Sprawozdanie z projektu

Przedmiot

Biologically Inspired Artificial Intelligence

Temat projektu

Przewidywanie ataku serca

Rok akademicki	Prowadzący	Skład sekcji
2020/2021	mgr inż. Marcin Wierzchanowski	Michał Opiełka Krystian Stebel

1. Temat projektu

Wybrany przez nas temat projektu to “Przewidywanie ataku serca”. Celem projektu jest przewidzenie ataku serca bazując na zestawie 13 zmiennych odzwierciedlających wyniki badań medycznych pacjenta oraz jego samopoczucie.. Główną inspiracją podczas wyboru tematu były zbiory danych na stronie kaggle.com.

2. Analiza zadania

2.1. Prezentacja zestawu danych

W celu wykonania zadania posłużyliśmy się zestawem danych ze strony kaggle.com (link w pkt. 6). Zestaw składa się z 303 rekordów, 13 zmiennych oraz jednego pola wynikowego. Do zmiennych należą:

- wiek - zakres od 29 do 77 lat
- płeć - oznaczana przez zera i jedynki - przeważają mężczyźni
- rodzaj bólu w klatce piersiowej - jest klasyfikowany według 3 kryteriów - gdzie występuje, dlaczego występuje, czy ustępuje po odpoczynku
- ciśnienie tętnicze - ciśnienie spoczynkowe po przyjęciu do szpitala
- poziom cholesterolu - wyrażany w mg/dL
- nadmierny poziom cukru - jeżeli poziom cukru przekracza 120 mg/dL oznaczamy jako 1, w przeciwnym razie 0
- wynik EKG w spoczynku - 0 oznacza prawidłowy kształt wykresu, 1 i 2 oznaczają kształty charakterystyczne dla konkretnych zaburzeń
- maksymalne tętno
- występowanie bólu po wysiłku fizycznym - 1 jeśli występuje, 0 jeśli nie występuje
- obniżenie odcinka ST - różnica między najniższymi punktami odcinka ST wykresu podczas spoczynku i wysiłku fizycznego w mm
- kierunek zbocza odcinka ST - wartość 0 dla zbocza wznoszącego, 1 dla zbocza płaskiego i 2 dla zbocza opadającego
- liczba głównych naczyń wieńcowych zabarwionych podczas badania - wynik reprezentuje liczbę głównych naczyń wieńcowych zabarwionych w badaniu. Badaniu polega na wprowadzeniu do ciała radioaktywnego barwnika i wykonaniu zdjęcia rentgenowskich w celu wykrycia zaburzeń w sercu
- wynik testu obciążenia z użyciem talu - test polega na wprowadzeniu do ciała radioaktywnego pierwiastka talu i wykonania obrazu serca z pomocą kamery gamma. Wartość 0 - czyli wynik w normie, oznacza że w obu przypadkach udało się zaabsorbować izotop. Wartość 1 oznacza że nie udało się to w żadnym przypadku

Pewną wadą zestawu była spora ilość parametrów kategorycznych, które w procesie oczyszczania danych zostały zamienione na numeryczne. Należą do nich między innymi płeć czy rodzaj bólu w klatce piersiowej.

Ostatecznie jednak uważamy, że był to jeden z bogatszych zestawów danych reprezentujących tematykę medyczną

2.2. Technologie

Zdecydowaliśmy się na wybór języka Python głównie z powodu jego prostoty a także intuicyjności. Dodatkowo zdecydowaliśmy się na biblioteki takie jak:

- pandas - odpowiada za wczytanie zestawu danych z pliku *.csv
- numpy - pozwala na wykonywanie operacji numerycznych
- tensorflow oraz keras - udostępniają interfejs pozwalający na tworzenie sieci neuronowych
- sklearn - biblioteka ułatwiająca uczenie maszynowe

2.3. Propozycja rozwiązania

Tworząc rozwiązanie wykorzystaliśmy ANN czyli sztuczną sieć neuronową. Sieć składa się z trzech warstw:

- wejściowej odbierającej dane wejściowe,
- ukrytej przetwarzającej dane wejściowe
- wyjściowej, odpowiadającej za wyniki

Postawiony problem został rozwiązany w sześciu krokach:

- pobieranie danych - wykorzystanie biblioteki pandas do pobrania zestawu 303 rekordów z pliku *.csv
- oczyszczanie danych - polega na zamianie parametrów kategorycznych na numeryczne. Przykładowo kolumna reprezentująca rodzaj bólu w klatce piersiowej zawierała wartości od 0 do 3. Po zamianie na parametry numeryczne dysponowaliśmy czterema kolumnami zawierającymi tylko wartości 0-1
- podział na zestaw treningowy i testowy - 80% danych zostało przeznaczone na zestaw treningowy, 20% na zestaw testowy.
- normalizacja - przeskalowanie danych wejściowych w taki sposób, aby znajdowały się w porównywalnym zakresie np. [0,1] lub [-1, 1]. Chcieliśmy uniknąć sytuacji, w której dysponujemy dwoma wartościami o bardzo rozbieżnych zakresach - jedną od 0 do 0.5 i drugą od 0 do 1000.
  Zmiana o 0.5 to zmiana aż o 100% w pierwszym przypadku i tylko 0.05% w drugim. Prowadziłoby to do powstawania zaburzeń podczas mnożenia przez wagi. Zastosowanie normalizacji daje pewność, iż do takich zaburzeń nie dojdzie.
- regresja - uśrednienie wyników i dopasowanie ich do przebiegu pewnej funkcji
- prezentacja wyniku - prezentacja wyniku w konsoli

3. Specyfikacja oprogramowania

Projekt jest aplikacją konsolową napisaną w języku Python. Dane wejściowe podawane są przy użyciu pliku *.csv, a wynik wyświetlany jest na ekranie oraz zapisywany w pliku *.txt.
Przykładowe polecenie pozwalające na uruchomienie programu:

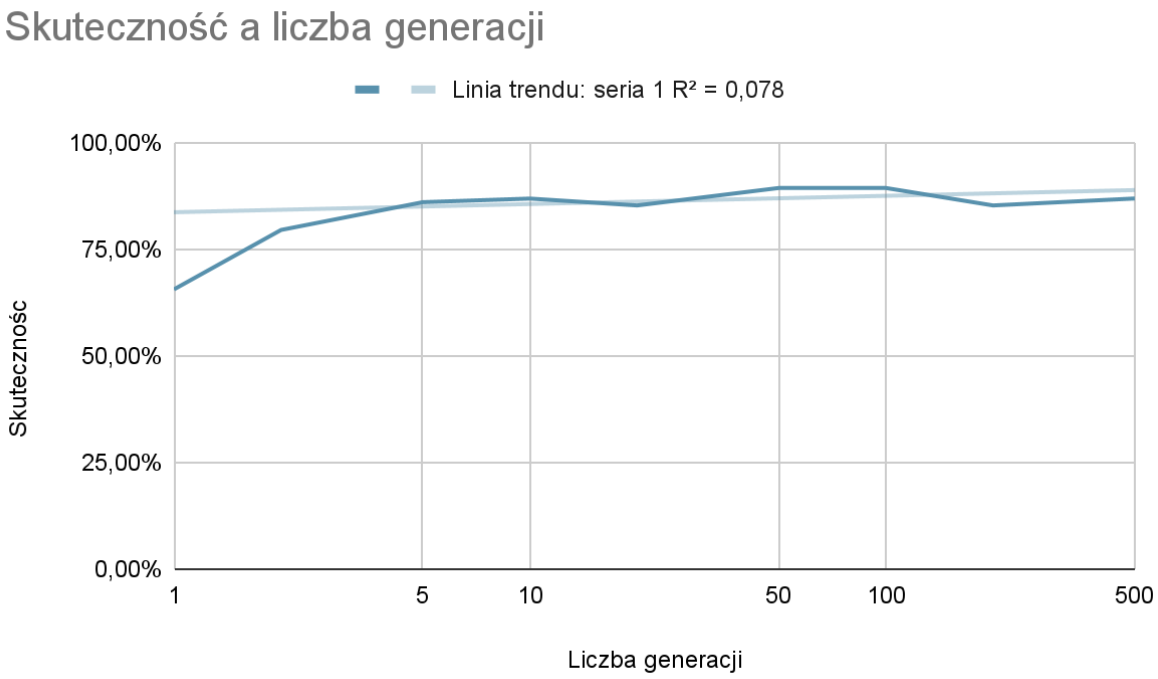
python predict.py -i ../DATA/heart.csv -o ../DATA/res.txt -u 20 -g 100

Parametr -i pozwala na dołączenie pliku *.csv zawierającego zbiór danych.
Parametr -o wskazuje plik wynikowy *.txt.
Parametr -u oznacza liczbę jednostek.
Parametr -g oznacza liczbę generacji.

