МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»

Кафедра «Программное обеспечение
автоматизированных систем»

Анализ данных. Часть 2. Классификация и регрессия. Методы машинного обучения

Методические указания к выполнению лабораторных работ
для бакалавров направлений
09.03.03 «Прикладная информатика»,
09.03.04 «Программная инженерия»

Курган 2026

Кафедра: «Программное обеспечение автоматизированных систем»

Дисциплины: «Интеллектуальный анализ данных» (09.03.03), «Методы и алгоритмы анализа данных» (09.03.04)

Составитель: старший преподаватель Ю. В. Адаменко

Утверждены на заседании кафедры «5» декабря 2025 г.

Печатается в соответствии с планом издания, утвержденным методическим советом университета «13» декабря 2025 г.

Наивный байесовский классификатор

Теоретические сведения

В основе NBC (Naïve Bayes Classifier) лежит теорема Байеса:

$$P(c/d) = \frac{P(d/c)P(c)}{P(d)},$$

где P(c / d) — вероятность что документ d принадлежит классу c, именно её нам надо рассчитать;

P(d / c) — вероятность встретить документ d среди всех документов класса c;

P(c) — безусловная вероятность встретить документ класса c в корпусе документов;

P(d) — безусловная вероятность документа d в корпусе документов.

Ее смысл можно выразить следующим образом. Теорема Байеса позволяет переставить местами причину и следствие. Зная, с какой вероятностью причина приводит к некоему событию, эта теорема позволяет рассчитать вероятность того, что именно эта причина привела к наблюдаемому событию.

Цель классификации состоит в том, чтобы понять, к какому классу принадлежит документ, поэтому нам нужна не сама вероятность, а наиболее вероятный класс. Байесовский классификатор использует оценку апостериорного максимума (Maximum a posteriori estimation) для определения наиболее вероятного класса. Можно сказать, что это класс с максимальной вероятностью.

$$c_{map} = \arg\max_{c \in C} \frac{P(d/c)P(c)}{P(d)}.$$

То есть надо рассчитать вероятность для всех классов и выбрать тот класс, который обладает максимальной вероятностью. Обратите внимание, знаменатель (вероятность документа) является константой и никак не может повлиять на ранжирование классов, поэтому в нашей задаче мы можем его игнорировать. Соответственно,

$$c_{map} = \arg\max_{c \in C}[P(d/c)P(c)]. \quad (1)$$

Далее делается предположение условной независимости. Байесовский классификатор представляет документ как набор слов, вероятности которых условно не зависят друг от друга. Этот подход иногда еще называется bag of words model. Исходя из этого предположения условная вероятность документа аппроксимируется произведением условных вероятностей всех слов входящих в документ.

$$P(d/c) \approx P(w_1/c)P(w_2/c)...P(w_n/c) \approx \prod_{i=1}^{n} P(w_i/c). \quad (2)$$

Этот подход также называется Unigram Language Model.

Подставив полученное выражение (2) в формулу (1), мы получим:

$$c_{map} = \arg\max_{c \in C}[\prod_{i=1}^{n} P(w_i / c)P(c)] \quad . \qquad (3)$$

Проблема арифметического переполнения

При достаточно большой длине документа придется перемножать большое количество очень маленьких чисел. Для того чтобы при этом избежать арифметического переполнения снизу, зачастую пользуются свойством логарифма произведения log ab = log a+log b. Переписываем формулу (3) с использованием логарифма:

$$c_{map} = \arg\max_{c \in C}[\sum_{i=1}^{n} \log P(w_i / c) + \log P(c)] \quad . \qquad (4)$$

Основание логарифма в данном случае не имеет значения. Вы можете использовать как натуральный, так и любой другой логарифм.

Оценка параметров Байесовской модели

Оценка вероятностей Pc и $P(w_i/c)$ осуществляется на обучающей выборке. Вероятность класса мы можем оценить как:

$$P(c) = \frac{D_c}{D} \quad , \qquad (5)$$

где Dc – количество документов принадлежащих классу c, а D – общее количество документов в обучающей выборке.

Оценка вероятности слова в классе может делаться несколькими путями. Например, по multinomial bayes model (11.6):

$$P(w_i / c) = \frac{W_{ic}}{\sum_{i'' \in V} W_{i'c}} \quad , \qquad (6)$$

W_{ic} – количество раз сколько i-ое слово встречается в документах класса c;

V – словарь корпуса документов (список всех уникальных слов).

Другими словами, числитель описывает, сколько раз слово встречается в документах класса (включая повторы), а знаменатель – это суммарное количество слов во всех документах этого класса.

Проблема неизвестных слов

По формуле (6), если на этапе классификации вам встретится слово, которого вы не видели на этапе обучения, то значения W_{ic} , а следовательно и $P(w_i / c)$ будут равны нулю. Это приведет к тому, что документ с этим словом нельзя будет классифицировать, так как он будет иметь нулевую вероятность по всем классам. Типичным решением проблемы неизвестных слов является аддитивное сглаживание (сглаживание Лапласа). Идея заключается в том, что мы предполагаем, будто видели каждое слово на один раз больше, то есть прибавляем единицу к частоте каждого слова.

$$P(w_i / c) = \frac{W_{ic} + 1}{\sum_{i'' \in V} (W_{i'c} + 1)} = \frac{W_{ic} + 1}{|V| + \sum_{i'' \in V} W_{i'c}} \quad . \qquad (7)$$

Подставив выбранные нами оценки в формулу (4), получаем окончательную формулу (8), по которой происходит байесовская классификация.

$$c_{map} = \arg\max_{c \in C} [\sum_{i=1}^{n} \log \frac{W_{ic} + 1}{|V| + \sum_{i'' \in V} W_{i'c}} + \log \frac{D_c}{D}] \quad . \qquad (8)$$

Реализация классификатора

Для реализации Байесовского классификатора необходима обучающая выборка, в которой проставлены соответствия между текстовыми документами и их классами. Затем нам необходимо собрать следующую статистику из выборки, которая будет использоваться на этапе классификации:

− относительные частоты классов в корпусе документов, то есть как часто встречаются документы того или иного класса;

− суммарное количество слов в документах каждого класса;

− относительные частоты слов в пределах каждого класса;

− размер словаря выборки. Количество уникальных слов в выборке.

Совокупность этой информации мы будем называть моделью классификатора. Затем на этапе классификации необходимо для каждого класса рассчитать значение выражения (9) и выбрать класс с максимальным значением.

$$\sum_{i \in Q}^{n} \log \frac{W_{ic} + 1}{|V| + L_c} + \log \frac{D_c}{D}] \quad , \qquad (9)$$

где

Dc – количество документов в обучающей выборке, принадлежащих классу c;

D – общее количество документов в обучающей выборке;

|V| – количество уникальных слов во всех документах обучающей выборки;

Lc – суммарное количество слов в документах класса c в обучающей выборке;

Wic— сколько раз i-е слово встречалось в документах класса c в обучающей выборке;

Q – множество слов классифицируемого документа (включая повторы).

Формирование вероятностного пространства

В простейшем случае вы выбираете класс, который получил максимальную оценку. Но если вы, например, хотите помечать сообщение как спам, только если соответствующая вероятность больше 80 %, то сравнение логарифмических оценок вам ничего не даст. Оценки, которые выдает данный алгоритм не удовлетворяют двум формальным свойствам, которым должны удовлетворять все вероятностные оценки:

− они все должны быть в диапазоне от нуля до единицы;

− их сумма должна быть равна единице.

Для того чтобы решить эту задачу, необходимо из логарифмических оценок сформировать вероятностное пространство. А именно: избавиться от логарифмов и нормировать сумму по единице.

$$P(c/d) = \frac{e^{q_c}}{\sum_{c' \in C} e^{q_{c'}}} \quad (10)$$

Здесь qc — это логарифмическая оценка алгоритма для класса c, а возведение e (основания натурального логарифма) в степень оценки используется для того чтобы избавиться от логарифма ($a^{\log_a x} = x$). Таким образом, если вы в расчетах использовали не натуральный логарифм, а десятичный, вам необходимо использовать не e, а число 10.

2 Задание и порядок выполнения работ

Задание. Используя Excel или Calc, проведите классификацию твитов по принадлежности к Madrill.

Исходные данные находятся в файле «Mandrill.xlsx».

В нем содержатся данные о твитах, содержащих упоминание слова Mandrill. На первом листе (AboutMandrillApp) твиты относится к сервису Mandrill, на втором листе (AboutOther) – нет. Третий лист (TestTweets) содержит тестовый набор твитов.

Mandrill — это «дочерний» сервис MailChimp, предназначенный для отправки транзакционных писем, т. е. писем, уведомляющих пользователя об определённых событиях: регистрации, смене пароля, оформлении заказа, оплате счетов и т. п.

Ход работы.

Для выполнения работы выполните следующие этапы.

1 Осуществите токенизацию твитов на листах AboutMandrillApp и AboutOther. Для этого:

1.1 Преобразуйте все буквы в строчные в столбце B2. (функция СТРОЧН)

1.2 Уберите лишнюю пунктуацию, для этого последовательно в ячейках C2-H2 замените знаки «. », «: », «?», «!», «;» и «,» на пробелы. (функция ПОДСТАВИТЬ). При этом обратите внимание точку и двоеточие нужно искать с пробелом после них.

1.3 Разделите каждый твит на токены:

1.3.1 Создайте два новых листа AppTokens и OtherTokens.

1.3.2 Предполагаем, что каждый твит содержит не более 30 слов. Соответственно, вам нужно 30*150=4500 строк, для того чтобы записать все слова. Озаглавьте ячейку A1 Tweet, выделите диапазон A2:A4501 и с помощью специальной вставки вставьте значения твитов из соответствующего столбца H.

1.3.3 В столбце B нужно найти позиции пробелов в твитах. Назовите стобец Space Position. Введите для первых 150 твитов начальное значение 0. Далее (начиная с повтора набора твитов, строка 152) рассчитайте положение следующего пробела, при этом предусмотрите проверку на ошибку для случая,

если твит содержкит менее 30 слов. Например, для B152 формула будет иметь вид:

=ЕСЛИОШИБКА(НАЙТИ(" ";A152;B2+1);ДЛСТР(A152)+1)

1.3.4 В столбце C (Token) извлеките токены с помощью фукнции ПСТР. Предусмотрите проверку на ошибку для коротких твитов. Если ошибка есть, замените этот токен, например на «.». Например, для C2 формула будет иметь вид:

=ЕСЛИОШИБКА(ПСТР(A2;B2+1;B152-B2-1);".")

1.3.5 В стоблце D (Length) посчитайте длину токена (функция ДЛСТР).

Вид листа после выполнения этой части задания представлен на рисунке 1.

	A	B	C	D
1	Tweet	Space Position	Token	Length
26	@icntmx yep we'd be glad to would you mind submitting a request at http://help.mandrill.co	0	@icntmx	7
27	@jeremyweir if you submit a request at http://help.mandrill.com we'll get back to you with s	0	@jeremywei	11
28	@josscrowcroft mind submitting a request via http://help.mandrill.com with some additional	0	@josscrowcr	14
29	@juanpabloaj official clients have inline doc info but our support team can help with example	0	@juanpablo	12
30	@kanonbulle no issues delivering to the hotmail domain currently mind submitting a request	0	@kanonbull	11
31	@kennydude @devongovett yeah mandrill is well worth a look.	0	@kennydud	10
32	@kennyfraser already cleaned up client & delisted ip but #enginehosting support can share s	0	@kennyfrase	12
33	@khiger вот сервис http://mandrill.com/	0	@khiger	7
34	@ljharb when i looked last year mandrill's pricing and api was a bit confusing but it now seer	0	@ljharb	7
35	@mandrill realised i did that about 5 seconds after hitting send	0	@mandrill	9
36	@mandrillapp could you add the defaults (if any) to your smtp header docs http://help.mand	0	@mandrillap	12
37	@mandrillapp increases scalability (http://bit.ly/14myvuh) then decreases pricing (http://bi	0	@mandrillap	12
38	@mandrillapp there are some issues with your mandrill npm module what's your preferred wa	0	@mandrillap	12
39	@mandrillapp tried refreshing the link (line 184 on the homepage) goes here http://help.man	0	@mandrillap	12
40	@mandrillapp we cannot even find out the last email sent to as mandrill page crashes when e	0	@mandrillap	12
41	@mandrillapp yeap that's what i meant throttling - throttling before it gets to mandrill	0	@mandrillap	12
42	@manojranaweera looks like bulk increasing volume considerably our support team can help v	0	@manojrana	15
43	@marcelosomers @nathansmith fwiw we dumped postmark in favor of mandrill highly recom	0	@marceloso	14
44	@masuga use a service like mandrill not ee mail.	0	@masuga	7
45	@matt_pickett if u want to reach out to other mailchimp/mandrill users try http://awe.sm/r0j	0	@matt_pick	13

Рисунок 1 – Вид листа AppToken

2 Рассчитайте условную вероятность каждого токена.

2.1 Выделите на листе AppTokens область с токенами длиной (C1:D4501) и создайте сводную таблицу на лист AppTokensProbabality. В конструкторе сводных таблиц отфильтруйте токены по длине более 3 (чтобы избавиться от коротких несмысловых токенов) и расположите их по горизонтали. Затем в окне значений установите значение для подсчета количества каждого токена. В результате вы получите перечень всех токенов в твитах с указанием их количества.

2.2 Осуществите дополнительное сглаживание. Для этого в столбце C (Заголовок Add One To Everything) прибавьте к количеству каждого токена единицу. Найдите сумму числа токенов после сглаживания (после последнего числа токенов).

2.3 В столбце D (название P(Token|App) рассчитайте вероятность токена по формуле:

$$P_i = \frac{N_i}{\sum_i N_i},$$

где Ni – количество i-го токена;

Pi – вероятность i-го токена

2.4 Найдите в столбце E (название — LN(P)) натуральные логарифмы вероятностей. Вид листа AppTokensProbabality после расчета вероятностей представлен на рисунке 2.

