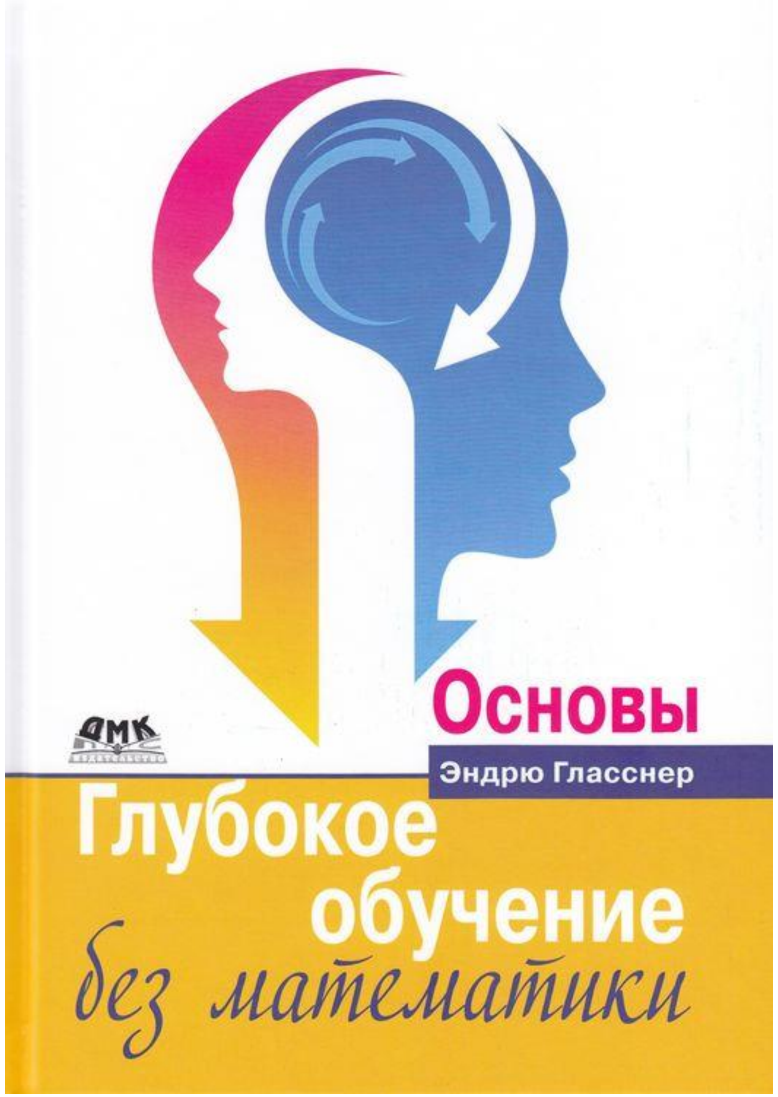
Линейная алгебра

Модуль 2


Основы
Эндрю Гласснер
Глубокое обучение
без математики


VALUES
CAREER
SUCCESS
SOLUTION
STRATEGY
SKILLS
GOALS
Практика
Эндрю Гласснер
Глубокое обучение
без математики

Задачи машинного обучения

OLS - регрессия

- Регрессия – задача машинного обучения с учителем

- В задаче регрессии по набору признаков предсказывается значение целевой переменной

- Целевая переменная - любое число

- OLS – один из методов

Простая и множественная регрессия

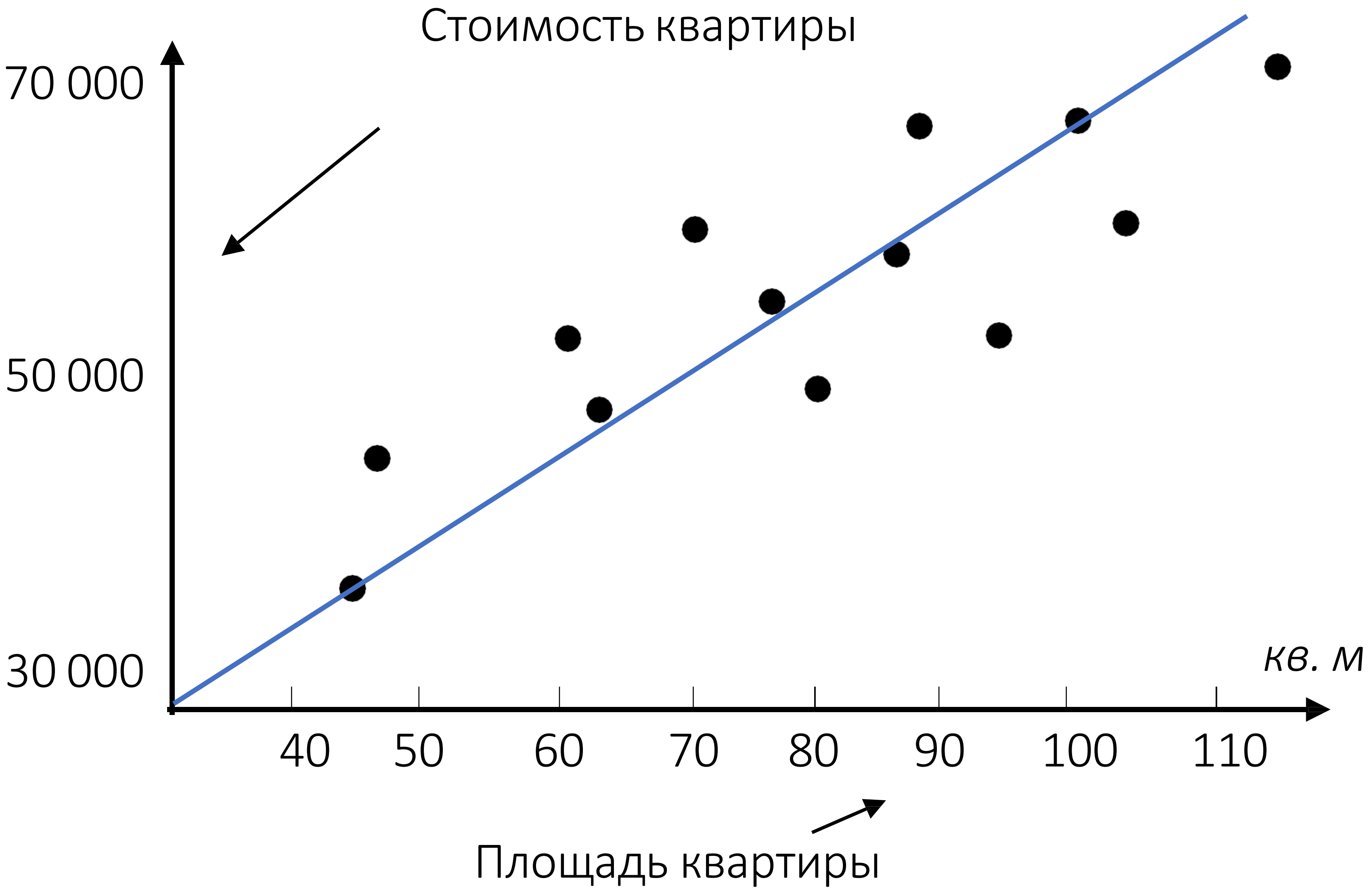
$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_k x_k$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{pmatrix}, \quad \begin{pmatrix} x_{11} \\ x_{12} \\ \cdots \\ x_{1N} \end{pmatrix}, \ldots, \begin{pmatrix} x_{k1} \\ x_{k2} \\ \cdots \\ x_{kN} \end{pmatrix}$$

