

Линейни регресионни модели в машинното обучение

Д-р Людмил Йовков

Софийски университет „Св. Климент Охридски“, БАН

София, 30 ноември 2023



За автора



- СУ, „Приложна математика“ (BSc, 2013)
- СУ, „Изчислителна математика и математическо моделиране“ (MSc, 2015)
- БАН, „Математическо моделиране на кристализацията на метални сплави“ (PhD, 2021)
- Practical Data Science with MATLAB Specialization (MathWorks, 2022)
- Advanced Data Analytics Specialization (Google Professional Certificate, 2023)
- Machine Learning Specialization (Stanford University, 2023)

Съдържание

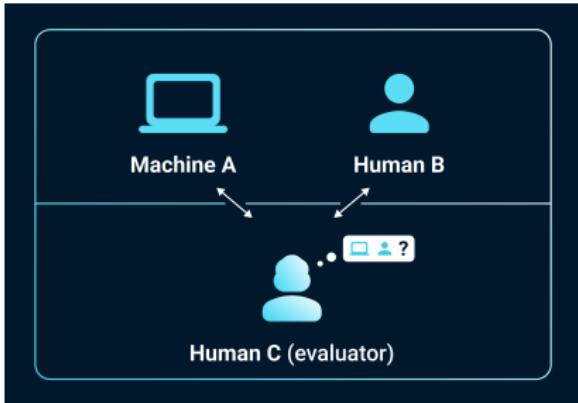
- ① Глава 1. Машинно обучение и класове машинно обучение
- ② Глава 2. Едномерна линейна регресия
- ③ Глава 3. Многомерна линейна регресия
- ④ Глава 4. Приложение на линейните регресионни модели в практиката

Линейни регресионни модели в машинното обучение. Глава 1



Глава 1. Машинно обучение и класове машинно обучение

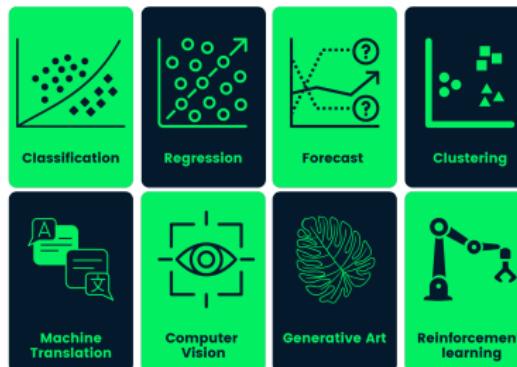
Исторически преглед



- ➊ 1950: тест на Тюринг
- ➋ 1957: Франк Розенблат, изкуствена невронна мрежа
- ➌ 1959: Артър Самюел, IBM, игра на пулове
- ➍ 1967: метод на най-близките съседи (k -means)
- ➎ 1990: data-driven approach
- ➏ 1997: IBM, Deep Blue, игра на шах
- ➐ 2006: Джофри Хинтън, понятието „дълбоко обучение“
- ➑ 2012: Google Brain, Google X Lab, идентифициране на обект котка
- ➒ 2014: Facebook, DeepFace
- ➓ 2020: OpenAI, GPT-3
- ➔ след 2020: квантови изчисления

Какво е машинно обучение?

1 Същност и основни концепции

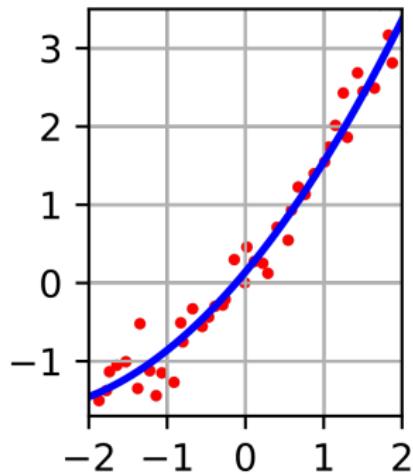


2 Класове машинно обучение

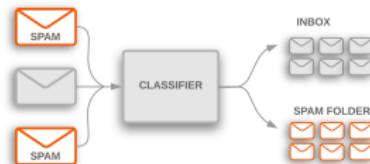
- надзорувано машинно обучение (supervised machine learning)
- ненадзорувано машинно обучение (unsupervised machine learning)
- подсилващо обучение (reinforcement learning)

Надзиравано машинно обучение

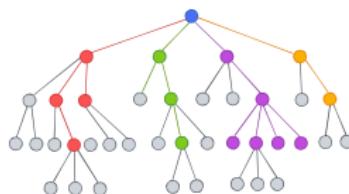
- 1 Същност
- 2 Основни алгоритми: регресионни и класифициращи



Фигура: Регресионни алгоритми



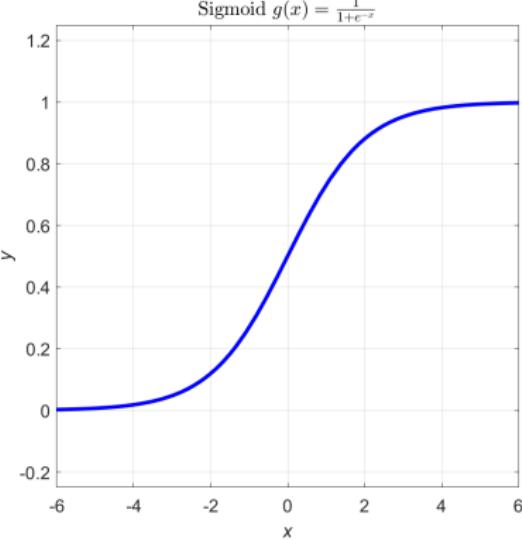
(а) Логистична регресия,
наивен Бейс, SVM метод



(б) Регресионни
дървета

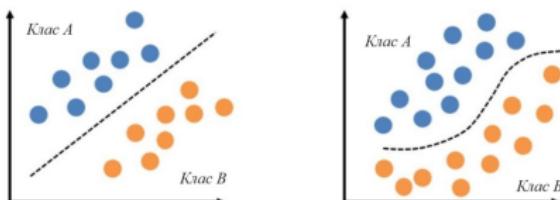
Фигура: Класифициращи
алгоритми

Надзиравано машинно обучение: логистична регресия



Фигура: σ -функция

$$g(z) = \frac{1}{1 + e^{-z}}, z = ax + b, \theta \in (0; 1)$$
$$x^* \in \mathbb{R} \Rightarrow g(z(x^*)) = g^* \in (0; 1)$$
$$\Rightarrow (x^*; z(x^*)) \in \begin{cases} \text{клас } A, & g^* \leq \theta \\ \text{клас } B, & g^* > \theta \end{cases}$$

