**Deep Learning TensorFlow + Keras**

**w języku programowania Python**

AI, Deep i Machine Learning

Na samym początku chcielibyśmy zaznaczyć kilka ważnych spraw. W krótkim wstępie przedstawimy Wam proste definicje **sztucznej inteligencji (AI – Artificial Intelligence), uczenia maszynowego (Machine Learning) i uczenia głębokiego (Deep Learning).** Również postaramy się zaprezentować nasze środowisko pracy, czyli język programowania **Python**. Dlaczego **Python**? Co ma w sobie takiego? Dlaczego jest numerem jeden? Opowiemy o tym w tej oto prezentacji 😊.

**Czym jest Deep Learning?**

Na początku zakolegujmy się z twierdzeniem **Deep Learningu**. Inaczej, w języku polskim - uczenie głębokie, to jeden z najszybciej rozwijających się gałęzi sztucznej inteligencji. Polega na tworzeniu sieci neuronowych, czyli takich systemów informatycznych, których budowa i funkcjonowanie przypomina pracę ludzkiego mózgu. Samo pojęcie jest dziedziną szerszego pojęcia zwanego Machine Learningiem, natomiast obydwa pojęcia ściśle nawiązują to sztucznej inteligencji.  
Aby dowiedzieć się czym jest **Deep Learning**, powinniśmy najpierw poznać dwa pojęcia.  
**Artificial Intelligence** (sztuczna inteligencja)oraz **Machine Learning** (uczenie maszynowe). A więc omówmy to po kolei!

**Czym jest sztuczna inteligencja?**

Sztuczna Inteligencja z języka angielskiego **Artificial Intelligence** w skrócie **AI** jest dziedziną wiedzy obejmująca m.in. obliczenia ewolucyjne, sieci neuronowe (deep learning), sztuczne życie czy robotykę. Sztuczna Inteligencja w szczególności kojarzona jest z Inteligencją.

**Inteligencja (naturalna inteligencja)**, jest zdolnością do postrzegania, analizy i adaptacji zmian względem otoczenia. Zdolność rozumienia, uczenia się oraz wykorzystywania posiadanej wiedzy i umiejętności w wielu sytuacjach. W wielu przypadkach te sytuacje są uznane przez nas za trudne.

Mówiąc o **sztucznej inteligencji** rozszerzamy pojęcie **inteligencji**. Sztuczną inteligencją możemy nazwać maszyny czy programy komputerowe, które próbują naśladować człowieka. Chcemy zatem uznać za „inteligentne” takie programy, które są w stanie rozwiązywać trudne zadania. Ponieważ w naturalny sposób celem konstruowania takich programów jest umożliwienie zautomatyzowania pewnych czynności wykonywanych przez ludzi, interesować nas będzie rozwiązywanie zadań, które według opinii człowieka wymagają inteligencji.

**Sztuczną inteligencję** można definiować inaczej, na przykład:

**- SI** to nauka mająca za zadanie nauczyć maszyny zachowań podobnych do ludzkich,  
**- SI** to nauka o tym, jak nauczyć maszyny robić rzeczy które obecnie ludzie robią lepiej.  
**- SI** to nauka o komputerowych modelach wiedzy umożliwiających rozumienie, wnioskowanie i działanie.

Pojęcie **sztucznej inteligencji** pojawiło się w latach 50 ubiegłego wieku i z racji bardzo ograniczonej mocy obliczeniowej, rozwój tej dziedziny był bardzo powolny. W dzisiejszych czasach pracę nad sztuczną inteligencją stały się bardzo popularne głównie z dwóch powodów:

Moc obliczeniowa super komputerów, a nawet personalnych komputerów pozwalają na wykonywanie skomplikowanych obliczeń wymaganych w algorytmach sztucznej inteligencji. Ogromna ilość danych, które trzeba przetworzyć - człowiek nie jest w stanie odczytać, usystematyzować, przeliczyć oraz zapisać ciągle zmieniających się danych.

**Czym jest Machine Learning?**

***Uczenie maszynowe***dziedzina zajmująca się analizą danych, na których podstawie pozyskuję informację. Zawiera się w pojęciu sztucznej inteligencji. Algorytmy uczenia maszynowego zdolne są do przetwarzania ogromnej ilości danych, w których odnajduje wzorce, reguły i zależności. Wykorzystanie wiedzy z analiz uczenia maszynowego można zauważyć m.in. filtrach antyspamowych, prężnie działających wyszukiwarkach internetowych, systemach rekomendacyjnych i innych. W odróżnieniu od zwykłych algorytmów, uczenie maszynowe nie posiada z góry narzuconych kroków do osiągnięcia celu jak to ma miejsce np. w algorytmie poszukiwania w grafie. Algorytm uczenia maszynowego sam odnajduję sekwencje kroków, jakie powinien wykonać, aby otrzymać wynik a proces zdobywania wiedzy na podstawie przykładowych danych nazywany jest właśnie uczeniem algorytmu.