Регрессия

Квартира	Цена, у.е.	Площадь, м
1	20	46
2	21	51
3	22	56
4	23	60
5	32	79
6	36	73
7	39	83
8	41	79
9	43	96
10	45	96

Регрессия


Стоимость квартиры
70 000
50 000
30 000
кв. м
40
50
60
70
80
90
100
110
Площадь квартиры

Регрессия

Коэффициенты, которые мы хотим найти

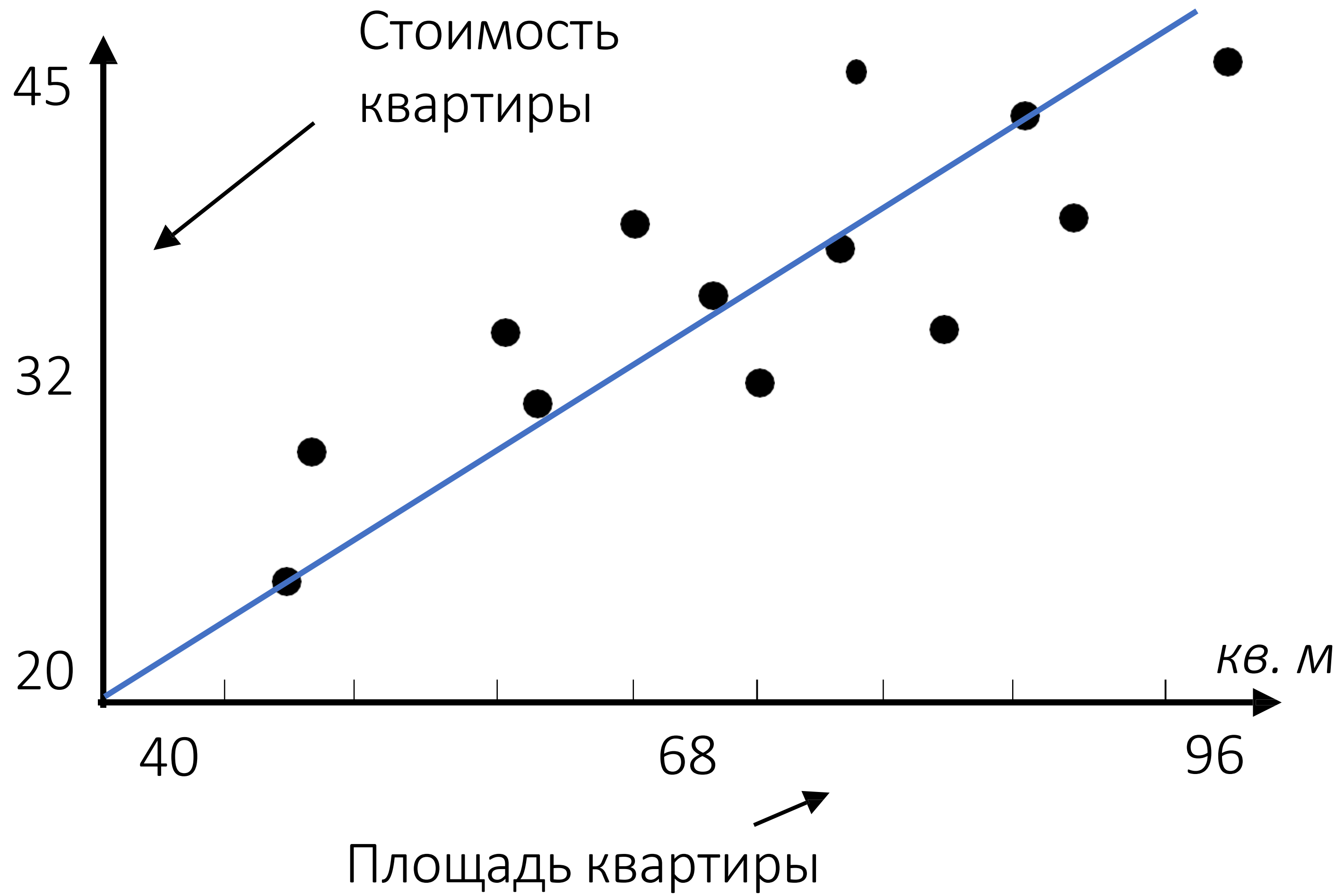
$$\boxed{\textit{Цена} = A \times \textit{Площадь} + B}$$

