Inteligencja obliczeniowa, przetwarzanie danych

Grzegorz Madejski

Co to za przedmiot? Skąd nazwa?

Sztuczna inteligencja: zdolność maszyny do poprawnego interpretowania danych zewnętrznych, uczenia się z nich, i używania ich do osiągnięcia określonych celów dzięki adaptacji (Kaplan, Haenlein); ogólniej: dziedzina informatyki zajmująca się technikami wykorzystującymi powyższą zdolność

Inteligencja obliczeniowa: poddziedzina sztucznej inteligencji skupiająca się na adapcyjnych algorytmach, które umożliwiają lub usprawniają inteligentne zachowanie w złożonym lub zmieniającym się środowisku (Engelbrecht)

Zakres inteligencji obliczeniowej

Trzy filary:

- Sztuczne sieci neuronowe
- Bioinspirowane algorytmy metaheurystyczne:
  - Algorytmy ewolucyjne i genetyczne
  - Inteligencja roju i algorytmy mrówkowe
- Logika rozmyta

Celem dzisiejszego wykładu jest przegląd technik stosowanych do przygotowania zbioru danych do treningu modelu (do klasyfikacji lub regresji).

Taki preprocessing danych może się składać z kilku kroków, które po kolei omówimy.

Techniki omówione na tym wykładzie to zaledwie ułamek wszystkich stosowanych technik przetwarzania danych. Zachęcam wszystkich do rozeznania się, czy istnieją inne ciekawe algorytmy.

Praca w zakresie sztucznej inteligencji to w 80% przygotowywanie danych, and 19% narzekanie na przygotowywanie danych i 1% optymalizacja algorytmów uczących.

Krok I: Usuwanie błędnych danych

Krok II: Usuwanie brakujących danych

Krok III: Optymalizowanie struktury zbioru danych

Krok IV: Normalizacja danych

Krok I: Usuwanie błędnych danych

- Różne modele AI potrzebują danych do trenowania. Dane są często dostępne w formie datasetów – prostych zbiorów danych, najczęściej w formie tabelki csv.
- Przykłady: pomiary kwiatów (irysów) oraz diagnoza cukrzycy na podstawie parametrów medycznych.

	sepal length	sepal width	petal length	petal width	target
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Krok I: Usuwanie błędnych danych

