

Научно Исследовательский Семинар 2016

# Математические модели в экономике

Sergei Koltcov  
**skoltsov@hse.ru**  
<https://linis.hse.ru>



# Содержимое



**НИС  
2016**

**7. Элементы теории вероятности.** Понятие о правиле Байеса. Применение теории вероятности для анализа оценки надежности компаний

**8. Применение классификаторов для анализа акций ценных бумаг.** Работа классификатора 'Naive Bayes'.

**9. Качество классификации.** Precision, Reccall, Confusion matrix, F measure, ROC, AUC.

**10. Работа классификатора 'SVM'.**

**11. Работа классификатора 'Логистическая регрессия'.**

**11. Сравнение трех классификаторов.**

# Различия в подходах к теории вероятностей

Случайная величина — это величина, которая принимает в результате опыта одно из множества значений, причём появление того или иного значения этой величины до её измерения нельзя точно предсказать.

1. В частотном подходе (классический подход) предполагается, что случайность есть объективная неопределенность. Вероятность рассчитывается из серии экспериментов и является мерой случайности как эмпирической данности. Исторически частотный подход возник из практической задачи: анализа азартных игр — области, в которой понятие серии испытаний имеет простой и ясный смысл.
2. В байесовском подходе предполагается, что случайность характеризует наше незнание. Например, случайность при бросании кости связана с незнанием динамических характеристик игровой кости, сопротивления воздуха и так далее.

Многие задачи частотным методом решить невозможно (точнее, вероятность искомого события строго равна нулю). В то же время интерпретация вероятности как меры нашего незнания позволяет получить отличный от нуля осмысленный ответ.

# Понятие вероятности

**Вероятность события** — Вероятностью события  $A$  называют отношение числа благоприятствующих этому событию исходов к общему числу всех равновозможных несовместных элементарных исходов. Например. Вероятность того, что на кубике выпадет четное число, равна следующему отношению  $P=3/6=1/2$ .



**Условной вероятностью** события  $A$  при условии, что произошло событие  $B$ , называется число  $P(A|B)=P(B, A)/ P(B)$ ,  
 $P(B, A)$  — произведение вероятностей,  $P(B)$  — полная вероятность события  $B$ .

**Например.** В урне 3 белых и 3 черных шара. Из урны дважды вынимают по одному шару, не возвращая их обратно. Найти вероятность появления белого шара при втором испытании (**событие  $A$** ), если при первом испытании был извлечен черный шар (**событие  $B$** ).

**Решение задачи:**

**Событие  $B$**  — это вытаскивание первого шага (а именно черного). Вероятность события  $B=3/6=1/2$  — вер. вытащить черный шар.

**События  $A$**  — это вытаскивание второго шара (а именно белого), так как в урне осталось 5 шаров, то вероятность этого события  $A=3/5$

Таким образом, совместная вероятность событий  $A$  и  $B$  это произведение вероятностей этих событий  $P(B, A) = (3/6) * (3/5) = 9/30$

Полная вероятность события  $B=1/2$

**Итоговый результат:**  $\{3/6 * 3/5\} / (1/2) = 3/5$

# Формула Байеса

**Байесовская вероятность** — это интерпретация понятия вероятности, используемое в байесовской теории. Вероятность определяется как степень уверенности в истинности суждения.

$$P(A | B) = \frac{P(B | A) \cdot P(A)}{P(B)}$$

**$P(A)$**  — **априорная вероятность** гипотезы  $A$  (заранее известная вероятность);

**$P(A|B)$**  — вероятность гипотезы  $A$  при наступлении события  $B$  (**апостериорная вероятность**);

**$P(B|A)$**  — вероятность наступления события  $B$  при истинности гипотезы  $A$ ;

**$P(B)$**  — полная вероятность наступления события  $B$ .

**$P(A|B)$**  — вероятность наступления события  $A$  при истинности гипотезы  $B$ ;

Формула Байеса позволяет «переставить причину и следствие»: по известному факту события вычислить вероятность того, что оно было вызвано данной причиной. Таким образом, формула Байеса может быть использована для разработки алгоритмов классификации.



# Априорные и апостериорные суждения

1. Предположим, мы хотим узнать значение некоторой неизвестной величины.
2. У нас имеются некоторые знания, полученные до (a priori) наблюдений/эксперимента. Это может быть опыт прошлых наблюдений, какие-то модельные гипотезы, ожидания.
3. В процессе наблюдений эти знания подвергаются постепенному уточнению. После (a posteriori) наблюдений/эксперимента у нас формируются новые знания о явлении.
4. Будем считать, что мы пытаемся оценить неизвестное значение величины  $P(A|B)$  посредством наблюдений некоторых ее косвенных характеристик (гипотез).
5. В зависимости от уровня вероятности мы можем принять или отвергнуть нашу гипотезу (предсказание)

Если у нас много событий то мы предполагаем что они не зависят друг от друга. Например, мы считали что процесс вытаскивания шара из урны не зависит от цвета шара. В связи с таким допущением алгоритм называется «**наивным**».

# Пример оценки надежности компании

Пусть нам нужно оценить надежность компании. Мы предполагаем, что у нас есть три гипотезы о надежности ( $\Pr(\theta_{i:1,2,3})$ ). 1. Средняя надежность. 2. Высокая надежность. 3. Низкая надежность.

## Априорные значения

Номер гипотезы $i$	Средняя надежность (Pr1)	Высокая надежность (Pr2)	Низкая надежность (Pr3)
$\Pr(\theta_i)$ (число компаний имеющих разные уровни надежности)	0.5 (50%)	0.3 (30%)	0.2 (20%)
Число компаний имеющие прибыль $\Pr(y_1; \theta_i)$	0.4 (40%)	0.8 (80%)	0.3 (30%)
Число компаний, осуществляющие своевременный расчет с гос. $\Pr(y_2; \theta_i)$	0.7 (70%)	0.9 (90%)	0(0%)

Вопрос, как будут меняться вероятности гипотез (**Pr1, Pr2, Pr3**) если мы наблюдаем какую либо величину? Расчет вероятности гипотез ведется при помощи формулы Байеса.

$$\Pr(\theta_j|y) = \frac{\Pr(y|\theta_j) \Pr(\theta_j)}{\Pr(y)} = \frac{\Pr(y|\theta_j) \Pr(\theta_j)}{\sum_{i=1}^m \Pr(\theta_i) \Pr(y|\theta_i)}.$$

# Пример оценки надежности компании

Пусть мы наблюдаем компанию у которой есть прибыль. Тогда гипотеза (апостериорное значение) того, что данная компания относится к типу средней надежности будет рассчитываться следующим образом.

$$Pr1 = \frac{0.4 * 0.5}{0.4 * 0.5 + 0.8 * 0.3 + 0.3 * 0.2} = 0.4 \text{ (было 0.5)}$$

Вероятность гипотезы о высокой надежности:

$$Pr2 = \frac{0.8 * 0.3}{0.4 * 0.5 + 0.8 * 0.3 + 0.3 * 0.2} = 0.48 \text{ (было 0.3)}$$

Вероятность гипотезы о низкой надежности:

$$Pr3 = \frac{0.3 * 0.2}{0.4 * 0.5 + 0.8 * 0.3 + 0.3 * 0.2} = 0.12 \text{ (было 0.2)}$$

Таким образом мы получили апостериорные оценки, которые потом можно использовать как априорные.



# Пример оценки надежности компании

Предположим, что фирма, которая имеет прибыль, еще и платит своевременно долги.