**Uczenie maszynowe jest najczęściej wybieraną dziedziną sztucznej inteligencji – przez ostatnie lata rozwija się niesamowicie szybko.**

**Jak działa Deep Learning?**

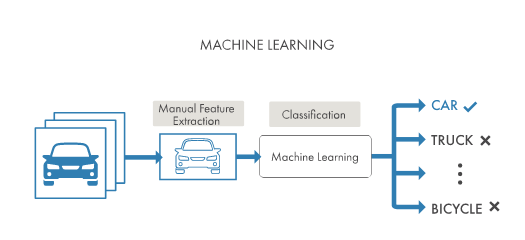
Jak wspomnieliśmy wcześniej, deep learning jest działem machine learningi. Poddział ten wykorzystuje wiele warstw tzw. „neuronów”, aby skutecznie wyodrębnić dane wyższego poziomu z dostępnych danych wejściowych. Dla przykładu – niższe warstwy mogą odpowiadać, za wyznaczanie krawędzi obrazka, gdy wyższe zajmują się rozpoznawaniem rzeczy, które głównie są istotne dla człowieka, jak twarze, litery czy cyfry. W skrócie Deep Learning tworzy sieć neuronową, która ma za zadanie symulować prace ludzkiego mózgu.

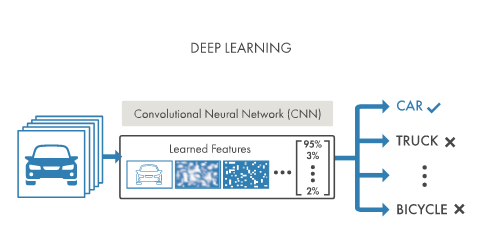
**Różnice pomiędzy Deep, a Machine Learningiem!**

Jako pierwsze zaprezentujmy działanie **uczenia maszynowego**. Będziemy pracować na kolekcji zdjęć z dwoma wyróżniającymi się dla człowieka obiektami. W jednej grupie zdjęć będą znajdować się pieski, natomiast w drugiej kotki. Aby pomóc naszemu algorytmowi to najpierw musimy „przedstawić” mu te obrazy. Jednak, skąd algorytm wie jakie to jest zwierzę? Odpowiedzą na to pytanie jest obszerność jak i poprawność danych wejściowych. Po prostu, osoba trzecia musi zdefiniować, które z dostępnych obrazków należą do jakiej grupy. Zaznaczyć charakterystyczne cechy obu zwierząt i zapodać je programowi. Takie dane są wystarczające, aby nasz algorytm uczenia maszynowego mógł zacząć trenować rozpoznawanie naszych zwierząt. Z każdym nowym obrazkiem algorytm analizuje i rozpoznaje cechy charakterystyczne dla zwierzęcia. A to wszystko w oparciu o poprzednie identyfikacje.

A jak działa **deep learning?** Deep learning używa innego podejścia do tematu, główna jego zaletą jest to, że niekoniecznie potrzebuję uporządkowanych danych z wyznaczonymi cechami charakterystycznymi obu zwierząt, aby je sklasyfikować. W takim przypadku surowe dane (zdjęcie), jest przesyłane pomiędzy różnymi poziomami sieci neuronowej, każdy z tych poziomów odpowiada za rozpoznanie i wydobycie konkretnej cechy zwierzęcia. Można powiedzieć, ze Deep learning uczy się od początku do końca. Sieć dostaje surowe dane i zadania do wykonania. Algorytm sam musi sklasyfikować, nauczyć się i ocenić czy na zdjęciu znajduję się piesek czy kotek.

Warto wspomnieć, że deep learning udoskonala się wraz ze wzrostem automatycznie przeanalizowanych danych.





