

Ответы на экзаменационные вопросы по курсу «Многомерный анализ данных»

Общие вопросы

1. Содержание и цели дисциплины «Многомерный анализ данных» и возможности Loginom по их реализации.

- **Содержание дисциплины:** Курс охватывает основы науки о данных и методы интеллектуального анализа данных. Рассматриваются этапы работы с данными – от сбора и подготовки до моделирования и визуализации, включая статистические основы, методы машинного обучения и бизнес-аналитики. Студенты изучают техники обработки многомерных данных: очистка, преобразование, статистический анализ, регрессионный, корреляционный, факторный, кластерный анализ, нейронные сети, ассоциативные правила и др.

- **Цели дисциплины:** Дать представление о современных системах анализа данных и научить извлекать из данных полезные знания для принятия решений. Курс направлен на знакомство с классовыми BI-системами – интеллектуальными информационными системами для бизнес-аналитики ¹. Студенты должны научиться применять методы Data Mining для решения реальных задач и понимать полный цикл анализа данных – от постановки задачи до получения результатов.

- **Возможности Loginom:** Платформа Loginom выбрана как практический инструмент, потому что она поддерживает **весь процесс анализа данных** – от интеграции и подготовки данных до их моделирования, прогнозирования и визуализации ². Loginom – это *low-code* аналитическая платформа, позволяющая без программирования выполнять сложные расчёты, консолидацию различных источников данных, очистку и трансформацию данных, построение моделей (включая статистические и ML-алгоритмы) и представление результатов в наглядной форме. Таким образом, все методы, изучаемые в курсе, могут быть реализованы на практике с помощью Loginom (например, кластеризация, нейросети, регрессия, ассоциативный анализ и пр.), что связывает теорию с реальными инструментами.

2. Понятия больших данных (Big Data) и датасетов. Причины их появления и реализация в интернет-пространстве.

- **«Большие данные» (Big Data):** термин, обозначающий экстремально большие, сложные и быстро изменяющиеся наборы данных, обработка которых затруднена традиционными методами. Big Data характеризуется объёмом, разнообразием источников и скоростью поступления. В современном мире огромное количество данных генерируется повсеместно: веб-сайты фиксируют каждое действие пользователей, смартфоны постоянно собирают данные о местоположении, умные устройства (IoT) регистрируют множество параметров повседневной жизни ³. Эти массивы информации и называются большими данными. Понятие появилось как ответ на **взрыв данных** в цифровую эпоху – объёмы информации выросли настолько, что потребовались новые технологии хранения и обработки (распределённые файловые системы, кластерные вычисления, облачные хранилища и т.д.). Кроме того, большие данные подразумевают извлечение знаний из неструктурированных источников (социальные сети, логи поиска, датчики и пр.), что тоже стимулировало развитие нового класса методов.

- **«Датасет» (Dataset):** это просто набор данных, обычно структурированный (таблица, коллекция записей), используемый для анализа или обучения моделей. Датасетами могут быть как небольшие таблицы для статистического анализа, так и фрагменты больших данных. Появление этого термина связано с ростом культуры открытых данных и машинного обучения – появилось множество общедоступных датасетов для исследований и соревнований (например, на платформах типа Kaggle).

- **Причины возникновения понятий:** термин *Big Data* возник вследствие упомянутого лавинообразного роста объёма данных в бизнесе, науке и интернете. Традиционные базы данных не справлялись с **масштабами и скоростью** генерации информации, потребовалась новая парадигма обработки. Понятие *dataset* стало популярным с развитием *Data Science* – для экспериментов и обучения моделей нужны структурированные наборы данных, и общество данных стало активно делиться такими наборами.

- **Реализация в интернет-пространстве:** эра больших данных реализована через распределённые системы хранения и вычислений (например, экосистема Hadoop, облачные сервисы типа AWS, Google BigQuery и т.п.), которые позволяют собирать и анализировать данные пользователей в интернете. Практически каждый крупный веб-сервис – социальные сети, поисковики, интернет-магазины – оперирует big data, собирая сведения о поведении миллионов пользователей (клики, лайки, покупки и т.д.). Эти данные хранятся в распределённых хранилищах и анализируются для извлечения знаний (таргетинг рекламы, рекомендации контента, аналитика трендов). Кроме того, появились **онлайн-репозитории датасетов** – порталы открытых данных (например, data.gov, открытые данные Всемирного банка, города и др.), где можно свободно получить большие наборы данных. В целом интернет-пространство не только генерирует Big Data, но и предоставляет инструменты для их агрегирования и обмена. Например, понятие data set реализуется через API и каталоги данных, позволяющие скачивать целевые наборы данных для анализа.

3. Отечественные и иностранные системы визуализации и анализа данных. Сравнение их особенностей.

- **Примеры отечественных систем:** Среди российских (отечественных) BI- и аналитических систем можно отметить платформу «Прогноз» (Prognost Platform) и интеллектуальную аналитическую платформу *Loginom* (ранее известна как Deductor) ⁴. Компания «Прогноз» разработала собственный комплекс для бизнес-аналитики, активно применяемый в госуправлении и крупных предприятиях РФ. *Loginom*, разработанный фирмой «Аналитические технологии» (Loginom Company), представляет современную российскую платформу анализа данных. Отечественные системы обычно имеют интерфейс на русском языке, учитывают локальные нормативы и специфику бизнеса, могут тесно интегрироваться с 1С и другими распространёнными в России ИТ-системами. Также плюсом может быть более низкая стоимость и локальная поддержка.

- **Примеры иностранных систем:** К крупнейшим мировым разработчикам средств анализа данных относятся **SAP, Oracle, IBM, SAS, Microsoft, Tableau, Qlik** и др. Например, *SAP BusinessObjects* – мощная BI-платформа от SAP AG; *Oracle BI Suite* – интегрированный набор бизнес-аналитики от Oracle; *IBM Smart Analytics* – решение IBM с OLAP, текстовой аналитикой и хранилищами данных; *SAS* – известный статистический и Data Mining пакет; *Microsoft* предлагает инструменты Power BI, Excel, SQL Server Analysis Services для аналитики ⁵ ⁶. Эти системы отличаются высокой масштабируемостью, богатыми возможностями для интеграции и глобальной поддержкой.

- **Сравнение особенностей:** И зарубежные, и отечественные системы стремятся обеспечить полный цикл бизнес-аналитики (ETL, хранение, анализ, визуализация). Зарубежные решения часто ориентированы на крупные международные корпорации: они интегрируются с различными корпоративными источниками данных, обладают модульностью и поддерживают множество языков.

Например, продукты SAP и Oracle могут работать с различными БД, ERP-системами и имеют преднастроенные модули под отрасли ⁷ ⁸. Отечественные системы обычно лучше приспособлены к российским реалиям – имеют встроенную поддержку местных форматов данных, бухгалтерских стандартов, русскоязычных справочников. В плане функциональности: **и те, и другие** предлагают схожий набор инструментов (OLAP-кубы, отчётность, дашборды, Data Mining алгоритмы). Однако зарубежные лидеры (SAP, Oracle, Microsoft) славятся надёжностью и масштабируемостью, в то время как российские продукты привлекают более низкой ценой, отсутствием языкового барьера и возможностью гибкой доработки под заказчика. Например, Loginom полностью веб-ориентирован и не требует установки на рабочие места, что упрощает развёртывание ⁹. В целом, выбор зависит от масштабов и требований организации: иностранные системы – стандарт де-факто для международного бизнеса, а отечественные – хорошая альтернатива для компаний, предпочитающих локальные решения с поддержкой на родном языке.

4. Задачи Data Mining и их классификация.

В интеллектуальном анализе данных (Data Mining) выделяют несколько основных типов задач, для решения которых строятся модели ¹⁰ ¹¹:

- **Прогнозирование (prediction):** предсказание будущих значений на основе имеющихся данных. Например, прогноз продаж на следующий месяц, предсказание нагрузки на сервер или времени его простоя ¹². По сути, это задачи регрессии или классификации во времени, где модель выдаёт ожидаемый результат.
- **Анализ рисков и вероятностей:** оценка вероятностных сценариев, выявление риска наступления определённых событий. Примеры: определение наиболее подходящих клиентов для целевой рассылки, оценка кредитного риска клиента, прогноз вероятности отказа оборудования, назначение вероятностей диагноза в медицине ¹³. Такие задачи часто решаются методами классификации или вероятностного моделирования.
- **Рекомендательные системы (ассоциации/рекомендации):** определение, какие продукты или действия связаны друг с другом, чтобы выдавать рекомендации. Классический пример – поиск продуктов, которые часто покупаются вместе, и формирование рекомендаций («с этим товаром часто берут...») ¹⁴. Сюда же относятся системы рекомендаций контента (фильмов, музыки) на основе предпочтений пользователей.
- **Поиск последовательностей (sequence discovery):** выявление закономерных последовательностей событий во времени. Например, анализ поведения покупателей: в какой последовательности они просматривают товары и совершают покупки, выявление типичных *паттернов* действий пользователя на сайте ¹⁵. Это важно для прогнозирования поведения (маркетинг, отток клиентов) и для последовательного паттерн-майнинга (например, в финансовых временных рядах).
- **Группировка (clustering):** разделение объектов или событий на однородные группы (кластеры) для исследования их свойств ¹¹. Цель – обнаружить скрытые сегменты в данных. Например, сегментация клиентов по схожести покупательского поведения, кластеризация товаров по признакам. Группировка позволяет понять структуру данных без заранее заданных классов.

Эти задачи охватывают основные направления Data Mining. В различных источниках могут добавляться и другие категории, например выявление аномалий (анализ отклонений) и др., но часто они смыкаются с перечисленными. В целом классификация задач Data Mining делит их на предсказательные (прогнозирование, классификация, риск-оценка) и описательные (кластеризация, ассоциации, последовательности) модели.

5. Основные понятия статистики и их использование при анализе данных.

Статистика предоставляет базовый понятийный аппарат для анализа данных ¹⁶ ¹⁷. К ключевым понятиям относятся:

- **Статистическое наблюдение:** процесс планомерного сбора исходных данных об исследуемом объекте или явлении ¹⁸. Например, проведение опроса населения, снятие показаний датчиков – это статистическое наблюдение. Результаты наблюдения служат основой для дальнейшего анализа.
- **Статистическая совокупность:** множество единиц (объектов) исследования, объединённых определёнными признаками ¹⁹. Совокупность может быть генеральной (все объекты, например всё население страны) или выборочной (подмножество объектов). Анализ данных обычно стремится делать выводы о генеральной совокупности на основе выборки.
- **Признак (переменная) и его разновидности:** признак – измеряемая характеристика объекта, по которой ведётся сбор данных ²⁰. Признаки бывают количественными (например, рост, доход) и качественными (пол, цвет). В статистике важно различать типы признаков, так как методы анализа для них разные. В многомерном анализе данные представляют собой совокупность признаков для каждой единицы.
- **Вариация:** изменчивость значений признака у разных единиц. Практически всегда в данных есть разброс – единицы отличаются по показателям. Статистика вводит меры вариации (размах, дисперсия, среднеквадратичное отклонение), чтобы количественно оценивать разброс. Анализ вариации помогает понять надёжность средних величин, выявить однородность/неоднородность групп и т.д.
- **Статистические показатели:** обобщающие характеристики совокупности. Например, *средние величины* (среднее арифметическое, медиана, мода) – отражают центр распределения; *меры разброса* (дисперсия, стандартное отклонение) – отражают степень вариативности; *коэффициенты корреляции* – измеряют связь между признаками, и т.д. Эти показатели позволяют сжато описать большие объёмы данных и сделать выводы. В анализе данных часто сначала рассчитывают такие статистики, чтобы получить первичное понимание (разведочный анализ данных – Descriptive statistics).
- **Статистическая закономерность:** объективная повторяющаяся тенденция в данных, выявленная на основе массовых наблюдений. Статистика учит, что при достаточно больших данных проявляются устойчивые закономерности, несмотря на случайность индивидуальных вариаций. Выявление статистических закономерностей (например, зависимость дохода от образования) – цель анализа, позволяющая делать прогнозы и управленческие выводы.
- **Фактор и факторный анализ:** фактор – переменная, влияющая на другой показатель. В статистике выделяют факторы (независимые переменные) и результаты (зависимые). Анализ влияния факторов – ключевая задача (регрессионный, корреляционный анализ). Факторный анализ в другом смысле – метод уменьшения размерности, ищущий скрытые факторы (см. ниже по спец. вопросам).

Использование этих понятий при анализе данных обеспечивает строгий и количественный подход. С их помощью аналитик систематизирует данные (разбивая совокупности, группируя по признакам), рассчитывает основные характеристики и проверяет гипотезы. Например, зная понятия среднего и дисперсии, можно сравнить две группы и сделать вывод, значимо ли их различие. Коэффициент корреляции помогает обнаружить взаимосвязи между переменными (скажем, рост и вес имеют высокую корреляцию). Весь инструментарий Data Mining базируется на статистических концепциях – от простого описания до построения моделей и проверки их значимости.

6. Общий алгоритм анализа данных по этапам.

