Klasyczny model regresji liniowej

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,m} \\ x_{2,1} & & x_{2,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_m \end{bmatrix}$$

$$Y = XB + \varepsilon$$

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

$$e = Y - \hat{Y} = Y - X\hat{B}$$ *(błąd/reszty)*

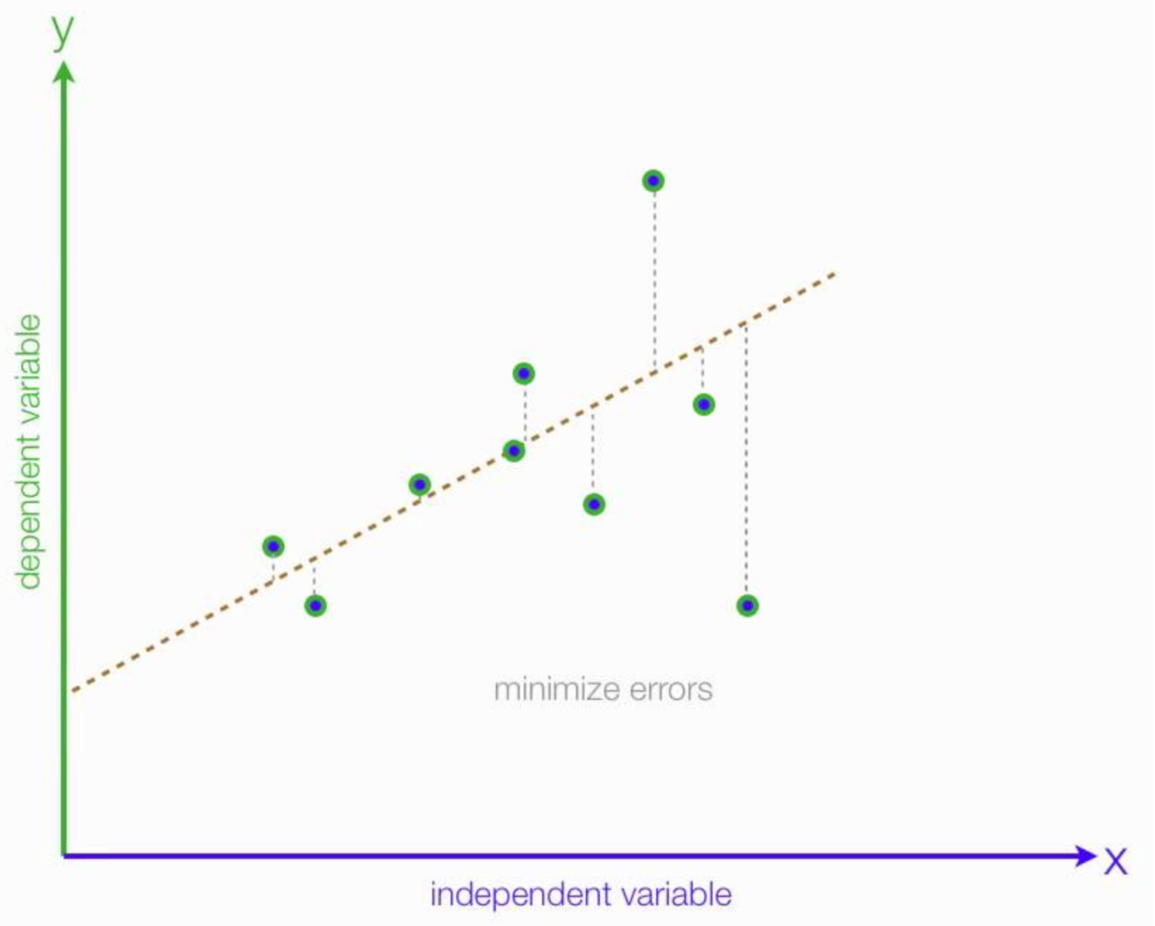
Metoda najmniejszych kwadratów (*ordinary least square/OLS*):

$$RSS = e^T e = \left(Y - X\hat{B}\right)^T \left(Y - X\hat{B}\right) = \left(Y^T - \hat{B}^T Y^T\right)\left(Y - X\hat{B}\right)$$

$$RSS = Y^T Y - 2\hat{B}^T X^T Y + \hat{B}^T X^T X\hat{B} \underset{B}{\rightarrow} \quad min$$

