

# Анализ данных, Loginom, вейвлет-формула и сети Кохонена: основные понятия

## Определение анализа данных

**Анализ данных** – это процесс сбора, очистки, преобразования и интерпретации наборов данных с целью извлечения полезной информации и принятия решений. В общем случае анализ данных включает изучение взаимосвязей между переменными и выявление закономерностей. Например, статистический анализ может определять степень связи (корреляции) между двумя или более показателями или строить модель для прогнозирования одного параметра по другим. Важной частью анализа данных является визуализация – представление результатов в виде графиков, диаграмм и отчетов, что облегчает понимание полученных инсайтов.

## Инструменты визуализации данных (кроме Loginom)

Для визуализации и анализа данных помимо платформы Loginom широко используются различные программные продукты (Business Intelligence системы и инструменты визуальной аналитики). К наиболее популярным относятся: **Tableau, Microsoft Power BI, Qlik Sense, TIBCO Spotfire, Google Data Studio** и др. Эти инструменты позволяют создавать интерактивные графики, дашборды и отчеты, упрощая восприятие больших наборов данных. Например, продукты вроде Tableau, Power BI или Qlik могут быть использованы для построения диаграмм, карт, сводных таблиц и других визуализаций, помогая бизнес-пользователям анализировать данные без необходимости программирования <sup>1</sup>. Кроме того, существуют открытые библиотеки для визуализации (Matplotlib, ggplot2 и пр.) и специализированные платформы (Oracle BI, IBM Cognos и т.д.), которые также применяются для представления данных в наглядной форме.

## Особенности и преимущества платформы Loginom

**Loginom** – это российская аналитическая платформа класса *low-code*, предназначенная для интеграции, очистки и продвинутого анализа данных. Её ключевая идея – визуальное конструирование процессов обработки данных с минимальным кодированием. Ниже приведены характерные особенности и преимущества Loginom:

1. **Low-code архитектура:** Позволяет выстраивать сложные сценарии анализа данных с помощью графического интерфейса, практически без ручного программирования <sup>2</sup>. Это делает платформу доступной для бизнес-пользователей, не обладающих глубокими навыками кодирования.
2. **Богатый набор компонентов:** В составе Loginom имеется более 60 готовых узлов для обработки данных – от базовых операций до алгоритмов машинного обучения <sup>3</sup>. Пользователь может применять преднастроенные модули для типовых задач (фильтрация, преобразование, моделирование и пр.), включая нейросети, регрессии, кластеризацию и т.д.

3. **Интеграция с различными источниками:** Платформа поддерживает подключение множества источников данных: файлы (Excel, CSV, XML, собственный формат LGD), базы данных (Oracle, MS SQL, PostgreSQL, ClickHouse, MySQL и др.), приложения (например, 1С:Предприятие), внешние сервисы (SOAP/REST API) и др. – это обеспечивает сквозную загрузку и консолидацию данных из разнородных систем <sup>4</sup>.
4. **Высокая производительность и масштабируемость:** Loginom рассчитан на обработку больших данных. Благодаря вычислениям *in-memory* и параллельной обработке, платформа демонстрирует высокую скорость работы, превосходя многие рыночные аналоги по производительности <sup>5</sup>. Это позволяет анализировать крупные наборы данных практически в реальном времени.
5. **Повторное использование и кастомизация компонентов:** Пользователи могут создавать собственные *производные компоненты* и подмодели. Один раз настроенный аналитический сценарий можно сохранить как компонент и затем многократно использовать его в других проектах <sup>6</sup>. Такая модульность упрощает тиражирование решений и накопление экспертизы в организации.
6. **Широкие возможности визуализации и отчетности:** В Loginom встроены инструменты для визуального исследования данных – поддерживаются OLAP-кубы, диаграммы, таблицы качества данных и специализированные визуализаторы <sup>7</sup>. Это позволяет не только подготовить данные, но и сразу построить интерактивные отчеты и дашборды для конечных пользователей.
7. **Развёртывание и лицензирование:** Платформа гибко разворачивается – доступна *Desktop*-редакция для персонального компьютера, серверная установка *on-premises*, а также облачная *SaaS*-версия. Модель лицензирования разнообразна: имеется бесплатная версия, варианты подписки, постоянной лицензии и пр. <sup>8</sup>. Таким образом, компания может подобрать удобный способ использования Loginom.
8. **Отечественная разработка и экосистема поддержки:** Loginom – 100% российский продукт, включённый в реестр отечественного ПО <sup>9</sup>. Имеются обширные русскоязычные документация, Wiki (450+ статей), форум вопросов и ответов, обучающие курсы (Loginom Skills) и более 150 видеуроков <sup>10</sup>. Платформа преподаётся в учебных программах свыше 70 вузов России, что подтверждает её признание в профессиональной среде <sup>11</sup>.

