Laboratorium maszynowej analizy danych

Laboratorium 2

Wprowadzenie do uczenia nadzorowanego. Problem przewidywania wartości. Regresja liniowa.

Wprowadzenie

Ogólny schemat postępowania przy projekcie uczenia maszynowego:

- zdefiniowanie celu/problemu do rozwiązania
- pozyskanie i przygotowanie danych
- **wybór modelu**
- **wytrenowanie modelu na danych uczących**
- **dostrojenie modelu**
- **wykorzystanie wytrenowanego modelu do prognozowania wyników dla nowych przypadków**
- wdrażanie, monitorowanie, utrzymanie i konserwacja systemu

Przykładowe kryteria podziału systemów uczenia maszynowego:

- sposób nadzorowania w fazie uczenia:
  - **uczenie nadzorowane (ang. *supervised learning*)** – modeluje się relacje między cechami danych i powiązanymi z nimi etykietami. Po zakończeniu tego procesu za pomocą modelu można etykietować nowe i nieznane dane:
    - klasyfikacja – modele, które przewidują etykiety należące do dwóch lub więcej dyskretnych kategorii.
    - **regresja – modele przewidują ciągłe etykiety**.
  - uczenie nienadzorowane (ang. *unsupervised learning*) – modelowanie zbioru cech bez znajomości poprawnych etykiet:
    - klasteryzacja – modele, które wykrywają i identyfikują grupy w danych.
    - redukcja wymiarowości – modele, które w danych wielowymiarowych wykrywają i identyfikują strukturę o mniejszej liczbie wymiarów.
    - Asocjacja - wykrywanie zależności między elementami zbioru danych.
    - Wykrywanie anomalii
  - uczenie półnadzorowane (ang. *semisupervised learning*)
  - uczenie przez wzmocnienie (ang. *reinforcement learning*) - metoda prób i błędów (sygnał wzmocnienia – nagroda (pozytywny) i kara (negatywny)).
- możliwość uczenia się w czasie rzeczywistym:
  - uczenie przyrostowe (ang. *online learning*)
  - uczenie wsadowe (ang. *batch learning*)
- uogólnianie:
  - uczenie z przykładów (ang. *instance-based learning*)
  - uczenie z modelu

Główne problemy uczenia maszynowego:

- niedobór danych uczących
- niereprezentatywne dane uczące
- dane kiepskiej jakości
- nieistotne cechy
- **przetrenowanie danych uczących (ang. *overfitting*)**
- **niedotrenowanie danych uczących (ang. *underfitting*)**

Każdy model musi być trenowany na pewnym zbiorze danych, ale potrzebny jest również oddzielny zbiór obiektów do weryfikacji, czy wytrenowany model działa w sposób zadowalający.

Dlatego też, przed przystąpieniem do wyboru modelu należy dobrać właściwy sposób walidacji modelu:

- **Podział na zbiór uczący/treningowy i testowy:**
  - Model jest uczony na danych uczących, a następnie wydajność modelu jest oceniana na danych należących do zbioru testowego.

  * *Losowanie warstwowe (ang. stratified sampling)*

- **Podział na zbiór treningowy, walidacyjny i testowy:**
  - Z zestawu danych uczących wydzielony zostaje zbiór walidacyjny w celu zweryfikowania kilku różnych modeli. Trenowanych jest wiele modeli mających różne wartości hiperparametrów za pomocą zredukowanego zbioru uczącego i dobierany jest model najlepiej sprawujący się wobec zbioru walidacyjnego. Następnie najlepszy model jest trenowany na pełnym zestawie uczącym (wraz z walidacyjnym), co pozwala uzyskać model ostateczny. Na koniec należy przeprowadzić sprawdzian wobec zbioru testowego, aby oszacować wartość błędu uogólnania.

- **Walidacja krzyżowa (kroswalidacja, ang. *cross-validation*)*:**
  - k-krotny sprawdzian krzyżowy (ang. *k-fold cross-validation*)- dane są dzielone na *k* części/podzbiorów. Model jest następnie trenowany za pomocą *k-1* podzbiorów, a ostatni podzbiór jest używany w charakterze zbioru uczącego. Operacja zostaje powtórzona *k* razy, przy czym za każdym razem jako testowy wykorzystywany jest inny podzbiór. Kolejnym krokiem jest uśrednianie wydajności działania modelu dla wszystkich *k* iteracji, aby w ten sposób otrzymać ostateczny wynik.
    - metoda *leaving-one-out*

