POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA
SPECJALNOŚĆ: Inżynieria systemów informatycznych

PRACA DYPLOMOWA MAGISTERSKA

Zastosowanie metod uczenia maszynowego w detekcji fałszywych informacji

Application of machine learning methods to fake news detection

AUTOR:
Inż. Dawid Mikowski

PROWADZĄCY PRACĘ:
Prof. Michał Woźniak

OCENA PRACY:

WROCŁAW, 2020

Spis treści

Wstęp

Motywacja

Ogromna w ostatnich latach popularność różnego rodzaju mediów społecznościowych, a co za tym idzie wymiana ogromnej ilości informacji między ludźmi powoduje, że tak jak nigdy wcześniej w historii spotkać się można z masową dezinformacją i skutkami jakie może ona ze sobą nieść [11]. Nieprawdziwe informacje stanowią realne zagrożenie dla każdego z nas, dlatego odnalezienie prostego sposobu na ich wykrycie staje się coraz ważniejsze i mogłoby zredukować lub całkowicie wyeliminować szansę na bycie oszukanym podczas pozyskiwania informacji z różnych stron internetowych. Popularne ostatnimi czasy algorytmy uczenia maszynowego mogą stanowić rozwiązanie tego problemu.

Cel pracy

Celem pracy jest sprawdzenie efektywności popularnych algorytmów uczenia maszynowego w rozwiązywaniu zadania klasyfikacji nieprawdziwych informacji. Badana metoda opiera się na analizie tekstu podzielonego na tak zwane ngramy czyli sekwencje jednego lub kilku znaków. Metoda ta jest stosowana podczas detekcji spamu w różnego rodzaju skrzynkach mailowych. W badaniu sprawdzano jak zmiana długości ngramów wpływa na poprawność algorytmów, czas ich uczenia oraz czas wykonywania przez nie predykcji.

Zawartość pracy

Praca składa się z sześciu rozdziałów. Pierwsze trzy rozdziały zawierają teoretyczne wprowadzenie do tematów związanych z celem pracy. Kolejne dwa rozdziały składają się z opisu metod jakie zostały wykorzystane do osiągnięcia celu oraz analizę wykonanych badań. Ostatni rozdział stanowi podsumowanie całej pracy.

Dokładny opis zawartości każdego z rozdziałów wygląda następująco:

- Pierwszy rozdział zawiera krótki opis historii nieprawdziwych informacji i ich rozprzestrzeniania, a także definicję pojęcia *fake news* i omówienie jego znaczenia w dzisiejszym świecie. Zostały w nim również opisane sposoby ochrony przed nieprawdziwymi informacjami oraz zagrożenia jakie stanowi stworzona w ostatnich latach technologia deepfake.
- Rozdział drugi zawiera opis technologii jaką jest uczenie maszynowe. Zostały omówione jedne z popularniejszych algorytmów uczenia maszynowego, które wykorzystano do wykonania badań. Został także uwzględniony opis zagrożenia jakie może stanowić uczenie maszynowe dla ludzkości.

- Trzeci rozdział krótko opisuje dziedzinę informatyki, którą jest przetwarzanie języków naturalnych. Zostały opisane przykłady jej zastosowania w dzisiejszym świecie. W rozdziale znajduje się również omówienie sposobów przygotowywania danych tekstowych. Ostatnim elementem rozdziału jest opis metod wektoryzacji, czyli zamiany tekstu na formę numeryczną.
- W rozdziale czwartym zawarty jest opis systemu, który został stworzony w celu wykonania badań. Opisuje wykorzystane w nim technologie oraz motywację wyboru każdego z nich. Zawarte są w nim wymagania funkcjonalne, które musi spełniać ten system oraz krótki opis jego implementacji.
- Piąty rozdział zawiera ocenę eksperymentalną wykonanych badań. Opisane są w nim warunki przeprowadzonego eksperymentu wraz ze specyfikacją danych, wyniki badań a także analiza wyników. Zakończenie rozdziału zawiera wnioski wynikające z analizy badań.

Rozdział 1

Informacje nieprawdziwe w dobie Internetu

Pojęcie *fake news* odnosi się do informacji, które pomimo że nie posiadają pokrycia z rzeczywistością są przedstawiane jako prawdziwe w mediach takich jak np. wiadomości, artykuły, portale społecznościowe itd. Zwrot ten jest neologizmem i w języku angielskim oznacza dosłownie "Fałszywe wiadomości". Tego rodzaju wiadomości często wykorzystywane są w tworzeniu treści humorystycznych takich jak satyra, jednak ich główne przeznaczenie sprowadza się do spełniania jednego z dwóch zadań:

- oszukać odbiorcę i wpłynąć na jego poglądy w sposób żądany przez autora danej informacji (propaganda),
- namówienie go na zakup czegoś czego w innym przypadku by on nie kupił (reklama).

Niektóre fałszywe informacje łączą oba cele.

Fake newsy stały się bardzo popularnym zagadnieniem w ostatnich czasach ponieważ internet, a w szczególności media społecznościowe dają możliwość przekazywania informacji z niespotykaną wcześniej prędkością dzięki czemu rozprzestrzenianie dezinformacji stało się zadaniem stosunkowo prostym.

Zagadnienie to zyskało ogromny rozgłos podczas kampanii wyborczej oraz prezydentury Donalda Trumpa, który zasłynął z częstego wykorzystywania tego zwrotu podczas wywiadów, debat oraz wypowiedzi w mediach społecznościowych takich jak Twitter. Do roku 2020 roku pojęcie *fake news* zostało umieszczone w słownikach języka angielskiego takich jak "Oxford English Dictionary", "Macmillan Dictionary".