Номер гипотезы $i$	Средняя надежность	Высокая надежность	Низкая надежность
$Pr(\theta_i)$ (число компаний имеющих разные уровни надежности)	0.4	0.48	0.12
Число компаний имеющие прибыль $Pr(y_1; \theta_i)$	0.4	0.48	0.12
Число компаний, осуществляющие своевременный расчет с гос. $Pr(y_2; \theta_i)$	0.7	0.9	0

Тогда новые вероятности гипотез рассчитываются на основании предыдущих расчетов.

$$Pr1 = \frac{0.4 * 0.7}{0.7 * 0.4 + 0.48 * 0.9 + 0 * 0.12} = 0.39 \text{ (было 0.4)}$$

$$Pr2 = \frac{0.48 * 0.9}{0.7 * 0.4 + 0.48 * 0.9 + 0 * 0.12} = 0.607 \text{ (было 0.48)}$$

$$Pr3 = \frac{0.12 * 0}{0.7 * 0.4 + 0.48 * 0.9 + 0 * 0.12} = 0 \text{ (было 0.12)}$$

# Вероятностная постановка задачи классификации

Пусть имеется множество объектов  $X$  и конечное множество классов  $Y$ . Требуется построить алгоритм способный классифицировать произвольный объект  $X$  в рамках заданного множества  $Y$ . Апостериорная вероятность принадлежности объекта  $X$  классу  $Y$  по формуле Байеса:

$$P(X | Y) = \frac{p(X, Y)}{P(X)} = \frac{p(X)P(Y | X)}{P(X)}$$

$P(X | Y)$  - Апостериорная вероятность

$p(X, Y)$  - Априорная вероятность

*Задача классификации заключается в расчете (оценке) апостериорной информации на основании априорной информации. Такая оценка может быть реализована при помощи формулы Байеса. Однако существует проблема оценивания априорной величины  $p(x, y)$*

# Другой пример работы байесовского алгоритма

Пусть у нас есть набор данных, содержащий один признак «Погодные условия» (weather) и целевую переменную «Игра» (play), которая обозначает возможность проведения матча. На основе погодных условий мы должны определить (предсказать), состоится ли матч.

1. Пусть у нас есть набор наблюдений

	A	B
1	weather	play
2	sunny	no
3	overcast	yes
4	rainy	yes
5	sunny	yes
6	sunny	yes
7	overcast	yes
8	rainy	no
9	rainy	no
10	sunny	yes
11	rainy	yes
12	sunny	no
13	overcast	yes
14	overcast	yes
15	rainy	no

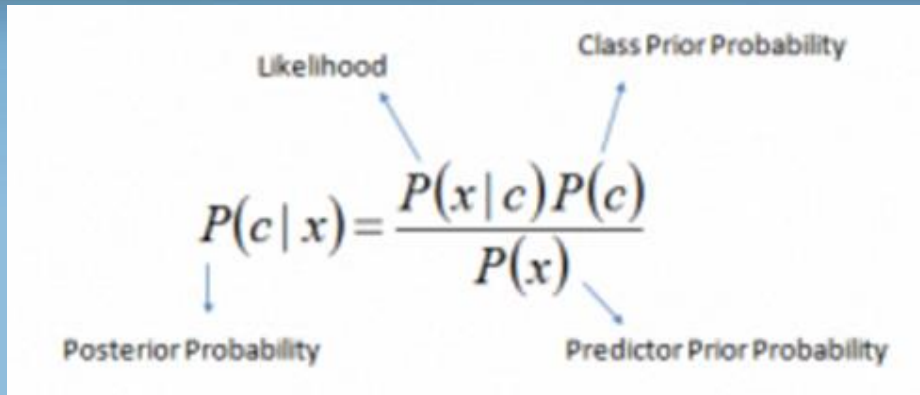
2. Преобразуем данные в таблицу частот

A	B	C
weather	no	yes
overcast	0	4
rainy	3	2
sunny	2	3
total	5	9

3. Преобразуем частоты в таблицу вероятности

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

# Другой пример работы байесовского алгоритма



The diagram shows the formula for the posterior probability: 
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$
 Four blue arrows point from text labels to parts of the formula: 'Likelihood' points to  $P(x|c)$ , 'Class Prior Probability' points to  $P(c)$ , 'Posterior Probability' points to  $P(c|x)$ , and 'Predictor Prior Probability' points to  $P(x)$ .

$P(c/x)$  – апостериорная вероятность данного класса  $c$  (т.е. данного значения целевой переменной) при данном значении признака  $x$ .

$P(c)$  – априорная вероятность данного класса.

$P(x/c)$  – правдоподобие, т.е. вероятность данного значения признака при данном классе. Это может быть произведением множества функций, в которых определена зависимость признака на множестве классов.

$P(x)$  – априорная вероятность данного значения признака.

**Задача: какова вероятность проведения матча в зависимости от погоды.**

Решение:

$X$  – это **Да** или **Нет** (то есть у нас два класса).

$C$  – типы погоды (overcast, sunny, rainy) - признаки

# Другой пример работы байесовского алгоритма

Вероятность проведения матча в солнечную погоду.

По формуле Байеса:  $P(\text{yes/sunny}) = \frac{P(\text{sunny/yes}) * P(\text{yes})}{P(\text{sunny})}$

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

$$P(\text{sunny/yes}) = 3/9$$

$$P(\text{yes}) = 9/14$$

$$P(\text{sunny}) = 5/14$$

$$P(\text{yes/sunn}) = \frac{\left(\frac{3}{9}\right) * \left(\frac{9}{14}\right)}{P\left(\frac{5}{14}\right)} = 0.6$$

$$P(\text{no/sunny}) = \frac{P(\text{sunny/no}) * P(\text{no})}{P(\text{sunny})}$$

$$P(\text{sunny/no}) = 2/5$$

$$P(\text{no}) = 5/14$$

$$P(\text{sunny}) = 5/14$$

$$P(\text{no/sunn}) = \frac{\left(\frac{2}{5}\right) * \left(\frac{5}{14}\right)}{P\left(\frac{5}{14}\right)} = 0.4$$

# Плюсы и минусы наивного байесовского алгоритма

## Положительные стороны:

1. Классификация, в том числе многоклассовая, выполняется легко и быстро.
2. НБА лучше работает с категориальными признаками, чем с непрерывными.

## Отрицательные стороны:

1. Если в тестовом наборе данных присутствует некоторое значение категориального признака, которое не встречалось в обучающем наборе данных, тогда модель присвоит нулевую вероятность этому значению и не сможет сделать прогноз. Это явление известно под названием «нулевая частота».
2. Еще одним ограничением НБА является допущение о независимости признаков. В реальности наборы полностью независимых признаков встречаются крайне редко.

## ОБЛАСТИ ПРИМЕНЕНИЯ

1. **Классификация в режиме реального времени.** НБА очень быстро обучается, поэтому его можно использовать для обработки данных в режиме реального времени (котировки акций).
2. **Многоклассовая классификация.** НБА обеспечивает возможность многоклассовой классификации.
3. **Классификация текстов, фильтрация спама, анализ тональности текста.** При решении задач, связанных с классификацией текстов, НБА превосходит многие другие алгоритмы. Благодаря этому, данный алгоритм находит широкое применение в области фильтрации спама (идентификация спама в электронных письмах) и анализа тональности текста (анализ социальных медиа, идентификация позитивных и негативных мнений клиентов).