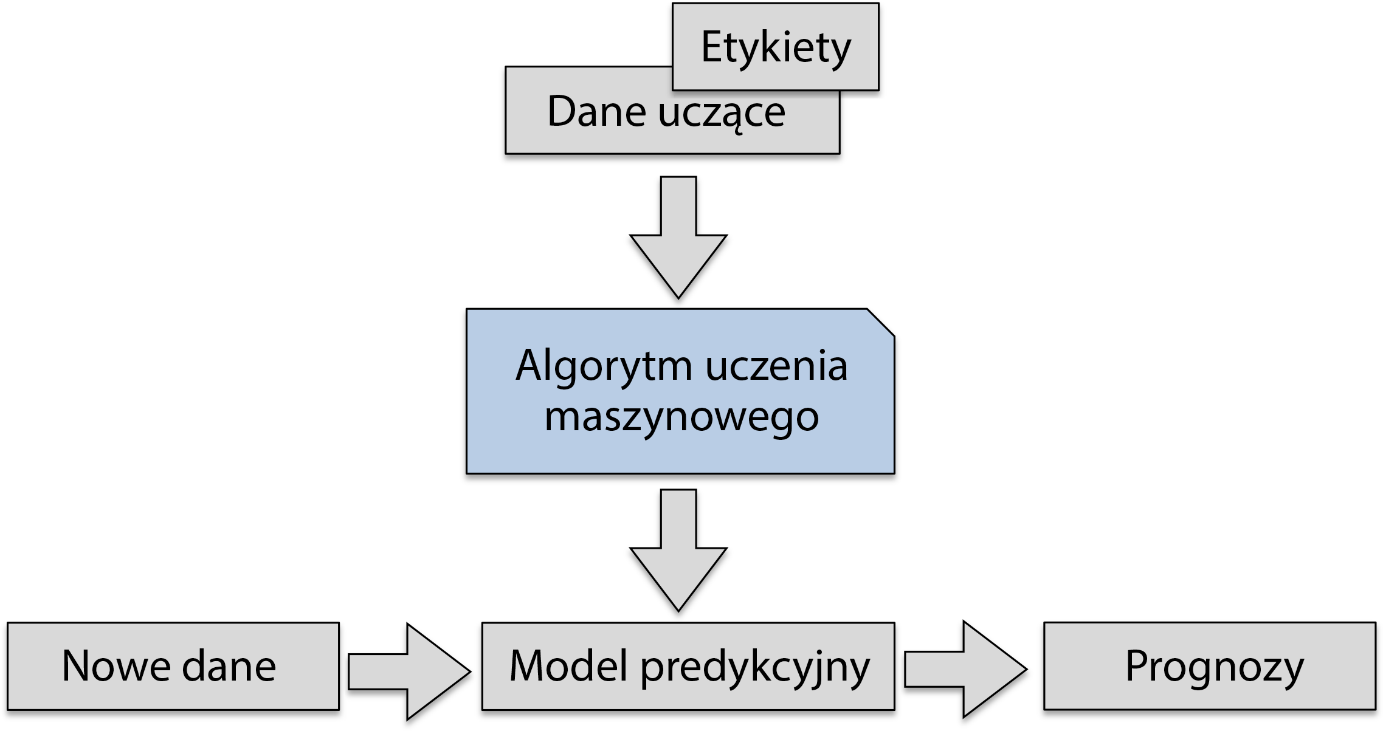
**O nie! Uczenie maszynowe ewoluuje!**

Mimo to, że staramy się rozróżniać te dwa pojęcia to tak naprawdę **Deep Learning jest Machine Learningiem!** Deep Learning powinniśmy traktować jako ewolucję Machine Learningu! Tak naprawdę Deep Learning to Machine Learning wykorzystujący zaprogramowaną sieć neuronowa!

**Jak może uczyć się maszyna?**

**Uczenie nadzorowane**

Ten typ uczenia maszynowego używa tzw. oznakowanych danych uczących, które to oznaczenia nazywane są etykietami. Zbiór danych posiada listę cech (np. wzrost, waga, wiek), na podstawie których można wywnioskować etykietę (np. płeć). Można przyjąć standard oznaczenia



Często stosowana jest praktyka dzielenia danych uczących na podzbiór treningowy oraz testowy w celu sprawdzenia wydajności modelu predykcyjnego. Algorytm uczenia maszynowego tworzy model predykcyjny na podstawie danych treningowych wraz z etykietami, po czym podzbiór testowy (bez podania etykiet) zostaje poddany predykcji a na końcu sprawdzana jest lista otrzymanych etykiet (prognoz) z listą rzeczywistą etykiet dla tego podzbioru.

