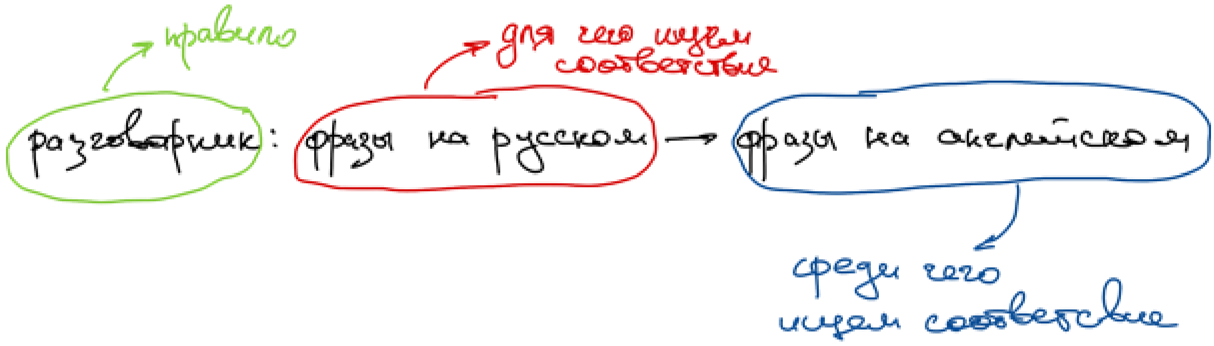
Урок 2. Понятие функции. Построение графиков функций с помощью SymPy

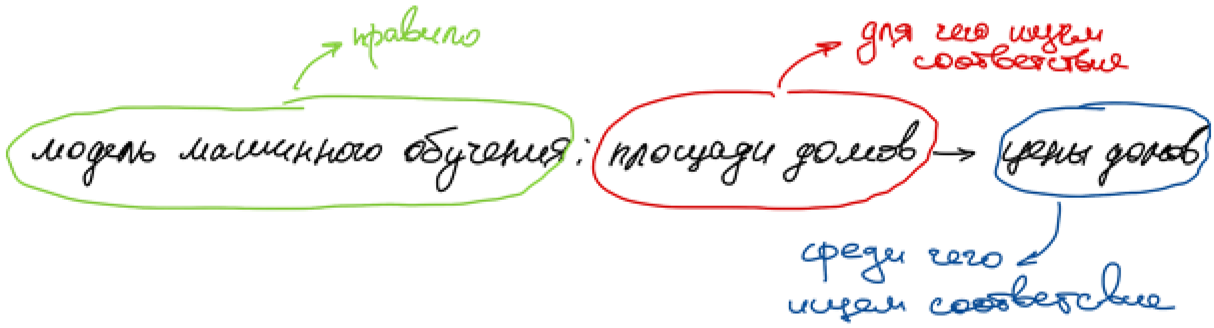
Понятие функции

Итак, начнём разбираться с тем, что такое функция. Это такое правило, обычно записанное в виде формулы, но не обязательно, которое каждому объекту одного множества ставит в соответствие объект из другого множества. Простейший пример функции — это обычный англо-русский разговорник. Каждой фразе ставится в соответствие её перевод. Давайте запишем это вот так:

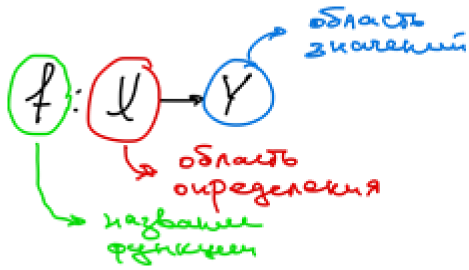


Помните, в нашем примере из первого модуля мы хотели предсказывать цены домов по их площади? С математической точки зрения, мы искали функцию, которая ставила бы в соответствие площади дома его цену. Найденная функция, она называется решающей, — и есть обученная модель. Поиск такой функции — это и есть процесс обучения.