Rys. 1.1: Post udostępniony przez Donalda Trumpa na portalu Twitter Źródło: https://twitter.com/

Według projektu "First Draft News" założonego przez dziewięć organizacji, w skład których wchodzą Google, Facebook oraz Twitter możemy wyróżnić siedem typów *fake newsów* [1]:

- satyra bądź parodia - nie ma na celu wyrządzić krzywdy ale może oszukać,
- fałszywe połączenie - nagłówki oraz obrazy nie mają powiązania z zawartością,
- myląca zawartość - tekst napisany w mylący sposób,
- fałszywy kontekst - prawdziwa zawartość powiązana ze złym kontekstem,
- oszukana zawartość - źródła pochodzenia informacji są fałszywe,
- zmanipulowana zawartość - prawdziwa zawartość zmanipulowana w odpowiedni sposób by oszukać odbiorcę,
- sfabrykowana zawartość - całkowicie zmyślona zawartość mająca na celu wyrządzić krzywdę.

Jak podaje słownik "Merriam Webster" po raz pierwszy wykorzystano zwrot *fake news* w roku 1890. Wiele nieprawdziwych wiadomości wzbudziło tak wielkie zainteresowanie, że na zawsze zapisały się one na kartach historii. Przykładami takich fake newsów są [10]:

- życie na księżycu - w roku 1835 nowojorska gazeta The Sun opublikowała 6 artykułów, które ze szczegółami opisywały odkrycie życia na księżycu. Odkrywcą według artykułu miał być znany astronauta John Herschel. Gazeta przyciagnęła ogromne zainteresowanie co znacznie podniosło jej popularność, w tym samym roku przyznała się ona do nieprawdziwości artykułów,
- Kuba Rozpruwacz -w roku 1888 w Londynie popełniono serię brutalnych zabójstw, których popełnienie przypisano seryjnemu mordercy o pseudonimie Kuba Rozpruwacz. Wiele gazet w tamtym okresie wykorzystywało historię o zabójstwach w celu przyciągnięcia uwagi przechodniów, nie zawsze przekazując prawdziwe informacje. Jednym z przykładów była para sprzedawców gazet William Macdonald oraz George Write, którzy sprzedając swoje artykuły wykrzykiwali o aresztowaniu Kuby Rozpruwacza, informacja ta była jednak całkowicie fałszywa, ponieważ morderca nie został nigdy odnaleziony.

Jednym z najsłynniejszych działań wykorzystujących *fake newsy* w złych celach jest propaganda, czyli według definicji Słownika Języka Polskiego "technika sterowania poglądami i zachowaniami ludzi polegająca na celowym, natarczywym, połączonym z manipulacją oddziaływaniu na zbiorowość" [5]. Pomimo iż najczęściej propaganda ma charakter polityczny nie jest to jedyne jej zastosowanie. Najstarszym przykładem pisemnej propagandy są opisy podbojów Dariusza Wielkiego datowane na rok 515 p.n.e. Od tego czasu w historii ludzkości można znaleźć wiele przypadków wykorzystania tego typu dezinformacji w krajach takich jak Starożytny Rzym, Niemcy podczas II Wojny Światowej, a nawet w dzisiejszych czasach Korea Północna. Propagandę można podzielić na 3 różne typy:

- biała propaganda - źródło pochodzenia informacji jest prawdziwe i podane,
- szara propaganda - źródło pochodzenia informacji jest dla odbiorcy nieznane i może się on jedynie domyślać,
- czarna propaganda - źródło pochodzenia informacji jest umyślnie sfałszowane w celu wyrządzenia szkody.

1.1. Sposoby rozprzestrzeniania fałszywych informacji

Wraz ze zmianami w sposobach rozprzestrzeniania informacji na świecie zmieniało się także podejście do tworzenia *fake newsów* w odpowiedni sposób oszukujących osoby, do których były one skierowane.

1.1.1. Gazety

Wykorzystanie *fake newsów* w gazetach miało na celu przyciągnąć uwagę, a co za tym idzie zwiększyć sprzedaż danej gazety. Stało się to na tyle popularne, że spowodowało narodziny

nowego pojęcia "żółtej prasy", czyli takiej, której działanie polega na zamieszczaniu w nagłówkach w pełni lub częściowo nieprawdziwych informacji, aby przyciągnąć uwagę przechodnia za wszelką cenę, nawet jeśli wiąże się to z utratą wiarygodności.

Dziennikarz Frank Luther Mott wyróżnia 5 cech charakteryzujących żółtą prasę [17]:

- napisane dużą czcionką straszące nagłówki na temat mniej ważnych wydarzeń,
- nadmierna liczba zdjęć i rysunków,
- zawarcie sfałszowanych wywiadów, mylących nagłówków, pseudonauki oraz nieprawdziwych informacji od ludzi podających się za ekspertów,
- dodanie w pełni kolorowych dodatków do gazet w niedzielę,
- stawianie siebie jako słabszego w walce przeciwko systemowi.

Rys. 1.2: Przykład żółtej prasy z roku 1993 Źródło: https://www.nytimes.com/

1.1.2. Telewizja

Wynalezienie telewizji na początku XX wieku zmieniło całkowicie sposób, w jaki ludzie pozyskiwali wiadomości ze świata. Aby pozyskać informacje na temat najnowszych wydarzeń, nie było konieczne kupienie gazety, a nawet wyjście z domu. W tym celu wystarczyło posiadać dostęp do telewizji i urządzenie do jej odbioru. Połączenie zarówno obrazu jak i dźwięku zmusiło osoby chcące oszukać swoich odbiorów do stworzenia nowych technik pozwalających w wiarygodny sposób przedstawić kłamstwo.