# Классификация на основе Naïve Bayes в Orange

**Котировка акций Газпрома: с 01.01.2016 – 18.03.2016**

<DATE>	<TIME>	<OPEN>	<HIGH>	<LOW>	<CLOSE>	difference
20160104	110000	135.89	136.65	135.62	135.96	0.07
20160104	120000	135.92	135.92	134.42	134.67	-1.25
20160104	130000	134.71	135.24	134.42	135.15	0.44
20160104	140000	135.12	135.22	134.7	135.22	0.1
20160104	150000	135.23	135.49	134.88	135.11	-0.12
20160104	160000	135.11	135.35	134.94	134.97	-0.14
20160104	170000	134.97	135.3	134.8	135.13	0.16
20160104	180000	135.13	135.4	134.96	135.15	0.02
20160104	190000	135.14	135.45	134.9	134.91	-0.23
20160105	110000	134.85	135.35	134.85	135.06	0.21
20160105	120000	135.06	136.16	135	135.93	0.87
20160105	130000	135.92	135.94	135.1	135.31	-0.61
20160105	140000	135.3	135.3	134.61	135.07	-0.23
20160105	150000	135.08	135.23	134.76	135.22	0.14
20160105	160000	135.21	135.75	135.01	135.64	0.43
20160105	170000	135.64	137.28	135.53	136.74	1.1
20160105	180000	136.71	137.15	136.18	136.19	-0.52
20160105	190000	136.23	136.58	136.08	136.45	0.22
20160106	110000	136.35	136.93	136.17	136.51	0.16
20160106	120000	136.51	136.72	136	136.46	-0.05

**Задача** заключается в том ,что бы построить на основе Байесовского классификатора алгоритм предсказания котировок ценных бумаг.

**Решение:** будем использовать пакет Orange.

1. Возьмем исходный датасет.
2. Обучим классификатора на наших данных.
3. Посмотрим как классификатор предсказывает на наших данных котировки акций Газпрома.

# Классификация на основе Naïve Bayes в Orange

## Данные в формате Orange

	A	B	C	D	E	F
1	<OPEN>	<HIGH>	<LOW>	<CLOSE>	difference	up_down
2	c	c	c	c	c	d
3						class
4	135.89	136.65	135.62	135.96	0.07	up
5	135.92	135.92	134.42	134.67	-1.25	down
6	134.71	135.24	134.42	135.15	0.44	up
7	135.12	135.22	134.7	135.22	0.1	up
8	135.23	135.49	134.88	135.11	-0.12	down
9	135.11	135.35	134.94	134.97	-0.14	down
10	134.97	135.3	134.8	135.13		
11	135.13	135.4	134.96	135.15		
12	135.14	135.45	134.9	134.91		
13	134.85	135.35	134.85	135.06		
14	135.06	136.16	135	135.93		
15	135.92	135.94	135.1	135.31		

Data Table (1)



Info  
465 instances (no missing values)  
5 features (no missing values)  
Discrete class with 3 values (no missing values)  
No meta attributes

Restore Original Order

Variables

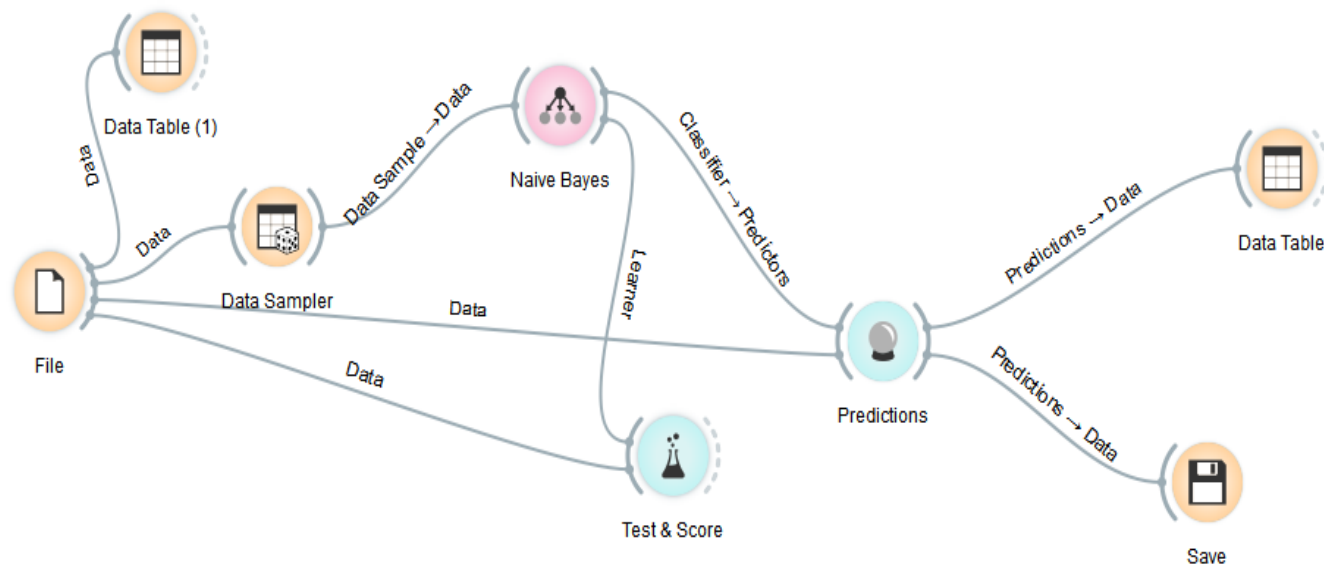
- ☒ Show variable labels (if present)
- ☒ Visualize continuous values
- ☒ Color by instance classes

Открываем новый проект в Orange, и кладем на холст виджеты:

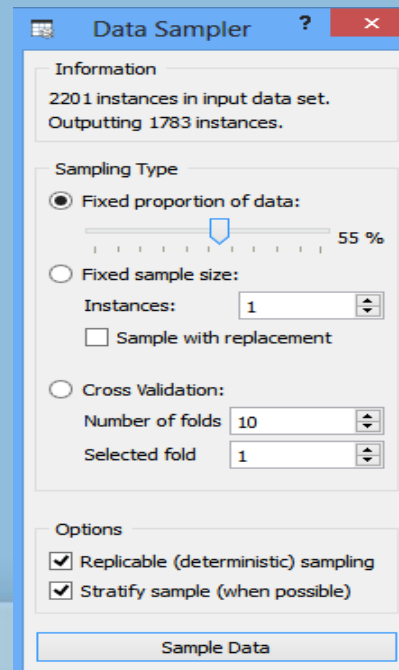
1. File (опция Data).
2. Data Table. (опция Data).
3. Кликаем на File и загружаем данные по Газпрому.
4. Кликаем на Data Table и смотрим что загрузилось в Orange.

	<OPEN>	<HIGH>	<LOW>	<CLOSE>	difference	up_down
1	135.890	136.650	135.620	135.960	0.070	up
2	135.920	135.920	134.420	134.670	-1.250	down
3	134.710	135.240	134.420	135.150	0.440	up
4	135.120	135.220	134.700	135.220	0.100	up
5	135.230	135.490	134.880	135.110	-0.120	down
6	135.110	135.350	134.940	134.970	-0.140	down
7	134.970	135.300	134.800	135.130	0.160	up
8	135.130	135.400	134.960	135.150	0.020	up
9	135.140	135.450	134.900	134.910	-0.230	down
10	134.850	135.350	134.850	135.060	0.210	up

# Классификация на основе Naive Bayes в Orange



Добавим в наш проект следующие виджеты: 1. Data Sampler (опция Data). 2. Naive Bayes (опция Classify). 3. Predictions ( опция Evaluate). 4. Test & Score. 5. Data Table и Save.



