

## Наш второй компонент — «Бинарная сегментация»

Если первый компонент, который мы рассмотрели — Генератор календаря, — был относительно простым, то второй компонент будет прикладным — из бизнес-области по клиентской аналитике. Мы спроектируем и реализуем в виде компонента расчетный метод сегментации клиентов и добавим в него элементы экспертизы предметной области. Метод называется «Бинарная сегментация», он позволяет разделить клиентов на сегменты на основании изучения их активности в четырех периодах — отчетном и трех предшествующих. Рассмотрим его подробнее.

### 1. Выбор метода

Бинарная сегментация — простой метод поведенческой сегментации, активно применяемый маркетологами, который позволяет оценить, на какой стадии отношений с компанией находится тот или иной клиент — кто совершает визиты постоянно, кто вдруг перестал приходить, какие клиенты скорее всего потеряны, а каких еще можно удержать. В результате клиента можно отнести к одному из шестнадцати сегментов. Принадлежность к сегменту определяется на основании поведения клиента в течение четырех периодов: если был хотя бы один визит, периоду присваивается код 1, нет — 0. Таким образом, для каждого клиента описывается четырехзначным кодом, состоящим из нулей и единиц, отсюда и название метода — «бинарная сегментация».

В качестве основного источника литературы будем опираться на статью Натальи Устименко «**Бинарная сегментация**»<sup>1</sup>. Рекомендуем предварительно ознакомиться с ней. В ней кроме проведения непосредственно сегментации описывается, как оценить значимость сегмента, то есть определить, насколько, к примеру, сегмент 0101 лучше или хуже сегмента 0001. Для этого необходимо подсчитать, какое количество исследуемых клиентов совершили визит в следующем периоде, и определить их долю в общем

---

<sup>1</sup> Статья доступна по ссылке <https://esputnik.com/blog/binarnaya-segmentaciya>

числе клиентов сегмента. Полученный показатель будет представлять собой вероятность того, что клиент определенного сегмента совершит визит в следующем периоде.

Также в статье предлагается некая начальная интерпретация для сегментов, к примеру, указывается, что сегмент 1111 — это постоянные, или VIP-клиенты, а 0000 — скорее всего, потерянные. Проанализировав ее и дополнив, мы решили разделить итоговые шестнадцать сегментов на четыре более крупных сегмента, конкретнее определяющие характер отношения клиента с компанией на момент проведения сегментации: **Активный** (сегменты 0011, 0111, 1011 и 1111), **Спящий** (0010, 0100, 0110, 1010, 1100, 1110), **Реактивированный** (0001, 0101, 1001, 1101) и **Ушедший** (0000, 1000). Будем считать это некоторой экспертизой и также реализуем ее в компоненте.

## 2. Проектирование внешней оболочки

Метод мы будем реализовывать в виде компонента в Loginom. Уточним постановку задачи: нам требуется компонент, с помощью которого мы сможем определить факт визита клиента в каждый из четырех исследуемых периодов и на основании этого разделить клиентов на шестнадцать сегментов, а также подсчитать ряд статистик для этих сегментов:

- значимость сегмента (вес клиента сегмента);
- численность сегмента;
- количество клиентов, совершивших визит в следующем периоде;
- вероятность совершения покупки в следующем периоде для клиента сегмента.

Кроме этого мы хотим задать интерпретацию для каждого сегмента.

Исходя из требований, понимаем, что на вход необходимо подать набор данных с исторической информацией о датах визитов клиентов (визитом может быть покупка или любая транзакция).

Для проведения сегментации нам также понадобится информация о периодах, которые мы будем исследовать. Подать данные только за нужные периоды нельзя — нас интересуют даже те клиенты, которые не совершали визитов ни в один из исследуемых периодов, но приходили до этого — они

составят сегмент 0000. Поэтому на входе компонента в переменных будем задавать дату, на которую необходимо проводить сегментацию, и уже внутри компонента определять на ее основании исследуемые периоды. Кроме того, для различных компаний свойственна разная частота визитов, поэтому мы не можем знать заранее, какой тип периода — квартал, месяц, неделя и т.д. — нужно будет исследовать. Эта информация также будет задаваться с помощью переменной.

Шестнадцать итоговых сегментов мы решили сгруппировать в четыре, для которых подобрали словесную интерпретацию, поэтому понадобится еще один вход переменных — для задания пар *Код сегмента–Наименование*. Такая реализация позволит легко изменить названия сегментов, если указанные по умолчанию по какой-либо причине не подходят.

Выходом компонента будет набор данных с клиентами, где для каждого указаны код и наименование сегмента, к которому он относится, а также вес клиента. Второй выход — статистики по шестнадцати сегментам.

Рассмотрим входы и выходы, которые были созданы на основании наших рассуждений.

### **Вход «Входной набор данных»**

Так как нас интересует исключительно информация, приходил или нет клиент в определенный период, входной набор данных будет содержать поля:

1. **Идентификатор (ID)** — идентификатор клиента;
2. **Дата (Date)** — дата визита;

### **Вход «Переменные»**

На этом порте задаются параметры обработки:

1. **Дата актуальности (RelevanceDate)** — дата, на которую проводится сегментация;
2. **Тип периода (PeriodType)** — на вход подаются даты визитов, и в данной переменной задается информация о том, в период какого типа их необходимо преобразовать:

- $d$  – день;
- $w$  – неделя;
- $m$  – месяц;
- $q$  – квартал;
- $y$  – год.

По умолчанию используется значение *месяц* ( $m$ ).

### Вход «Сегменты»

Здесь задаются пары *Код сегмента–Наименование*. Названия сегментов прописаны в метках переменных, а коды — в значениях (см. рис. 1).