	A	B	C	D	E
1	length	(Multiple Items)			
2					
3	Count of token				
4	Row Labels	Total	Add One To Eve	P(Token|App)	LN(P)
5	friday	1	2	0,000829876	-7,094234846
6	‘migrate’	1	2	0,000829876	-7,094234846
7	“mandrill”	1	2	0,000829876	-7,094234846
8	(@mandrillapp)	1	2	0,000829876	-7,094234846
9	(0.0.3)	1	2	0,000829876	-7,094234846
10	(0.0.4)	1	2	0,000829876	-7,094234846
11	(1.0.19)	1	2	0,000829876	-7,094234846
12	(1.0.25)	1	2	0,000829876	-7,094234846
13	(confirmação	1	2	0,000829876	-7,094234846
14	(http://j.mp/10tohxc	1	2	0,000829876	-7,094234846
15	(http://j.mp/10tohxg	1	2	0,000829876	-7,094234846
16	(i.e	1	2	0,000829876	-7,094234846
17	(ils	1	2	0,000829876	-7,094234846
18	(line	1	2	0,000829876	-7,094234846
19	(one-to-one	1	2	0,000829876	-7,094234846
20	(some	1	2	0,000829876	-7,094234846
21	(their	1	2	0,000829876	-7,094234846

Рисунок 2 – Вид листа AppTokensProbabality

2.5 Аналогично получаем лист OtherTokensProbabality, по твитам, не имеющим отношения в Mandrill.com.

Таким образом вы получите модель наивного байесовского классификатора, содержащую две таблицы с условными вероятностями.

3 Тестирование модели

3.1 В столбцах D – J осуществите обработку текста твитов (аналогично п.1.1, 1.2).

3.2 Для этого создайте лист TestPredictions и вставьте в него столбцы Number и Class из TestTweets. Столбец C назовите Prediction. В нем будете размещать предполагаемые значения классов. В столбец D скопируйте обработанные твиты с листа TestTweets.

3.3 Извлеките токены. Так как, в отличие от таблиц вероятностей, нет необходимости комбинировать токены по всем твитам, токенизацию можно осуществить боле простым способом.

Выделите твиты D2:D21 и выберите «Текст по столбцам» в меню «Данные». В Мастере текстов выберите «С разделителями», в качестве разделителей выберите знаки табуляции и пробела, «Считать последовательные разделители одним», ограничитель строк установите на «нет». В результате твиты будут разбросаны по столбцам в виде отдельных токенов.

3.4 Рассчитайте вероятности отношения токенов к приложению Mandrill и к другим.

3.4.1 Начиная со столбца D, строка 25, рассчитайте вероятности принадлежности токенов к приложению Mandrill. Используйте функцию ВПР. Вы должны найти соответствующий токен с листа TestPredictions на листе AppTokensProbabality в столбце A и взять значение соответствующей вероятности из столбца E.

Но важно обработать следующие условия:

- вероятность редких слов (отсутствующих в классификаторе) необходимо принять равной LN(1/общее число токенов на листе AppTokensProbabality), используйте функцию ЕНД;

- вероятность коротких токенов (число знаков меньше либо равно 3) следует принять равной 0.

Например, для ячейки D25 формула будет иметь вид:

=ЕСЛИ(ДЛСТР(D2)<=3;0;ЕСЛИ(ЕНД(ВПР(D2;$AppTokensProbability.$A$5:$E$827;5;0));
LN(1/$AppTokensProbability.$C$828);ВПР(D2;$AppTokensProbability.$A$5:$E$827;5;0)))

3.4.2 Просуммируйте в столбце С вероятности по каждой строке.

Вид листа TestPredictions после выполнения этих операций представлен на рисунке 3.

	A	B	C	D	E	F	G	H	I	J	K
1	Number	Class	Prediction	Tokens							
2	1	APP		just	love	@mandrillap	transactiona	email	service	-	http://manc
3	2	APP		@rossdeane	mind	submitting	a	request	at	http://help.r	with
4	3	APP		@veroapp	any	chance	you'll	be	adding	mandrill	support
5	4	APP		@elie__	@camj59	jparle	de	relai	smtp	1	million
6	5	APP		would	like	to	send	emails	for	welcome	password
7	6	APP		from	coworker	about	using	mandrill	"i	would	entrust
8	7	APP		@mandrill	realised	i	did	that	about	5	seconds
9	8	APP		holy	shit	it's	here	http://www.mandrill.com/			
10	9	APP		our	new	subscriber	profile	page	activity	timeline	aggregate
11	10	APP		@mandrillap	increases	scalability	(	http://bit.ly/)		then	decreases
12	11	OTHER		the	beets	rt	@missmya	#nameanam	mandrill		
13	12	OTHER		rt	@luissand0v	fernando	vargas	mandrill	mexican	pride	mma
14	13	OTHER		photo	oculi-ds	mandrill	by	natalie	manuel	http://tmblr.co/zjqanxhd	
15	14	OTHER		@mandrill	me	neither	we	can	be	:sadpanda	together
16	15	OTHER		@mandrill	n	/	(	k	*	(	n
17	16	OTHER		megaman	x	-	spark	mandrill	acapella	http://youtu	@youtube
18	17	OTHER		@angeluser	storm	eagle	ftw	nomás	no	dejes	que
19	18	OTHER		gostei	de	um	vídeo	@youtube	http://youtu	aspark	...
20	19	OTHER		what	is	2-year-old	mandrill	jj	thinking	in	this
21	20	OTHER		120	years	of	moscow	zoo	-	mandrill	-

Рисунок 3 – Вид листа TestPredictions после расчета вероятностей твитов принадлежности к Madrill

3.4.3 Начиная с ячейки D48, расчитайте вероятности отношения токенов к другим объектам.

3.5 Проклассифицируйте твиты в столбце С (диапазон С1:С21). Нужно сравнить полученные вероятности принадлежности для Madrill и не Madrill. Наибольшее значение вероятности относит твит к соответствующему классу. Используйте фукнцию ЕСЛИ. Принадлежность к Madrill обозначьте APP, принадлежность к другим — OTHER. Пример результата представлен на рисунке 11.4.

Сделайте выводы.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Number	Class	Prediction	Tokens									
2	1	APP	APP	just	love	@mandrillap	transactional	email	service	-	http://mand	sorry	@sendgrid
3	2	APP	APP	@rossdeane	mind	submitting	a	request	at	http://help.n	with	account	details
4	3	APP	APP	@veroapp	any	chance	you'll	be	adding	mandrill	support	to	vero
5	4	APP	APP	@elie__	@camj59	jparle	de	relai	smtp	1	million	de	mail
6	5	APP	APP	would	like	to	send	emails	for	welcome	password	resets	payment
7	6	APP	APP	from	coworker	about	using	mandrill	"i	would	entrust	email	handling
8	7	APP	APP	@mandrill	realised	i	did	that	about	5	seconds	after	hitting
9	8	APP	APP	holy	shit	it's	here	http://www.mandrill.com/					
10	9	APP	APP	our	new	subscriber	profile	page	activity	timeline	aggregate	engagement	stats
11	10	APP	APP	@mandrillap	increases	scalability	(	http://bit.ly/)		then	decreases	pricing	(
12	11	OTHER	OTHER	the	beets	rt	@missmya	#nameanam	mandrill				
13	12	OTHER	OTHER	rt	@luissand0v	fernando	vargas	mandrill	mexican	pride	mma		
14	13	OTHER	OTHER	photo	oculi-ds	mandrill	by	natalie	manuel	http://tmblr.co/zjqanxhdswlr			
15	14	OTHER	OTHER	@mandrill	me	neither	we	can	be	:sadpanda	together	:(	
16	15	OTHER	OTHER	@mandrill	n	/	(	k	*	(	n	-	k
17	16	OTHER	OTHER	megaman	x	-	spark	mandrill	acapella	http://youtu	@youtubeさんから		
18	17	OTHER	OTHER	@angeluserr	storm	eagle	ftw	només	no	dejes	que	se	le
19	18	OTHER	OTHER	gostei	de	um	vídeo	@youtube	http://youtu	aspark	...	mandrill's	stage
20	19	OTHER	APP	what	is	2-year-old	mandrill	jj	thinking	in	this	pic	http://ow.ly/
21	20	OTHER	OTHER	120	years	of	moscow	zoo	-	mandrill	-	nocta	cccp
22													
23													
24			Sum of conditional probabilities										
25	1		-65,538186	-5,302475	-6,6887697	-5,1483247	-5,3024754	-4,4915452	-5,3024754	0	-5,8414719	-7,0942348	-6,1779441
26	2		-74,482537	-7,094235	-5,3024754	-5,3894868	0	-4,9541687	0	-4,6518878	-4,1764641	-5,5901574	-5,1483247
27	3		-44,882623	-7,094235	0	-7,0942348	-7,0942348	0	-7,0942348	-3,2335051	-6,1779441	0	-7,0942348
28	4		-109,77229	-6,68877	-5,9956226	-7,0942348	0	-7,0942348	-5,7079405	0	-6,6887697	0	-6,1779441
29	5		-82,76288	-5,995623	-5,9956226	0	-5,1483247	-5,3894868	0	-7,0942348	-7,0942348	-7,0942348	-7,0942348
30	6		-58,475042	-5,590157	-7,0942348	-5,4847969	-5,3024754	-3,2335051	0	-5,9956226	-7,0942348	-4,4915452	-7,0942348
31	7		-50,883523	-6,68877	-7,0942348	0	0	-5,5901574	-5,4847969	0	-7,0942348	-6,6887697	-7,0942348
32	8		-34,149418	-7,094235	-7,0942348	-7,0942348	-6,1779441	-6,6887697	0	0	0	0	0
33	9		-80,353798	0	0	-7,0942348	-7,0942348	-6,1779441	-7,0942348	-7,0942348	-7,0942348	-7,0942348	-7,0942348

Рисунок 4 – Вид листа TestPredictions после расчета вероятностей принадлежности твитов к Madrill

3 Искусственные нейронные сети

1 Искусственные нейронные сети

В продуктах Data Mining для решения задач классификации и регрессии используют так называемые нейронные сети прямого распространения (их также называют многослойным персептроном). Такая сеть состоит из совокупности узлов (нейронов), соединенных между собой связями. Каждый узел является своеобразным обрабатывающим модулем. Все связи имеют определенный вес (числовой параметр), а ориентация соединяющих линий соответствует пути прохождения сигнала (рисунок 5).

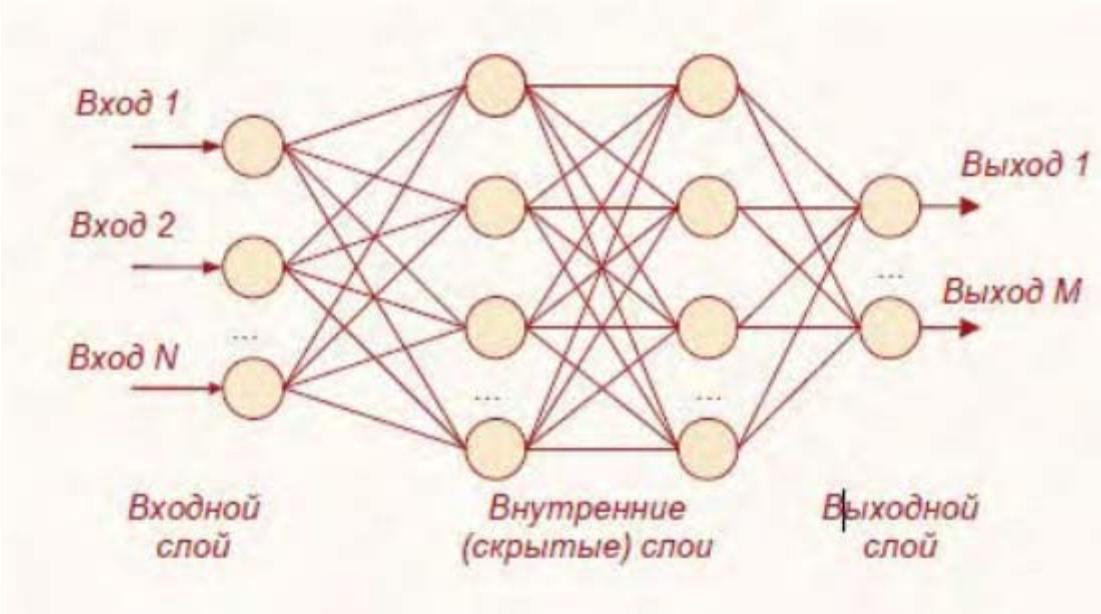


Рисунок 5 – Многослойный персептрон

Существует три типа узлов: входной, скрытый и выходной. Входные узлы формируют первый слой сети. В большинстве НС каждому из них соответствует один входной атрибут (возраст, пол, доход и т. д.). Перед обработкой исходное

значение входного признака должно быть отмасштабировано (чаще всего в диапазоне от – 1 до 1).

Скрытые узлы расположены в промежуточных слоях; они получают входные сигналы от узлов предыдущего слоя, производят с ними определенные вычисления и результат обработки передают на вход узлов следующего слоя. Каждый нейрон скрытого слоя соединен со всеми нейронами предыдущего. Наличие скрытого слоя крайне важно, поскольку это позволяет моделировать нелинейные зависимости между входами и выходами сети.

Выходные узлы соответствуют зависимым (предсказываемым) переменным модели; на выходе сети формируется вещественное число в диапазоне от 0 до 1. В принципе НС может иметь несколько выходных узлов, однако почти всегда ее можно представить как совокупность сетей с одним выходом.

В режиме прогнозирования НС работает довольно просто: поступающие сигналы подаются на входы и «прогоняются» через сеть, в результате чего на выходе генерируется рассчитанное значение; затем оно подвергается денормализации в исходное значение (для непрерывных) или в исходное состояние (для дискретных атрибутов).