Целевая переменная

Признак объекта

Регрессия

Цена = 1.7368 x Площадь + 15.974



Оценка качества модели: OLS (МНК)

Вывод OLS

$$Y_i = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_k X_{ki} + \varepsilon_i, i = 1, \ldots, n$$

Обозначим

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \ldots \\ Y_n \end{pmatrix}, X_1 = \begin{pmatrix} X_{11} \\ X_{12} \\ \ldots \\ X_{1n} \end{pmatrix}, \ldots, X_k = \begin{pmatrix} X_{k1} \\ X_{k2} \\ \ldots \\ X_{kn} \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \ldots \\ \varepsilon_n \end{pmatrix}$$

Тогда уравнение регрессии можно переписать в векторном виде

$$Y = \beta_0 + \beta_1 X_2 + \ldots + \beta_k X_k + \varepsilon$$

Вывод OLS

$$Y_i = \beta_0 + \beta_1 X_{2i} + \ldots + \beta_k X_{ki} + \varepsilon_i, \; i = 1, \ldots, n$$

Если ввести матрицу наблюдений X размера (nxk) и вектор коэффициентов β размера (kx1)

$$X = \begin{pmatrix} 1 & X_{11} & \ldots & X_{k1} \\ 1 & X_{12} & \ldots & X_{k2} \\ \ldots & \ldots & \ldots & \ldots \\ 1 & X_{1n} & \ldots & X_{kn} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \ldots \\ \beta_k \end{pmatrix},$$

то уравнение регрессии можно переписать в матричном виде:

$$Y = X\beta + \varepsilon$$

Вывод OLS

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon,$$

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \ldots + \hat{\beta}_k X_{ki}, i = 1, \ldots, n$$

$$e_i = Y_i - \hat{Y}_i, i = 1, \ldots, n$$

Вывод OLS

$$RSS = \sum_{i=1}^{n} e_i^2 \to \min$$

$$Y = X\beta + \varepsilon$$

$$\hat{Y} = X\hat{\beta}$$

$$e = Y - \hat{Y} = Y - X\hat{\beta}$$

Вывод OLS

$$e = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

$$\sum_{i=1}^{n} e_i^2 = (e_1, \dots, e_n) \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} = e'e$$

Вывод OLS

$$RSS(\hat{\beta}) = e'e = (Y - X\hat{\beta})'(Y - X\hat{\beta}) =$$

$$= (Y' - \hat{\beta}'X')(Y - X\beta) =$$

$$= Y'Y - \hat{\beta}'X'Y - Y'X\hat{\beta} + \hat{\beta}'X'X\hat{\beta}$$

$$\underset{(1\times n)}{Y'}\ \underset{(n\times k)}{X}\ \underset{(k\times 1)}{\hat{\beta}} \quad \Rightarrow (Y'X\hat{\beta}) = (Y'X\hat{\beta})' = \hat{\beta}'X'Y$$

$$RSS(\hat{\beta}) = Y'Y - 2\hat{\beta}'X'Y + \hat{\beta}'X'X\hat{\beta}$$

Вывод OLS

$$RSS(\hat{\beta}) = Y'Y - 2\hat{\beta}'X'Y + \hat{\beta}'X'X\hat{\beta}$$

$$\frac{\partial RSS(\hat{\beta})}{\partial \hat{\beta}} = -2X'Y + 2X'X\hat{\beta} = 0$$

$$X'X\hat{\beta} = X'Y$$

$$\boxed{\hat{\beta} = (X'X)^{-1}X'Y}$$

Вывод OLS: геометрическая интерпретация

$$Y = \hat{Y} + e \Leftrightarrow |e| \to \min \Leftrightarrow |e|^2 \to \min$$

$$|e| \to \min \Rightarrow e \perp \Omega$$

$$\Rightarrow e \perp X_j \ \forall \ j = 1, \ldots, k \Rightarrow$$

$$X'_j e = 0$$

Вывод OLS: геометрическая интерпретация

$$X'e = 0$$

$$X'(Y - \hat{Y}) = 0$$

$$X'(Y - X\hat{\beta}) = 0$$

$$X'Y - X'X\hat{\beta} = 0$$

$$X'X\hat{\beta} = X'Y$$

$$\hat{\beta}_{OLS} = (X'X)^{-1}X'Y$$

Теорема Гаусса-Маркова

Если модель множественной линейной регрессии

$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon$,

1) Правильно специфицирована

2) Не существует линейной связи между регрессорами

3) Возмущения имеют нулевое мат. ожидание $E(\varepsilon_i) = 0$,

4) Дисперсии возмущений одинаковы $D(\varepsilon_i) = \sigma_\varepsilon^{\,2}$,

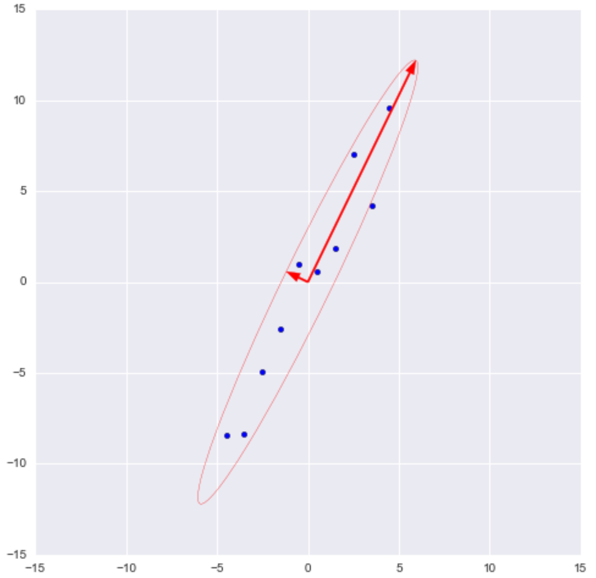
5) Возмущения с разными номерами не коррелируют

$Cov(\varepsilon_i, \varepsilon_j) = 0$

Тогда оценки МНК являются BLUE (Best Linear Unbiased Estimator).