Процесс анализа данных обычно разбивается на **последовательность этапов**, каждый из которых важен для конечного результата. Общий алгоритм включает следующие шаги:

1. **Формулирование задачи и сбор исходных данных.** На этом этапе уточняется цель анализа (какую проблему решить, какие вопросы ответить) и определяется, какие данные необходимы. Производится сбор данных из доступных источников: базы данных, файлы, результаты наблюдений или экспериментов. Возможно, требуется интегрировать данные из нескольких источников в единый набор.
2. **Предварительная обработка данных (очистка).** Это критически важный этап подготовки данных. Реальные данные часто «загрязнены» ошибками и шумами: могут быть пропущенные значения, опечатки, дубликаты записей, противоречивая информация, anomalously ошибочные цифры или выбросы, шума и др. ²¹ . На этапе очистки устраняются подобные проблемы: редактируются или удаляются аномалии, заполняются пропуски, фильтруется шум, удаляются дубликаты и противоречивые записи ²² ²³ . Цель – получить корректный и согласованный набор данных, пригодный для анализа.
3. **Трансформация и подготовка данных.** После очистки данные могут требовать дальнейших преобразований: приведение форматов и единиц измерения к единому виду, нормализация числовых признаков (масштабирование), кодирование категориальных переменных (например, замена текстовых категорий на числовые индикаторы), агрегирование или детализация данных по необходимости, вычисление новых признаков (feature engineering). Эти шаги улучшают качество данных и адаптируют их под требования конкретных алгоритмов. В Logitom, например, к трансформации относят квантование, табличные замены значений, формирование скользящих окон и изменение структуры набора данных ²⁴ .
4. **Разведочный анализ данных (EDA).** Перед сложным моделированием обычно выполняется промежуточный анализ: рассчитываются статистические показатели, строятся простые визуализации (гистограммы, диаграммы, ящики с усами). Это помогает понять распределения, выявить явные зависимости или аномалии, наметить гипотезы. Хотя EDA не всегда выделяют как отдельный формальный этап, на практике он присутствует для получения интуиции о данных.
5. **Моделирование (построение моделей Data Mining).** Ключевой этап – применение алгоритмов анализа данных для извлечения знаний. Выбор метода зависит от задачи (см. вопрос 4 выше): может выполняться кластерный анализ, классификация, регрессионное моделирование, построение решающих деревьев, нейронных сетей, факторный анализ и пр. Здесь происходит «обучение» моделей на подготовленных данных. Например, строится регрессионная модель для прогноза показателя или кластеризуются клиенты на сегменты. В Logitom этот этап называется собственно *Data Mining* – построение моделей с использованием нейросетей, деревьев решений, карт Кохонена, ассоциативных правил и других методов ²⁵ .
6. **Оценка и интерпретация моделей.** Построенную модель необходимо оценить: насколько она точна и полезна. Проводится проверка на отложенных данных или кросс-валидация, вычисляются метрики качества (точность, MSE, R^2 , etc.). Если модель недостаточно качественна, может потребоваться возврат к предыдущим этапам (например, улучшить признаки, собрать больше данных или выбрать другой алгоритм). Также важно **интерпретировать** результаты: статистические модели позволяют выяснить, какие факторы значимы (например, коэффициенты регрессии), а сегменты кластеров – как они характеризуются.

7. Презентация результатов и принятие решений. Окончательный этап – представить найденные знания в удобной форме: отчёт, графики, дашборд, рекомендации. Часто используются средства визуализации, построение отчётности или OLAP-кубов для демонстрации результатов. На основе анализа формулируются выводы и предложения (например, какие действия предпринять в бизнесе). Если цель анализа – внедрить модель (предиктивная аналитика), то на этом шаге модель разворачивается в продакшн: например, на её основе строится автоматизированная система (рекомендательный сервис, система детекции мошенничества и т.д.).

Важно отметить, что анализ данных **итеративен**: возможны циклы возврата. Например, обнаружив на этапе моделирования проблемы с данными, возвращаются к их очистке или сбору дополнительных сведений. Тем не менее, описанный общий алгоритм – классическая последовательность, обеспечивающая структурированный подход к получению знаний из данных.

Вопросы по направлениям анализа

1. Варианты «загрязнённости» данных и способы их очистки в Loginom.

- **Виды «грязных» данных:** Реальные данные часто содержат различные дефекты. Основные варианты: (а) **Противоречивые или дублирующие сведения** – одна и та же сущность представлена несколькими записями, либо данные разных источников не совпадают (например, разночтения в написании названий, задвоения клиентов); (б) **Пропущенные значения** – не указаны данные в некоторых полях из-за сбоев или отсутствия информации; (с) **Аномальные значения (выбросы)** – отдельные наблюдения сильно выбиваются из общей массы (например, ошибка ввода: возраст 200 лет); (д) **Шум** – случайные флуктуации или помехи в данных, особенно в временных рядах (например, помехи датчиков); (е) **Ошибки ввода и опечатки** – человеческий фактор при занесении данных (некорректный формат, ошибочно лишний знак, названия с ошибками и т.п.) ²¹ ²⁶ .

- **Способы очистки в Loginom:** Платформа Loginom предоставляет инструменты для автоматизированной очистки данных, многие из которых входят в так называемую **парциальную обработку данных** (partial data preprocessing) ²⁷ . Методы очистки включают:

- **Удаление дублей и разрешение противоречий:** есть узел обнаружения дубликатов, который выявляет повторяющиеся записи. Дубликаты можно либо удалить, либо объединить, сформировав «золотую запись» (master record). Противоречивые сведения разрешают с помощью правил приоритетов или справочников – в Loginom можно на этапе интеграции данных настроить **стандартизацию справочников** (НСИ) для приведения разных написаний к одному варианту ²⁸ .

- **Обработка пропущенных данных:** Loginom предоставляет алгоритмы восстановления пропусков. Для временных упорядоченных данных применяется **интерполяция** – пропущенное значение заменяется рассчитанным по соседним точкам (например, метод аппроксимации на основе фильтра Калмана) ²⁹ ³⁰ . Для неупорядоченных данных используется метод наиболее вероятного значения – на основе всей выборки вычисляется и подставляется наиболее правдоподобное значение пропуска ³¹ ³² . Кроме того, Loginom поддерживает простые подходы: удалить записи с пропусками (если их мало) или заполнить средним/модой – выбор зависит от ситуации.

- **Выявление и корректировка выбросов:** В Loginom есть узел «Удаление аномалий», который автоматически находит значения, сильно отклоняющиеся от нормы, и может их корректировать или помечать ³³ . Например, можно задать степень чувствительности к выбросам, и алгоритм заменит слишком экстремальные точки на более правдоподобные (например, обрезка по квантилям или заменой на границы доверительного интервала). Также возможна ручная фильтрация – отсечение наблюдений, выходящих за определённые границы.

- *Сглаживание шума*: Для шумных рядов Loginom реализует **спектральные методы очистки** (Фурье-фильтрацию) и **вейвлет-фильтрацию**. Эти методы подробно раскрыты в следующем вопросе; кратко – высокочастотный шум убирается путём фильтрации частотного спектра (убираются компоненты выше заданного порога) ³⁴, а вейвлет-преобразование позволяет сгладить ряд, подавляя резкие выбросы в данных за счёт настройки уровня разложения ³⁵ ³⁶. Такие узлы используются на этапе очистки для получения более ровного сигнала без помех.

- *Исправление ошибок ввода*: Полностью автоматизировать устранение всех опечаток сложно, но Loginom может помочь. Во-первых, при импорте данных задаются форматы и типы полей – несоответствие формату сразу выявляется и можно настроить правила (например, обрезать лишние символы, конвертировать запятые в точки и т.п.). Во-вторых, для текстовых полей практикуют очистку с помощью словарей: например, можно подключить справочник названий и сравнить с ним поступающие значения, исправляя очевидные опечатки (в лекции упоминается использование тезаурусов для исправления ошибок) ²⁶. В сложных случаях, конечно, может потребоваться ручная правка. В любом случае, Loginom позволяет выделить такие подозрительные записи (фильтром по условию либо отдельным узлом контроля качества данных) и вынести их на проверку.

Таким образом, для каждого типа «загрязненности» данных в Loginom предусмотрены соответствующие методы очистки – либо встроенные узлы, либо комбинация инструментов. Благодаря этому можно автоматизировать процесс приведения данных в приемлемый для анализа вид с минимальным участием человека.

2. Использование Фурье- и вейвлет-анализа для очистки данных. Их использование в Loginom.

- **Фурье-анализ для очистки данных**: Преобразование Фурье представляет сигнал (или временной ряд данных) в виде суммы синусоидальных компонент с разными частотами. При очистке данных часто применяют *фильтрацию высоких частот*, т.е. удаление шумовой компоненты. Например, выполняется **быстрое преобразование Фурье (FFT)**, затем обнуляются спектральные компоненты выше некоторого порога частоты – тем самым устраняется высокочастотный шум ³⁴. После обратного преобразования Фурье получаем сглаженный сигнал, где мелкие колебания (вероятно являвшиеся шумом) приглушены, а низкочастотная структура (тренды, циклы) сохранена. Таким образом, Фурье-анализ позволяет **сгладить ряд данных**, выделить основные циклы и тенденции, убрав случайные пики. В контексте очистки это полезно, когда данные очень зашумлены – спектральный фильтр будет действовать как «шумоподавитель». В Loginom подобный подход реализован в узле сглаживания: пользователь может указать порог частоты или степень сглаживания, и алгоритм сам отсекает высокочастотные компоненты шума ³⁷. Кроме того, Фурье-методы применяются при анализе сезонных колебаний: вырезая определённые гармоники, можно удалять сезонный шум или, наоборот, выделять сезонный сигнал.

- **Вейвлет-анализ для очистки данных**: Вейвлет-преобразование – более сложный инструмент, разлагающий сигнал на базис «маленьких волн» (вейвлетов), локализованных во времени. Главное преимущество – *одновременное* представление данных в домене времени и частоты, что особенно полезно для нестационарных рядов (где свойства сигнала меняются со временем). Для очистки данных применяют **дискретное вейвлет-преобразование** с последующей реконструкцией, отбрасывая компоненты, отвечающие за шум. Проще говоря, на различных масштабах анализируются коэффициенты вейвлета, и мелкомасштабные (шумовые) колебания можно обнулить или подавить. В Loginom при использовании вейвлет-фильтрации настраивается **глубина разложения** (сколько уровней масштабов рассматривать) и **порядок (тип) вейвлета** ³⁵. Порядок вейвлета влияет на степень гладкости восстановленного ряда: при низком порядке остаются резкие

выбросы, при более высоком – выбросы сильнее сглаживаются ³⁶. Таким образом, можно отфильтровать шум, сохранив основные тренды и локальные особенности. Вейвлет-анализ хорошо убирает *случайные флуктуации* и помехи, минимально искажая сигнал, что особенно ценно, когда шум накладывается на значимые резкие изменения (вейвлет способен отличить один от другого).

- **Использование в Logiplot:** Платформа Logiplot включает и спектральное (Фурье), и вейвлет-преобразования как опции узлов предварительной обработки (парциальной обработки). Например, в узле «Сглаживание» можно выбрать метод Фурье-фильтрации или вейвлет-преобразования. В случае Фурье – задаётся граничная частота или критерий фильтрации, после чего узел автоматически пропускает только компоненты ниже порога (ниже заданной частоты), уменьшая шум ³⁴. При использовании *вейвлет-преобразования* – пользователь указывает глубину и тип вейвлета; Logiplot разложит ряд, отбросит детали нужного уровня и соберёт сигнал обратно, получая сглаженные данные. Оба метода активно применяются для очистки временных рядов (например, финансовых данных, показаний датчиков): Logiplot позволяет нажатием нескольких кнопок применить эти мощные математические подходы, без необходимости вручную писать код. Таким образом, Фурье- и вейвлет-анализ дополняют стандартные способы очистки, обеспечивая **гибкую фильтрацию шума** и улучшая качество данных перед дальнейшим анализом.

3. Понятие регрессионного анализа и его практическое применение.

- **Понятие регрессионного анализа:** Регрессионный анализ – это статистический метод для исследования зависимости одной переменной (называемой *зависимой переменной*, или результативного признака) от одной либо нескольких других переменных (*независимых переменных*, факторов) ³⁸. Проще говоря, регрессия строит математическую модель, которая описывает, как изменение входных факторов влияет на выход. Классический случай – **линейная регрессия**, где зависимость аппроксимируется линейной функцией. Цель регрессионного анализа – **предсказание** (прогнозирование) значения зависимой переменной по заданным значениям факторов, а также **оценка влияния факторов** (выявление, какие переменные и насколько сильно воздействуют на результат) ³⁹. Регрессия основывается на методе наименьших квадратов: подбирает коэффициенты модели так, чтобы минимизировать сумму квадратов отклонений прогнозируемых значений от фактических ⁴⁰ ⁴¹. Существует множество видов регрессии помимо линейной – полиномиальная, логистическая (для бинарных исходов), регрессия на основе деревьев решений, и др., но суть всегда в моделировании отношения между переменными.

- **Практическое применение:** Регрессионный анализ является одним из самых распространённых инструментов анализа данных, применяемым в разнообразных сферах:

- **Экономика и финансы:** моделирование рыночных трендов, прогноз цен, спроса и предложения. Например, регрессия может использоваться для **прогноза продаж** на основе затрат на рекламу, цен или сезонных факторов ⁴². Также – для оценки влияния макроэкономических показателей (ставок, инфляции) на прибыль компании.

- **Маркетинг и продажи:** компании применяют регрессию, чтобы понять, как различные факторы влияют на объём продаж. Например, **модель зависимости продаж от бюджета на рекламу, ценовой политики и времени года** поможет оптимизировать расходы ⁴³. Регрессионный анализ может показать, какая часть продаж объясняется рекламой, а какая – другими факторами, тем самым оценить ROI маркетинговых кампаний.

- **Социальные науки (социология, психология):** исследование связей между социальными факторами. Пример – регрессия для оценки влияния уровня образования, дохода и окружения на *уровень счастья* человека ⁴⁴. Или анализ факторов, влияющих на результаты тестов у студентов (часы подготовки, метод обучения, питание и т.п.). Регрессия позволяет проверить значимость каждого фактора и предсказывать исход (например, удовлетворённость жизнью).