После того как архитектура нейронной сети сформирована (задано число слоев и нейронов в каждом слое), запускается процесс обучения сети; суть его состоит в нахождении оптимальных значений весовых коэффициентов. Это сложный вычислительный процесс, который может длиться довольно долго. На его начальном этапе в качестве весов используют случайные числа. Затем на каждой итерации все примеры из обучающей выборки «прогоняют» через сеть с текущими весовыми коэффициентами, определяют выходные значения и величину ошибок. На основе информации об ошибках, по специальному правилу (которое зависит от выбранного алгоритма обучения) веса, НС корректируют, добиваясь все более точной ее работы.

Здесь необходимо более подробно рассмотреть понятие активационной функции, получившей свое название от биологического термина «активация» (возбуждение нервной клетки). Каждый нейрон в НС, как указывалось ранее, представляет собой элементарный блок, который суммирует входные сигналы и генерирует на их основе выходной (аналогичный уровню возбуждения в биологии).

Суммирование чаще всего осуществляется путем расчета средней взвешенной (линейной комбинации входных сигналов и их весов), а для определения выходного значения как раз и используется активационная функция.

Существует несколько несложных аналитических функций, удовлетворяющих данному требованию. Чаще всего в НС используют две из них:

– сигмоиду – $f = 1/(1 + e^{a})$ (рисунок 7.2);

– гиперболический тангенс – $f = th(a) = (e^{a} – e^{-a})/ (e^{a} + e^{-a})$ (рисунок 7.3);

где a – параметр крутизны функции, f – выходное значение.

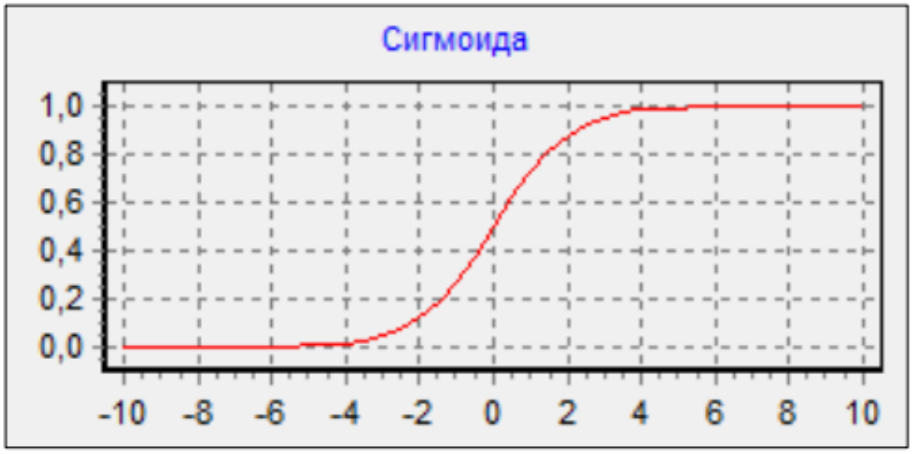


Рисунок 6 – Сигмоида

Рисунок 7 – Гиперболический тангенс

Важнейший вопрос, возникающий при построении нейросетевой модели, – определение архитектуры НС, в частности количества скрытых слоев и нейронов в них. При его решении рекомендуется руководствоваться следующими правилами:

1 Количество скрытых слоев в большинстве случаев не должно быть больше двух.

2 Если две обученные нейросети имеют одинаковый порядок ошибок обучения и обобщения, предпочтение следует отдать более простой (содержащей меньше скрытых слоев и нейронов).

3 Количество примеров обучающей выборки должно быть в 1,5–2 раза больше числа связей (весов).

4 Количество нейронов в скрытых слоях можно приблизительно рассчитать по формуле $c\sqrt{mn}$, где n – число входных нейронов, m – число выходов сети, c – константа (по умолчанию c = 4).

5 Другой существенный вопрос – определение момента, когда обучение сети следует прекратить. Проблема состоит в том, что слишком долгое обучение может привести к адаптации параметров НС (весов) к любым нерегулярностям в обучающих данных (так называемое переобучение сети).

Истинная цель обучения сети состоит в таком подборе ее архитектуры и параметров, которые обеспечат минимальную погрешность распознавания тестового множества данных, не участвовавших в обучении.

2 Пример работы многослойного персептрона

Рассмотрим решение задачи регрессии с помощью многослойного персептрона на примере прогнозирования результата умножения двух чисел.

1) для этого потребуется файл multi.txt.

В файле содержится таблица со следующими полями: Аргумент1, Аргумент2 – множители, Произведение – их произведение. Импортировав данные из файла, можно посмотреть результат умножения, используя визуализатор «Таблица» (рисунок 8).

Таблица

1 / 63

аргумент 1	аргумент 2	произведение
0	1	0
1	0	0
0	3	0
3	0	0
0	5	0
5	0	0
0	7	0
7	0	0
1	3	3
3	1	3
1	5	5
5	1	5
1	4	4
4	1	4
1	7	7
7	1	7

Рисунок 8 – Визуализатор «Таблица»

2) предположим, что нужно построить нейросетевую модель, на вход которой подаются два множителя, а на выходе получается их произведение.

Для этого, находясь на узле импорта, следует вызвать Мастер обработки и в его окне выбрать обработчик Нейросеть, после чего перейти к следующему шагу. Во втором окне нужно установить назначение полей: Аргумент1 и Аргумент2 представить, как входные, а поле Произведение – как выходное.

3) на следующем шаге предлагается настроить разбиение исходного множества данных на обучающее и тестовое. Оставим опции, принятые по умолчанию (рисунок 9).

4) в третьем окне Мастера нужно указать параметры архитектуры многослойного персептрона и активационной функции.

Для данной задачи выбираем один скрытый слой с четырьмя нейронами (рисунок 10).

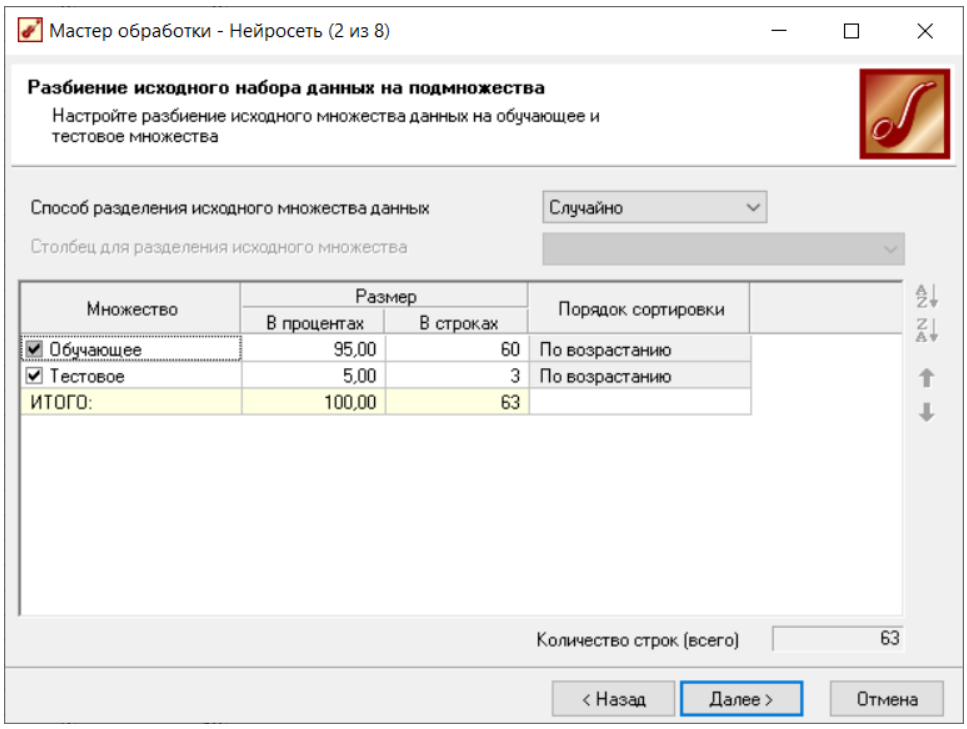


Рисунок 9 – Разбиение исходного множества данных

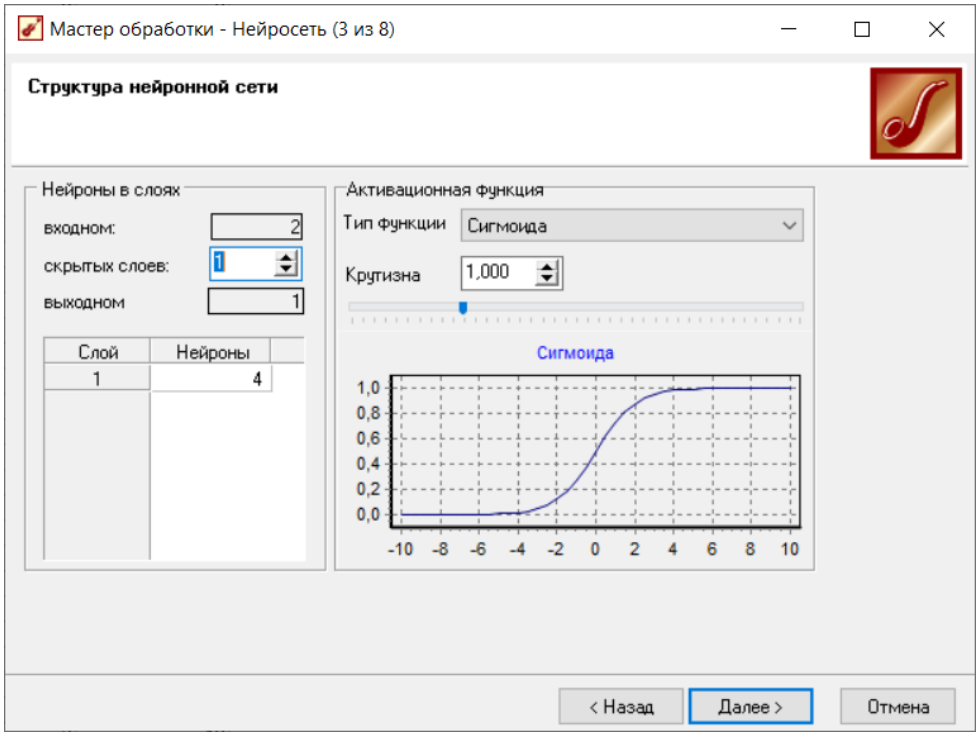


Рисунок 10 – Структура нейронной сети

5) вслед за этим выбирают алгоритм обучения многослойного персептрона и указывают его параметры (рисунок 11).

6) далее нужно настроить условия остановки обучения.

Примем, что пример следует считать распознанным, если ошибка станет менее 0,005, и укажем в поле Эпоха 10 000.

В следующем окне Мастер предложит запустить процесс обучения, в ходе которого можно наблюдать как величину ошибки, так и процент распознанных

примеров. Параметр Темп обновления показывает, через какое количество эпох обучения начинает выводиться данная информация (рисунок 12).

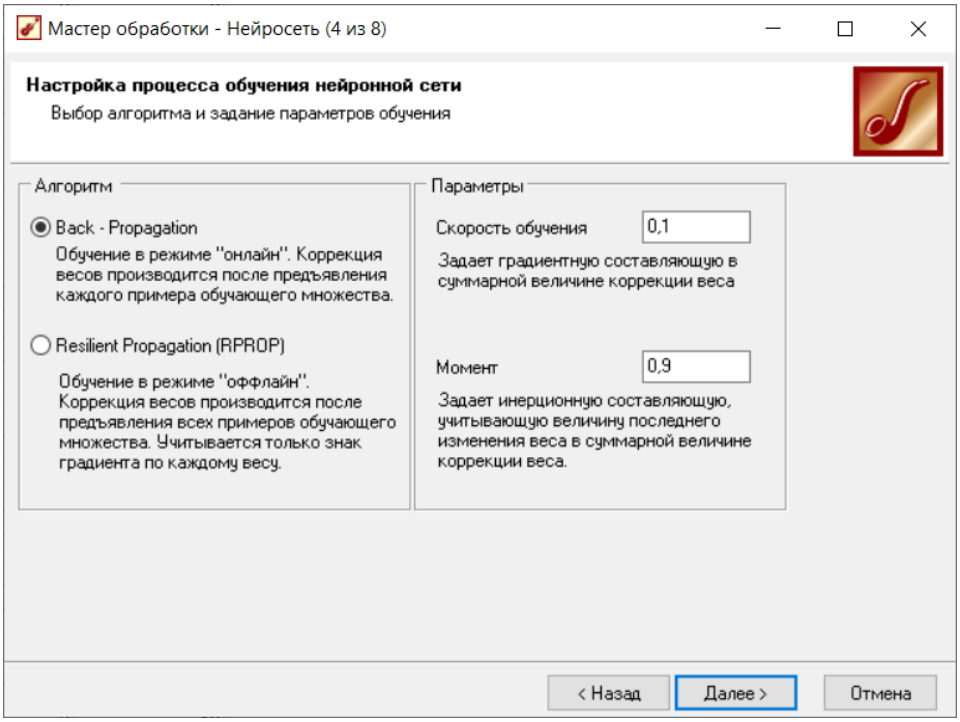


Рисунок 11 – Выбор алгоритма обучения и настройка его параметров

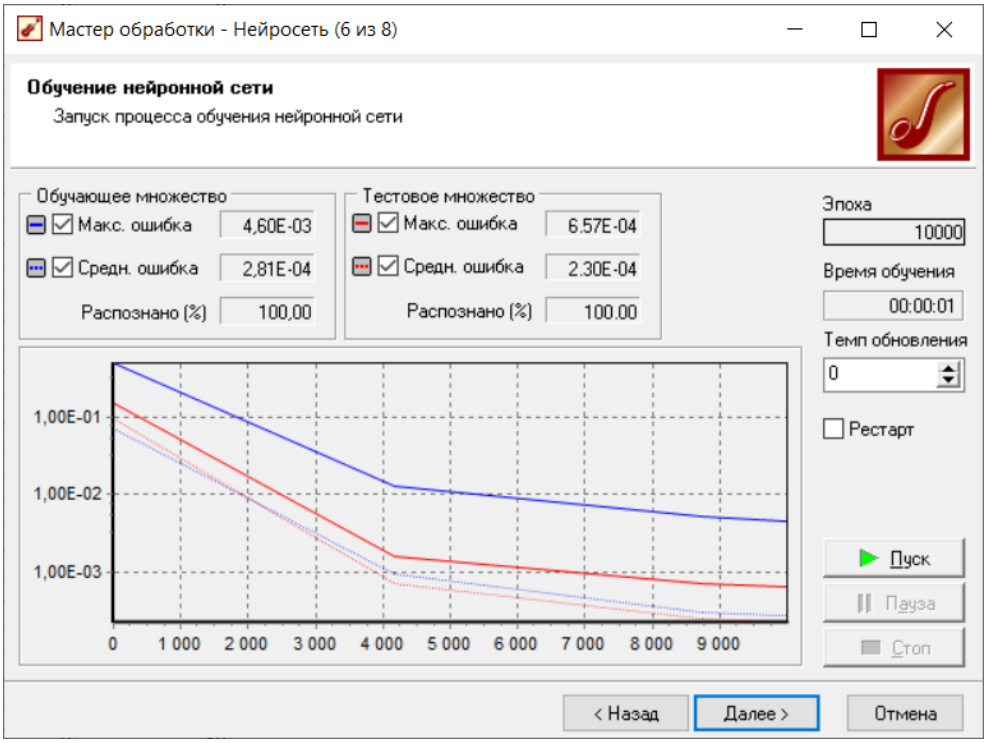


Рисунок 12 – Процесс обучения нейронной сети

7) после того как процесс обучения сети завершится, выберем следующие визуализаторы: Граф нейросети, Диаграмма рассеяния, Что-если.