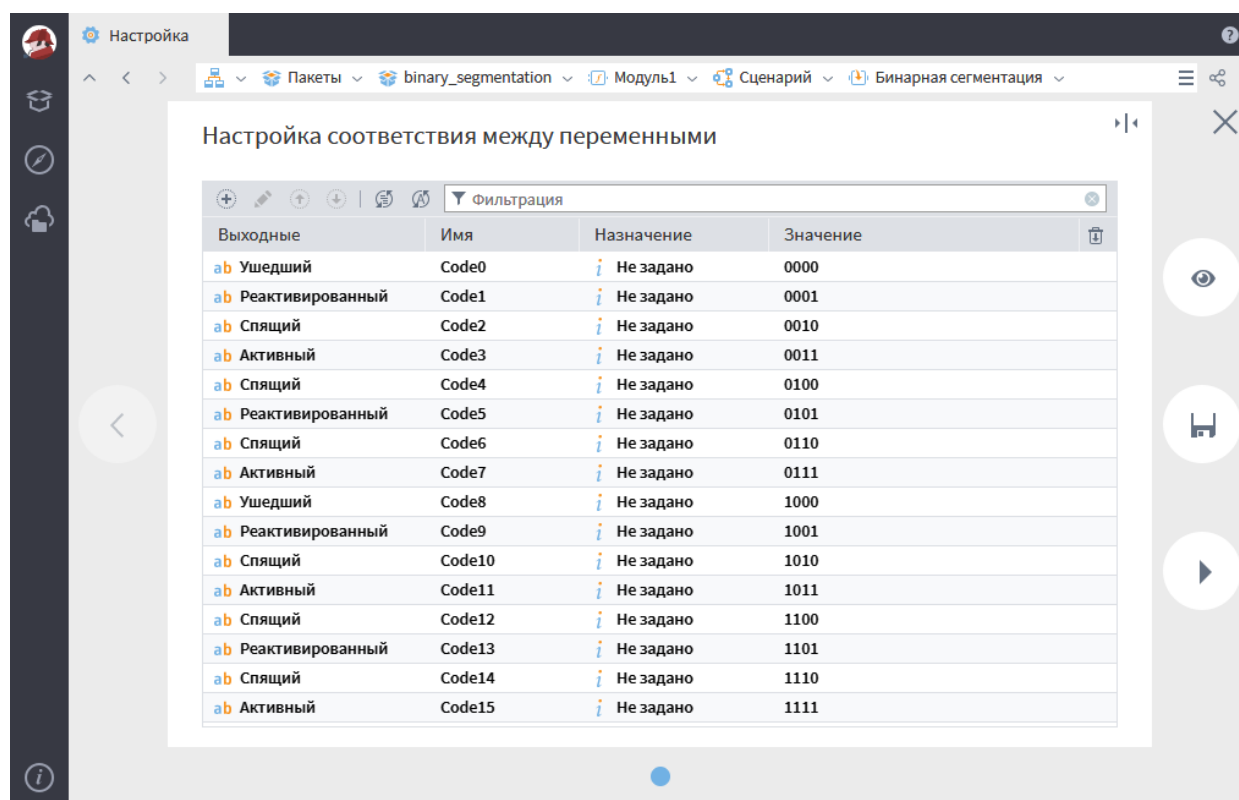


Рисунок 1 — Переменные порта «Сегменты»

### Выход «Набор данных»

Данный набор состоит из пяти полей:

1. **Идентификатор (ID)** — идентификатор клиента;
2. **Код сегмента (Code)** — четырехзначный код сегмента, в который попал клиент;
3. **Сегмент (Segment)** — наименование сегмента;

4. **Вес клиента (Weight)** — значимость клиента. Одинакова для всех клиентов, входящих в один сегмент.

### Выход «Статистики по сегментам»

Здесь выводятся статистики по шестнадцати сегментам, на каждый сегмент приходится одна запись:

1. **Код сегмента (Code)** — четырехзначный код сегмента вида XXXX, где X может принимать значения **1** или **0**.
2. **Сегмент (Segment)** — наименование сегмента;
3. **Численность сегмента (Count)** — количество клиентов в сегменте;
4. **Вес клиента (Weight)** — значимость клиента сегмента;
5. **Вероятность (Probability)** — вероятность того, что клиент совершит визит в периоде, следующем за периодом актуальности;
6. **Кол-во активных в след. периоде (CountNextPeriod)** — количество клиентов в сегменте, сделавших покупку в периоде, следующем за периодом актуальности.

## 3. Разработка сценария

При проектировании сценария компонента рекомендуется для начала определиться с его архитектурой и общим алгоритмом действий, выделить основные части. Необходимо также учитывать возможность использования компонентов, которые были созданы ранее как части других сценариев или библиотек.

Рассмотрим, какие основные действия нам необходимо выполнить для проведения бинарной сегментации.

1. Определение периода, к которому относится дата каждого визита.
2. Определение четырех периодов, для которых будет проводиться сегментация.
3. Формирование набора для сегментации (заполнение пропущенных периодов, отсечение лишних, установка наличия визита в периоде).
4. Определение сегментов и весов для каждого клиента.
5. Расчет статистик по сегментам.

На рис. 2 можно увидеть, как выглядит наш сценарий.

