Wprowadzenie do analizy danych i uczenia maszynowego

Uczenie maszynowe I — regresja

Franciszek Saliński

Koło Naukowe Data Science, Wydział MiNI PW

13 grudnia 2025

Agenda

1. Zbiory treningowy/walidacyjny/testowy
2. Overfitting i underfitting
3. Data leakage
4. Przygotowanie danych:
   - data cleaning
   - feature engineering
5. Zadanie regresji
6. Regresja liniowa
7. Regularyzacja (Ridge, Lasso, Elastic Net)
8. Ocena modelu regresyjnego

Zbiory treningowy/walidacyjny/testowy

- Dane dzielimy typowo na:
  - **zbiór treningowy** (train) — służy do uczenia modelu,
  - **zbiór walidacyjny** (validation) — optymalizacja hiperparametrów,
  - **zbiór testowy** (test) — niezależna próba pozwalająca na końcową ocenę jakości modelu.
- Zbiór testowy:
  - powinien być *"niewidzialny"* dla modelu w trakcie całego procesu trenowania,
  - umożliwia symulowanie działania modelu w przyszłości.

Data leakage

- **Data leakage** to wyciek informacji ze zbioru walidacyjnego/testowego do modelu.
- Przykład:
  - standaryzacja zmiennych wykonana na *całym* zbiorze danych,
  - imputacja braków danych przy użyciu informacji z całego zbioru.
- Skutek:
  - model widzi informacje, które pochodzą ze zbioru testowego,
  - zawyżone wyniki na walidacji/testach, gorsze działanie w realnym świecie.
- Ważne: wszystkie przekształcenia danych powinny być **uczone tylko na train**, a dopiero później stosowane do val/test.

Overfitting i underfitting

- **Underfitting**:
  - model jest zbyt prosty,
  - nie potrafi uchwycić wzorców w danych,
  - duży błąd zarówno na zbiorze treningowym, jak i testowym.
- **Overfitting**:
  - model jest zbyt złożony,
  - uczy się dokładnie dopasować do treningowego zbioru zamiast ogólnej zależności,
  - niski błąd na treningu, wysoki na teście.
- Zwykle szukamy kompromisu, modelu, który jest:
  - dostatecznie elastyczny, by uchwycić zależność,
  - ale na tyle prosty, by radził sobie na niezależnych próbkach.

Jak walczyć z overfittingiem?

- Poprawny podział na train / val / test.
- Uproszczenie modelu.
- **Regularyzacja** — kara za zbyt duże wagi (o tym za chwilę).
- Zbieranie większej liczby danych (jeśli to możliwe) lub dobranie większego zbioru treningowego.
- Tworzenie komitetów modeli (ensemble learning) — bardziej złożony temat.

Przygotowanie danych: data cleaning

- **Braki danych**:
  - usuwanie obserwacji (puste lub prawie puste wiersze, duplikaty),
  - imputacja (np. średnią, medianą, modelami).
- **Wartości odstające (outliers)**:
  - mogą mocno wpłynąć na modele regresji, w szczególności liniowe,
  - Jak sobie radzić?
    - winsoryzacja,
    - transformacje cech
- **Formaty danych**:
  - daty, teksty, kategorie często mogą być niejednolite
  - spójne jednostki

Przygotowanie danych: feature engineering

- **Tworzenie nowych cech**:
  - np. cena za m^2, wskaźniki proporcji w danych finansowych.
- **Encoding zmiennych kategorycznych**:
  - one-hot encoding,
  - ordinal/label encoding.
- **Skalowanie cech**:
  - standaryzacja (średnia 0, wariancja 1),
  - min-max scaling,
  - ważne szczególnie w regresji.

Kodowanie zmiennych kategorycznych — przykład

Dane oryginalne:

ID	Marka
1	Audi
2	Citroen
3	Audi
4	Volksvagen

Label encoding

ID	Marka
1	0
2	1
3	0
4	2

One-hot encoding

ID	Audi	Citroen	Volksvagen
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1

Label encoding zwykle używamy, gdy kategorie można sensownie uszeregować (ordinal encoding) lub gdy jest ich bardzo dużo.