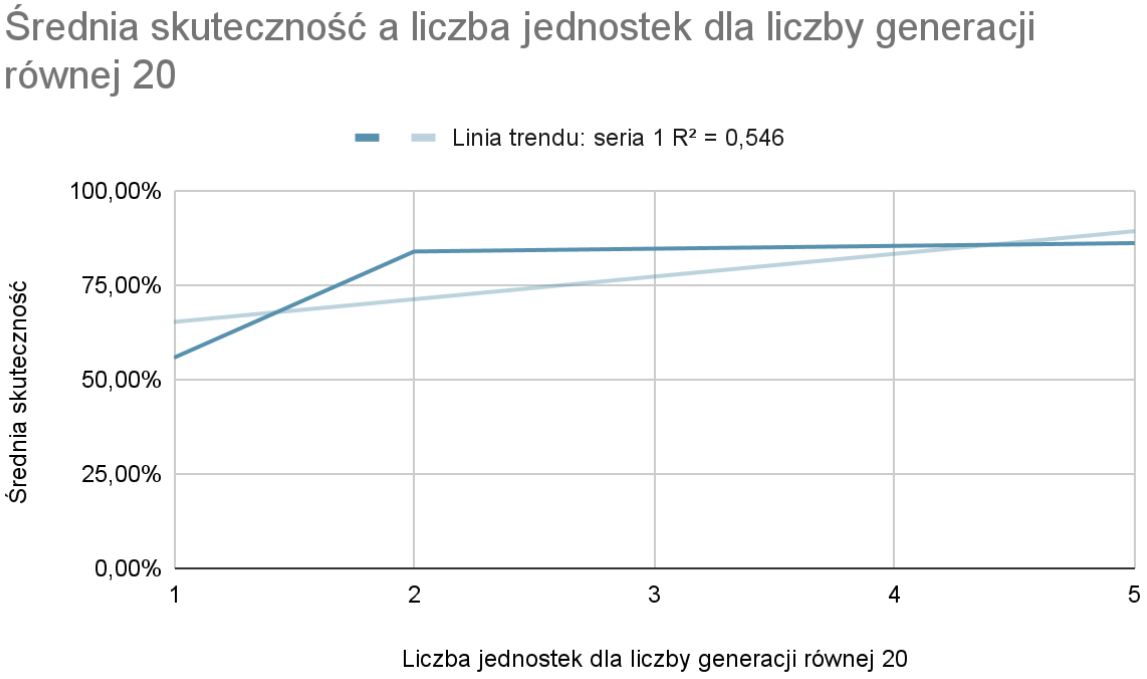
4. Testowanie

Program był testowany przy użyciu zestawu danych testowych, który został wydzielony ze zbioru początkowego, stanowiąc jego 20%. Rekordy treningowe jak i testowe zostały wybrane losowo.

Liczba generacji dla liczby jednostek równej 20	Skuteczność		Średnia
1	73,77%	57,38%	65,58%
2	81,97%	77.05%	79.51%
5	86,89%	85.24%	86.01%
10	86,89%	86.88%	86.86%
20	85,25%	85.24%	85.25%
50	88,52%	90.16%	89.34%
100	88,52%	90.16%	89.34%
200	85,25%	85.24%	85.25%
500	85.24%	88.52%	86.88%



Liczba jednostek dla liczby generacji równej 20	Skuteczność		Średnia
1	55,74%	55.74%	55,74%
2	81,97%	85.72%	83,85%
5	86,89%	85.24%	86,07%
10	85,25%	86.52%	85,89%
20	85,25%	88.52%	86,89%
50	91,80%	88.54%	90,16%
100	88,52%	80,34%	84,43%
200	85,25%	83,61%	84,43%
500	85,25%	83,61%	84,43%



Jak widać wzrost liczby jednostek lub generacji ma tylko niewielki wpływ na uzyskaną dokładność - dzieje się tak dlatego, że model bardzo szybko uzyskuje dokładność powyżej 80%, co znacząco zmniejsza potencjał dalszego wzrostu. Zestaw danych to 303 rekordy - 20% z nich (zestaw testowy) to 60 lub 61 rekordów - stąd wyniki procentowe się powtarzają. Od 5 pokolenia/jednostki wzwyż mówimy o różnicach jednego lub dwóch trafień, możliwe więc że efekty byłyby widoczne dla większego zestawu danych
Testowanie wykazało skuteczność rzędu ok. 84%

5. Podsumowanie

Projekt pozwolił nam na zapoznanie się z zagadnieniami związanymi ze sztuczną inteligencją oraz skłonił do nauki nowej technologii jaką jest Python.
Stworzenie powyższego programu nie było bardzo wymagające, gdyż dostępnych jest wiele bibliotek języka Python znacząco usprawniających pracę. Należą do nich m.in. numpy czy sklearn. Umożliwiają wykonywanie bardzo złożonych operacji jednocześnie nie wymagając od programisty pisania wielu linii kodu. Projekt można dalej rozwinąć poprzez dodanie możliwości zapisu wytrenowanego modelu do pliku oraz dostrojenie hiperparametrów przy użyciu frameworka Keras Tuner. Można również skorzystać z danych zwracanych przez metodę fit() (obiekt History) do wygenerowania przez program wykresu krzywej uczenia się (learning curve).

6. Źródła

- https://www.kaggle.com/ronitf/heart-disease-uci
- https://www.kaggle.com/onatto/predicting-heart-disease-a-detailed-guide

- https://www.kaggle.com/ronitf/heart-disease-uci/discussion/82648

7. Link do projektu
    - https://github.com/ancf/heart-disease-prediction