Zbiór treningowy, walidacyjny i testowy
Train vs validation vs test sample

Zbiór treningowy
Jest to zbiór danych, na którym trenujemy model.

Zbiór walidacyjny
Jest to próbka danych używana do zapewnienia bezstronnej oceny dopasowania modelu wytrenowanego na zbiorze treningowym. Często jest również wykorzystywana do dostrajania hiperparametrów modelu.

Zbiór testowy
Jest to próbka danych używana do zapewnienia bezstronnej oceny ostatecznego dopasowania modelu do zestawu danych szkoleniowych. Schowaj ją do szuflady i nie wyciągaj, dopóki nie będziesz mieć pewności, że masz już gotowy model.

Przykład z życia
Wyobraź sobie, że uczysz się do matury.
Książki i zeszyty są Twoim **zbiorem treningowym**, który umożliwia zdobywanie wiedzy i naukę.
Zbiorem walidacyjnym są próbne testy maturalne, dzięki którym możesz ocenić przygotowanie do egzaminu. Jeśli widzisz, że brakuje Ci wiedzy z jakiegoś zakresu, to masz jeszcze czas, aby to nadrobić.
Matura będzie ostatecznym sprawdzianem i jest ona właśnie **zbiorem testowym**, który weryfikuje przygotowanie do matury.

Widzenie komputerowe
Computer vision

Termin **widzenie komputerowe** składa się z dwóch pojęć:

- *komputer* - maszyna, która jest zdolna do wykonywania obliczeń, procesów i operacji zgodnie z instrukcją dostarczoną przez program lub sprzęt,
- *widzenie* - definiowane jako zrozumienie środowiska poprzez zobaczenie w tym środowisku obiektów.

Łącząc te dwa terminy można powiedzieć, że widzenie komputerowe to proces, w którym maszyna lub system generuje zrozumienie informacji wizualnych.

Zatem poprzez **computer vision** rozumiemy nie tylko zdolność widzenia obrazów (wzroku), ale naśladowanie percepcji – zdolności ludzi do zrozumienia tego, co widzą.

Wartość F1
F1 score

Wartość F1 jest wyliczana jako średnia harmoniczna Precision i Recall, w związku z tym określana jest jako kompromis pomiędzy tymi dwoma metrykami.

Jeśli optymalizujesz jedną z metryk, robisz to kosztem drugiej i średnia harmoniczna zaczyna szybko spadać. Natomiast wynik jest największy, gdy zarówno Precision, jak i Recall są sobie równe. Innymi słowy średnia harmoniczna bardziej "karze", gdy jeden z wyników jest zły (bliższy wartości 0).

F1 przyjmuje wartości od 0 (bardzo zły model) do 1 (idealnie rozpoznaje).

$$F_1 - score = \left(\frac{Recall^{-1} + Precision^{-1}}{2}\right) = 2 \cdot \frac{(Recall \cdot Precision)}{(Recall + Precision)}$$

Kiedy? F1 jest przede wszystkim używany, gdy masz niezbalansowane klasy. Dodatkowo możesz używać go, gdy kluczowe są False Negatives (FN) i False Positives (FP).
Jeśli chcemy wskazać, która z metryk jest dla nas istotniejsza: czy recall czy precision można użyć bardziej ogólnej metryki Fβ:

$$F_\beta = (1 + \beta^2) \cdot \frac{Recall \cdot Precision}{Recall + (\beta^2 \cdot Precision)}$$

Jeżeli obie miary są ważne to podstawiamy za β=1, jeżeli bardziej zależy nam na Precission to β powinna być z przedziału [0,1], zaś w przeciwnym przypadku, gdy bardziej zależy na Recall to β>1.

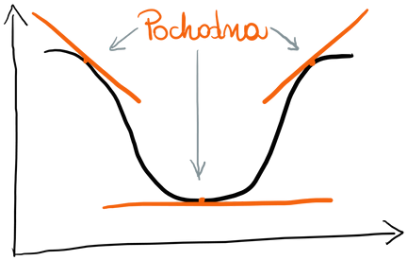
Gradient i pochodna
Gradient and derivative

Gradient jest wektorem, którego zwrot wskazuje kierunek najszybszego wzrostu wartości funkcji, a którego długość odpowiada wzrostowi wartości tej funkcji na jednostkę długości.

$$\nabla_x f(x)$$

Inaczej mówiąc, gradient funkcji f jest wektorem zawierającym wszystkie jej pochodne kierunkowe w pewnym punkcie x.