Przykładem osoby, która w znakomity sposób wykorzystała siłę daną mu przez telewizję był Edward Bernays nazywany "Ojcem public relations". W roku 1929 został on zatrudniony aby wypromować papierosy firmy "Lucky Strike", stworzona przez niego reklama ukazywała kobiety palące papierosy podczas marszu. Ponieważ kobiety palące były uznawane w tamtych czasach za temat taboo, autor reklamy nazwał ją w gazetach walką o prawa kobiet. Reklama ta spowodowała tak duże spopularyzowanie palenia papierosów, że właśnie Edwardowi Bernays przypisuje się ich dużą sprzedaż przez kolejne lata. Był on także osobą dzięki której w dzisiejszych czasach diamenty są uznawane za symbol miłości po tym jak został zatrudniony do wypromowania diamentów firmy "De Beers" [14].

1.1.3. Internet

Pojawienie się internetu wpłynęło na każdy element życia codziennego. Czynności takie jak komunikacja, rozrywka, a także rozprzestrzenianie informacji uległy zmianom tak wielkim, że ciężko wyobrazić sobie w jaki sposób działały one wcześniej.

Dzięki pojawieniu się takich portali społecznościowych jak Facebook, Twitter oraz Instagram każdy użytkownik internetu może opowiedzieć o swoich przemyśleniach lub wydarzeniach z życia każdej osobie zainteresowanej. Portale te pozwoliły nie tylko na przekazywanie informacji o sobie, ale także opowiadanie o wydarzeniach ze świata przez wszystkie chętne osoby. Wraz z rozwojem internetu stopniowo zwiększa się ilość osób czerpiących informacje na temat wydarzeń z portali społecznościowych i do dnia dzisiejszego w Ameryce wynosi 68% dorosłych osób.

Udostępnienie każdej osobie możliwości wypowiedzenia się doprowadziło do sytuacji, w której duża część informacji w internecie jest całkowicie fałszywa bądź w pewien sposób zmanipulowana poprzez osobę nieobiektywnie opisującą wydarzenia. Ogrom informacji można zauważyć na podstawie ilości potwierdzonych *fake newsów* związanych z wydarzeniami wokół wirusa COVID-19, których do czerwca 2020 jest aż 110. Niektóre z nich to [2]:

- szczepionka na koronawirusa jest ukrywana od marca,
- aspiryna jest lekarstwem na COVID-19,
- komary przenoszą koronawirusa,
- kraje bez sieci 5G są wolne od koronawirusa ,
- koronawirus nie zagraża uczestnikom zgromadzeń religijnych,
- koronawirus to środek do zmniejszenia populacji Ziemi.

Osoby oszukane *fake newsami* grają istotną rolę w dalszym ich rozprzestrzenianiu poprzez udostępnianie informacji swoim rozmówcom m.in. znajomym, rodzinie. Rozprzestrzenianie informacji jest to cecha internetu, która daje niespotykaną wcześniej efektywność oszukiwania dużej ilości ludzi w szybkim czasie.

Na powyższym obrazie ukazany jest post udostępniony na portalu Facebook z którego wynika, że jeżeli osoba jest w stanie wstrzymać oddech na 10 sekund to nie jest ona zarażona koronawirusem. Treść postu wskazywała, że metoda ta według ekspertów z Japonii miała być skuteczna. Jak się później okazało informacja ta była całkowicie fałszywa. Nie powstrzymało to jednak ponad 2,4 tysiąca ludzi przed udostępnieniem jej swoim rozmówcom. [6]

1.2. Sposoby ochrony przed nieprawdziwymi informacjami

Rozwój tak potężnego narzędzia jak internet spowodował, że fałszywe informacje stały się poważnym zagrożeniem w dzisiejszym świecie. Aby zapobiec oszukaniu dużej ilości społeczeństwa znaleziono różne sposoby na ochronę przed nieprawdziwymi informacjami, niektóre z nich to:

- IFLA "How to spot fake news" jest to stworzona przez międzynarodową instytucję reprezentującą interesy bibliotekarzy i pracowników informacji infografika przedstawiająca listę rzeczy, które należy zrobić podejrzewając, że jesteśmy oszukiwani.
  Czynności które są na niej zawarte to:
  - sprawdzenie źródła informacji,
  - dokładne przeczytanie treści,
  - sprawdzenie autora,
  - analiza odnośników,
  - sprawdzenie dat związanych,
  - upewnienie się, że informacja nie jest formą żartu,

Rys. 1.3: Przykład fałszywej informacji na portalu Facebook Źródło: https://www.facebook.com/

Rys. 1.4: Infografika stworzona przez IFLA Źródło: https://www.ifla.org/

  - obiektywna ocena informacji,
  - zapytanie ekspertów.
- strony internetowe stworzone w celu walki z dezinformacją istnieje wiele witryn gdzie eksperci sprawdzają nadesłane przez użytkowników informacje pod względem ich zgodności z prawdą. Przykładami takich portali są: *fakenews.pl, snopes.com, FactCheck.org, factchecker.in*
- grupy osób powołanych przez rząd do walki z fałszywymi informacjami, najlepszym przykładem takiej grupy są tzw. *Litewskie Elfy* - jest to grupa ochotników, którzy w wolnym czasie przeglądają fora internetowe oraz media społecznościowe sprawdzając znajdujące się w nich informacje. W przypadku znalezienia nieprawdziwej informacji osoby te informują administratora strony o problemie, co najczęściej prowadzi do jego rozwiązania. Nazwa grupy wzięła się stąd, że ludzie ci walczą z grupami powszechnie zwanymi *trollami*. Grupa ta zyskała na popularności w takim stopniu, że rozpoczęto organizację kolejnych oddziałów w innych krajach między innymi na Łotwie. [3]

Pomimo istnienia wielu sposobów ochrony przed fałszywymi informacjami ich ilość powoduje, że bardzo łatwo zostać oszukanym. Z tego powodu bardzo pomocne byłoby stworzenie oprogramowania pozwalającego na automatyczne rozpoznanie czy informacja jest prawdziwa. Implementacja takiego systemu w mediach społecznościowych jak Facebook lub Twitter pozwoliłaby na usunięcie kłamstw zanim trafiłyby one do użytkowników. Popularne w ostatnich czasach algorytmy uczenia maszynowego *ML* oraz analizy języka naturalnego *NLP* osiągają zaskakująco dobrą poprawność w klasyfikacji różnego rodzaju tekstów, więc ich wykorzystanie w rozwiązaniu takiego problemu mogłoby być bardzo pomocne.