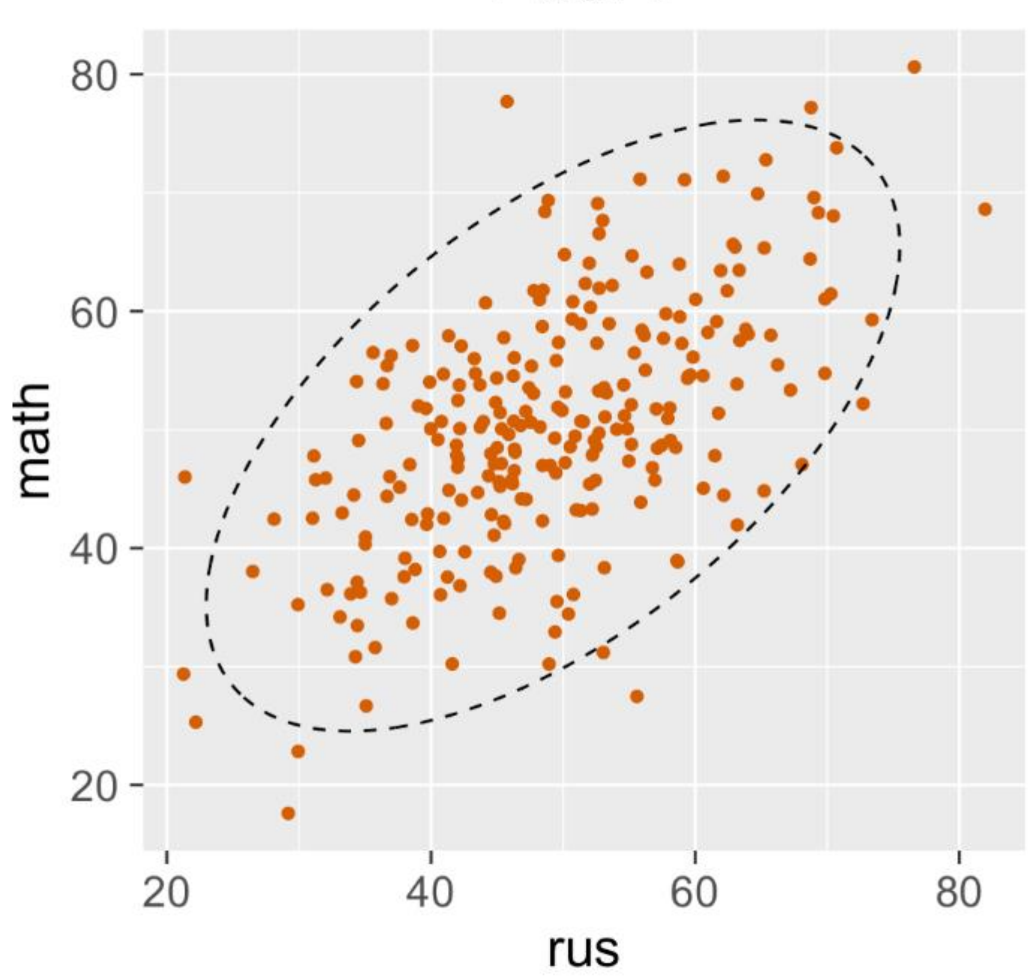
PCA – идея метода

- Новое признаковое пространство
- Новые оси координат (главные компоненты) ортогональны
- Значения новых признаков – линейная комбинация предыдущих
- Необходимо сохранить как можно больше изменчивости (дисперсии) исходных данных



PC: пример на игрушечной выборке

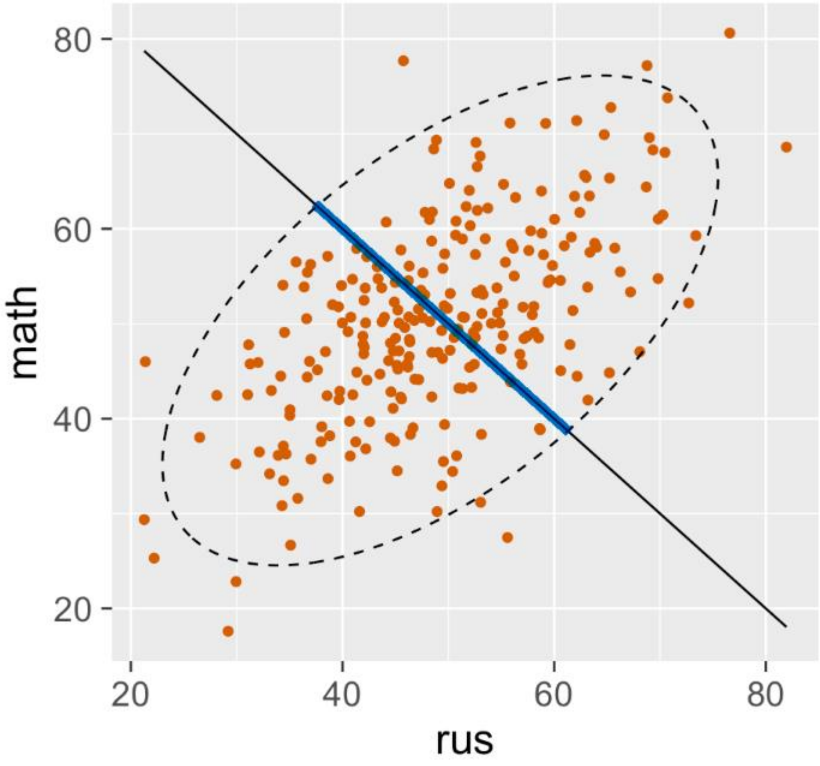
```
##          rus     math
## 1 38.62011 33.67848
## 2 46.22913 54.53733
## 3 46.40963 38.32976
## 4 53.17011 51.07601
## 5 62.86754 65.64322
```