Równanie normalne:

$$\boldsymbol{\hat{B} = \left(X^T X\right)^{-1} X^T Y}$$



Klasyczny model regresji liniowej

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

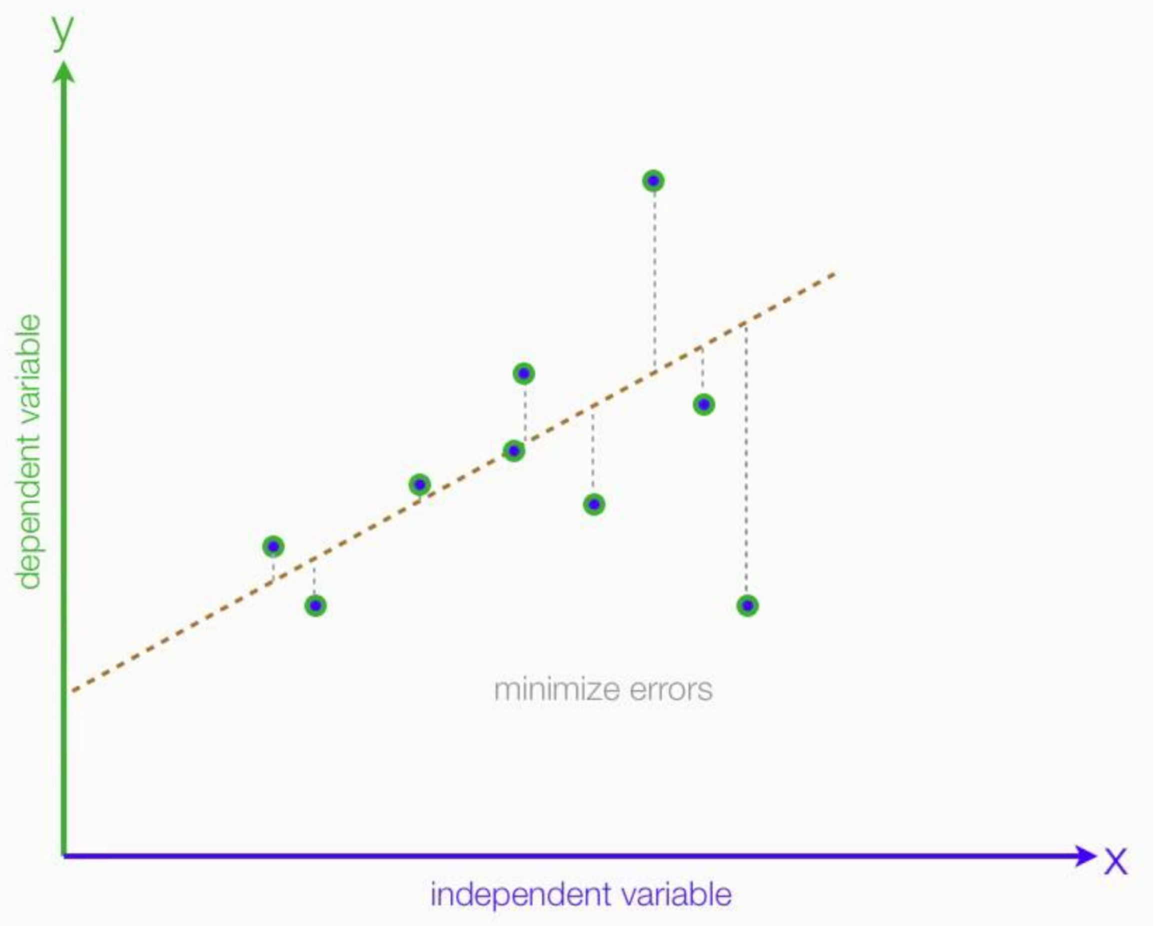
$$e = Y - \hat{Y} = Y - X\hat{B}$$

$$RSS = e^T e = \left(Y - X\hat{B}\right)^T \left(Y - X\hat{B}\right) = \left(Y^T - \hat{B}^T Y^T\right)\left(Y - X\hat{B}\right)$$

$$RSS = [e_1 \quad e_2 \quad \dots \quad e_n] \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = e_1^2 + e_2^2 + e_3^2 + \dots + e_n^2$$