- *Медицина и биология*: анализ влияния факторов риска на здоровье. Например, регрессионная модель может оценить, как вес, давление, уровень холестерина и возраст влияют на вероятность сердечно-сосудистого заболевания ⁴⁵ . С помощью логистической регрессии можно предсказывать вероятность болезни (выход – от 0 до 1) и выделять наиболее опасные факторы. В фармакологии – моделирование дозы и эффекта.

- *Инженерия и производство*: тут регрессия помогает **моделировать технологические процессы**. Например, зависимость прочности материала от температуры и времени обработки, или качество продукта от настроек оборудования. Регрессионный анализ выявляет, какие параметры производства критичны, и позволяет предсказывать выход (например, % брака) при заданных условиях ⁴⁶ . Это важно для оптимизации процессов и контроля качества.

- *Спорт*: в спортивной аналитике регрессия используется для **прогноза результатов матчей** или оценки вклада отдельных игроков. Например, регрессионная модель может предсказывать количество побед команды в сезоне на основе статистики игроков, бюджета, тренерского состава и т.д. ⁴⁷ . Это помогает менеджерам принимать решения по трансферам, стратегии игры.

Подводя итог, регрессионный анализ – универсальный метод, применимый везде, где нужно понять и количественно описать влияние факторов на результат. В практических задачах он ценен двояко: во-первых, даёт **прогнозы** (численные оценки, чего ожидать при заданных входных условиях), во-вторых, позволяет **делать выводы** о важности тех или иных причинных факторов, что поддерживает принятие решений на основе данных.

4. Основные задачи корреляционного и факторного анализа. Их решение в Loginot.

- **Задачи корреляционного анализа**: Корреляционный анализ нацелен на количественную оценку взаимосвязи между переменными. Главная задача – измерить, насколько изменение одной переменной сопровождается систематическим изменением другой. Для этого рассчитываются коэффициенты корреляции (например, Пирсона для количественных данных), которые принимают значения от -1 до 1 и показывают тесноту и направление связи (положительная, отрицательная). Практически корреляционный анализ решает такие задачи: (а) **Отбор значимых факторов**: если у нас много потенциальных независимых переменных, можно вычислить корреляцию каждого фактора с целевой переменной и отбросить те, у которых связь слабая или нулевая ⁴⁸ ⁴⁹ . Например, в Loginot реализована автоматическая фильтрация факторов: задаётся порог значимости, и все входные поля, имеющие модуль корреляции с выходом меньше порога, исключаются как незначимые ⁴⁸ . (b) **Обнаружение мультиколлинеарности**: анализ корреляций между самими независимыми признаками помогает найти сильно коррелированные между собой переменные (избыточные). Это важно перед построением регрессий – сильно коррелирующие факторы приводят к неустойчивости модели. (с) **Классификация связей**: выявление пар признаков с высокой положительной или отрицательной корреляцией, что может указывать на определённые закономерности или взаимозаменяемость признаков. Например, по корреляционной матрице продаж товаров можно заметить, что у двух товаров коэффициент -0.8 (сильная отрицательная корреляция) – значит, они *заменители* (рост продаж одного ведёт к падению другого), а у другой пары +0.7 – *сопутствующие товары* ⁵⁰ ⁵¹ .

- **Реализация в Loginot**: Loginot имеет встроенные средства для корреляционного анализа. Во-первых, это **визуализатор «Матрица корреляции»**, позволяющий рассчитать и увидеть матрицу коэффициентов для набора признаков ⁵² . На ней сразу видны пары с сильной связью. Во-вторых, как упомянуто, узлы моделирования (например, узел регрессии) могут выполнять предобработку – автоматическое исключение факторов с низкой корреляцией с целевым полем ⁴⁸ . В-третьих, есть возможность экспортировать коэффициенты корреляции в табличный вид для отчётов. Таким

образом, Loginom помогает как на этапе предварительного анализа (построить корреляционную матрицу), так и на этапе отбора переменных. Задача по сути сводится к выявлению *существенных* связей – слабые связи отсеиваются, сильные фиксируются для дальнейшей интерпретации.

- **Задачи факторного анализа:** Факторный анализ – методика сокращения размерности данных и выявления скрытых (латентных) факторов. Его главная задача: **обнаружить набор гипотетических факторов, которые объясняют наблюдаемые взаимосвязи между множеством исходных переменных**, при этом число факторов намного меньше числа исходных показателей. Практически это позволяет решить две проблемы: (а) **Упрощение модели:** заменить десятки коррелированных между собой показателей несколькими независимыми факторами, не сильно потеряв в информации. Выявленные факторы часто имеют содержательную интерпретацию (например, совокупность связанных признаков может отражать один общий «размер компании» или «уровень технологии») ^{53 54}. (б) **Борьба с избыточностью и коллинеарностью:** когда признаки коррелируют, факторный анализ группирует их в факторы, устраняя мультиколлинеарность. В результате можно **снизить размерность** данных, ускорить обучение моделей и упростить интерпретацию результатов ^{55 56}. Например, в большом социологическом опросе сотня вопросов может свестись к нескольким факторам (скажем, «социальный статус», «консерватизм» и т.п.). Ещё одна задача – **построение новых признаков (факторов)** для последующего анализа, особенно перед использованием моделей типа нейросетей, где лишние измерения увеличивают требования к объёму выборки ⁵⁷.

- **Реализация в Loginom:** В Loginom есть специальный обработчик «Факторный анализ», который реализует метод главных компонент (PCA) ⁵⁸. Пользователь может указать исходный набор переменных, а инструмент выдаст новые переменные – главные компоненты, ранжированные по вкладу в общую дисперсию. Loginom позволяет настроить, сколько факторов оставить: можно задать суммарный процент объяснённой дисперсии, и платформа автоматически выберет минимальное число факторов, покрывающее, например, 90% вариации данных ^{59 60}. Также возможно вручную просмотреть собственные значения и решить, сколько факторов значимо. Результат факторного анализа – набор факторных нагрузок (коэффициентов, показывающих связь исходных переменных с фактором) и сами рассчитанные значения факторов для каждого наблюдения. Эти новые факторы затем можно использовать в дальнейших моделях как независимые переменные вместо исходного множества коррелированных показателей. Таким образом, Loginom полностью автоматизирует решение задачи факторного анализа: несколько кликов – и данные сведены к компактному факторному описанию.

В итоге: **корреляционный анализ** отвечает на вопрос «какие переменные связаны и как сильно?», а **факторный анализ** – «какие скрытые причины стоят за этими связями?». Оба инструмента дополняют друг друга (часто факторный анализ проводится на основе корреляционной матрицы) и оба поддерживаются функционалом Loginom для удобства аналитика.

5. Цели кластерного анализа при анализе данных. Основные алгоритмы кластерного анализа.

- **Цели кластерного анализа:** Кластерный анализ (кластеризация) преследует цель **разделить объекты на однородные группы** без заранее заданных меток, чтобы лучше понять структуру данных. Основная идея – объекты внутри одного кластера максимально похожи друг на друга, а объекты из разных кластеров – существенно различаются ^{61 62}. Цели применения кластеризации включают:

- **Сегментация и понимание данных:** выделение природных групп позволяет аналитикам понять, какие типичные категории присутствуют. Например, кластеризация клиентов банка может выявить сегменты: «консервативные вкладчики», «активные заёмщики», «молодёжь – цифровые

пользователи» и т.п. Это даёт ценные инсайты для маркетинга и обслуживания.

- *Упрощение дальнейшего анализа*: разбив данные на группы, можно затем изучать каждую группу отдельно более подходящими методами. Например, в одном кластере можно применять один метод прогнозирования, в другом – другой (адаптация моделей под сегменты) ⁶³ ⁶⁴ . Так повышается точность и осмысленность анализа.

- *Компактное представление данных*: кластеризация позволяет заменить множество близких объектов одним «прототипом» (например, центроидом кластера). Это сокращает данные – вместо хранения миллионов записей можно хранить усреднённые данные по кластерам, что удобно для некоторых задач (сжатие, визуализация) ⁶⁵ ⁶⁶ .

- *Поиск новизны (аномалий)*: объекты, которые не присоединились ни к одному кластеру или образовали крошечный кластер, часто являются выбросами или редкими случаями. Поэтому кластеризация помогает обнаруживать нетипичные наблюдения (аномалии), что важно, например, для поиска мошеннических транзакций (они могут образовать отдельный небольшой кластер) ⁶⁷ ⁶⁸ .

В целом, цель кластерного анализа – **открытие скрытых структур** в данных: будь то рыночные сегменты, темы в текстах, группы схожих изображений и т.д. Он относится к описательным методам Data Mining и часто служит отправной точкой для других анализов.

- **Основные алгоритмы кластерного анализа**: Существует несколько классов алгоритмов кластеризации, каждый со своими особенностями:
- *Методы разбиения (центроидные)*: самый известный – алгоритм **K-средних (K-means)**. Пользователь задаёт число кластеров K , алгоритм случайно выбирает K начальных центров, затем итеративно распределяет точки по ближайшим центрам, пересчитывает центры как средние по кластерам, и так до сходимости ⁶⁹ ⁷⁰ . K-means быстр и прост, но требует заранее знать K и чувствителен к выбросам (одно далеко лежащее значение может сильно сместить центр). Существуют улучшения K-means: например, **K-medians** (центр рассчитывается как медиана – более устойчиво к выбросам) ⁷¹ ; **K-medoids** (выбирает центром реальный объект из кластера); **G-means** – алгоритм, который сам определяет оптимальное число кластеров, добавляя кластеры, пока данные внутри кластеров не будут нормально распределены ⁷² ⁷³ . В Logitom, например, кластеризация реализована на основе K-means, а если число кластеров неизвестно, используется G-means, который подбирает число автоматически ⁷² .
- *Иерархические методы*: эти алгоритмы строят вложенную иерархию кластеров (кластерное дерево – дендрограмму). Бывают **агломеративные** (bottom-up) – стартуют с каждого объекта как отдельного кластера и постепенно объединяют самые близкие кластеры, пока не останется заданное число кластеров или один кластер; и **дивизивные** (top-down) – наоборот, начинают со всех данных в одном кластере и рекурсивно делят. Иерархическая кластеризация позволяет увидеть структуру на разных уровнях детализации. Она не требует задавать K заранее, но обычно сложнее вычислительно ($O(n^2)$). Классический пример – алгоритм ближайшего соседа, или метод Уорда (Ward's method), минимизирующий увеличение внутрикластерной дисперсии при объединении.
- *Плотностные алгоритмы*: например, **DBSCAN** и его улучшения. Эти методы ищут области повышенной плотности точек и объявляют их кластерами, а точки в разреженных областях – шумом. DBSCAN не требует заранее число кластеров, способен выявлять кластеры произвольной формы и отсеивать выбросы. Параметры – радиус окрестности и минимальное число точек для формирования кластера. Хорошо работает, когда кластеры в данных имеют разную форму и размер, но явно выражены плотностью.

- *Алгоритмы на графах и сетях*: сюда относятся методы на основе минимального остовного дерева, разреза графа, а также *самоорганизующиеся карты Кохонена* (нейронные сети для кластеризации, см. следующий вопрос). Карты Кохонена можно рассматривать как особый вид кластеризации с сохранением топологии – они формируют двумерную решётку прототипов, «похожа» ли на соседей.
- *Размытые алгоритмы*: например, **Fuzzy C-means** (С-средних) – расширение K-means, где каждый объект не жестко принадлежит кластеру, а имеет степень принадлежности к каждому кластеру ⁷⁴. Это позволяет учитывать неопределенность: объект может частично относиться к двум кластерам, если находится на границе. Размытая кластеризация полезна, когда границы групп нечеткие.

В Loginom доступны по крайней мере центроидные методы (K-means, G-means) через узел «Кластеризация» ⁶² ⁷⁵. Также можно реализовывать и другие алгоритмы (например, иерархическую кластеризацию) через скрипты или внешние модули, но основные встроенные – К-средних и его вариации. Loginom автоматически визуализирует результаты кластеризации (например, можно построить диаграммы кластеров) и предоставляет сведения о центрах кластеров, размере кластеров и пр. Главное – правильно выбрать алгоритм под задачу: если известно число групп, применяют K-means; если нет – G-means; если кластеры сложной формы – можно прибегнуть к другим средствам. В любом случае, кластерный анализ – мощный инструмент разведочного анализа, и знание основных алгоритмов позволяет гибко сегментировать данные для достижения целей, перечисленных выше.

6. Возможности использования нейронных сетей в Loginom.

- **Neural Networks в анализе данных**: Искусственные нейронные сети – класс моделей машинного обучения, имитирующих работу мозга. Они особенно хороши для поиска сложных нелинейных зависимостей и паттернов в данных. В контексте анализа данных нейронные сети применяются для задач классификации, прогнозирования, распознавания образов, кластеризации и др. (в зависимости от архитектуры сети). Примеры: многослойный персептрон (MLP) для прогнозирования и классификации, рекуррентные сети для временных рядов, сверточные сети для изображений и т.д.

- **В Loginom**: Платформа Loginom имеет встроенную поддержку технологий машинного обучения, включая **нейронные сети** ⁷⁶. Это позволяет пользователю без глубокого программирования воспользоваться силой нейросетевых моделей. Конкретные возможности:

- *Обучение моделей на основе нейронных сетей*: Loginom предоставляет узлы для создания и обучения нейронной сети. Например, есть узел типа «Нейронная сеть» (или он может быть частью узла Data Mining общего), который позволяет настроить архитектуру (количество скрытых слоёв, нейронов, функцию активации) и обучить сеть на предоставленном наборе данных. Обучение происходит внутри платформы – пользователь лишь запускает процесс, а алгоритм сам подбирает веса сети, минимизируя ошибку. Таким образом можно решать задачи **регрессии и классификации** (контролируемое обучение) с помощью нейросети. Loginom включает этот метод наряду с деревьями решений, SVM и другими для задач классификации ⁷⁷.