Граф нейросети позволяет представить нейронную сеть графически, со всеми нейронами и синоптическими связями. При этом можно увидеть не только

структуру НС, но и значения весов для всех связей. В зависимости от веса их цвет меняется, а соответствующее числовое значение можно определить на цветовой шкале, расположенной в нижней части окна (рисунок 13).

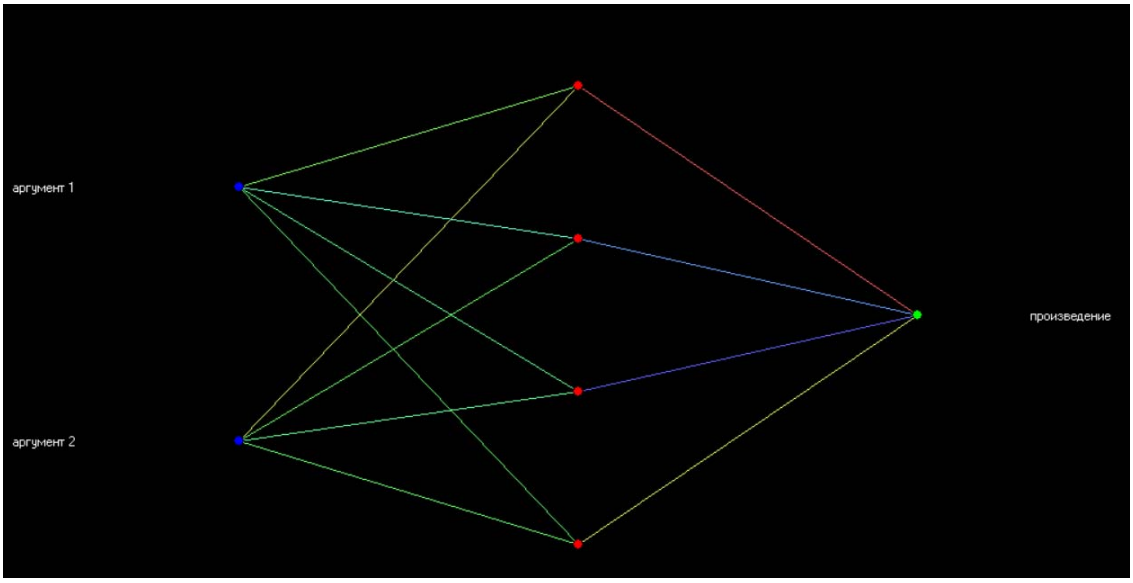


Рисунок 13 – Граф нейронной сети

Диаграмма рассеяния позволяет оценить качество полученной модели. Он показывает отклонение прогнозируемых данных от эталонных. Красные кружки на диаграмме соответствуют примерам из обучающей выборки, причем их абсцисса равна эталонному значению, а их ордината – выходному значению, рассчитанному обученной моделью. Прямая диагональная линия представляет собой линию точных значений; чем ближе к ней кружок, тем меньше ошибка модели (рисунок 14).

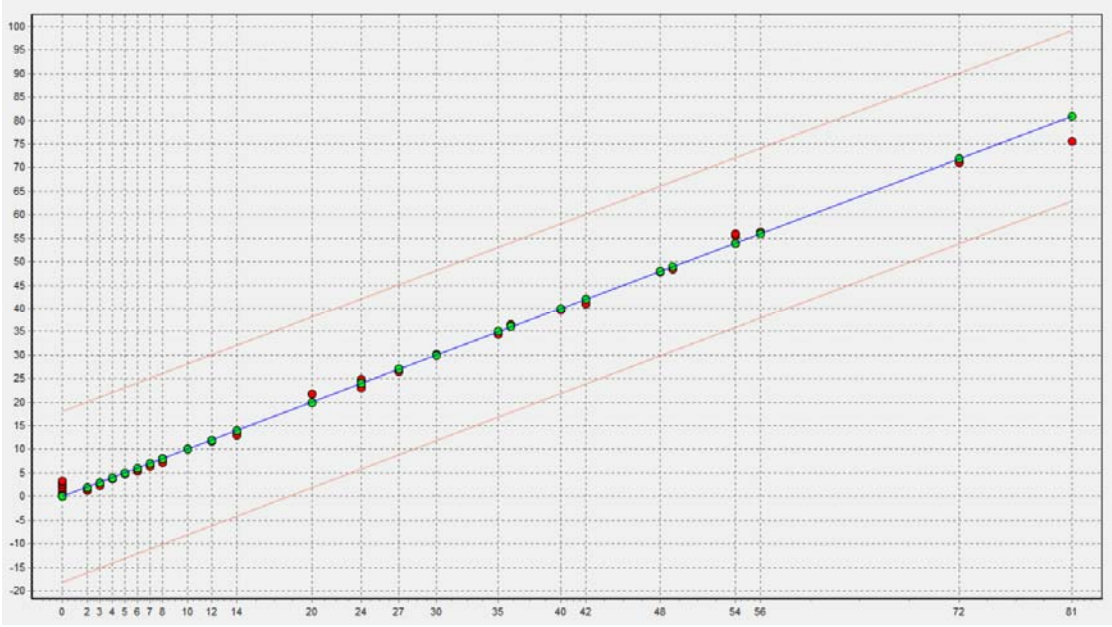


Рисунок 14 – Визуализатор «Диаграмма рассеяния»

Визуализатор «Что-если» дает возможность провести эксперимент, введя любые значения множителей Аргумент1 и Аргумент2 и рассчитав результат по модели (рисунок 15).

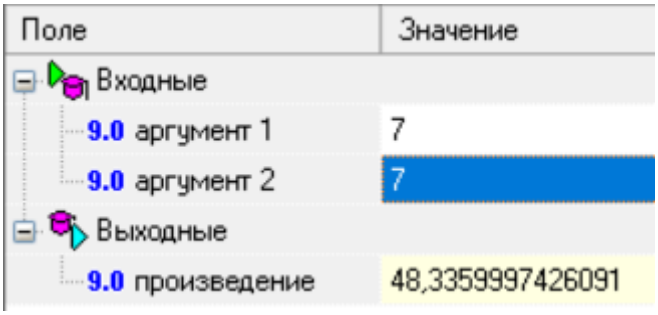


Рисунок 15 – Визуализатор «Что-если»

Так, в обучающей выборке не было примера, в котором первый аргумент равен 7, а второй – 7. Если ввести эти данные в визуализатор, получим 48,33, что весьма близко к истине.

Сохраните проект в файле L1.ded.

3 Аппроксимация многомерных функций

При решении самых разнообразных задач (инженерных, экономических, научных) нередко возникает потребность подобрать непрерывную функцию, наиболее точно выражающую фактически наблюдаемые взаимосвязи между параметрами.

Предположим, что имеется набор пар данных типа вход-выход {(x1,y1), (x2,y2),..., (xn,yn)}, которые генерируются неизвестной функцией, искаженной шумом. Задача аппроксимации состоит в нахождении неизвестной функции

y = F(x), которая в точках x1, x2, xn принимает значения, как можно более близкие к y1, y2,..., yn.

На практике вид искомой функции чаще всего определяют с помощью точечного графика, позволяющего наглядно проследить характер зависимости между входными и выходными параметрами.

Так, на рисунке видно, что на графике слева взаимосвязь переменных близка к линейной; поэтому фактические значения лучше всего аппроксимируются прямой линией. Отклонения от этой линии можно интерпретировать как случайные колебания. Напротив, на графике справа реальная взаимосвязь величин x и y явно имеет нелинейный характер: какую бы прямую линию мы ни провели, отклонения точек от нее будут слишком большими, чтобы считаться случайными. В данном случае необходимо использовать параболу второго или третьего порядка, и тогда можно получить достаточно хорошее приближение (рисунок 16).

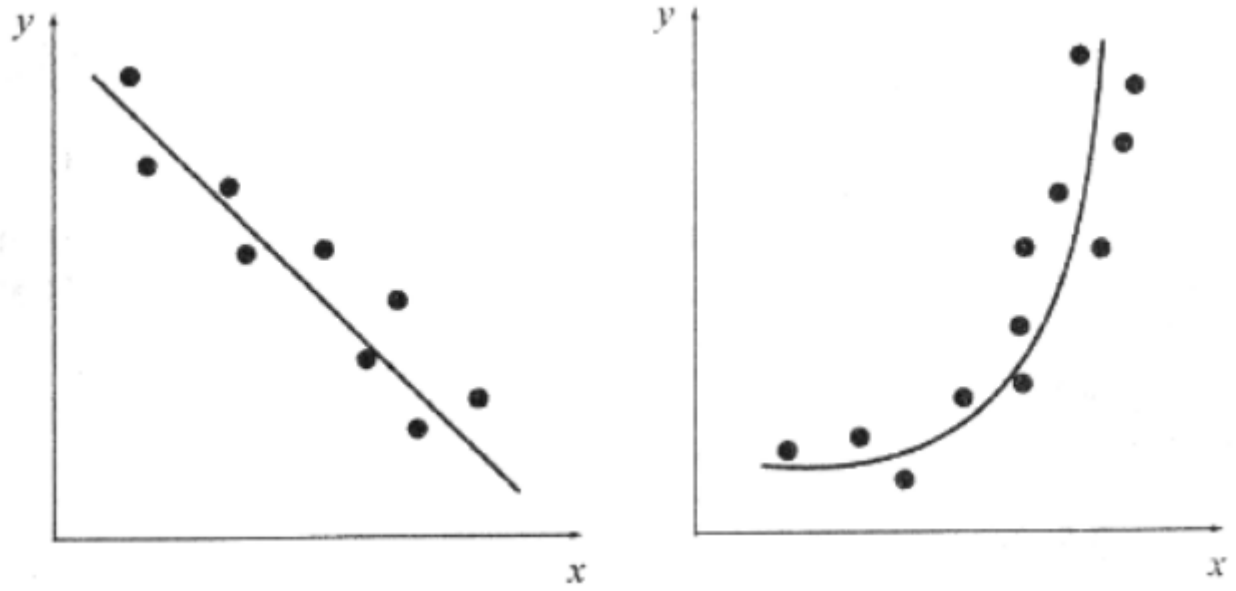


Рисунок 16 – Взаимосвязь переменных

Ситуация заметно осложняется, когда необходимо проанализировать зависимость выходной переменной y не от одной, а от нескольких входных переменных сразу. В этом случае аппроксимация осуществляется с помощью многомерной функции y = F(x), где x = [x1, x2, xn] – вектор с n компонентами. Естественно, графический метод, позволяющий использовать для решения задачи геометрическую интуицию, здесь не подходит. Модели на основе искусственных нейронных сетей снимают эту проблемы, поскольку:

– доказано, что нейронные сети – универсальные аппроксиматоры и позволяют имитировать любую непрерывную функцию с заданной точностью;

– исследователь избавлен от необходимости самостоятельно выдвигать гипотезы о виде приближающей функции;

– существуют быстрые алгоритмы обучения соответствующих нейронных сетей. В силу указанных причин нейронные сети стали широко использоваться при решении сложных задач, требующих построения аппроксимирующих зависимостей.

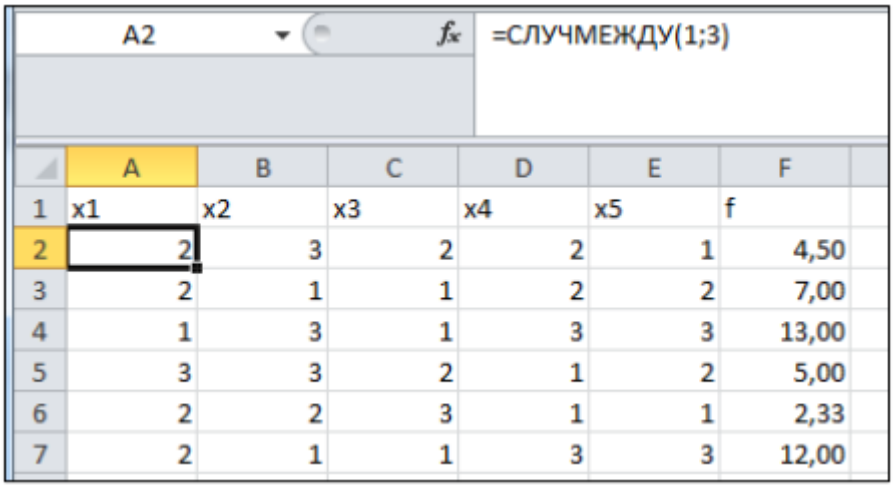
Задание

Для многомерной нелинейной функции $f = \frac{x_1+x_2}{x_3} + x_4 x_5$ постройте нейронную сеть, позволяющую аппроксимировать ее значение (предполагается $1 \leq x_i \leq 3$; i = 1,...,5).