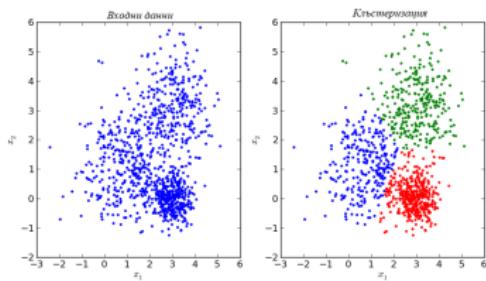


Фигура: Класова отделимост

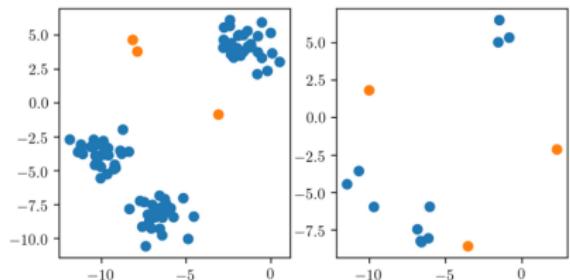
Ненадзорано машинно обучение

1 Същност

2 Основни алгоритми: клъстеризация, засичане на аномалии

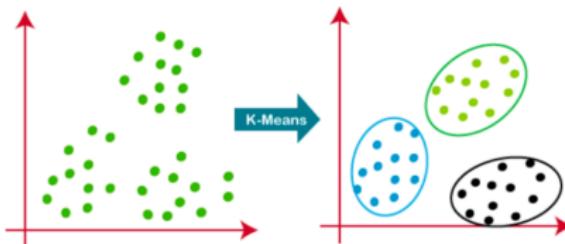


Фигура: Клъстеризация



Фигура: Засичане на аномалии

Ненадзорано машинно обучение: k-means



Фигура: Метод на най-близките съседи (k-means)

$$\mathbb{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}\},$$

$$\mathbf{x}^{(j)} \in \mathbb{R}^n, j = \overline{1, m},$$

$$\mathbb{M} = \{\mu_1, \mu_2, \dots, \mu_K\},$$

$$\mu_s \in \mathbb{R}^n, s = \overline{1, K}$$

μ_s : центроиди

■ Стъпка 1. Инициализация

$$\mu_j = \overset{(0)}{\mu_j}, j = \overline{1, K}$$

■ Стъпка 2. Клъстеризация

$$\forall \mathbf{x}^{(j)} \in \mathbb{X}, \mu_s \in \mathbb{M} : d_{j,s} = \|\mathbf{x}^{(j)} - \mu_s\|$$

$$d_{j,p} = \min \|\mathbf{x}^{(j)} - \mu_p\|, 1 \leq p \leq K$$

$$\Rightarrow \mathbf{x}^{(j)} \in \text{клъстер}_p$$

■ Стъпка 3. Обновяване на μ_s

$$\overset{(r)}{\mu_j} = \text{mean}\{\text{текущ клъстер}\}, r \geq 1$$

■ Повторение на 2. и 3. до сходимост

Линейни регресионни модели в машинното обучение. Глава 2



Глава 2. Едномерна линейна регресия

Постановка на регресионната задача

Дадено:

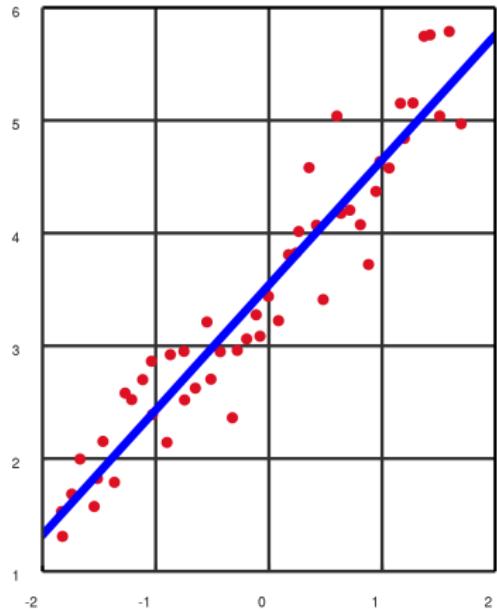
$$\mathbf{x} = (x_1, x_2, \dots, x_n),$$

$$\mathbf{y} = (y_1, y_2, \dots, y_n)$$

Входни данни: $\left\{ (x_j; y_j) \right\}_{j=1}^n$

\mathbf{x}	x_1	x_2	\cdots	x_n
\mathbf{y}	y_1	y_2	\cdots	y_n

Търсим: $\hat{y}(x) = w\mathbf{x} + b$
така, че $y_j \approx \hat{y}(x_j)$, $j = \overline{1, n}$, в
средноквадратичен смисъл.



Фигура: Зависимост данни —
регресионна права

Постановка на регресионната задача

■ Апроксимация

$$\hat{y} = w\mathbf{x} + b, \quad (w, b) \in \mathbb{R}^2$$

■ Целева функция $J(w, b)$

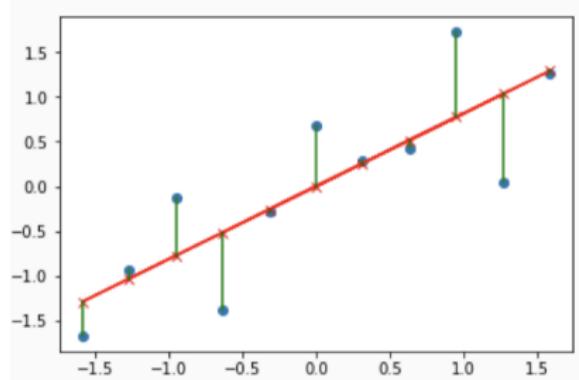
$$r_j = \hat{y}_j - y_j, \quad j = \overline{1, n} : \text{резидуали}$$

$$\text{Loss}_j = r_j^2 = (\hat{y}_j - y_j)^2, \quad j = \overline{1, n}$$

$$\text{Cost } J(w, b) = \frac{1}{n} \sum_{j=1}^n r_j^2 = \frac{1}{n} \sum_{j=1}^n \text{Loss}_j$$

■ Оптимизационна задача

$$\underset{(w,b) \in \mathbb{R}^2}{\operatorname{argmin}} J(w, b)$$



Фигура: Резидуали

Предварителни сведения от анализа

Необходимо условие за минимум в \mathbb{R}^n

Ако функцията $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, има минимум в точката \mathbf{x}^* и е диференцируема в тази точка, то

$$\frac{\partial f}{\partial x_i}(\mathbf{x}^*) = 0, i = \overline{1, n}.$$

Достатъчно условие за минимум в \mathbb{R}^n

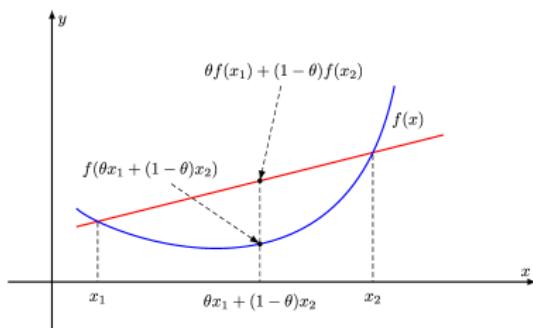
Ако функцията $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, е двукратно диференцируема в точката \mathbf{x}^* , частните ѝ производни $\frac{\partial f}{\partial x_i}$, $i = \overline{1, n}$, се анулират в \mathbf{x}^* и матрицата на Хесе е положително дефинитна, то $f(\mathbf{x})$ достига минимум в \mathbf{x}^* .