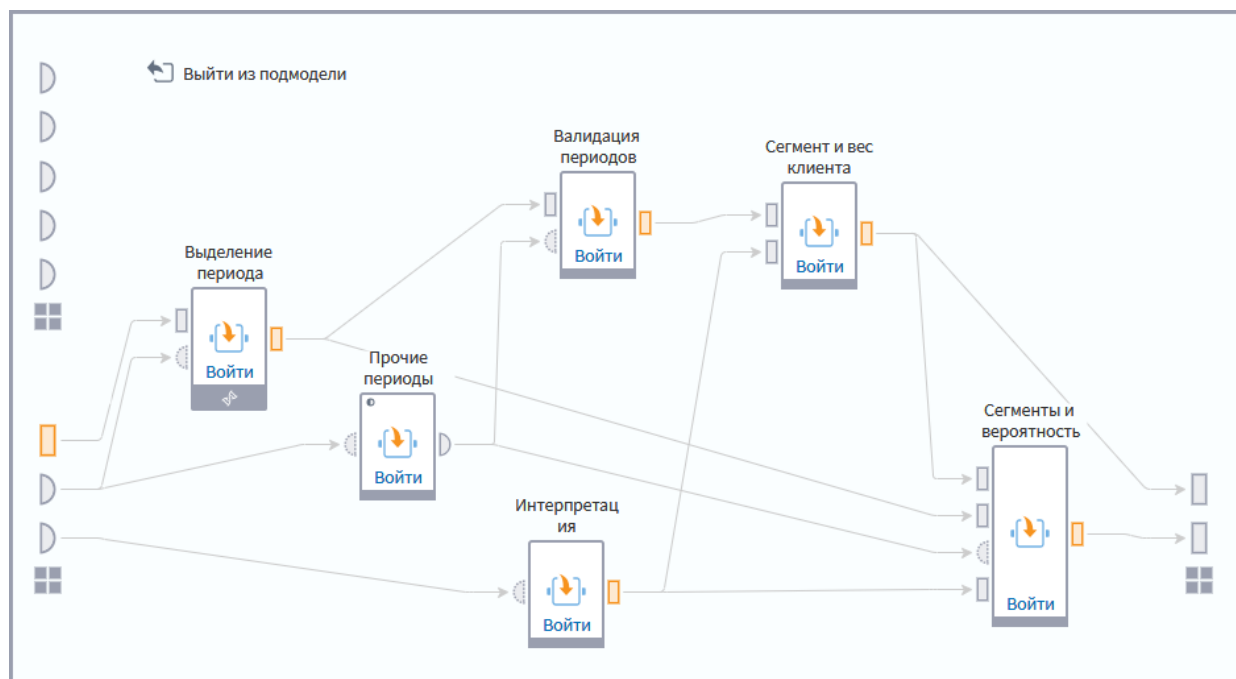


Рисунок 2 — Сценарий бинарной сегментации

Несмотря на то, что в Loginom есть возможность проектирования без данных, лучше использовать небольшой тестовый набор при разработке сценария. Это позволит сразу проверить корректность настроек узлов и соответствие результата обработки нашим ожиданиям. В данном сценарии будем использовать набор **transactions.lgd**, фрагмент которого представлен на рис. 3. Разберем сценарий подробнее.

transactions.lgd • Набор данных • Быс...		
#	ab Идентификатор	7 Дата
1	C0100000199	20.08.2017, 0:00
2	C0100000199	28.06.2017, 0:00
3	C0100000199	29.12.2017, 0:00
4	C0100000343	27.07.2017, 0:00
5	C0100000343	02.02.2017, 0:00
6	C0100000343	12.07.2017, 0:00
7	C0100000343	02.02.2017, 0:00
8	C0100000343	07.09.2017, 0:00
9	C0100000343	13.05.2017, 0:00
10	C0100000375	22.09.2017, 0:00
69 215	C0100000375	02.05.2017, 0:00

[Заккрыть](#)

Рисунок 3 — Фрагмент набора данных transactions.lgd

## 1. Определение периода визита

В большинстве аналитических задач оперируют не конкретными датой и временем, когда произошло событие, а периодом, к которому это событие относится. Для этих целей в открытой библиотеке Loginom Main Library уже имеется компонент **Выделение периода**. Воспользуемся им в нашем сценарии для определения, к какому периоду относится та или иная дата визита клиента.

На вход компонента подается наш набор данных, а также переменные, из которых используется одна — **Тип периода** — позволяющая определить, какими периодами мы будем оперировать при решении задачи. Как уже было сказано выше, по умолчанию мы будем использовать значение *m* — месяц.

На входе набора данных необходимо проставить соответствие полей: *Идентификатор (ID)*–*ID объекта (ObjectID)*, так как в компоненте **Выделение периода** используются абстрактные наименования. На выходе к нашему набору данных добавляется еще одно поле — **Период**.

Подробнее с описанием компонента можно познакомиться в Loginom e-Learning в статье **Выделение периода**<sup>2</sup>.

## 2. Определение периодов для сегментации

Как мы уже знаем, бинарная сегментация проводится по четырем периодам — периоду актуальности и трем предшествующим. Дополнительно мы рассчитываем вероятность визита в следующем периоде, значит, нам нужно получить пять периодов. Для их определения будем использовать дату, на которую проводится сегментация, заданную в переменной **Дата актуальности**.

В сценарии за эту задачу отвечает узел **Прочие периоды**, на вход которого мы подаем переменные с порта **Переменные** внешней подмодели, а на выходе получаем нужные пять периодов, тоже хранимые в переменных. Рассмотрим структуру подмодели подробнее (см. рис. 4).

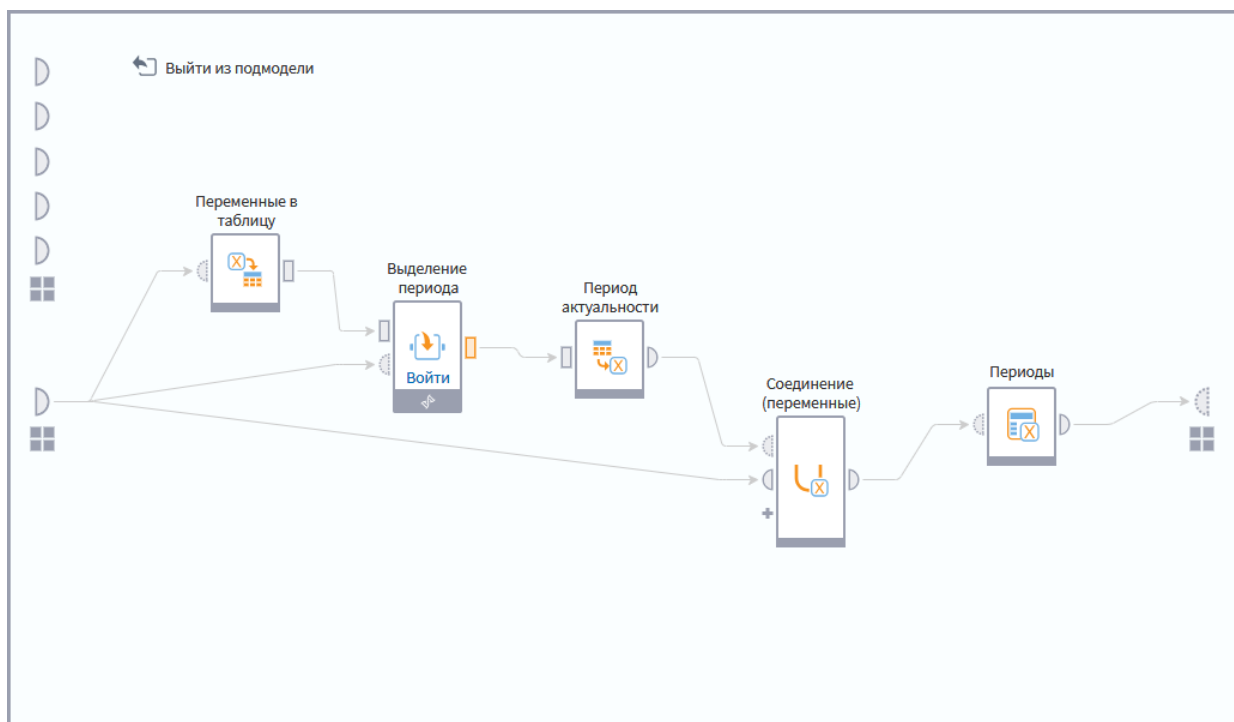


Рисунок 4 — Структура узла «Прочие периоды»