  * *stosowana przy małych zbiorach danych.*

Wybór/selekcja modelu

W zależności od problemu, dysponując odpowiednio przygotowanymi danymi, można przystąpić do wyboru modelu uczenia maszynowego. Problem przewidywania wartości jest to zdanie regresyjne, które może być realizowane za pomocą dowolnego modelu regresyjnego:

- modelu regresji liniowej,
- modelu regresji wielomianowej,
- regresyjnej maszyny wektorów nośnych,
- regresji *k*-najbliższych sąsiadów,
- regresyjnego lasu losowego,
- Gaussian Naive Bayes,

- sztucznej sieci neuronowej,
- …

Rodzaje problemów regresyjnych:

- regresja prosta
- regresja wieloraka (system wykorzysta do prognozowania wyniku wielu cech)
- regresja wielomianowa

Regresja liniowa dla jednej zmiennej

$$\hat{y} = w_0 + w_1 x_1$$

gdzie:
$\hat{y}$ – wartość przewidywana, w_1 – nachylenie (ang. *slope),* w_0 – punkt przecięcia z osią y (ang. *intercept*) lub punkt obciążenia (ang. *bias*).

Regresja liniowa dla wielu zmiennych

$$\hat{y} = w_0 + w_1 x_1 + \ w_2 x_2 + \ \ldots + \ w_n x_n$$

$$\hat{y} = w_0 + \overrightarrow{w_n x_n}$$

gdzie:
$\hat{y}$ - prognozowana wartość, n – liczba cech, x_i - wartość i-tej cechy, w_j - j-ty parametr modelu.

Regresja wielomianowa

$$\hat{y} = w_0 + w_1 x^1 + w_2 x^2 + \ \ldots + \ w_k x^k$$

gdzie:
k – stopień wielomianu.

Wytrenowanie modelu na danych uczących

Jest to etap bezpośredniego powstawania modelu. Dane treningowe są przedstawiane w wybranym algorytmie, a algorytm na ich podstawie buduje reguły. Wytrenowany model jest uogólnieniem zależności, które można zaobserwować w prezentowanych danych.

Dostrojenie modelu/optymalizacja modelu

Wykorzystując modele uczenia maszynowego dąży się do maksymalizacji oczekiwanej użyteczności, a dokładniej, idea realizowana jest w postaci „odwróconej" jako **minimalizowanie funkcji straty**.

Aby zmierzyć wydajność modelu uczenia maszynowego, używamy funkcji kosztu. W przypadku regresji liniowej funkcja kosztu przyjmuje postać:

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

Celem trenowania modelu regresji liniowej jest znalezienie takich wartości wag, które minimalizują funkcję kosztu. Do tego często używa się algorytmów takich jak spadek gradientowy (ang. *gradient descent*). Przy metodzie gradientu prostego wagi aktualizujemy w oparciu o wyznaczoną wartość pochodnej.

$$w_{j\,new} = w_{j\,old} - \alpha \frac{\partial \sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\partial w_j}$$

W powyższym równaniu α określa wielkość kroku inaczej szybkość uczenia (ang. *learning rate*) w problemach uczenia maszynowego przy każdej iteracji podczas dążenia do minimalizacji funkcji straty i jest to parametr strojenia w algorytmie optymalizacyjnym.

Istotną konsekwencją połączenia statystyki z uczeniem maszynowym jest to, że błąd generalizacji można przedstawić jako sumę trzech różnych rodzajów błędów:

- Błąd obciążenia (ang. *bias*)
- Błąd wariancji (ang. *variance*)
- Błąd nieredukowalny (ang. *irreductible error*)

Podczas walidacji musimy wziąć pod uwagę kompromis między błędem obciążenia a wariancją (ang.*bias-variance tradeoff*). Model o wysokiej wartości obciążeni może nadmiernie upraszczać problem (niedopasowanie), podczas gdy model o wysokiej wariancji może nadmiernie komplikować problem (przeuczenie). Zmiana hiperparametrów pozwala nam znaleźć odpowiednią równowagę między błędem obciążenia, a wariancją oraz poprawić dokładność walidacji.

