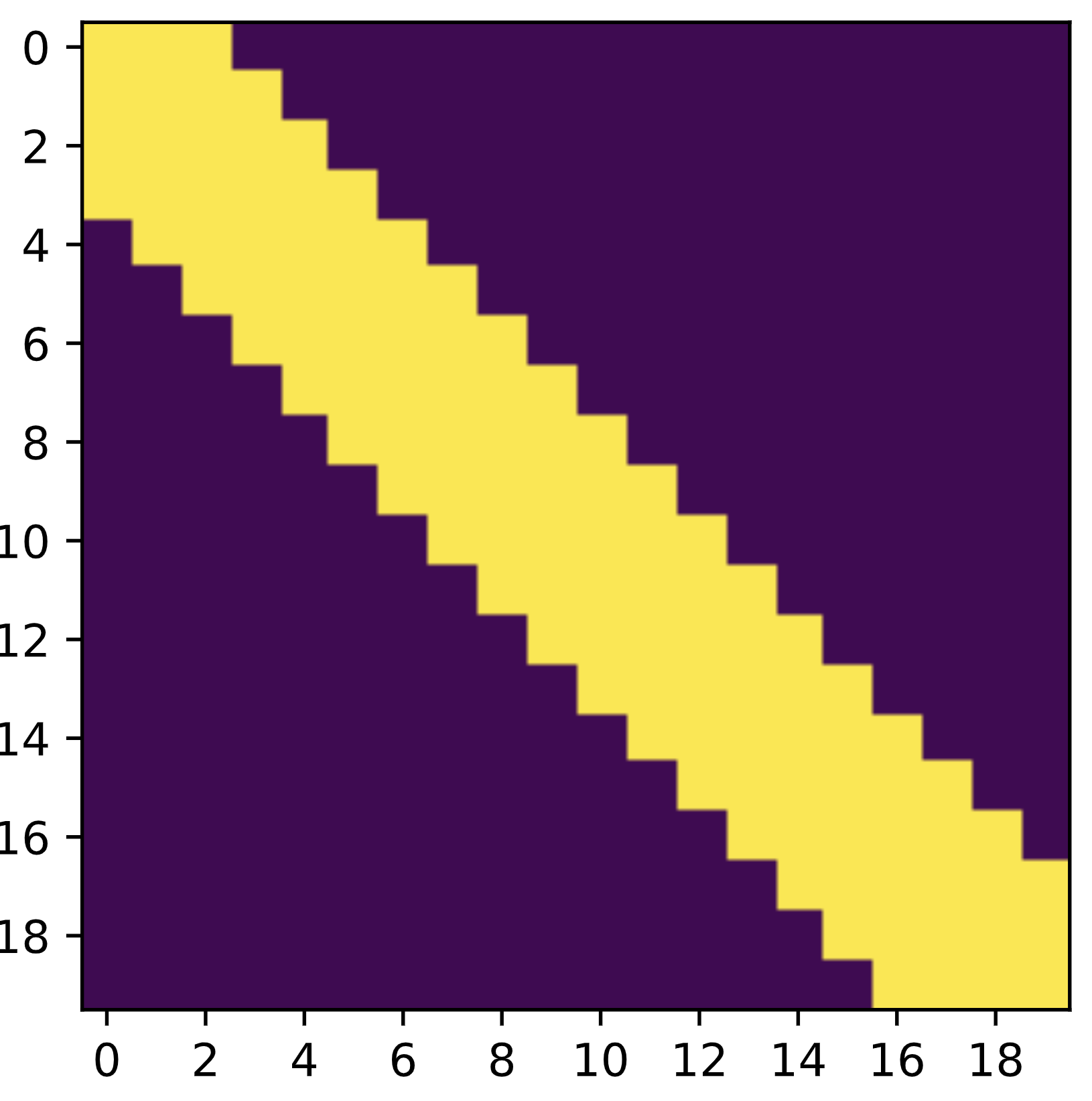


Рассмотрим математическую постановку DTW-алгоритма: даны два временных ряда и . Далее, считается прямоугольная матрица локальных потерь порядка , элемент которой соответствует расстоянию между -ым элементом временного ряда и -ым элементом временного ряда , в качестве такого расстояния, как правило, берется либо , либо . После чего находится такой путь в матрице из левого верхнего угла в правый нижний, который, во-первых, минимизирует расстояние между рядами и . Тогда DTW-расстояние вычисляется по следующей формуле:

Однако, количество всевозможных путей в матрице экспоненциально возрастает с ростом входных данных. Поэтому, чтобы гарантировать нахождение пути за полиномиальное время (в нашем случае за ), вводят локальное ограничение на соседние пары индексов в пути :

Фактически, это ограничение означает, что путь должен быть монотонным, другими словами, в рамках пути по матрице можно двигаться либо вправо, либо вниз, либо вправо вниз по диагонали.

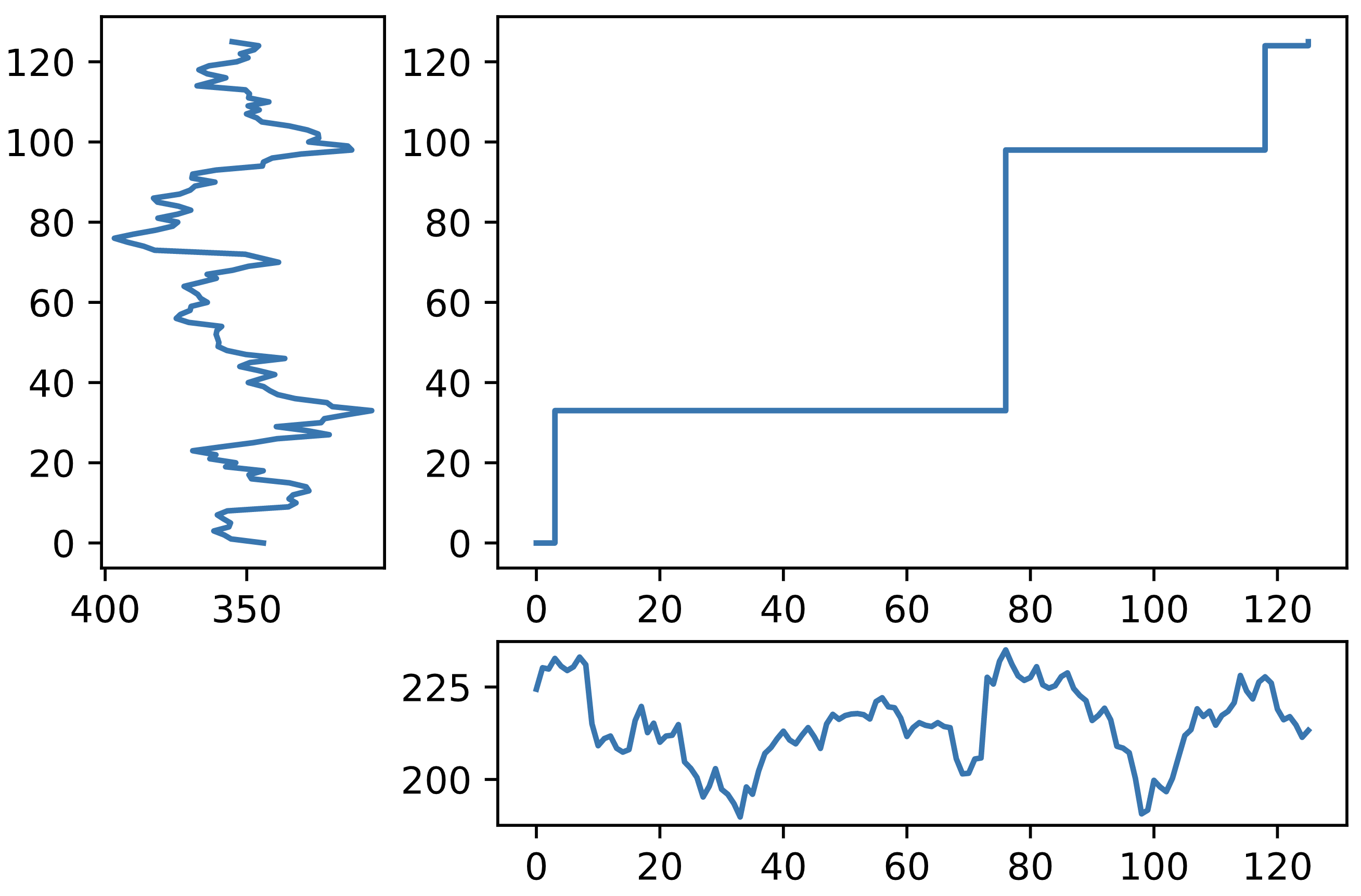
Стоит добавить, что иногда, помимо локального ограничения на соседние пары индексов, также вводятся глобальные ограничения, например, ограничение на ширину окна пути по матрице локальных потерь, в пределах которого разрешается прокладывать путь. Существуют различные ограничения на ширину окна, однако, классическим считается ограничение Сакэ-Чиба (англ. Sakoe-Chiba) – это ограничение на движение в рамках симметричной области вдоль главной диагонали матрицы локальных потерь. Пример такого глобального ограничения Саке-Чиба представлен на рисунке ниже.