<sup>2</sup> Описание компонента доступно в разделе **Ресурсы-> Библиотеки компонентов->Loginom Main Library->ETL и подготовка данных**



Первые три узла позволяют получить переменную **Период актуальности**: мы делаем из переменной **Дата актуальности** поле, затем снова используем узел **Выделение периода**, чтобы получить период нужного типа, к которому эта дата относится, и переносим его в переменные.

При использовании узла **Выделение периода** в данном случае есть интересный нюанс: по умолчанию на вход узла подается два поля — **ID объекта** строкового типа и **Дата** типа дата/время, но у нас есть только дата. Поэтому в узле **Переменные в таблицу** мы также делаем поле из переменной строкового типа, которая у нас имеется — **Тип периода**, — и затем на входе узла выделения периода указываем ее в качестве соответствия для поля **ID объекта**. Второй вариант — создать пустое поле строкового типа в калькуляторе и использовать его в качестве соответствия, потому что в данном случае идентификатор не является необходимым для расчетов.

Далее мы соединяем все наши переменные в единый набор и с помощью калькулятора рассчитываем три предыдущих и один последующий период относительно периода актуальности, используя формулы добавления заданного количества периодов (*AddMonth()*, *AddWeek()* и т.д. в зависимости от нужного типа периода).

Таким образом, на выходе узла **Прочие периоды** получаем набор из шести переменных (см. рис. 5).







Прочие периоды • Выходные переменные • Быстрый про...			
№	Имя	Метка	Значение
1	 RelevancePeriodM1	Период -1	01.04.2017, 0:00
2	 RelevancePeriodM2	Период -2	01.03.2017, 0:00
3	 RelevancePeriodM3	Период -3	01.02.2017, 0:00
4	 RelevancePeriodP1	Период +1	01.06.2017, 0:00
5	 RelevancePeriod	Период актуальности	01.05.2017, 0:00
6	 PeriodType	Тип периода	m
<a href="#">Заккрыть</a>			

Рисунок 5 — Переменные на выходе узла «Прочие периоды»

### 3. Формирование набора для сегментации

Эта часть сценария (см. рис. 6) заключена в подмодель **Валидация периодов**. Так как мы используем данные по датам визитов клиентов, очевидно, что, если клиент уже приходил ранее, но не совершал визитов в какой-то из исследуемых периодов, записей за этот период по нему не будет. Но для сегментации нам необходимо, чтобы по каждому клиенту было четыре записи — по одной на исследуемый период, — поэтому основная задача этого фрагмента — добавить недостающие периоды и обозначить, что визитов в них не было.

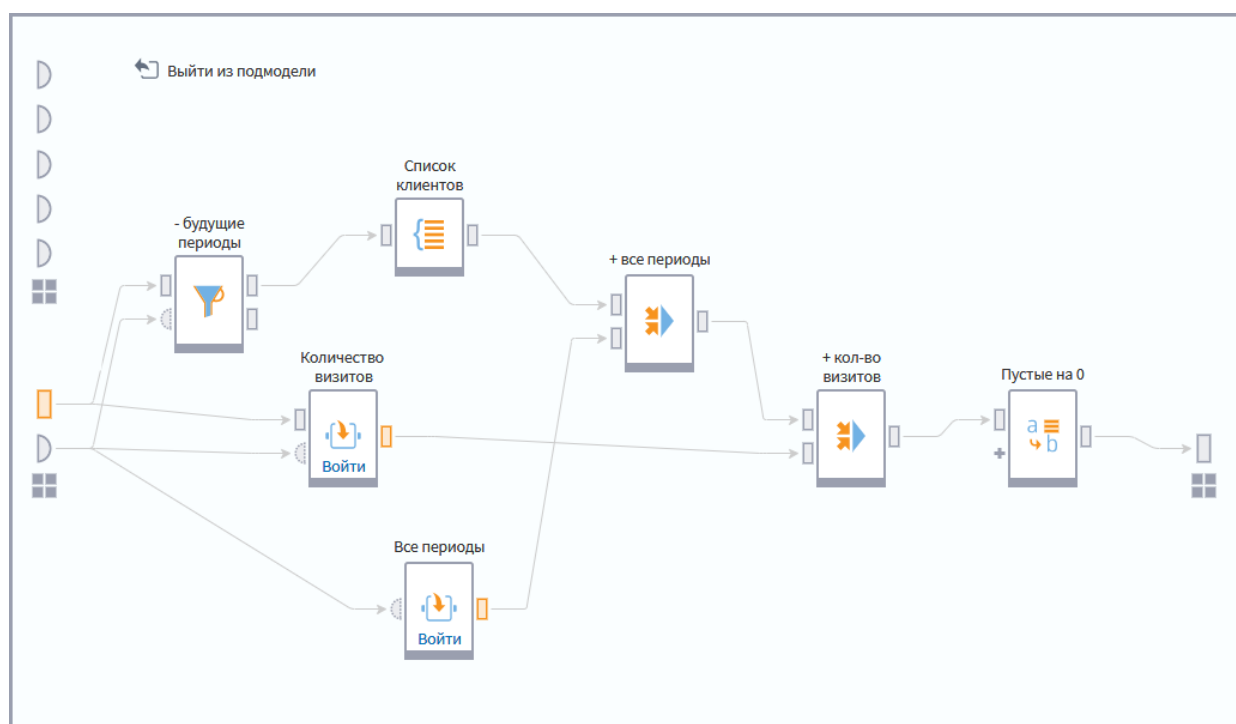


Рисунок 6 — Структура узла «Валидация периодов»

Для начала с помощью фильтра мы отсекаем все периоды после периода актуальности, затем с помощью группировки по полю **Идентификатор** получаем полный список клиентов, которые хоть раз совершили визит до или во время заданного периода.