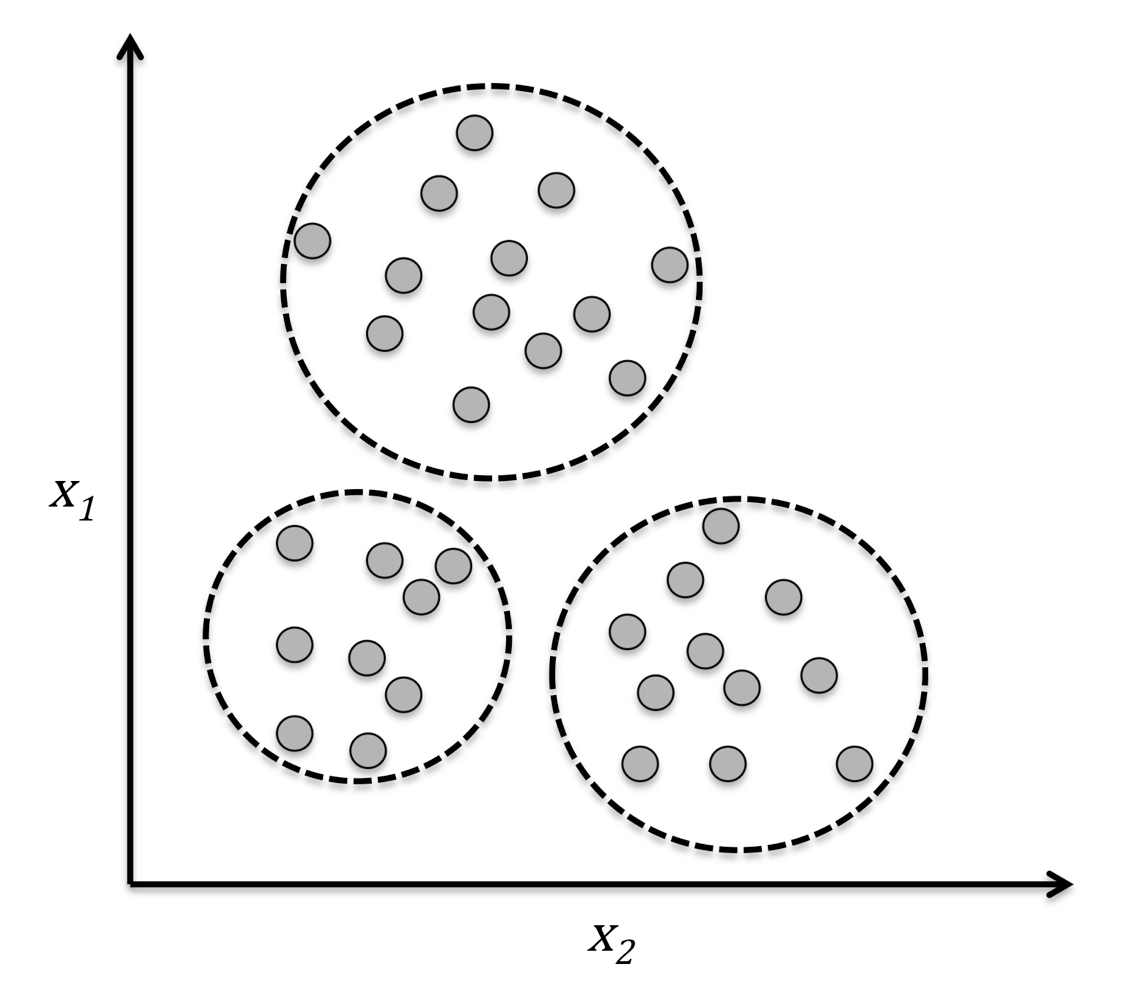
Wyniki wydajności modelu predykcyjnego mogą mieć kluczowe znaczenie. W przypadku niewystarczającej wydajności programista może zmienić algorytm uczący, szukać błędu w danych lub próbować użyć parametrów algorytmu, aby ten lepiej dostosował się do przekazanych mu danych. Jeżeli model osiągnie już właściwą skuteczność, można swobodnie zaimplementować go do całości systemu jak np. model predykcyjny klasyfikujący wiadomość jako spam lub zwykłą wiadomość można zaimplementować do systemu zarządzającego skrzynką poczty elektronicznej.

**Uczenie nienadzorowane**

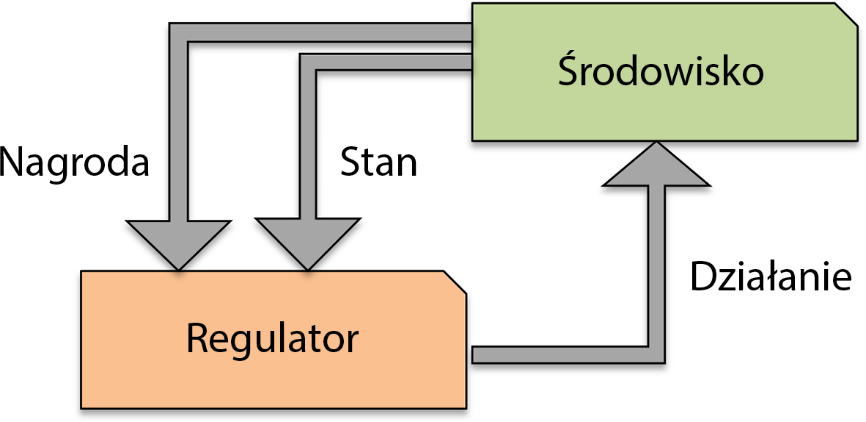
Uczenie nienadzorowane używane jest do analiz nieznanych lub zbyt złożonych struktur danych. W przeciwieństwie do uczenia nadzorowanego nie występuję tutaj pojęcie etykiet tj. dane, które zostają poddane analizie, nie mają kolumny wynikowej, dzięki której można było odnaleźć ścieżkę i zależności w cechach, które prowadziły do konkretnego wyniku. Algorytm uczący odnajduję grupy danych na podstawie podobieństw względem wybranych cech (np. stosunek wagi do wzrostu mógłby utworzyć grupy ludzi o podobnych wymiarach).