Таким образом, нахождение DTW-расстояния сводится к задаче динамического программирования, с состоянием динамики, которое выражается в следующей формуле:

Где – матрица состояний динамики, элемент равен следующему выражению:

Существует классический вариант визуализации пути в матрице локальных потерь, который представлен на рисунке ниже. На рисунке ниже представлена визуализация результата DTW-алгоритма для двух финансовых активов компаний Visa и MasterCard.

**

## **1.4. Кластеризация финансовых активов по описательным статистикам**

В предыдущем подразделе был рассмотрен подход кластеризации финансовых активов в признаковом пространстве, сформированном на значениях стоимости финансового актива в единицу времени, то есть, на значениях временного ряда.

Однако, есть альтернативные подходы к формированию признакового пространства финансовых активов. Один из таких альтернативных подходов – формирование признаков на основе описательных статистик временных рядов.

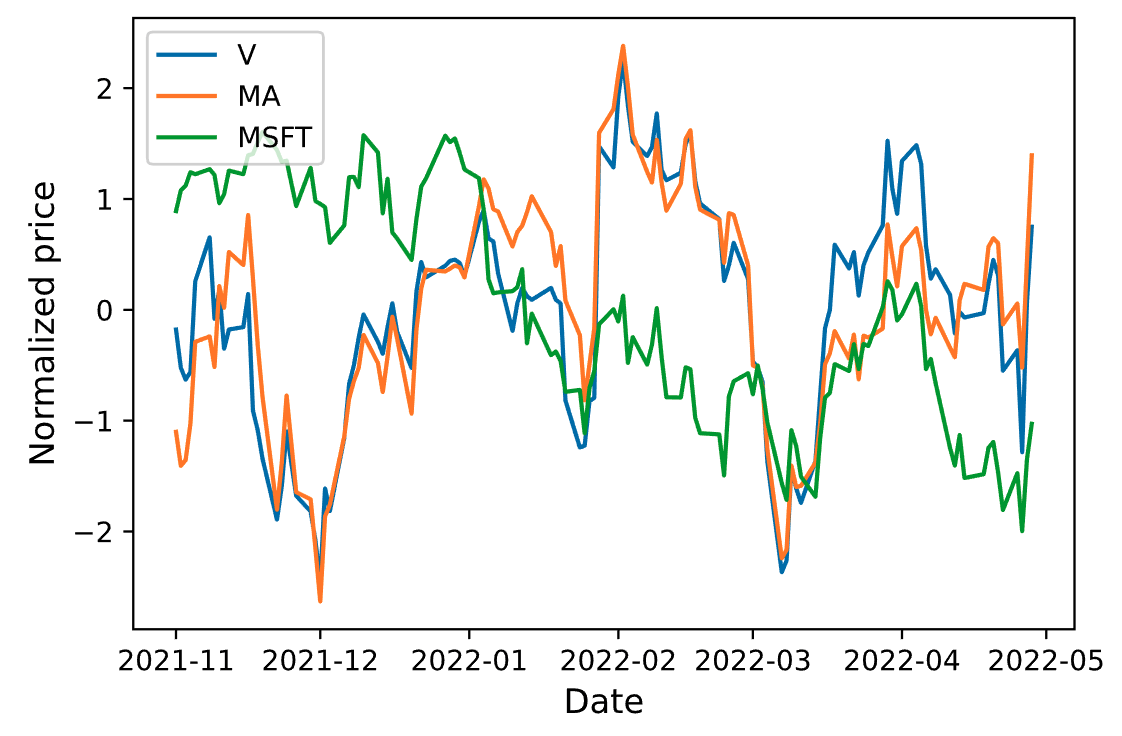
Идея состоит в следующем: если временные ряды имеют схожую динамику, то и их описательные статистики временных рядов не должны значительно различаться.

Тогда, способ кластеризации финансовых активов по описательным активам, состоит в том, чтобы для каждого ряда посчитать большое количество описательных признаков, а затем в полученном признаковом пространстве кластеризовать финансовые активы.

Как и для обычных признаков, можно считать такие стандартные описательные статистики, как среднее, медиана, дисперсия, стандартное отклонение, выборочные квантили. Однако, поскольку работа идет конкретно с временными рядами, то гарантируется, что наблюдения упорядочены во времени, тогда можно считать и статистики временных рядов – автокорреляция различных порядков, стационарность, сезонные компоненты, а также, обычные описательные статистики в окне – например, среднедневное значение, среднемесячное значение и другие всевозможные варианты.

Стоит отметить, что в данном случае, учитывание в признаковом пространстве автокорреляции решает проблему, зависимости пары временных рядов с некоторым лагом, которая в предыдущих подпунктах решалась с использованием алгоритма динамической трансформации временной шкалы.

В качестве примера схожих финансовых активов, у которых схожи описательные статистики, рассмотрим три компании – Microsoft, Visa и MasterCard, динамика нормализованных ежедневных цен закрытия которых представлена на рисунке ниже.



Из графика видно, что динамика Mastercard и Visa, можно сказать, идентична, – оно и понятно, потому что обе компании предоставляют международные услуги проведения платежных операций. Microsoft же, в свою очередь, имеет совсем другую динамику, поскольку деятельность Microsoft связана непосредственно с разработкой программного обеспечения.

Различные описательные статистики Microsoft, Visa и MasterCard представлены на рисунке ниже.

Изображение выглядит как стол

Автоматически созданное описание

Как видно, Visa и Mastercard схожи не только по динамике, но и по значениям описательных статистик, когда Microsoft, в свою очередь, имеет совсем другие описательные статистики.