W celu rozwiązania problemu związanego z **przetrenowaniem modelu** stosuje się jego regularyzację (ograniczenie) – technika redukcji kary: im mniej stopni swobody, tym trudniej przetrenować model wobec danych czy też krzywe uczenia. W przypadku modelu liniowego regularyzacja jest przeważnie osiągana poprzez ograniczenie wag modelu. Chcąc zredukować wariancję w przygotowanym modelu liniowym można zastosować regularyzację:

- regresja metodą LASSO (ang. *least shrinkage and selection operator regression*) inaczej regularyzacja L_1:

$$J(w) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{m}|w_j|$$

gdzie:
λ - współczynnik regularyzacji L_1.

- regresja grzbietowa (ang. *ridge regression*) inaczej regularyzacja L_2 lub regularyzacja Tichonowa:

$$J(w) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{m} w_j^2$$

gdzie:
λ - współczynnik regularyzacji L_2.

- metoda elastycznej siatki (ang. *elastic net*), czyli L_1+L_2:

$$J(w) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{m}|w_j| + \lambda_2 \sum_{j=1}^{m} w_j^2$$

gdzie:
λ_1 - współczynnik regularyzacji L_1,
λ_2 - współczynnik regularyzacji L_2.

Większość algorytmów uczenia maszynowego wymaga określenia wartości tzw. **hiperparametrów**. Wartości hiperparametrów należy ustalić przed dopasowaniem modelu do danych. Przeszukiwanie zakresu hiperparametrów w celu wyboru najlepszego modelu można realizować za pomocą:

- własnoręcznego doboru wartości hiperparametrów,
- przeszukiwania gridowego (ang. *grid search*),
- przeszukiwania losowego (ang. *random search*),
- optymalizacji bayesowskiej (ang. *Bayesian optimization*),
- optymalizacji opartej na gradientach (ang. *Gradient-based Optimization*).

Wykorzystanie wytrenowanego modelu do prognozowania wyników dla nowych przypadków/ewaluacja modelu

Do ewaluacji otrzymanego modelu regresyjnego wykorzystuje się następujące miary/wskaźniki wydajności:

Średni błąd bezwzględny (ang. *Mean Absolute Error*, MAE):

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

gdzie:
n – liczba obserwacji, y_i - rzeczywista lub zaobserwowana wartość dla obserwacji i, $\hat{y}_i$ - prognozowana wartość modelu dla y_i.

Jest to średni błąd, którego możemy oczekiwać, jeśli użyjemy danej linii do przewidywania. Im niższy wskaźnik MAE, tym lepiej.

Błąd średniokwadratowy (ang. *Mean Square Error*, MSE):

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

Niższe MSE wskazuje, że przewidywania modelu są bliższe rzeczywistym wartościom.

Pierwiastek błędu średniokwadratowego (ang. *Root Mean Square Error*, RMSE)

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

Podobnie jak w przypadku MSE, niższe wartości RMSE wskazują lepiej dopasowaną linię – taką, która pozwala dokonywać dokładniejszych prognoz.

Wskaźniki MSE i RMSE są mniej odporne na elementy odstające niż miara MAE. Jeżeli liczba elementów wykładniczo maleje, generalnie zaleca się użycie miary RMSE.

Średni bezwzględny błąd procentowy (ang. *Mean Absolute Percentage Error*, MAPE):

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|} \times 100$$

Współczynnik determinacji, R^2- to wskaźnik, który informuje o wydajności modelu:

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

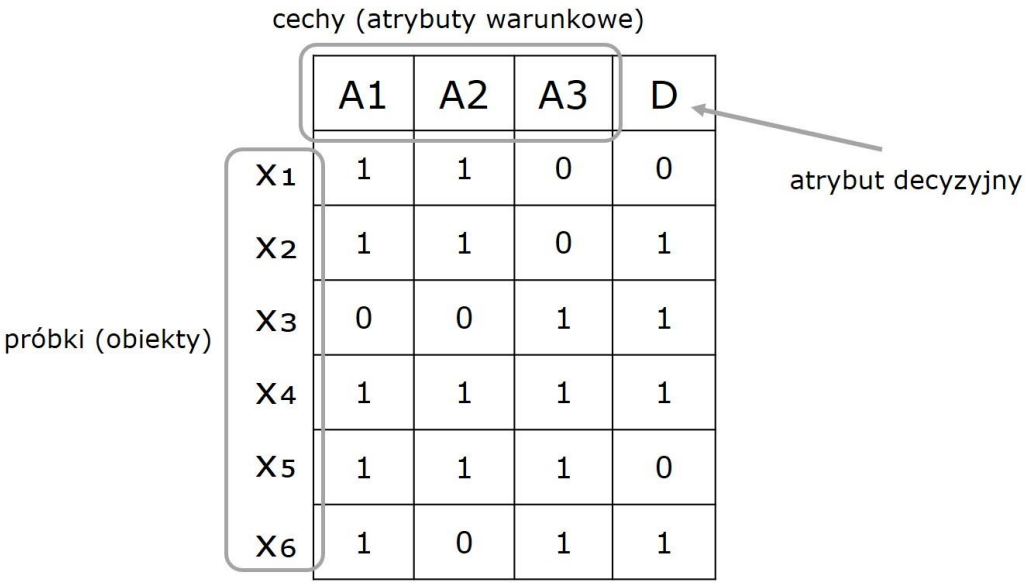
Im bliższa 1 będzie wartość R^2, tym większa liczba wariancji w wektorze docelowym jest wyjaśniona przez cechy.