Задание необходимо выполнять в следующем порядке.

1 Подготовить обучающую выборку средствами приложения Microsoft Excel и оформить ее в виде файла *.txt.

Чтобы создать набор случайных чисел, нужно использовать функцию Excel СЛУЧМЕЖДУ() (рисунок 17).



	A	B	C	D	E	F
1	x1	x2	x3	x4	x5	f
2	2	3	2	2	1	4,50
3	2	1	1	2	2	7,00
4	1	3	1	3	3	13,00
5	3	3	2	1	2	5,00
6	2	2	3	1	1	2,33
7	2	1	1	3	3	12,00

Рисунок 17 – Создание обучающей выборки

Рассчитать значение заданной функции в соседнем столбце.

2 Провести обучение нескольких нейронных сетей (как минимум двух).

3 Проверить качество каждой обученной сети с помощью диаграммы рассеяния, отражающей близость обученной модели к исходной. Выбрать наилучшую модель и оценить точность аппроксимации.

4 Провести исследования для функции $f = x_1 - 20\sin(x_2) + 5x_3 + \frac{x_4}{e^{x_5}}$.

Сохраните проект в файле L1.ded.

Аппроксимация многомерных функций

Рассмотрим пример построения системы оценки кредитоспособности физического лица.

Предположим, что эксперты определили основные факторы, определяющие кредитоспособность. Ими оказались возраст, образование, площадь квартиры, наличие автомобиля и т. д. В организации была накоплена статистика возвратов или невозвратов взятых кредитов. Эта статистика представлена таблицей (файл CreditSample.txt).

Необходимо обучить нейросеть для принятия решения о выдаче кредита физическому лицу.

1) импортировать данные в проект L2.ded из файла CreditSample.txt.

При импорте настроить поля следующим образом (рисунок 18).

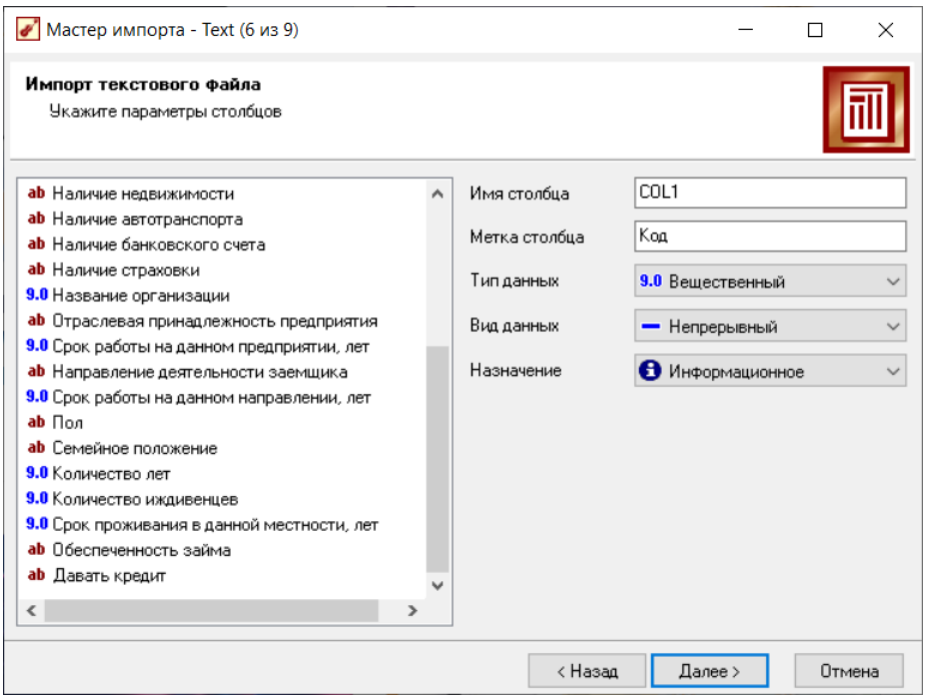


Рисунок 18 – Настройка полей

2) в Мастере обработки выбрать Нейросеть. Задать входные и выходные данные следующим образом (рисунок 19).

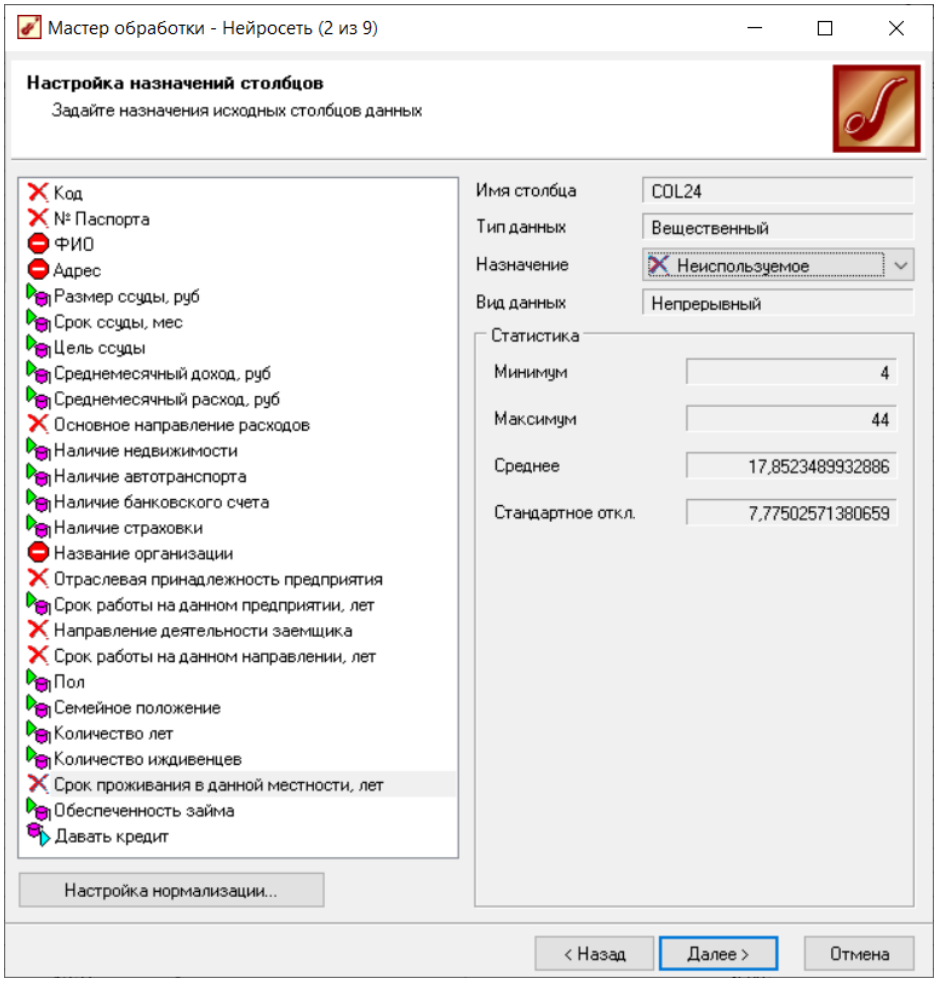


Рисунок 19 – Задание входных и выходных параметров

Провести нормализацию полей:

– непрерывные (числовые) поля привести в диапазоне [–1, 1];

– поля «Наличие недвижимости», «Наличие автотранспорта», «Наличие банковского счета», «Наличие страховки», «Пол», «Семейное положение» – уникальные значения;

– поле Цель ссуды – битовая маска;

– поля «Обеспеченность займа» и «Давать кредит» – уникальные значения (поменять местами Ложь и Истина) (рисунок 20).

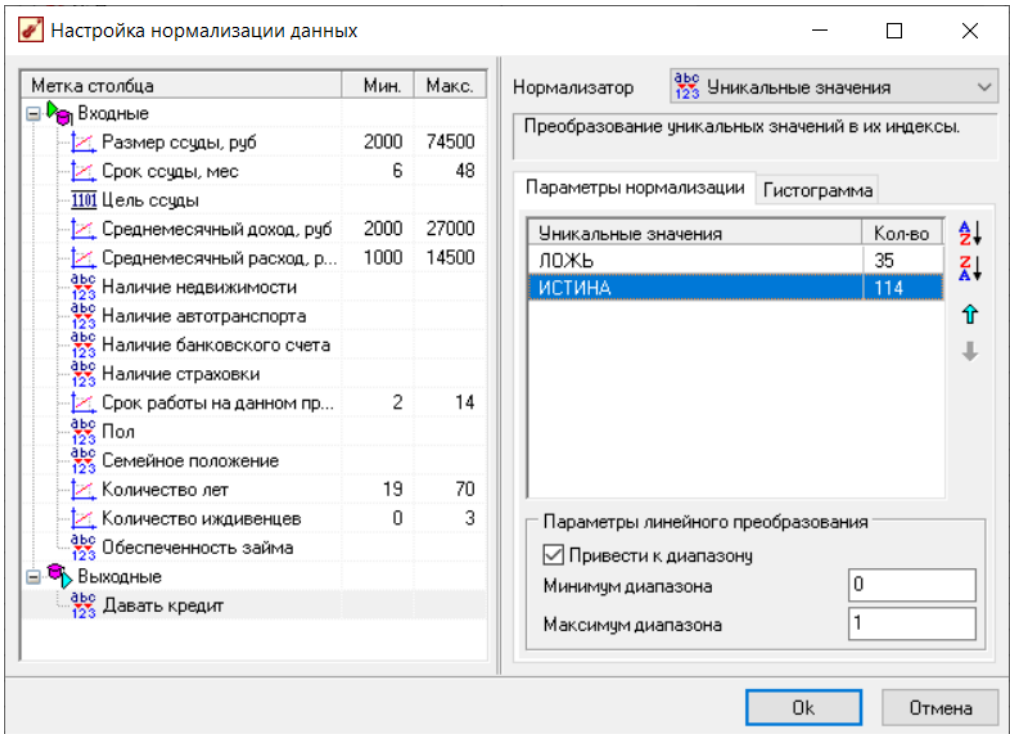


Рисунок 20 – Настройка нормализации данных

На этом нормализация закончена.

3) обучающую выборку разобьем на обучающее и тестовое множества так, как программа предлагает это сделать по умолчанию, т. е. в обучающее множество попадут случайные 95 % записей, а остальные 5 % – в тестовое.

Конфигурация сети будет такой: во входном слое – 17 нейронов, то есть по одному нейрону на один вход.

Сделаем один скрытый слой с двумя нейронами. В выходном слое будет один нейрон, на выходе которого будет решение о выдаче кредита.

Выберем алгоритм обучения сети Resilient Propagation с настройками по умолчанию. Условие окончания обучения оставим без изменения.

4) обученную таким образом нейросеть можно использовать для принятия решения о выдаче кредита физическому лицу. Это можно сделать, применяя анализ «Что-если».

После изменения в этой таблице входных полей система сама принимает решение о выдаче кредита и в поле «Давать кредит» проставляет либо «Истина», либо «Ложь».

Кроме такой таблицы анализ «Что-если» содержит диаграмму, на которой

отображается зависимость выходного поля от одного из входных полей при фиксированных значениях остальных полей.

Например, требуется узнать, на какой срок ссуды кредита может рассчитывать человек, обладающий определенными характеристиками. Это можно определить по диаграмме (рисунок 21).

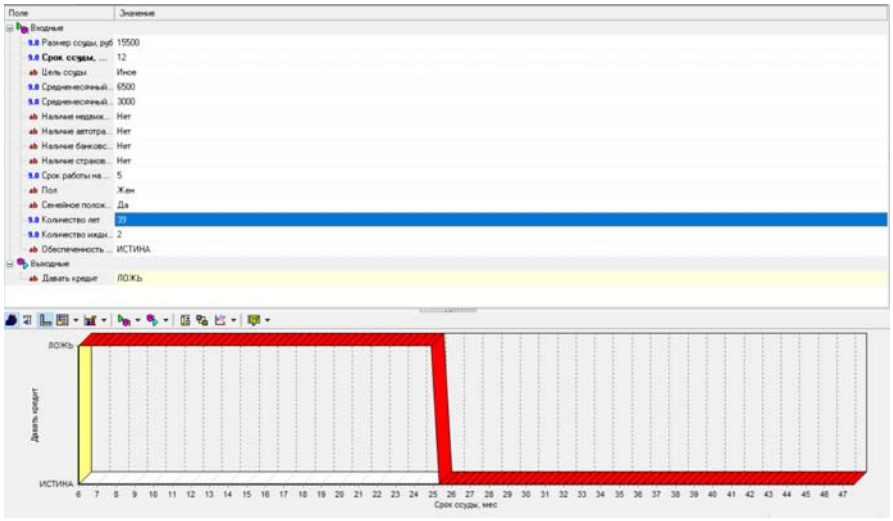


Рисунок 21 – Диаграмма анализа «Что-если»

Видно, что если срок ссуды увеличится до 26 месяцев, то параметр «Давать ссуду» изменится на «Истина» (рисунок 22).