## **1.5. Кластеризация финансовых активов по скрытому векторному представлению**

Другой, более современный подход к формированию признакового пространства – обучение искусственных нейросетей, имеющих архитектуру автокодировщика (англ. autoencoder), и затем, взятие эмбеддинга (векторного представления) на основе скрытого латентного представления для описания свойств финансового актива как временного ряда.

Рассмотрим подробнее архитектуру автокодировщика – это такая специальная архитектура искусственных нейронных сетей, при которой структура нейросети состоит из двух частей: первая – это энкодер (англ. encoder), а вторая – декодер (англ. decoder).

Автокодировщики используются для решения таких задач, как генерация новых объектов (из того же распределения, что и выборка), понижения размерности, формирования эмбеддингов на основе скрытого латентного представления автокодировщика. Ключевой особенностью автокодировщиков является то, что такой алгоритм обучения является обучением без учителя.

Алгоритм обучения автокодировщика следующий: во-первых, энкодер преобразует входной вектор в его скрытое представление преобразованием , затем декодер восстанавливает сигнал по его скрытому представлению преобразованием . Затем, по результату работы автокодировщика для входного вектора считается значение функции ошибки , функционал которой минимизируется по весам автокодировщика методом обратного распространения ошибки и оптимизации весов автокодировщика одним из вариаций градиентного спуска.

Поскольку и – это векторы в , то функция ошибки в случае автокодировщика – это одна из классических функций ошибок в задаче регрессии, например MSE (минимизация суммы квадратов ошибок) или MAE (минимизация суммы модулей ошибок).

Стоит отметить, что одной из отличительных особенностей автокодировщиков является то, что в случае большой выборки, автокодировщик будет очень долго обучаться, чтобы начать показывать хорошие результаты, потому что в таком случае автокодировщик будет выучивать распределение входных данных, которых оказалось много.

## **1.6. Алгоритмы решения задачи кластеризации**

Рассмотрим несколько классических алгоритмов решения задачи кластеризации, которые будут использованы далее, и результаты которых будут сравнены далее.

Первый алгоритм кластеризации – KMeans. Алгоритм KMeans – один из наиболее популярных методов кластеризации, ввиду его простоты и сходимости.

Исходя из названия, в алгоритме KMeans заранее задается число кластеров , после чего алгоритм стремится минимизировать сумму квадратов отклонений объектов от центра кластера этого объекта. В качестве центра тяжести кластера, как правило, берут центр масс объектов в кластере.

Математическая постановка алгоритма состоит в следующем: имеется набор наблюдений , каждое наблюдение – это -мерный вектор, тогда результат работы алгоритма разбивает наблюдений на подмножеств так, чтобы минимизировать внутрикластерную дисперсию (сумму квадратов отклонений). Задача алгоритма KMeans – найти такое разбиение , что:

Алгоритм KMeans является итерационным: на каждой итерации алгоритма пересчитывается центр масс для каждого кластера, после чего для каждого наблюдения находится ближайший центр масс и данное наблюдение помечается номером кластера, который имеет данный центр масс. Алгоритм завершается, когда на какой-либо итерации не произошло изменения суммарного внутрикластерного расстояния, однако программно прекращение работы алгоритма происходит, когда на какой-либо итерации изменение суммарного внутрикластерного расстояние оказалось меньше, чем на .

Стоит отметить, что существует несколько вариаций алгоритма в зависимости от того, как инициализируются центры кластеров по умолчанию. Есть несколько вариантов инициализации центра кластеров:

* Во-первых, инициализация центров кластеров случайными объектов из выборки, то есть, выбирается случайных объектов из выборки и эти объекты назначаются центрами кластеров;
* Во-вторых, инициализация центров кластеров происходит случайно по каждой компоненте всех векторов, другими словами, для каждого центра кластеров каждая компонента вектора инициализируется случайно из какого-либо распределения, например, из равномерного;
* В-третьих, инициализация центров кластеров происходит в соответствии с алгоритмом kmeans++, который заключается в нахождении «оптимальных» начальных значений центров кластеров, интуитивно смысл алгоритма состоит в том, чтобы распределить центры кластеров равномерно (в смысле плотности соседних объектов) по всему пространству признаков.

К преимуществам KMeans можно отнести гарантированную сходимость алгоритма, простоту использования и быстроту реализации, а к недостаткам – необходимость выбора количества кластеров , чувствительность к выбросам, зависимость результатов работы алгоритма от изначальной инициализации центров кластеров, а также тот факт, что не гарантируется нахождение глобального оптимума суммарного внутрикластерного расстояния, как правило, находится лишь один из локальных минимумов.

Второй алгоритм кластеризации – DBSCAN (англ. density-based spatial clustering of applications with noise). В современном сообществе машинного обучения, данный алгоритм многими считается лучшим алгоритмом кластеризации, поскольку в данном алгоритме не требуется заранее указывать количество кластеров, поскольку алгоритм хорошо умеет работать с выбросами, а также, так как данный алгоритм основан на плотности объектов в признаковом пространстве, то алгоритм группирует объекты, которые имеют высокую плотность и расположены тесно, а объекты, имеющие низкую плотность, алгоритм помечает как выбросы.

У алгоритма DBSCAN есть два гиперпараметра – это минимальное число точек в кластере и радиус эпсилон-окрестности . Работа алгоритма DBSCAN начинается с произвольной точки выборки, после чего находится число точек в -окрестности данной точки – в случае, если это число не меньше , то точка и все ее соседи в -окрестности начинают образовывать кластер, в противном случае, точка помечается выбросом, шумом, однако, заметим, что эта точка может быть позже найдена в -окрестности какой-либо точки из ее соседей и включена в кластер.

Другими словами, если найдена точка с высокой плотностью, то она и все ее соседи в -окрестности добавляются в один кластер. Данный процесс продолжается, пока не будет найден полный состав данного кластера, после чего выбирается новая случайная точка и процесс повторяется для нее и ее соседей, и так до тех пор, пока все точки выборки не будут помечены шумом либо конкретным номером кластера.

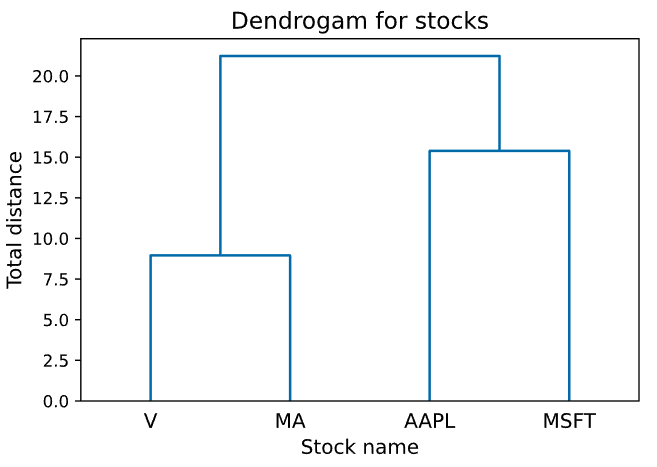
Стоит отметить, что DBSCAN может использоваться с любой функцией расстояния, а также DBSCAN может работать на основе предподсчитанной матрицы расстояний.

К преимуществам DBSCAN можно отнести то, что DBSCAN устойчив к выбросам, не требует априорного задания числа кластеров, имеет всего два интерпретируемых гиперпараметра, которые могут гибко настраиваться, а также, что DBSCAN находит плотные кластеры произвольной формы. К недостаткам же, как правило, относят лишь то, что для большого числа признаков и большого количества элементов в выборке DBSCAN алгоритм будет работать очень долго.