Таким образом, Loginom сочетает простоту визуального *low-code* инструмента с мощностью аналитической платформы промышленного уровня. Она позволяет организациям без привлечения больших ИТ-ресурсов внедрять решения для **ETL, BI, Data Mining** и **ML**, существенно ускоряя работу с данными и принятием решений <sup>12</sup> <sup>13</sup>.

## Основные части интерфейса Loginom

Интерфейс Loginom с выделенными основными блоками: (1) Главное меню и адресная строка наверху, (2) Панель компонентов слева, (3) Рабочая область сценария (*canvas*) по центру, (4) Панель процессов внизу.

Интерфейс Loginom веб-ориентированный и предоставляет пользовательскую среду для построения и выполнения сценариев анализа данных. Основные части интерфейса следующие <sup>14</sup> <sup>15</sup>:

- **Главное меню** – расположено в верхней части окна. Через него пользователь может создавать и открывать проекты (*пакеты*), настраивать параметры платформы, запускать/

завершать работу и т.д. Состав пунктов главного меню может меняться в зависимости от версии (Desktop или Server) и прав пользователя <sup>16</sup> .

- **Адресная строка** – панель под главным меню, отображающая путь к текущему открытому объекту (проекту, узлу сценария или другому элементу) <sup>17</sup> . Позволяет быстро ориентироваться, в каком разделе или пакете ведется работа.
- **Рабочее пространство** – основная область экрана, где разрабатываются сценарии обработки данных. Здесь пользователь *визуально* соединяет узлы (блоки-компоненты), задаёт их параметры, настраивает подключения к данным, создает отчёты и т.п. В режиме разработки сценария рабочее пространство представляет собой холст, на котором отображаются узлы и связи между ними <sup>18</sup> . При переходе к другим страницам (например, настройкам или администрированию) содержимое рабочей области меняется соответствующим образом.
- **Панель процессов** – нижняя область интерфейса, в которой отображается служебная информация о выполняемых процессах и узлах сценария <sup>19</sup> . Во время запуска сценария здесь можно видеть прогресс выполнения, сообщения об ошибках, логирование и т.д., что помогает отладке и контролю процесса анализа.
- **Панель компонентов** – расположена слева (вертикальная колонка). Содержит библиотеку всех доступных узлов (компонентов) разбитых по категориям: источники данных, преобразования, моделирование, визуализация и прочее <sup>20</sup> . Чтобы добавить операцию в сценарий, достаточно перетащить нужный компонент из этой панели на холст рабочей области. Панель компонентов можно скрывать или разворачивать при необходимости.
- **Инспектор свойств** – всплывающая панель (обычно скрыта по умолчанию), которая показывается при выборе конкретного узла сценария. В инспекторе отображаются свойства и настройки выделенного компонента, их можно редактировать <sup>21</sup> <sup>22</sup> . Открыть или закрыть инспектор можно кнопкой на верхней панели инструментов.

**Верхняя панель инструментов** – это набор кнопок и элементов управления, находящийся прямо над рабочим холстом сценария (под адресной строкой). Данная панель облегчает манипуляцию узлами и сценарием в целом <sup>23</sup> . На ней, в частности, доступны следующие операции <sup>24</sup> <sup>25</sup> :

- Показать/скрыть панель компонентов (левая колонка) для освобождения места под холст <sup>26</sup> .
- Переключение режимов: режим *выбора/редактирования* объектов сценария и режим *навигации* (прокрутки) по области сценария <sup>27</sup> .
- Представление сценария: стандартный графический вид (блок-схема) или компактный вид *таблицы* сценария <sup>28</sup> . Табличное представление перечисляет все узлы списком.
- Выравнивание узлов – автоматическое расположение или выравнивание выделенных блоков по левому/правому краю, по центру, по вертикали или горизонтали и т.д., для аккуратной компоновки схемы <sup>29</sup> .
- **Выполнение сценария:** кнопки для запуска всех узлов сразу, либо запуска выделенного узла (с выполнением предшествующих) <sup>30</sup> . Здесь же можно деактивировать узлы, настроить порядок их выполнения вручную <sup>31</sup> , а также запустить *обучение* моделей (например, переобучение нейросети) <sup>32</sup> .
- Операции редактирования: открыть настройки узла, добавить *заметку* на холст (текстовый комментарий), копировать/вырезать/вставить узлы, клонировать узел, удалить выбранные элементы <sup>33</sup> . Также есть функция свернуть группу узлов в подмодель или развернуть подмодель обратно <sup>34</sup> , создать новый производный компонент из выделенного узла <sup>35</sup> , показать связанные узлы и т.п.

- Отображение вспомогательной информации: включение показа количества строк на выходных портах узлов <sup>36</sup>, открытие мини-карты сценария для навигации (масштабирования) <sup>37</sup>, и переключение инспектора свойств <sup>38</sup>.

Таким образом, интерфейс Loginom интуитивно разделен на области: сверху – навигация и инструменты, слева – библиотека компонентов, центр – визуальное моделирование сценария, снизу – мониторинг выполнения. Это обеспечивает удобство разработки сложных процессов анализа данных без программирования.