- Przykłady: pomiary kwiatów (irysów) oraz diagnoza cukrzycy na podstawie parametrów medycznych.

```
data.head(10)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

Krok I: Usuwanie błędnych danych

- Nasza baza danych może być strukturalnie błędna (np. nierówne wymiary kolumn lub wierszy, brak cudzysłowów ograniczających teksty, brak średników pomiędzy danymi), co uniemożliwia nam jej wczytanie do Pythona. Takie błędy należy poprawić na samym początku.
- Następnie warto sprawdzić czy kolumny są dobrze sformatowane i mają dane tego samego typu. Bywa, że w obrębie jednej kolumny są pomieszane teksty i liczby.
- Uwaga! Czasem w kolumnie liczbowej mogą pojawić się napisy „n/a” lub podobne, sugerujące brak danych. O nich powiemy dalej.

Krok I: Usuwanie błędnych danych

- Następnie warto zastanowić się, czy w obrębie danych występują błędy.
- W przypadku danych kategorycznych (jakościowych) niektóre wartości mogą nie wchodzić w skład domyślnego zakresu, np. kolumna z wynikiem testu na chorobę zawiera wartości negative, positive, ale jeden rekord danych zawiera test_positive. Wówczas oczywiście dane poprawiamy.
- Bywa jednak, że dane są niejasne i nie wiemy co oznaczają. Jeśli dla powyższego przykładu pojawiłaby się wartość done. To nie wiemy nic o diagnozie. Wówczas dobrym pomysłem jest zamiana tych danych na brak danych (np. n/a).

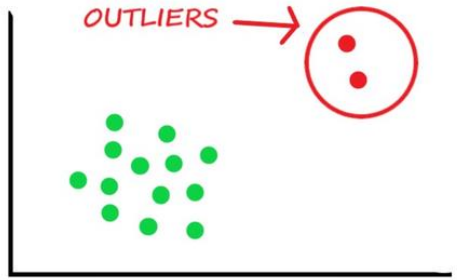
Krok I: Usuwanie błędnych danych

Ptaki lecące prędkością nadświetlną?

W datasecie wszystko jest możliwe.
Zdarzyło mi się ☺

Krok I: Usuwanie błędnych danych

- Jeszcze więcej problemów sprawiają dane liczbowe.
- Jeśli długość płatka irysa wynosi -0.5 to wiemy, że mamy do czynienia z błędem, i taką liczbę również można podmienić na n/a.
- Jeśli wiemy, że irysy mają płatki w granicach 10 cm, to gdy pojawi się liczba 30, to również klasyfikuje się ona jako błąd do usunięcia.