В узле **Все периоды** мы переносим четыре переменные: **Период -3**, **Период -2**, **Период -1** и **Период актуальности** — в строки таблицы, и столбец **Значения**, содержащий теперь список дат, переименовываем в **Период (Period)**, после чего сортируем его по возрастанию и передаем на выход подмодели. С помощью полного соединения добавляем это поле к списку

клиентов в узле **+ все периоды**, получая таким образом записи по каждому исследуемому периоду для каждого клиента.

Теперь необходимо посчитать количество визитов клиентов, что реализовано в одноименной подмодели (структуру см. на рис. 7). Здесь нам понадобятся только данные за исследуемые периоды, поэтому для начала отфильтруем их. Далее в калькуляторе добавим поле целого типа **Визит (Visit)**, в области кода для которого напомним цифру 1. С помощью узла группировки **Кол-во визитов** подсчитаем количество для каждого клиента в каждом периоде, задав в качестве групп поля **Идентификатор** и **Период**, а в качестве показателя — **Визит** с функцией агрегации *Количество*. Полю на выходе узла оставим имя **Visit** и метку **Визит**.

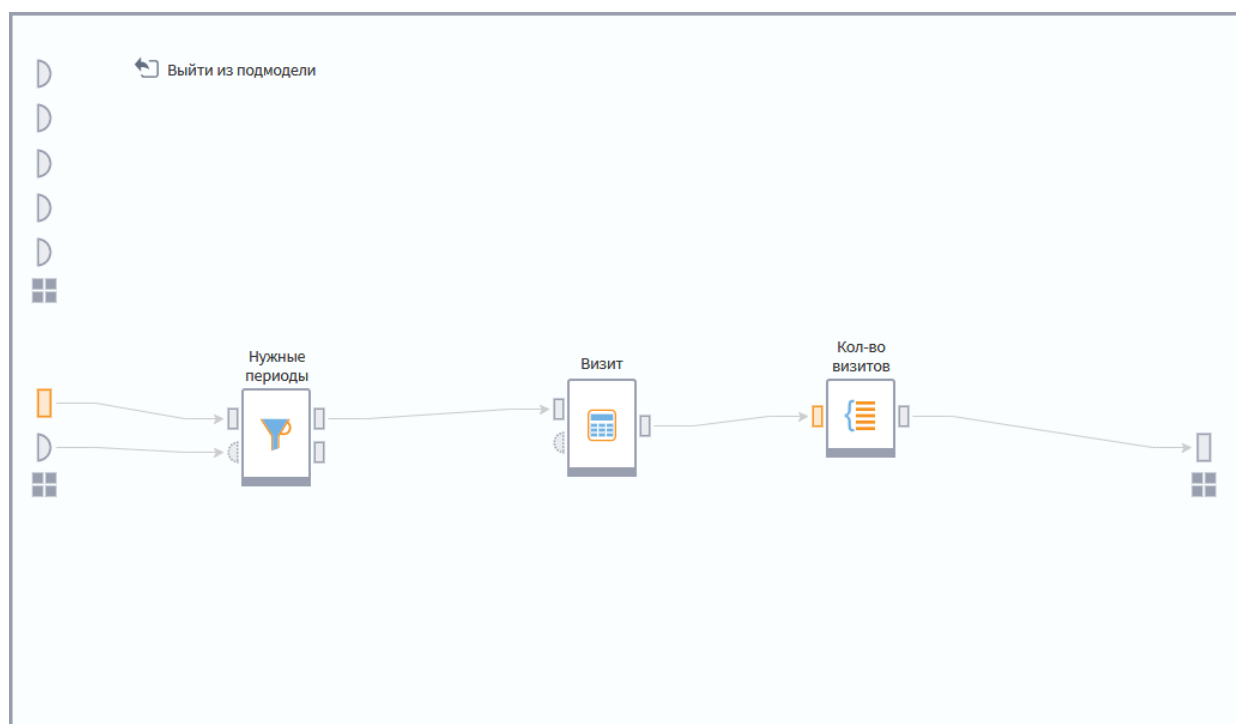


Рисунок 7 — Структура узла «Количество визитов»

Данные с выхода подмодели **Количество визитов** подаются на вход *Присоединяемая таблица* узла слияния **+ кол-во визитов** (см. рис. 6). На вход *Главная таблица* подаются данные с узла **+ все периоды**, используется левое соединение по полям **Идентификатор** и **Период**.

В узле замены пустые значения в поле **Визит** заменяются нулями, а значения больше одного — единицами, так как нас интересует не количество, а

сам факт визита. После всех этих преобразований на выходе подмодели **Валидация периодов** получаем набор данных, готовый для проведения сегментации (см. рис. 8).

Валидация периодов • Выходной набор данных...			
#	Идентификатор	Период	Визит
1	C0100000343	01.02.2017, 0:00	1
2	C0100000343	01.03.2017, 0:00	0
3	C0100000343	01.04.2017, 0:00	0
4	C0100000343	01.05.2017, 0:00	1
5	C0100000375	01.02.2017, 0:00	0
6	C0100000375	01.03.2017, 0:00	0
7	C0100000375	01.04.2017, 0:00	0
8	C0100000375	01.05.2017, 0:00	1
9	C0100000482	01.02.2017, 0:00	0
10	C0100000482	01.03.2017, 0:00	1
38 872	C0100000482	01.04.2017, 0:00	1
<a href="#">Заккрыть</a>			

Рисунок 8 — Набор данных на выходе узла «Валидация периодов»

#### 4. Определение сегментов и весов

Для следующей части сценария, заключенной в подмодель **Сегмент и вес клиента**, нам понадобится информация о названиях сегментов, поэтому для начала кратко рассмотрим узел **Интерпретация**: в нем мы переводим переменные с порта **Сегменты** в строки таблицы, после чего переименовываем поле **Метка** в **Сегмент (Segment)**, а поле **Значение** — в **Код сегмента (Code)**.

Вернемся непосредственно к структуре узла **Сегмент и вес клиента** (см. рис. 9).