Предварителни сведения от анализа

Изпъкналост в \mathbb{R}^n

Функцията $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, се нарича изпъкнала в множеството $D \subseteq \mathbb{R}^n$, ако за всеки две точки $\mathbf{x}_1, \mathbf{x}_2 \in D$ и за всяко $\theta \in (0; 1)$ е изпълнено неравенството

$$f(\theta\mathbf{x}_1 + (1 - \theta)\mathbf{x}_2) \leq \theta f(\mathbf{x}_1) + (1 - \theta)f(\mathbf{x}_2).$$



Графиката на изпъкнала функция е разположена „под хордата“.

Фигура: 1D изпъкнала функция

Предварителни сведения от анализа

Критерий за изпъкналост в \mathbb{R}^n

Нека реалнозначната функция $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, е двукратно диференцируема в множеството $M \subseteq \mathbb{R}^n$ и за всяко $\mathbf{x} \in M$ съответната матрица на Хесиан

$$H = \begin{bmatrix} f_{x_1 x_1} & f_{x_1 x_2} & \cdots & f_{x_1 x_n} \\ f_{x_2 x_1} & f_{x_2 x_2} & \cdots & f_{x_2 x_n} \\ \vdots & \vdots & \cdots & \vdots \\ f_{x_n x_1} & f_{x_n x_2} & \cdots & f_{x_n x_n} \end{bmatrix}$$

е положително дефинитна. Тогава $f(\mathbf{x})$ е изпъкнала в M .

Изпъкналост на целевата функция $J(w, b)$

Лема

Функцията $J(w, b)$ е изпъкната навсякъде в \mathbb{R}^2 .

Доказателство. За вторите частни производни на целевата функция

$\tilde{J}(w, b) = nJ(w, b) = \sum_{j=1}^n [y_j - (wx_j + b)]^2$ последователно получаваме

$$\frac{\partial^2 \tilde{J}}{\partial w^2} = \sum_{j=1}^n 2x_j^2, \quad \frac{\partial^2 \tilde{J}}{\partial b^2} = \sum_{j=1}^n 2, \quad \frac{\partial^2 \tilde{J}}{\partial w \partial b} = \frac{\partial^2 \tilde{J}}{\partial b \partial w} = \sum_{j=1}^n 2x_j$$

$$\Rightarrow H = \begin{bmatrix} \frac{\partial^2 \tilde{J}}{\partial w^2} & \frac{\partial^2 \tilde{J}}{\partial w \partial b} \\ \frac{\partial^2 \tilde{J}}{\partial b \partial w} & \frac{\partial^2 \tilde{J}}{\partial b^2} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n 2x_j^2 & \sum_{j=1}^n 2x_j \\ \sum_{j=1}^n 2x_j & \sum_{j=1}^n 2 \end{bmatrix}.$$

Изпъкналост на целевата функция $J(w, b)$

Пресмятаме главните минори $|H_{(1,1)}|$ и $|H_{(2,2)}|$:

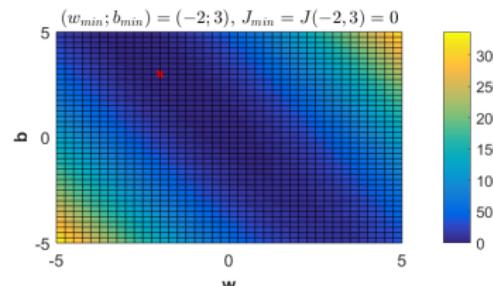
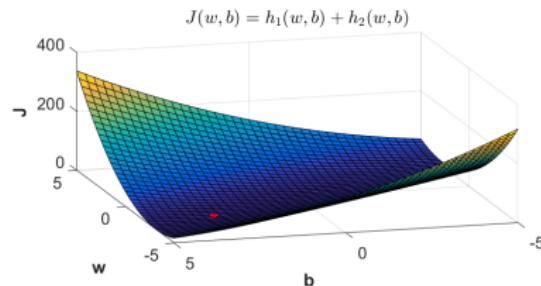
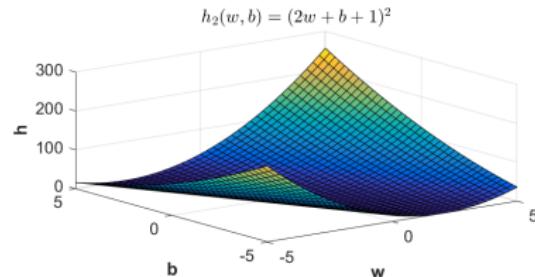
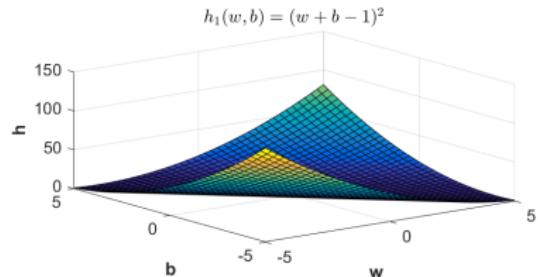
$$\frac{1}{2}|H_{(1,1)}| = \sum_{j=1}^n x_j^2 > 0,$$

$$\frac{1}{4}|H_{(2,2)}| = n \sum_{j=1}^n x_j^2 - \left(\sum_{j=1}^n x_j \right)^2 =$$

$$\begin{aligned} &= (x_1 - x_2)^2 + (x_1 - x_3)^2 + (x_1 - x_4)^2 + \cdots + (x_1 - x_n)^2 + \\ &\quad + (x_2 - x_3)^2 + (x_2 - x_4)^2 + \cdots + (x_2 - x_n)^2 + \\ &\quad + (x_3 - x_4)^2 + \cdots + (x_3 - x_n)^2 + \\ &\quad \cdots \\ &\quad + (x_{n-1} - x_n)^2 > 0 \end{aligned}$$

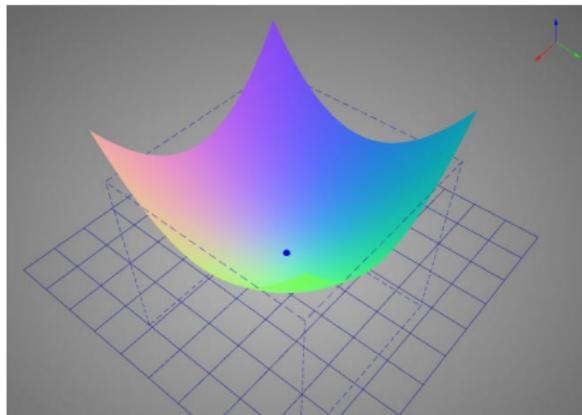
$\Rightarrow H$ — положително дефинитна $\Rightarrow J(w, b)$ — изпъкнала ■

Изпъкналост на целевата функция $J(w, b)$



Фигура: Глобален минимум на изпъкната функция — пример

Директен метод за получаване глобалния минимум



За целевата функция

$$J(w, b) = \frac{1}{n} \sum_{j=1}^n [y_j - (wx_j + b)]^2$$

от НДУ за минимум имаме

$$\frac{\partial J}{\partial w} = -\frac{2}{n} \sum_{j=1}^n [y_j - (wx_j + b)] x_j = 0,$$

$$\frac{\partial J}{\partial b} = -\frac{2}{n} \sum_{j=1}^n [y_j - (wx_j + b)] = 0.$$