- Nie zawsze jednak jest jasne, jaki przedział danych liczbowych jest dobry. Są pewne techniki, które usuwają dane odstające (ang. outliers). W wielu przypadkach dane te są błędne i można je wykluczyć ze zbioru danych.

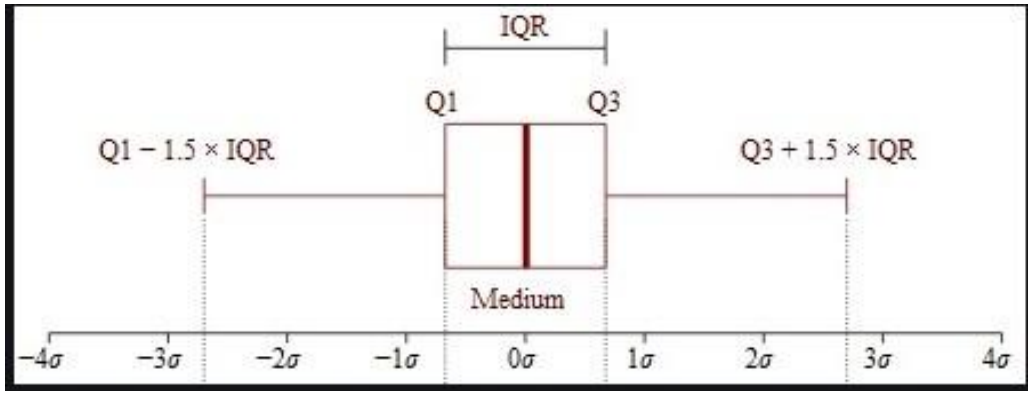


Krok I: Usuwanie błędnych danych

- Jak zidentyfikować obserwacje odstające do ich usunięcia? Są na to różne sposoby.
- Można wykorzystać rozstęp ćwiartkowy czy międzykwartylowy (interquartile range, IQR), czyli odległość między pierwszym a trzecim kwartylem. Następnie wykorzystać go do obliczenia przedziału „dobrych” danych.

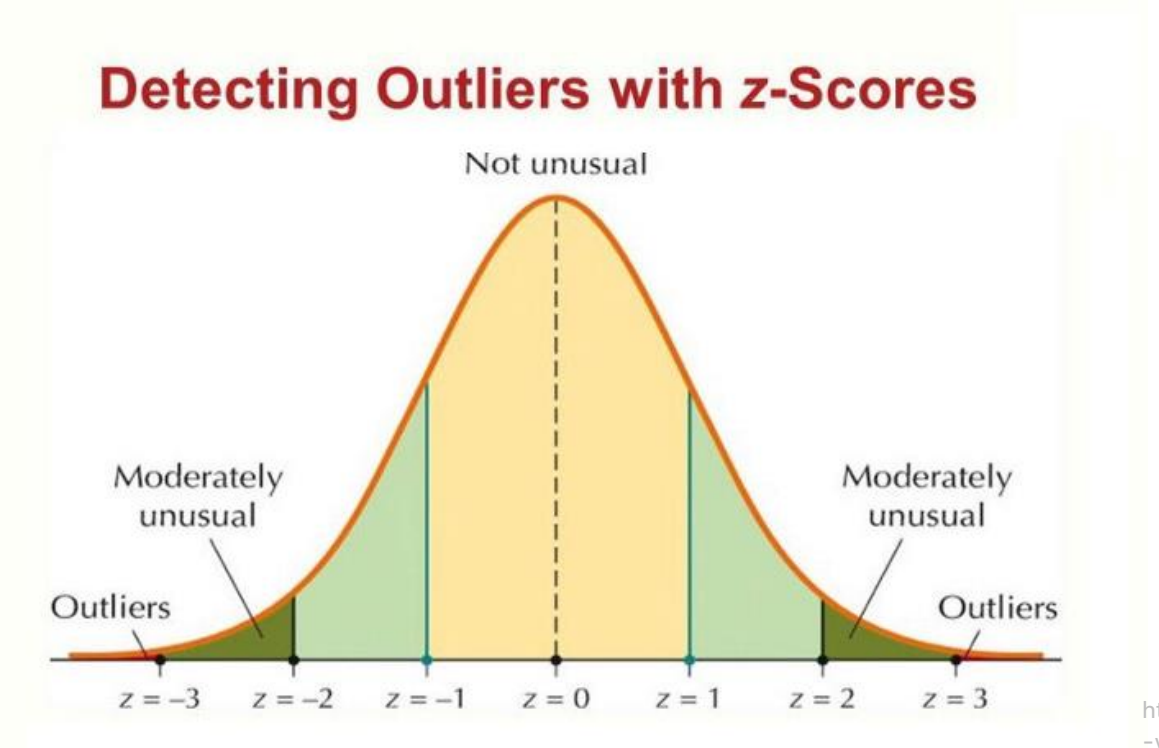
$$IQR = Q3 - Q1$$

$$PrzedziałDobrychDanych = [\ Q1-1.5*IQR,\ Q3+1.5*IQR]$$



Krok I: Usuwanie błędnych danych

- Jak zidentyfikować obserwacje odstające do ich usunięcia? Są na to różne sposoby.
- Drugi sposób to obliczanie tzw. z-score (standardowy wynik). Jest to stopień odchylenia wartości x od średniej i dany jest wzorem z = (x – średnia)/odch.stand.



https://www.analyticsvidhya.com/blog/2022/08/dealing-with-outliers-using-the-z-score-method/

Krok I: Usuwanie błędnych danych

Krok II: Usuwanie brakujących danych

Krok III: Optymalizowanie struktury zbioru danych

Krok IV: Normalizacja danych

Krok II: Usuwanie brakujących danych

- Brakujące dane (ang. missing data/values) to bardzo częsty problem w bazach danych.
- Braki danych mogą wynikać z usterek bazy danych, lub ze sposobu pobierania danych (np. ankietowani nie zawsze wszystko wypełnią).
- Gdy baza danych jest bardziej pusta niż pełna, trzeba się zastanowić, czy w ogóle wystarczy do naszych badań.
- Brakujące dane mogą być oznaczone jako: puste komórki, białe znaki, symbol „?”, „-” lub „--”, napis „b/d” (brak danych), „n/a” lub „NA” (not available), „NaN” (not a number) lub inne.
- Są dwa sposoby na pozbycie się brakujących danych...