1.3. Deep fake

Data pojawienia się technologii deep fake nie jest dokładnie znana jednak zaczęła zyskiwać na popularności w grudniu 2017 roku, kiedy użytkownik o pseudonimie “deepfakes” umieścił na portalu *Reddit* film pornograficzny w którym główną rolę grała aktorka Gal Gadot znana z filmu “Wonder Woman”. Aktorka jednak nigdy nie brała udziału w tego typu filmach a całe nagranie zostało stworzone wykorzystując algorytmy sztucznej inteligencji, która pozwoliła na zamianę twarzy dowolnego aktora z nagrania twarzą Gal Gadot w bardzo realistyczny sposób.

Sytuacja ta przyciągnęła zainteresowanie wielu użytkowników i już w styczniu 2018 roku pojawiła się aplikacja o nazwie “FakeApp”, dzięki której każdy może zamienić twarze znajdujące się na nagraniu na twarze kogoś innego. Powszechny dostęp oraz prostota w użytkowaniu powodują, że filmy “deep fake” stanowią niespotykane wcześniej zagrożenie mogąc oskarżyć dowolną osobę o wykonanie czynności, z którymi nie miała ona żadnego związku.

Aby zwrócić uwagę na niebezpieczeństwo firma buzzfeed w kwietniu 2018 roku umieściła na swoim kanale *Youtube* film, w którym Barack Obama opowiada o zagrożeniu płynącym z dezinformacji jednak słowa które wypowiada nie są naprawdę mówione przez niego, a pochodzą z nagrania aktora Jordana Peele, które następnie zostało podrobione techniką “deep fake”.

Filmy “deep fake” doprowadziły do sytuacji gdzie nie istnieje sposób przekazu informacji, który daje stu procentową wiarygodność, co w jeszcze większym stopniu zwiększa znaczenie odnalezienia uniwersalnej i efektywnej metody walki z dezinformacją.

Rys. 1.5: Klatka z filmu firmy Buzzfeed Źródło: https://www.youtube.com/

Rozdział 2

Uczenie maszyn

Uczenie maszynowe jest dziedziną algorytmów komputerowych, które automatycznie poprawiają swoją poprawność poprzez doświadczenie. Określa się je jako poddziedzinę sztucznej inteligencji. Algorytmy te pozwalają na zbudowanie matematycznego modelu na podstawie przykładowych danych nazywanych danymi treningowymi, co pozwala im na wykonywanie predykcji lub decyzji bez potrzeby ich dokładnego zaimplementowania przez programistę.

Uczenie maszynowe stosuje się do wielu zadań, między innymi do filtrowania poczty mailowej ze spamu, reklam internetowe, wykrywania twarzy na zdjęciach oraz nagraniach, a w przyszłości może pomóc w stworzeniu technologii takich jak autonomiczne samochody. Sam termin został spopularyzowany przez informatyka Arhura Samuela w roku 1959, był on autorem pierwszego działającego systemu tego typu, jego program automatycznie grał w warcaby i uczył się na podstawie poprzednich potyczek [15].

W dzisiejszych czasach implementacja takich systemów opiera się na wykonaniu następujących czynności:

- preprocessing danych:
  - czyszczenie,
  - uzupełnianie,
  - redukcja wymiarowości.
- ekstrakcja cech,
- uczenie modelu,
- sprawdzenie poprawności modelu.

Odpowiednie wykonanie etapu ekstrakcji cech jest kluczowym elementem do stworzenia poprawnego modelu, jest on opisany w kolejnym rozdziale pracy.

2.1. Rodzaje uczenia na podstawie przykładów

Algorytmy uczenia maszynowego uczące się na podstawie przykładów można podzielić na dwa główne rodzaje w zależności od problemów, które mają one rozwiązywać:

- uczenie nadzorowane - jest to najczęściej wykorzystywany rodzaj uczenia maszynowego, polega on na tym, że maszyna uczy się na podstawie przykładów zawartych w danych treningowych. Uczenie nadzorowane można porównać do nauczyciela i ucznia gdzie dane pełnią rolę nauczyciela, a program ucznia. Algorytmy tego typu potrafią znaleźć odpowiednie zależności na podstawie etykiet przypisanym danym, które następnie wykorzystują w celu predykcji wcześniej nie analizowanych danych. Ważnym zagadnieniem w przypadku uczenia nadzorowanego jest tak zwany overfitting polegający na przeuczeniu programu jednym zestawem

treningowym, przez co traci on umiejętność generalizacji problemu i nie jest w stanie poprawnie podejmować predykcji danych niewystarczająco podobnych do treningowych.
Przykładowe zastosowania:

- klasyfikacja - przewidywanie kategorii:
  - rozpoznawanie elementów na zdjęciu,
  - filtrowanie spamu w skrzynce mailowej.
- regresja - przewidywanie liczb:
  - przewidywanie trendów finansowych lub ekonomicznych,
  - prognozowanie pogody.