Въвеждаме означенията

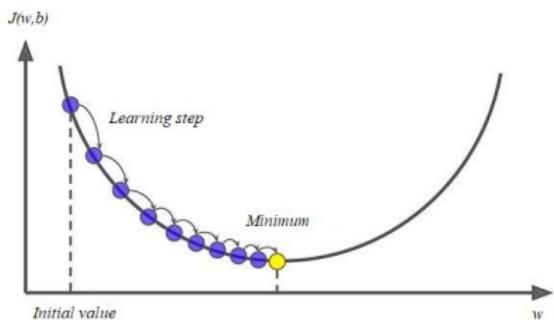
$$\beta_1 = \sum_{j=1}^n x_j, \quad \beta_2 = \sum_{j=1}^n x_j^2,$$

$$\beta_3 = \sum_{j=1}^n x_j y_j, \quad \beta_4 = \sum_{j=1}^n y_j$$

и от уравненията получаваме

$$w_{\min} = \frac{n\beta_3 - \beta_1\beta_4}{n\beta_2}, \quad b_{\min} = \frac{\beta_1\beta_3 - \beta_2\beta_4}{\beta_2^2 - n\beta_2}.$$

Приближено решаване на оптимизационната задача за минимум: градиентен метод



Фигура: Метод на най-бързото спускане (gradient descent)

Градиентен метод:

- $j = 0 : \mathbf{p}^0 = (w^0, b^0)$
- $j \geq 1 : \mathbf{p}^{j+1} = (w^{j+1}, b^{j+1})$

$$w^{j+1} = w^j - \alpha \frac{\partial J}{\partial w}(w^j, b^j)$$

$$b^{j+1} = b^j - \alpha \frac{\partial J}{\partial b}(w^j, b^j)$$

Стоп-критерий:

- $\|\mathbf{p}^{j+1} - \mathbf{p}^j\| < \varepsilon, \varepsilon \in \mathbb{R}^+$
- достигнат макс. брой итерации n_{\max}
- Точност: линейна

Оценка на апроксимацията — регресионни метрики

Регресионни метрики

$$\text{Sum of squares due to error } SSE = \sum_{j=1}^n (y_j - \hat{y}_j)^2$$

$$\text{Total sum of squares } SST = \sum_{j=1}^n (y_j - \bar{y})^2$$

$$\text{Sum of squares of the regression } SSR = \sum_{j=1}^n (\hat{y}_j - \bar{y})^2$$

$$\text{Coefficient of determination } R^2 = \frac{SSR}{SST} \in (0; 1)$$

$$\text{Mean squared error } MSE = \frac{SSE}{n}$$

$$\text{Rooted mean squared error } RMSE = \sqrt{MSE}$$

Оценка апроксимацията — обучаващо и тестово множество



- Обучаващо множество (training set) — построяване на регресионния модел
- Тестово множество (test set) — потвърждаване на резултатите от регресионния модел
- Променливо съотношение (training set) : (test set)

Линейни регресионни модели в машинното обучение. Глава 3

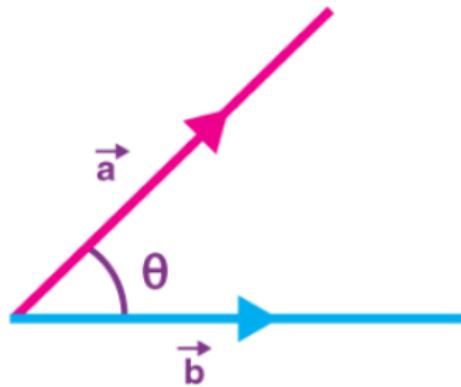


Глава 3. Многомерна линейна регресия

Постановка на регресионната задача

Скалярно произведение

Нека $\vec{a} = (a_1, a_2, \dots, a_n)$ и $\vec{b} = (b_1, b_2, \dots, b_n)$. Скалярно произведение на векторите $\vec{a} \in \mathbb{R}^n$ и $\vec{b} \in \mathbb{R}^n$ наричаме реалното число $\vec{a} \cdot \vec{b} = \|\vec{a}\| \cdot \|\vec{b}\| \cdot \cos \angle(\vec{a}, \vec{b}) = \sum_{k=1}^n a_k b_k$.



Пример

$$\vec{a} = (1, 0, 2, 3), \vec{b} = (0, 4, 1, 2)$$

$$\vec{a} \cdot \vec{b} = 0 + 0 + 2 + 6 = 8$$

$$\|\vec{a}\| = \sqrt{14}, \|\vec{b}\| = \sqrt{21}$$

$$\cos \theta = \frac{8}{\sqrt{294}}, \theta \approx 62.1882^\circ$$

Постановка на регресионната задача

\mathbf{x}_1	\mathbf{x}_2	\cdots	\mathbf{x}_n	y
$x_1^{(1)}$	$x_2^{(1)}$	\cdots	$x_n^{(1)}$	y_1
$x_1^{(2)}$	$x_2^{(2)}$	\cdots	$x_n^{(2)}$	y_2
\vdots	\vdots	\cdots	\vdots	\vdots
$x_1^{(m)}$	$x_2^{(m)}$	\cdots	$x_n^{(m)}$	y_m

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{bmatrix},$$

$$\mathbf{y} = (y_1, y_2, \dots, y_m)^T \in \mathbb{R}^m,$$

$$\mathbf{w} = (w_1, w_2, \dots, w_n) \in \mathbb{R}^n,$$

$$\mathbf{X} \in \mathbb{R}^{m \times n}$$

- Променливи на модела (features): $x_k (k = \overline{1, n})$
- Параметри на модела (parameters): $w_k (k = \overline{1, n}), b$
- $\mathbf{x}^{(j)} = \mathbf{X}(j, :)$: j -ти тренировъчен пример, $j = \overline{1, m}$
- $\mathbf{x} = (x_1, x_2, \dots, x_n)$: произволен тренировъчен пример
- Приближаваща функция:

$$\hat{y}(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b = \mathbf{w} \cdot \mathbf{x} + b$$

Постановка на регресионната задача

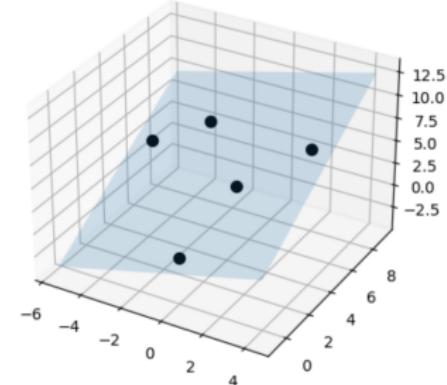
■ Целева функция $J(\mathbf{w}, b)$

$$\begin{aligned}\text{Loss}_j &= h_j(\mathbf{w}, b) = (y_j - \hat{y}_j)^2 \\ &= [y_j - (\mathbf{w} \cdot \mathbf{x}^{(j)} + b)]^2 \\ \text{Cost } J(\mathbf{w}, b) &= \frac{1}{m} \sum_{j=1}^m h_j(\mathbf{w}, b)\end{aligned}$$