Informacje dodatkowe

Reprezentacja danych w Scikit-Learn

Tablica informacyjna:

- Wiersze – poszczególne elementy zbioru danych (próbki/obiekty)
- Kolumny – wielkości odnoszące się do każdego z tych elementów (cechy/atrybuty warunkowe)



Macierz cech – dwuwymiarowa macierz o wymiarach [liczba_próbek, liczba_cech], często przechowywana w zmiennej o nazwie ***X***.

Tablica wartości docelowych (tablica etykiet) inaczej **wektor wartości docelowych** – zazwyczaj jest jednowymiarowa, jej długość jest równa liczbie próbek. Zwyczajowo oznaczana jako ***y***.

Aby użyć Scikit-Learn do regresji, należy postępować według następujących kroków:

1. Wybrać klasę modelu poprzez import odpowiedniej klasy z ScikitLearn.
2. Wybrać hiperparametry.
3. Zapisać dane w macierzy cech (*X*) i wektorze wartości docelowych (*y*).
4. Podzielić dane na zbiór treningowy i testowy.
5. Utworzyć model i dopasować model do danych. Należy wywołać metodę *fit()*.
6. Zastosowanie modelu na nowych danych. Do znalezienia etykiet dla nieznanych danych można wykorzystać metodę *predict()*, która ma za zadanie wyliczać przewidywane wartości w oparciu o macierz próbek i wektor wag.
7. Ewaluacja modelu.

Zadania wprowadzające

Należy skorzystać z funkcji *make_regression()* służącej do generowania zbiorów danych (z biblioteki *sklearn* należy zaimportować *datasets*):

- Zbiór danych składa się z 500 punktów, liczba cech = 1, rozrzut danych jest na poziomie 10, *random_state* = 101).
- Należy przygotować wykres punktowy X i y.
- Macierz cech należy zestandaryzować.
- Zbiór danych należy podzielić na zbiór treningowy i testowy, przy czym zbiór testowy ma stanowić 20% całego zbioru danych.
- Następnie należy stworzyć model regresji liniowej.
- Sprawdzić predykcję modelu do zbioru treningowego, a następnie narysować wykres punktowy *X_train* vs *y_train* wraz z krzywą regresji.
- Jakie są przewidywane wartości dla danych ze zbioru testowego? Należy narysować wykres punktowy *X_test* vs *y_test* wraz z krzywą regresji.
- Należy przygotować wykres wartości resztowych dla zbioru treningowego i testowego (na osi *x* znajdują się wartości etykiet (*y_train/y_test*, a na osi *y* znajduję się różnica *y_pred_train – y_train*,/*y_pred_test – y_test*). Czy dla większości wartości występują błędy większe/mniejsze?
- Wyznaczyć wartość MAE, MSE, RMSE, R^2 dla zbioru treningowego i testowego. Jak można ocenić działanie modelu?

Zadania

Proszę o pobranie danych ze źródła: https://www.kaggle.com/code/shreayan98c/boston-house-price-prediction/input.

Zbiór danych dotyczy cen nieruchomości w Bostonie. Celem jest przewidywanie ceny (kolumna MEDV) nieruchomości w zależności od innych cech. Zbiór danych dotyczy 506 nieruchomości opisanych za pomocą 13 cech + cena:

CRIM - współczynnik przestępczości w mieście,
ZN - odsetek "dużych działek" - powyżej 2500 m^2,
INDUS - odsetek terenów industrialnych w mieście,
CHAS - jeśli teren znajduje się przy rzece Charles -1, w pozostałych przypadkach 0,
NOX - stężenie tlenków azotu,
RM - średnia ilość pomieszczeń w budynku,
AGE - odsetek "starych budynków" - powstałych przed 1940 r.,
DIS - ważona odległość od urzędów pracy w Bostonie,
RAD - wskaźnik dostępności do głównych dróg,
TAX - wartość podatku od nieruchomości liczona od 10 tys. dolarów,
PTRATIO - stosunek liczby uczniów na nauczycieli w mieście,
B - odsetek osób pochodzenia afroamerykańskiego,
LSTAT - odsetek mieszkańców zaliczany do ubogich (odsetek ubóstwa),
MEDV - mediana wartości domów z danego terenu (w tys. dolarów).