**Data Sampler:** данный виджет позволяет разбить нашу коллекцию на две части, и одну передать дальше для обучения.

**Fixed proportion of data** - можно задать процент данных, которые будут использованы для обучения (в данной примере 45%).

**Fixed sample size percentage** - можно задать фактическое число наблюдений в данных.

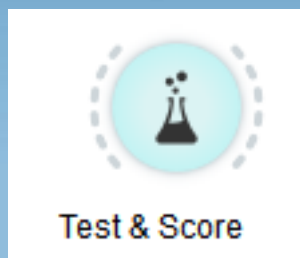
# Классификация на основе Naive Bayes в Orange

**Naive Bayes:** данный виджет который обучается на заданной коллекции данных.

**Predictions:** данный виджет на входе получает данные из двух источников (1. Классификатор. 2. Оригинальные данные). На выходе виджета исходные данные + предсказания по данным. Эти совместные данные можно отправить в виджет Data Table.

Data Table										
<div>Info</div> <div>465 instances (no missing values)</div> <div>5 features (no missing values)</div> <div>Discrete class with 3 values (no missing values)</div> <div>4 meta attributes (no missing values)</div> <div>Restore Original Order</div> <div>Variables</div> <div><input checked="" type="checkbox"/> Show variable labels (if present)</div> <div><input checked="" type="checkbox"/> Visualize continuous values</div> <div><input checked="" type="checkbox"/> Color by instance classes</div> <div>Selection</div> <div><input type="checkbox"/> Select full rows</div>										
	<OPEN>	<HIGH>	<LOW>	<CLOSE>	difference	up_down	Naive Bayes	Naive Bayes(down)	Naive Bayes(flat)	Naive Bayes(up)
1	135.890	136.650	135.620	135.960	0.070	up	down	0.894	0.008	0.098
2	135.920	135.920	134.420	134.670	-1.250	down	down	0.971	0.019	0.010
3	134.710	135.240	134.420	135.150	0.440	up	up	0.015	0.022	0.962
4	135.120	135.220	134.700	135.220	0.100	up	up	0.016	0.023	0.961
5	135.230	135.490	134.880	135.110	-0.120	down	down	0.859	0.057	0.085
6	135.110	135.350	134.940	134.970	-0.140	down	down	0.859	0.057	0.085
7	134.970	135.300	134.800	135.130	0.160	up	up	0.016	0.023	0.961
8	135.130	135.400	134.960	135.150	0.020	up	down	0.859	0.057	0.085
9	135.140	135.450	134.900	134.910	-0.230	down	down	0.859	0.057	0.085
10	134.850	135.350	134.850	135.060	0.210	up	up	0.016	0.023	0.961
11	135.060	136.160	135.000	135.930	0.870	up	up	0.015	0.011	0.974
12	135.920	135.940	135.100	135.310	-0.610	down	down	0.971	0.019	0.010
13	135.300	135.300	134.610	135.070	-0.230	down	down	0.859	0.057	0.085
14	135.080	135.230	134.760	135.220	0.140	up	up	0.016	0.023	0.961
15	135.210	135.750	135.010	135.640	0.430	up	up	0.016	0.023	0.961
16	135.640	137.280	135.530	136.740	1.100	up	up	0.014	0.003	0.983
17	136.710	137.150	136.180	136.190	-0.520	down	down	0.986	0.001	0.013
18	136.230	136.580	136.080	136.450	0.220	up	up	0.013	0.001	0.985
19	136.350	136.930	136.170	136.510	0.160	up	up	0.013	0.001	0.985
20	136.510	136.720	136.000	136.460	-0.050	down	down	0.888	0.004	0.108
21	136.440	136.700	136.140	136.240	-0.200	down	down	0.888	0.004	0.108
22	136.220	136.300	135.400	135.450	-0.770	down	down	0.979	0.010	0.012
23	135.420	135.850	135.300	135.680	0.260	up	up	0.016	0.023	0.961

# Качество классификации на основе Naive Bayes в Orange



Данный виджет автоматически рассчитывает основные метрики качества:

1. Precision (точность).
2. Recall (качество).
3. F1 (F мера).
4. AUC.

**Test & Score**

**Sampling**

☐ Cross validation  
Number of folds: 20

☐ Leave one out

☒ Random sampling  
Repeat train/test: 10  
Relative training set size: 59 %

☐ Test on train data  
☐ Test on test data

Apply

**Target class**  
(Average over classes)

**Evaluation Results**

Method	AUC	CA	F1	Precision	Recall
Naive Bayes	0.977	0.980	0.977	0.974	0.980

Report



# МЕРЫ КАЧЕСТВА КЛАССИФИКАТОРОВ ДЛЯ БИНАРНЫХ КЛАССОВ

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Precision: число правильно предсказанных положительных значений поделенных на число предсказанных классификатором положительных значений.

Recall : число правильно предсказанных положительных значений поделенных на число положительных ответов в данных.

## Confusion Matrix

		Predicted 1	Predicted 0
True 1	True 1	true positive	false negative
	True 0	false positive	true negative

		Predicted 1	Predicted 0
True 1	True 1	TP	FN
	True 0	FP	TN

		Predicted 1	Predicted 0
True 1	True 1	hits	misses
	True 0	false alarms	correct rejections

		Predicted 1	Predicted 0
True 1	True 1	$P(pr1 tr1)$	$P(pr0 tr1)$
	True 0	$P(pr1 tr0)$	$P(pr0 tr0)$

**TP** – число правильно предсказанных положительных значений

**FN** – число неправильно предсказанных положительных значений

**FP** – число неправильно предсказанных негативных значений

**TN** – число правильно предсказанных негативных значений



# МЕРЫ КАЧЕСТВА КЛАССИФИКАТОРОВ ДЛЯ БИНАРНЫХ КЛАССОВ

F – measure

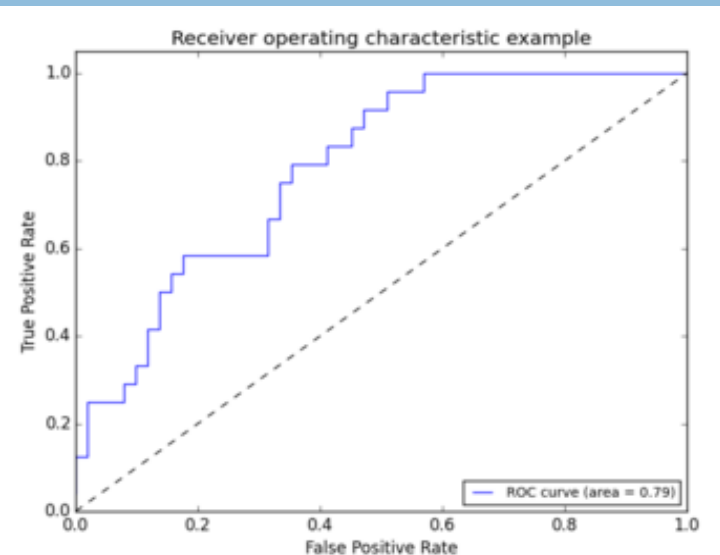
$$F - measure = \frac{(\beta^2 + 1) * precision * recall}{\beta^2 * precision + recall}$$



$$\frac{(\beta^2 + 1)tp}{(\beta^2 + 1)tp + \beta^2 fn + fp}$$

$\beta$  – обычно берут равной 1. **F measure = 2 \* (precision \* recall) / (precision + recall)**