## Порядок анализа данных (workflow) в Loginom

Процесс анализа данных типично проходит через несколько последовательных этапов. Ниже описан общий порядок анализа данных и соответствующие функции в Loginom:

- **Подключение данных (подсоединение):** На первом шаге необходимо подключиться к источникам данных. Это может быть импорт файлов (Excel, CSV и др.), подключение к базе данных (SQL), обращение к веб-сервису или другим системам. В Loginom для этого служат компоненты из категории «Подключения», позволяющие настроить соединение с нужным хранилищем данных. Например, можно добавить узел подключения к базе MS SQL или импортировать таблицу из Excel. На этом этапе происходит извлечение сырых данных в рабочее пространство системы <sup>39</sup>.
- **Очистка и подготовка данных:** Сырые данные часто содержат ошибки, пропуски, дубликаты, лишние столбцы и т.п. Этап очистки включает фильтрацию и исправление выбросов, удаление дубликатов, заполнение пропущенных значений, приведение типов данных, нормализацию показателей и другие преобразования. В Loginom предусмотрены компоненты для очистки – например, «Дубликаты и противоречия» (поиск повторов), «Заполнение пропусков», «Фильтр строк», преобразование форматов дат, замена значений и др. Также сюда относится **трансформация данных** – агрегирование, объединение таблиц, расчет новых полей (с помощью узла «Калькулятор» или скриптов), кодирование категориальных признаков и т.д. Результатом этапа является **подготовленный набор данных**, пригодный для анализа.
- **Анализ данных / моделирование:** На подготовленных данных выполняются собственно методы анализа – это может включать описательную статистику, поиск зависимостей, построение моделей и проверку гипотез. Часто анализ начинают с простых статистических оценок и визуализации, чтобы понять структуру данных. Далее, в зависимости от целей, применяются различные методы: корреляционный анализ, факторный анализ, кластеризация, регрессионный анализ, построение прогнозных моделей (регрессии, решающие деревья, нейронные сети и др.), классификация и т.п. В Loginom многие из этих методов реализованы как отдельные узлы в категории «Data Mining» (например, компоненты **Корреляционный анализ, Факторный анализ, Кластеризация, Линейная регрессия, Логистическая регрессия, Нейросеть** и др. <sup>40</sup> <sup>41</sup>). Пользователь может добавлять нужный метод на схему, настраивать его параметры (выбирать переменные, задавать настройки алгоритма) и запускать вычисление. Если анализ носит исследовательский характер, могут проводиться эксперименты с разными методами и параметрами, сравнение результатов (например, пробуют несколько моделей и оценивают их качество).
- **Вейвлет-анализ:** Особое место среди методов занимает **вейвлет-анализ** – инструмент для работы с временными рядами и сигналами. Вейвлет-преобразование позволяет представить сигнал во временно-частотной области, выявляя локальные особенности, тренды и

периодичности, которые не видны при стандартном фурье-анализе. Суть метода заключается в разложении исходного ряда с помощью специальной «маленькой волны» – *вейвлета*, который масштабируется и сдвигается вдоль оси времени <sup>42</sup> <sup>43</sup>. **Вейвлет-преобразование** непрерывного сигнала  $x(t)$  относительно выбранной материнской функции  $\psi(t)$  определяется формулой:

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi^*\left(\frac{t-b}{a}\right) dt,$$

где  $a > 0$  – масштаб (степень сжатия/растяжения вейвлета),  $b$  – сдвиг по времени,  $a *$  обозначает комплексно-сопряженную функцию (если вейвлет комплексный) <sup>44</sup> <sup>45</sup>. Меняя параметр  $a$ , вейвлет «растягивается» или «сжимается», что позволяет анализировать как низкочастотные, так и высокочастотные компоненты сигнала, а параметр  $b$  задаёт положение во времени. Таким образом, вейвлет-анализ даёт двухмерное представление данных (время и масштаб), раскрывая, например, моменты начала/конца каких-либо периодических компонентов, всплески активности и т.п. Практическое применение вейвлет-анализа – обработка временных рядов (финансовых, метеорологических), денойзинг сигналов, сжатие данных (формат JPEG2000 использует вейвлет-сжатие) и многое другое <sup>46</sup> <sup>47</sup>. В контексте задач анализа данных владение вейвлет-методами расширяет возможности исследования временных зависимостей и аномалий.

- **Интерпретация и визуализация результатов:** Полученные модели или статистические выводы необходимо проинтерпретировать в терминах предметной области. На завершающем этапе аналитик формирует выводы: какие факторы влияют на целевой показатель, каковы прогнозные значения, какие группы клиентов выделились при кластеризации и т.д. Для наглядности результаты представляют в виде диаграмм, графиков, карт – это может быть итоговый **дашборд** или отчет для бизнес-пользователей. В LogiNot итоговую визуализацию можно настроить с помощью узлов-визуализаторов (диаграммы, таблицы, OLAP-кубы и пр.), либо экспортировать результаты во внешние BI-системы при необходимости.