Krok II: Usuwanie brakujących danych

- (Deletion) Brakujące dane można usunąć wraz z całym wierszem lub kolumną, w których się znajdują. Ma to sens, gdy dany wiersz lub kolumna są w większości puste (i być może też mało ważne). Ale uwaga, zbyt dużo usuwania może spowodować poważne straty informacji. Baza może stać się za mała do dalszych badań.
- (Imputation) Brakujące dane można też zastępować nowymi wygenerowanymi danymi (przypisanie = imputacja). Co prawda są to dane „fałszywe”, ale ratują inne dane przed usunięciem i na ogół nie wpływają bardzo na wydajność algorytmów uczących się ze zbioru.

Krok II: Usuwanie brakujących danych

Jakie dane można wstawić w puste komórki bazy danych? Jest wiele technik imputacji, między innymi:

- Wstaw w pustą komórkę średnią lub medianę dla danej zmiennej (kolumny). Działa dla kolumn numerycznych. Szybkie, ale dość prymitywne.
- Wstaw w pustą komórkę dominantę (wartość najczęstszą) dla danej kolumny. Dobre dla kolumn z wartościami kategorycznym. Szybkie i prymitywne.
- Wstaw w pustą komórkę średnią/medianę/dominantę dla danej zmiennej (kolumny), ale uwzględniając tylko próbki/wiersze z tą samą klasą, co edytowany wiersz. Trochę mądrzejsze, bo zwraca uwagę na inne dane.
- Wstaw w pustą komórkę taką wartość, jaką wyznacza k najbliższych sąsiadów badanego rekordu (odległości po reszcie kolumn). Dość mądra technika, bo analizuje inne kolumny. Przez to niestety wolna.
- Inne zaawansowane techniki: alg. uczące, deep learning, interpolacja.

Krok II: Usuwanie brakujących danych

Przydatne linki:

- https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779
- https://realpython.com/python-data-cleaning-numpy-pandas/
- https://towardsdatascience.com/data-cleaning-with-python-and-pandas-detecting-missing-values-3e9c6ebcf78b
- https://www.freecodecamp.org/news/how-to-handle-missing-data-in-a-dataset/