The F measure (F1, Fscore) можно интерпретировать как взвешенное среднее precision и recall. Если F1=1, то классификатор отработал на 100% и F1=0 тогда классификатор не справился с задачей.



$$ROC = \frac{P(x|positive)}{P(x|negative)}$$

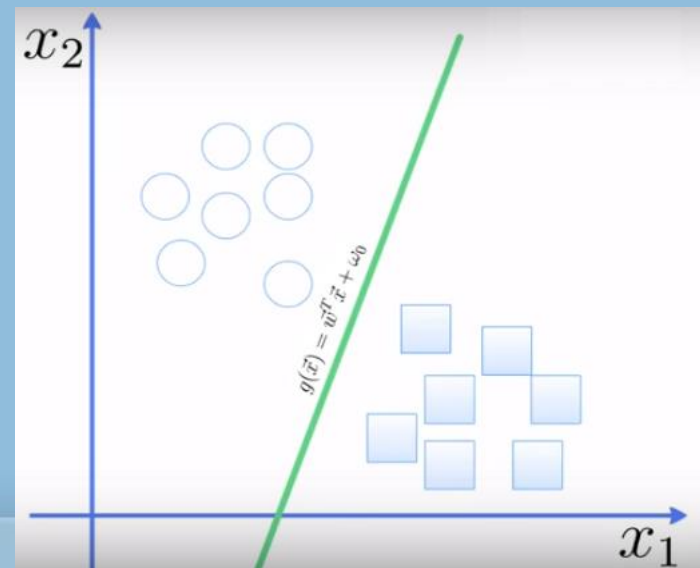
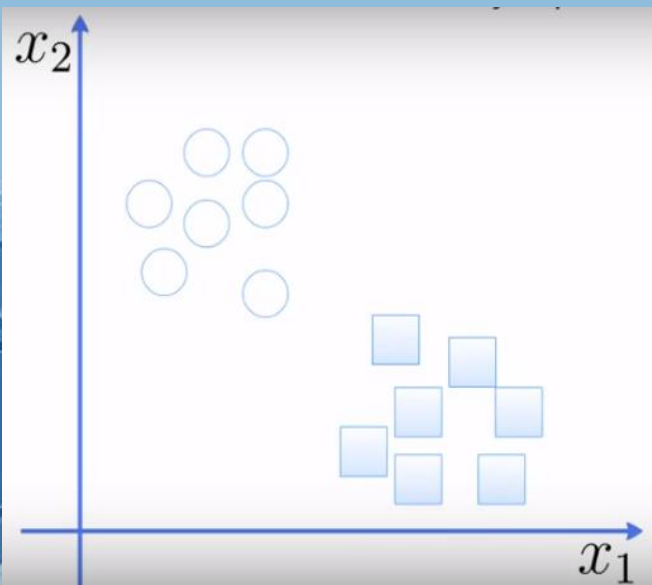
Рассчитывает отношение числа правильно распознанных случаев к числу не правильных. Процесс расчета таков: берутся данные, последовательно, и в них вычисляется это отношение. В какой то момент отношение становится константой.

AUC – интеграл под кривой.

# Метод опорных векторов (SVM)

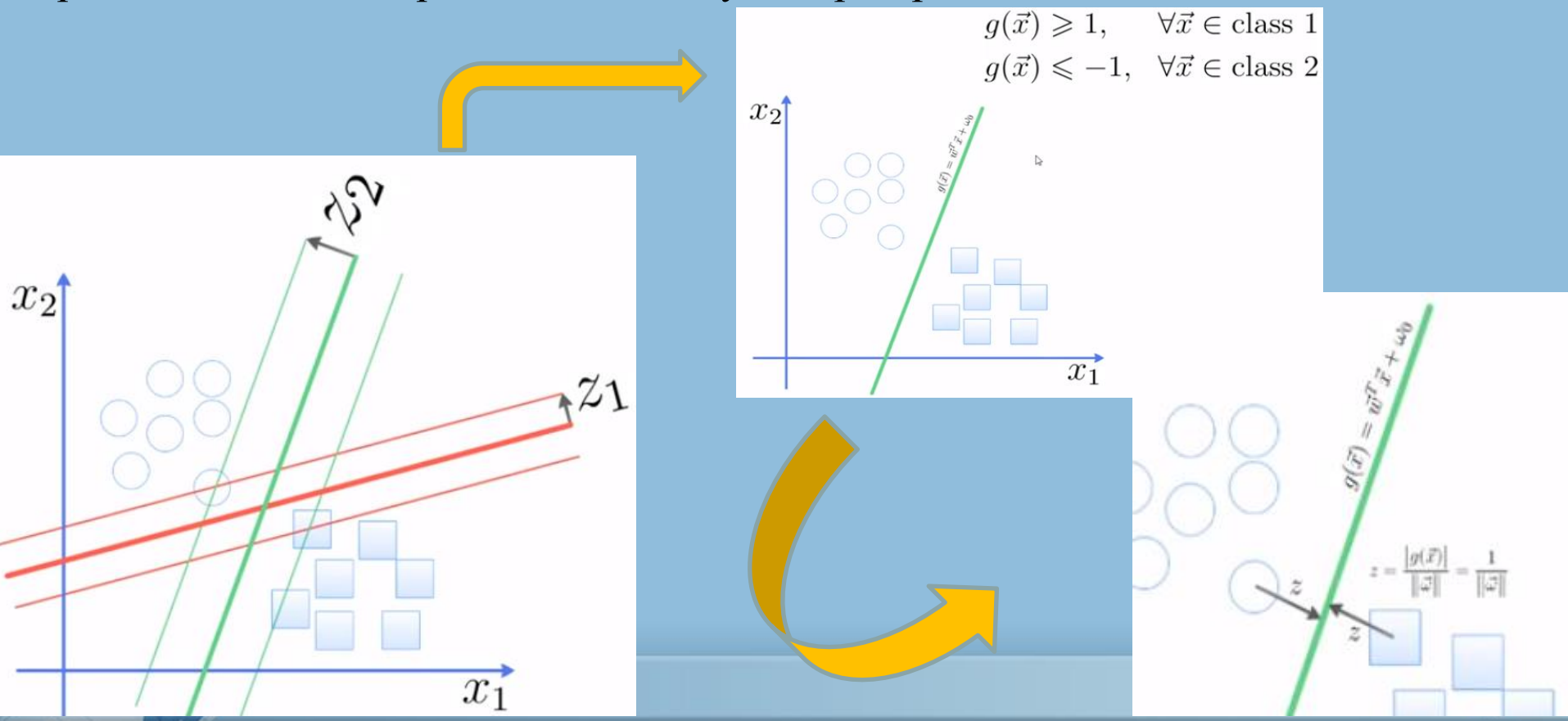
Каждый объект данных (например, документ, котировки ценных бумаг или компании) представлен как вектор в  $\mathbf{R}$  мерном пространстве (последовательность чисел). Пусть у нас есть тестовая коллекция, в которой есть набор объектов (features) и есть набор классов. Математическая задача обучения заключается в том что бы найти функцию, которая адекватно сопоставляла объекты и классы, то есть найти такую функцию, которая эффективно разделяла бы объекты в пространстве features.