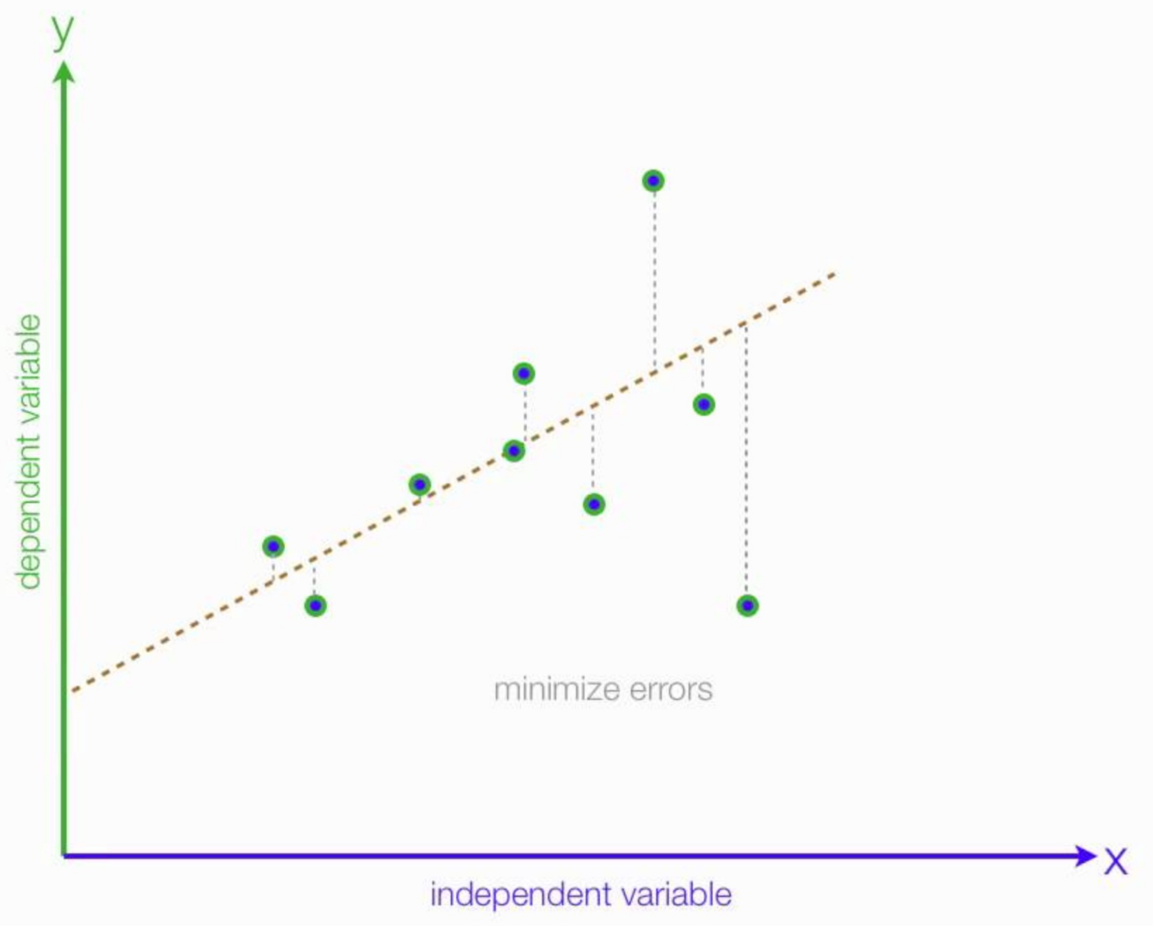
$$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - \boldsymbol{x_i}\hat{\boldsymbol{b}}\right)^2$$

$$e = \begin{bmatrix} y_1 - \widehat{y_1} \\ y_2 - \widehat{y_2} \\ \vdots \\ y_n - \widehat{y_n} \end{bmatrix} = \begin{bmatrix} y_1 - \widehat{b_1} x_1 + b_0 \\ y_2 - \widehat{b_1} x_2 + b_0 \\ \vdots \\ y_n - \widehat{b_1} x_n + b_0 \end{bmatrix}$$



Klasyczny model regresji liniowej

1. Liniowość - zmienne losowe Y_i, $X_{i,k}$ należą do L^2 i spełniają:
$$Y = XB + \varepsilon$$
2. Ścisła egzogeniczność:
$$\mathbb{E}(\varepsilon|X) = 0$$
3. Liniowa niezależność obserwacji.
4. Sferyczność błędu, czyli:
    1. Homoskedastyczność (stałość wariancji):
    $$\mathbb{E}(\varepsilon^2|X) = \sigma^2$$
    2. Brak korelacji reszt
    $$\mathbb{E}\left(\varepsilon_i \varepsilon_j \middle| X\right) = 0$$
5. Gaussowskość błędu:
$$\varepsilon|X \sim \mathcal{N}$$



Twierdzenie Gaussa-Markova:
Jeśli zachodzą 1 – 4 to estymator najmniejszych kwadratów:

$$\boldsymbol{B} = \left(\boldsymbol{X^T X}\right)^{-1} \boldsymbol{X^T Y}$$

jest:
najlepszym nieobciążonym liniowym estymatorem.
(*BLUE*: best linear unbiased estimator *)*

https://mst.mimuw.edu.pl/wyklady/ekn/wyklad.pdf
http://coin.wne.uw.edu.pl/pstrawinski/notatki/kmrl.pdf

Klasyczny model regresji liniowej

Ocena jakości modelu R^2; $adjusted\ R^2$

$$R^2 := \frac{\sum_{i=1}^{n} (\hat{y}_i - \overline{y})^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2}$$

Współczynnik determinacji, inaczej zwany współczynnikiem określoności lub R-kwadrat jest miarą tego, jaki procent zmienności zmiennej zależnej (objaśnianej) jest wyjaśniany za pomocą zmiennej niezależnej (czynnik zmienna objaśniająca, predyktor) bądź modelu statystycznego. Innymi słowy, współczynnik determinacji informuje nas, ile nasz model, nasz badany czynnik wyjaśnia zgromadzone dane pomiarowe (zmienną zależną).