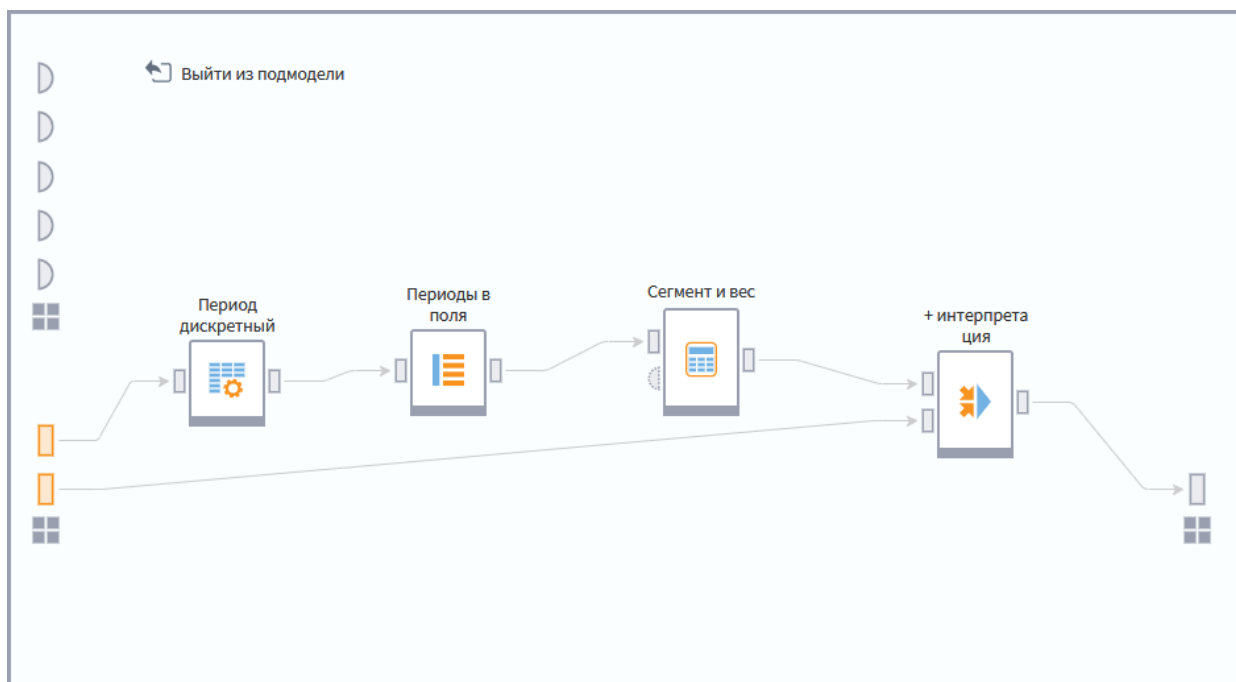


Рисунок 9 — Структура подмодели «Сегмент и вес клиента»

На входе узла у нас список клиентов, по каждому из которых есть четыре записи: период и отметка о визите в этот период. Нам необходимо, чтобы периоды представляли собой не строки, а поля, тогда по каждому клиенту будет только одна запись. Для этого нам понадобится узел **Кросс-таблица**, но, прежде чем использовать его, необходимо изменить вид данных поля **Период**. По умолчанию поле типа дата/время имеет непрерывный вид, но для того чтобы столбец можно было использовать в качестве колонки в настройках кросс-таблицы, он должен быть дискретного вида.

Вид данных меняем с помощью узла **Параметры полей**, которому мы задали метку **Период дискретный**. Далее в настройках узла **Периоды в поля** делаем следующие настройки:

1. **Колонки** — *Период*;
2. **Строки** — *Идентификатор*;
3. **Факты** — *Визит*, функция агрегации — *Минимум*.

Далее с помощью формулы *Concat()* в калькуляторе соединяем значения полей **Период -3**, **Период -2**, **Период -1** и **Период актуальности** в единую строку вида XXXX, где X — либо 0, либо 1, в зависимости от значения в каждом поле. Новый столбец назовем **Код сегмента (Code)**.

Там же создадим поле **Вес клиента (Weight)**, для вычисления которого используется формула:

$$1 * (1\text{-я цифра кода сегмента}) + 2 * (2\text{-я цифра кода сегмента}) + \\ + 4 * (3\text{-я цифра кода сегмента}) + 8 * (4\text{-я цифра кода сегмента}).$$

Используя левое соединение по полю **Код сегмента** в узле слияния **+ интерпретация**, добавим названия сегментов к нашему набору данных. На выходе подмодели **Сегмент и вес клиента** оставим 5 полей (см. рис. 10) и передадим их на первый выход — **Набор данных** — внешней подмодели. По сути, бинарная сегментация на этом завершена — мы получили по каждому клиенту информацию, к какому сегменту он относится. Однако для проведения анализа результатов сегментации нам будут полезны показатели, рассмотренные ранее, поэтому продолжим расчеты.

Сегмент и вес клиента • Выходной набор данных • Быстрый просмотр дан... X				
#	ab Идентификатор	ab Код сегмента	ab Сегмент	12 Вес клиента
1	C0100000343	1001	Реактивированный	9
2	C0100000375	0001	Реактивированный	8
3	C0100000482	0110	Спящий	6
4	C0100000689	0001	Реактивированный	8
5	C0100001116	1001	Реактивированный	9
6	C0100001139	1011	Активный	13
7	C0100001244	1001	Реактивированный	9
8	C0100001405	0000	Ушедший	0
9	C0100001916	0010	Спящий	4
10	C0100002002	0000	Ушедший	0
9 718	C0100002206	1000	Ушедший	1
<a href="#">Заккрыть</a>				

Рисунок 10 — Набор данных на выходе узла «Сегмент и вес клиента»

## 5. Расчет статистик по сегментам

Последний крупный фрагмент сценария заключен в подмодель **Сегменты и вероятность** (см. рис. 11), где проводится расчет статистик по шестнадцати сегментам и вероятности совершения визита клиентом сегмента в

следующем периоде. Для этого на вход подмодели необходимо подать несколько наборов данных с выходов следующих узлов:

1. сегмент и вес клиента;
2. выделение периода;
3. прочие периоды;
4. интерпретация.