Zadanie 1.

1.1. Import modułów (należy zaimportować klasę *LinearRegression* z biblioteki *sklearn.linear_model*).

1.2. Otwarcie pliku z danymi. Należy zaimportować nazwy kolumn i stworzyć obiekt *df*.

1.3. Sprawdzenie podstawowych statystyk.

1.4. Sprawdzenie kompletności danych.

1.5. Czy typy danych są akceptowalne?

1.6. Należy stworzyć wykresy pudełkowe dla wszystkich kolumn.

1.7. Korzystając z metody IQR, czyli rozstępu międzykwartylowego, należy wyznaczyć % wartości odstających dla każdej z kolumn. Dana wartość jest traktowana jako odstająca, gdy $x_i < Q1 - 1.5 \cdot IQR$ (wartość za mała) lub $x_i > Q3 + 1.5 \cdot IQR$ (wartość za duża).

1.8. Należy utworzyć macierz korelacji dla wszystkich cech. Następnie należy narysować wykres przedstawiający tę macierz (można skorzystać z *sns.heatmap*).

1.9. Wykres *pairplot* dla wszystkich kolumn.

1.10. Należy stworzyć obiekt *df_selected*, który będzie zawierać najbardziej skorelowane cechy o współczynniku korelacji <-0.5 oraz >0.5. Należy narysować wykres *pairplot*.

1.11. Do zmiennej *X* należy zapisać wszystkie kolumny (13) z zestawu danych (macierz cech) oprócz kolumny MEDV (można skorzystać z metody *drop* z argumentem *axis=1*).

1.12. Do zmiennej *y* należy zapisać dane z kolumny MEDV.

1.13. Do zmiennych *X_train, X_test, y_train, y_test* należy zapisać dane powstałe z podziału *X* i *y* na dane uczące i testowe (z biblioteki *sklearn.model_selection* należy zaimportować *train_test_split*). Zbiór testowy ma stanowić 20% zbioru danych, a random_state = 101).

1.14. Należy stworzyć obiekt regresji liniowej *model.* Dla obiektu *model* należy wywołać metodę *fit()*, która ma nauczyć model w jaki sposób odgadywać wartość nieruchomości w oparciu dane społeczno-gospodarcze na zbiorze treningowym.

> Wyniki obliczeń są zapisywane w atrybutach modelu. W *Scikit-Learn* wszystkie parametry, których wartości zostały ustalone poprzez wywołanie *fit*, zawierają na końcu nazw symbol podkreślenia.
>
> Np. dla regresji liniowej pojedynczej zmiennej:
>
> *model.coef_* - nachylenie,
>
> *model.intercept_* - punkt przecięcia prostej dopasowanej do zbioru danych.

1.15. Należy wyświetlić wartości współczynników dopasowania. Jak interpretować wartości poszczególnych współczynników?

1.16. W zmiennej *y_pred* należy zapisać wynik predykcji dla *X_test*.

1.17. Należy wyświetlić wykres punktowy *y_test* vs *y_pred*.

1.18. Ewaluacja modelu, czyli należy wyznaczyć wartość MAE, MSE, RMSE, R^2 (z biblioteki *sklearn.metrics* należy zaimportować odpowiednie wskaźniki wydajności).

1.19. Należy zastosować regularyzacje (z biblioteki *sklearn.linear_model* należy zaimportować *Ridge, Lasso, ElasticNet*).

1.19.1. Regresja grzbietowa (współczynnik regularyzacji, alpha = 0.5). Należy stworzyć obiekt *ridge.* Dla obiektu *ridge* należy wywołać metodę *fit().* Należy wyświetlić wartości współczynników dopasowania. W zmiennej *y_pred* należy zapisać wynik predykcji dla *X_test*. Należy przeprowadzić ewaluację modelu *ridge* jak w 1.18.