- uczenie nienadzorowane - w przeciwieństwie do uczenia nadzorowanego, opiera się na braku nauczyciela, a zadaniem maszyny jest znalezienie wzorców i zależności między analizowanymi obiektami samodzielnie. Dwie metody które są najczęściej wykorzystywane w uczeniu nienadzorowanym to:
  - analiza składowych głównych - polega na zmniejszaniu wymiarowości danych poprzez odnajdywanie, a następnie odrzucanie cech, które niosą ze sobą najmniejszą ilość informacji,
  - analiza skupień (Klasteryzacja) - pozwala na identyfikację oraz grupowanie danych podobnych, które nie są w żaden sposób oznaczone. Może także pomóc w odnalezieniu anomalii nie pasujących do żadnej z wydzielonych grup.

  Wykorzystanie tego typu algorytmów pozwala na badanie danych nieoznaczonych, które są znacznie częściej spotykane niż dane oznaczone.
  Przykładowe zastosowania:
  - redukcja wymiarów:
    - wizualizacja danych “big data”,
    - kompresja danych.
  - klasteryzacja:
    - spersonalizowane reklamy,
    - systemy rekomendacyjne.

2.2. Algorytmy klasyfikacji

Z problemem klasyfikacji można się spotkać wszędzie tam, gdzie wykorzystując zbiór zmiennych objaśniających należy wskazać wartość przyjmowaną przez zmienną modelowaną. W problemach klasyfikacji zmienna modelowana może przyjmować wartości binarne (klasyfikacja dwuklasowa) lub jedną z wielu etykiet (klasyfikacja wieloklasowa). Aby wybrać odpowiedni do rozwiązywanego problemu algorytm klasyfikacji należy wziąć pod uwagę cztery czynniki:

- złożoność czasowa - jak długo trwa uczenie oraz predykcja nowych danych,
- interpretowalność - jak łatwo można wytłumaczyć decyzję podjętą przez system,
- skalowalność - jak dużą ilość zasobów zużywa dany algorytm,
- czynnik ludzki - aby poprawnie ustawić parametry algorytmu potrzebna jest wiedza na temat jego działania, ponieważ poprawne ich ustawienie może mieć większe znaczenie dla efektywności niż dobór najlepszego algorytmu. Często lepiej jest wykorzystać algorytm znany, a nie optymalny.

W dalszej części rodziału opisane zostały najpopularniejsze metody klasyfikacji.

2.2.1. *k*-NN

Algorytm K najbliższych sąsiadów (K nearest neighbours) jest algorytmem stosowanym zarówno w problemach regresyjnych, jak i klasyfikacji. Kroki jego działania można opisać w następujący sposób:

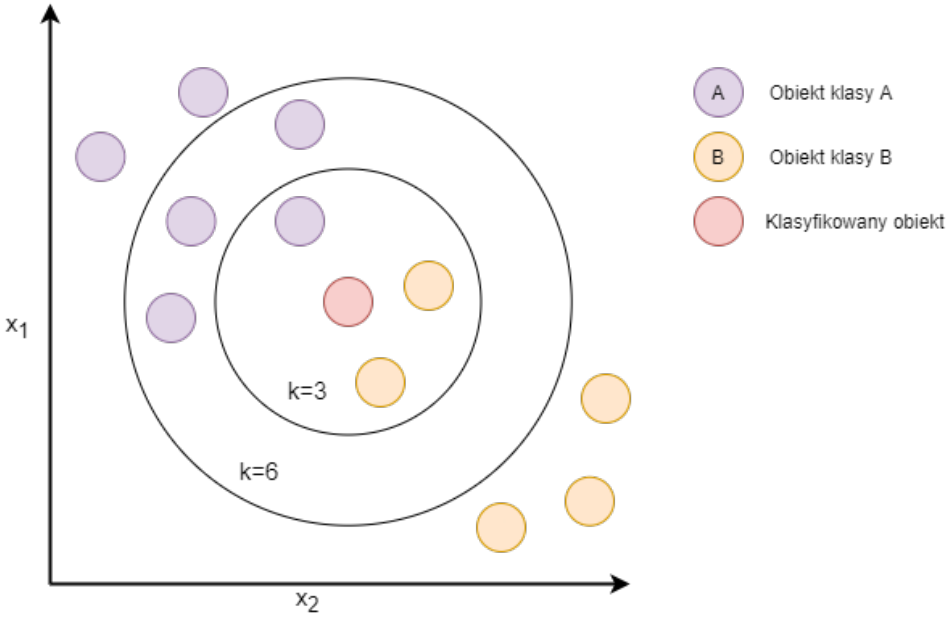
- umieszczenie wszystkich obiektów posiadających N cech jako punkty w N-wymiarowej przestrzeni,
- obliczenie odległości między obiektem, którego etykieta będzie przewidywana, a każdym innym obiektem,
- przypisanie do obiektu etykiety, którą posiada większość z K najbliższych obiektów.

Faza uczenia w *k*-NN polega wyłącznie na wczytaniu danych treningowych do pamięci, przez co jest ona bardzo szybka.

Najważniejszą kwestią w poprawnym implementowaniu algorytmu *k*-NN jest odpowiednie wybranie liczby K, której optymalna wartość będzie się różnić w zależności od danych. W problemach klasyfikacji często wybiera się K o wartości nieparzystej aby uniknąć problemu remisu podczas zliczania sąsiadów. W przypadku problemów regresyjnych zamiast wykonywać głosowanie, predykcję wykonuje się na podstawie średniej wartości K najbliższych sąsiadów [16].