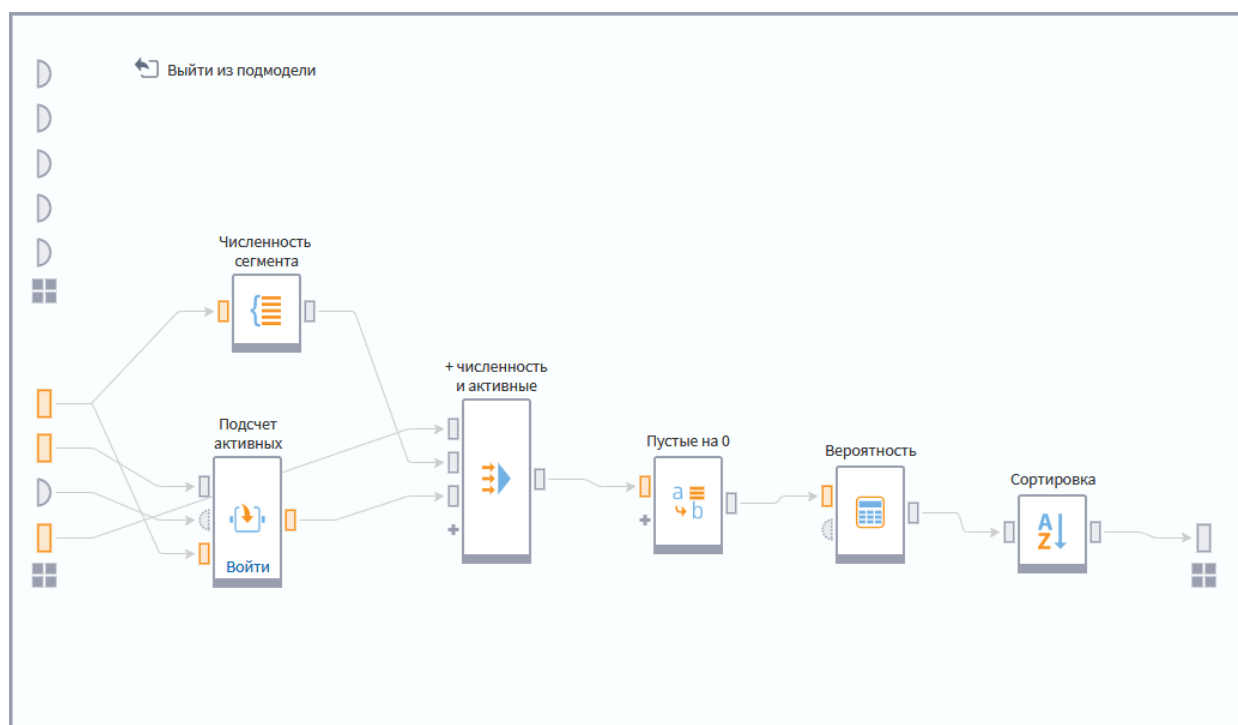


Рисунок 11 — Структура узла «Сегменты и вероятность»

Первые три из этих наборов данных подаются на вход узла **Подсчет активных**. Рассмотрим для начала его внутреннюю структуру (см. рис. 12).

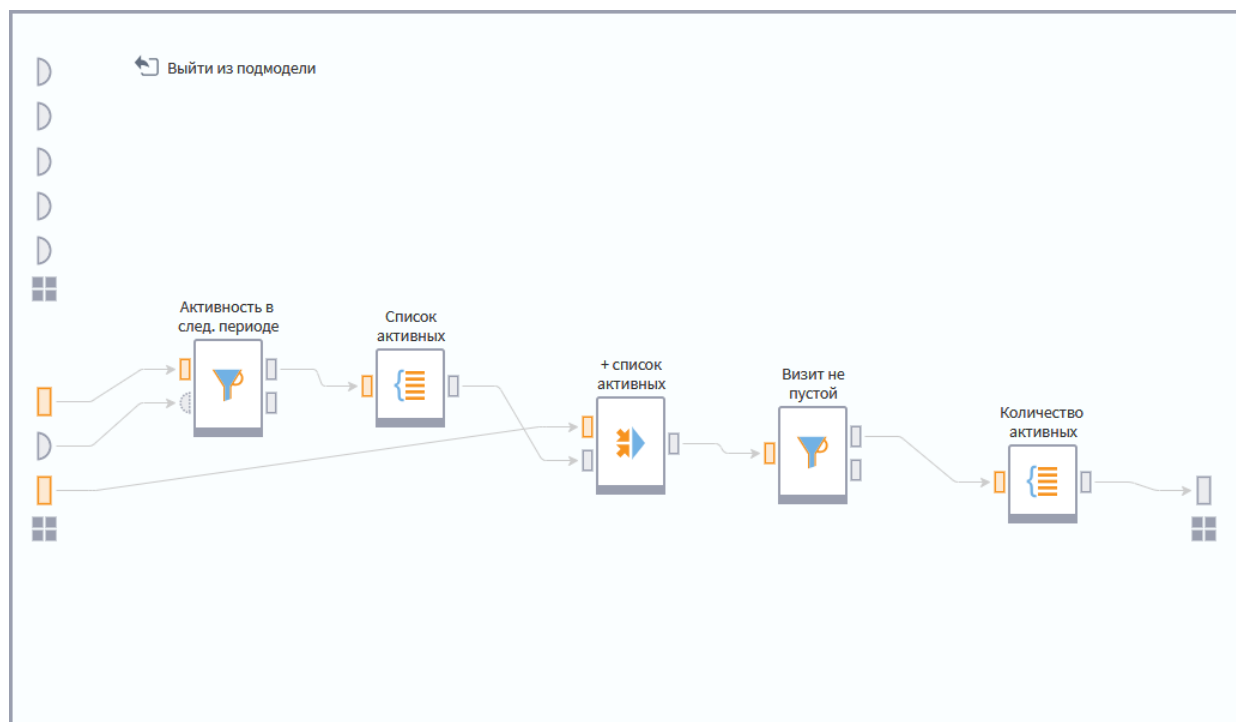


Рисунок 12 — Структура узла «Подсчет активных»

Здесь мы в первую очередь отфильтровываем тех клиентов, которые совершали визиты в период, равный значению переменной **Период +1**, то есть в следующем после периода актуальности. Фильтр используется на наборе данных с узла **Выделение периода**, где есть все исторические сведения о визитах.