PC: пример на игрушечной выборке

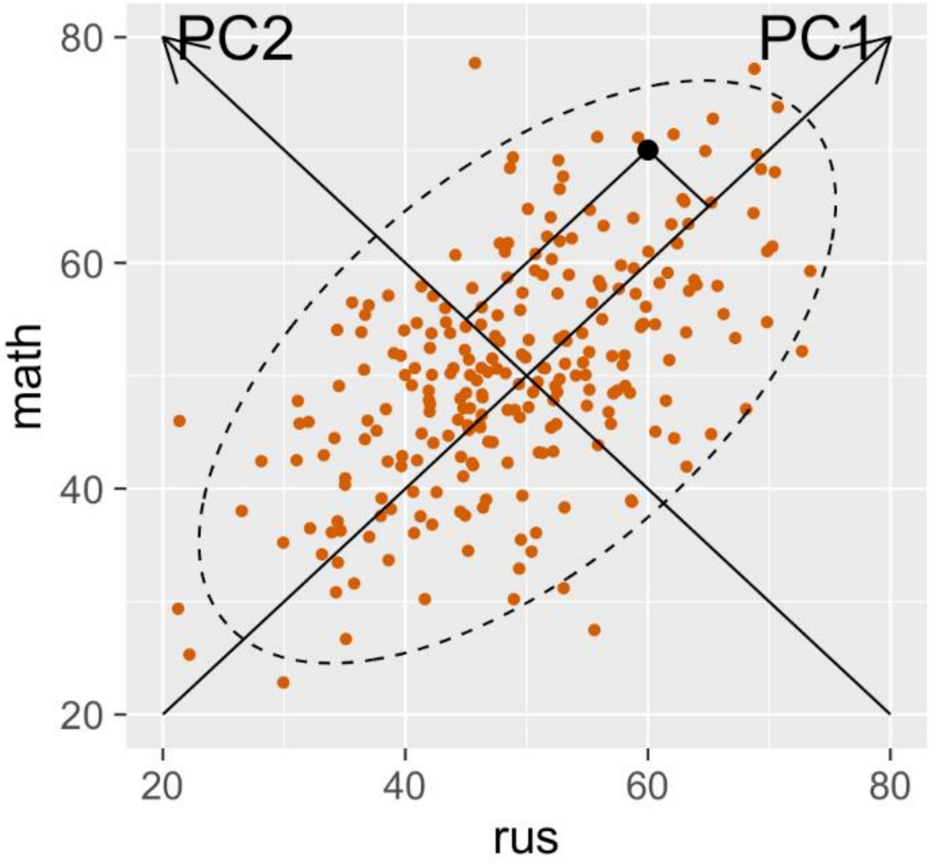
Как закодировать два числа в одном?