- **Принятие решений:** Собственно цель анализа – выработать рекомендации или решения на основе обнаруженных инсайтов. Например, по итогам анализа данных могут быть предложены изменения в бизнес-процессе, таргетированные акции для сегментов клиентов, настройка параметров оборудования и т.д. Этот шаг лежит уже вне самой системы анализа данных, но качество и правильность предыдущих этапов напрямую влияют на эффективность решения.

Отметим, что анализ данных редко бывает линейным одноразовым процессом – часто он итеративен. Аналитик может возвращаться к предыдущим этапам: например, после построения модели обнаружить, что нужны дополнительные преобразования данных или что в наборе есть выбросы, исказившие результаты, – тогда выполняется повторная очистка или сбор новых данных. Среда LogiNot поддерживает такой интерактивный подход: сценарий можно легко модифицировать, добавляя или переупорядочивая шаги, а затем запускать снова на обновленных данных.

## Корреляционный анализ

**Корреляционный анализ** – это статистический метод, с помощью которого измеряется теснота связи между двумя или более переменными. Проще говоря, корреляция показывает, насколько изменения одной величины сопутствуют систематическим изменениям другой величины <sup>48</sup>. Вычисляется корреляция обычно через коэффициенты корреляции. Наиболее известен коэффициент Пирсона (для линейной связи), который принимает значения от -1 до +1: положительное значение означает прямую связь (при росте одного показателя другой тоже возрастает), отрицательное –

обратную связь (один растет, другой снижается), а близкий к нулю коэффициент говорит об отсутствии линейной зависимости.

Пример: если мы проводим корреляционный анализ между количеством часов, затраченных студентом на подготовку, и его баллом экзамена, то положительная корреляция укажет, что большее время подготовки связано с более высоким баллом. Однако важно понимать, что корреляция – это не причинно-следственная зависимость. Высокая корреляция лишь свидетельствует о статистической взаимосвязи, но не доказывает, что одно событие вызывает другое. Например, может обнаружиться высокая корреляция между объемом продаж мороженого и числом утоплений – однако оба этих фактора связаны не напрямую, а через третью переменную (жаркую погоду). Поэтому корреляционный анализ используют для выявления связей и формирования гипотез, но выводы о причинности требуют дополнительных исследований <sup>49</sup> <sup>50</sup>.

В Loginom для автоматизации корреляционного анализа имеется специальный компонент «**Корреляционный анализ**», который по входным данным вычисляет матрицу корреляций между признаками и может выявлять наиболее связанные пары факторов <sup>51</sup>. Результаты корреляционного анализа часто применяются на этапе предварительного отбора переменных для моделей (например, удаляют один из двух сильно коррелирующих признаков, чтобы избежать коллинеарности) и для общего понимания структуры данных.

## Регрессия (регрессионный анализ)

**Регрессионный анализ** – это совокупность методов, предназначенных для моделирования зависимости одной переменной от другой или нескольких других. В таком анализе выделяют зависимую переменную (что мы пытаемся предсказать или объяснить) и независимые переменные (факторы-признаки, по которым идет прогноз) <sup>52</sup>. Цель регрессии – подобрать такую функцию (модель), которая наилучшим образом описывает влияния независимых переменных на зависимую. Например, в бизнесе регрессия может использоваться, чтобы спрогнозировать продажи ( $Y$ ) на основе факторов  $X$ : цена товара, бюджет рекламы, сезонность и т.д.

Наиболее распространенным частным случаем является **линейная регрессия**, где предполагается линейная зависимость:  $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$ . Классическим методом оценки коэффициентов  $\beta$  в линейной регрессии служит метод наименьших квадратов – коэффициенты подбираются так, чтобы минимизировать сумму квадратов отклонений прогнозируемых значений от фактических <sup>53</sup>. В результате получается уравнение регрессии, позволяющее по значениям  $X$  получить оценку  $Y$ .

Помимо прогнозирования, регрессионный анализ решает задачу выяснения степени влияния факторов: значения коэффициентов при  $X$  показывают, как изменение соответствующего фактора сказывается на целевой переменной (при прочих равных). Важным статистическим показателем качества модели является  $R^2$  – доля дисперсии зависимой переменной, объясненная регрессией.

Пример: пусть у нас данные по домам –  $Y$  = цена продажи дома,  $X_1$  = площадь дома,  $X_2$  = число комнат. Построив регрессию, можем получить формулу вида: *цена* = 50 + 0.2·*площадь* + 10·*число комнат* (условно). Такой результат означал бы, что каждый дополнительный квадратный метр площади в среднем прибавляет 0.2 (тыс. \$) к цене, а каждая комната – 10 (тыс. \$), при этом базовая

цена (для нулевой площади и комнат) была бы 50 тыс. \$. Конечно, реальность сложнее, но модель дает приблизительные оценки и позволяет прогнозировать цену для новых домов.