Третий алгоритм кластеризации – иерархическая кластеризация. Иерархическая кластеризация – такие алгоритмы кластеризации, которые направлены на создание иерархии, которую обычно визуализируют в виде дерева вложенных кластеров. Иерархическая кластеризация бывает агломеративная, когда новые кластеры создаются путем объединения более мелких кластеров, и дивизионная, когда новые кластеры создаются путем разбиения больших кластеров на более мелкие. Здесь и далее будут рассматриваться только алгоритмы агломеративной кластеризации, поскольку они более популярны, чем алгоритмы дивизионной кластеризации.

Для визуализации результатов иерархической кластеризации используется дендрограмма – такое дерево, построенное по матрице расстояний объектов выборки, в котором в листьях дерева находятся объекты выборки, а в узлах дерева находятся подмножества объектов выборки, то есть кластера, при этом объединение двух узлов создает новый узел, в котором находится объединение подмножеств объектов его узлов-потомков.

Пример дендрограммы для выборки из четырех объектов финансового рынка, четыре компании – Visa, Mastercard, Apple, Microsoft, представлен на рисунке ниже



В данном случае видно, что в дереве 4 листа – это названия компаний, и 3 узла, первый узел – объединение Visa и Mastercard в один кластер, второй узел – объединение Apple и Microsoft в один кластер, третий узел – объединение всей выборки в один кластер. Каждый узел находится на какой-либо высоте по оси ординат – это значит, что суммарное внутрикластерное расстояние объектов данного кластера равно отметке по высоте, на которой находится данный узел в дендрограмме.

Понятно, что для того, чтобы получить кластера, необходимо обрезать дерево по определенной высоте, тогда после обрезания дерева по определенной высоте, останутся поддеревья, и в корне каждого полученного поддерева (поддеревом может быть и дерево из одной вершины, из одного объекта) будет состав кластера.

Стоит отметить, что дендрогромма – это ни в коем случае не алгоритм кластеризации, это лишь визуализация интуиции, стоящей за алгоритмами агломеративной кластеризации.

Более формально, алгоритм агломеративной кластеризации состоит в следующем:

* Для каждой пары объектов выборки считаются парные расстояния по заданной метрике расстояний, которая, на самом деле, математически может быть абсолютно любой, и получается матрица расстояний
* Каждый объект данных рассматривается как отдельный кластер;
* Два кластера, которые являются ближайшими по какому-либо межкластерному расстоянию, объединяются в один кластер, после чего считается (либо обновляется) матрица расстояний между кластерами;
* Третий пункт повторяется, пока не останется один кластер

В случае, когда расстояние считается между объектами, все довольно понятно – это, как правило, какое-либо классическое расстояние, например – Eвклидово расстояние, расстояние Чебышева или манхэттенское расстояние.

Однако, не совсем очевидный момент – это то, как считается расстояние между кластерами, ведь кластер – совокупность объектов, и не совсем понятно, как можно считать расстояние между двумя совокупностями объектов. Эта проблема решается какой-либо агрегацией объектов в кластерах для подсчитывания расстояния между кластерами.

Существует несколько классических расстояний между кластерами:

* Метод одиночной связи – расстоянием между кластерами и является минимальное расстояние между всеми парами объектов из этих кластеров:
* Метод полной связи – расстоянием между кластерами и является максимальное расстояние между всеми парами объектов из этих кластеров:
* Метод средней связи – расстоянием между кластерами и является среднее расстояние между всеми парами объектов из этих кластеров:
* Метод центроидной связи – расстоянием между кластерами и является квадрат расстояния между центрами кластеров:
* Метод связи Уорда – расстоянием между кластерами и является минимизации суммы квадратов расстояний объектов внутри кластеров, такой подход, с точки зрения статистики, фактически, приводит к минимизации дисперсии каждого отдельного кластера, тогда итоговая формула выглядит следующим образом:

Преимуществами агломеративной иерархической кластеризации являются интуитивная интерпретируемость алгоритма, гибкая настраиваемость ввиду большого количества различных возможных расстояний между кластерами и объектами выборки, интерпретируемая визуализация результатов работы алгоритма.

К недостаткам агломеративной иерархической кластеризации же можно отнести вычислительную стоимость работы алгоритма, потому что для большого количества элементов выборки расчет матрицы попарных расстояний – это очень дорогая операция, чувствительность к выбросам и шуму, а также необходимость определения числа кластеров.

**1.7. Сбор и обработка данных финансовых активов индекса S&P 500**

В качестве исходных данных будет рассматриваться дневная динамика цены акций всех компаний индекса S&P 500 на апрель 2022 года. Рассматриваемый горизонт времени – 2 года, с мая 2020 года по май 2022 года.

Ежедневные данные о дневной цене открытия, закрытия, средней цене за день были загружены с сервиса Yahoo Finance при помощи библиотеки на Python yfinance.

Поскольку в индекс S&P 500 входит более пятисот компаний, то процесс парсинга данных, даже при помощи уже готового модуля для загрузки, совсем небыстрый и нестабильный. Для стабильного процесса сбора исходных данных на языке Python был написан собственный парсер ежедневных цен акций компаний индекса S&P 500, особенностью которого является поддержка логирования процесса парсинга, а также возможность задания количества компаний индекса S&P 500 для парсинга. Пример запуска парсера из консоли с поддержкой логирования представлен на рисунке ниже

Изображение выглядит как текст

Автоматически созданное описание

Также отметим, что вся работа по парсингу и обработке исходных данных, а также дальнейшая кластеризация на основе исходных данных, велась через ssh-соединение к виртуальной машине с операционной системой Linux, дистрибутив Ubuntu.

На парсинг динамики ежедневных цен на акции всех компаний индекса S&P 500 понадобилось около двух с половиной часов, а именно, 168 минут.

После того как данные были собраны, задача состоит в их непосредственной обработке – обработке пропущенных значений, аномальных значений и различных всевозможных проблем.

Во-первых, в ходе обработки данных, были убраны из рассмотрения такие компании, как Constellation Energy Corporatio, Organon & Co., Paramount Global, Warner Bros. Discovery, Inc., поскольку в данных финансовых активах оказались пропущенные значения.

Во-вторых, исходные данные были приведены из csv-файла к удобному датафрейму, в котором в каждой строке находится название компании, аббревиатура названия компании, сектор, страна, индустрия и дневные цены закрытия акций. Такая обработка позволяет в дальнейшем удобно нормализовать данные, удобно считать различные статистики и аггрегировать результаты кластеризации.

В-третьих, несмотря на то, что данные рассматриваются за двухлетний период, с мая 2020 года по май 2022 года, было решено сократить рассматриваемый промежуток времени до интервала с июня 2021 года по май 2022. Сделано было такое решение с целью уменьшения размерности признакового пространства данных для ускорения работы алгоритмов кластеризации путем ускорения расчетов расстояний между временными рядами. Такое решение позволило в относительных показателях увеличить скорость работы алгоритмов кластеризации путем увеличения скорости расчета расстояний между рядами в среднем в четыре раза, поскольку рассматриваемый период уменьшился приблизительно в четыре раза, а расстояние считается за линейное время, следовательно, расстояние стало считаться в четыре раза быстрее.

В-четвертых, исходные данные приведены к нормальной форме путем вычитания из дневных цен закрытия акций каждой компании ее среднедневной цены закрытия и деления на ее среднеквадратическое отклонение. Такая нормализация приводит данные к стандартному распределению с нулевым средним и единичной дисперсии, таким образом для всех компаний индекса S&P 500 гарантируется сравнимость значений дневных цен закрытия для расчета расстояний в алгоритмах кластеризации.