Bardziej zaawansowanym przykładem mógłby być system rekomendacji na podstawie popularności danego produktu w określonej grupie:

Dane posiadają listę użytkowników, każdy użytkownik posiada listę tytułów filmów jako listę cech i przypisaną do nich flagę czy tytuł został obejrzany przez użytkownika. Algorytm uczenia nienadzorowanego określi grupy użytkowników, którzy oglądali te same tytuły oraz zależności między grupami. Algorytm może wykazać, że użytkownikom z grupy A warto zaproponować tytuły z grupy B, jeżeli grupa A oraz B leżą blisko siebie na wykresie zależności.

**Uczenie przez wzmacnianie**

Kolejnym typem uczenia maszynowego jest uczenie przez wzmacnianie [12]. Algorytm analizuje przychodzące dane, podejmuję pewną decyzję i na podstawie rezultatów tej decyzji określa, czy to była dobra decyzja, czy też nie, aby zastosować ewentualna regulacje. Rys. 7 przedstawia schemat uczenia przez wzmacnianie, na którym widać, iż regulator podejmuje działanie na środowisko po, czym otrzymuje stan tego środowiska oraz nagrodę, dzięki czemu może podjąć następne działanie z poprawką na błędy.



Za przykład można przyjąć robota zwiadowczego dla którego środowiskiem będzie teren, który robot powinien przejechać a nagrodą jest dojechanie do określonego celu. Robot będzie do tej pory wykonywał działanie (jazdę wraz z manewrami) dopóki nagroda nie będzie spełniona, czyli nie dojedzie do określonego położenia. Stan mógłby w tym przypadku przedstawiać pewnego rodzaju przeszkody (ukształtowanie terenu) które robot powinien omijać.

Kolejnym bardzo popularnym przykładem wykorzystania uczenia przez wzmacnianie jest system aplikacji szachowej w którym nagrodą jest zwycięstwo, stan określa aktualne położenie figur a działanie to pojedynczy ruch na szachownicy (środowisku).

**Do czego wykorzystywany jest Deep Learning?**

Algorytmy sztucznej inteligencji są zaimplementowane w większość systemów informatycznych jakie w tej chwili funkcjonują na świecie. Każdy człowiek korzystający z komputera w dzisiejszych czasach doświadcza potęgi sztucznej inteligencji, może nawet nie zdając sobie sprawy z jej istnienia. Wspomniane już silniki wyszukiwarek internetowych czy systemy rekomendacyjne to tylko nieliczne przykłady wykorzystania tej dziedziny informatyki.

Sztuczna inteligencja jest używana również do przetwarzania wizyjnego, przetwarzania języka naturalnego, rozpoznawania mowy, gier oraz robotyce.

**Przetwarzanie wizyjne**

Wspomniane uczenie nadzorowane bardzo często wykorzystywane jest do przetwarzania wizyjnego a dokładniej do operacji na obrazach. Przykładem takiego zastosowania może być zdjęcie przedstawiające obiekty (np. kwiaty), system na podstawie binarnych danych z tego obrazka oraz etykiety przydzielonej do zdjęcia, uczy się wzorców, które elementy odpowiadają za odpowiednią klasyfikacje, jaki to kwiat bądź inny obiekt. Stosując analogie do opisanego przykładu uczenia nadzorowanego, można ten przykład zaprezentować w formie listy zbinaryzowanych pikseli ‘X’ do której jest przypisana wyjściowa etykieta ‘y’. Tak stworzonego modelu predykcyjnego można użyć w systemach takich, jak kamera w smartfonie, w której po zrobieniu zdjęcia wyświetlałaby się lista odnalezionych obiektów na zdjęciu wraz z prawdopodobieństwem prognozy.

**Przetwarzanie języka naturalnego**

Systemy zajmujące się przetwarzaniem języka naturalnego mają na celu analizę ze zrozumieniem podanego teksu, generowanie tekstu np. samo piszące się artykuły.

Najbardziej podstawową funkcją tego przetwarzania może być autokorekta błędów ortograficznych w pisanym tekście przez użytkownika.

**Rozpoznawanie mowy**

W dobie tak szeroko rozwiniętej technologii, inteligentny asystent komputerowy staję się coraz częściej używaną funkcją. Największe firmy jak Google, Microsoft czy Apple rozwijają technologie wykorzystującą algorytmy sztucznej inteligencji do przetwarzania mowy. Systemy takie jak Siri, Cortana czy Google Now mają na celu przeanalizować wypowiedziane frazy, zrozumieć kontekst i wykonać odpowiednie dla kontekstu działanie. Przykładem może być wypowiedzenie polecenia ‘Pogoda’ po którym inteligentny system uruchomi aplikację pogoda a w niektórych przypadkach za pomocą syntezatora mowy, asystent podyktuje warunki atmosferyczne wyświetlone w tej aplikacji.