То есть если использовать запись из предыдущего примера, это будет выглядеть так:



Давайте обобщим:



В начале идёт название функции (f), затем большая буква X — она обозначает множество, объектам которого что-то ставится в соответствие, — в нашем примере это были площади домов. Это множество называется областью определения функции. Область определения решающей функции — это множество объектов, для которых мы хотим что-то предсказывать. Справа стоит Y — это множество, объекты которого сопоставляются объектам множества X. В нашем примере это были цены. Это множество называется областью значений функции. Область значений решающей функции — это то, что мы хотим предсказать для наших объектов. Стрелка из X в Y обозначает, откуда куда функция работает.

Тогда для нашего примера с домами всё будет выглядеть так:

функция

Область определения

модель машинного обучения: площади домов → цены домов

Область значений

Модель машинного обучения — это наша функция, она играет роль f(x). Множество площадей домов — это наша область определения, то есть икс большое (X), а множество всевозможных цен домов — это наша область значений, то есть то, что играет роль игрек большого (Y).

Таким образом, в переводе с математического эта запись:

область значений

$$f: X \to Y$$

область определения

название функции

звучит так: «Есть функция f, которая ставит в соответствие объектам множества X объекты множества Y». То есть наша модель будет находить для каждой площади соответствующую цену. Важно отметить, что далее мы будем иметь дело только с числовыми функциями, то есть такими, что и объекты области определения (множества X) и объекты множества значений (множества Y) являются числами. Объекты множества X мы будем называть аргументами функции и обозначать x (маленькое), а объекты множества Y — значениями функции и обозначать y (маленькое).

Итак, общие вещи мы обсудили, давайте рассмотрим пример функции, заданной с помощью формулы. Например, вот так:

$$f(x) = x + 1$$

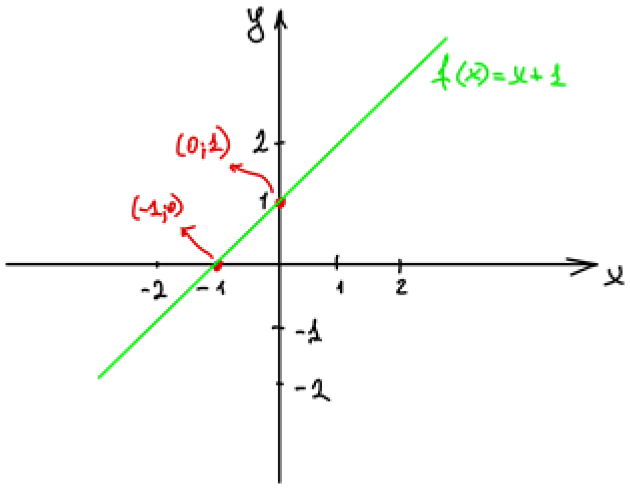
Что есть в этой записи? Давайте разбираться по порядку. В начале идёт f(x). Это, конечно, не умножение f на скобку, в которой стоит x, это целостная запись. Она означает, что есть функция f, которая зависит от аргумента x. В правой части формулы находится выражение, зависящее от x. Чтобы найти значение функции f (то есть игрек) для конкретного значения x, нужно подставить это значение вместо x в это выражение. Например, если мы хотим найти значение игрек при x = 10, то мы сделаем следующее:

$$f(10) = 10 + 1 = 11$$

Мы подставили 10 вместо x и нашли значение функции f, соответствующее значению аргумента 10, оказалось, что y = 11.

Кратко вспомним, что представляет собой график функции. В числовом виде функции удобны для работы с помощью компьютера, но для того, чтобы мы могли увидеть какие-то особенности глазами, придумали рисовать графики. График — это множество точек на плоскости или в пространстве, первой координатой которых является значение аргумента, а второй — значение функции при таком значении аргумента. То есть подставляя разные значения x в функцию мы получаем пары «аргумент — значение», которые затем отмечаем на графике.