В Logiplot компонент **«Линейная регрессия»** строит регрессионную модель по указанным полям: на выходе дается статистика (значимость факторов,  $R^2$ , среднеквадратичная ошибка и т.п.) и сами коэффициенты уравнения <sup>40</sup>. Также Logiplot поддерживает полиномиальную регрессию, регрессию с регуляризацией и некоторые другие варианты через скрипты Python/R при необходимости.

Регрессионный анализ является одним из базовых инструментов аналитика данных, широко применим в экономике, технике, социальных науках – всюду, где нужно понять и количественно оценить зависимости между переменными.

## Факторный анализ

**Факторный анализ** – это метод статистического анализа, используемый для выявления скрытых факторов (причинных признаков), влияющих на набор наблюдаемых переменных. Он относится к методам снижения размерности данных. Предполагается, что множество исходных признаков обусловлено действием меньшего числа латентных (ненаблюдаемых напрямую) факторов, и цель анализа – обнаружить эти факторы и структуру взаимоотношений между ними <sup>54</sup>.

Проще говоря, факторный анализ пытается объяснить корреляции между исходными переменными через несколько общих факторов. Например, если у нас есть результаты экзаменов студентов по разным предметам, можно предположить, что за высокими корреляциями между оценками по физике, математике и информатике стоит скрытый фактор “технические способности”, а за корреляцией между оценками по литературе и истории – фактор “гуманитарные склонности”. Факторный анализ позволил бы количественно выделить эти факторы и посчитать, как сильно каждый предмет на них “нагружен”.

Методология: на вход подается корреляционная матрица исходных переменных. Затем алгоритм находит такие комбинации переменных (факторы), которые объясняют наибольшую часть дисперсии данных. Первый фактор объясняет максимальную возможную часть общей вариации, второй – большую часть оставшейся, и так далее. Обычно практик оставляет лишь несколько первых факторов, которые в сумме объясняют достаточную долю (~70-90%) дисперсии, а остальные мелкие факторы отбрасываются как шум. Каждый фактор – это линейная комбинация исходных переменных, и каждому исходному признаку соответствует *факторная нагрузка* – коэффициент связи с фактором <sup>55</sup> <sup>56</sup>. Высокие нагрузки указывают, какие переменные тесно связаны с данным фактором.

Факторный анализ решает две основные задачи: (1) **обобщение и сжатие информации** – заменить множество коррелирующих переменных несколькими интегральными факторами; (2) **поиск скрытых причин** – понять, какие сущностные силы стоят за наблюдаемыми данными <sup>57</sup> <sup>58</sup>. После выделения факторов часто их стараются интерпретировать осмысленно, присвоив им названия (например, “интеллектуальный фактор”, “социально-экономический статус” и т.д., в зависимости от предметной области).

В Logiplot компонент **«Факторный анализ»** автоматизирует вычисление – он может выполнить метод главных компонент (как частный случай факторного анализа) или другие техники и отчитаться

о доле объясненной дисперсии, матрице факторов и нагрузок <sup>59</sup> <sup>60</sup>. Часто после факторного анализа выполняют *вращение факторов* (Varimax или др.), чтобы получить более интерпретируемую структуру – т.е. стремятся чтобы у каждой переменной была высокая нагрузка на один фактор и низкая на остальные, что упрощает трактовку факторов.

Факторный анализ широко используется в социологии и психологии (выявление скрытых черт личности по опросникам), в финансах (определение основных движущих сил рынка), в маркетинге (выделение групп схожих характеристик потребителей) и т.д.

## Кластерный анализ

**Кластерный анализ** (кластеризация) – совокупность методов анализа данных, цель которых – разделение множества объектов (наблюдений) на группы (*кластеры*) таким образом, чтобы объекты внутри одного кластера были максимально похожи друг на друга, а объекты из разных кластеров – различны. В формальном определении кластерный анализ – это многомерная статистическая процедура, которая берет данные об выборке объектов и упорядочивает (группирует) эти объекты в относительно однородные группы <sup>61</sup>. Кластеризация относится к методам **обучения без учителя**, поскольку группы выявляются без заранее известной правильной классификации.

Простой пример: имея базу клиентов с разными характеристиками (возраст, доход, частота покупок, предпочитаемый вид товара и пр.), кластеризация может автоматически разбить клиентов на сегменты – например, выделить кластер “молодые активные онлайн-покупатели” и “консервативные возрастные клиенты офлайна” и т.д., исходя только из имеющихся данных, без ручной метки классов. Это позволяет лучше понимать структуру данных и применять таргетированные стратегии для разных кластеров.

Существуют разные алгоритмы кластерного анализа: иерархические (постепенное объединение или разделение кластеров с построением дендрограммы), метод k-средних (k-means, разбивает на заданное число кластеров, итеративно уточняя центры кластеров), плотностные методы (DBSCAN – ищет области высокой плотности точек) и многие другие. Выбор меры сходства/различия также важен – часто используется евклидово расстояние для количественных данных или другие метрики.