$$PC_1 = rus + math$$

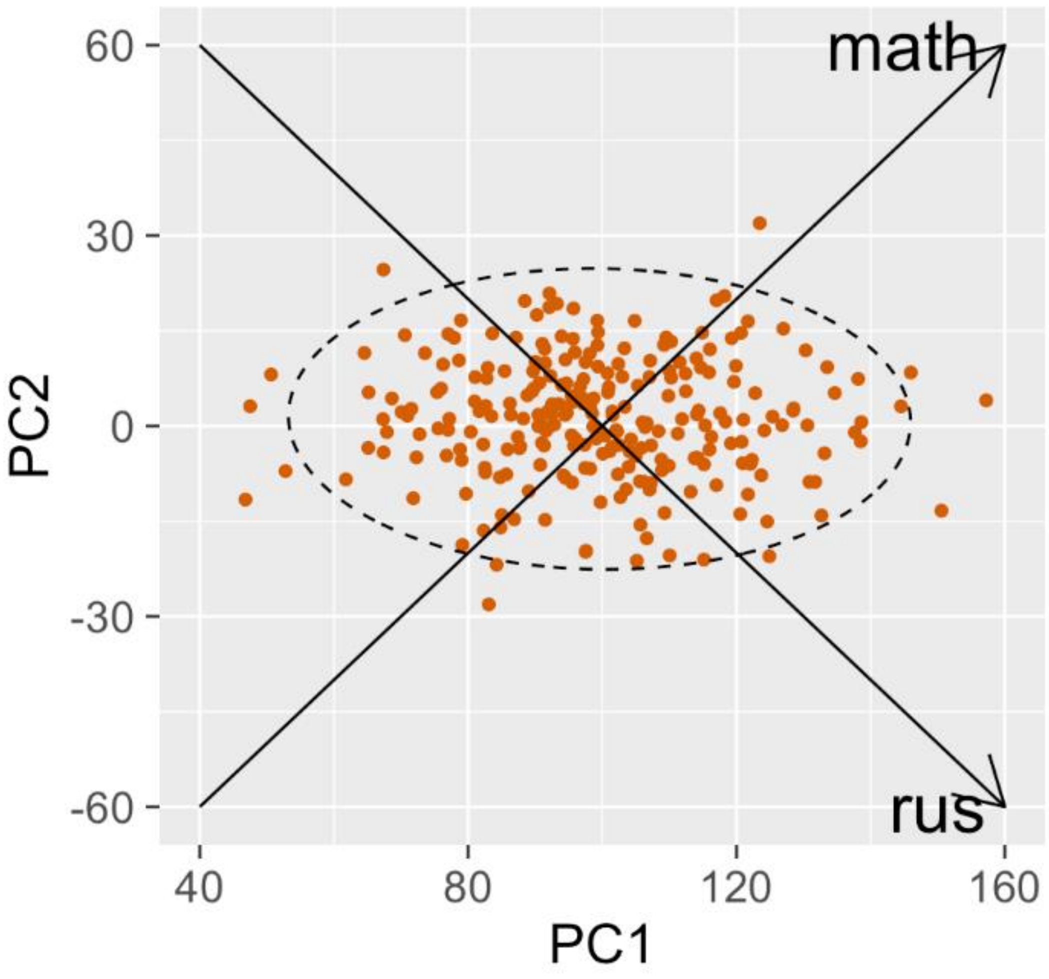


PC: пример на игрушечной выборке

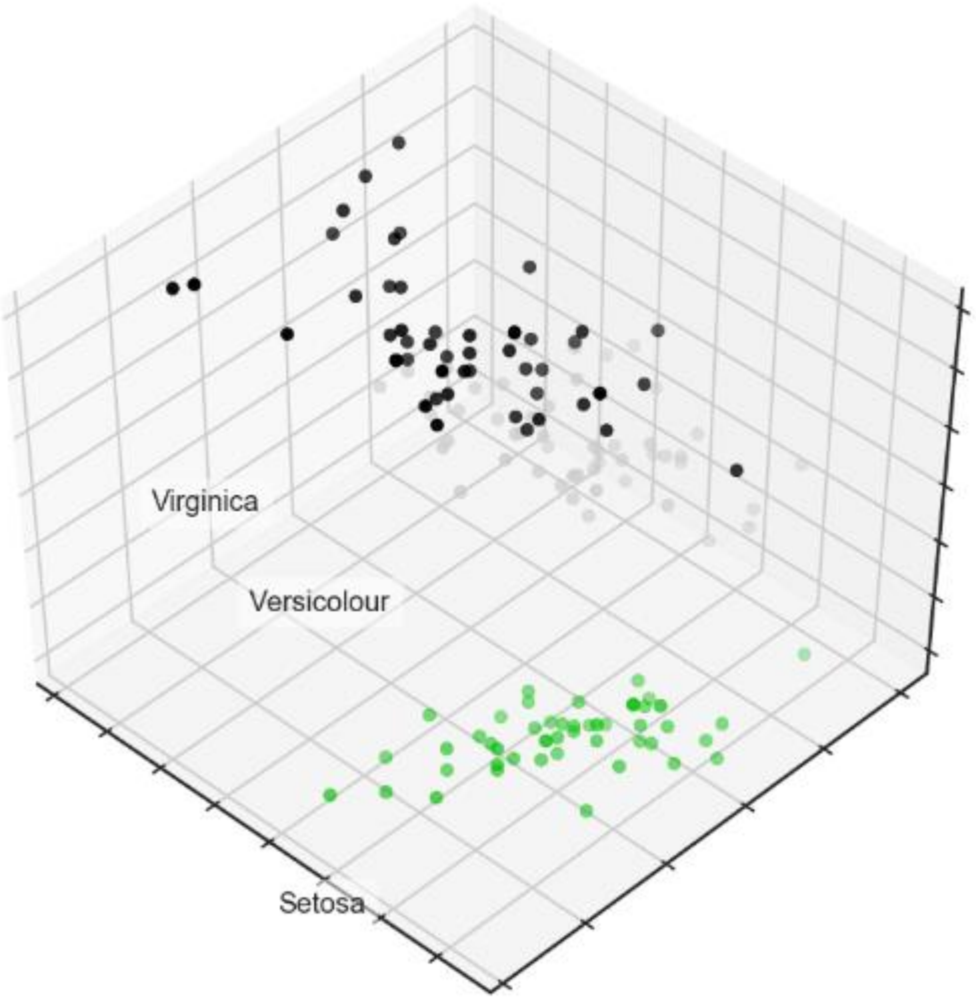
Идея: новая система координат!