После того как данные очищены, обработаны и нормализованы, поскольку в алгоритмах кластеризации из модуля для Python scikit-learn нет DTW-расстояния, то необходимо было посчитать квадратную матрицу попарных DTW-расстояний между каждой парой объектов. Поскольку заполнение квадратной матрицы выполняется за квадратическое время, а всего в рассматриваемом наборе данных более 500 компаний, то матрица расстояний считалась довольно долго – около получаса.

## **1.8. Кластеризация финансовых активов индекса S&P 500**

После сбора и обработки данных можно приступать к кластеризации финансовых активов из индекса S&P 500.

Первый использованный алгоритм кластеризации – иерархическая кластеризация. В качестве матрицы расстояний для построения дендрограммы и самого алгоритма кластеризации использовалась посчитанная ранее матрица попарных DTW-расстояний. Метрикой расстояния между кластерами было выбрано расстояние, рассчитываемое по метод связи Уорда.

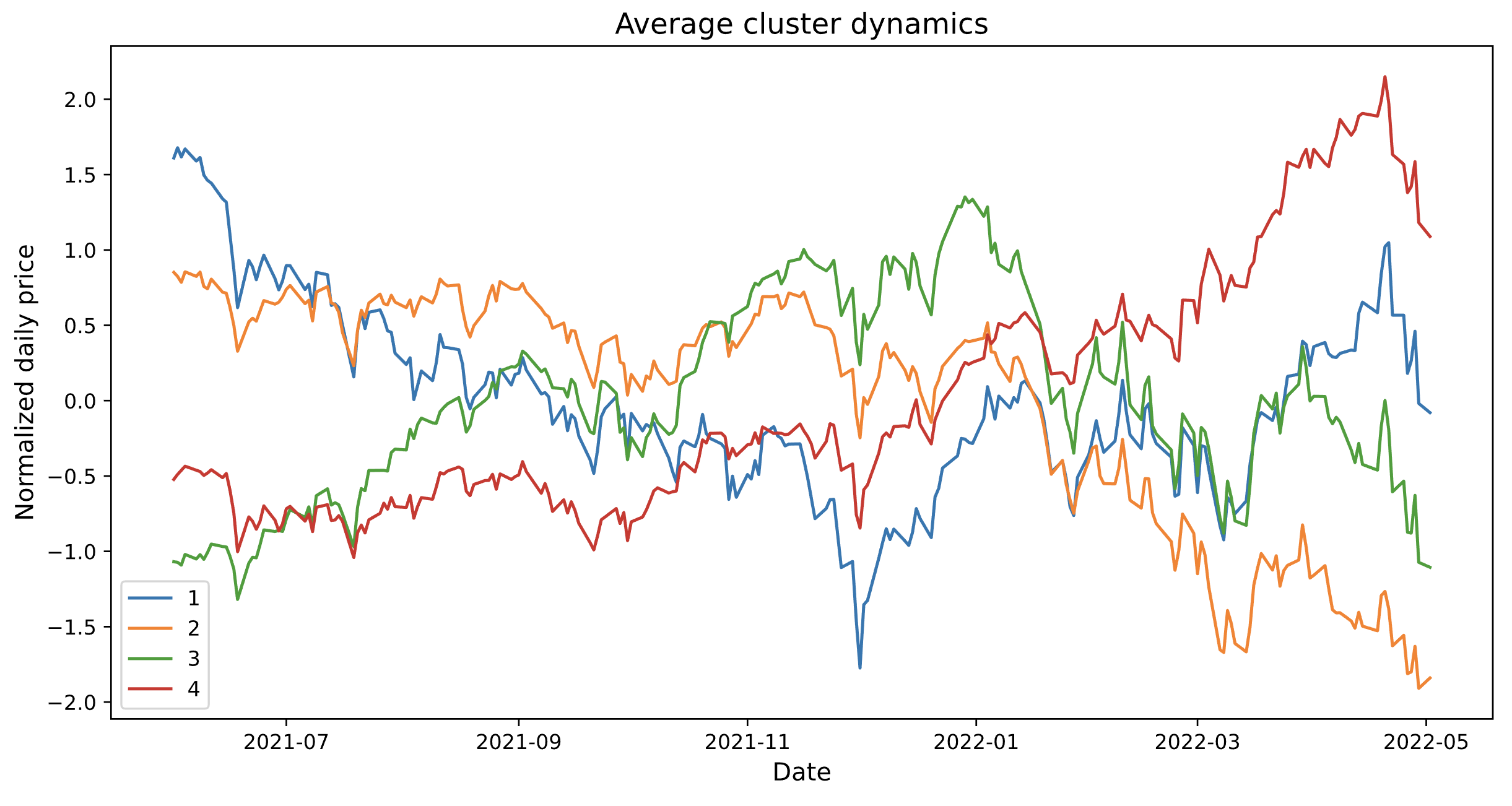
Поскольку объектов кластеризации много и задача специфическая, то в иерархической кластеризации порог по расстоянию сечения дендрограммы подбирался из интерпретируемости кластеров и однородности кластеров.

В результате, после выбора порога по расстоянию, было получено 4 кластера, сводная таблица статистик и размеров, по которым приведена на рисунке ниже. Дендрограмма представлена в приложении **ТУТ ПРИЛОЖЕНИЕ ДОПИЛИТЬ**

Изображение выглядит как стол

Автоматически созданное описание

Графики средней динамики нормализованных дневных цен закрытия каждого кластера представлен на рисунке ниже.



Полученные кластера можно интерпретировать следующим образом:

1. Первый кластер – компании индекса S&P 500, акции которых в середине рассматриваемого периода довольно сильно упали в цене, однако к концу периода практически вернулись к ценам начала периода, однако их доходность за рассматриваемый период, в любом случае, отрицательная и составляет в среднем -10%.
2. Второй кластер – компании индекса S&P 500, акции которых в течение всего рассматриваемого периода падали, акции этих компаний имеют за рассматриваемый период нисходящий тренд и в среднем упали на 26%.
3. Третий кластер – компании индекса S&P 500, акции которых к середине рассматриваемого периода довольно сильно выросли в цене, однако к концу периода практически вернулись к ценам начала периода, их доходность к концу периода практически нулевая.
4. Четвертый кластер – компании индекса S&P 500, акции которых в течение всего рассматриваемого периода росли, акции этих компаний имеют восходящий тренд и в среднем выросли на 20%, однако, стоит отметить, что показатели среднеквадратического отклонения доходности этого кластера значительно выше чем у всех остальных кластеров – это значит что разброс доходности акций этих компаний относительно средней доходности в кластере значительно выше, чем у других кластеров.

Второй использованный алгоритм кластеризации – DBSCAN. В качестве метрики расстояния у алгоритма из пакета scikit-learn было выбрана опция «precomputed», что значит, что алгоритм будет выполнен на посчитанной ранее матрицы DTW-расстояний.

В случае DBSCAN’а, в отличие от иерархической кластеризации, несмотря на выбранные параметры , алгоритм, по причине специфики работы алгоритма DBSCAN, выдавал неравномерные по численности и составу кластера, однако, как и в случае алгоритма иерархической кластеризации, параметры подбирались исходя из интерпретируемости кластеров.