Zalety	Wady
prosty do zrozumienia i interpretacji,	przechowuje wszystkie dane treningowe w pamięci, co skutkuje wysoką złożonością pamięciową,
szybkość fazy uczenia,	wrażliwy na ilość danych i nieistotne cechy,
nie wykonuje generalizacji danych, przez co jest efektywny w przypadku danych nieliniowych.	kosztowny obliczeniowo.

Tab. 2.1: Wady i zalety *k*-NN



Rys. 2.1: Graficzne przedstawienie algorytmu *k*-NN Źródło: Własne

2.2.2. SVM

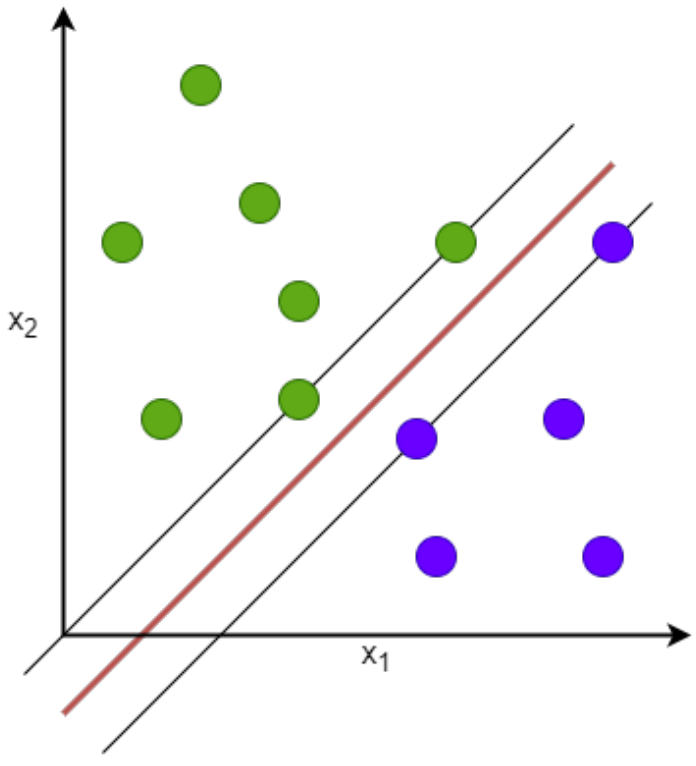
Klasyfikator SVM (Support vector machines) można wykorzystywać zarówno w problemach regresyjnych i klasyfikacyjnych. Służą one do klasyfikacji binarnej, co oznacza, że obiekty należy podzielić na dokładnie dwie klasy. SVM polega na znalezieniu takiej prostej lub płaszczyzny (w zależności od liczby cech), która w jak najlepszy sposób dzieli obiekty dwóch klas. Czasami idealny podział jest niemożliwy, z czym klasyfikator ten radzi sobie w jeden z dwóch sposobów:

- ignorowanie punktów, które uniemożliwiają podział,
- wykorzystanie tak zwanych kernel trick, które przekształcają dane do wyższego wymiaru w którym podział jest możliwy.

Aby odnaleźć optymalne rozwiązanie SVM skupia się na punktach jak najbardziej skrajnych obu klas, są one nazywane wektorami nośnymi. Zmiana ich położenia lub usunięcie całkowicie zmienia położenie płaszczyzny [16].

Zalety	Wady
efektywny w wielowymiarowych przestrzeniach,	nie jest on przystosowany do dużej ilości danych treningowych,
działa nawet w przypadku, gdzie liczba danych treningowych jest mniejsza od ilości ich cech,	nie radzi sobie dobrze jeżeli obiekty różnych klas nachodzą na siebie
przechowuje w pamięci tylko niewielką ilość danych treningowych.	trudne do zinterpretowania wyniki predykcji.

Tab. 2.2: Wady i zalety SVM



Rys. 2.2: Wizualizacja algorytmu SVM Źródło: Własne

2.2.3. MLP

Działanie algorytmu Perceptronu wielowarstwowego (Multi layered perceptron) opiera się na wykorzystaniu modelu sztucznego neuronu, który na podstawie określonej funkcji aktywacji oblicza na wyjściu pewną wartość na podstawie ważonych sum danych wejściowych.

Funkcje aktywacji dzieli się na:

- funkcje progowe z wyjściem binarnym,
- funkcje liniowe z wyjściem ciągłym,
- funkcje nieliniowe z wyjściem ciągłym.

W sieciach wielowarstwowych najczęściej wykorzystuje się funkcje nieliniowe ponieważ wykazują one największe zdolności do nauki.

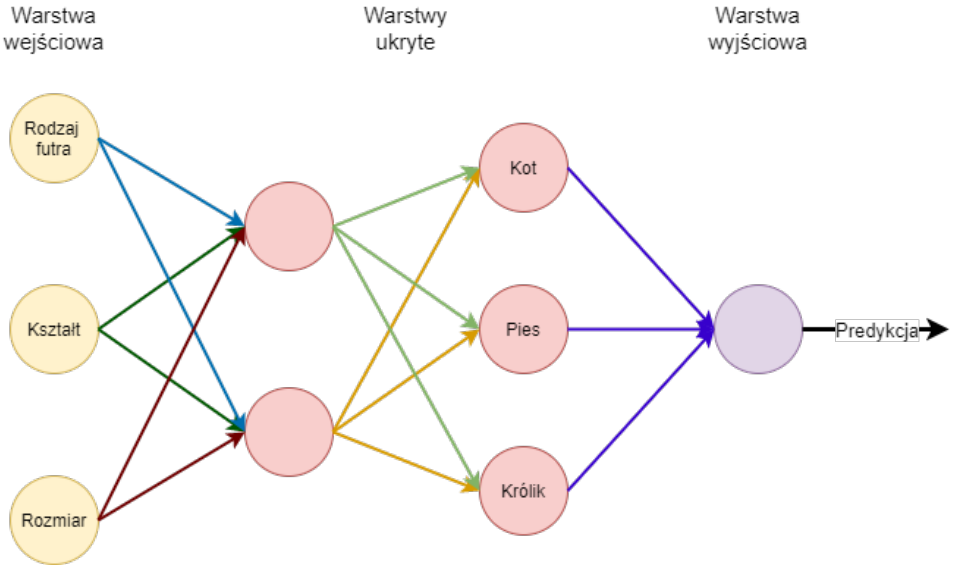
Perceptron wielowarstwowy składa się z trzech warstw sztucznych neuronów:

- warstwy wejściowej - są to neurony zapewniające całej sieci informacje, czyli zazwyczaj dane treningowe. Nie wykonują one żadnych obliczeń, a jedynie przekazują informacje do kolejnych warstw,
- warstwy ukrytej - wykonują one odpowiednie obliczenia zmieniając i przekazując informacje z warstwy wejściowej do wyjściowej. Sieć może składać się z dowolnej liczby warstw ukrytych,
- warstwy wyjsciowej - wykonują one ostatnie obliczenia na danych, a następnie zwracają wynik.

Uczenie takiej sieci polega na modyfikacji wag na wejściu neuronów, aby poprawić wyniki klasyfikacji. Podczas gdy perceptron jednowarstwowy, czyli nieposiadający żadnej warstwy ukrytej może nauczyć się wyłącznie funkcji liniowych perceptron wielowarstwowy pozwala na nauczenie się skomplikowanych funkcji nieliniowych [16].

Zalety	Wady
szybkie wykonywanie predykcji po nauczeniu modelu,	brak możliwości interpretacji wyniku predykcji,
efektywny dla nieliniowych danych posiadających wiele cech takich jak zdjęcia,	uczenie bardzo złożone obliczeniowo, co skutkuje długim czasem uczenia,
działają najlepiej dla dużej ilości danych treningowych.	użytkownik ma niewielki wpływ na działanie sieci.

Tab. 2.3: Wady i zalety MLP



Rys. 2.3: Graficzne przedstawienie perceptronu wielowarstwowego Źródło: Własne

2.2.4. Decision trees

Algorytm Decision trees polega na stworzeniu drzewa decyzyjnego, w którym korzeń i węzły są poszczególnymi cechami zbioru danych, a liście odpowiadają klasom, które tym danym należy przypisać, ostatnim elementem są krawędzie, którymi reprezentuje się wartości cech. Kroki algorytmu są następujące:

1. wybranie najlepszej cechy,
2. dodanie gałęzi odpowiadających poszczególnym wartościom wybranej cechy,
3. podział zbioru danych na podzbiory zgodnie z wartościami cechy,
4. jeżeli wszystkie elementy podzbiorów należą do tej samej klasy zakończenie gałęzi liściem. W przeciwnym wypadku powtórzenie kroków od 1 do 4 dla każdego podzbioru.

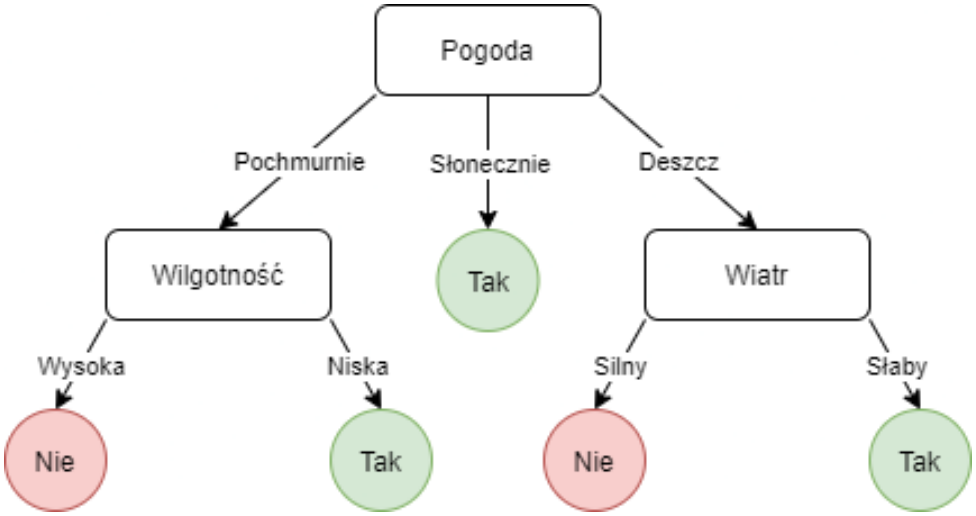
Aby dokonać optymalnego podziału algorytm musi wybrać cechy powodujące jak najlepsze podzielenie różnych klas, a więc niosące ze sobą największą ilość informacji. Wykorzystuje się w tym celu entropię, której wartość pozwala określić średnią ilość informacji niesioną przez poszczególne cechy [16].

$$H(x) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i). \tag{2.1}$$

Wzór na entropię 2.1, gdzie $p(x_i)$ jest prawdopodobieństwem wybrania klasy x_i

Zalety	Wady
łatwe do zinterpretowania wyniki oraz wygodne do zwizualizowania,	wrażliwy na przeuczenie,
automatyczna selekcja ważnych cech,	potrzeba stworzenia nowego drzewa podczas dodawania nowych danych,
obecność cech zbędnych nie ma wpływu na poprawność,	małe zmiany w danych treningowych maja duży wpływ na predykcję.
szybkie wykonywanie predykcji.	