- *Использование нейросетей для прогнозирования*: Обучив сеть, её можно применять для предсказания на новых данных. В Loginom после обучения узел нейронной сети может выдавать прогнозы (например, вероятности классов). Это даёт возможность строить сложные прогнозные модели, которые учитывают нелинейные сочетания факторов. Преимущество – нейронные сети могут выявить скрытые взаимосвязи, которые не явны при линейных или деревьях, особенно если данных много и они сложные.

- *Нейронные сети для кластеризации*: Loginom поддерживает **самоорганизующиеся карты Кохонена** – особый вид нейронных сетей для неконтролируемого обучения (см. следующий вопрос).

Карты Кохонена можно создать и обучить в LogiDom, результатом будет разбиение данных на кластеры (узел «Карта Кохонена»). Таким образом, нейросетевые методы охватывают не только прогноз, но и сегментацию. В документации указано, что LogiDom поддерживает анализ данных от простой логики до машинного обучения, включая нейросети ⁷⁶, а упоминаются конкретно *карты Кохонена* как один из инструментов Data Mining ⁷⁸.

- *Low-code реализация*: Все настройки нейронных сетей (выбор количества нейронов, эпох обучения, скорости обучения) доступны через интерфейс. То есть, даже не будучи программистом, аналитик может экспериментировать с архитектурой сети, опираясь на визуальные подсказки. LogiDom снимает необходимость писать код на Python/R для типовых сетевых моделей – всё делается компонентно.

- *Интеграция с другими узлами*: Нейронные сети могут быть частью комплексного сценария: например, данные сначала очищаются, затем через узел «Нормализация» приводятся к единому масштабу, потом поступают в узел «Нейронная сеть» для обучения модели, а результат идёт в узел «Применение модели». В результате, в одной среде можно полностью реализовать цикл построения и внедрения нейросетевой модели. Также, результаты сети можно визуализировать (например, построить ROC-кривую для классификации, используя вычисленные вероятности).

- *Примеры использования*: Если нужен прогноз сложного показателя, где зависимость нелинейна и традиционные регрессии недостаточно точны – можно попробовать нейронную сеть. В практике LogiDom нейросети могут применяться, скажем, для **кредитного скоринга** (предсказания вероятности дефолта клиента по множеству параметров), для **прогноза оттока клиентов**, для **распознавания категорий товаров по текстовому описанию** (сеть как классификатор) и т.п. Важный плюс – нейронные сети часто дают более высокую точность на больших массивах разнородных данных, а LogiDom позволяет их обучать параллельно с другими методами и сравнивать результаты.

В заключение: возможности LogiDom в части нейронных сетей делают современный инструмент ML доступным бизнес-пользователям. Можно решать широкий спектр задач – от предсказания числовых показателей до автоматической классификации клиентов – используя мощности нейросетей, не выходя из интерфейса LogiDom.

7. Сети Кохонена: назначение, отличие алгоритма их построения от стандартных методов кластерного анализа.

- **Назначение сетей Кохонена**: Самоорганизующаяся карта Кохонена (Self-Organizing Map, SOM) – это вид нейронной сети без учителя, предназначенный в первую очередь для **кластеризации и визуализации многомерных данных**. Её назначение – проецировать данные высокой размерности на низкоразмерное (обычно 2D) пространство нейронов, сохраняя топологическую близость: схожие объекты попадают на соседние нейроны карты ⁷⁹. Практически, карта Кохонена разбивает данные на кластеры (каждый нейрон сети представляет кластер), но дополнительно даёт структуру этих кластеров – их можно визуально расположить на решётке и увидеть отношения между кластерами. SOM часто используют для сегментации, когда важно не только получить группы, но и расположить их в некотором порядке, выявить *карту* данных. Например, в маркетинге карта Кохонена может разместить клиентов на двумерном пространстве таким образом, что соседние области – схожие сегменты, а дальние – сильно отличающиеся сегменты. То есть, помимо кластеризации, сети Кохонена выполняют роль *инструмента снижения размерности для визуального анализа*. Кроме того, SOM применяется там, где обычный кластерный анализ затруднён из-за формы кластеров или необходимости мягкого плавного разделения – Кохонен обеспечивает сглаженное распределение весов и может выявлять кластеры произвольной формы.

- **Отличие алгоритма построения от стандартных методов кластеризации:** Алгоритм обучения сети Кохонена существенно отличается от, скажем, K-means или иерархической кластеризации:

- *Обучение с конкуренцией и кооперацией:* В сети Кохонена нейроны расположены обычно на двумерной решётке (например, 10x10 узлов). Изначально каждому нейрону назначается случайный *весовой вектор* такой же размерности, как входные данные. Затем алгоритм итеративно перебирает обучающие примеры: для каждого входного вектора находится нейрон, чей весовой вектор наиболее близок к входу – это *нейрон-победитель* (Best Matching Unit). После этого выполняется **правило обучения Кохонена**: победитель и его соседи по решётке обновляют свои веса – сдвигаются ближе к входному вектору^{80 81}. Весы корректируются по формуле, содержащей коэффициент обучения и функцию соседства, которая максимальна для победителя и уменьшается с удалением от него (гауссова или другая). Таким образом, не только выигравший нейрон учится (как было бы в конкурентной обучении без соседства), но и его окружение. Это ключевое отличие: **нейроны самоорганизуются с учётом топологии**, тогда как K-means, например, не имеет понятия соседних кластеров – каждый центр обновляется независимо.

- *Отсутствие глобальной целевой функции:* Стандартные методы (K-means) явно минимизируют сумму квадратов расстояний внутри кластеров. У сети Кохонена нет явной целевой функции, она использует *эвристическое обучение*, следствием которого является упорядочивание весов. Соответственно, критерий сходимости – изменения весов становятся малыми после нескольких эпох обучения. SOM не гарантирует оптимальности кластеризации по какому-либо критерию, но на практике даёт хорошее разбиение плюс упорядоченность.

- *Постепенное снижение параметров:* В процессе обучения SOM уменьшается коэффициент обучения и радиус соседства. Сначала нейроны подстраиваются грубо, захватывая большие области (большой радиус соседства – почти вся карта двигается), потом радиус сокращается, и к концу обучения только ближайшие соседи корректируются. Это создает эффект, что сначала карта выстраивает грубую топологию, а потом «тонко настраивается» локально. K-means же не имеет такого поэтапного сокращения радиуса – он с самого начала оперирует заданным K, без понятия масштабов.

- *Результат – упорядоченная карта:* Выход сети Кохонена – это решётка нейронов, каждый с весовым вектором. Эти весовые векторы являются центроидами кластеров, аналогично как в K-means, но они расположены на сетке. Близкие по данным центроиды оказываются соседями на сетке благодаря алгоритму. В K-means центры кластеров – просто точки в пространстве данных, без «географии» между ними. В SOM же можно сказать: кластер A соседствует с B на карте, значит, эти кластеры более похожи между собой, чем, скажем, кластеры A и C, находящиеся далеко на карте. Это **топологическое упорядочивание** – уникальное свойство SOM.

- *Отсутствие необходимости задавать число кластеров явно:* В SOM фактически число кластеров равно числу нейронов на карте. Но часто карту берут довольно «передетализированной» (например, 100 нейронов), а потом схожие нейроны можно объединить или рассматривать области карты. То есть SOM предоставляет более тонкое разбиение, из которого аналитик может извлечь нужное число кластеров, смотря на карту (например, выделить зоны плотности). В K-means приходится *заранее* фиксировать K или использовать правила (ELBOW, BIC) для выбора K.

- **Итого, чем алгоритм Кохонена отличается от стандартной кластеризации:** (1) *Наличие топологии нейронов:* кластеры (нейроны) «знают» о соседях и обучаются совместно – у классических алгоритмов кластеры обособлены. (2) *Процесс обучения с локальным обновлением:* Кохонен – итеративный, «биологически правдоподобный» процесс конкуренции и сотрудничества; K-means – итеративное глобальное пересчитывание центров. (3) *Выход – не просто группы, а структурированная карта:* позволяет визуализировать, например, с помощью теплокарты (U-matrix) расстояния между нейронами, выделять области на карте, что

значительно облегчает **интерпретацию**. (4) *SOM лучше сохраняет геометрию данных*: близкие по исходным признакам объекты будут близки на карте – K-means сохраняет это лишь в пределах кластера, а отношения между кластерами не отражаются.

В Loginom карты Кохонена поддерживаются как один из инструментов кластерного анализа. Пользователь может задать размер решётки (например, 10x10), запустить обучение – и получить карту. Далее Loginom позволяет визуализировать результат: например, отобразить карту, раскрашенную по какому-то признаку, или показать, какие нейроны (кластеры) содержат наибольшее число объектов. От аналитика не требуются тонкие настройки – платформа сама реализует алгоритм Кохонена. Практически это даёт альтернативу K-means: если нужна более наглядная сегментация или предполагаются сложные отношения между группами, сеть Кохонена – отличный выбор. Она особенным образом дополняет стандартную кластеризацию, предоставляя **самоорганизующуюся проекцию данных**, что уникально.

8. Понятие ассоциативного анализа данных и его использование на практике. Возможности в Loginom.

- **Понятие ассоциативного анализа**: Речь идёт об **поиске ассоциативных правил** в данных (affinity analysis). Ассоциативный анализ выявляет закономерности вида «Если X, то Y» в большой базе транзакций или событий. Классический пример – *анализ рыночной корзины*: по данным о покупках выявляются правила типа «если куплен товар A, то с высокой вероятностью в той же корзине будет товар B»⁸². Ассоциативные правила состоят из двух частей: **условие (Left-hand side)** и **следствие (Right-hand side)**, например: {молоко, хлеб} \Rightarrow {масло}, читается: «если в корзине есть молоко и хлеб, то обычно есть и масло». Для каждого правила вычисляются меры *поддержки* и *достоверности* (confidence). **Поддержка (support)** – доля транзакций, содержащих одновременно условие и следствие⁸³. **Достоверность (confidence)** – условная вероятность появления следствия при наличии условия⁸⁴, т.е. число транзакций с обоими / число транзакций с условием. Например, правило «молоко \Rightarrow хлеб» с достоверностью 0.75 значит: в 75% случаев, когда покупают молоко, вместе берут и хлеб⁸⁵. Ассоциативный анализ ищет все правила, превышающие заданные пороги поддержки и достоверности. Таким образом, понятие ассоциативного анализа – это **автоматическое обнаружение скрытых связей между событиями** (товарами в корзине, симптомами болезней, последовательностями действий пользователей и т.д.).

- **Использование на практике**: Ассоциативные правила впервые получили известность в розничной торговле (отсюда термин market basket analysis)⁸². Практические применения:

- *Ритейл и продажи*: выявление часто совместно покупаемых товаров для оптимизации ассортимента, мерчандайзинга (раскладка товаров вместе), проведения перекрёстных акций («купил принтер – предложи картридж»). Например, анализ чеков может показать неожиданную связь – классический (хотя и легендарный) пример: подгузники и пиво часто покупают вместе, что позволяет соответствующим образом располагать товары.

- *Интернет-торговля*: ассоциативный анализ лежит в основе рекомендаций типа «С этим товаром также покупают...». Большие онлайн-площадки анализируют миллионы корзин покупателей, чтобы формировать блоки рекомендаций на страницах товара.

- *Банковское дело*: по набору услуг, которыми пользуется клиент, могут быть правила типа «если у клиента есть зарплатный проект и кредитная карта, то с вероятностью X% он возьмёт потребительский кредит». Это помогает прогнозировать потребности и предлагать дополнительные продукты.

- *Телеком*: анализ оттока – ассоциативные правила могут выявить комбинации условий, ведущие к уходу клиента (например: {плохое качество связи, нет тарифных опций} \Rightarrow {отток} с высокой достоверностью). Тогда можно превентивно работать с такими клиентами.

- *Медицина*: поиск сочетаний симптомов и диагнозов. Правила вида «{симптом А, симптом В} ⇒ {заболевание Х} (confidence 90%)» помогают при диагностике – если у пациента совместно наблюдаются А и В, есть высокая вероятность Х. Или в фармакологии – какие препараты часто назначаются совместно.
- *Безопасность*: выявление типичных сочетаний действий злоумышленников. Например, в логах системы: {вход в систему ночью, запуск несанкционированного процесса} ⇒ {инцидент безопасности}. Это помогает создавать правила для систем мониторинга.
- *Общее*: ассоциативный анализ можно применять ко многим видам транзакционных данных – веб-клики (последовательности посещённых страниц), голосования (часто совпадающие ответы депутатов – «коалиции интересов»), тексты (часто встречающиеся вместе слова – для тематического моделирования).

Главное – ассоциативные правила эффективны, когда у нас много однотипных транзакций, содержащих наборы элементов, и мы хотим выявить *локальные паттерны совместного появления элементов*.

- **Возможности Loginom**: В Loginom есть узел «Ассоциативные правила», который автоматически выполняет поиск правил по заданным параметрам ⁸⁶ ⁸⁷. Пользователь настраивает минимальную поддержку и достоверность (и/или другие метрики, например лифт), указывает, какие поля считать «элементами транзакции». Затем Loginom применяет алгоритм (обычно **Apriori** или похожий) для генерации правил ⁸⁸. Apriori работает итеративно: находит частые одиночные элементы, затем строит частые пары, тройки и т.д., пока возможно. В Loginom это прозрачно – на выходе узла пользователь получает список правил с расчетом поддержки, confidence, и других метрик (например, **lift** – повышение вероятности по сравнению со случайным, **leverage** и др., которые тоже есть в теории правил ⁸⁹). Есть возможность ограничить максимальный размер комбинации, или зафиксировать, что должно быть в левой/правой части.