Pochodna kierunkowa funkcji (x,y) to wartość, która reprezentuje tempo zmian w kierunku wektora jednostkowego u.
Pojęcie pochodnej można interpretować jako tempo zmian f(x,y), które można traktować jako nachylenie funkcji w punkcie (x0,y0).



Przykład. Wyobraź sobie, że w jednym garnku gotujesz zupę dla 300 osób. Aby zupa się zmieściła garnek musi być olbrzymi. We wnętrzu tak wielkiego naczynia temperatura rozkłada się nierównomiernie – gdzieś będzie cieplej a gdzieś zimniej. Na szczęście masz pod ręką funkcję T(x,y,z), która mówi o temperaturze w garnku. Zmienne x, y, z to konkretny punkt w przestrzeni, w którym ta funkcja zwraca nam temperaturę. Wówczas gradient funkcji T wskazywałby nam kierunek, gdzie temperatura jest najmniejsza.

Kurtoza
Kurtosis

Kurtoza jest miarą występowania wartości odstających.

Kurtoza z próby wyraża się wzorem:

$$Kurt = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^4}{\sigma^4} - 3$$

gdzie:
x_i - i-ta wartość cechy
μ - wartość oczekiwana w populacji
σ - odchylenie standardowe
n - liczebność próby

Kurtoza jest średnią (lub wartością oczekiwaną) standaryzowanych danych podniesionych do czwartej potęgi.

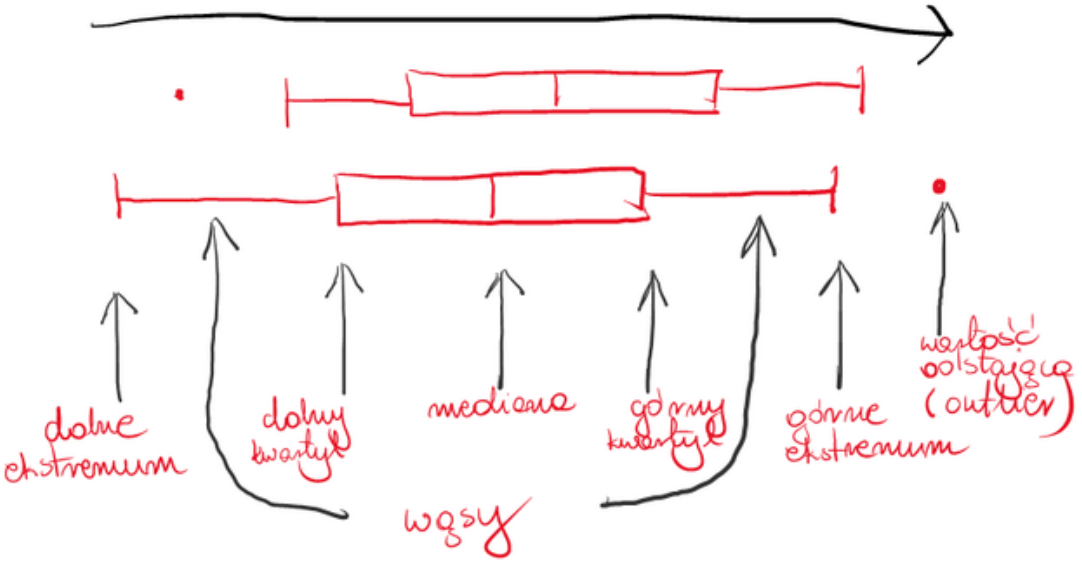
Wszelkie standaryzowane wartości, które są mniejsze niż 1 (czyli są w obrębie jednego odchylenia standardowego od średniej) praktycznie nie przyczyniają się do kurtozy. Wynika to z tego, że podniesienie liczby mniejszej niż 1 do czwartej potęgi sprawia, że będzie bliżej zera.

Jedynymi wartościami danych, które przyczyniają się do kurtozy w jakikolwiek znaczący sposób, są dane znajdujące się poza regionem odchylenia standardowego, czyli wartości odstające.

Wykres pudełkowy
Box plot

Wykres pudełkowy (skrzynkowy) tworzy się rysując prostokąt (pudełko, skrzynkę). Jego lewy lub dolny bok (zależy czy wykres prezentujemy w postaci pionowej lub poziomej) wyznaczony jest przez pierwszy kwartyl, zaś przeciwny bok przez trzeci kwartyl. Wewnątrz prostokąta znajduje się linia określająca medianę.