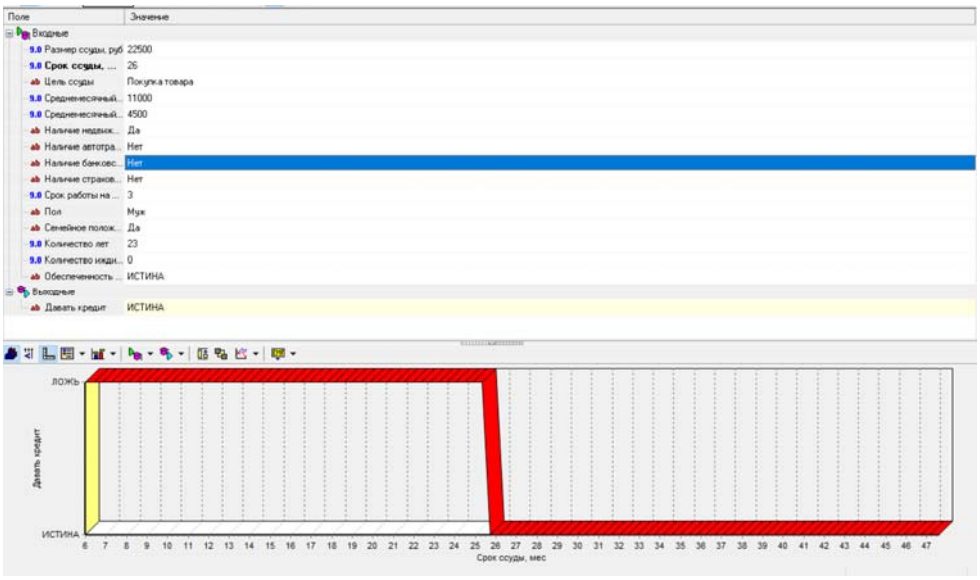


Рисунок 22 – Диаграмма анализа «Что-если»

5) в Таблице сопряженности можно увидеть точность классификации на обучающей и тестовой выборке (рисунок 23).

Давать кредит

Фактически	Классифицировано		
	ИСТИНА	ЛОЖЬ	Итого
ИСТИНА	110	4	114
ЛОЖЬ	2	33	35
Итого	112	37	149

Рисунок 23 – Таблица сопряженности

6) на Графе нейросети можно представить нейронную сеть графически, со всеми нейронами и синоптическими связями (рисунок 24).

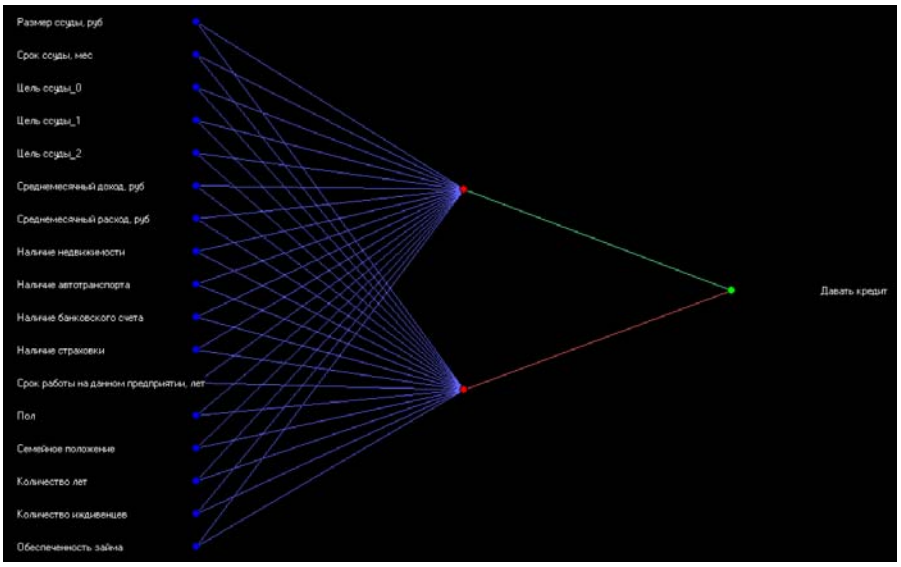


Рисунок 24 – Граф нейронной сети

Сохраните проект в файле L2.ded.

3 Ассоциативные правила

1 Алгоритм поиска ассоциативных правил

Ассоциативные правила позволяют находить закономерности между связанными событиями. Примером такого правила служит утверждение, что покупатель, приобретающий «Хлеб», приобретет и «Молоко». Впервые эта задача была предложена для поиска ассоциативных правил для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины (market basket analysis).

Пусть имеется база данных, состоящая из покупательских транзакций. Каждая транзакция – это набор товаров, купленных покупателем за один визит. Такую транзакцию еще называют рыночной корзиной. Целью анализа является установление следующих зависимостей: если в транзакции встретился некоторый набор элементов X, то на основании этого можно сделать вывод о том, что другой набор элементов Y также должен появиться в этой транзакции. Установление таких зависимостей дает нам возможность находить очень простые и интуитивно понятные правила.

Ассоциативное правило состоит из двух наборов предметов, называемых

условие (англ: antecedent) и следствие (англ: consequent), записываемых в виде X → Y , что читается «из X следует Y». Таким образом, ассоциативное правило формулируется в виде «Если условие, то следствие».

При поиске ассоциативных правил используют показатели, позволяющие оценить значимость правила. В этой связи можно выделить объективные и субъективные меры значимости правил. Объективными являются такие меры, как поддержка и достоверность, которые могут применяться независимо от конкретного приложения. Субъективные меры связаны со специальной информацией, определяемой пользователем в контексте решаемой задачи. Такими субъективными мерами являются лифт и левередж.

Ассоциативные правила описывают связь между наборами предметов, соответствующим условию и следствию. Эта связь характеризуется двумя показателями – поддержкой (s) и достоверностью(c).

Правило «Из X следует Y» имеет поддержку s, если s % транзакций из всего набора содержат наборы элементов X и Y.

Достоверность правила показывает, какова вероятность того, что из X следует Y. Правило «Из X следует Y» справедливо с достоверностью c, если c % транзакций из всего множества, содержащих набор элементов X, также содержат набор элементов Y.

Покажем на конкретном примере: пусть 75 % транзакций, содержащих хлеб, также содержат молоко, а 3 % от общего числа всех транзакций содержат оба товара. 75 % – это достоверность правила, а 3 % – это поддержка.

Лифт (L) – это отношение частоты появления условия в транзакциях, которые также содержат и следствие, к частоте появления следствия в целом. Значения лифта большие, чем единица, показывают, что условие более часто появляется в транзакциях, содержащих и следствие, чем в остальных. Можно сказать, что лифт является обобщенной мерой связи двух предметных наборов: при значениях лифта >1 связь положительная, при 1 она отсутствует, а при значениях <1 – отрицательная.

Другой мерой значимости правила является левередж. Это разность между наблюдаемой частотой, с которой условие и следствие появляются совместно (т. е. поддержкой ассоциации), и произведением частот появления (поддержек) условия и следствия по отдельности.

Такие меры, как лифт и левередж, могут использоваться для последующего ограничения набора рассматриваемых ассоциаций путем установки порога значимости, ниже которого ассоциации отбрасываются.

Алгоритмы поиска ассоциативных правил предназначены для нахождения всех правил вида «Из X следует Y», причем поддержка и достоверность этих правил должны находиться в рамках некоторых, наперед заданных, границ, называемых соответственно минимальной и максимальной поддержкой и минимальной и максимальной достоверностью.

Границы значений параметров поддержки и достоверности выбираются таким образом, чтобы ограничить количество найденных правил. Если поддержка имеет большое значение, то алгоритмы будут находить правила, хорошо известные аналитикам или настолько очевидные, что нет никакого

смысла проводить такой анализ. С другой стороны, низкое значение поддержки ведет к генерации огромного количества правил, что, конечно, требует существенных вычислительных ресурсов. Тем не менее большинство интересных правил находится именно при низком значении порога поддержки, хотя слишком низкое значение поддержки ведет к генерации статистически необоснованных правил. Таким образом, необходимо найти компромисс, обеспечивающий, во-первых, интересность правил и, во-вторых, их статистическую обоснованность. Поэтому значения этих границ напрямую зависят от характера анализируемых данных и подбираются индивидуально.

Простейший алгоритм поиска состоит в том, что для всех ассоциаций, которые могут быть построены на основе базы данных, определяются поддержка и достоверность, а затем отбираются те из них, для которых эти показатели превышают заданное пороговое значение. Однако в большинстве случаев такое элементарное решение неприемлемо, поскольку число ассоциаций, которое при этом придется анализировать, слишком велико. Так, если выборка содержит всего 100 предметов, количество образуемых ими ассоциаций будет порядка 10^{31}, а в реальных ситуациях (например, при анализе покупок в супермаркете) номенклатура учитываемых продуктов может достигать нескольких тысяч и более. Очевидно, что никаких вычислительных мощностей на такой расчет не хватит.

Поэтому на практике при поиске ассоциативных правил используют различные приемы, которые позволяют уменьшить пространство поиска до размеров, обеспечивающих приемлемые затраты машинного времени. Сейчас одним из наиболее распространенных является алгоритм a priori, основанный на понятии популярного набора (часто встречающийся предметный набор). Этот термин обозначает предметный набор, частота появления которого в общей совокупности транзакций превышает некоторый заранее заданный уровень.

Таким образом, алгоритм a priori включает два этапа:

– поиск популярных наборов;

– формулировка ассоциативных правил, удовлетворяющих заданным ограничениям по уровням поддержки и достоверности.

В Deductor Studio для решения задач ассоциации используется обработчик Ассоциативные правила. В нем реализован алгоритм a priori. Обработчик требует на входе два поля: идентификатор транзакции и элемент транзакции.

2 Создание ассоциативных правил для анализа покупательских корзин для стимулирования спроса

Розничная сеть по продаже бытовой химии поставила задачу анализа покупательских корзин для оптимизации их размещения на витринах и проведения кросс-продаж. Отдел маркетинга предоставил 5 000 чеков, в которых отражены покупки, сделанные предыдущими клиентами магазинов. Стоят следующие задачи:

– предсказать, какие товары покупатели могут выбрать в зависимости от того, что уже есть в их корзинах;

– выявить наиболее популярные товарные наборы, состоящие из более чем 1 предмета;

– предложить рекламные акции типа «Каждому купившему A и B товар C в подарок».

1) импортируйте в новый проект файл Чеки.txt.

Поле ID (идентификатор транзакции) – это номер чека или код клиента. А поле ITEM (элемент транзакции) – это наименование товара в чеке.

Оба поля (идентификатор и элемент транзакции) должны быть дискретного вида.

2) к узлу импорта добавим обработчик Ассоциативные правила. Столбец ID сделаем идентификатором транзакции, а ITEM – ее элементом (рисунок 25).

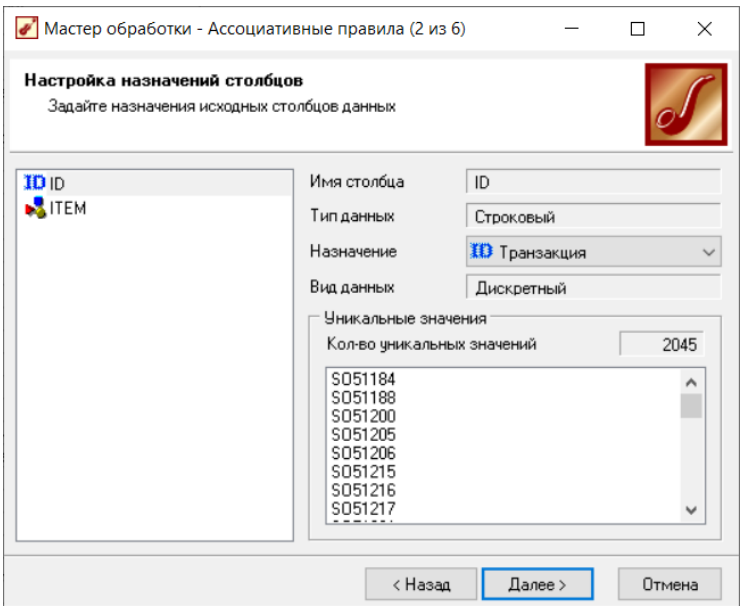


Рисунок 25 – Назначение исходных столбцов данных

3) настроить параметры построения ассоциативных правил (параметры алгоритма a priori) (рисунок 26).

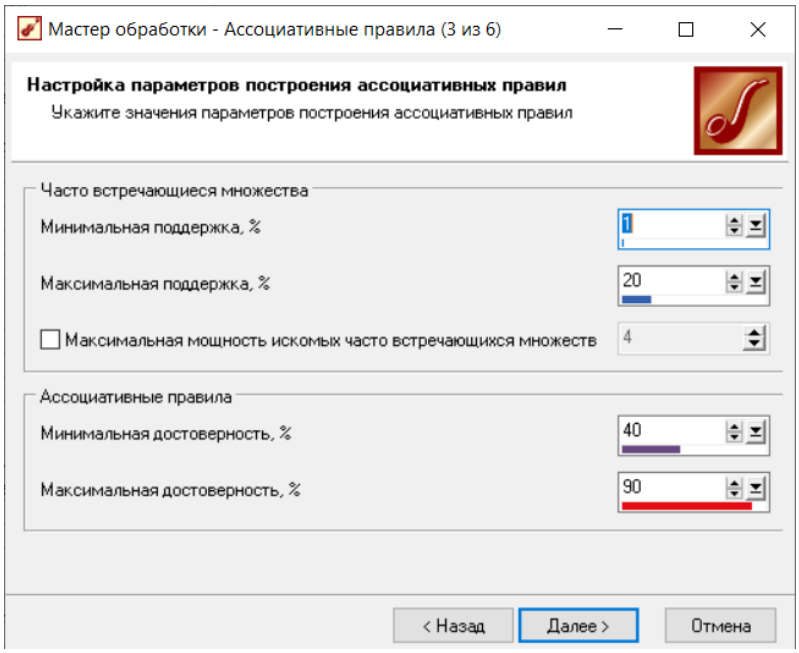


Рисунок 26 – Настройка параметров построения ассоциативных правил

Здесь для изменения доступны следующие параметры:

– минимальная и максимальная поддержка в % – ограничивают пространство поиска часто встречающихся предметных наборов. Эти границы определяют множество популярных наборов, из которых и будут создаваться ассоциативные правила;

– минимальная и максимальная достоверность в % – в результирующий набор попадут только те ассоциативные правила, которые удовлетворяют условиям минимальной и максимальной достоверности;

– максимальная мощность искомых часто встречающихся множеств – параметр ограничивает длину k-предметного набора. Например, при установке значения 4-й шаг генерации популярных наборов будет остановлен после получения множества 4-предметных наборов. В конечном итоге это позволяет избежать появления длинных ассоциативных правил, которые трудно интерпретируются.