Zadanie regresji

- W zadaniu regresji przewidujemy **ciągłą wartość liczbową**.
- Przykłady:
  - cena mieszkania,
  - zużycie energii w budynku w danym dniu.
- Dane:
  - macierz danych $X \in \mathbb{R}^{n \times p}$, gdzie n to liczba obserwacji, a p liczba zmiennych
  - wektor etykiet $y \in \mathbb{R}^n$.
- Celem jest znalezienie funkcji $f(x)$ takiej, aby:

$$f(x) \approx y$$

na nowych, niewidzianych wcześniej obserwacjach.

Regresja liniowa: model

- Załóżmy, że istnieje w przybliżeniu liniowa zależność między cechami a y.
- Model regresji liniowej ma postać:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p,$$

gdzie:

  - β_0 — wyraz wolny,
  - $\beta_1, \ldots, \beta_p$ — wagi (parametry) modelu.

- Interpretacja geometryczna:
  - model szuka **hiperpłaszczyzny** w przestrzeni cech, która najlepiej przybliża punkty danych

Uczenie regresji liniowej

- Mamy dane treningowe:

$$(x^{(i)}, y^{(i)}), \quad i = 1, \ldots, n.$$

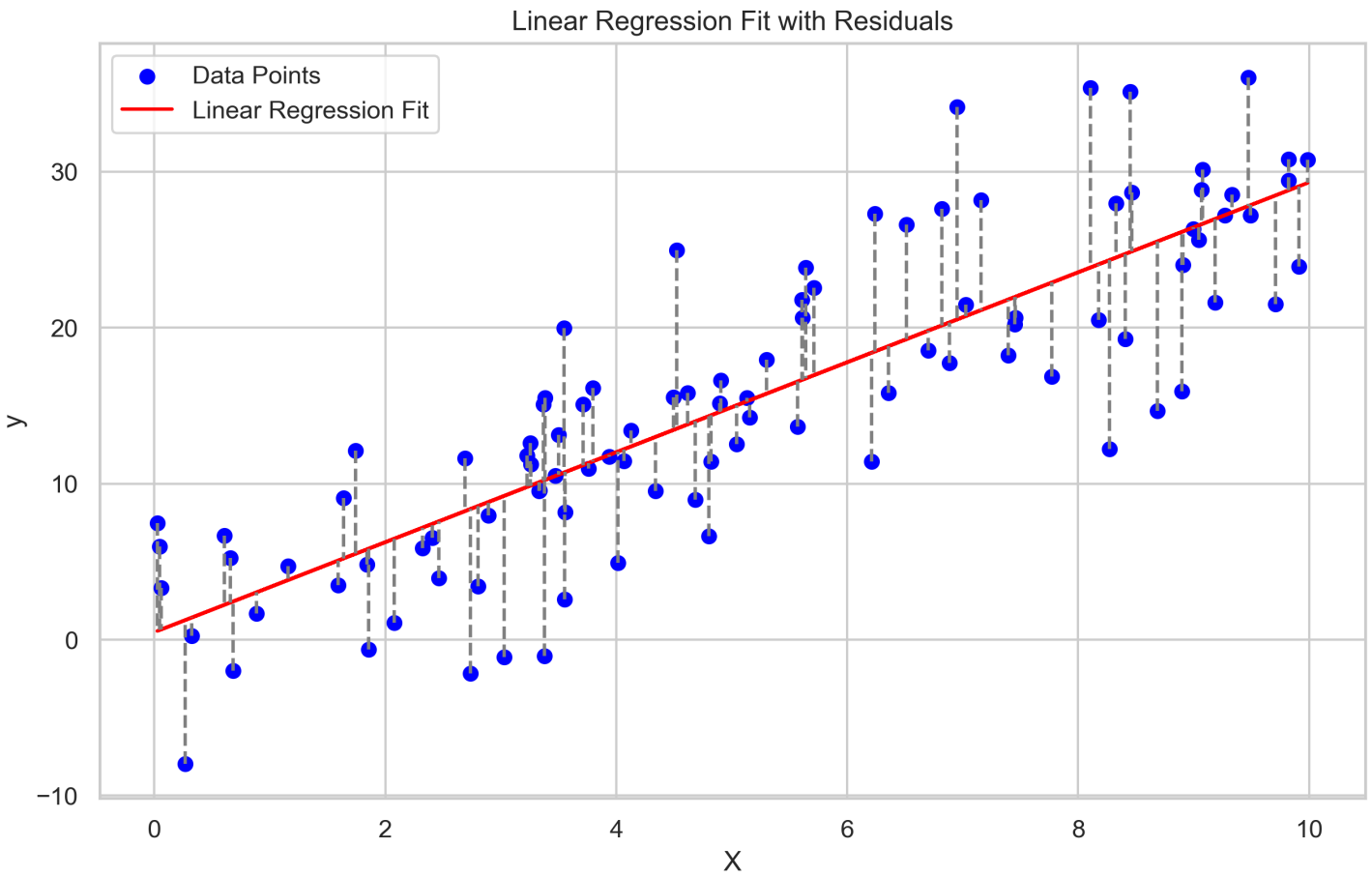
- Chcemy dobrać parametry $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$ tak, aby błąd predykcji był jak najmniejszy.
- Typowa funkcja kosztu: **średni błąd kwadratowy** (MSE):