В Loginom кластерный анализ представлен компонентом **«Кластеризация»**. Он позволяет выполнить автоматическую кластеризацию данных (например, метод k-средних) и получить для каждого наблюдения метку кластера <sup>62</sup>. Кроме того, для углубленного анализа результатов есть визуализатор *«Профили кластеров»*, который показывает характеристики полученных кластеров (средние значения переменных по кластерам и др.) – это помогает интерпретировать смысл групп.

Кластерный анализ широко применяется: в маркетинге – для сегментации клиентов, в биологии – для классификации схожих видов или генов, в банковской сфере – для выявления групп схожих сделок (например, чтобы обнаружить аномальные транзакции), в обработке изображений – для сегментации пикселей по цвету, и т.д. Ключевая особенность – метод позволяет *открыть новые, заранее не заданные категории* в данных, что делает его ценным инструментом разведочного анализа.



## Логистическая регрессия

**Логистическая регрессия** – это статистическая модель, применяемая для прогнозирования вероятности наступления некоторого события на основе набора признаков. В отличие от обычной (линейной) регрессии, логистическая регрессия используется, когда зависимая переменная является бинарной (два возможных исхода, например: событие произошло или нет, клиент ушел или остался, 0/1) <sup>63</sup>. Модель возвращает на выходе вероятность принадлежности наблюдения к классу “1” (событие произошло).

Название связано с использованием *логистической функции* (сигмоиды) для связи линейной комбинации входных переменных с вероятностью. Функция сигмоиды имеет вид  $f(z) = \frac{1}{1+e^{-z}}$  – она преобразует любое реальное число  $z$  в диапазон (0,1), что естественно интерпретируется как вероятность. В логистической регрессии сначала вычисляется линейный взвешенный суммарный показатель  $z = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n$  (похож на линейную регрессию), а затем к нему применяется сигмоида, давая прогнозируемую вероятность:  $P(y = 1 | x) = f(z) = \frac{1}{1+e^{-z}}$  <sup>64</sup> <sup>65</sup>. Если полученная вероятность выше некоторого порога (обычно 0.5), то модель классифицирует наблюдение как класс “1”, иначе – “0”.

Например, логистическая регрессия может прогнозировать вероятность того, что посетитель сайта совершит покупку (да/нет) на основе таких признаков, как время, проведенное на сайте, число просмотренных страниц, источник трафика и т.п. Модель выдаст вероятность в процентах, и на основе порога можно решить, считать ли данного посетителя потенциальным покупателем (которому, например, показать специальное предложение).

Преимущество логистической регрессии – интерпретируемость: коэффициенты  $\theta_i$  можно трансформировать в **относительные шансы** (odds ratio), показывающие влияние признака. Кроме того, модель не требует нормального распределения признаков и хорошо работает при большом количестве наблюдений.

В Logipom логистическая регрессия реализована как компонент **«Логистическая регрессия»**, который позволяет обучить такую модель на размеченных данных (где известен факт наступления события) <sup>62</sup>. После обучения компонент выдает оценку качества (например, Accuracy, ROC-AUC) и коэффициенты модели. Логистическая регрессия относится к методам классификации и часто используется в задачах кредитного скоринга (вероятность дефолта), в маркетинге (вероятность отклика на кампанию), медицине (вероятность заболевания по симптомам) и во многих других областях, требующих предсказать бинарный исход.

## Нейронные сети

**Искусственные нейронные сети** – это класс моделей машинного обучения, вдохновленных устройством биологических нейронных сетей мозга. Математически нейронная сеть представляет собой совокупность взаимосвязанных между собой искусственных нейронов (узлов), объединенных в слои. Каждый нейрон выполняет простое преобразование входных сигналов в выходной (как правило, суммируя входы с определенными весами и пропуская через нелинейную функцию активации). Нейронные сети способны учиться на данных, автоматически настраивая веса связей между нейронами, и таким образом извлекать сложные зависимости.

Классическая архитектура – **многослойный перцептрон** (MLP) – состоит из входного слоя (принимает исходные признаки), нескольких скрытых слоев нейронов и выходного слоя (выдает прогноз). Обучение сети происходит обычно методом обратного распространения ошибки: сеть прогоняет обучающие примеры, вычисляет ошибку прогноза и корректирует веса, чтобы уменьшить эту ошибку. После многократного повторения (эпох) сеть настраивает свои параметры таким образом, что может выдавать правильные ответы не только на обучающие, но и на новые примеры (если обучена удачно и не переобучена).

Искусственные нейронные сети – очень мощный инструмент, потому что они универсальные аппроксиматоры: при достаточном числе нейронов они способны приблизить сколь угодно сложную зависимость. В современных условиях нейронные сети применяются для задач распознавания изображений, обработки естественного языка, прогнозирования временных рядов, генерации контента и др. С появлением глубоких нейронных сетей (*deep learning*, с десятками и сотнями слоев) удалось достичь прорывных результатов во многих задачах ИИ.