Krok I: Usuwanie błędnych danych

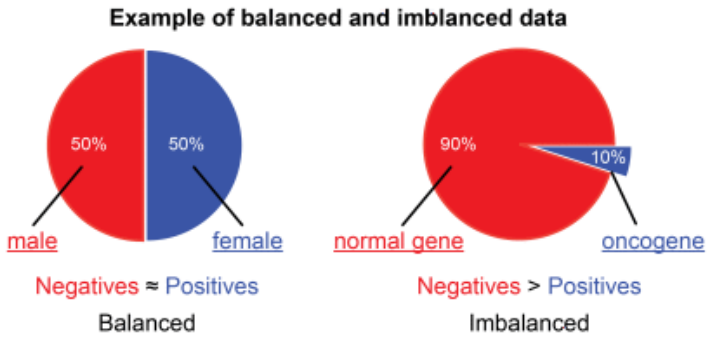
Krok II: Usuwanie brakujących danych

Krok III: Optymalizowanie struktury zbioru danych

Krok IV: Normalizacja danych

Krok III: Optymalizowanie struktury datasetu

- Struktura datasetu może nie być wygodna do uczenia maszynowego
- Niezrównoważony zbiór danych (imbalanced data) to dość częsty problem, w którym liczba próbek zmiennej z jedną wartością może być znacznie większa niż liczba próbek z drugą wartością. Z powodu tego niezrównoważenia podczas uczenia model algorytmu może w ogóle nie uczyć się danych rzadko występujących w bazie danych. Jest to szczególnie niebezpieczne gdy niezbalansowana jest kolumna z klasą.
- Być może widać było to przy okazji diabetes.csv na poprzednich laboratoriach. Liczba osób chorych jest znacznie mniejsza niż liczba osób zdrowych. Ciężko nauczyć się rozpoznawać chore osoby.



Krok III: Optymalizowanie struktury datasetu

- Jak uczyć na niezbalansowanym zbiorze danych?
- Jeśli dysproporcje pomiędzy wartościami klas są znaczne np. 1:10, 1:100 czy 1:1000, to warto je zbalansować. Przy łagodnym niezrównoważeniu np. 1:2, 2:3 nie warto tego robić.
- Najłatwiejszą techniką jest wyrównywanie liczebności obu wartości poprzez:
  - usuwanie nadmiarowych próbek z klasy bardziej licznej (tzw. under-sampling)
  - dodawanie kopii próbek z klasy mniej licznej (tzw. over-sampling)
- Inną techniką jest generowanie syntetycznych próbek. Można spróbować do tego wykorzystać dedykowany algorytm np. SMOTE.
- Dodatkowe info: https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/ , https://elitedatascience.com/imbalanced-classes , https://www.analyticsvidhya.com/blog/2022/05/handling-imbalanced-data-with-imbalance-learn-in-python/

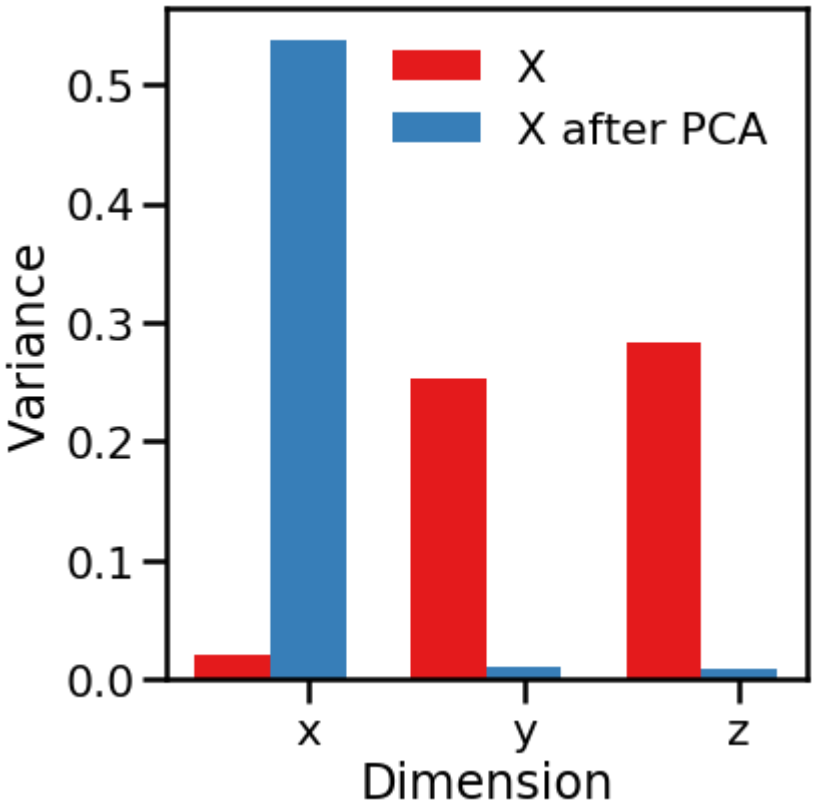
Krok III: Optymalizowanie struktury datasetu