Loginom позволяет легко **применять ассоциативные правила на практике**: после генерации их можно отфильтровать, отсортировать по интересующим метрикам и представить бизнесу. Например, правило «молоко ⇒ хлеб (75%)» буквально будет выведено, и известно, что support, скажем, 30% (т.е. в 30% чеков есть и молоко и хлеб) ⁸⁵. Это ценная информация для розницы. Технически, решение в Loginom автоматизирует весь pipeline: агрегирует транзакции, прогоняет алгоритм, выдает результат. Такой анализ, который вручную занял бы много времени, здесь выполняется быстро даже на больших данных.

Подведём итог: **ассоциативный анализ** ищет скрытые «если – то» шаблоны в данных; на практике он широко используется для рекомендаций и анализа корзин; а Loginom предоставляет готовый инструмент для этой задачи, значительно облегчая его проведение и позволяя бизнес-пользователям обнаруживать неочевидные взаимосвязи в своих данных.

Вопросы по технике работы в Loginom

1. Основные особенности и возможности Loginom.

Loginom – современная российская аналитическая платформа класса BI/Data Mining, отличающаяся удобством *low-code* подхода. Её основные особенности и возможности:

- **Комплексный аналитический процесс:** Loginom покрывает все этапы работы с данными – от подключения и извлечения данных до подготовки, анализа и финальной визуализации ². Пользователю доступны инструменты ETL (extract-transform-load), статистического анализа, построения моделей, прогнозирования и представления результатов без необходимости переключаться между разными программами.
- **Интеграция данных из разных источников:** платформа умеет подключаться к популярным базам данных (SQL, NoSQL), импортировать данные из файлов CSV, Excel, JSON, XML, а также получать данные через API, из web-сервисов и соцсетей ⁹⁰. Наличие множества коннекторов и возможности настройки подключений позволяет консолидировать разнородные данные в одном месте.
- **Сложные вычисления и трансформации без программирования:** Loginom предоставляет богатый набор узлов для обработки данных – фильтрация, сортировка, группировка, соединение (join) таблиц, вычисление новых показателей по формулам, очистка от выбросов и пропусков и т.д. Все это делается визуально, перетаскиванием компонентов и настройкой параметров, что снижает порог входа для аналитиков, не владеющих языками программирования.
- **Машинное обучение и Data Mining:** платформа имеет встроенные алгоритмы интеллектуального анализа: регрессия (линейная, логистическая), решения дерева, случайный лес, нейронные сети, кластеризация (k-means, G-means), факторный анализ (метод главных компонент), ассоциативные правила, анализ временных рядов и др. Т.е. покрываются практически все типы задач анализа данных ⁹¹ ⁹². Пользователь может настраивать эти алгоритмы через понятный интерфейс и быстро получать модели.
- **Прогнозирование и оптимизация:** Loginom позиционируется как инструмент предиктивной аналитики – позволяет строить прогнозы (например, продажи, спрос) на основе исторических данных. Также доступна оптимизация – поиск наилучших решений по моделям (например, подбор оптимального набора параметров для максимизации прибыли, используя перебор или сценарный анализ).
- **Анализ больших данных:** Платформа заточена под работу с очень большими наборами данных – реализованы механизмы потоковой обработки, кэширования, оптимизации вычислений с учётом объёма (в лекциях отмечено, что Loginom поддерживает функции анализа Big Data) ⁹³. При необходимости можно подключать распределённые хранилища.
- **Web-ориентированность и многопользовательский режим:** Loginom – полностью веб-приложение (сервер + браузерный интерфейс) ⁹. Это означает кроссплатформенность (работает через браузер в Windows, Linux, Mac) и отсутствие сложной установки на компьютеры аналитиков. Кроме того, предусмотрен **многопользовательский доступ** – несколько пользователей могут работать одновременно, имея свои учётные записи и разграниченные права доступа ⁹⁴ ⁹⁵. Администратор может управлять правами на проекты, узлы, данные, обеспечивая безопасность и совместную работу команды.
- **API и встраивание:** Для интеграции с другими системами Loginom предоставляет API (Application Programming Interface). Это позволяет автоматизировать обмен данными – например, другой софт может вызывать расчёты Loginom или получать от него результаты

через REST/SOAP веб-сервисы ⁹⁶ ⁹⁷ . Такая открытость делает Loginom не изолированной системой, а частью общей ИТ-инфраструктуры предприятия.

- **Визуализация, отчётность и OLAP:** Loginom обладает средствами визуального представления результатов. Встроенный инструментарий отчётности позволяет строить таблицы, графики, диаграммы, дашборды прямо внутри платформы ⁹⁸ ⁹⁹ . Есть поддержка интерактивного анализа – построение OLAP-кубов (см. вопрос 6 ниже), что даёт возможность пользователям самостоятельно «крутить» данные (slice and dice) в реальном времени. Отчёты могут экспортироваться (в PDF, картинки) либо предоставляться пользователям через веб-интерфейс.
- **Простота использования (интуитивный UI):** Интерфейс Loginom разработан таким образом, что для решения аналитических задач не требуется умение программировать. Компонентный подход (перетаскивание узлов на рабочее поле), мастера настройки, подсказки, цветовая индикация состояния узлов – всё это помогает быстро освоиться. Платформа на русском языке и снабжена обширной справочной информацией. По сути, её может использовать бизнес-аналитик, знакомый с концепциями данных, но не являющийся ИТ-специалистом.
- **Проектный подход и воспроизводимость:** Все действия в Loginom оформляются в виде *проекта*, который можно сохранить, повторно запустить с новыми данными, модифицировать. Это обеспечивает воспроизводимость анализа и удобство сопровождения – всегда можно открыть сохранённый сценарий и увидеть, как получены те или иные результаты.
- **Расширяемость:** Если стандартных узлов недостаточно, Loginom позволяет расширять функциональность: можно использовать узел скрипта (Python, R, SQL) для выполнения специфических вычислений, подключать внешние библиотеки и модули. Таким образом, продвинутые пользователи не ограничены встроенным списком возможностей.

В целом, Loginom сочетает **широкий функционал** (как у дорогих иностранных BI-платформ) с **удобством и локализацией** под российского пользователя. Его ключевые преимущества: интеграция данных, автоматизация анализа, доступность методов Data Science, совместная работа и наглядная подача результатов.

2. Структура панелей Loginom и их использование при анализе.

Интерфейс Loginom можно разделить на несколько основных областей (панелей), каждая из которых выполняет свою функцию в процессе анализа ¹⁰⁰ ¹⁰¹ :

- **Левая боковая панель (Главное меню и навигация):** Это вертикальное меню на самом левом краю экрана. Оно содержит кнопки: «Меню», «Пакеты», «Навигация», «Файлы», «Процессы».
- **Раздел «Меню»** позволяет создавать новый проект (пакет), открывать существующие, сохранять проект, настраивать параметры системы и т.д.
- **Раздел «Пакеты»** отображает список всех открытых проектов (пакетов) и их содержимое. Здесь можно переключаться между проектами, добавлять в проект новые компоненты, настраивать свойства проекта.
- **Раздел «Навигация»** показывает иерархию объектов внутри текущего проекта – например, сценарии, источники данных, подключения. Помогает быстро перемещаться к нужному узлу или модулю внутри большого проекта.
- **Раздел «Файлы»** используется для управления файлами, связанными с проектом (например, локальные копии импортированных файлов, промежуточные сохранения данных). Можно

просмотреть доступные наборы данных, загруженные в проект, выполнить импорт новых файлов.

- **Раздел «Процессы»** – панель мониторинга выполнения (она дублирует функциональность нижней панели «Процессы», см. ниже, но может быть раскрыта здесь). Она показывает, какие узлы сценария выполняются, завершены или ожидают активации.

Левая панель служит **навигационным «хребтом»**: через неё пользователь управляет жизненным циклом проекта, открывает нужные окна. Например, чтобы подключиться к базе данных, сначала через «Пакеты» откроем раздел подключений, а чтобы отслеживать прогресс – откроем «Процессы».

- **Верхняя панель (панель вкладок и адресная строка)**: Сверху интерфейса находятся вкладки открытых пакетов и адресная строка. Каждая открытая вкладка представляет отдельный пакет (проект) или отдельный объект внутри пакета (например, открытый узел-сценарий). Адресная строка отображает путь навигации внутри пакета – например, «Пакет_1 > Сценарий_А». Также на верхней панели есть элементы управления навигацией (стрелки вперёд/назад, подобно браузеру, для перехода по истории просмотров) ¹⁰².
- **Использование**: верхняя панель позволяет быстро переключаться между несколькими одновременно открытыми проектами или разделами одного проекта. Например, можно параллельно открыть два сценария в разных вкладках и переключаться для сравнения. Адресная строка помогает понять, где мы находимся (особенно при глубокой вложенности, например внутри подмодели сценария).
- **Рабочая область (центральная панель)**: Это основное пространство справа от главного меню, где происходит построение и настройка аналитических схем. Рабочая область, в свою очередь, разделена на две части:
 - **Левая часть рабочей области – панель компонентов и структуры**: Здесь отображаются **«Компоненты»** – библиотека всех доступных узлов и действий, сгруппированных по категориям ¹⁰³ ¹⁰⁴. Например, категории: «Источники данных», «Преобразования», «Аналитика», «Машинное обучение», «Вывод данных» и т.д. Пользователь может сворачивать/разворачивать эти списки. Чтобы использовать компонент, его нужно **перетащить мышью** на правую часть – область построения сценария ¹⁰⁵. Кроме того, на левой панели часто отображается структура решения: иерархия пакета – например, список сценариев внутри пакета, список подключений, наборов данных. Выбрав, скажем, конкретный сценарий, мы увидим содержащиеся в нём узлы. По сути, эта панель сочетает библиотеку инструментов и проводник по текущему решению.
 - **Правая часть рабочей области – область построения сценария и просмотра данных**: Это ключевое место, где графически конструируется *сценарий анализа (workflow)*. Пользователь размещает узлы (из левой панели) и соединяет их портами (линии связи между узлами), формируя поток данных. Каждый узел представлен прямоугольником с иконками входных/выходных портов. Здесь можно перемещать узлы, выстраивать последовательность расчётов, группировать узлы. При клике на узел обычно справа открывается окно его настроек (мастер настройки) – оно может всплывать над рабочей областью или в ней. Также эта область используется для **визуализации результатов**: например, если выполнить узел и открыть результат (быстрый просмотр), появится таблица данных или график прямо в рабочей области, либо в виде всплывающего окна. Таким образом, правая часть – это и конструктор, и отладчик: тут же можно видеть и данные, проходящие через узлы, и сами процессы.

Использование: рабочая область – центр активности аналитика. Например, для создания сценария нужно из панели «Компоненты» взять узел «Импорт Excel» и перетащить на холст, затем взять узел «Фильтр» и соединить с импортом, далее узел «Регрессия» и т.д. – получим цепочку. Если требуется просмотреть промежуточный результат – достаточно щёлкнуть по выходному порту нужного узла и выбрать «Быстрый просмотр», чтобы увидеть данные в табличной форме. Можно также строить графики: например, подключить узел визуализации (диаграмму) к данным. В процессе разработки сценария левая панель помогает выбрать нужные узлы, а правая – непосредственно собрать и настроить алгоритм анализа.

- **Нижняя панель (панель «Процессы»):** В нижней части окна находится горизонтальная панель процессов, по умолчанию скрытая (её можно закрепить, развернув) ¹⁰⁶. Панель «Процессы» показывает информацию о выполнении узлов в текущей сессии Loginom ¹⁰⁷. Структура представления: порядковый номер процесса, название процесса (обычно соответствует запуску сценария или части сценария) и список узлов, выполняющихся внутри процесса с указанием статуса и процента выполнения ¹⁰⁸. Каждый раз при запуске сценария создаётся запись «Активация узлов» и далее перечисляются узлы, которые выполняются (возможно, параллельно). Завершённые узлы могут отмечаться зелёным, текущие – иконкой выполнения, упавшие – красным.
- *Использование:* Эта панель служит **монитором выполнения и отладки**. Если сценарий сложный и включает, скажем, обращение к базе или обучение модели, на панели процессов видно, сколько % уже выполнено, не завис ли процесс. В случае ошибок узел подсвечивается – можно кликнуть и прочитав сообщение об ошибке (например, отказ в подключении к БД). Также через контекстное меню панели процессов можно останавливать выполнение (прерывать процесс). Для больших проектов с вложенными подмоделями структура процессов помогает понять, на каком этапе сейчас находится расчёт, так как показывает иерархию: процесс > подпроцесс > узел ¹⁰⁹. Например, «Процесс 1 – Сценарий: Скоринг (узлы...)» – Подпроцесс: Обучение модели – узел Регрессия...».

Итого, **структура панелей Loginom** выглядит так: слева – главное меню и инструменты, сверху – навигация по вкладкам, центр – конструирование сценариев и результаты, снизу – состояние процессов. Такое разделение облегчает работу: каждый участок экрана отвечает за свою задачу, и в комплексе они позволяют эффективно строить, запускать и контролировать анализ данных. При анализе данных аналитик обычно циклически: выбирает компонент (левая панель), настраивает его (центральная область), запускает выполнение (кнопка запуска на панели инструментов или контекстно) и смотрит на нижнюю панель, затем просматривает результат в центральной – и так далее, пока не будет готов весь проект.

3. Анализ назначения и использования всех узлов Loginom.