Linie rozciągające się równolegle od skrzynek nazywane są „wąsami" i służą do wskazania zmienności poza dolnym i górnym kwartylem. Standardowo wąsy mają długość półtorej wartości rozstępu ćwiartkowego.



Chociaż wykresy pudełkowe mogą wydawać się prymitywne w porównaniu do histogramu lub wykresu gęstości, to ich zaletą jest, że zajmują mniej miejsca. Jest to przydatne przy porównywaniu rozkładów między wieloma grupami lub zestawami danych.

Wykres bąbelkowy
Bubble chart

Wykres bąbelkowy to wykres rozrzutu rozbudowany o kolejny (trzeci) wymiar, którym jest rozmiar kropek. Podobnie jak wykres rozrzutu jest osadzony w kartezjańskim układzie współrzędnych.

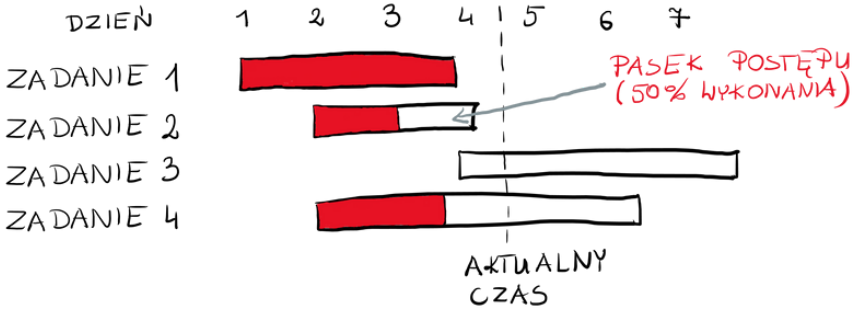
Mając trzy zmienne numeryczne można je zaprezentować w taki sposób, że pierwsza wartość będzie na osi X, druga zmienna na osi Y, a trzecia będzie rozmiarem kropki. Ten trzeci wymiar reprezentowany jest na podstawie pola koła. Dodatkowo można jeszcze wykorzystać różne kolory do rozróżniania kategorii lub reprezentowania dodatkowej zmiennej danych.

Wykres bąbelkowy najczęściej używany jest do pokazywania relacji między zmiennymi. Łatwo na nich odnaleźć wszelkie wzorce wśród danych.

Uwaga!
Jest duża różnica, w jakiej kolejności i jak przedstawione zostaną prezentowane zmienne. Dlatego trzeba świadomie wybrać ich kolejność i wiedzieć co chcemy pokazać i jaką historię opowiedzieć na wykresie.

Wykres Gantta
Gantt Chart

Wykres Gantta wyświetla listę działań (lub zadań) wraz z ich długością trwania w czasie. W prosty sposób można odczytać, kiedy zaczyna się i kończy każde działanie. Dodatkowo od razu można zobaczyć na nim równolegle wykonane zadania. Z tego względu wykres ten jest przede wszystkim używany jako narzędzie do zarządzania projektami.



Wykres Gantta to rodzaj wykresu kolumnowego używanego do zilustrowania planów i harmonogramów.

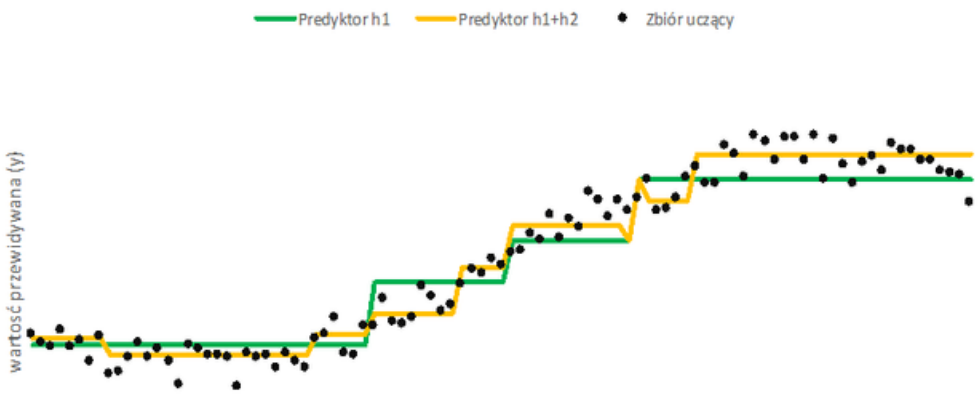
Wiersze przedstawiają działania, a kolumny są użyte jako skala czasu. Czas trwania każdej czynności jest reprezentowany przez długość słupka wykreślonego w skali czasu. Początek słupka to początek czynności, a jego koniec to moment, w którym czynność powinna się zakończyć.