- Kolejnym problemem z datasetem może być jego rozmiar.
- Bywa, że baza danych ma mnóstwo kolumn i przetwarzanie jej na maszynie jest bardzo wolne.
- Co jeszcze gorsze, bywa, że kolumny mają bardzo rozproszone dane (ang. sparse data) np. bardzo dużo nic nieznaczących zer i niewiele ważnych jedynek. Jak temu zaradzić?
- https://www.kdnuggets.com/2023/04/best-machine-learning-model-sparse-data.html

Users					Movies						Target
A	B	C	D	E	Parasite	Joker	Avengers	Spotlight	The Great Beauty	There will be blood	Rating
1	0	0	0	0	1	0	0	0	0	0	5
1	0	0	0	0	0	1	0	0	0	0	4
1	0	0	0	0	0	0	1	0	0	0	4
0	1	0	0	0	1	0	0	0	1	0	2
0	1	0	0	0	0	0	0	1	0	0	4
0	1	0	0	0	0	0	0	0	1	0	3
0	0	1	0	0	0	0	1	0	0	0	5
0	0	0	1	0	0	0	0	0	0	1	4
0	0	0	0	1	0	0	1	0	0	0	4

Krok III: Optymalizowanie struktury datasetu

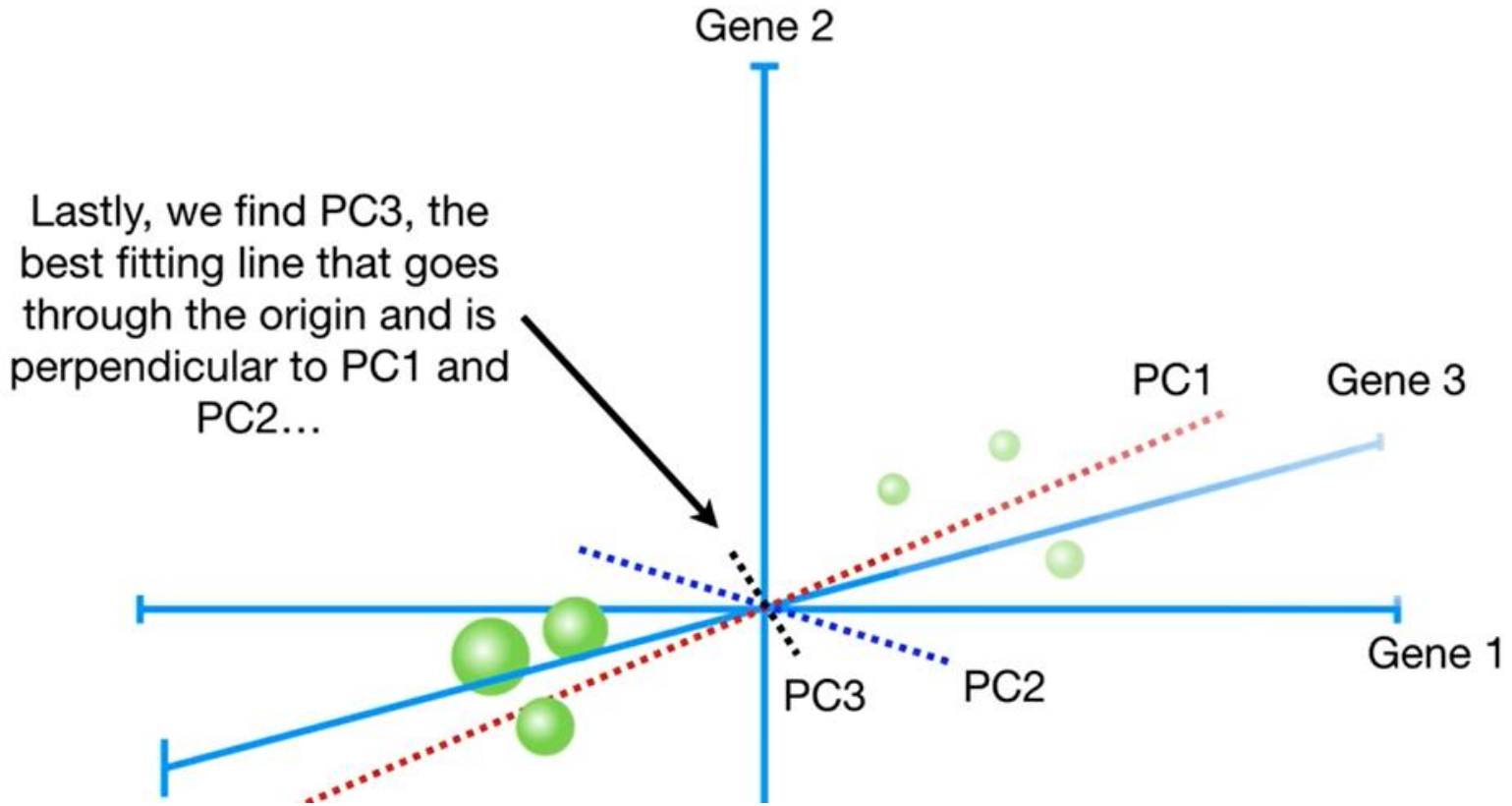
- Jeśli kolumna ma bardzo mało danych można ją usunąć.
- Inną techniką jest kompresja informacji w kolumnach za pomocą algorytmu PCA (principal component analysis, analiza głównych składowych).
- Algorytm ten, za pomocą sprytnych technik statystyczno-algebraicznych, kondensuje informacje z bazy danych w nowo utworzonych zmiennych/kolumnach. Najpierw „upycha” jak najwięcej informacji w pierwszej nowej kolumnie. To co się nie udało, upycha do drugiej, resztki do trzeciej i tak dalej. W konsekwencji, pierwsze kolumny zawierają najwięcej informacji, a ostatnie najmniej. Ostatnie kolumny można usunąć z bazy danych bez dużej straty informacji.
- Linki: https://marcioodwyer.wordpress.com/2019/01/09/principal-component-analysis-a-toy-example/ , https://www.kaggle.com/code/shrutimechlearn/step-by-step-pca-with-iris-dataset



Krok III: Optymalizowanie struktury datasetu

Bardzo fajny film wyjaśniający PCA:

https://www.youtube.com/watch?v=FgakZw6K1QQ&ab_channel=StatQuestwithJoshStarmer



Krok III: Optymalizowanie struktury datasetu