Платформа Loginom оперирует понятием *узел* (node) – это блок сценария, выполняющий определённую операцию над данными. Существует множество видов узлов, и чтобы охватить «все», их удобно разделить на категории по назначению. Рассмотрим основные группы узлов и как они используются:

- **Узлы источников данных (импорт):** Предназначены для загрузки данных в сценарий из внешних источников. Примеры: «Импорт из CSV/Excel», «Импорт из базы данных (SQL запрос)», «Импорт JSON/XML», «Импорт из OLAP/1C», узлы подключения к конкретным системам (SAP, Salesforce – при наличии). Назначение: взять данные «снаружи» и преобразовать в табличный формат внутри Loginom. Использование: обычно первым шагом сценария стоит узел импорта

– его настраивают (например, указав путь к файлу, или соединение к БД и текст запроса), выполняют – и на выходе узла получается таблица данных для дальнейшей обработки. Эти узлы могут иметь различные порты (например, некоторые – порт подключения). В одном сценарии можно использовать несколько источников, если нужно объединять данные из разных мест.

• **Узлы подготовки и преобразования данных:** Самая обширная группа – узлы, которые изменяют, фильтруют, объединяют и агрегируют данные. Сюда входят:

- *Фильтр* – отбор строк по заданному условию (например, оставить только клиентов старше 30 лет).
- *Сортировка* – упорядочение строк по одному или нескольким полям.
- *Объединение (Join)* – соединение двух потоков данных по ключам (аналог SQL JOIN): Inner, Left, Right, Full joins – объединяет таблицы, добавляя поля.
- *Объединение последовательное (Union, Append)* – склеивание таблиц одинаковой структуры друг под другом (например, объединить данные из двух филиалов).
- *Группировка (Aggregation)* – группировка записей по указанным полям с вычислением агрегатных функций (суммы, средние и т.п.) ¹¹⁰. Например, посчитать сумму продаж по каждому региону.
- *Расчёт поля* – добавление нового поля, вычисленного по формуле из других полей (например, вычислить прибыль = выручка - себестоимость). Пользователь задаёт выражение, Loginom применяет к каждой строке.
- *Транспонирование, Pivot (Озеро данных)* – преобразование строк в столбцы или наоборот (широкий формат в длинный и обратно). Например, поворот таблицы для построения OLAP-куба.
- *Очистка данных:* отдельные узлы для разных аспектов: «Заполнение пропусков», «Удаление выбросов», «Выявление дубликатов», «Выравнивание временных рядов», «Нормализация» (приведение чисел к единому масштабу) и др. Их назначение – улучшить качество данных.
- *Соединение с эталонными данными:* например, «Присоединение справочника» – подтянуть к основным данным расшифровки кодов (коды регионов заменить на названия через справочник).

Использование: эти узлы строят последовательность очистки и преобразований. Например, после импорта данных о продажах можно поставить узел «Заполнение пропусков» (чтобы заменить недостающие цены на средние), затем «Фильтр» (убрать нулевые продажи), потом «Группировка» (суммировать продажи по товарным категориям) и т.д. Каждый узел берёт входной поток (таблицу), выполняет свое преобразование, выдаёт новый поток. Пользователь настраивает параметры узла (через мастер настройки – например, для фильтра задаёт условие), а Loginom выполняет эти шаги. Визуально такой сценарий выглядит как цепочка узлов, соединённых стрелками. Эти подготовительные узлы часто используются в *комбинации*: реальный сценарий может иметь разветвление (например, один поток данных фильтруется по году 2021, другой по 2022, затем объединяются). Благодаря богатой библиотеке узлов подготовки, можно реализацию почти любого ETL-процесса сделать внутри Loginom.

• **Узлы анализа данных и моделирования:** Это узлы, реализующие алгоритмы Data Mining, статистики и машинного обучения. Они обычно более «тяжёлые» и могут требовать обучения (для контролируемого обучения). Основные:

- *Регрессия*: узел «Линейная регрессия» – строит модель линейной зависимости целевой переменной от факторов. На вход подаётся обучающая выборка, узел настраивается (выбирается целевое поле, независимые поля), по выполнении внутри узла хранится модель. Этот же узел может выдавать на выходе рассчитанные коэффициенты, статистику модели или прогнозы (если на вход подать новые данные). Также есть «Логистическая регрессия» для бинарной классификации.
- *Дерево решений*: узел, который обучает решающее дерево (например, для классификации клиентов на купит/не купит). Настройки: целевой признак, критерий разбиения, максимальная глубина. Результат – дерево, которое можно визуализировать.
- *Кластеризация*: узел «Кластеризация (K-means)» – разбивает данные на указанное число кластеров. Нужно настроить: список полей для расчёта расстояний, число кластеров (если K-means), либо диапазон (если G-means автоподбор). После выполнения узел выдаёт для каждой строки номер кластера, а также может выдавать таблицу центроидов.
- *Нейронная сеть*: узел «Нейронная сеть» (например, многослойный персептрон) – позволяет настроить архитектуру (сколько нейронов в скрытом слое), функцию активации, число эпох обучения. После обучения может применяться для прогнозирования.
- *C- Методы классификации*: кроме деревьев и логистической регрессии, Loginom поддерживает, например, метод опорных векторов (SVM) и k-ближайших соседей (k-NN)* для задач классификации ⁷⁷. Их узлы настраиваются параметрами (тип ядра для SVM, число соседей для k-NN и т.д.).
- *Факторный анализ*: узел «Факторный анализ (PCA)» – уменьшает размерность данных, генерируя главные компоненты (см. ранее). Пользователь указывает, какие поля анализировать и сколько факторов оставить (или долю дисперсии), на выходе получает преобразованную таблицу с новыми признаками-факторами.
- *Ассоциативные правила*: узел «Поиск ассоциативных правил» – принимает на вход транзакционные данные (например, покупки по чекам) и выдаёт список правил вида $X \Rightarrow Y$ с вычисленной поддержкой и уверенностью (см. вопрос 8 выше). Настройки – минимальная поддержка, достоверность и прочие ограничения.
- *Анализ временных рядов*: имеются узлы для сглаживания, сезонной декомпозиции, некоторые модели прогнозирования (например, простые экспоненциальное сглаживание). Сложные модели (ARIMA, ETS) можно реализовать через R-скрипт узел, если требуется.

Использование аналитических узлов обычно двухфазное: **обучение** модели и **применение**. Многие узлы (дерево, нейросеть, регрессия) сначала конфигурируются и обучаются на исторических данных – при этом внутри узла сохраняется модель. Затем этот же узел или специальный узел «Применить модель» используется, чтобы сделать прогнозы на новых данных. В Loginom это реализовано удобно: после выполнения узла с обученной моделью можно подключить к нему новые данные (вход порт «новые данные») – на выходе узел выдаст результат (прогноз, метки классов). Кроме того, модель можно сохранить как компонент и reuse в другом месте. Так, аналитические узлы решают задачи Data Mining end-to-end внутри сценария.

- **Узлы визуализации и вывода результатов**: К этой категории относятся компоненты, предназначенные для представления результатов анализа и экспорта данных во внешние форматы:
- *Таблица (Viewer)*: узел, позволяющий отобразить данные в табличном виде с возможностью сортировки, фильтрации прямо в интерфейсе. В режиме разработки чаще используют контекстный «быстрый просмотр», но можно встроить таблицу как финальный объект на дашборде.

- **Диаграммы и графики:** Logiplot имеет узлы для построения различных графиков – гистограммы, линейные графики, круговые диаграммы, столбчатые, коробчатые (box plot) и пр. Настроив такой узел, аналитик определяет, какие данные по какой оси откладывать, категории, цвета, и на выходе получает визуальный элемент. Эти узлы используются при создании итоговых **дашбордов** или отчетов: например, отобразить динамику продаж по месяцам (линейный график) или структуру затрат (круговая диаграмма).
- **OLAP-куб:** особый узел, который агрегирует данные по нескольким измерениям и позволяет затем интерактивно их «крутить». Пользователь задаёт меры (числовые показатели, например сумма продаж) и измерения (например, по годам, по регионам, по продуктам), узел строит многомерный куб. В интерфейсе затем можно разворачивать измерения по осям таблицы, проваливаться (drill-down) в детали. OLAP-куб – мощный способ визуализировать данные в разрезах (подробнее см. следующий вопрос). В Logiplot результаты куба могут быть представлены в виде сводной таблицы для конечного пользователя.
- **Экспорт данных:** узлы, позволяющие вывести результаты наружу: «Экспорт в Excel/CSV», «Экспорт в базу данных», «Экспорт в HTML/PDF отчет». Они берут на вход итоговую таблицу или набор графиков и записывают в указанный файл или систему. Например, после анализа можно выгрузить очищенные данные обратно в хранилище или сформировать Excel-отчет для руководства.
- **Отчет (Dashboard):** скорее, не один узел, а возможность собрать несколько визуальных элементов (таблицы, диаграммы) на одном листе с текстовыми комментариями, логотипами и т.п. Logiplot позволяет проектировать такие отчеты в режиме конструктора и затем обновлять их по данным. По сути, это итог презентации результатов анализа.

Использование узлов вывода обычно завершает сценарий. Например, после всех расчётов ставится узел «Экспорт в Excel» – и при запуске сценария получаем на диске файл отчета. Либо на веб-портале Logiplot конечный пользователь заходит и видит интерактивный отчет с фильтрами и графиками, подготовленный с помощью соответствующих визуализаторов.

- **Узлы управления потоком и логики сценария:** Кроме узлов, непосредственно работающих с данными, Logiplot предоставляет компоненты для организации сложных сценариев:
- **Подмодель:** узел, внутри которого можно вложить другие узлы (то есть целый подпроцесс). Подмодель позволяет *инкапсулировать* часть логики – например, сложную комбинацию очистки данных – в один узел, чтобы не загромождать основную схему. Подмодель имеет свои входные и выходные порты (их можно настроить), и ее содержимое выполняется как единое целое. Используется для структурирования больших проектов и повторного использования кода (одну подмодель можно вставлять в разные места сценария).
- **Цикл:** узел, который повторно выполняет вложенную подмодель для набора значений. Например, цикл по списку файлов – будет выполнять подмодель импорта и обработки для каждого файла из папки. Или цикл по значению переменной – позволяет реализовать поочередную обработку разных групп данных. По сути, «Цикл» содержит внутри себя подмодель (как тело цикла) ¹¹¹. В Logiplot цикл удобен для автоматизации однотипных операций.
- **Условный узел (ветвление):** компонент, который реализует условие «если-то». Он имеет два выходных порта – для случаев, когда условие выполняется (True) и не выполняется (False). В зависимости от заданного логического выражения будет активирован тот или иной последующий путь сценария. Это позволяет делать разветвления – например, если данных меньше определенного объема, применить упрощенный метод, иначе – основной.

- *Переменные и скрипты:* есть узлы для задания **управляющих переменных** (параметров сценария, которые можно менять без правки узлов) и для выполнения скриптов на внешних языках. Узел «Выполнить R-скрипт» или «Python» позволяет вписать код, который выполнится над входными данными и выдаст результат. Это нужно, когда встроенных компонентов недостаточно или для очень специфических вычислений. Также можно использовать SQL-запросы внутри узла, если это проще для определенной трансформации.
- *Логирование и отладка:* узлы, которые пишут служебную информацию (например, «Записать в лог»), измеряют время выполнения (таймер) и т.п. – помогают отслеживать работу сценария.

Эти управляющие узлы делают сценарии гибкими и адаптивными. С их помощью Logiplot поддерживает не просто линейный поток, а **полноценный алгоритм** с циклами, условиями, разбиением на подзадачи. Например, можно настроить сценарий так, что он проходит по всем таблицам в базе (цикл), для каждой выполняет анализ, и в зависимости от результата (условие) либо отправляет уведомление, либо переходит к следующей.

Подводя итог, **«все узлы Logiplot»** – это богатый каталог инструментов для работы с данными. Их назначение разнообразно: от простых трансформаций до сложного моделирования. Использование узлов интуитивно: аналитик **визуально собирает цепочку** из этих блоков, настраивая каждый под свои нужды. Понимание назначения каждой категории узлов позволяет эффективно строить сценарии анализа: от загрузки сырья, через очистку и анализ, к выдаче инсайтов – на каждом шаге есть подходящий инструмент. В совокупности узлы Logiplot покрывают практически любые задачи подготовки и анализа данных, что делает платформу универсальным решением для аналитиков.

4. Типы портов узлов Logiplot и варианты их использования.

Каждый узел в Logiplot имеет специальные точки подключения – *порты*, через которые узлы обмениваются данными или сигналами. Существуют разные типы портов, и важно понимать их назначение:

- **Входные и выходные порты данных (табличные порты):** Это основные порты, обозначенные обычно значком таблицы или стрелками. **Входной порт данных** принимает набор данных (таблицу) от предыдущего узла, **выходной порт данных** передает результат обработки дальше по сценарию. Большинство узлов подготовки и анализа имеют хотя бы один входной порт (например, узел «Фильтр» – один вход данных) и один выходной. Некоторые узлы имеют несколько входных: например, узел «Объединение (Join)» имеет два входных порта – для левой и правой таблицы, которые он объединяет; узел «Ассоциативные правила» может иметь отдельный вход для транзакций и вход для справочников. Соответственно, выходных портов тоже может быть несколько: узел «Разделение набора» может один вход разделить на два выхода (например, обучающая и тестовая выборки). **Использование:** данные порты служат для основного потока данных – их соединяют линиями между узлами, выстраивая цепочку обработки. При соединении действует правило: *соединять можно только порты совместимых типов*. Обычно это означает, что выход таблицы можно подключить ко входу таблицы. Нельзя, например, напрямую соединить выход узла с самим собой (циклическая связь запрещена). Если узел не получает нужный вход (не подключен) – при выполнении будет ошибка или узел будет ждать данных. Состояние порта отображается цветом (например, серый – нет данных, зелёный – данные готовы). В практике построения сценария пользователь соединяет выход *табличного* порта одного узла со входом другого простым перетаскиванием мыши – появится линия связи. Это задаёт последовательность передачи наборов данных по сценарию.