Z R^2 (R-kwadrat) wiąże się z nieodłącznym problemem – dodatkowe zmienne wejściowe sprawią, że R-kwadrat pozostanie taki sam lub wzrośnie (jest to spowodowane tym, jak R-kwadrat jest obliczany matematycznie). Dlatego nawet jeśli dodatkowe zmienne wejściowe nie wykazują związku ze zmiennymi wyjściowymi, R-kwadrat wzrośnie.

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-k-1}$$

n – ilość obserwacji
k – ilość zmiennych objaśniających (bez wyrazu wolnego)

Metoda Najmniejszych Kwadratów

Złożoność obliczeniowa

Równanie normalne:

$$\boldsymbol{\widehat{B} = \left(X^T X\right)^{-1} X^T Y}$$

Najbardziej złożone obliczeniowo jest znalezienie macierzy odwrotnej:

$$\boldsymbol{W^{-1} = \left(X^T X\right)^{-1}}$$

Wyznaczanie:

Metoda dopełnień algebraicznych

$$W^1 = \frac{1}{detW}(W^D)^T$$

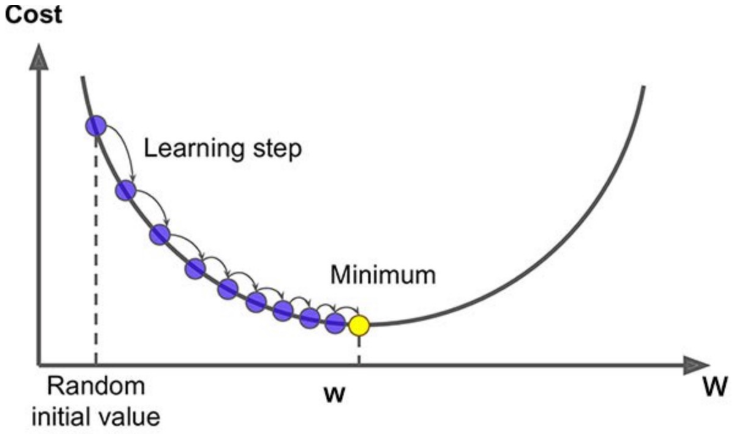
Metoda eliminacji Gaussa-Jordana

$$[W|I] \longmapsto [I|W^{-1}]$$

W zależności od implementacji algorytmu złożoność obliczeniowa odwrócenia macierzy wynosi zazwyczaj od około $\boldsymbol{O(n^{2.4})}$ do $\boldsymbol{O(n^3)}$.

Podwojenie liczby kolumn wydłuża czas obliczeń o około od $\boldsymbol{2^{2.4} = 5.3}$ do $\boldsymbol{2^3 = 8}$.

Gradient prosty *Gradient descent*



$$\theta_j := \theta_j - \eta \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\eta - learning\ rate\ (0.01)$$

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ x_{2,1} & & x_{2,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix}$$

$$\boldsymbol{x_i} = [x_{i,1} \quad x_{i,2} \quad \dots \quad x_{i,n}]$$

$$RSS(\theta) = e^T e = (Y - X\theta)^T (Y - X\theta) = (Y^T - \theta^T Y^T)(Y - X\theta)$$

$$RSS(\theta) = Y^T Y - 2\theta^T X^T Y + \theta^T X^T X \theta$$

$$\frac{\partial}{\partial \theta} RSS(\theta) = -2X^T Y + 2X^T X\theta = 2X^T (X\theta - Y)$$

$$\frac{1}{m} RSS(\theta) = J(\theta)$$

```
for iteration in range(n_iterations):
    gradients = 2/m * X.T.dot(X.dot(theta) - y)
    theta = theta - eta * gradients
```