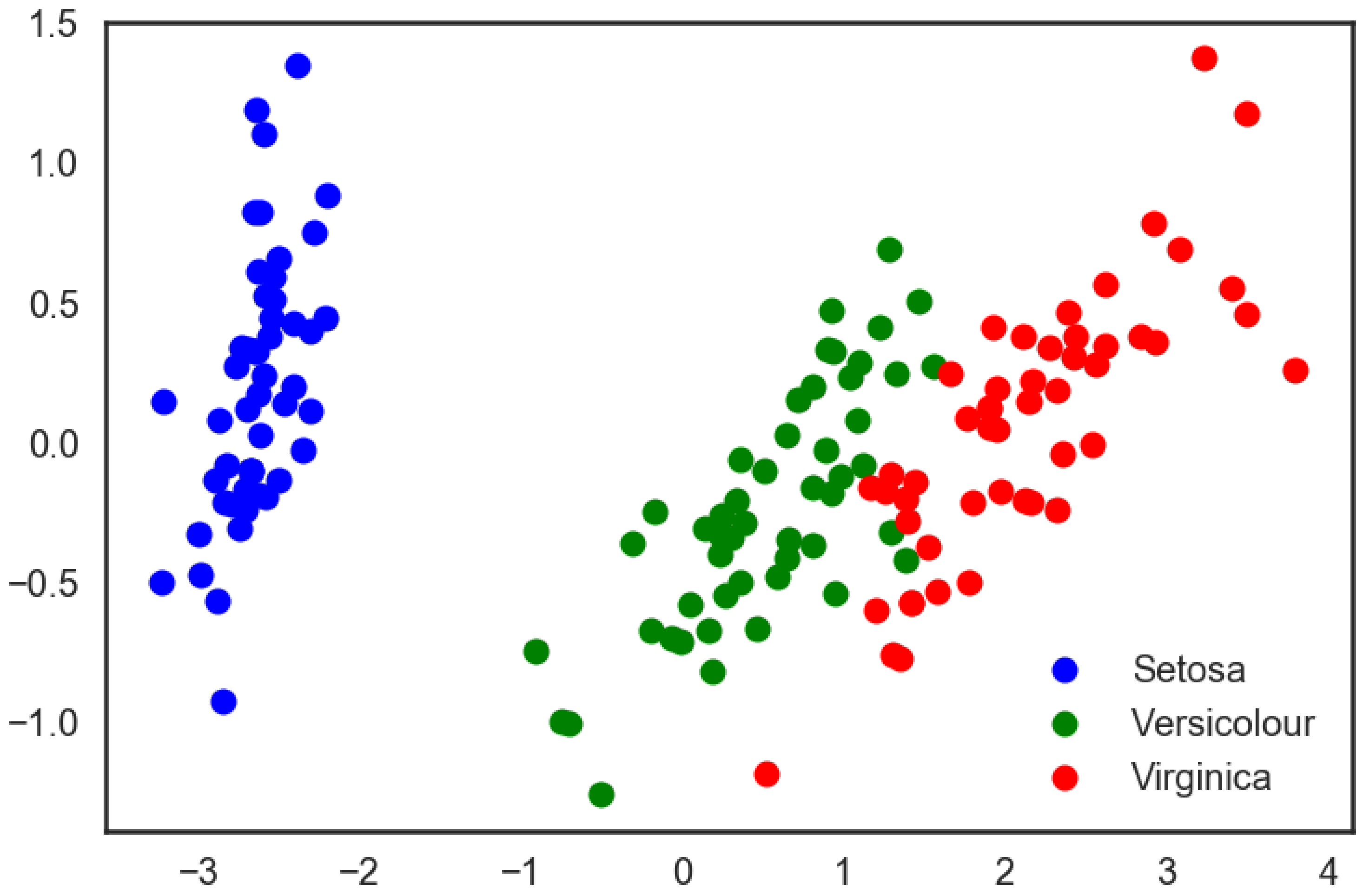
PC: пример на игрушечной выборке



PC: пример на цветках ириса

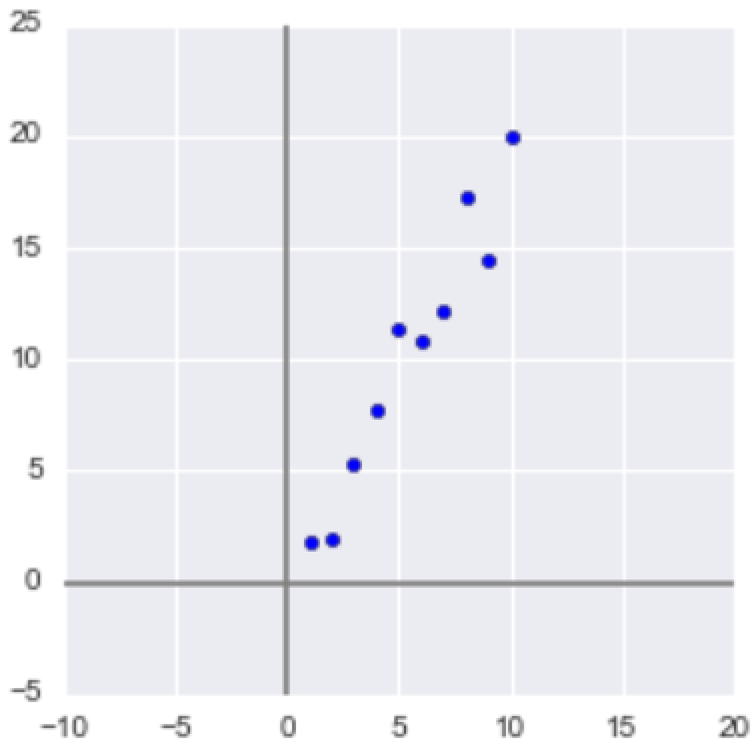


```
pca = decomposition.PCA(n_components=2)
X_centered = X - X.mean(axis=0)
pca.fit(X_centered)
X_pca = pca.transform(X_centered)
```



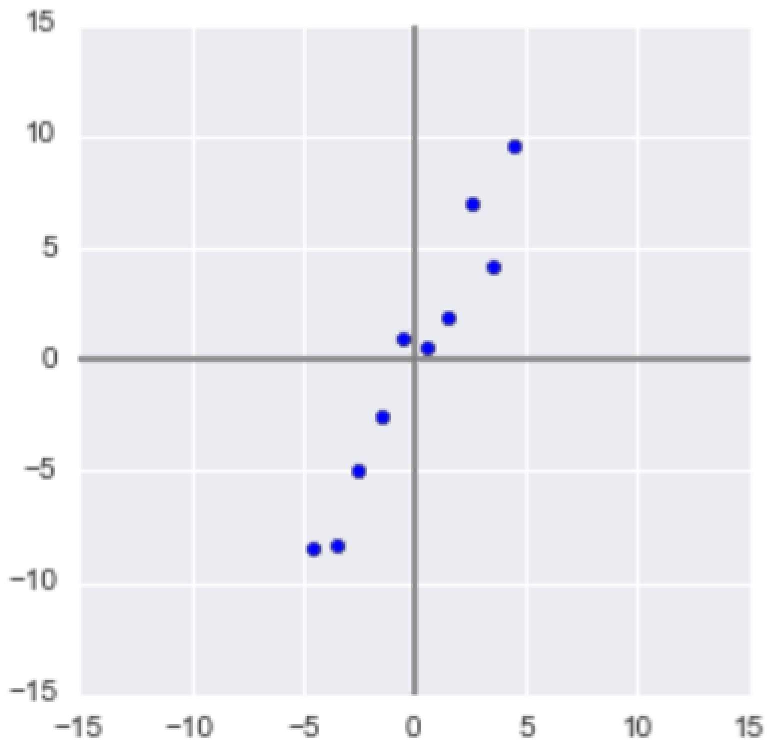
PC: пример на игрушечной выборке

```
x = np.arange(1,11)
y = 2 * x + np.random.randn(10)*2
X = np.vstack((x,y))
print X

OUT:
[[  1.            2.            3.            4.            5.
    6.            7.            8.            9.           10.
]
 [  2.73446908   4.35122722   7.21132988  11.24872601   9.
58103444
  12.09865079  13.78706794  13.85301221  15.29003911  18.0
998018 ]]
```