Построим график для нашей функции f(x) = x + 1. Графиком этой функции является прямая. Для прямой достаточно рассчитать только две точки, чтобы её построить. Сначала рассчитаем точку с x = 0, потому что с 0 проще всего работать. f(0) = 0 + 1 = 1, значит первая точка будет (0; 1). Для второй точки возьмём x = -1, тогда f(-1) = -1 + 1 = 0, значит вторая точка (-1, 0). Начертим график:



Итак, в машинном обучении мы ищем функции, и чтобы их искать, нужно знать, какими свойствами они обладают. Математики придумали ряд информативных свойств, по наличию или отсутствию которых функции часто делятся на виды. Вместе с этим были придуманы методы определения, обладает функция тем или иным свойством или нет, которые мы будем изучать в рамках курса.

Давайте разбираться, чем характеризуются функции и зачем это нужно.

Построение графиков функций с помощью SymPy

Помогать нам в этом непростом деле будет знакомая нам с предыдущих модулей библиотека SymPy, с помощью которой можно сильно упростить себе жизнь при работе с математическими выражениями и функциями. В этом уроке мы научимся создавать функции с помощью SymPy, а также строить их графики. Таким образом мы избежим необходимости рисовать эскизы графиков с помощью карандаша и линейки. Хвала прогрессу!

В начале работы необходимо импортировать все функции SymPy и вызвать команду для красивой отрисовки формул, точно так же, как в первом модуле. Сейчас мы добавим к этим двум строкам ещё одну, она нужна для построения графиков:

In [7]:
```
from sympy import *
from sympy.plotting import plot
```

In [2]:
```
init_printing(use_unicode=False, wrap_line=False, no_global=True)
```

Создадим символьную переменную x, которая будет служить нам аргументом функции:

In [3]:
```
x = Symbol('x')
```

Теперь мы можем создать математическую функцию. Это делается точно так же, как и создание выражения, которое мы делали в первом модуле. Дело в том, что на самом деле мы и создавали тогда функции, определявшиеся этими выражениями, только ещё не знали об этом.

Давайте создадим функцию f(x) = x + 1, которая служила нам примером в этом уроке:

```
In [4]:  f = x + 1
```

Всё! Вот так просто, словно в тетрадке написали. Напечатаем нашу функцию:

```
In [18]:  f
Out[18]:  x + 1
```

Отлично, всё верно: x + 1. Теперь мы можем находить значения этой функции, вводя разные значения аргумента. Это делается с помощью функции sub, которая принимает два аргумента: название переменной и значение этой переменной, от которого мы хотим найти значение функции. Найдём значение при x = 0, что мы уже делали вручную:

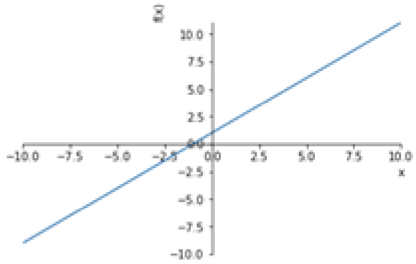
```
In [5]:  f.subs(x, 0)
Out[5]:  1
```

Замечательно, результат совпал нашим. Проверим так же значение при x = -1:

```
In [6]:  f.subs(x, -1)
Out[6]:  0
```

Тоже совпал. Теперь давайте построим график этой функции. Это делается очень просто, с помощью команды plot. В качестве аргумента передаётся функция:

```
In [23]:  plot(f)
```



Прекрасно. Как видите, всё очень просто и удобно. Мы будем активно пользоваться этим функционалом в следующих уроках модуля.

Практика

Теперь к практике. Импортируйте библиотеку и модуль plot, вызовите команду для красивой отрисовки формул. Затем, создав переменную, создайте функцию f(x) = -3x + 5 и постройте её график. Найдите значения этой функции для значений $x = \frac{5}{3}$ и x=2.

Проделайте то же самое для следующих функций и их точек. С некоторыми из них мы встретимся в будущем и разберём более подробно, но уже сейчас можно посмотреть на их графики:

1. $f(x) = 3x^2 + 3x - 6$. Найдите значения функции при x=0.5 и x=7.
2. $f(x) = 2x^4 + x^3 - 6x^2 + 7x + 1$. Найдите значения функции при x=3 и x=–5.
3. $f(x) = (x^2)^x$. Найдите значения функции при x=0, x=10.