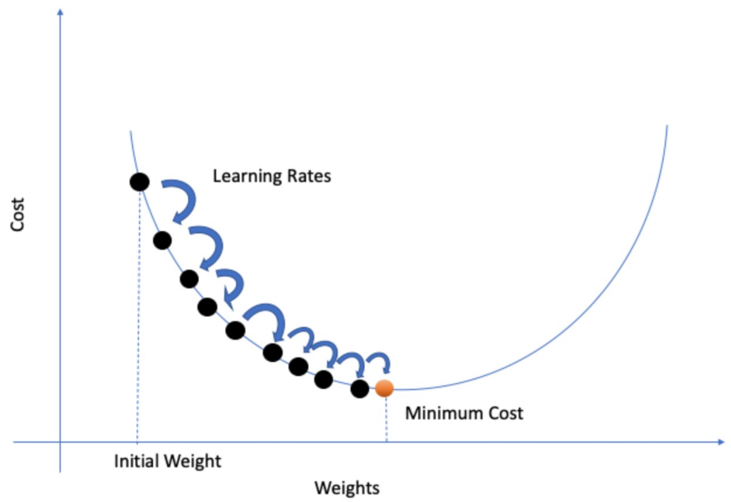
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

$$h_\theta(x_i) = \boldsymbol{\theta^T x_i}$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\boldsymbol{\theta^T x_i} - y_i\right)^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} 2\left(\boldsymbol{\theta^T x_i} - y_i\right) x_{i,j} = \frac{1}{m} \sum_{i=1}^{m} \left(\boldsymbol{\theta^T x_i} - y_i\right) x_{i,j}$$

Stochastyczny Gradient *Stochastic Gradient Descent*



$$\theta_j := \theta_j - \eta \frac{\partial}{\partial \theta_j} J_i(\theta_0, \theta_1)$$

$$\eta - learning\ rate\ (0.01)$$

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ x_{2,1} & & x_{2,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix}$$

$$\boldsymbol{x_i} = [x_{i,1} \quad x_{i,2} \quad \cdots \quad x_{i,n}]$$

$$i - ustalone, losowe$$

$$J_i(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

$$J_i(\theta_0, \theta_1) = (h_\theta(x_i) - y_i)^2$$

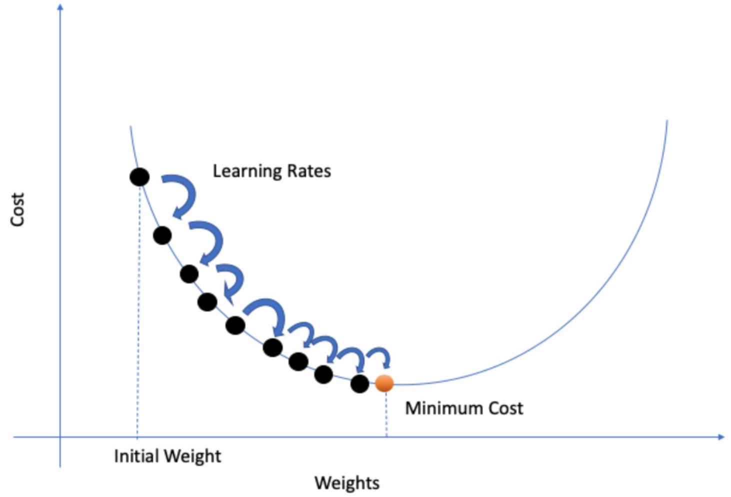
$$h_\theta(x_i) = \boldsymbol{\theta^T x_i}$$

$$J_i(\theta_0, \theta_1) = \left(\boldsymbol{\theta^T x_i} - y_i\right)^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = 2\left(\boldsymbol{\theta^T x_i} - y_i\right) x_{i,j}$$

```python
for epoch in range(n_epochs):
    for i in range(m):
        random_index = np.random.randint(m)
        xi = X[random_index:random_index+1]
        yi = y[random_index:random_index+1]
        gradients = 2 * xi.T.dot(xi.dot(theta) - yi)
```

Gradient z minigrupami *Mini-batch Gradient Descent*



$$\theta_j := \theta_j - \eta \frac{\partial}{\partial \theta_j} J_I(\theta_0, \theta_1)$$

$$\eta - learning\ rate\ (0.01)$$

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ x_{2,1} & & x_{2,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix}$$

$$\boldsymbol{x_i} = [x_{i,1} \quad x_{i,2} \quad \dots \quad x_{i,n}]$$

$$i - ustalone, losowe$$

$$J(\theta_0, \theta_1) = \frac{1}{2|I|} \sum_{i \in I} (h_\theta(x_i) - y_i)^2$$