- **Порты порядка выполнения (управляющие порты, flow ports):** Особый тип портов, отвечающих не за передачу данных, а за **синхронизацию и последовательность** выполнения узлов. Они могут быть не видны по умолчанию (в Loginom скрыты для упрощения, но их можно отобразить). Управляющие порты часто обозначаются треугольными метками (например, на узлах подмоделей или циклов). Для чего они нужны: допустим, два узла не связаны потоком данных (не передают таблицы друг другу), но мы хотим, чтобы один выполнялся строго после другого. Например, сначала выполнить узел «Удалить старые данные из БД», а затем узел «Загрузить новые данные» – между ними может не быть таблицы, но нужно задать порядок. Вот тут и используют порт порядка выполнения: соединяют выход «по завершении» первого узла со входом «запуск» второго. Тогда Loginom знает, что сначала должен отработать первый, затем второй, хоть они и не связаны данными. В сложных сценариях с параллельными ветками такие порты помогают координировать потоки (что-то вроде блок-схемы, где есть стрелки перехода). **Использование:** управлять портами обычно пользуются при необходимости – их надо включить (через кнопку на панели инструментов или в меню узла «Показать порты порядка выполнения»). После этого появятся треугольные порты, которые можно соединять. Например, у узла «Экспорт» часто имеет смысл дождаться выполнения всех вычислительных узлов – можно соединить от них управляющие выходы к экспорту, чтобы он стартовал только когда всё готово.
- **Порты управляющих переменных (параметрические порты):** Ещё один вид портов, тоже скрытых по умолчанию. Они служат для передачи **значений параметров или переменных** в узлы. Например, можно завести переменную «Порог» и подключить её к узлу «Фильтр» – тогда вместо фиксированного числа в условии фильтра узел возьмёт текущее значение из переменной. Управляющие переменные могут поступать извне сценария (например, пользователь на дашборде выбрал значение – оно передалось в сценарий через порт) или рассчитываться другим узлом. По сути, это способ гибко менять поведение узла без редизайна схемы. **Использование:** чтобы воспользоваться, надо в контекстном меню узла выбрать «Показать порт управляющих переменных». Появится порт, обычно отличающийся цветом. Его можно соединить с узлом, который генерирует переменную (скажем, узел «Задать переменную»). Например, в сценарии сегментации клиента можно сделать переменную «Возрастной_порог» – задать ей значение 30 узлом «Задать var», и подключить к фильтру: `Возраст > $Возрастной_порог`. Теперь легко изменить порог, просто изменив значение переменной (например, вынести на интерфейс). Без таких портов пришлось бы редактировать узел. Также управляющие порты применяются для динамической настройки узлов: например, узел SQL-запроса может принимать на вход переменную с текстом запроса или параметром, что позволяет менять запрос на лету.
- **Специальные порты:** Некоторые узлы имеют специфические порты. Например, узел «Обучение модели» может иметь порт для входа *обучающих данных* и отдельный порт для входа *контрольных (тестовых) данных* – для сразу оценки модели. Узел «Ассоциативные правила» – порт для входа основных транзакций и порт для «элементов справочников» (чтобы добавить описание продуктов, которые фигурируют в правилах). Подмодель-узел может иметь несколько **входов и выходов**, настроенных пользователем (например, подмодель, которая принимает две таблицы – продажи и затраты – и возвращает таблицу прибыли и какой-то показатель). Эти порты пользователь задаёт в настройках подмодели как публичные. Их использование аналогично – нужно соединить с соответствующими данными/узлами вне подмодели.

В целом, **тип порта определяет, что по нему передается**: данные, сигнал порядка или значение переменной. *Варианты использования портов*: - **Последовательная цепочка (data ports)**: стандартный случай – выходные данные одного шага идут на вход следующего. - **Параллель с синхронизацией (flow ports)**: например, два независимых расчёта выполняются параллельно, а потом управляющие выходы обоих сходятся на вход узла экспорта – тем самым экспорт ждёт завершения обоих расчетов. - **Параметризация (variable ports)**: настройка сценария извне – допустим, через дашборд передаём дату, и эта дата по управляющему порту входит в узел фильтра, ограничивая выборку.

Важно помнить, что по **табличным портам** нельзя передать управляющие сигналы, а по управляющим – данные: Loginom контролирует типы соединений. Среда сама окрашивает порты (например, синий – таблица, оранжевый – переменная, чёрный треугольник – последовательность) и подсказывает, к какому типу порта можно присоединиться при протягивании связи. Если попытаться соединить несовместимые – связь не создастся.

Знание типов портов и умение их использовать позволяют создавать гибкие сценарии: например, организовать **итерации** и условия (за счёт управляющих портов) и **адаптивные алгоритмы** (за счёт переменных). В простых случаях можно обходиться только табличными портами – Loginom сам выполняет узлы последовательно по наличию данных. Но для сложных orkflow-ов эти механизмы незаменимы, обеспечивая корректный порядок выполнения и *параметризацию* сценариев.

5. Варианты настройки узлов.

Каждый узел в Loginom обладает параметрами, которые нужно настроить под конкретную задачу. Существуют различные способы и уровни настройки узлов: от использования мастеров до ручного ввода формул. Основные варианты настройки:

- **Мастер настройки узла (Wizard)**: При двойном клике на узел или выборе опции «Настроить...» открывается диалоговое окно мастера. Мастер интерактивно проводит через шаги конфигурации. Например, для узла «Импорт файла» мастер сначала спрашивает путь к файлу, потом предлагает настроить формат (разделитель, кодировку), отобразит образец данных, далее шаг – выбор листа Excel или диапазона, и т.д. Для узла «Регрессия» мастер первым шагом попросит выбрать целевую переменную и входные, следующим – параметры обучения (нормализация, разбиение на обуч./тест), и т.д. Мастер – основной способ настройки, особенно для сложных узлов, так как он структурирует процесс и обеспечивает ввод всех необходимых параметров. Пользователю не нужно помнить все свойства – мастер подскажет последовательность. В некоторых узлах (фильтр, расчёт поля) мастер фактически представляет собой окно с формулой и подсказками по синтаксису.
- **Настройка через свойства (табличный интерфейс)**: Для ряда узлов настройка может осуществляться в табличной форме. Например, узел «Соединение (Join)» – настройка соответствия полей может быть через таблицу: в левой колонке поля первой таблицы, в правой – поля второй, и выпадающим списком выбирается, какие сопоставить. Узел «Сортировка» при настройке показывает две панели: слева список всех полей, справа – порядок сортировки, куда переносишь нужные поля и задаёшь возр/убывание. Это тоже разновидность визуальной настройки – drag-and-drop в таблице параметров. В узле «Группировка» например одна панель – выбор группировочных полей, другая – выбор агрегатных операций для остальных полей. Такой подход удобен, когда нужно настроить

соответствия или списки – вместо вписывания текста, пользователь кликами формирует конфигурацию.

- **Графический интерфейс сопоставления полей («Связи»):** В случаях, когда надо связать поля входа и выхода, Loginom предлагает графический режим. Например, при настройке порта, если входная структура известна, можно выбирать соответствие входных и выходных полей либо в режиме «Таблица», либо в режиме «Связи» ¹¹². В режиме «Связи» отображаются два столбца с именами полей, и пользователь мышью соединяет соответствующие поля. Это наглядно при перетаскивании – особенно полезно, если поля имеют разные названия или когда создаются новые вычисленные поля на основе входных. Это помогает избежать ошибок в сопоставлении. Аналогично может настраиваться, скажем, узел «Расчет показателей» – графически связать, какие входы преобразуются в какие выходы.
- **Использование формул и выражений:** Некоторые узлы требуют написания выражений (синтаксис Loginom или встроенного языка). Например, в узле «Фильтр» условие пишется как логическое выражение (e.g., `Возраст > 30 AND Доход < 50000`). В узле «Вычислить поле» – формула для нового поля (e.g., `Прибыль = Выручка - Себестоимость`). Loginom предоставляет редактор формул с подсветкой синтаксиса, автодополнением имен полей и функций. Можно использовать богатую библиотеку встроенных функций (математических, текстовых, дат и т.д.). Таким образом, настройка осуществляется путем написания необходимого выражения. Это вариант *ручной настройки*, требующий знания синтаксиса, но дающий большую гибкость – можно задать произвольную логику, которую не всегда можно вставить выпадающими списками. В мастере узла обычно для этого выделено поле ввода формулы.
- **Настройка через скрипты:** В ситуациях, когда встроенные средства не покрывают задачу, узел может быть настроен путем включения скрипта. Например, узел «SQL запрос» – по сути, текст SQL является настройкой узла (пишется `SELECT ...`). Узел «R-скрипт» или «Python» – пользователь настраивает узел, вводя код на соответствующем языке. Это самый продвинутый способ настройки, требующий навыков программирования, но дающий максимальную свободу. Практически, это встраивание своего алгоритма в узел. Ещё пример: узел «Вывод HTML» может позволять встраивать HTML/JavaScript код в отчет, что тоже можно считать настройкой узла путем редактирования шаблона кода.
- **Использование шаблонов и повторного использования настроек:** Loginom позволяет копировать узлы или сохранять настроенные узлы как шаблоны (компоненты) для повторного использования. Это значит, что единожды настроив сложный узел, можно его дублировать. Например, настроили сложное вычисление – можно скопировать узел, вставить вновь – настройки сохранятся. Также можно экспортировать узел/подмодель настроенную и потом импортировать в другой проект – фактически, reuse. Это не вариант настройки с нуля, но вариант *применения ранее выполненной настройки*.
- **Пакетная настройка через порт переменных:** Если в проекте введены управляющие переменные, часть узлов может получать настройки не вручную, а извне. Например, вместо жесткого значения порога в узле, мы привязываем переменную `X`. Тогда настройка узла сводится к указанию «брать значение из переменной X» – а само значение устанавливается в другом месте. Это тоже вариант: *параметрическая настройка*. Она удобна тем, что чтобы изменить поведение нескольких узлов, можно поменять одну переменную.

В пользовательском интерфейсе Loginom реализована контекстная проверка корректности настроек. Если узел настроен не полностью или с ошибкой (например, не выбрано ключевое поле для соединения, или синтаксическая ошибка в формуле), узел подсвечивается красным, а мастер

указывает на проблему. Так что процесс настройки итеративный: ввёл параметры – проверил – исправил, если нужно.

Например, **настройка узла «Регрессия»**: первый шаг – выбираем входные и выходное поля из списка (таблично, галочками), второй – отмечаем опции (стандартизовать ли переменные, разбить ли на обуч./тест с указанием процента), третий – смотрим сводку или дополнительные настройки (критерии, метрики). После этого узел готов. Другой пример: **узел «Сортировка»** – при настройке появляется окно с двумя панелями: слева все поля, справа – пока пусто. Пользователь выделяет слева поле «Дата» и нажимает кнопку «Добавить» (или двойной клик) – поле появляется справа, можно выбрать порядок «Возрастание». Затем добавляет «Регион» – он добавится вторым уровнем сортировки. Нажимает ОК – узел настроен (будет сортировать сначала по дате, внутри даты по региону).

Таким образом, **варианты настройки узлов** охватывают: интерактивные мастера, таблично-графические сопоставления, ввод формул, написание скриптов и использование переменных. Какой вариант применяется – зависит от конкретного узла и предпочтений пользователя. Loginom стремится обеспечить удобные мастера для всех типовых настроек, сводя ручной код к минимуму. Однако, опытный пользователь всегда может переключиться на ручной режим (ввести выражение, написать скрипт) для реализации нестандартной логики. В итоге, система настройки узлов гибкая: от «пошагового диалога» до «сделай всё сам». Это позволяет начинающим быстро освоиться (благодаря мастерам), а профессионалам – не быть ограниченными рамками и кастомизировать поведение узлов вплоть до кода.

6. Возможности визуализации результатов анализа. Понятие и использование OLAP-кубов.

- **Возможности визуализации в Loginom**: Платформа предоставляет разнообразные способы представить результаты анализа в понятной и наглядной форме:

- *Табличные представления*: любой набор данных можно отобразить в виде таблицы, настроить формат отображения чисел, подсветку определённых значений (условное форматирование). Это базовый, но важный способ – иногда результаты (например, список аномальных клиентов) целесообразно представить списком.

- *Диаграммы и графики*: Loginom поддерживает построение стандартных диаграмм: линейные графики (динамика во времени), столбчатые (сравнение категорий), круговые (структура процентов), точечные (распределение), пузырьковые, гистограммы (распределение частот), ящиков с усами (распределение и выбросы) и другие. Пользователь может настраивать цвета, легенды, подписи осей. Графики динамически связаны с данными – при обновлении данных график перестраивается. Есть и специфичные визуализации, например дендрограммы для кластеров, деревья решений (отрисовываются структурой), тепловые карты.

- *Комбинированные дашборды*: результаты разных узлов можно вывести совместно на интерактивный дашборд. Например, на одном экране: фильтр по дате (выбор периода), график продаж по месяцам, таблица топ-10 товаров и пирог долей по регионам. Эти элементы могут быть связаны: выбирая точку на графике, можно настроить фильтрацию таблицы и диаграммы. Loginom позволяет создавать такие дашборды для пользователей, которые не будут запускать сценарий вручную, а просто просматривать актуальную информацию.

- *Отчётность для печати*: встроенный генератор отчётов даёт возможность подготовить макет отчёта, включающий текст, изображения (логотипы), таблицы и графики, и сгенерировать его в PDF или Excel. Это важно для финальной документации: например, ежемесячный отчёт о продажах – можно автоматизировать его выпуск на основе шаблона, и Loginom сам подставит новые данные и

сохранит PDF.