Oznaczanie pasków kolorami umożliwia podzielenie działań na grupy. Aby pokazać procent ukończenia czynności, słupek można częściowo wypełnić, zacieniować lub użyć innego koloru, aby odróżnić to, co zostało zrobione, od tego, co pozostało do zrobienia.

Przyspieszenie
Boosting

Boosting to metoda tworzenia modelu polegająca na tworzeniu serii słabszych modeli (weak learners), którego zadaniem jest poprawienie błędów wcześniejszego modelu.

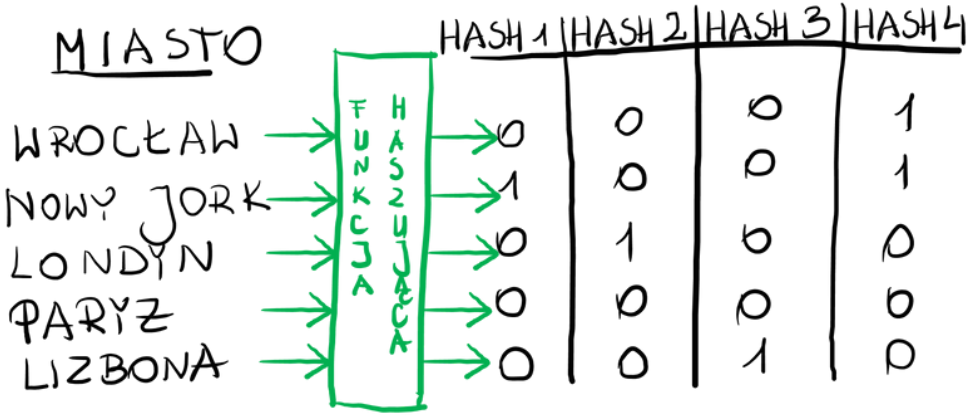
Przykład. W pierwszym kroku budujemy proste drzewo decyzyjne. Następnie, mając informacje o działaniu tego drzewa, budujemy kolejne drzewko, które próbuje naprawić błąd tego wcześniejszego. Naprawiamy błędy wcześniejszego drzewa, czyli wyliczamy reszty (wartość rzeczywista minus przewidywania pierwszego drzewa) i dla nich budujemy kolejne drzewo. I tak dalej. Uruchamiamy kolejne iteracje działające w taki sam sposób, tzn. próbując naprawiać wszystkie wcześniejsze decyzje.



Kodowanie skrótu
Hash labeling

Funkcja haszująca (lub funkcja skrótu) przyporządkowuje dowolnie dużej wartości krótszą wartość o stałym rozmiarze. Jest bardzo często wykorzystywana w kryptografii.

Niestety, jak chcemy zakodować więcej danych w mniejszej liczbie danych, to mogą wystąpić tzw. kolizje, czyli jedna wartość hashująca może odpowiadać np. dwóm wartościom.
Zatem hash encoding jest stworzony na kształt działania funkcji hashującej. Hash encoding to inaczej One-Hot encoding, ale z określoną długością. Oczywiście jeśli liczba możliwych "skrotów" będzie odpowiadać liczbie unikalnych wartości, to będzie działać dokładnie jak One-Hot encoding.



Analiza głównych składowych (PCA)
Principal Component Analysis (PCA)