Często wraz z systemami przetwarzania mowy współuczestniczą algorytmy przetwarzania języka naturalnego. System przetwarzania mowy zamienia nagraną frazę na tekst i przy użyciu przetwarzania języka naturalnego analizowany jest kontekst wypowiedzianej frazy, po czym generowana jest odpowiedź i w formie zwrotnej algorytm z użyciem syntezatora mowy odpowiada.

**Gry**

W grach zastosowanie znajduję bardzo wiele algorytmów sztucznej inteligencji. Uczenie przez wzmacnianie jest wykorzystywane w elementach gry, które powinny, zachowywać się ‘inteligentnie’ tj. wykonywały, pewna czynność do momentu aż osiągną, nagrodę przy zmniejszaniu popełnianych błędów i eliminując wszystkie przeszkody.

Statystyki w grze mogą wykorzystywać zarówno uczenie nadzorowane, jak również uczenie nienadzorowane. Atrybuty gracza mogą być zbierane i przypisane do wyników danej rozgrywki co może być podstawą do stworzenia modelu predykcyjnego, który pozwoli oszacować skuteczność gracza w następnej rozgrywce przy posiadaniu danych atrybutów. Natomiast uczenie nienadzorowane mogłoby wyszczególnić grupy atrybutów, które mają cechy wspólne względem pewniej zmiennej.

**Robotyka**

Wspomniany już wcześniej robot zwiadowczy to tylko jedno z niewielu zastosowań sztucznej inteligencji w robotyce. Bardzo wiele maszyn produkcyjnych używa algorytmów sztucznej inteligencji w celach poprawy jakości wykonywanej pracy. Za przykład można podać maszynę sortującą warzywa:

Urządzenie posiada kamerę, taśmę produkcyjną oraz podajnik wraz z dwoma zsypami na warzywa dobrej, oraz gorszej jakości. Kamera za pomocą algorytmów rozpoznawania obrazu klasyfikuję dane warzywo jako dobrej jakości lub gorszej i ta decyzja jest wykorzystywana w podajniku, do którego zsypu wrzucić warzywo.

**Odkrywanie leków i toksykologia**

Duży procent leków kandydujących nie uzyskał zgody organu nadzoru. Niepowodzenia te są spowodowane niewystarczającą skutecznością (efektami ubocznymi, lub niezadowalającymi efektami końcowymi) lub nieprzewidzianymi skutkami toksycznymi. W badaniach zbadano zastosowanie Deep Learningu do przewidywania celów biomolekularnych.

AtomNet to system głębokiego uczenia się oparty na strukturze projektowania leków. AtomNet zastosowano do przewidywania nowych kandydujących biomolekuł na cele związane z chorobą, takie jak wirus Ebola czy stwardnienie rozsiane.

Python

**Co to jest Python?**

Wysoko poziomowy język programowania, wydany został początkiem lat 90 przez Guido van Rossum. Język ten jest rozwijany jako projekt z otwartym kodem źródłowym i jest dostępny w dwóch wersjach: **Python 2** oraz **Python 3**. Z racji dużych różnic oraz brak wsparcia dla kilku ważnych modułów, zdecydowano się rozwijać dwie wersje niezależnie od siebie.

**Jakie są główne cechy języka Python?**

Język programowania Python jest atrakcyjny dla początkującego programisty z racji na jego prostotę oraz fakt, że jest on darmowy. Niemniej jednak Python dzięki swojej skalowalności jest odpowiednim narzędziem dla profesjonalistów, którzy tworzą duże systemy komercyjne. W szczególności warto nadmienić tu kilka najważniejszych cech tego języka.

**Interpretowany!**

Napisany kod źródłowy poddany jest analizie i każde przeanalizowane polecenie zostaje od razu wykonane. Proces ten może okazać się dłuższy niż wykonanie klasycznie skompilowanego programu, lecz w przypadku wprowadzenia zmian bądź debugowaniu napisanego programu, interpreter zyskuje, na czasie względem kompilowanych języków co stanowi, jeden z wielu pozytywnych stron tego języka. Możliwe jest wykonanie poleceń z poziomu interpretera (linia po linii) lub wykorzystanie edytora tekstu i stworzenie tzw. skryptu z rozszerzeniem **.py**.

**Wieloparadygmatowy!**

Python umożliwia wykorzystanie kilku paradygmatów programowania (obiektowy, imperatywny, funkcyjny) przy czym nie ogranicza w wyborze jednego w skali programu. Programista może łączyć najlepsze rozwiązania z wybranych paradygmatów.

**Przenośny?!**

Omawiany język programowania dostępny jest na wszystkich liczących się platformach tj. Windows, Mac Os oraz Linux. Program napisany na jednej platformie, można w prosty sposób skopiować i działanie nie ulegnie zmianie oraz nie będzie konieczności wprowadzenia żadnych zmian.

**Typuje dynamicznie!**

Typ danych jest automatycznie przypisywany na podstawie wartości, jaka w danym momencie jest przypisana do danej zmiennej. Usprawnia to szybkie pisanie programu, gdyż programista nie musi zastanawiać jaki typ powinien zastosować w danym momencie. Python umożliwia, również nadpisywanie typów tj. raz przypisany typ można, zmienić przypisując, do zmiennej obiekt innego typu.