■ Оптимизационна задача

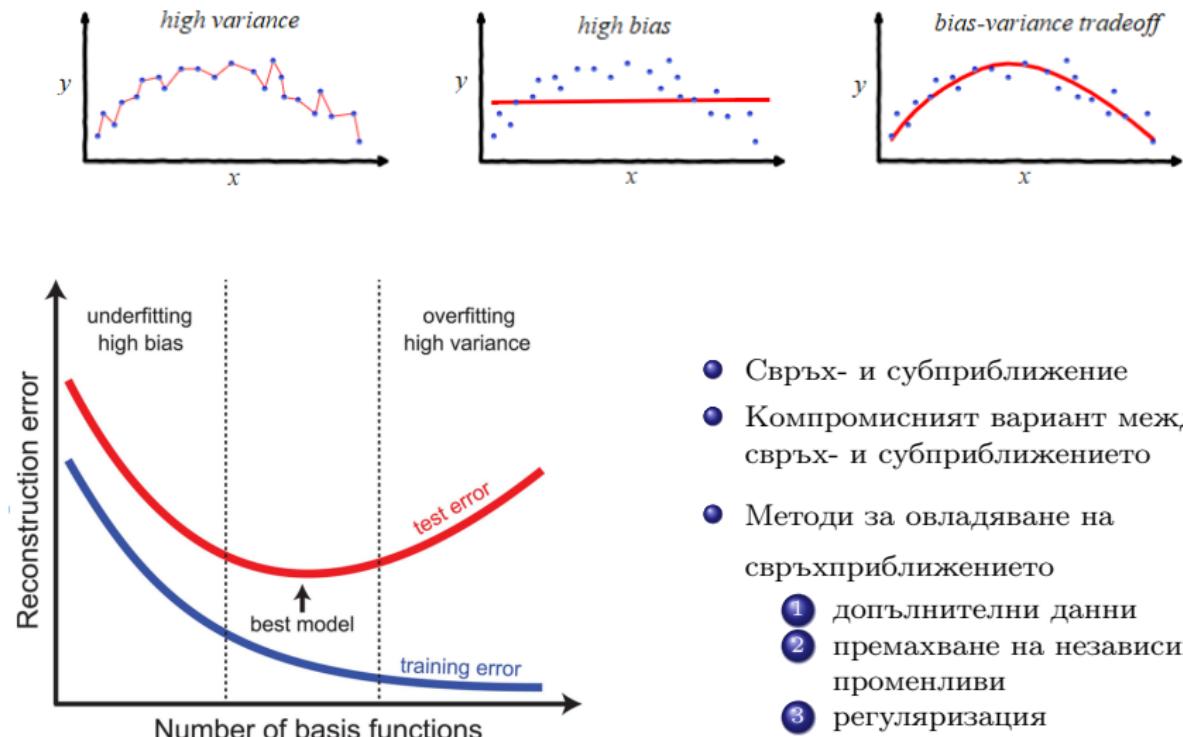
$$\underset{(\mathbf{w}, b) \in \mathbb{R}^{n+1}}{\operatorname{argmin}} J(\mathbf{w}, b)$$

■ Изпъкналост на $J(\mathbf{w}, b)$ и единственост на оптималното решение



Фигура: Многомерна линейна регресия: хиперравнина на най-добро среднокв. приближение

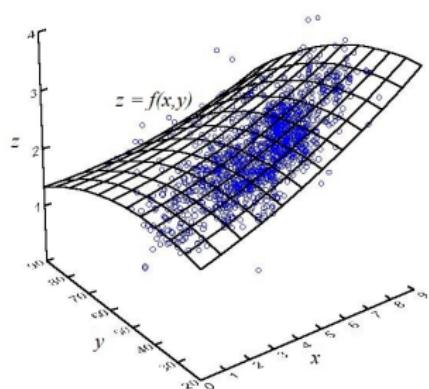
Върху въпроса за свръх- и субприближението



Фигура: Баланс bias — variance

Многомерна регресия с нелинейни функции

■ Мотивация



Фигура: Нелинейна многомерна регресия

$$\hat{y}(\mathbf{x}) = b + \sum_{k=1}^p w_k f_k(\mathbf{x}) \\ = b + w_1 f_1(\mathbf{x}) + \cdots + w_p f_p(\mathbf{x}),$$

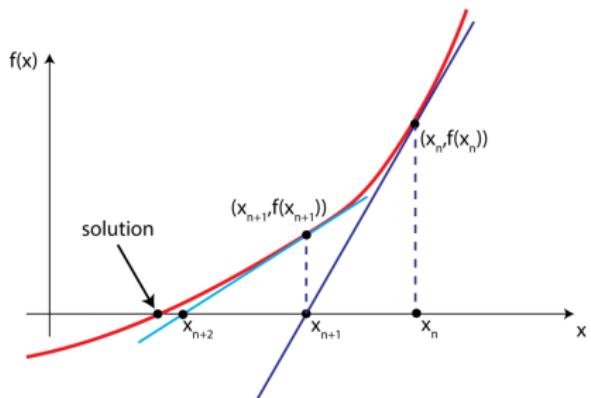
$\left\{f_k(\mathbf{x})\right\}_{k=1}^p$ — нелинейни относно \mathbf{x}

■ Избор на функциите $f_k(\mathbf{x})$

- $f_k(\mathbf{x}) = \mathbf{x}^\alpha$, $\mathbf{x} \neq \mathbf{0}$, $\alpha \in \mathbb{Q}$
- $f_k(\mathbf{x}) = a^{\mathbf{x}}$, $a > 0$, $a \neq 1$
- $f_k(\mathbf{x}) = \log_a \mathbf{x}$, $a > 0$, $\mathbf{x} > \mathbf{0}$
- $f_k(\mathbf{x}) = \sin \mathbf{x}$, $f_k(\mathbf{x}) = \tan \mathbf{x}$
- рационални функции на \mathbf{x}

Приближено решаване на оптимизационната задача за минимум: метод на Нютон

■ Метод на Нютон за нелинейното уравнение $f(x) = 0$



Фигура: Метод на Нютон за решаване на нелинейното уравнение $f(x) = 0$

- Същност на метода
- Итерационна формула:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Точност: квадратична

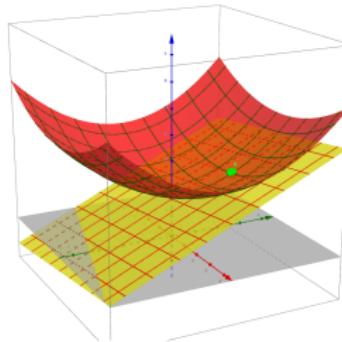
$$e_n = |x_n - x^*|$$

$$e_{n+1} = |x_{n+1} - x^*|$$

$$\Rightarrow e_{n+1} \leq \text{const} \cdot e_n^2$$

Приближено решаване на оптимизационната задача за минимум: метод на Нютон

- Метод на Нютон за нелинейната система $\mathbf{f}(\mathbf{x}) = \mathbf{0}$



Фигура: Метод на Нютон за решаване на нелинейната система $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ – общ случай

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$$

$$f_k(\mathbf{x}) = f_k(x_1, x_2, \dots, x_n), k = \overline{1, n}$$

$$\mathbf{f} = (f_1, f_2, \dots, f_n)$$

- Начално приближение:

$$\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$$

- Итерационна формула:

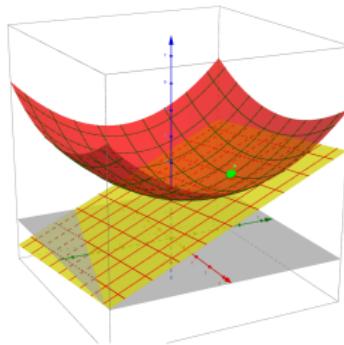
$$\mathbf{x}^{n+1} = \mathbf{x}^n - \mathbf{J}_\mathbf{f}^{-1}(\mathbf{x}^n) \cdot \mathbf{f}(\mathbf{x}^n)$$

$$\mathbf{J}_\mathbf{f} = \frac{D[f_1, f_2, \dots, f_n]}{D[x_1, x_2, \dots, x_n]} : \text{якобиан}$$

- Точност: квадратична

Приближено решаване на оптимизационната задача за минимум: метод на Нютон

- Метод на Нютон за системата $dJ(\mathbf{w}, b) = \mathbf{0}$



Фигура: Метод на Нютон за решаване на системата $dJ(\mathbf{w}, b) = \mathbf{0}$

$$\mathbf{b} := \mathbf{w}_{n+1}, \mathbf{p} = (w_1, w_2, \dots, w_n, w_{n+1}) \in \mathbb{R}^{n+1}$$

$$\frac{\partial J}{\partial w_k}(\mathbf{p}) = f_k(w_1, w_2, \dots, w_{n+1}), k = \overline{1, n+1}$$

$$\mathbf{f} = (f_1, f_2, \dots, f_n, f_{n+1})$$

- Начално приближение:

$$\mathbf{p}^0 = (w_1^0, w_2^0, \dots, w_n^0, w_{n+1}^0)$$

- Итерационна формула:

$$\mathbf{p}^{n+1} = \mathbf{p}^n - \mathbf{J}_f^{-1}(\mathbf{p}^n) \cdot \mathbf{f}(\mathbf{p}^n) \parallel \mathbf{J}_f(\mathbf{p}^n).$$

$$\Rightarrow \mathbf{J}_f(\mathbf{p}^n)\mathbf{p}^{n+1} = \mathbf{J}_f(\mathbf{p}^n)\mathbf{p}^n - \mathbf{f}(\mathbf{p}^n)$$

Линейни регресионни модели в машинното обучение. Глава 4



Глава 4. Приложения на линейните регресионни модели в практиката

Апроксимация на количество емисии от CO₂

- Постановка на проблема

	1	2
	Years	Carbon_Emission
1	1980	338.7000
2	1982	341.1000
3	1984	344.4000
4	1986	347.2000
5	1988	351.5000
6	1990	354.2000
7	1992	356.4000
8	1994	358.9000
9	1996	362.6000
10	1998	366.6000
11	2000	369.4000

Years → x

Carbon_Emissions → $y(x)$

$$\text{len}(x) = \text{len}(y) = 11$$

$$y \approx \hat{y} = wx + b$$

- train : test = 70% : 30%
- $\hat{y} = 1.5509x - 2732.5824$
- Регресионни метрики

MSE_{train} = 0.32935, RMSE_{train} = 0.5739

MSE_{test} = 0.139, RMSE_{test} = 0.3728

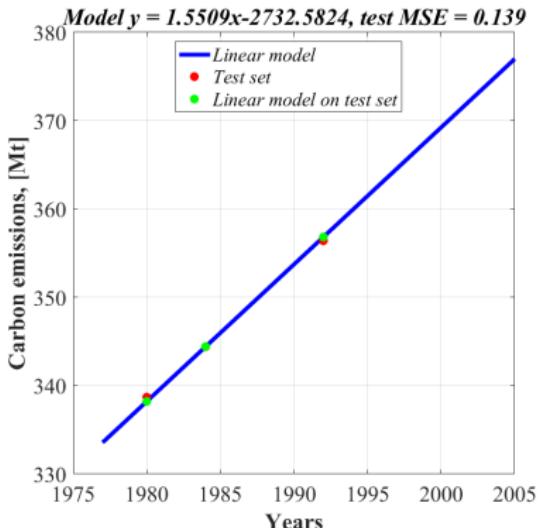
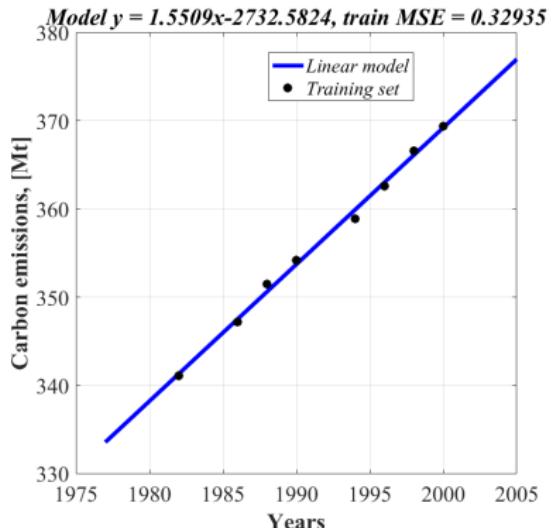
- Моделна прогноза

$$x = 2002 \text{ г.} : y \approx 372.3343 \text{ Mt}$$

$$x = 2005 \text{ г.} : y \approx 376.9970 \text{ Mt}$$

$$x = 2008 \text{ г.} : y \approx 381.6397 \text{ Mt}$$

Апроксимация на количество емисии от CO₂



Фигура: Линейна регресионна зависимост година — общо количество въглеродни емисии

Линейна апроксимация на MPG

● Постановка на проблема

data_table					
	1	2	3	4	5
	Acceleration	Displacement	Horsepower	MPG	Weight
1	12	307	130	18	3504
2	11.5000	350	165	15	3693
3	11	318	150	18	3436
4	12	304	150	16	3433
5	10.5000	302	140	17	3449
6	10	429	198	15	4341
7	9	454	220	14	4354
8	8.5000	440	215	14	4312
9	10	455	225	14	4425
10	8.5000	390	190	15	3850

Фигура: Множество от входни данни с 4 независими променливи $x_{1,2,3,4}$ и 1 зависима променлива y , $\text{len}(x_i) = \text{len}(y) = 93$ (след предварителна обработка на данните).

Horsepower $\rightarrow x_1$

Weight $\rightarrow x_2$

Acceleration $\rightarrow x_3$

Displacement $\rightarrow x_4$

MPG $\rightarrow y$

$$\hat{y} \approx \hat{y} = w_1 x_1 + w_2 x_2 + b$$

● train : test = 70% : 30%

● Регресионно уравнение

$$\hat{y} = -0.044094x_1 - 0.0069462x_2 + 48.878$$

● Регресионни метрики

$$\text{RMSE}_{\text{train}} = 3.77, R^2_{\text{train}} = 0.807$$

$$\text{RMSE}_{\text{test}} = 24.7412, R^2_{\text{test}} = 0.98657$$

Линейна апроксимация на MPG

y_pred	y_test	error
9.9935	15	5.0065
9.4453	14	4.5547
8.2195	14	5.7805
25.332	18	7.3317
28.03	24	4.0298
28.377	26	2.3771
26.516	21	5.5156
29.885	25	4.8854
26.955	25	1.9553
20.957	22	1.0429
20.613	18	2.6125
18.708	18.5	0.20834
33.07	29.5	3.5701
27.886	26.5	1.3858
23.763	19	4.763
10.516	16.5	5.9837
27.635	29	1.3655
24.92	24	0.92006
31.861	36	4.1389
31.03	36	4.9697
30.254	36	5.7459
30.197	34	3.8032
23.571	25	1.4293
24.187	38	13.813
24.247	22	2.2466
24.418	27	2.5822
26.368	31	4.6318