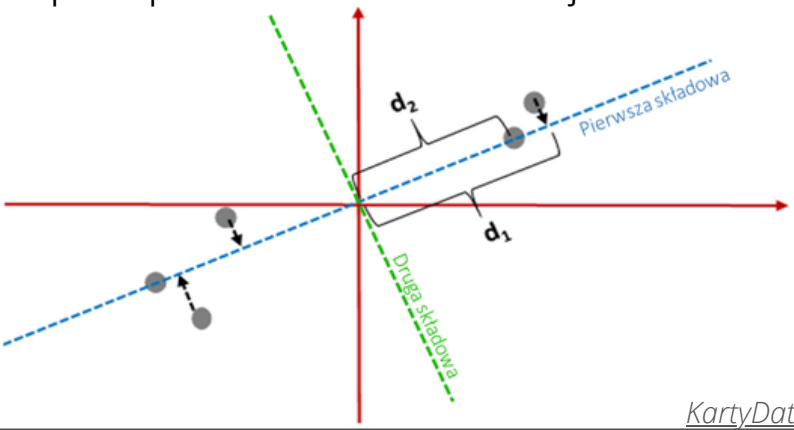
Analiza składowych głównych to najbardziej popularny algorytm redukcji wymiarów. W dużym skrócie polega na rzutowaniu danych do przestrzeni o mniejszej liczbie wymiarów tak, aby jak najlepiej zachować strukturę danych.

Służy przede wszystkim do redukcji zmiennych opisujących dane zjawisko oraz odkrycia ewentualnych prawidłowości między cechami. Dokładna analiza składowych głównych umożliwia wskazanie tych zmiennych początkowych, które mają duży wpływ na wygląd poszczególnych składowych głównych, czyli tych, które tworzą grupę jednorodną.

Sposób działania:

Krok 1. Przesuwamy układ współrzędnych w „środek danych", aby można było szukać najlepszej płaszczyzny do zrzutowania danych.

Krok 2. Szukamy krzywych (płaszczyzn), które maksymalizują sumę wszystkich kwadratów odległości punktów od początku układu współrzędnych. Najpierw PC1, potem PC2 prostopadłe do PC1, potem PC3 prostopadłe do PC1 i PC2 i tak dalej.

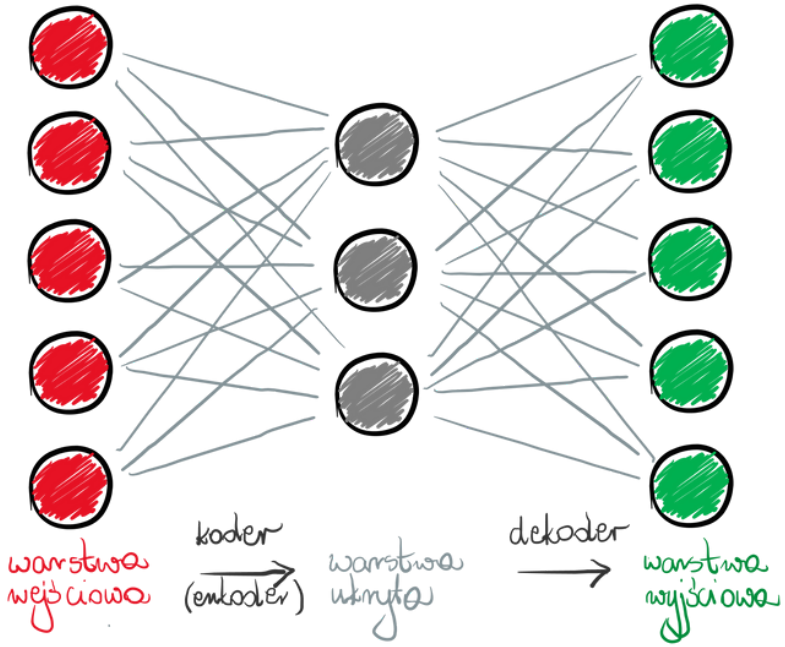


Autoenkodery
Autoencoders

Autoenkoderami nazywamy sieci neuronowe zdolne do uczenia się reprezentacji danych wejściowych (kodowań).

Autoenkoder zawsze składa się z dwóch elementów:

- kodera (ang. encoder), który przekształca dane wejściowe do postaci reprezentacji,
- dekodera (ang. decoder), który przekształca dane z postaci reprezentacji do danych wyjściowych



Inaczej mówiąc autoenkodery patrzą na dane wejściowe, przekształcają je w pewną reprezentację wewnętrzną, a potem odtwarzają wynik przypominający dane otrzymane na wejściu.

Jak radzić sobie z przetrenowaniem sieci

Methods to avoid overfitting neutral networks

Sieci neuronowe są bardzo potężnymi algorytmami, które bardzo prosto potrafią się dopasować do danych. Aby sobie poradzić z przetrenowaniem można zastosować poniższe metody:

1) Więcej danych do treningu

Jeśli mamy bardzo dużo danych to najprostszym sposobem jest po prostu trenowanie sieci na większej liczbie danych. Dla przykładu jeśli dograliśmy informacje na podstawie ostatniego kwartału wówczas okres można rozszerzyć do całego roku.