Итоговые параметры алгоритма , по которым, в результате работы алгоритма, было получено 10 кластеров, сводная таблица статистик и размеров, по которым приведена на рисунке ниже.

Изображение выглядит как стол

Автоматически созданное описание

Из таблицы видно, что 1–10 кластера состоят всего из двух компаний, это объясняется тем, что акции данных компаний «аномально» похожи друг на друга с точки зрения динамики их дневной цены закрытия. А алгоритм DBSCAN, в свою очередь, работает так, что объединяет в кластеры именно «аномально» похожие друг на друга объекты, которые имеют низкую плотность.

В качестве примера «аномально» похожих друг на друга компаний с точки зрения динамики их дневной стоимости закрытия акций приведем кластер под номером три, состоящий из двух компаний – Carnival Corporation & plc и Norwegian Cruise Line Holdings, динамика нормализованных днедневных цен закрытия которых представлена на рисунке ниже.



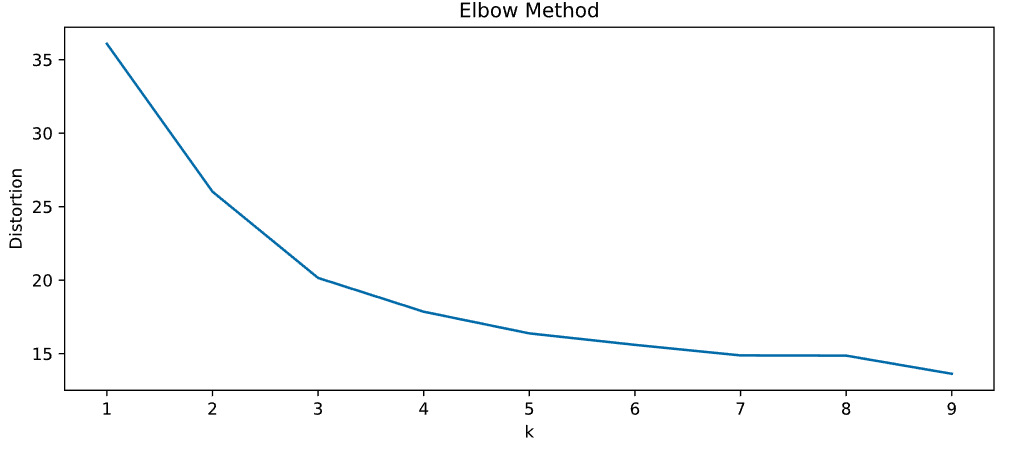
Из графика видно, что динамика дневных цен закрытия идентичная с точностью до случайного шума, однако, если посмотреть на род деятельности этих компаний – это оказывается не удивительно. Carnival Corporation & plc и Norwegian Cruise Line Holdings – это круизные операторы, соответственно любые новости, связанные с пандемией коронавируса оказывают на них абсолютно одинаковое влияние. В данном же случае, за рассматриваемый промежуток времени, виден нисходящий тренд, связанный с негативными новостями пандемии коронавируса в середине-конце 2021 года и с рекордной инфляцией в Америке в начале-середине 2022 года, которая оказывает негативное влияние на все компании индекса S&P 500.

Интерпретировать кластера, полученные алгоритмом DBSCAN, можно следующим образом: -1 и 0 кластера – обычные компании индекса S&P 500, 2–10 кластера – компании индекса. S&P 500, для каждой из которых есть компания индекса S&P 500, которая имеет точно такую же динамику дневной цены закрытия. С точки зрения портфельной теории, в данном случае, брать сразу две компании из 2–10 кластеров нельзя, потому что это контр диверсифицирует инвестиционный портфель, в случае падения цены на акции одной из двух компании, также упадет цены на акции другой компании кластера.

Следующий использованный алгоритм кластеризации – Kmeans. Особенность практической работы с ним в случае решения данной задачи состоит в том, что в пакете scikit-learn не реализована метрика DTW-расстояния, и, в силу специфики самого алгоритма KMeans (а именно из-за того, что в алгоритме считаются расстояния до центров масс кластера, а центры масс на каждой итерации меняются), нет возможности предварительно посчитать матрицу попарных DTW-расстояний, как это было в случае DBSCAN и иерархической кластеризации.

Поэтому, вместо пакета scikit-learn для Python было решено воспользоваться пакетом tslearn для Python, который предоставляет инструменты машинного обучения для анализа временных рядов, в частности, в данном пакете реализован алгоритм KMeans с доступным DTW-расстоянием.

В случае KMeans, в отличие от иерархической кластеризации и DBSCAN, оптимальное число кластеров выбиралось по графику каменистой осыпи, который представлен на рисунке ниже

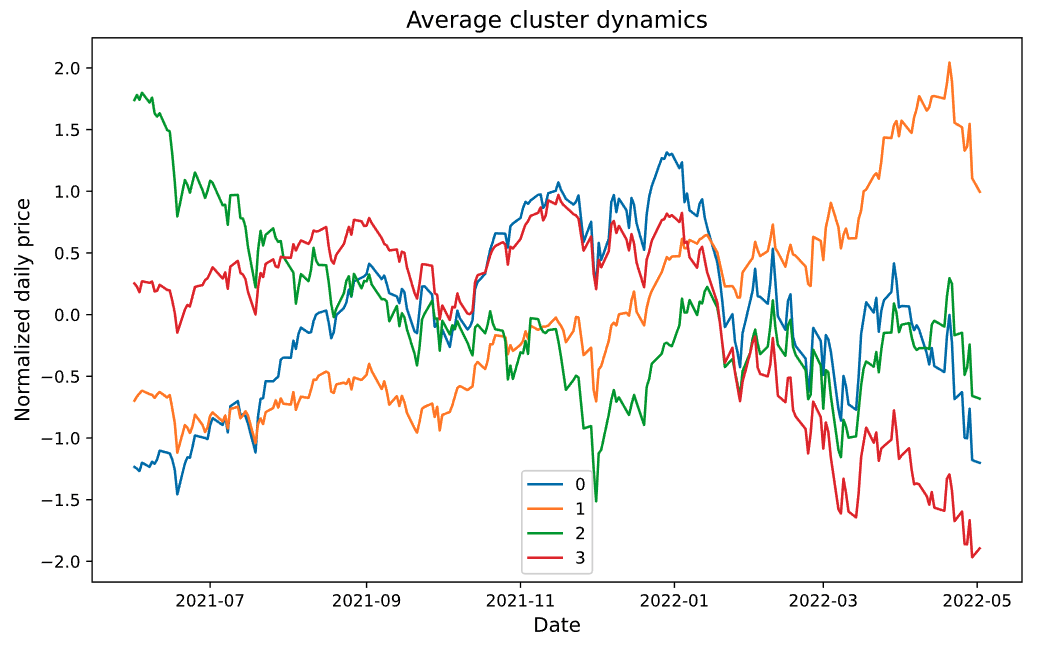


На графике каменистой осыпи по оси абсцисс изображается число кластеров, по оси ординат изображается среднее из суммы квадратов расстояний до центра кластера, которое при увеличении количества кластеров будет монотонно убывать. Обычно берется такое число кластеров, при котором убывание значений среднего квадратов расстояний слева направо максимально замедляется. В данном случае, таким числом было взято четыре кластера, сводная таблица статистик и размеров, по которым приведена на рисунке ниже.

Изображение выглядит как стол

Автоматически созданное описание

График средней динамики нормализованных дневных цен закрытия каждого кластера представлен на рисунке ниже.