```
from sklearn import datasets
from sklearn.decomposition import PCA
import pandas as pd

iris = datasets.load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_names)
y = pd.Series(iris.target, name='FlowerType')
print(X.head())



pca_iris = PCA(n_components=2).fit(iris.data)
print(pca_iris)
print(pca_iris.explained_variance_ratio_)
print(pca_iris.components_)
print(pca_iris.transform(iris.data))
```

Uruchomimy i przeanalizujemy fragment kodu po lewej.

Krok I: Usuwanie błędnych danych

Krok II: Usuwanie brakujących danych

Krok III: Optymalizowanie struktury zbioru danych

Krok IV: Normalizacja danych

Krok IV: Normalizacja danych

By algorytmy lepiej uczyły się na danych, dobrym pomysłem jest ich normalizacja. Są tutaj dwa standardowe wzory:

- Skalowanie, normalizacja min-max (ang. scale, min-max scaling) ściska dane z kolumny do przedziału [0, 1]. Dla kolumny obliczamy max i min, następnie każdą wartość kolumny x, zamieniamy na y wg wzoru:

$$y = \frac{x - min}{max - min}$$

- Skalowanie jest bardzo ważne i z reguły poprawia stabilność uczenia i skuteczność algorytmów. Kolumny/zmienne o różnych zakresach dostają ten sam zakres i tę samą „ważność”.
- Uwaga: dane mogą być zgniecione przez za duże wartości!

Krok IV: Normalizacja danych

By algorytmy lepiej uczyły się na danych, dobrym pomysłem jest ich normalizacja. Są tutaj dwa standardowe wzory:

- Standaryzacja, normalizacja z-score (ang. z-score normalization, standarization) ściska dane z kolumny do tak, aby przeciętnie były oddalone od 0 o 1. Innymi słowy dane przyjmują rozkład normalny. Dla kolumny obliczamy średnią mean i odchylenie standardowe sd, a następnie każdą wartość kolumny x, zamieniamy na y wg wzoru:

$$y = \frac{x - mean}{sd}$$

- Standaryzacja jest bardziej radykalna niż skalowanie, bo zmienia rozkład danych (kształt dystrybucji). Jest ważna dla niektórych algorytmów np. LDA.

A co ze zdjęciami?

A co ze zdjęciami?

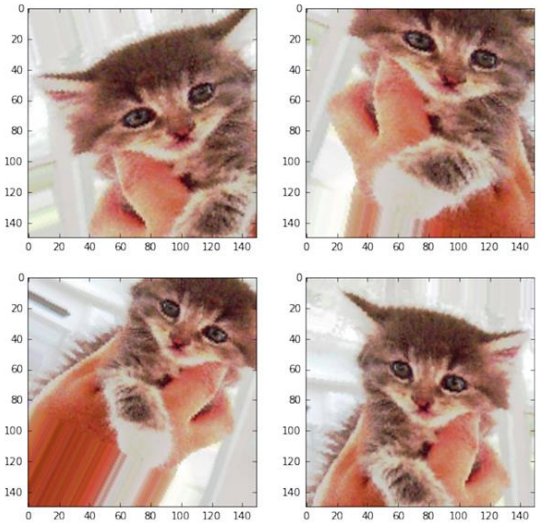
Bazy danych ze zdjęciami traktowane są trochę inaczej niż te numeryczno-tekstowe. Tylko niektóre techniki wymienione wyżej mają tu zastosowanie:

- Poprawianie niezrównoważonych danych (imbalanced data). Jeśli obrazków otagowanych różnymi klasami nie jest mniej więcej tyle samo.
- Normalizacja: skalowanie. Częstą taktyką jest skalowanie zakresu barwnego pikseli z [0, 255] do [0, 1]. Pomaga to w nauce konwolucyjnych sieci neuronowych.
- Bazy numeryczne były kompresowane za pomocą PCA. Bazy obrazkowe kompresuje się poprzez zmniejszenie wymiarowości obrazków.
- Jeśli wyuczanie trzykanałowych (RGB) kolorowych obrazków jest za trudne lub baza jest za duża, można też rozważyć konwersje obrazków do skali szarości (jeden kanał).

A co ze zdjęciami?

Dodatkowymi technikami stosowanymi na bazach danych ze zdjęciami jest augmentacja obrazów (image augmentation). Transformujemy zdjęcie za pomocą różnych operacji, by uzyskać więcej zdjęć z tym samym obiektem, ale nieco zmienionym. Dzięki temu algorytmy uczące staną się bardziej wszechstronne w wykrywaniu obiektu na zdjęciu. Technikami stosowanymi do augmentacji są:

- Rotacja
- Skalowanie
- Przesunięcie
- Przechylenie
- Zaszumienie
- Odbicie pionowe i poziome.



Linki: https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/ https://www.analyticsvidhya.com/blog/2022/09/training-a-cnn-from-scratch-using-data-augmentation/