$$J(\beta) = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \beta_0 - \sum_{j=1}^{p} \beta_j x_j^{(i)} \right)^2 .$$

- Uczenie polega na minimalizacji $J(\beta)$
- Geometrycznie, minimalizacja MSE jest równoważna **rzutowaniu ortogonalnemu** wektora y na **przestrzeń rozpiętą przez kolumny macierzy** X.

Wizualizacja regresji liniowej

Dopasowana linia regresji wraz z zaznaczonymi błędami $y - \hat{y}$.



Rysunek: Regresja liniowa z jedną zmienną wyjaśniającą

Ograniczenia regresji liniowej

- Zakłada **liniową** zależność między cechami a wynikiem.
- Wrażliwa na wartości odstające.
- Problemy przy **współliniowości** cech, gdy cechy są silnie skorelowane, parametry modelu stają się niestabilne.
- Przy dużej liczbie cech istnieje ryzyko **overfittingu**.
- Rozwiązaniem wielu z tych problemów jest **regularyzacja**.

Regularyzacja: po co?

- Intuicja:
  - chcemy, aby model nie dopasowywał się zbyt dokładnie do danych treningowych,
  - unikamy bardzo dużych wartości wag.
- Dodajemy do funkcji kosztu **karę** za duże wartości β_j:

$$J_{\mathsf{r}}(\beta) = J(\beta) + \mathsf{r}(\beta).$$

- Skutki regularyzacji:
  - bardziej stabilne parametry,
  - często lepsza generalizacja,
  - w przypadku L1 automatyczna selekcja zmiennych.

Regresja Ridge (L2)

- W regresji Ridge dodajemy karę w postaci normy L2 wag:

$$J_{\text{Ridge}}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2,$$

  gdzie $\lambda \geqslant 0$ jest hiperparametrem regularyzacji.
- Im większe λ, tym bardziej ściągamy wagi w stronę zera,
- Ridge nie zeruje wag, ale zmniejsza ich wartości.