Интерпретировать полученные алгоритмом KMeans кластеры можно абсолютно таким же образом, каким интерпретировались кластеры, полученные иерархической кластеризации, с точностью до номера кластера:

1. Нулевой кластер – компании индекса S&P 500, которые выросли в цене в середине рассматриваемого периода, однако в конце периода вернулись к цене начала периода.
2. Первый кластер – компании индекса S&P 500, которые имеют восходящий тренд и росли в цене в течение всего рассматриваемого периода.
3. Второй кластер – компании индекса S&P 500, которые упали в цене в середине рассматриваемого периода, однако в конце периода вернулись к цене начала периода.
4. Третий кластер – компании индекса S&P 500, которые имеют нисходящий тренд и падали в цене в течение всего рассматриваемого периода.

# **2. ПРОГНОЗИРОВАНИЕ ДОХОДНОСТИ ФИНАНОСОВЫХ АКТИВОВ**

## **2.1. Постановка задачи прогнозирования доходности финансовых активов как временных рядов**

Временной ряд – это последовательность значений , которые описывают какой-либо процесс, протекающий во времени, измеренные в последовательные моменты времени , которые, как правило, являются равными промежутками времени .

Временной ряд, содержащий во времени значения одной переменной, называется одномерным, а две и более переменных называется многомерным. В нашем случае, когда речь идет о динамике цен финансовых активов либо о доходности финансовых активов, последовательность цен или доходностей финансового актива во времени является временным рядом, причем одномерным.

Задача прогнозирования временного ряда состоит в нахождении вектора значений , которые будут описывать динамику ряда на значений вперед, основываясь на предыдущих значениях ряда .

Существует несколько различных подходов к задаче прогнозирования временных рядов: статистические модели прогнозирования, стандартные регрессионные модели машинного обучения, регрессионные модели глубинного обучения, основанные на рекуррентных нейронных сетях. Особенности и различия этих подходов будут рассмотрены далее.

Очевидно, что каждый временной ряд – это сложный процесс, на динамику которого оказывает влияние множество различных факторов. Однако, принято выделять четыре компоненты, на которые можно разбить динамику временного ряда:

1. Сезонность – это периодические регулярные предсказуемые колебания, которые соответствуют сезонным изменениям, например, – месяц, квартал или полугодие. В качестве примера временного ряда с ярко выраженной сезонностью можно привести продажи пуховиков и шуб, зимой продажи шуб и пуховиков высокие, а летом низкие.
2. Тренд – основная тенденция изменения временного ряда. Тренд может быть восходящим и нисходящим, а также линейным и нелинейным. Пример временного ряда с ярко выраженным трендом можно привести численность населения земного шара, которая начала экспоненциально расти в двадцатом веке.
3. Случайный остаток – разница между значением ряда, и суммы трендовой и сезонной компоненты. Формально, данная компонента не прогнозируема и возникает из-за случайных краткосрочных колебаний в самом процессе временного ряда.

Также, принято выделять стационарные и нестационарные временные ряды. Стационарный временной ряд – такой ряд, у которого постоянны среднее значение, дисперсия и ковариация соседних элементов ряда в последовательности, а нестационарный временной ряд – такой временной ряд, который не является стационарным. Математически стационарный временной ряд описывается следующим образом:

Принято считать, что проще работать со стационарными временными рядами – в них отсутствует трендовая составляющая, а также у многих статистических моделей прогнозирования стационарность ряда является одной из предпосылок.

Существует ряд статистических тестов, которые проверяют стационарность метод при помощи инструментов математической статистики. Наиболее популярный такой статистический тест проверки стационарности ряда – тест Дики-Фуллера, который проверяет нулевую гипотезу о нестационарности временного ряда.

Существует несколько преобразований, которые позволяют преобразовать нестационарный временной ряд в стационарный временной ряд:

1. Взятие разностей ряда -го порядка
2. Аппроксимирование ряда линейной моделью
3. Преобразование Бокса-Кокса с параметром (который нужно подбирать, например, методом максимального правдоподобия), которое выполняется для временных рядов с положительными значениями – в случае, если у временного ряда есть отрицательные значения, то можно сдвинуть все значения на :

Финансовые временные ряды являются совсем необычными временными рядами – динамика финансовых временных рядов во многом определяется микроструктурой финансового рынка, текущими новостями, начиная от мировых новостей, заканчивая публикацией финансовых результатов компанией. В финансовых временных рядах, как правило, всегда присутствуют тренд и цикличность, а сезонность отсутствует. Также, одной из особенностей финансовых временных рядов является высокая рыночная волатильность.

Существует несколько объектов прогнозирования – это цена финансового актива и доходность финансового актива. Прогнозирование доходности особенно интересно для инвесторов, поскольку, инвестору не важна цена актива через некоторый промежуток времени, ему важна именно доходность финансового актива через некоторый промежуток времени. Прогнозирование цены финансового актива компании более интересно для самой компании, поскольку это позволяет составлять финансовые планы и бюджеты на рассматриваемый период, исходя из прогноза динамики цены акций.

Именно поэтому в данной выпускной квалификационной работе будет рассмотрен именно подход прогнозирования доходности финансового актива.

## **2.2. Постановка задачи классификации динамики доходности финансовых активов**

Альтернативной подходу прогнозирования является подход к анализу динамики финансовых активов с точки зрения динамики в следующую единицу времени.

Такая задачи классификации предполагает классификацию знака доходности актива в следующую единицу времени. Биржевым рыночным языком, это классификация рынка на медвежий и бычий. Медвежий рынок – период времени на финансовом рынке, когда цена финансового актива имеет нисходящий тренд, а бычий рынок – период времени на финансовом рынке, когда цена финансового актива имеет восходящий тренд. Математически это описывается следующим образом:

Где – класс рынка в единицу времени , – доходность финансового актива в единицу времени .

Прогнозы, полученные по данной методологии, являются не предсказанием изменения доходности актива или доходности актива в следующую единицу времени, а степень их отклонения от состояния равновесия, установившуюся на данный момент.

Данную задачу можно решать любыми моделями классификации классического машинного и глубинного обучения, которые с определенной точностью будут предсказывать знак доходности актива в будущую единицу времени.

Для достижения приемлемых результатов классификации в таком случае необходим очень тщательный отбор признаков, поиск дополнительных параметров и статистических данных.

По результатам ответов алгоритма и истинным меткам классов можно построить матрицу ошибок:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | True Positive (TP) | False Positive (FP) |
|  | False Negative (FN) | True Negative (TV) |

Показатели качества прогнозирования знака доходности можно оценивать по классическим метрикам бинарной классификации:

1. Accuracy – точность алгоритма, доля правильных ответов алгоритма:
2. Precision – точность, доля объектов, классифицированных алгоритмом положительными и при этом являющимися положительными:
3. Recall – полнота, доля объектов положительно классифицированных алгоритмом объектов из всех положительных классов:
4. F1 – среднее гармоническое полноты и точности, метрика, учитывающая в равной мере и полноту и точность:
5. RocAuc – вероятность того, что случайно выбранная пара истинной и предсказанной меток класса будет правильно отранжирована.

Качество прогнозов алгоритма по описанным выше метрикам бинарной классификации должно производиться на тестовой выборке. Тестовая выборка – отложенная часть исходных данных, которая не участвует в обучении алгоритма и по которой оцениваются метрики качества классификации. В нашем случае, поскольку мы имеем дело с временными рядами, тестовая выборка обязательно должна быть по времени позже обучающей выборки.