- *Средства взаимодействия*: визуализация результатов – это не только статичные графики. Loginom поддерживает OLAP-анализ (см. ниже), который можно считать видом визуализации (сводные таблицы с возможностью детализации). Также возможна интеграция с BI-инструментами – например, подготовить агрегированные данные в Loginom и передать их в Tableau/PowerBI для визуализации – но чаще всего возможностей самой платформы достаточно.

- *Визуализация в процессе анализа*: Отметим, что Loginom предоставляет визуализацию не только на финальном этапе, но и для промежуточного анализа: есть **визуальные узлы** типа Scatter Plot, которые можно встроить в сценарий, чтобы изучить данные (хотя они и не нужны в финальном отчёте). Это помогает аналитику видеть распределения, взаимосвязи на этапе EDA.

Итого, возможности визуализации широки – от простых таблиц до интерактивных панелей. Все графики и таблицы можно настраивать под задачи (подписи, цвета, сортировка), что позволяет привести их к **презентабельному виду для принятия решений**.

- **Понятие OLAP-куба**: OLAP (On-Line Analytical Processing) куб – это многомерная структура данных, предназначенная для быстрого вычисления агрегированных показателей по различным измерениям. Проще говоря, OLAP-куб позволяет **сводить большие таблицы в удобные разрезы**: имеются числовые показатели (*меры*, например, сумма продаж, количество клиентов) и несколько категорий или атрибутов (*измерения*: время, регион, продукт, канал продажи и т.д.), по которым эти показатели можно группировать. Кубом называют многомерный массив, где измерения – оси массива, а ячейки содержат агрегаты (суммы, средние). Преимущество OLAP-куба в том, что он предрассчитан – агрегаты вычислены заранее (или быстро считаются при первом запросе и кешируются), поэтому пользователь в режиме реального времени может менять срезы и получать ответ практически мгновенно ¹¹³. OLAP позволяет делать **срезы (slice)** – фиксировать одно измерение на определённом значении (например, только 2025 год) и смотреть куб для этого среза, **dice (подвыборка)** – взять подмножество по нескольким измерениям (например, продажи по двум регионам за Q1), **drill-down/roll-up** – проваливаться на более детальный уровень или сворачивать до более крупного. Например, измерение "Дата" можно иметь уровни Год > Квартал > Месяц – roll-up суммирует месяцы в квартал, drill-down раскрывает год на кварталы. Таким образом, OLAP-куб – мощный инструмент для **оперативного анализа агрегированных данных**, очень удобный для конечных пользователей (финансистов, менеджеров), т.к. не требует знаний SQL – все действия выполняются интерактивно через таблицы и графические контролы.

- **Использование OLAP-кубов в Loginom**: Loginom включает функциональность OLAP. Это проявляется, во-первых, как одна из опций при визуализации: есть компонент «OLAP-куб/Сводная таблица». Аналитик на этапе построения отчёта определяет, какие поля будут измерениями, какие показатели агрегировать, и строит куб. Во-вторых, Loginom может подключаться к внешним OLAP-хранилищам (например, Microsoft SSAS, если настроен коннектор) – но чаще используется собственный механизм.

В Loginom OLAP-куб представляется пользователю как интерактивная **сводная таблица**: по строкам и столбцам располагаются значения измерений, внутри – агрегаты. Пользователь (либо аналитик в режиме разработки, либо конечный пользователь, если ему дать интерактивный доступ) может перетаскивать поля измерений между осями «Строки», «Колонки» и «Фильтры» – тем самым перестраивая отчет. Например, изначально куб

показывает *Продажи по Регионам* (строки) за *Годы* (колонки). Пользователь может перетащить измерение «Продуктовая категория» в строки – теперь увидит разбивку продаж по регионам и категориям. Или убрать «Год» в фильтры, а вынести «Квартал» в колонки – более детально по времени. Все эти операции происходят **онлайн**, без ожидания минутных пересчетов – система либо уже рассчитала нужные агрегаты, либо мгновенно суммирует из исходных (в случае небольших данных). Лекции отмечают, что OLAP дает возможность пользователям оперативно получать агрегированную информацию на базе больших массивов данных в реальном времени.

Для аналитика настройка OLAP-куба в Loginom состоит в указании: 1) *Источника данных* – таблицы с детальными записями (факты). 2) *Мер* – какие числовые поля агрегировать (сумма, счет, максимум и пр.). Например, «Сумма продажи (SUM Sales)», «Кол-во транзакций (COUNT)». 3) *Измерений* – какие поля выступают измерениями; возможно указать иерархии (например, из даты выделить Год-Квартал-Месяц). После этого Loginom строит внутреннюю структуру куба. Можно настроить дефолтное представление (какие измерения сразу на каких осях). Затем при выполнении сценария куб будет доступен либо в интерфейсе Loginom (в режиме дизайна/отчета), либо даже можно опубликовать его в веб-интерфейсе для бизнес-пользователей. Те смогут взаимодействовать с кубом: фильтровать значения (например, выбрать конкретный регион), разворачивать и сворачивать иерархии (drill), менять группировки. OLAP-куб также часто позволяет строить над собой графики: например, выделив ряд или колонку сводной таблицы, построить график тенденции.

Использование OLAP-кубов особенно оправдано, когда данные имеют *много измерений и требуется гибкий анализ*. Вместо того чтобы заранее предугадать все разрезы и делать десятки отчетов, аналитик выдает один куб – и пользователи сами формируют нужные им отчеты «на лету». Например, отдел продаж может сам посмотреть продажи *по продуктам в разрезе кварталов по каждому региону за интересующий год*, просто перетащив поля. Без OLAP это был бы отдельный запрос или отчет.

Подчеркнем, OLAP-куб эффективен на **агрегированных запросах**. Если нужно увидеть конкретные сделки – это не к кубу, а к исходным транзакциям (подробный отчет). Но чтобы проанализировать итоги, тенденции, доли – куб идеален.

В Loginom OLAP реализован как один из компонентов аналитической платформы (в перечне возможностей он прямо указан). Под капотом, вероятно, используется гибрид: для небольших наборов – расчет на лету, для больших – может агрегировать. Для пользователя это прозрачно.

В итоге, **визуализация в Loginom** позволяет превращать сухие цифры в понятные графики и таблицы, а **OLAP-кубы** дают мощный интерактивный инструмент «*поворота данных*» для детального изучения результатов. В связке они обеспечивают, что результаты анализа не останутся в виде сложных моделей, а будут представлены так, чтобы менеджмент мог принять решения, основываясь на них.

7. Возможности инкапсуляции элементов сценария анализа данных.

Инкапсуляция в контексте Loginom означает способность объединять группу узлов сценария в один логический блок, скрывая внутреннюю сложность и, при желании, повторно используя этот блок. Это

важно для организации крупных проектов, повышения наглядности и модульности. Возможности инкапсуляции включают:

- **Подмодели (сабмодели):** Главный механизм инкапсуляции – узел «Подмодель». Подмодель – это контейнер, внутри которого можно разместить произвольный фрагмент сценария (набор узлов и связей между ними), как бы вложенный под-сценарий ¹¹¹. Внешне подмодель выглядит как единый узел, у которого можно настроить входные и выходные порты для передачи данных внутрь и наружу. Все детали реализации скрыты внутри – чтобы их посмотреть или изменить, нужно «войти» в подмодель (двойным кликом или через навигацию). Инкапсуляция здесь проявляется в том, что на верхнем уровне схемы вместо десятка узлов мы видим один узел-подмодель с понятным названием, например «Очистка данных» или «Расчет показателей». Это значительно улучшает читаемость сложных сценариев. **Использование:** Аналитик может сгруппировать несколько этапов обработки, выделив их в подмодель. Для этого в LogiNot выделяются нужные узлы -> команда "Объединить в подмодель". Система создаст новый узел, перенесёт внутрь него выделенные компоненты, а на внешнем уровне останутся только порты связи. Затем можно переименовать подмодель по смыслу. Теперь, когда нужно понять общий ход сценария, не приходится разбираться в деталях этих этапов – название подмодели говорит само за себя. При необходимости, конечно, можно раскрыть и посмотреть, что внутри (инкапсуляция не препятствует доступу разработчику, но скрывает от поверхностного обзора).
- **Иерархия подмоделей:** В LogiNot подмодели могут быть вложенными (подмодель внутри подмодели). Это позволяет строить многоуровневую декомпозицию алгоритма. Например, на первом уровне подмодель «Моделирование», внутри неё подмодель «Кластеризация клиентов», внутри которой, например, подмодель «Стандартизация показателей». Такая иерархия способствует повторному использованию логики и изоляции изменений: если поменять что-то внутри, внешний интерфейс подмодели (порты) может остаться тем же, и остальная схема не затронется.
- **Параметризация подмоделей (инкапсуляция логики с параметрами):** При создании подмодели разработчик может определить **порты управляющих переменных** для неё – т.е. сделать подмодель, зависящую от внешнего параметра. Например, можно создать подмодель «Отбор аномалий» с параметром «Порог_сигмы» (сколько сигм считать выбросом). Внутри подмодели этот параметр используется в формуле фильтра. Внешне подмодель имеет порт для приема значения порога. Теперь её можно использовать в разных проектах: подключая разные значения порога. Это способ инкапсулировать алгоритм, но оставив ему гибкость настройки через параметры.
- **Библиотеки и повторное использование модулей:** LogiNot позволяет сохранять подмодели (и даже целые сценарии) в виде *пакетов* или компонентов, которые затем можно импортировать в другие проекты. Таким образом, команды или отдельные аналитики могут наработать библиотеку стандартных решений. Например, однажды настроенную подмодель «Очистка контактных данных» (которая вычищает телефонные номера, e-mail, дубли) можно сохранить и потом применять во всех проектах, где нужны те же шаги, вместо того чтобы заново строить каждый раз. Это и есть инкапсуляция – внутренняя реализация упакована, а наружу предъявляется готовый блок, который достаточно подключить и передать ему данные.
- **Инкапсуляция через циклы и условия:** Узлы «Цикл» и «Если», по сути, тоже дают элемент инкапсуляции – они содержат внутри себя подмодель (тело цикла или ветвление). Например, узел «Цикл по значению поля» инкапсулирует логику повторного выполнения: внутри у него

есть подмодель, которую он выполняет для каждого уникального значения указанного поля. Внешне видно только сам узел цикла, а детали (какие узлы выполняются на каждой итерации) спрятаны внутри подмодели цикла. Это пример *шаблона проектирования*: цикл – как обёртка над подмоделью. Для условий аналогично: узел «Если» содержит две внутренние подмодели – одна для ветки TRUE, другая для FALSE. Снаружи же видно просто один узел условного оператора. Это и повышает наглядность (на схеме один значок вместо двух сценариев), и группирует связанную логику, и облегчает модификацию (изменения внутри ветвей не влияют на остальную схему).

- **Инкапсуляция и безопасность:** В многопользовательском режиме инкапсуляция полезна тем, что можно скрыть от определённых пользователей внутренности узла, дав им доступ лишь к интерфейсу (портам) компонента. Например, внешнему консультанту можно дать готовый компонент прогноза, не раскрывая ноу-хау, как именно он реализован внутри. Он сможет его использовать как чёрный ящик. Loginom поддерживает разграничение доступа на уровне узлов, так что это возможно.

Пример: Допустим, имеется сценарий, прогнозирующий отток клиентов. Он включает: загрузку данных, подготовку (много шагов очистки), построение модели (кластеризация + классификация), генерацию отчёта. Без инкапсуляции это десятки узлов в одной плоскости – трудно читать. Применяем инкапсуляцию: объединяем шаги очистки в подмодель "Очистка данных", шаги кластеризации и сегментации – в подмодель "Сегментация", шаги классификации в "Прогноз оттока". На верхнем уровне останутся 4 узла: Импорт, Очистка данных, Моделирование (внутри которого две подмодели-сегментация и прогноз), Отчет. Схема стала **лаконичной и ясной**. При этом, если нужно улучшить очистку – открываем подмодель, правим детали, внешняя схема не меняется. Если хотим этот же блок очистки применить в другом проекте – просто импортируем или копируем подмодель "Очистка данных".

Таким образом, **инкапсуляция элементов сценария** в Loginom реализована через подмодели, циклы, условия и повторно используемые компоненты. Это приносит следующие выгоды: - *Модульность:* сложный процесс разбивается на логические модули с чёткими интерфейсами (портами), что облегчает разработку (можно разрабатывать и тестировать по частям) и отладку. - *Повторное использование:* один раз разработанный модуль можно вставить в другие сценарии, экономя время и обеспечивая единообразие подходов. - *Удобство чтения:* сценарии верхнего уровня становятся короче и понятнее, показывая только высокоуровневую последовательность шагов. - *Абстракция:* пользователи, которым не нужны детали, могут работать с компонентом как с чёрным ящиком, уменьшая риск что-то сломать внутри.

Loginom фактически поддерживает концепцию «*аналитической сборки из блоков*», где каждый блок может быть комплексным внутри, но простым снаружи. Это ключ к масштабируемости проектов: когда анализ растёт, его всегда можно рефакторить, упаковывая части в подмодели. В результате платформа подходит не только для разовых задач, но и для построения больших промышленных аналитических процессов, содержащих сотни узлов, потому что с инкапсуляцией ими можно управлять и поддерживать их структуру в порядке. Каждый элемент сценария – как кирпичик в стене, и возможность делать из кирпичиков блоки, а из блоков – большие секции, существенно упрощает строительство «аналитической системы» в Loginom.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 все лекции.docx

file:///file_000000004368720ab2b5431f484bb9a4