Пока оставим все настройки на данной вкладке по умолчанию.

Нажатие на кнопку «Пуск» приведет к работе алгоритма поиска ассоциативных правил. По окончании его работы справа в полях появится следующая информация (рисунок 27):

– количество множеств – число популярных наборов, удовлетворяющих заданным условиям минимальной поддержки и достоверности;

– количество правил – число сгенерированных ассоциативных правил.

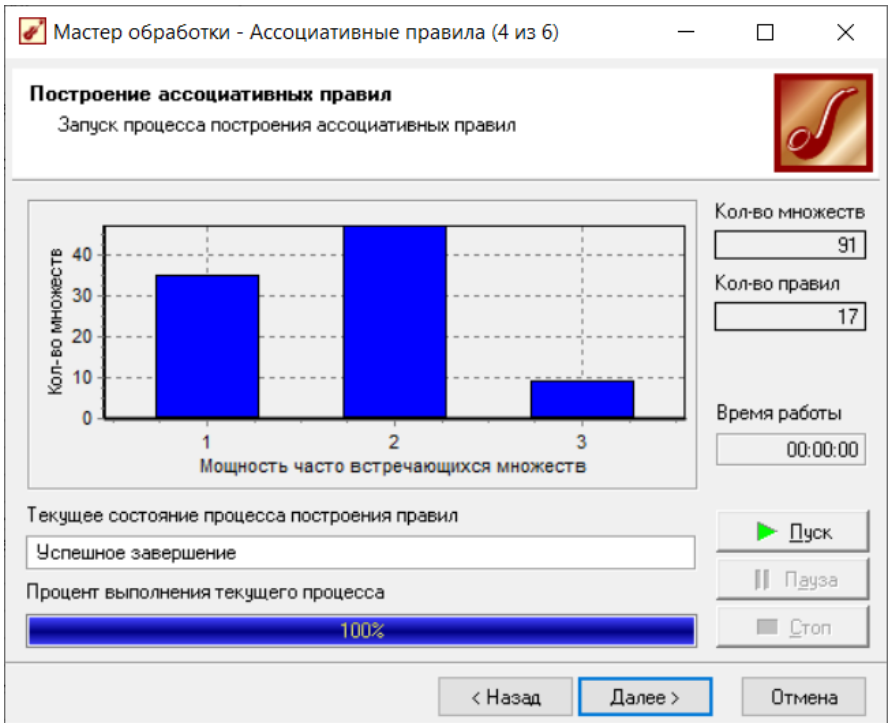


Рисунок 27 – Результат алгоритма a priori

4) далее выбираем все доступные специализированные визуализаторы («Правила», «Популярные наборы», «Дерево правил», «Что-если») и визуализатор «Таблица».

Все эти визуализаторы, кроме «Что-если», отображают результаты работы

алгоритма в различных формах.

5) на вкладке «Популярные наборы», как следует из названия, отображается множество найденных популярных предметных наборов в виде списка.

Кнопка  предлагает на выбор несколько вариантов сортировки списка, а кнопка  вызывает окно настройки фильтра множеств.

Например, задав в фильтре минимальное значение поддержки 6 % и отсортировав их по убыванию поддержки, получим следующие 16 популярных наборов. Параметр «Мощность» указывает на количество товаров в одном наборе (рисунок 28).

№	Номер множества	ab. Элементы	Поддержка		|s| Мощность
			Кол-во	%	
1	10	Мыло кусковое	392	19,17	1
2	35	Чистящий порошок универсальный	292	14,28	1
3	6	Зубная паста	288	14,08	1
4	3	Гель для туалетов	221	10,81	1
5	11	Освежитель воздуха	208	10,17	1
6	9	Мыло жидкое	200	9,78	1
7	12	Отбеливатель	199	9,73	1
8	32	Стиральный порошок ручной	195	9,54	1
9	59	Мыло жидкое Мыло кусковое	167	8,17	2
10	26	Средство для ухода за мебелью	154	7,53	1
11	24	Средство для мытья посуды	152	7,43	1
12	19	Пятновыводитель	146	7,14	1
13	34	Стиральный порошок-автомат	144	7,04	1
14	21	Сода кальцинированная	133	6,50	1
15	67	Мыло кусковое Средство для мытья посуды	131	6,41	2
16	29	Средство для чистки плит	129	6,31	1

Рисунок 28 – Использование фильтра минимальной поддержки

6) на вкладке «Дерево правил» предлагается еще один удобный способ отображения множества ассоциативных правил, которое строится либо по условию, либо по следствию. При построении дерева правил по условию, на первом (верхнем) уровне находятся узлы с условиями, а на втором уровне – узлы со следствием. В дереве, построенном по следствию, наоборот, на первом уровне располагаются узлы со следствием.

Справа от дерева расположен список правил, построенный по выбранному узлу дерева (рисунок 29).

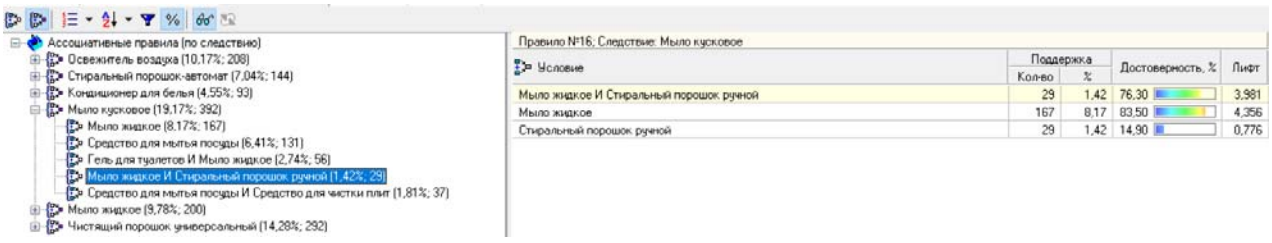


Рисунок 29 – Список правил выбранного узла

Для каждого правила отображаются поддержка, достоверность и лифт. Если дерево построено по условию, то вверху списка отображается условие правила, а список состоит из его следствий. Тогда правила отвечают на вопрос, что будет при таком условии. Если же дерево построено по следствию, то вверху списка отображается следствие правила, а список состоит из его условий.

Эти правила отвечают на вопросы, что нужно, чтобы было заданное следствие, или какие товары нужно продать для того, чтобы продать товар из следствия.

7) на вкладке «Правила», помимо самих ассоциативных правил, приводятся их основные расчетные характеристики: поддержка, достоверность и лифт. На рисунке показаны ассоциативные правила, отсортированные по убыванию лифта (рисунок 30).

№	Номер правила	Условие	Следствие	Поддержка		Достоверность	Лифт
				Кол-во	%		
1	4	Стиральный порошок-авто	Кондиционер для белья	79	3,86	54,86	12,064
2	3	Кондиционер для белья	Стиральный порошок-авто	79	3,86	84,95	12,064
3	1	Бумажное полотенце	Освежитель воздуха	52	2,54	59,77	5,876
4	2	Запасной баллон для осве	Освежитель воздуха	53	2,59	59,55	5,855
5	9	Сода кальцинированная	Чистящий порошок универ	96	4,69	72,18	5,055
6	14	Зубная паста Сода кальцинированная	Чистящий порошок универ	28	1,37	71,79	5,028
7	13	Гель для туалетов Сода кальцинированная	Чистящий порошок универ	34	1,66	70,83	4,961
8	11	Средство от накипи	Чистящий порошок универ	76	3,72	69,72	4,883
9	15	Мыло кусковое Отбеливатель	Мыло жидкое	25	1,22	46,30	4,734
10	8	Салфетки бумажные	Освежитель воздуха	50	2,44	47,17	4,638
11	7	Средство для мытья посу	Мыло кусковое	131	6,41	86,18	4,496
12	12	Гель для туалетов Мыло жидкое	Мыло кусковое	56	2,74	84,85	4,426
13	17	Средство для мытья посу Средство для чистки плит	Мыло кусковое	37	1,81	84,09	4,387
14	6	Мыло кусковое	Мыло жидкое	167	8,17	42,60	4,356
15	5	Мыло жидкое	Мыло кусковое	167	8,17	83,50	4,356
16	16	Мыло жидкое Стиральный порошок руч	Мыло кусковое	29	1,42	76,32	3,981
17	10	Средство для чистки каф	Чистящий порошок универ	42	2,05	48,28	3,381

Рисунок 30 – Сортировка ассоциативных правил по убыванию лифта

Поэкспериментируйте с настройкой фильтра в визуализаторах «Правила», «Популярные наборы» и «Дерево правил».

3 Интерпретация ассоциативных правил

Теперь остановимся на наиболее важном этапе – интерпретации ассоциативных правил. Дело в том, что ассоциативные правила сами по себе, как результат работы некоторого алгоритма, еще не готовы к использованию. Их нужно проинтерпретировать, т. е. понять, какие из ассоциативных правил представляют интерес, действительно ли правила отражают закономерности или, наоборот, являются артефактом. Это требует тщательной работы аналитика и понимания предметной области, в которой решается задача ассоциации.

Все множество ассоциативных правил можно разделить на три вида:

– полезные правила – содержат действительную информацию, которая ранее была неизвестна, но имеет логичное объяснение. Такие правила могут быть использованы для принятия решений, приносящих выгоду.

– тривиальные правила – содержат действительную и легко объяснимую информацию, которая уже известна. Такие правила, хотя и объяснимы, но не могут принести какой-либо пользы, т. к. отражают или известные законы в исследуемой области, или результаты прошлой деятельности. При анализе рыночных корзин в правилах с самой высокой поддержкой и достоверностью окажутся товары-лидеры продаж. Практическая ценность таких правил крайне низка.

– непонятные правила – содержат информацию, которая не может быть объяснена. Такие правила могут быть получены или на основе аномальных значений, или глубоко скрытых знаний. Напрямую такие правила нельзя использовать для принятия решений, т. к. их необъяснимость может привести к непредсказуемым результатам. Для лучшего понимания требуется дополнительный анализ.

Варьируя верхним и нижним пределами поддержки и достоверности, можно избавиться от очевидных и неинтересных закономерностей. Как следствие, правила, генерируемые алгоритмом, принимают приближенный к реальности вид. Понятия «верхний предел» и «нижний предел» очень сильно зависят от предметной области, поэтому не существует четкого алгоритма их выбора. Но есть ряд общих рекомендаций.

Полезные советы:

– большая величина параметра Максимальная поддержка означает, что алгоритм будет находить хорошо известные правила или они будут настолько очевидными, что в них нет никакого смысла. Поэтому ставить порог Максимальная поддержка очень высоким (более 20 %) не рекомендуется;

– большинство интересных правил находится именно при низком значении порога поддержки, хотя слишком низкое значение поддержки ведет к генерации статистически необоснованных правил. Поэтому правила, которые кажутся интересными, но имеют низкую поддержку, дополнительно анализируйте по лифту, а при необходимости рассчитывайте для них левередж;

– ограничивайте мощность часто встречающихся множеств – правила с большим числом предметов в условии трудно интерпретируются и воспринимаются;

– уменьшение порога достоверности приводит к увеличению количества правил. Значение минимальной достоверности не должно быть слишком маленьким, т. к. ценность правила с достоверностью 5 % чаще всего настолько мала, что это и правилом считать нельзя;

– правило с очень большой достоверностью (>8590 %) практической ценности в контексте решаемой задачи не имеет, т. к. товары, входящие в следствие, покупатель, скорее всего, уже купил.

1) при настройках алгоритма a priori по умолчанию получили 17 правил.

Например, первое правило Бумажное полотенце → Освежитель воздуха имеет S = 2,54 %; C = 59,77 % и L = 5,88 (рисунок 31).

3	1	Бумажное полотенце	Освежитель воздуха	52	2,54	59,77	5,876
4	2	Запасной баллон для осве	Освежитель воздуха	53	2,59	59,55	5,855

Рисунок 31 – Фрагмент сортировки по правилам

Это означает следующее:

– ожидаемая вероятность покупки набора Бумажное полотенце + Освежитель воздуха равна 2,54 %;

– если клиент положил в корзину товар Бумажное полотенце, то с вероятностью 59,77 % он купит и Освежитель воздуха;

– клиент, купивший Бумажное полотенце, в 5,8 раз чаще выберет Освежитель воздуха, нежели любой другой товар.

2) анализ полученных правил позволяет прийти к выводу, что многие из них тривиальны:

– Мыло кусковое, Чистящий порошок, Зубная паста, Гель для туалета часто встречаются в условиях и следствиях правил, это лидеры продаж магазина (см. популярные наборы), поэтому и правила с ними имеют высокую достоверность (рисунок 32);

1	6	Мыло кусковое	Мыло жидкое	167	8,17	42,60	4,356
2	15	Мыло кусковое Отбеливатель	Мыло жидкое	25	1,22	46,30	4,734
3	8	Салфетки бумажные	Освежитель воздуха	50	2,44	47,17	4,638
4	10	Средство для чистки каф	Чистящий порошок универ	42	2,05	48,28	3,381
5	4	Стиральный порошок-авт	Кондиционер для белья	79	3,86	54,86	12,064
6	2	Запасной баллон для осве	Освежитель воздуха	53	2,59	59,55	5,855
7	1	Бумажное полотенце	Освежитель воздуха	52	2,54	59,77	5,876
8	11	Средство от накипи	Чистящий порошок универ	76	3,72	69,72	4,883
9	13	Гель для туалетов Сода кальцинированная	Чистящий порошок универ	34	1,66	70,83	4,961
10	14	Зубная паста Сода кальцинированная	Чистящий порошок универ	28	1,37	71,79	5,028
11	9	Сода кальцинированная	Чистящий порошок универ	96	4,69	72,18	5,055
12	16	Мыло жидкое Стиральный порошок руч	Мыло кусковое	29	1,42	76,32	3,981
13	5	Мыло жидкое	Мыло кусковое	167	8,17	83,50	4,356
14	17	Средство для мытья посу Средство для чистки плит	Мыло кусковое	37	1,81	84,09	4,387
15	12	Гель для туалетов Мыло жидкое	Мыло кусковое	56	2,74	84,85	4,426
16	3	Кондиционер для белья	Стиральный порошок-авт	79	3,86	84,95	12,064
17	7	Средство для мытья посу	Мыло кусковое	131	6,41	86,18	4,496

Рисунок 32 – Следствие правил с высокой достоверностью

– группа правил Стиральный порошок-автомат → Кондиционер для белья и наоборот тривиальны сами по себе: люди часто покупаю эти товары вместе.