## **2.3. Алгоритмы прогнозировании временных рядов и классификации**

Первая группа алгоритмов прогнозирования – статистические модели временных рядов. К ним можно отнести все модели, связанные с авторегрессией и скользящем средним – это популярные на данный момент AR, MA, ARMA, ARIMA, SARIMA.

Простейшие статистические модели временных рядов, которые лежат в основе более сложных статистических моделей временных рядов – это авторегрессионная модель (англ. auto regreesive, AR) и модель скользящего среднего (англ. moving average, MA).

Авторегрессионная модель предполагает, что значение временного ряда в момент времени зависит от предыдущих значений этого же ряда, математически эта модель описывается следующим образом:

Модель скользящего среднего предполагает, что значение временного ряда в момент времени зависит от значений ошибок прошлых периодов, математически эта модель описывается следующим образом:

Более сложной моделью является композиция моделей авторегрессии и скользящего среднего, которая называется . Параметры и показывают количество лагов в модели авторегрессии и модели скользящего среднего соответственно, а – показатель, соответствующий значению единицы времени. Математически модель описывается следующим образом:

Стоит отметить, что важной особенностью моделей AR, MA, ARMA является предпосылка о стационарности временного ряда, к которому будут применяться эти модели. Предположение о стационарности – это довольно серьезное предположение, которое далеко не всегда выполняется на практике, один из способов решения данной проблемы – взятие разностей временного ряда. На таком подходе базируется модель , которая является модификацией модели и не предполагает стационарность временного ряда. Параметр показывает степень взятия разностей временного ряда: .

Стоит отметить, что все модели AR, MA, ARMA являются частными случаями модели ARIMA с разными параметрами . Также, важно, что все модели AR, MA, ARMA, ARIMA предполагали, что слагаемое ошибки временного ряда имеет одинаковую дисперсию. Оценкой волатильности можно считать дисперсию временного ряда – тогда, получается, что при сильной волатильности наблюдается непостоянство дисперсии, гетероскедастичность. Отсюда возникает проблема, связанная с прогнозированием финансовых активов – на финансовых рынках особенно сильная волатильность, что проявляется в случайных отклонениях от среднего курса.

Решает данную проблему модель GARCH (англ. Generalized AutoRegressive Conditional Heteroscedasticity), которая моделирует динамику дисперсии (волатильности) разложением дисперсии в две модели, по аналогии с ARMA, – модель авторегрессии и модель скользящего среднего. GARCH модель обладает двумя параметрами , которые, по аналогии с ARMA, показывают количество лагов в модели авторегрессии и модели скользящего, тогда модель можно математически описать следующим образом:

Математически серьезной является композиция моделей ARIMA и GARCH – такая модель предполагает, что временной ряд можно разложить на две составляющие – основной тренд, который прогнозируется моделью ARIMA, и волатильность, которая прогнозируется моделью GARCH.

Вторая группа алгоритмов прогнозирования временных рядов – это классические модели машинного обучения, например, линейная регрессия, метод опорных векторов или ансамблевые алгоритмы над решающими деревьями.

Ключевой особенностью этих алгоритмов в решении задачи прогнозирования временных рядов является необходимость тщательнейшего отбора и генерации всевозможных признаков, потому что, в противном случае, данные алгоритмы потерпят крах в решении задачи прогнозирования временных рядов, поскольку они не будут учитывать специфику, особенности и скрытую информацию, которая содержится во временных рядах.

Существуют классические трюки в генерации и отборе признаков в случае решения задачи прогнозирования временных рядов при помощи классических методов машинного обучения. Генерировать признаки можно многими способами:

1. Использовать в качестве одного из признаков время – отображать каждую единицу времени в числовое значение так, чтобы на отображенных значениях соблюдалась монотонность по аргументу времени.
2. Использовать предыдущие значения временного ряда – в качестве признаков можно брать предыдущих значений временного ряда, таким образом будет учитываться «история» временного ряда.
3. Найти количественные признаки, которые могут оказывать влияние на значения временного ряда.
4. Математически преобразовать количественные признаки (в том числе, признак времени) по отдельности или по парам, – например, создать признаки попарных произведений по всем парам, квадраты, корни квадратные из всех признаков по отдельности.
5. Если есть категориальные признаки, то в группировке по категориальным признакам считать агрегации, – например, среднее, моду, медиану, сумму

Отбирать признаки тоже можно большим количеством способов:

1. В случае, если модель машинного обучения является линейной, то регуляризацией можно отбирать факторы – если коэффициент при факторе в L1 или L2 регуляризации имеет околонулевой вес, то, скорее всего, данный фактор не имеет значительного влияния.
2. В случае, если модель машинного обучения является ансамблевым методом над решающими деревьями, то для каждого признака можно посчитать его значимость (англ. feature importance) – важные признаки будут иметь высокие показатели важности, а неважные – наоборот.
3. Также, в случае использования ансамблевых алгоритмов, можно воспользоваться современным методом отбора признаков – метод SHAP (англ. shapley additive explanation) – данных подход использует предспосылки теории игр для оценивания вклада каждого признака в финальное предсказание.

Третий подход к прогнозированию временных рядов – использование современных методов глубинного обучения, основанных на рекуррентных нейронных сетях.

Рекуррентная нейронная сеть – такая нейронная сеть, в которой связи между элементами образуют направленную последовательность. Более формально, это сети, в которой каждое следующее значение таргета зависит от его предыдущих значений.

Из определения понятно, что рекуррентные нейронные сети отлично подходят для обработки последовательностей – будь то видеофрагмент, состоящих из отдельных кадров, или аудиофрагмент, состоящий из отдельных звуков, или временной ряд, состоящий из последовательности значений во времени.

Принцип работы рекуррентного слоя нейронной сети состоит в том, что на очередном шаге на вход сети поступает вектор , по которому вычисляется скрытое (латентное) представление слоя по следующей формуле:

Где – какая-либо функция активации (например, ReLU), – это веса, которые будут оптимизироваться в данном слое нейронной сети каким-либо градиентным методом, – скрытое представление на предыдущем шаге А выходом рекуррентного слоя является , где – веса, которые будут оптимизироваться каким-либо градиентным методом.

Скрытое представление рекуррентного слоя можно интерпретировать как память рекуррентного слоя – в нем содержится информация о том, что «видела» нейронная сеть на предыдущих шагах обучения.

Проблемой классической рекуррентной нейронной сети является то, что информация, которую «видела» рекуррентная нейронная сеть, не хранится достаточно долго в скрытом состоянии . Данную проблему решает вариация архитектуры рекуррентных нейронных сетей, которая называется нейронная сеть с долгой краткосрочной памятью (англ. long short-term memory, LSTM).

В LSTM, в отличие от обычной рекуррентной нейронной сети, сеть посредством совмещения в одном слое сигмоидного слоя активации и слоя поэлементного умножения – такой слой возвращает значения в диапазоне от нуля до единицы, которые обозначают, какую долю информации следует сохранить для дальнейшего обучения.

**2.4. Регрессионный подход к прогнозированию динамики индекса S&P 500**

фыв

**2.5. Классификационный подход к прогнозированию динамики индекса S&P 500**

вфыа

# **3. ФОРМИРОВАНИЕ ОПТИМАЛЬНОЙ СТРУКТУРЫ ИНВЕСТИЦИОННОГО ПОРТФЕЛЯ**

Тут будет какой то текст

# **ЗАКЛЮЧЕНИЕ**

Тут будет какой то текст

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Тут будет какой то текст