**Sam zarządza pamięcią!**

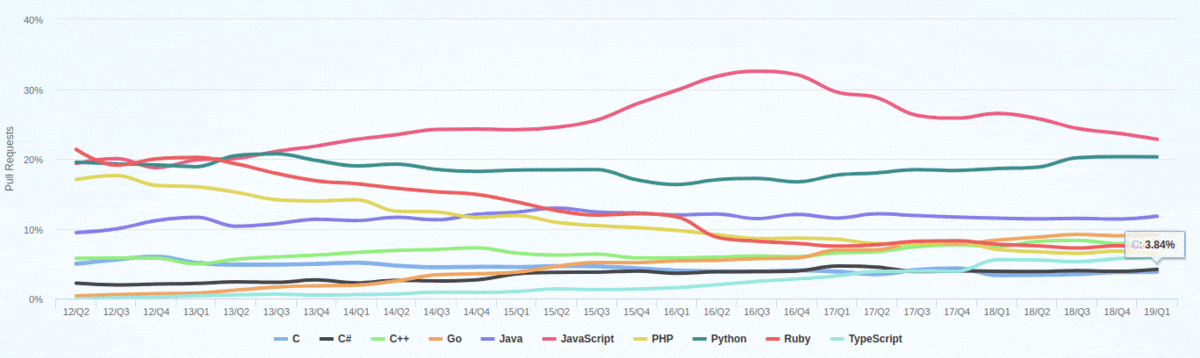
Python wykorzystuje mechanizm tzw. Garbage collector, który to ma za zadanie odciążyć programistę z przejmowania się alokacją pamięci. Mechanizm ten, sam usuwa z pamięci podręcznej, nieużywane obiekty.

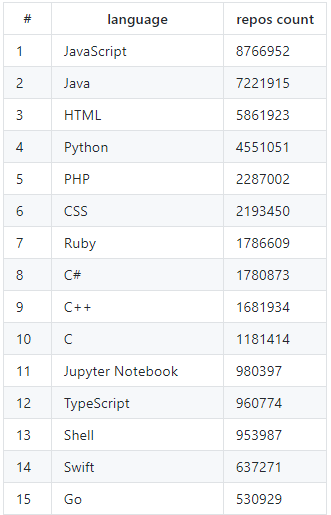
**Multum bibliotek oraz wsparcie społeczności!**

Język ten udostępnia ogromną ilość wbudowanych bibliotek pomagających programiście tworzyć programy. Fakt, iż Python posiada otwarty kod źródłowy, sprawia, że bardzo wiele ludzi angażuję, się w rozwój tego języka tworząc dodatkowe biblioteki.

**Właśnie dlatego!**

Właśnie dlatego na przestrzeni ostatnich lat Python wysunął się na podium w świecie programistycznym. W 2019 roku język Python umiejscawiał się w top 3 najczęściej wybieranych języków programowania.

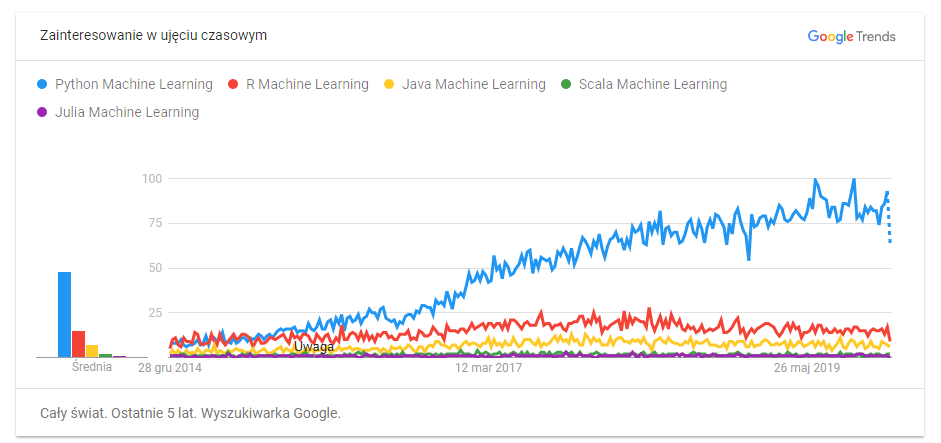


 Wszystkim dobrze znana strona GitHub, udostępniła w 6 grudnia 2019 roku ilość repozytoriów stworzonych na ich stronie pogrupowanych językami programowania. Oto wyniki:

Jak widzimy uplasował się na czwartym miejscu (Tak naprawdę to trzecim, wszyscy wiemy, ze HTML to nie język programowania 😊). Zaraz za JavaScriptem i Java.