– правила типа Запасной баллон для освежителя → Освежитель воздуха (и наоборот) тоже тривиальны, так как никому не нужен запасной баллон без освежителя.

А вот правило Салфетки бумажные → Освежитель воздуха не понятное: почему салфетки бумажные покупаются именно с освежителем воздуха.

Однако обратимся к рисунку, в котором сделана попытка классифицировать все правила на тривиальные и непонятные.

Тот факт, что при достоверности 42–43 % встречаются тривиальные ассоциативные правила, говорит о том, что интересные правила содержатся при

меньших значениях достоверности.

Попробуем сделать следующее:

– запустим алгоритм a priori с интервалом допустимой достоверности от 25 до 40 % (рисунок 33);

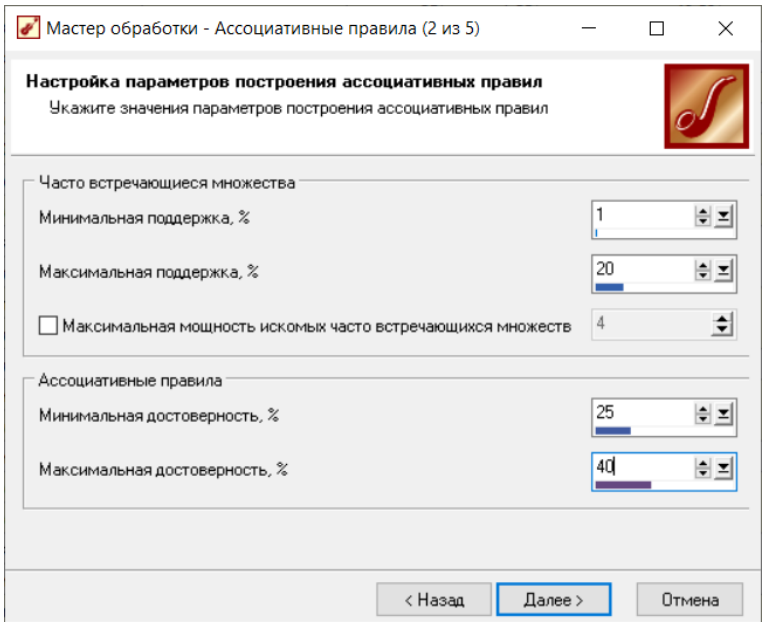


Рисунок 33 – Настройка интервала допустимой достоверности

– не будем рассматривать правила, в следствиях и условиях которых содержатся Гель для туалета, Зубная паста, Мыло жидкое, Мыло кусковое, Освежитель воздуха, Чистящий порошок универсальный – это снова будут тривиальные правила (рисунок 34).

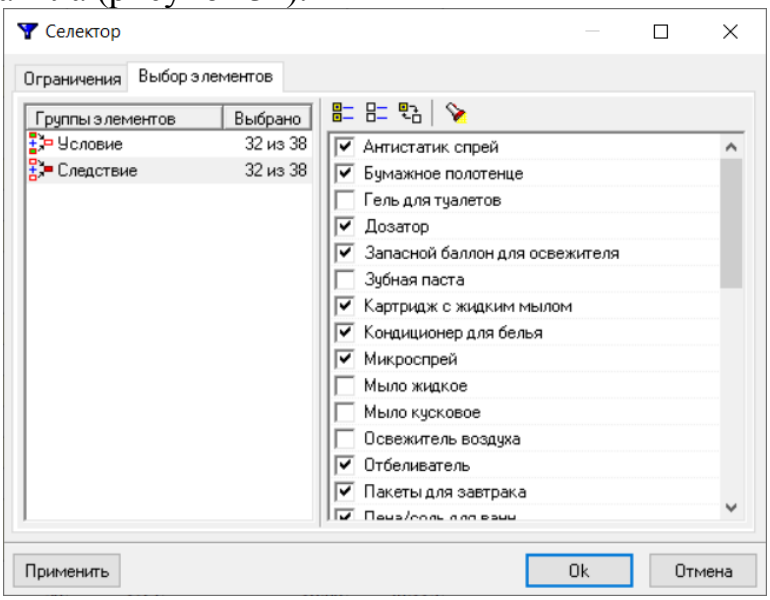


Рисунок 34 – Выбор селектора

В итоге получим дополнительные правила, которые имеют достоверность меньше 40 % (рисунок 35). Как видно, все полученные правила можно назвать полезными: они не очевидны, но понятны.

№	Номер правила	Условие	Следствие	Поддержка		Достоверность	Лифт
				Кол-во	%		
1	44	Пена/соль для ванн	Мыло кусковое	23	1,12	26,14	4,080
			Средство для мытья посуд				
2	35	Антистатик спрей	Мыло кусковое	28	1,37	27,72	4,328
			Средство для мытья посуд				
3	27	Пятновыводитель	Отбеливатель	41	2,00	28,08	2,886
4	45	Мыло кусковое	Средство для чистки плит	37	1,81	28,24	4,477
		Средство для мытья посуд					
5	28	Пена/соль для ванн	Средство для мытья посуд	25	1,22	28,41	3,822
6	46	Средство для чистки плит	Мыло кусковое	37	1,81	28,68	4,477
			Средство для мытья посуд				
7	32	Средство для мытья посуд	Средство для чистки плит	44	2,15	28,95	4,589
8	30	Средство для чистки плит	Салфетки бумажные	38	1,86	29,46	5,683
9	2	Антистатик спрей	Средство для мытья посуд	30	1,47	29,70	3,996
10	13	Запасной баллон для осве	Пена/соль для ванн	28	1,37	31,46	7,311
11	14	Пена/соль для ванн	Запасной баллон для осве	28	1,37	31,82	7,311
12	33	Средство для чистки плит	Средство для мытья посуд	44	2,15	34,11	4,589
13	29	Салфетки бумажные	Средство для чистки плит	38	1,86	35,85	5,683
14	43	Мыло кусковое	Средство для мытья посуд	21	1,03	38,89	5,232
		Отбеливатель					

Рисунок 35 – Результат работы алгоритма

Например, возьмем правило номер 27 Пятновыводитель → Отбеливатель. Проанализируем это правило с помощью лифта. Его величина равна 2,886, что больше чем 1, значит, с помощью правила предсказать покупку отбеливателя вероятнее, чем случайным угадыванием. Как можно применить на практике это правило? Это зависит от конкретных целей. Приведем всевозможные варианты.

Пятновыводитель → Отбеливатель

1 Разместите их рядом на витрине.

2 Разместите их на большом расстоянии друг от друга.

3 Сформируйте подарочные наборы «Пятновыводитель + Отбеливатель»

4 Сформируйте подарочные наборы «Пятновыводитель + Отбеливатель + плохо продаваемый товар».

5 Поднимите цену на одно, снизьте на другое.

6 Закажите комплекты пятновыводителей и отбеливателей одного бренда и серии.

Какова вероятность того, что клиент, купивший Антистатик-спрей, купит и Средство для мытья посуды?

4 Визуализатор «Что-если» в ассоциативных правилах

1) кроме уже изученных визуализаторов «Правила», «Популярные наборы» и «Дерево правил», в Deductor Studio к узлу «Ассоциативные правила» доступен визуализатор «Что-если». Он позволяет ответить на вопрос, что мы получим в качестве следствия, если выберем данные условия, например, какие товары, приобретаются совместно с выбранными товарами (рисунок 36).

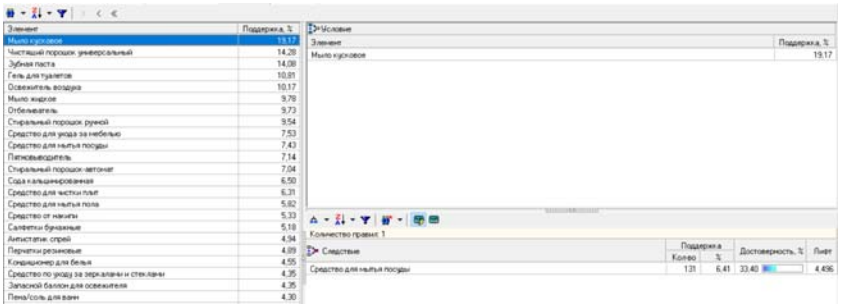
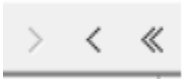


Рисунок 36 – Визуализатор «Что-если»

В окне слева расположен список всех элементов транзакций. Справа от каждого элемента указана поддержка: сколько раз данный элемент встречается в транзакциях. В правом верхнем углу расположен список элементов, входящих в условие, выбирается он с помощью двойного щелчка левой кнопки мыши или вспомогательных кнопок  . Это, например, список товаров, которые приобрел покупатель. Для них можно найти следствие, нажав на кнопку  (Вычислить правила) или  (Автоматически вычислить правила). Причем в условие могут входить несколько элементов, или товаров. Тогда в следствие попадут все товары, условия которых удовлетворяют списку ассоциативных правил.

2) например, клиент заказал Зубную паста и Пятновыводитель. Что еще ему можно предложить? Поскольку у нас имеются два правила, а именно:

а) Зубная паста → Чистящий порошок;

б) Пятновыводитель → Отбеливатель;

то в следствие попадут два элемента – Чистящий порошок и Отбеливатель.

Это проиллюстрировано на рисунке 37.

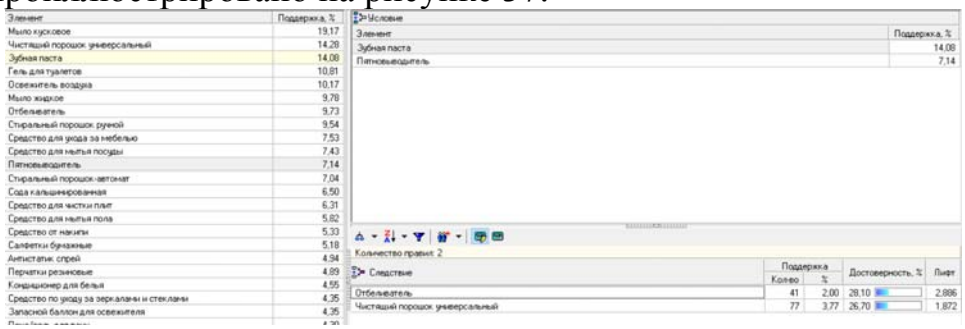


Рисунок 37 – Настройка интерактивного отчета

Информация о купленных товарах вносится продавцом в отчет, и тут же формируется набор предложений, который озвучивается клиенту.

С помощью встроенных функций можно настроить интерактивный отчет с предложениями-напоминаниями для клиента. Для этого используются следующие кнопки:

 Автоматически вычислить правила – позволяет рассчитывать новый набор следствий после каждого добавления товара в набор условий;

 Порядок сортировки,  Направление сортировки – позволяют упорядочить «список предложений» по одному из выбранных параметров (поддержка, достоверность, лифт);

 Фильтрация правил – задаются ограничения на формируемый «список предложений»;

 Тип определения лучшего правила – при повторении наименований товаров в «списке предложений» позволяет отображать один из них в соответствии с лучшей характеристикой.

Результаты сохраните в файле L3.ded.

Задание

1 Определите, что приобретет клиент купивший: Гель для туалетов; Средство для мытья посуды и Пятновыводитель, с величиной лифта больше 3.

2 Небольшая сеть из трех магазинов, продающих мелкие штучные товары, желает провести исследование связанных покупок. По мнению специалистов компании, знание того, какие товары покупаются совместно, поможет правильно расположить их на витринах. Таблица данных содержится в файле Чеки.txt и включает два столбца: Транзакция и Товар. Опираясь на имеющиеся данные, выполните следующие действия.

1) решите задачу поиска ассоциаций в Deductor (достоверность 20–50, правил – 17).

2) выделите непонятные, на ваш взгляд, ассоциативные правила, а также правила, представляющие интерес. Сколько правил попало в эти категории?

3) найдите правило, имеющее максимальный лифт.

4) заказчика данного исследования интересует, какие товары покупают с поздравительной открыткой (используйте все визуализаторы).

Сколько таких товаров оказалось в выбранном перечне? Какая из ассоциаций представляет в этом плане наибольший интерес (имеет максимальный лифт)?

Результаты сохраните в файле L4.ded.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Макшанов А. В. Технологии интеллектуального анализа данных : учебное пособие / А. В. Макшанов, А. Е. Журавлев. – 2-е изд., стер. – Санкт-Петербург : Лань, 2019. – 212 с. // Электронно-библиотечная система «Лань». – URL: (дата обращения: 01.10.2025).

2 Мицель А. А. Прикладная математическая статистика : учебное пособие / А. А. Мицель. – Томск : ТУСУР, 2019. – 113 с. – URL: https://edu.tusur.ru/publications/9151 (дата обращения: 01.10.2025).

3 Форман Д. Много цифр: Анализ больших данных при помощи Excel / Д. Форман ; перевод А. Соколовой. – Москва : Альпина Паблишер, 2016. – 461 с. // Электронно-библиотечная система Индекс «Лань» – URL: https://e.lanbook.com/book/87871 (дата обращения: 01.10.2025).

Адаменко Юлия Владимировна

Анализ данных. Часть 2. Классификация и регрессия. Методы машинного обучения
Методические указания к выполнению лабораторных работ
для бакалавров направлений
09.03.03 «Прикладная информатика»,
09.03.04 «Программная инженерия»

Редактор Н. М. Быкова

БИЦ Курганского государственного университета.
640020, г. Курган, ул. Советская, 63/4.
Курганский государственный университет.