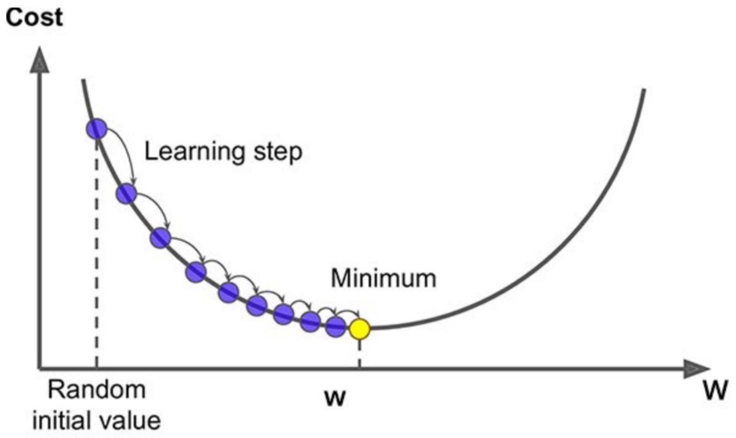
$$h_\theta(x_i) = \boldsymbol{\theta^T x_i}$$

$$J(\theta_0, \theta_1) = \frac{1}{2|I|} \sum_{i \in I} \left(\boldsymbol{\theta^T x_i} - y_i\right)^2$$

$$\frac{\partial}{\partial \theta_j} J_I(\theta_0, \theta_1) = \frac{1}{2|I|} \sum_{i \in I} 2\left(\boldsymbol{\theta^T x_i} - y_i\right) x_{i,j} = \frac{1}{2|I|} \sum_{i \in I} \left(\boldsymbol{\theta^T x_i} - y_i\right) x_{i,j}$$

Spadek po gradiencie

Gradient prosty
Gradient descent



$$\theta_j := \theta_j - \eta \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\eta - learning\ rate\ (0.01)$$

Obliczamy koszt i pochodną dla całego zbioru obserwacji x.

Stochastyczny gradient
Stochastic Gradient Descent

Obliczamy koszt i pochodną dla losowo wybranej obserwacji x.

Gradient z minigrupami
Mini-batch Gradient Descent

Obliczamy koszt i pochodną dla losowo wybranego podzbioru (batch) obserwacji x.

Zmienne kategoryczne/dyskretne *Dummy Variable*

One-Hot Encode (OHE)

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

.unique()

New York
California
Florida

One-Hot Encode (OHE)

State_California	State_Florida	State_New York
0	0	1
1	0	0
0	1	0

	R&D Spend	Administration	Marketing Spend	Profit	State_California	State_Florida	State_New York
0	165349.20	136897.80	471784.10	192261.83	0	0	1
1	162597.70	151377.59	443898.53	191792.06	1	0	0
2	153441.51	101145.55	407934.54	191050.39	0	1	0
3	144372.41	118671.85	383199.62	182901.99	0	0	1
4	142107.34	91391.77	366168.42	166187.94	0	1	0

Pułapka zmiennych kategorycznych *Dummy Variable Trap*

Dla modelu regresji

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

	R&D Spend	Administration	Marketing Spend	Profit	State_California	State_Florida	State_New York
0	165349.20	136897.80	471784.10	192261.83	0	0	1
1	162597.70	151377.59	443898.53	191792.06	1	0	0
2	153441.51	101145.55	407934.54	191050.39	0	1	0
3	144372.41	118671.85	383199.62	182901.99	0	0	1
4	142107.34	91391.77	366168.42	166187.94	0	1	0

R&D Spend	Administration	Marketing Spend	Profit	State_California	State_Florida	State_New York	intercept
165349.20	136897.80	471784.10	192261.83	0	0	1	1
162597.70	151377.59	443898.53	191792.06	1	0	0	1
153441.51	101145.55	407934.54	191050.39	0	1	0	1
144372.41	118671.85	383199.62	182901.99	0	0	1	1
142107.34	91391.77	366168.42	166187.94	0	1	0	1

$$B = \left(X^T X\right)^{-1} X^T Y$$

$$det\left(X^T X\right) = 0$$

$$\left(X^T X\right)^{-1} - nie\ istnieje$$

Zmienne kategoryczne/dyskretne

Inne metody

https://contrib.scikit-learn.org/category_encoders/

```
pip install category_encoders
```

```
import category_encoders as ce

encoder = ce.BackwardDifferenceEncoder(cols=[...])
encoder = ce.BaseNEncoder(cols=[...])
encoder = ce.BinaryEncoder(cols=[...])
encoder = ce.CatBoostEncoder(cols=[...])
encoder = ce.CountEncoder(cols=[...])
encoder = ce.GLMMEncoder(cols=[...])
encoder = ce.HashingEncoder(cols=[...])
encoder = ce.HelmertEncoder(cols=[...])
encoder = ce.JamesSteinEncoder(cols=[...])
encoder = ce.LeaveOneOutEncoder(cols=[...])
encoder = ce.MEstimateEncoder(cols=[...])
encoder = ce.OneHotEncoder(cols=[...])
encoder = ce.OrdinalEncoder(cols=[...])
encoder = ce.SumEncoder(cols=[...])
encoder = ce.PolynomialEncoder(cols=[...])
encoder = ce.TargetEncoder(cols=[...])
encoder = ce.WOEEncoder(cols=[...])
encoder = ce.QuantileEncoder(cols=[...])

encoder.fit(X, y)
X_cleaned = encoder.transform(X_dirty)
```

Regresyjne Lasy Losowe *(Bagging)*

Step 1: Bootstraping

Losujemy obserwacje ze zbioru danych aby stworzyć zbiór o tym samym rozmiarze.

Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Sugar
No	No	No	125	90.01
Yes	Yes	Yes	180	121.3
Yes	Yes	No	210	96.50
Yes	No	Yes	167	137.67

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Sugar
Yes	Yes	Yes	180	121.3
No	No	No	125	90.01
Yes	No	Yes	167	137.67
Yes	No	Yes	167	137.67

Regresyjne Lasy Losowe

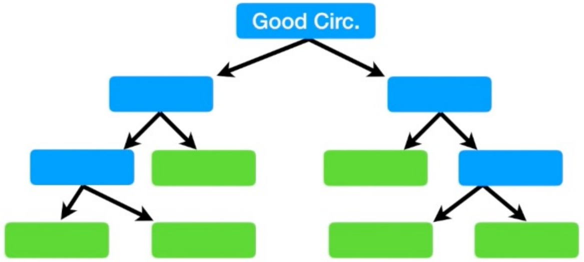
Step 2: Tworzymy drzewo losowe na wybranym zbiorze używając losowego podzbioru kolumn.

Losowo wybieramy podzbiór

Znajdujemy najlepszy podział korzenia.

Losowo wybieramy podzbiór dla węzła

Budujemy drzewo rozważając tylko podzbiór cech

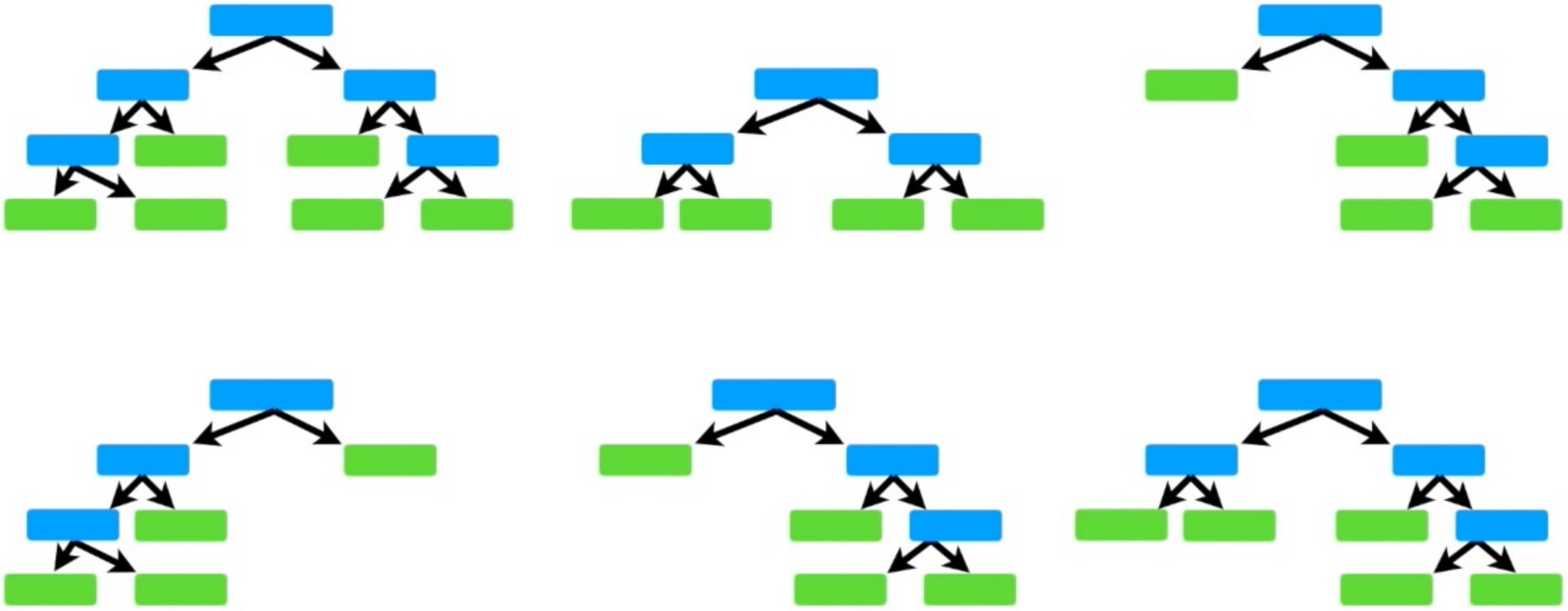


Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Sugar
Yes	Yes	Yes	180	121.3
No	No	No	125	90.01
Yes	No	Yes	167	137.67
Yes	No	Yes	167	137.67