Далее в узле группировки **Список активных** добавляем в группы поле **Идентификатор**, а в показатели — **Период** с функцией агрегации *Кол-во уникальных*, которому на выходе задаем имя **Визит** и метку **Visit**. Таким образом мы получим список клиентов, для каждого из которых есть отметка о визите — цифра 1 в соответствующем поле (так как после фильтрации в наборе по каждому клиенту остался только один период).

В результирующем наборе могут оказаться новые клиенты, которые вообще не совершали визиты до периода, следующего за периодом актуальности, они нас в данном случае не интересуют. Поэтому мы подаем набор на вход *Присоединяемая таблица* узла слияния **+ список активных**. На вход *Главная таблица* подаем данные с выхода узла **Сегмент и вес клиента** и используем левое соединение по полю **Идентификатор**.



Теперь у нас есть отметка для тех клиентов, которые совершили визит в следующем периоде. Фильтруем данные по полю **Визит** с условием *не пустое* и получаем набор, состоящий только из таких клиентов. Группируем его по полю **Код сегмента**, а в качестве показателя используем поле **Визит** с функцией агрегации *Количество*, которому зададим имя **Count-NextPeriod** и метку **Кол-во активных в след. периоде**. Получаем список сегментов, в которых совершались визиты в период, следующий за периодом актуальности, с указанием количества клиентов и передаем его на выход подмодели **Подсчет активных**.

Вернемся на один уровень вверх, к структуре подмодели **Сегменты и вероятность** (см. рис. 11). Подсчитаем численность каждого имеющегося сегмента с помощью группировки набора данных с узла **Сегменты и вероятность**: в качестве групп используем поля **Номер пачки** и **Код сегмента**, а в качестве показателей — **Идентификатор** с функцией агрегации *Количество* (выходное поле назовем **Численность сегмента (Count)**) и **Вес клиента** с агрегацией *Минимум*.

В наборах данных на выходах узлов **Численность сегмента** и **Подсчет активных** могут отсутствовать какие-то сегменты: допустим, за исследуемые периоды нет клиентов, которые совершали визиты в каждом, тогда в списке не будет сегмента 1111. Но на выходе бинарной сегментации нам необходима информация по всем сегментам, даже если таких клиентов не было. Здесь нам поможет набор данных с узла **Интерпретация** — в нем содержится полный список кодов с наименованиями сегментов. С помощью узла дополнения данных присоединим к этому набору численность и вес клиента, а также количество клиентов, активных в следующем периоде. Пустые значения, которые могут оказаться в полях **Численность сегмента** и **Кол-во активных в след. периоде** при отсутствии клиентов в сегменте, заменим нулями с помощью узла замены **Пустые на 0**.

Далее рассчитаем вероятность совершения клиентом визита в следующем периоде, формула для расчета выглядит следующим образом:

$$\begin{aligned} \text{Вероятность} &= \text{Доля клиентов, сделавших покупку в след. месяце} = \\ &= \text{Кол-во активных в след. периоде} / \text{Численность сегмента}. \end{aligned}$$

В выражении, используемом в калькуляторе, необходимо учесть, что, если значение в поле **Численность сегмента** равно нулю, вероятность также должна быть равна нулю. В противном случае в созданном поле **Вероятность (Probability)** может получиться значение  $\infty$ .

Отсортируем набор данных по уменьшению значения в поле **Вероятность**, чтобы первыми в списке оказались сегменты, для которых вероятность посещения в следующем периоде наиболее высока, и передадим набор данных на выход узла **Сегменты и вероятность**. Отсюда данные передаются на второй выходной порт внешней подмодели, который носит название **Статистики по сегментам** — и наш сценарий завершен. Полученный набор данных можно увидеть на рис. 13.

Бинарная сегментация • Статистики по сегментам • Быстрый просмотр данных							×
#	ab Код сегмента	ab Сегмент	12 Вес клиента	12 Численность сегмента	12 Кол-во активных в след. периоде	90 Вероятность	
1	1111	Активный	15	707	611	0,86	
2	1110	Спящий	7	160	97	0,61	
3	1101	Реактивированный	11	126	74	0,59	
4	0111	Активный	14	159	89	0,56	
5	1011	Активный	13	129	67	0,52	
6	0000	Ушедший	0	927	373	0,40	
7	1000	Ушедший	1	1 132	374	0,33	
8	0100	Спящий	2	1 398	376	0,27	
9	1100	Спящий	3	377	92	0,24	
10	0110	Спящий	6	365	85	0,23	
11	1010	Спящий	5	314	73	0,23	
12	0010	Спящий	4	1 374	258	0,19	
13	0101	Реактивированный	10	319	59	0,18	
14	0011	Активный	12	296	53	0,18	
15	1001	Реактивированный	9	371	57	0,15	
16	0001	Реактивированный	8	1 564	213	0,14	
							<a href="#">Заккрыть</a>

Рисунок 13 — набор данных на выходе «Статистики по сегментам»

## 4. Резюме

Узел, полученный в результате, изображен на рис. 14. Мы задали ему метку **Бинарная сегментация** — по названию метода, который был реализован. Для использования сегментации в других пакетах, задали узлу открытый модификатор доступа.



Рисунок 14 — Узел «Бинарная сегментация»

В процессе разбора сценария были рассмотрены некоторые общие моменты, которые могут помочь при проектировании. Перечислим их.

1. В первую очередь необходимо изучить методы, которые позволяют решить поставленную задачу, и определиться, какой подход лучше реализовать с учетом возможностей платформы.
2. Если возможно реализовать несколько методов расчета одних и тех же показателей в одном компоненте, и это имеет смысл (например, когда на разных данных следует использовать разные методы, но набор полей можно унифицировать), следует реализовать их и дать возможность задавать метод в переменных.
3. Если для какого-то метода существует словесная интерпретация результата, рекомендуется ее использовать.
4. Прежде, чем переходить непосредственно к созданию сценария, необходимо продумать его структуру и основные элементы.
5. На входе должны задаваться только необходимые параметры, все, что можно рассчитать на их основе, должно рассчитываться внутри компонента.
6. Если какие-то части сценария повторяются несколько раз, необходимо применять механизмы повторного использования. Если эти части используются в других сценариях и пакетах, имеет смысл создать их в качестве внешних компонентов в отдельном пакете — библиотеке компонентов, — которую можно подключать при необходимости.
7. Несмотря на возможность проектирования без данных, рекомендуется использовать небольшой набор при создании сценария для тестирования сделанных настроек узлов.