**No fajnie, ale dlaczego Deep Learning i Python?**

To pytanie nie jest wcale takie trudne! Głównym powodem jest ciągły wzrost popularności języka Python jak i samego terminu uczenia maszynowego. Co za tym idzie (jak wyżej wspomnieliśmy), o wiele większe wsparcie społeczności, bibliotek i tym podobnych.

Niżej przedstawiamy wykres, który pokazuje popularność wyszukiwanych fraz związanych z machine learningiem i językiem programowania.  


**I coś więcej?** Tak! Python jest prosty i łatwy do zrozumienia! Po za tym jest super efektywny dzięki funkcjom zawartym w samej podstawie języka.

**Ale co z tymi bibliotekami?** Python posiada wiele bibliotek, a w nich wsparcie, które nam w naszej sprawie najbardziej odpowiada. Mianowicie, biblioteki dla Deep Learningu, Machine Learningu jak i Artifical Inteligence!

Przedstawmy może kilka z nich:  
 **- TensorFlow\*,  
 - Keras\*,  
 - PyTorch,  
 - ApacheMXNet,  
 - Theano.**

**Czym jest TensorFlow**? Jedna z największych bibliotek z zakresu sztucznej inteligencji jaką można znaleźć na rynku. Opublikowana jako projekt z otwartym kodem źródłowym przez firmę Google, w celu podzielenia się własnymi rozwiązaniami a także udoskonaleniu ich z pomocą ogromnej społeczności. Moduł ten umożliwia prace nad uczeniem maszynowym z wykorzystaniem ukierunkowanym grafom, w których wierzchołki są operacjami wykonywanymi na danych przekazywanych przez krawędzie. Przewagą tej biblioteki jest możliwość wykonywania obliczeń nie tylko przy użyciu zwykłego procesora CPU, ale także procesora graficznych GPU co daje większa wydajność przy zaawansowanych obliczeniach na dużych zbiorach danych.

TensorFlow został pierwotnie opracowany przez naukowców i inżynierów pracujących w zespole Google Brain w ramach organizacji Google Intelligence Research do przeprowadzania uczenia maszynowego i głębokich sieci neuronowych. System jest na tyle ogólny, że można go stosować również w wielu innych domenach

**Czym jest Keras?** Keras to biblioteka sieci neuronowych typu open source napisana w języku Python. Kompatybilna z TensorFlow, Microsoft Cognitive Toolkit, R, Theano lub PlaidML. Zaprojektowany, aby umożliwić szybkie eksperymentowanie z głębokimi sieciami neuronowymi, koncentruje się na byciu przyjaznym dla użytkownika, modułowym i rozszerzalnym. Został on opracowany w ramach prac badawczych projektu ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), a jego głównym autorem i opiekunem jest François Chollet, inżynier Google.

W 2017 r. Zespół Google TensorFlow postanowił wesprzeć Keras w podstawowej bibliotece TensorFlow. Chollet wyjaśnił, że Keras został pomyślany raczej jako interfejs niż samodzielna platforma uczenia maszynowego. Oferuje wyższy poziom, bardziej intuicyjny zestaw abstrakcji, które ułatwiają opracowywanie modeli głębokiego uczenia się niezależnie od zastosowanegozaplecza obliczeniowego.

**Pozostałe biblioteki –** Język programowania Python jest bogaty w dostępne biblioteki z zakresu sztucznej inteligencji oraz przetwarzania danych, które można wymieniać bez końca, niemniej jednak wymienione powyżej biblioteki są najczęstszym wyborem programistów pracujących nad opisanymi dziedzinami. Listę większości dostępnych bibliotek można znaleźć na wiki pythona:

[*https://wiki.python.org/moin/PythonForArtificialIntelligence*](https://wiki.python.org/moin/PythonForArtificialIntelligence)

**A tak po za tym, do czego ten cały wąż?**

Python jest językiem programowania ogólnego zastosowania i dzięki wymienionym cechom w poprzednich podrozdziałach, świetnie się nadaje do tworzenia stron internetowych (django) sprawdza się również w narzędziach monitorujących oraz zarządzających siecią. W mniejszym stopniu, lecz jednak wartym wspomnienia, Python wykorzystywany jest do tworzenia gier oraz dzięki niemu można w szybki i łatwy sposób stworzyć aplikacje okienkowe. Zawierająca w sobie również statystykę oraz obliczenia naukowe, które są bardzo przydatne dla dziedziny, która właśnie opisujemy.

Język programowania, który tak często wspominamy używany jest także przez profesjonalistów w większych komercyjnych projektach. Znajduję potwierdzenie na liście światowej klasy firm które z niego korzystają. Na liście można zauważyć znane firmy takie jak Google, Yahoo czy Nasa które wykorzystują Pythona do swoich badań, wyszukiwarek, biznesowych aplikacji odpowiadających za planowanie czy nawet do tworzenia systemów rekomendacyjnych.

TUTAJ INSTALACJA