В начале разработки нейросеть и ее **гиперпараметры** (количество слоев, число нейронов в каждом, функция активации, скорость обучения и пр.) задает аналитик. Затем сеть обучается на данных. В процессе обучения она фактически самостоятельно *конструирует признаки* во внутренних слоях – каждый следующий слой формирует более абстрактные представления данных, опираясь на предыдущий. Например, при распознавании изображений: первые слои выделяют простые контуры, следующие – составляют из них формы, а последние – распознают конкретные объекты.

В Loginom имеется компонент «**Нейросеть (классификация)**» и «**Нейросеть (регрессия)**», что позволяет использовать многослойные перцептроны для соответствующих задач прямо в визуальном сценарии <sup>66</sup>. Пользователь задает архитектуру и параметры обучения, после чего узел обучает модель и выдает результаты (например, для классификации – метрики точности, матрицу конфузии и прогнозы для каждого примера). В более сложных случаях пользователь может интегрировать скрипты на Python/R в Loginom и воспользоваться любыми библиотеками глубокого обучения.

Важно отметить, что хотя нейронные сети зачастую превосходят другие алгоритмы по точности, они менее интерпретируемы – сложно понять, как именно сеть приняла то или иное решение. Поэтому в критически важных приложениях (медицина, банковский скоринг) нейросети применяют с осторожностью, либо дополняют методами объяснения моделей.

## Кластерный анализ с помощью сетей Кохонена

**Сети Кохонена** – это особый тип нейронных сетей, предназначенных для кластеризации и визуализации данных. Они реализуют алгоритм *самоорганизующейся карты* (Self-Organizing Map, SOM), предложенный финским ученым Теуво Кохоненом. Сеть Кохонена имеет двухслойную архитектуру: входной слой (просто распределяет входные сигналы) и выходной слой – собственно *карта* Кохонена, состоящая из нейронов, организованных чаще всего в виде двумерной решетки (например, узлы расположены в виде сетки 10x10) <sup>67</sup>. Каждый выходной нейрон сети ассоциирован с определенным *кластером* данных. В процессе работы сеть проецирует многомерные входные векторы данных на эту 2D-карту, размещая похожие объекты рядом друг с другом. Это дает одновременно кластеризацию и наглядную карту структуры данных <sup>68</sup>.

Работа сети Кохонена базируется на принципе **конкурентного обучения** (обучение без учителя). Алгоритм обучения можно описать так:

- Инициализация: каждому выходному нейрону присваивается случайный весовой вектор такой же размерности, как у входных данных (эти веса можно рассматривать как координаты “прототипа” кластера в пространстве признаков). Узлы выходного слоя имеют топологию (соседние узлы определяются по решетке).
- Для каждого обучающего примера (входного вектора) сеть вычисляет расстояние от этого вектора до весовых векторов всех нейронов карты (например, евклидово расстояние). Находит нейрон, чей вес наиболее близок к входному – это **нейрон-победитель** (Best Matching Unit, BMU), он считается центром, куда данный образ должен проецироваться <sup>69</sup>.
- Затем производится обновление весов: нейрон-победитель и его *соседи* на карте корректируют свои весовые векторы в сторону входного вектора. Формула обновления весов для нейрона  $l$  может быть записана как:  
$$W_l^{new} = W_l^{old} + \eta \cdot h_{j(x),l} \cdot (X - W_l^{old}),$$
где  $X$  – входной вектор,  $j(x)$  – индекс победившего нейрона для  $X$ ,  $\eta$  – коэффициент обучения (шага), а  $h_{j,l}$  – функция соседства, задающая влияние на нейрон  $l$  в зависимости от его расстояния от победителя  $j$  на карте <sup>69</sup> <sup>70</sup>. Ближайшие соседи получают наибольшую долю сдвига, дальние – меньшую. Со временем радиус соседства обычно уменьшают, фокусируя обучение на более локальных группах.
- Процесс повторяется многократно для всех образцов, обычно в течение нескольких эпох, пока сеть не “самоорганизуется” – т.е. веса стабилизируются.

По окончании обучения каждый выходной нейрон имеет свой весовой вектор, который является как бы *эталоном* одного из кластеров. При подаче новых данных сеть находит для них ближайший нейрон – тем самым классифицируя их в соответствующий кластер. Поскольку соседние нейроны обучались схожим образом, похожие данные активируют расположенные рядом узлы. Это означает, что SOM не только разделяет кластеры, но и упорядочивает их на карте так, что кластеры с общими свойствами оказываются соседними областями на карте (топологическая сохранность).

**Карты Кохонена** особенно ценны тем, что предоставляют наглядную визуализацию многомерных данных в двумерном пространстве. Каждому узлу карты можно сопоставить, сколько объектов к нему приписалось, или средние значения признаков для него, и отобразить эти данные цветом или диаграммой прямо на клетках карты. Получается «*портрет*» всего набора данных. Например, анализируя поведение клиентов банка, можно построить SOM, где каждая точка карты – некоторый тип клиентов, и затем раскрасить карту по признаку “рентабельность клиента” – это может выявить области карты (кластеры) с наиболее ценными клиентами, с оттоком и т.д. SOM таким образом служат инструментом **разведывательного анализа данных** – они часто используются, чтобы получить первое представление о структуре данных, найти выбросы, понять, существуют ли естественные группы <sup>71</sup> <sup>72</sup>.