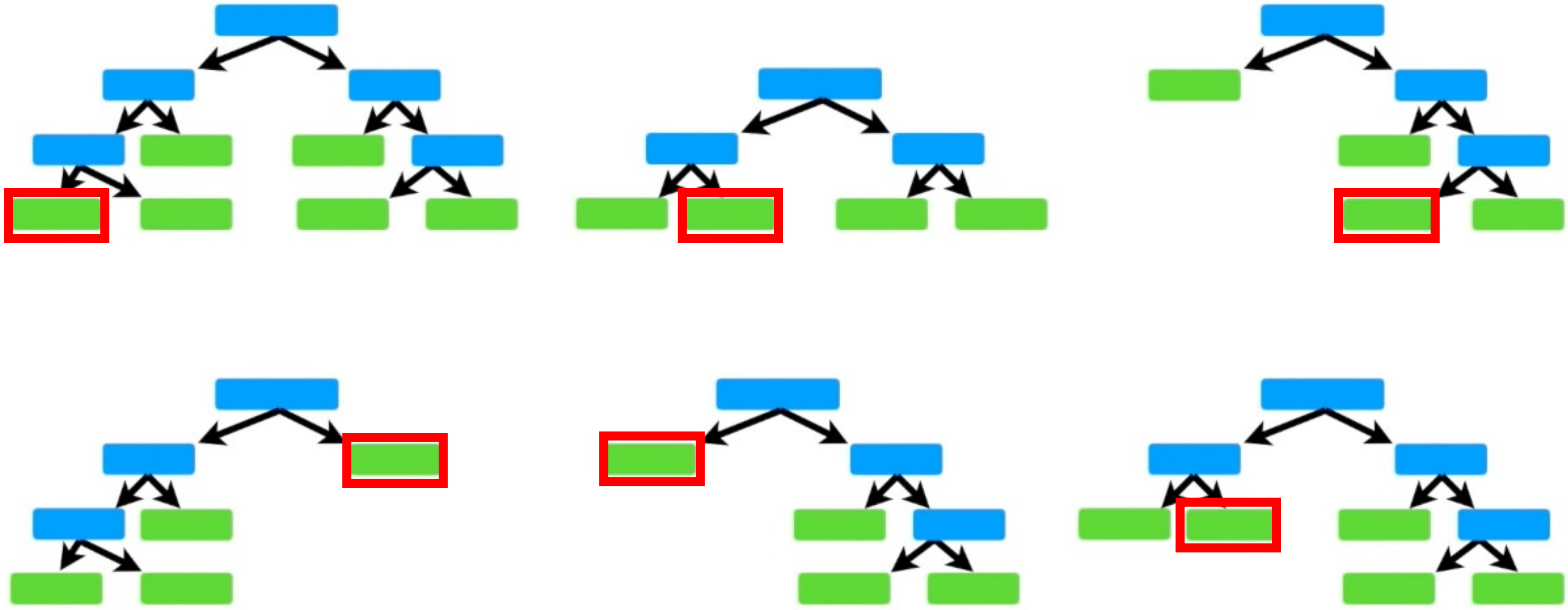
Regresyjne Lasy Losowe

Powtarzamy Step 1 i Step 2 tworząc kolejne drzewa.



Regresyjne Lasy Losowe

Wartość przewidywanej wartości to średnia ze wszystkich drzew.



Regresyjne Lasy Losowe

Out-Of-Bag Dataset

Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Sugar
No	No	No	125	90.01
Yes	Yes	Yes	180	121.3
Yes	Yes	No	210	96.50
Yes	No	Yes	167	137.67

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Sugar
Yes	Yes	Yes	180	121.3
No	No	No	125	90.01
Yes	No	Yes	167	137.67
Yes	No	Yes	167	137.67

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Sugar
Yes	Yes	No	210	96.50

Możemy przewidzieć wartości dla obserwacji z OOB a następnie policzyć na nim pewną metrykę.

W bibliotece sklearn moduł sklearn.ensemble.RandomForestRegressor posiada atrybut ***oob_score_***, który zwaraca R^2

.

oob_score_ : *float*

Score of the training dataset obtained using an out-of-bag estimate. This attribute exists only when `oob_score` is True.