**Рассмотрим пример на плоскости:** У нас есть два класса с двумя features ( $x_1$ ,  $x_2$ ). Нужно найти прямую линию, которая оптимально разделяла два класса.



# Метод опорных векторов (SVM)

Процесс расчета заключается в поиске двух опорных векторов при этом ищется максимум расстояния между опорными векторами. Вектор  $\mathbf{w}$  — перпендикулярен к разделяющей гиперплоскости. То, есть при разных вариантах проведения прямой линии, выбирается та линия у которой расстояние  $Z$  максимальное.

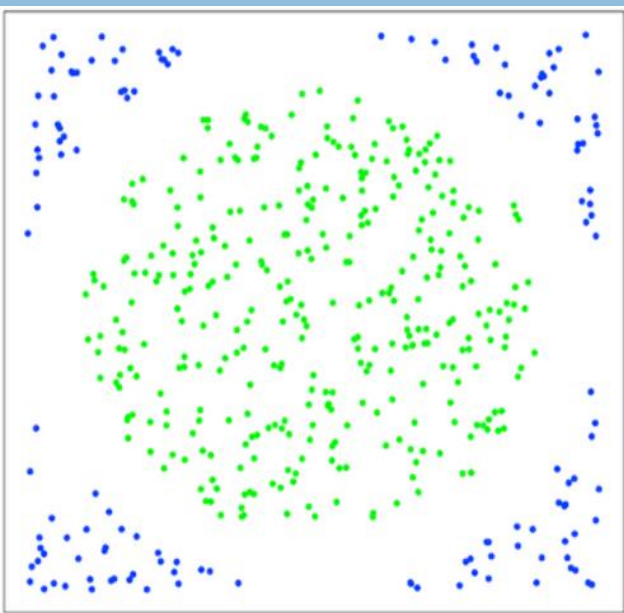


# Метод опорных векторов (SVM)

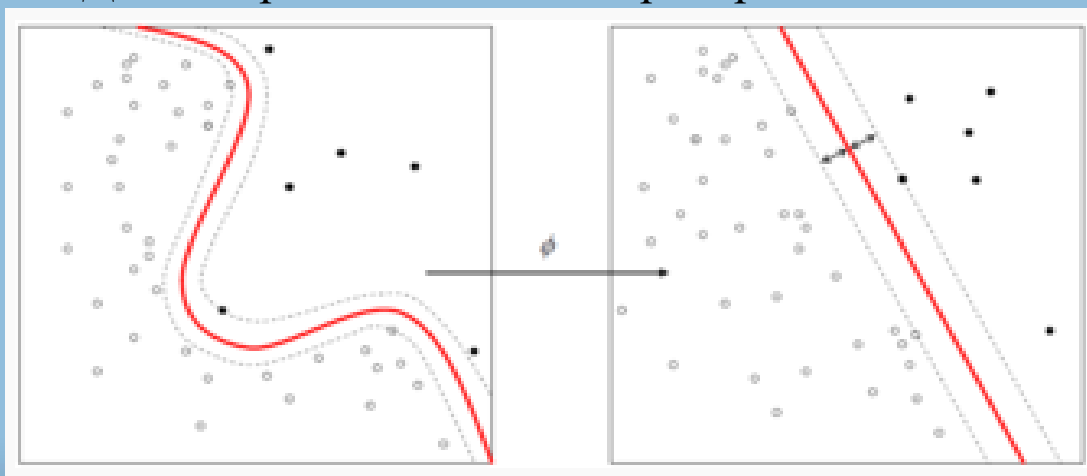
**Нахождение уравнения плоскости является** стандартной задачей квадратичного программирования и решается с помощью множителей Лагранжа. Собственно в этом заключается процесс обучения.

Как только плоскость найдена, берем новый объект и смотрим где он расположен относительно плоскости. Если справа, то принадлежит одному классу, если слева, то наш объект принадлежит другому классу.

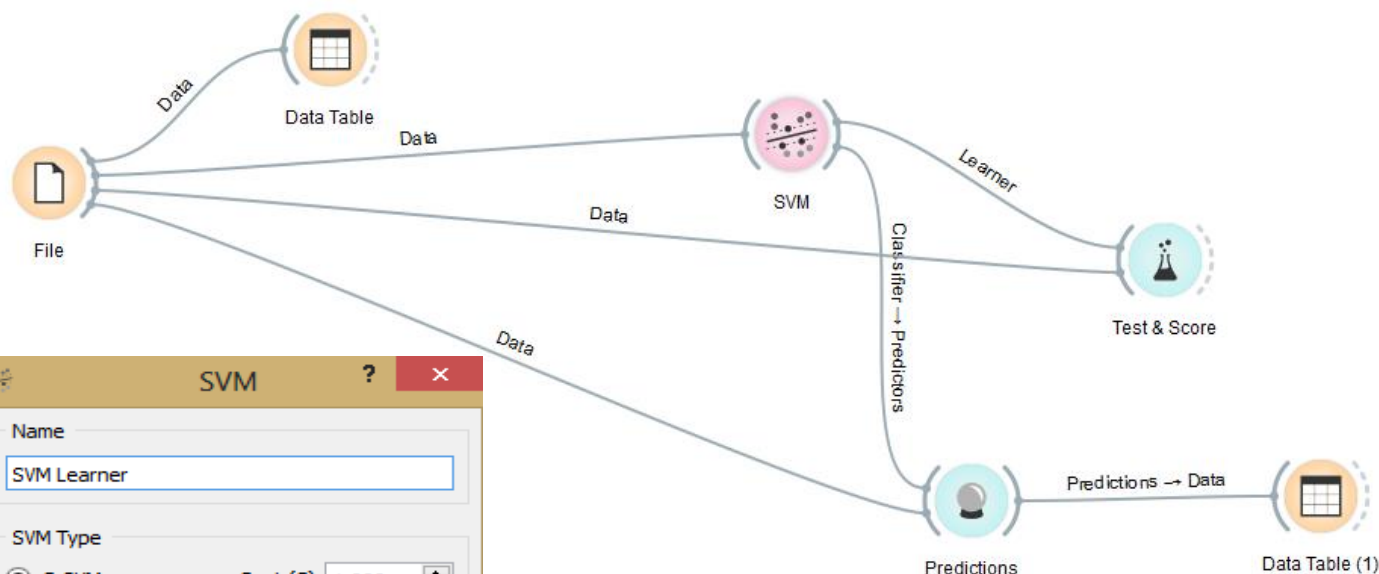
Однако, как правило на практике встречаются случаи когда объекты расположены, так что на плоскости невозможно провести разделяющую прямую. В этом случае плоскость вкладывается в пространство большей размерности. При вложении плоскость трансформируется таким образом, что бы появилась возможность провести разделяющую плоскость.