С точки зрения кластерного анализа, сеть Кохонена выполняет **автоматическую кластеризацию** сравнимо с другими методами, но обладает преимуществом сохранения топологии: результаты можно интерпретировать как карту, где расстояние между узлами отражает степень различия кластеров. Количество выходных нейронов (размер карты) определяет, сколько кластеров сеть потенциально может выявить <sup>73</sup>. Например, карта 10×10 содержит 100 нейронов – это максимальное число кластеров, но фактически близкие узлы могут составлять один крупный кластер

или иерархию. Аналитик может рассматривать мелкую сетку SOM и объединять соседние узлы в группы для нужной сегментации.

В Loginom есть специальный компонент **«Самоорганизующиеся сети»**, реализующий алгоритм Кохонена. Он позволяет настроить размер карты, параметры обучения и запустить кластеризацию. На выходе компонент дает для каждого входного объекта номер узла (координаты) на карте, к которому он приписан <sup>74</sup>. Также Loginom предоставляет визуализатор **«Карты Кохонена»** (профили), где можно изучать получившиеся кластеры более подробно – например, просматривать распределения значений признаков в каждом кластере, расстояния между узлами, и т.д., что облегчает интерпретацию самоорганизующейся карты.

**Алгоритмы Кохонена** работают по принципу "победитель получает всё": только один нейрон (и его окрестность) активируется для каждого входа <sup>75</sup>. Это отличает их от других нейросетей – здесь нет учителя и желаемого выхода, обучение направлено на упорядочение по сходству. По сути, сети Кохонена можно рассматривать как метод кластерного анализа: они разбивают данные на кластеры, как k-means, но дополнительно предоставляют структуру взаимного расположения этих кластеров.

Подводя итог: **сети Кохонена и их карты** – мощный инструмент для кластеризации, позволяющий визуализировать многомерные данные. Алгоритмы Кохонена самоорганизуются, настраивая нейроны-«центроиды» под данные. Близкие данные возбуждают соседние нейроны, формируя на карте области – кластеры. Эти карты нашли применение в анализе данных в самых разных областях: от диагностики в медицине до маркетинговой сегментации, от обработки сигналов до исследовательских задач в астрономии. Их ценность – в наглядности и способности выявлять структуры без предварительных предположений о числе или форме кластеров. <sup>67</sup> <sup>71</sup>

#### Источники:

1. Описание платформы Loginom – функциональность, интерфейс и возможности <sup>12</sup> <sup>76</sup>.
2. Сравнение Loginom с зарубежными системами и ключевые преимущества Loginom <sup>1</sup> <sup>77</sup>.
3. Руководство пользователя Loginom – основные блоки интерфейса и инструменты сценария <sup>14</sup> <sup>23</sup>.
4. Статья на Яндекс.Практикуме о корреляции – определение и примеры корреляционного анализа <sup>48</sup> <sup>49</sup>.
5. Википедия: *Регрессионный анализ* – суть метода и пример линейной регрессии <sup>52</sup>.
6. Википедия: *Факторный анализ* – назначение метода и выявление латентных факторов <sup>54</sup> <sup>55</sup>.
7. Википедия: *Кластерный анализ* – определение кластеризации и области применения <sup>61</sup>.
8. Википедия: *Логистическая регрессия* – модель для прогнозирования вероятностей событий <sup>63</sup>.
9. Документация Loginom – перечень компонентов Data Mining (корреляция, регрессия, логистическая регрессия, нейросеть и др.) <sup>40</sup> <sup>66</sup>.
10. Loginom Wiki: *Сеть Кохонена* – описание архитектуры и принципа работы самоорганизующихся сетей <sup>67</sup> <sup>75</sup>.
11. Википедия: *Нейронная сеть Кохонена* – алгоритм обучения самоорганизующейся карты и ее использование для визуализации данных <sup>69</sup> <sup>71</sup>.

12 13 76 Loginom: Описание, Функции и Интерфейс – 2025  
<https://soware.ru/products/loginom>

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 Страница «Сценарий» · Loginom Help

48 49 50 51 **Корреляция** — Википедия

https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D1%8F

54 55 56 57 58 59 60 Факторный анализ — Википедия

https://ru.wikipedia.org/wiki/  
%D0%A4%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B9\_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7

63 64 65 **Логистическая регрессия** — Википедия

https://ru.wikipedia.org/wiki/  
%D0%9B%D0%BE%D0%B3%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F\_%D1%80%D0%B5%D0%B3%D1%80%

68 Самоорганизующаяся карта Кохонена — Википедия  
https://ru.wikipedia.org/wiki/  
%D0%A1%D0%B0%D0%BC%D0%BE%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D1%83%D1%8E%D1%89%D0%B0%D1%8F%D1%81%

13