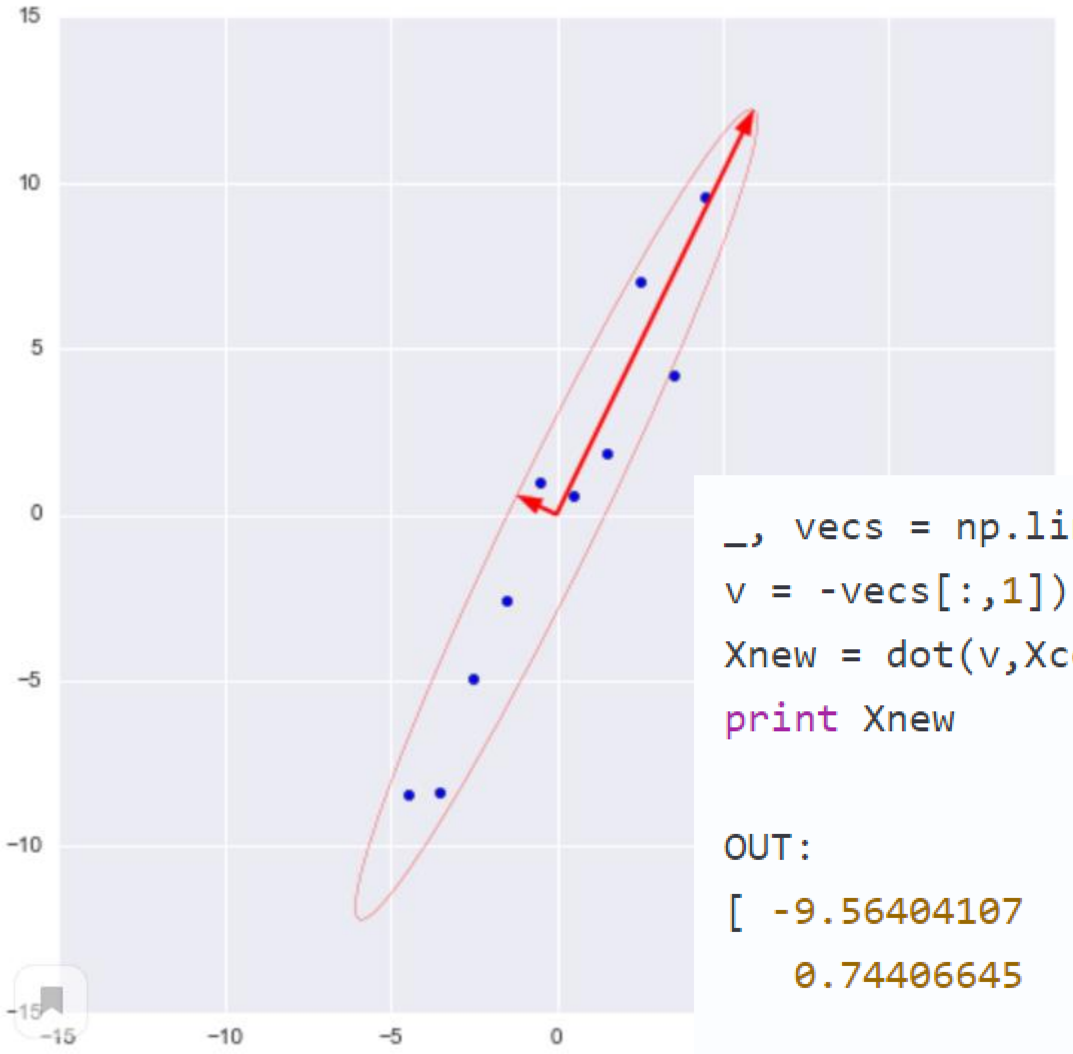


PC: пример на игрушечной выборке

```
Xcentered = (X[0] - x.mean(), X[1] - y.mean())
m = (x.mean(), y.mean())
print Xcentered
print "Mean vector: ", m

OUT:
(array([-4.5, -3.5, -2.5, -1.5, -0.5,  0.5,  1.5,  2.5,
3.5,  4.5]),
 array([-8.44644233, -8.32845585, -4.93314426, -2.5672313
6,  1.01013247,
         0.58413394,  1.86599939,  7.00558491,  4.2144064
7,  9.59501658]))
Mean vector:  (5.5, 10.314393916)
```



PC: пример на игрушечной выборке




```
_, vecs = np.linalg.eig(covmat)
v = -vecs[:,1])
Xnew = dot(v,Xcentered)
print Xnew

OUT:
[ -9.56404107  -9.02021624  -5.52974822  -2.96481262   0.68933859
   0.74406645   2.33433492   7.39307974   5.3212742   10.59672425]
```