Демонстрация подобного преобразования



# Метод опорных векторов (SVM) in Orange



**SVM** ? x

Name  
SVM Learner

SVM Type  
☒ C-SVM Cost (C) 1,000  
☐ v-SVM Complexity bound (v) 0,50

Kernel  
☒ Linear,  $x \cdot y$   
☐ Polynomial,  $(g \cdot x \cdot y + c)^d$   
☐ RBF,  $\exp(-g \|x - y\|^2)$   
☐ Sigmoid,  $\tanh(g \cdot x \cdot y + c)$   
g: 0,0000 c: 0,0000 d: 3,0

Optimization parameters  
Numerical Tolerance 0,0010000  
☒ Iteration Limit 100

Apply

## Типы kernels (ядер):

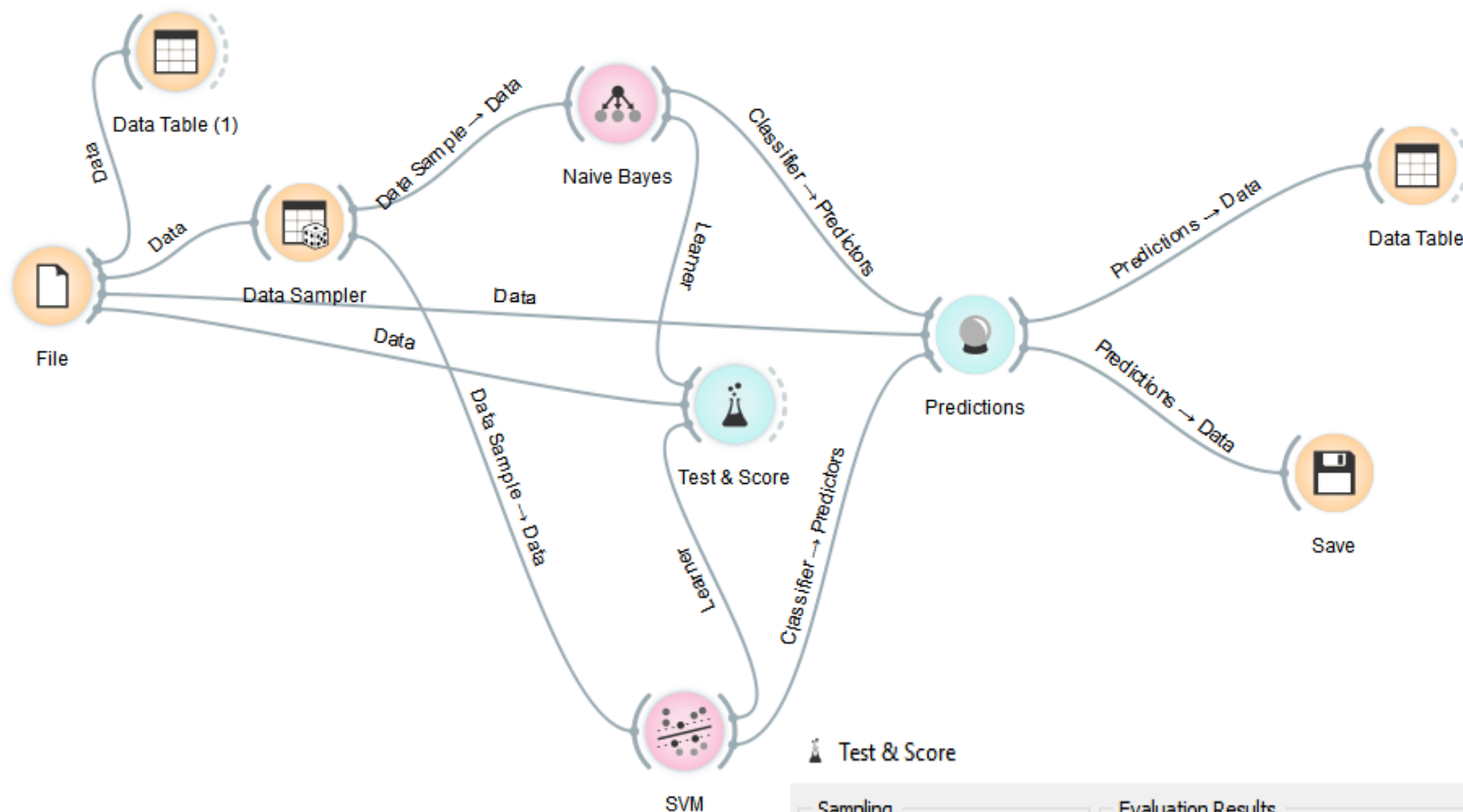
1. Линейное ядро.
  2. Полиномиальное ядро.
  3. Radial basis function kernel
  4. Hyperbolic Tangent (Sigmoid) Kernel
- c - const (default = 0)  
d - степень ядра

$$k(x, y) = (\alpha x^T y + c)^d$$

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

$$k(x, y) = \tanh(\alpha x^T y + c)$$

# Метод опорных векторов (SVM) in Orange



**SVM** дает лучшее  
предсказание на данных по  
котировкам Газпрома.

Sampling

☐ Cross validation  
Number of folds: 20

☐ Leave one out  
☒ Random sampling

Evaluation Results					
Method	AUC	CA	F1	Precision	Recall
Naive Bayes	0.977	0.980	0.977	0.974	0.980
SVM Learner	0.981	0.983	0.980	0.977	0.983



# Логистическая регрессия в задачах классификации

**Логистическая регрессия** – это разновидность множественной регрессии, общее назначение которой состоит в анализе связи между несколькими независимыми переменными (называемыми также регрессорами или предикторами) и зависимой переменной. Бинарная логистическая регрессия, как следует из названия, применяется в случае, когда зависимая переменная является бинарной (т.е. может принимать только два значения). Иными словами, с помощью логистической регрессии можно оценивать вероятность того, что событие наступит для конкретного испытуемого (больной/здоровый, возврат кредита/дефолт и т.д.).

$$y = a + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

Для решения проблемы задача регрессии может быть сформулирована иначе: вместо предсказания бинарной переменной, мы предсказываем непрерывную переменную со значениями на отрезке  $[0,1]$  при любых значениях независимых переменных. Это достигается применением следующего регрессионного уравнения (логит-преобразование):

$$p = \frac{1}{1+e^{-y}} ,$$

где  $P$  – вероятность того, что произойдет интересующее событие;  $e$  – основание натуральных логарифмов  $2,71\dots$ ;  $y$  – стандартное уравнение регрессии.

# Логистическая регрессия – как это работает

У нас есть линейная регрессия, в которой присутствуют переменные  $x_i$  и коэффициенты регрессии  $b_i$ .

$$y = a + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

У нас есть также обучающая коллекция, которая представлена в виде совокупности пар  $\{y_i, x_i\}$ . Коэффициенты  $b_i$  не известны. **Как можно найти эти коэффициенты из обучающей коллекции?**

Мы предполагаем что совместная вероятность появления величин  $y_i$  в выборке это произведение вероятностей, то есть:

$$\mathbf{b} = \operatorname{argmax}_{\theta} L(\mathbf{b}) = \operatorname{argmax}_{\theta} \prod_{i=1}^m \operatorname{Pr}\{y = y^{(i)} | x = x^{(i)}\}.$$

Что бы избавиться от произведений, можно взять логарифм от произведения, в результате этой операции получим сумму:

$$\log L(\mathbf{b}) = \sum_{i=1}^m \log \operatorname{Pr}\{y = y^{(i)} | x = x^{(i)}\}$$

Подставляя в последнюю формулу  $\{y_i, x_i\}$  и меняя коэффициенты  $b_i$ , так что бы в формуле получилось максимальное число. Такой подход называется **методом максимального правдоподобия**.