- Максимална абсолютна грешка върху тестовото множество

$$\max |y_i - \hat{y}_i| = 13.8132$$

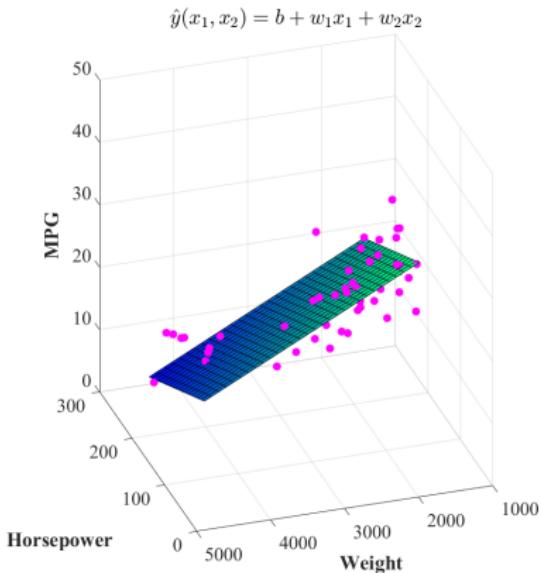
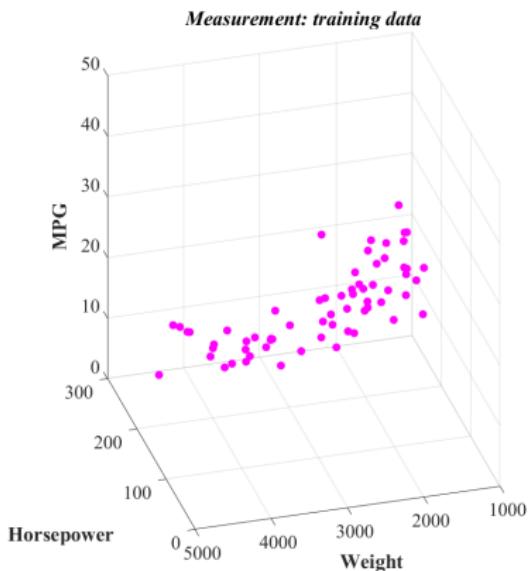
- Стандарти
 - кола среден размер: weight ≈ 3300 pounds, $180 < \text{hp} < 200$
 - кола голям размер: weight ≈ 4400 pounds, $200 < \text{hp} < 300$
- Моделна прогноза

$$x_1 = 120 \text{ hp}, x_2 = 3200 \text{ pounds} \Rightarrow \\ \hat{y}(x_1, x_2) = 21.3584 \approx 21.4 \text{ mpg}$$

$$x_1 = 250 \text{ hp}, x_2 = 4000 \text{ pounds} \Rightarrow \\ \hat{y}(x_1, x_2) = 10.0692 \approx 10.1 \text{ mpg}$$

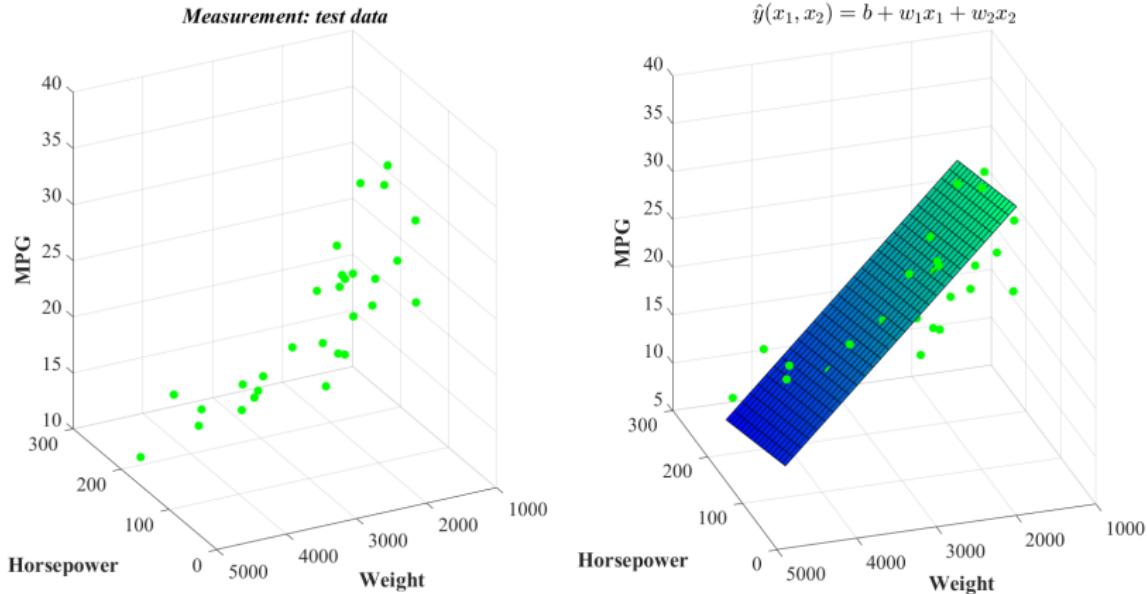
Фигура: Апроксимация върху тестовото множество

Линейна апроксимация на MPG



Фигура: Training set: $\text{RMSE}_{\text{train}} = 3.77$, $R^2_{\text{train}} = 0.807$

Линейна апроксимация на MPG



Фигура: Test set: $\text{RMSE}_{\text{test}} = 24.7412$, $R^2_{\text{test}} = 0.98657$

Нелинейна апроксимация на MPG

- Постановка на проблема

data_table 100x5 table

	1 Acceleration	2 Displacement	3 Horsepower	4 MPG	5 Weight
1	12	307	130	18	3504
2	11.5000	350	165	15	3693
3	11	318	150	18	3436
4	12	304	150	16	3433
5	10.5000	302	140	17	3449
6	10	429	198	15	4341
7	9	454	220	14	4354
8	8.5000	440	215	14	4312
9	10	455	225	14	4425
10	8.5000	390	190	15	3850

Фигура: Множество от входни данни с 4 независими променливи $x_{1,2,3,4}$ и 1 зависима променлива y , $\text{len}(x_i) = \text{len}(y) = 93$ (след предварителна обработка на данните).