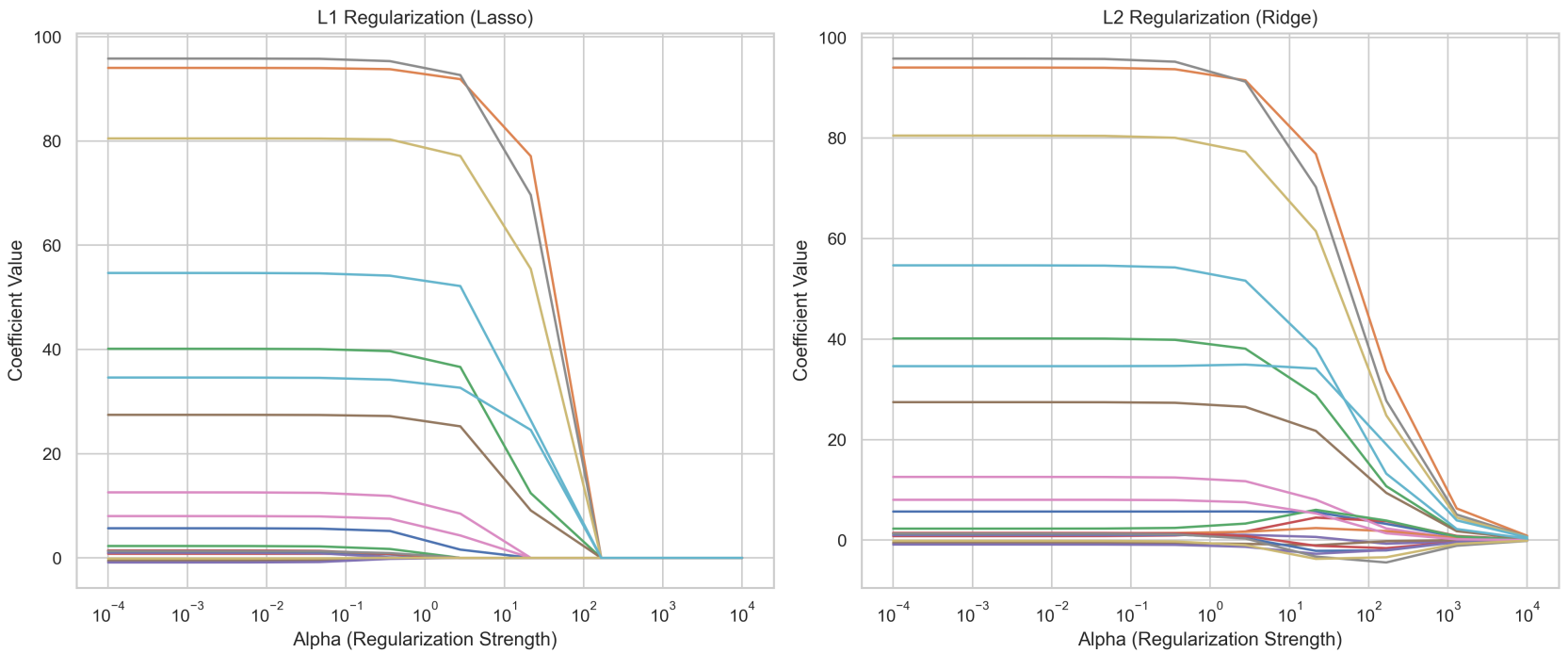
Regresja Lasso (L1)

- W regresji Lasso kara ma postać normy L1:

$$J_{\text{Lasso}}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\,.$$

- Skutek:
  - część wag jest dokładnie równa zero, więc Lasso wykonuje **selekcję cech**.
- Przy dużej liczbie cech może uprościć model do kilku najważniejszych predyktorów.

Lasso vs Ridge



Rysunek: Porównanie działania regularyzacji

Elastic Net

- Elastic Net łączy kary L1 i L2:

$$J_{\text{EN}}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda_1 \sum_{j=1}^{p}|\beta_j| + \lambda_2 \sum_{j=1}^{p}\beta_j^2.$$

- W praktyce często jest kompromisem między Ridge i Lasso.

Ocena modelu regresyjnego

- **Mean Squared Error**:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$

- **Root Mean Squared Error**:

$$\text{RMSE} = \sqrt{\text{MSE}}$$

- **Mean Absolute Error**:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y^{(i)} - \hat{y}^{(i)}|$$

- R^2 — wyjaśniona przez model liniowy część wariancji, (!) nie używać jako miary jakości predykcji:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^{n}(y^{(i)} - \bar{y})^2}$$