1.19.2. Regresja metodą lasso (współczynnik regularyzacji, alpha = 0.5). Należy stworzyć obiekt *lasso* i dalej postępować jak w 1.19.1.

1.19.3. Regresja elastycznej siatki (współczynnik regularyzacji, alpha = 0.5, l1_ratio = 0.5). Należy stworzyć obiekt *elastic* i dalej postępować jak w 1.19.1.

1.20. Do zmiennej *X_selected* należy zapisać kolumny 'RM', 'PTRATIO', 'LSTAT' (na podstawie analizy z zadania 1.8.-1.10.), a do zmiennej *y* należy zapisać dane z kolumny MEDV. Następnie należy wykonać kroki od 1.13 do 1.19.

Zadanie 2.

2.1. Z biblioteki *sklearn.preprocessing* należy zaimportować *StandardScaler*. Standaryzacja danych w zadaniu 2 ma być przeprowadzona na wszystkich kolumnach z pierwotnego zestawu danych (z zadania 1.11).

2.2. Należy stworzyć obiekt *scaler*, który będzie służył do standaryzacji danych:

scaler = StandardScaler()

2.3. Następnie dla obiektu *scaler* należy wywołać metodę *fit,* która dopasuje model do danych treningowych. WAŻNE !!! Trenowanie odbywa się na tylko na macierzy cech (*X_train*):

scaler.fit(X_train)

2.4. Następnie oryginalny treningowy zestaw cech zostanie przekształcony przy użyciu metody *transform()*:

scaled_X_train = scaler.transform(X_train)

2.5. Należy również przekształcić testowy zestaw cech (*X_test*) za pomocą *transform*:

scaled_X_test = scaler.transform(X_test)

2.6. Stworzenie i trenowanie modelu:

std_model.fit(scaled_X_train, y_train)

2.7. Należy wyświetlić wartości współczynników dopasowania.

2.8. W zmiennej *y_pred* należy zapisać wynik predykcji dla *scaled_X_test*.

2.9. Wykres punktowy *scaled_X_test* vs *y_test* z naniesioną krzywą regresji.

2.10. Ewaluacja modelu. Jaka jest różnica między przewidywanymi etykietami dla zbioru testowego, a rzeczywistymi wartościami? Czy uzyskane wyniki różnią się w porównaniu do danych nieprzekształconych?

2.11. Należy zastosować regularyzację jak w zadaniu 1.19.

2.12. Zbudować modele regresji liniowej, grzbietowej, lasso i elastycznej siatki dla zeskalowanych wybranych cech: RM, PTRATIO, LSTAT.

Zadanie dodatkowe

W celu poprawienia ewaluacji modeli regresyjnych można:

- usunąć wartości odstające dla MEDV, np. przy warunku, gdy MEDV >= 50.0,
- przeszukać i zastosować inne wartości współczynników regularyzacji,
- zastosować model regresji wielomianowej, itd.

Literatura

1. A. Géron, *Uczenie maszynowe z użyciem Scikit-Learn, Keras, i TensorFlow,* Wydanie III, Helion, 2023.
2. S.J. Russell, P. Norvig, *Sztuczna inteligencja. Nowe spojrzenie,* Wydanie IV, Helion, 2023.
3. A. Król-Nowak, K. Kotarba, *Podstawy uczenia maszynowego,* Wydawnictwa AGH, Kraków, 2022.
4. K. Gallatin, K. Albon, *Uczenie maszynowe w Pythonie. Receptury. Od przygotowania danych do deep learningu*, Wydanie II, Helion, 2023.
5. J. VanderPlas, *Python Data Science. Niezbędne narzędzia do pracy z danymi,* Wydanie II, Helion, 2023.

UWAGI DO SPRAWOZDANIA

Sprawozdanie zawiera rozwiązanie zadania 1 oraz zadania 2.

Materiały i metody.

Wykorzystywany zestaw danych dotyczy cen nieruchomości w Bostonie.

Do rozwiązania problemu regresyjnego przygotowane są 4 zestawy danych: nieprzetworzone, zredukowane, zestandaryzowane, zredukowane zestandaryzowane. Dla każdego zestawu danych budowane są 4 modele regresji (…).

Wyniki i dyskusja

Zestawienie wartości współczynników dopasowania oraz wskaźników MAE, MSE, RMSE, R^2 dla 4 modeli i 4 zestawów danych + dyskusja wyników. Czy redukcja cech wpłynęła na poprawę wskaźników ewaluacji?