Horsepower $\rightarrow x_1$

Weight $\rightarrow x_2$

Acceleration $\rightarrow x_3$

Displacement $\rightarrow x_4$

MPG $\rightarrow y$

$$y \approx \hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + b$$

- train : test = 70% : 30%
- Регресионно уравнение

$$\hat{y} = -0.1514x_1 - 0.0087179x_2 +$$

$$2.6725 \cdot 10^{-5}x_1 x_2 + 56.442$$

- Регресионни метрики

$$\text{RMSE}_{\text{train}} = 4.17, R^2_{\text{train}} = 0.753$$

$$\text{RMSE}_{\text{test}} = 17.2534, R^2_{\text{test}} = 0.83339$$

Нелинейна апроксимация на MPG

y_pred	y_test	error
17.567	16	1.5666
10.777	14	3.2229
11.077	14	2.9229
15.831	15	0.83096
14.032	14	0.031556
24.555	22	2.5547
26.897	21	5.8973
26.606	26	0.60551
27.605	28	0.39531
14.272	17.5	3.2284
14.002	16	1.9982
32.582	29	3.5819
18.728	18.5	0.22761
22.33	17.5	4.8303
22.125	20	2.1255
12.077	16.5	4.4231
16.468	13	3.4681
26.313	27	0.68683
24.581	24	0.58148
31.894	36	4.1065
32.174	37	4.8261
32.553	31	1.5534
31.957	38	6.0432
29.369	36	6.6311
26.334	26	0.33406
27.14	28	0.86016
26.276	31	4.7239

- Максимална абсолютна грешка върху тестовото множество

$$\max |y_i - \hat{y}_i| = 6.6311$$

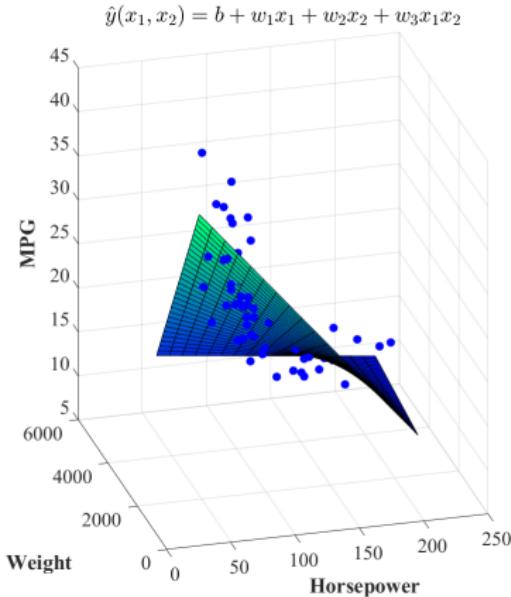
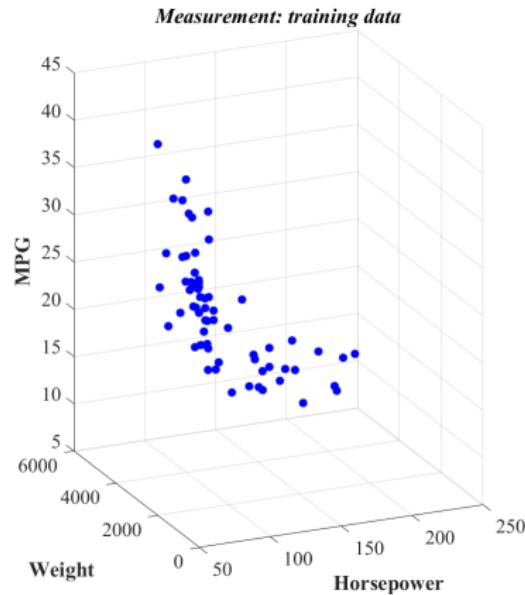
- Стандарти
 - кола среден размер: weight ≈ 3300 pounds, $180 < \text{hp} < 200$
 - кола голям размер: weight ≈ 4400 pounds, $200 < \text{hp} < 300$
- Моделна прогноза

$$x_1 = 120 \text{ hp}, x_2 = 3200 \text{ pounds} \Rightarrow \\ \hat{y}(x_1, x_2) = 20.6401 \approx 20.6 \text{ mpg}$$

$$x_1 = 250 \text{ hp}, x_2 = 4000 \text{ pounds} \Rightarrow \\ \hat{y}(x_1, x_2) = 10.4470 \approx 10.4 \text{ mpg}$$

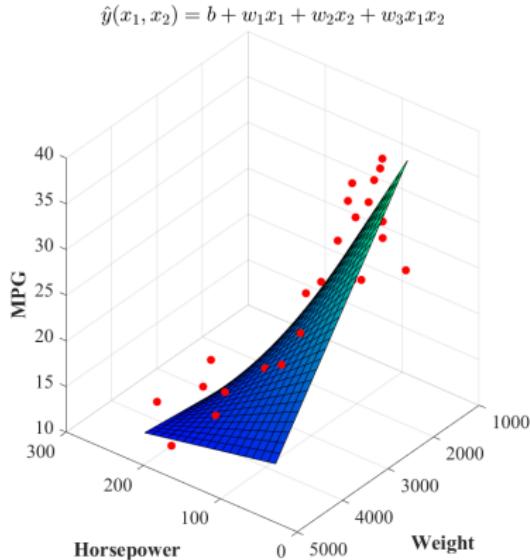
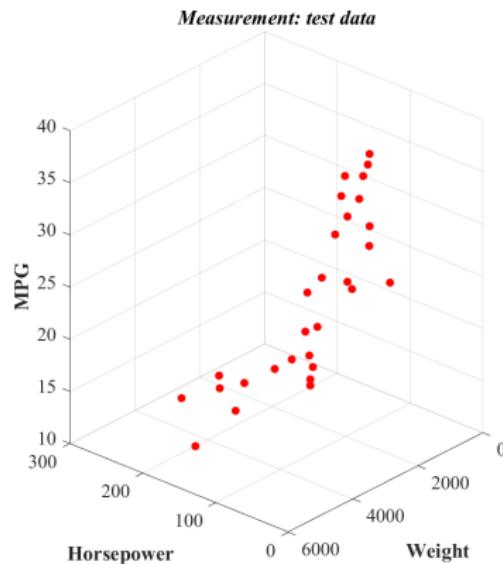
Фигура: Апроксимация върху тестовото множество

Нелинейна апроксимация на MPG



Фигура: Training set: $\text{RMSE}_{\text{train}} = 4.17$, $R^2_{\text{train}} = 0.753$

Нелинейна апроксимация на MPG



Фигура: Test set: $\text{RMSE}_{\text{test}} = 17.2534$, $R^2_{\text{test}} = 0.83339$

Информационни ресурси



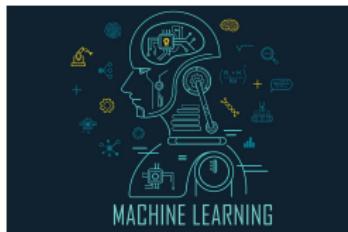
- Coursera: www.coursera.org
- DataCamp: www.datacamp.com
- Udemy: www.udemy.com
- Udacity: www.udacity.com
- RealPython: www.realpython.com
- LearnPython: www.learnpython.org
- TutorialsTeacher:
www.tutorialsteacher.com
- www.python.org
- www.pythonprogramming.net

Информационни ресурси



- Coursera: www.coursera.org
- LinkedIn: www.linkedin.com
- MathWorks: www.mathworks.com
- MATLAB product documentation

Информационни ресурси



- Coursera: www.coursera.org
- DataCamp: www.datacamp.com
- Udemy: www.udemy.com
- Udacity: www.udacity.com
- MathWorks: www.mathworks.com
- LinkedIn: www.linkedin.com
- TowardsDataScience:
www.towardsdatascience.com
- Deep learning:
www.ufldl.stanford.edu/tutorial/