2) Metoda porzucania (dropout)

Polega na losowym ustawieniu wychodzących krawędzi ukrytych jednostek (neuronów tworzących ukryte warstwy) na 0 przy każdej aktualizacji fazy treningu.

3) Regularyzacja wag

Kolejną techniką do walki z przeuczeniem jest dokładanie do warstw kar, aby ich wartości były mniejsze. Te metody nazywamy regularyzacjami wag.

4) Metoda wczesnego zakończenia (early stopping)

Polega na tym, aby zakończyć uczenie, gdy strata na zbiorze testowym nie rośnie. Czyli trenując dalej sieć nie poprawiamy mocy modelu na zbiorze testowym.

5) Upraszczanie architektury

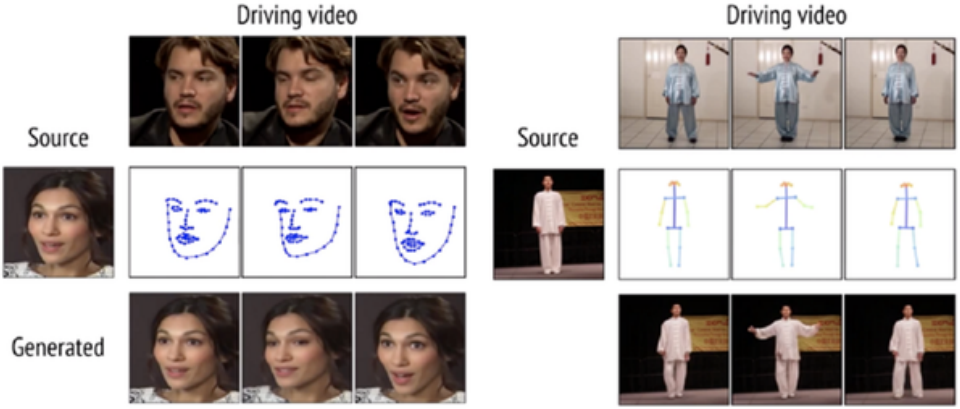
Jeśli skomplikowaliśmy architekturę sieci warto ją uprościć. Dzięki temu będzie mniej wag i sieć nie będzie w stanie zapamiętać wszystkich inforacji.

Deep fakes

Pojęcie ***deepfake*** powstało z połączenia dwóch angielskich słów: ***deep*** – jako skrót od głębokiego uczenia (*ang. deep learning*) oraz ***fake*** – czyli fałszywy.

Sformułowanie *deepfake* pojawiło się pierwszy raz w 2017 roku jako pseudonim użytkownika, który stworzył i opublikował filmy pornograficzne z wykorzystaniem wizerunków znanych gwiazd.

Jak działają algorytmy deepfake?
Większość deepfake wykorzystuje autoenkodery lub sieci GAN (*Generative Adversarial Network*).



Zagrożenie czy możliwości?
W zależności od intencji twórcy *deepfake* może stanowić zagrożenie, ale w niektórych sytuacjach może także okazać się pomocny. Niestety w mediach mówi się przede wszystkim o negatywnym wykorzystaniu tej technologii m.in. do manipulacji i próbie wpływania na innych ludzi.

źródło zdjęcia

Python

Python jest językiem programowania wysokiego poziomu, którego ideą jest czytelność i klarowność kodu. Python jest rozwijany jako projekt Open Source zarządzany przez organizację non-profit Python Software Fundation.
Python nie wymusza jednego stylu programowania i wspiera zarówno programowanie obiektowe, programowanie strukturalne jak i programowanie funkcyjne.
Posiada w pełni dynamiczny system typów oraz automatyczne zarządzanie pamięcią.
Jest to język o największej liczbie bibliotek wspomagających.

Troszkę historii
Python został stworzony w latach 90' przez Guido von Rossum'a jako następca języka ABC.
Nazwa języka nie pochodzi od zwierzęcia, lecz od serialu komediowego emitowanego w latach siedemdziesiątych przez BBC „Monty Python's Flying Circus". Projektant, będąc fanem serialu i poszukując nazwy krótkiej, unikalnej i nieco tajemniczej, uznał Python za najlepszą opcję.

źródło zdjęcia