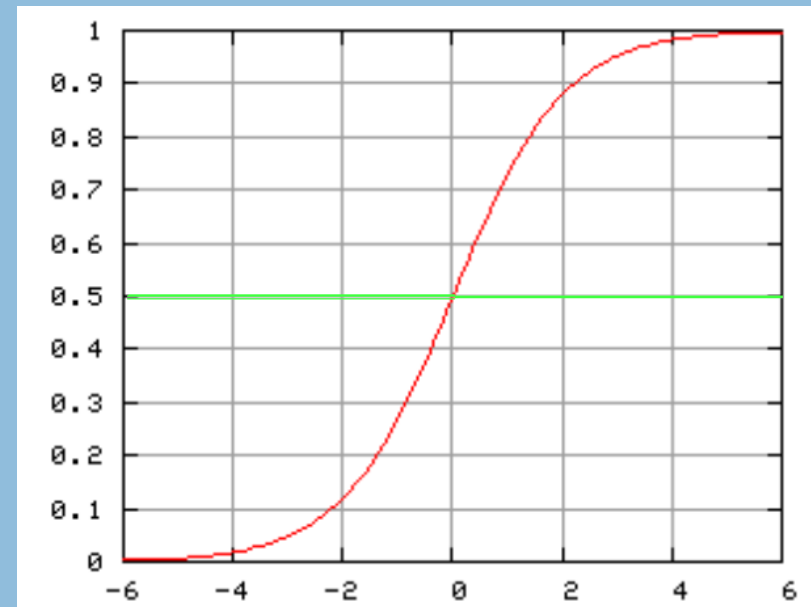
# Логистическая регрессия – как это работает

После того как нашли коэффициенты в линейной регрессии на основании обучающей коллекции:

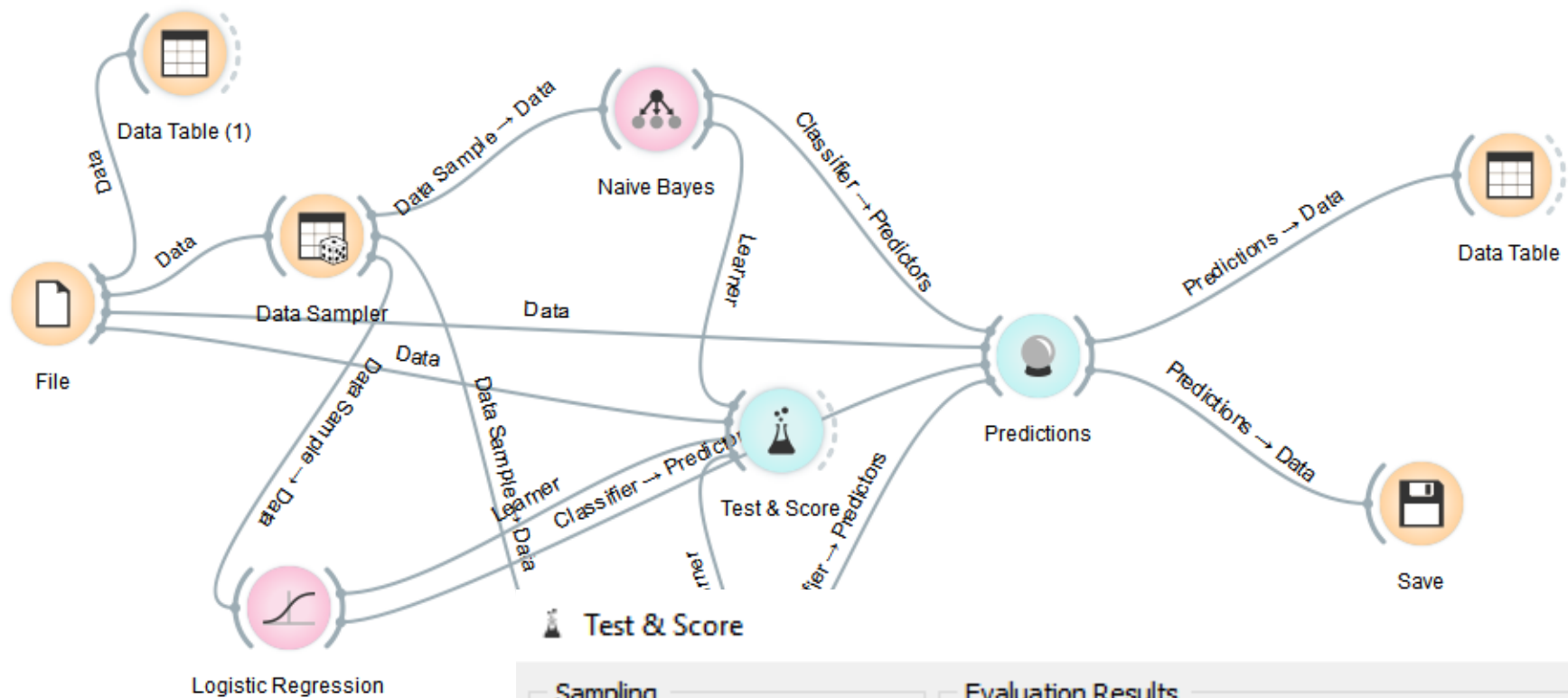
$$y = a + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

Мы теперь можем эти коэффициенты использовать для классификации следующим образом:

1. Берем величину вектор  $X(x_i)$  подставляем его в выше приведенную формулу ( $b_i$  известны).
2. Ищем  $Y = X(x_i) * B(b_i)$ .
3. Подставляем  $Y$  в логлит формулу.
4. Если  $p > 0.5$  то объект принадлежит одному классу если  $p < 0.5$ , то другому классу.



# Логистическая регрессия в Orange



## Sampling

☐ Cross validation

Number of folds: 20

☐ Leave one out

☒ Random sampling

Repeat train/test 10

## Evaluation Results

Method	AUC	CA	F1	Precision	Recall
Naive Bayes	0.977	0.980	0.977	0.974	0.980
SVM Learner	0.981	0.983	0.980	0.977	0.983
Logistic Regression	0.981	0.982	0.979	0.976	0.982

# Предсказание котировок акций Газпрома при помощи NB, SVM, Log. regression

	<OPEN>	<HIGH>	<LOW>	<CLOSE>	difference	up_down	Naive Bayes	SVM Learner	Logistic Regression
1	135.890	136.650	135.620	135.960	0.070	up	down	up	up
2	135.920	135.920	134.420	134.670	-1.250	down	down	down	down
3	134.710	135.240	134.420	135.150	0.440	up	up	up	up
4	135.120	135.220	134.700	135.220	0.100	up	up	up	up
5	135.230	135.490	134.880	135.110	-0.120	down	down	down	down
6	135.110	135.350	134.940	134.970	-0.140	down	down	down	down
7	134.970	135.300	134.800	135.130	0.160	up	up	up	up
8	135.130	135.400	134.960	135.150	0.020	up	down	up	down
9	135.140	135.450	134.900	134.910	-0.230	down	down	down	down
10	134.850	135.350	134.850	135.060	0.210	up	up	up	up
11	135.060	136.160	135.000	135.930	0.870	up	up	up	up
12	135.920	135.940	135.100	135.310	-0.610	down	down	down	down
13	135.300	135.300	134.610	135.070	-0.230	down	down	down	down
14	135.080	135.230	134.760	135.220	0.140	up	up	up	up
15	135.210	135.750	135.010	135.640	0.430	up	up	up	up
16	135.640	137.280	135.530	136.740	1.100	up	up	up	up
17	136.710	137.150	136.180	136.190	-0.520	down	down	down	down
18	136.230	136.580	136.080	136.450	0.220	up	up	up	up
19	136.350	136.930	136.170	136.510	0.160	up	up	up	up