Tab. 2.4: Wady i zalety Drzew decyzyjnych



Rys. 2.4: Przykładowe drzewo decyzyjne podejmujące decyzję czy wyjść na zewnątrz Źródło: Własne

2.2.5. RF

Algorytm lasów losowych (Random forest) w skrócie RF, polega na stworzeniu lasu składającego się z drzew decyzjnych, które są budowane w następujący sposób:

- z danych treningowych wybrane zostają losowe obiekty (może nastąpić wybranie tego samego obiektu dwukrotnie),
- wybrane obiekty stają się nowym zbiorem treningowym,
- w nowym zbiorze wybrana zostaje określona liczba losowych cech,
- tworzone zostaje drzewo decyzjne oparte tylko i wyłącznie na wybranych wcześniej cechach
- poprzednie kroki są powtarzane do czasu stworzenia określonej liczby drzew zazwyczaj jest to liczba między 64 a 128 drzewami.

Po stworzeniu wielu drzew decyzyjnych model jest gotowy aby wykonywać predykcje. Predykcja w algorytmie RF jest wykonywana na podstawie głosowania przez każde stworzone drzewo na klasy, w głosowaniu tym wybrana zostaje klasa, na którą głosowała większość drzew [12].

Zalety	Wady
odporny na overfitting,	ciężkie do zinterpretowania wyniki z powodu ilości drzew,
pojawienie się nowych danych nie wymaga tworzenia modelu od nowa,	długi czas uczenia,
zachowuje poprawność w przypadku brakujących danych.	nie sprawdza się dobrze w przypadku problemów regresyjnych

Tab. 2.5: Wady i zalety RF

2.2.6. Naive Bayes

Naiwny klasyfikator Bayesowski (Naive Bayes) jest probabilistycznym klasyfikatorem, który zakłada wzajemną niezależność wszystkich cech, stąd naiwność algorytmu. Stworzony przez niego model opiera się na obliczeniu prawdopodobieństw wystąpienia poszczególnych cech pod warunkiem, że występują one w danej klasie. Pozwala to przy użyciu twierdzenia Bayesa 2.2

obliczyć prawdopodobieństwa warunkowe określające z jak dużą pewnością występująca kombinacja cech prowadzi do wystąpienia różnych klas, na podstawie czego algorytm wykonuje predykcję [16].

$$P(A|B) = \frac{P(A|B)P(A)}{P(B)} \tag{2.2}$$

gdzie P(A) i P(B) prawdopodobieństwa wystąpienia zdarzeń A i B, P(A|B) prawdopodobieństwo wystąpienia zdarzenia A pod warunkiem wystąpienia zdarzenia B.

Algorytm ten jest często wykorzystywany w systemach czasu rzeczywistego, ponieważ działa bardzo szybko jednak jego poprawność w porównaniu do bardziej skomplikowanych algorytmów jest mniejsza.

Zalety	Wady
bardzo szybki nawet w przypadku przewidywania wieloklasowego, działa nawet dla małej ilości danych, dobra poprawność w przypadku klasyfikacji danych wielowymiarowych jak na przykład w przypadku klasyfikacji tekstu.	jeżeli pewna cecha nie pojawi się w danych treningowych, a istnieje w danych testowych algorytm nie będzie w stanie wykonać jego predykcji, przyjmuje niezależność cech co w rzeczywistości jest prawie niespotykane, mniejsza poprawność w porównaniu z bardziej skomplikowanymi algorytmami.

Tab. 2.6: Wady i zalety Naive Bayes

2.3. Zagrożenia

Ogromny rozwój uczenia maszynowego i sztucznej inteligencji spowodował, że znalazły one wiele zastosowań w różnych dziedzinach życia codziennego. Powoduje to, że wiele osób skupia się na pozytywach związanych z tymi technologiami, jednakże niektórzy dostrzegają ogromne zagrożenia płynące z niewłaściwego wykorzystania ich i katastrofalne skutki, do których mogą doprowadzić. Wśród takich osób znajdują się naukowcy tacy jak jak Stephen Hawking, Elon Musk, Steve Wozniak i Bill Gates. Według Elona Muska sztuczna inteligencja niesie ze sobą większe zagrożenie niż bomby atomowe [8].

Dwa główne sposoby w jaki sztuczna inteligencja może powodować zagrożenie to:

- AI zaprogramowane do czynienia krzywdy, czyli na przykład wykorzystanie uczenia maszynowego w takich narzędziach jak broń, która w niewłaściwych rękach stanowiłaby zagrożenie dla całej ludzkości,
- AI zaprogramowane w dobrych celach, jednak odnajdujące krzywdzący innych optymalny sposób osiągnięcia sukcesu, może to być związane ze złym przekazaniem celu jaki maszyna ma osiągnąć przez programistę. Przykładowo mógłby to być samojezdny samochód, który podczas jazdy zwraca uwagę tylko i wyłącznie na to by jak najszybciej dojechać do celu podróży. Samochód taki nie przestrzegał by istniejących praw drogowych ponieważ ograniczałyby one jego prędkość.

Przykłady te pokazują, że zagrożenie nie płynie z sytuacji gdzie maszyny odwracają się przeciwko ludzkości, a z sytuacji w której maszyny wykonują poświęcone im zadania za dobrze i nie w sposób przewidziany przez stwórcę systemu.

Istotnym problemem jest także bezrobocie będące skutkiem zastępowania ludzi maszynami, ponieważ są one znacznie tańsze w utrzymaniu niż pracownicy, a z wykorzystaniem uczenia maszynowego mogą one opanować niektóre zadania lepiej od ludzi. Rozwiązaniem na takie problemy mogłyby być obowiązkowo przestrzegane zasady bezpiecznego korzystania ze sztucznej inteligencji, których złamanie wiązałoby się z odpowiedzialnością karną.

Rozdział 3

Przetwarzanie języków naturalnych

Systemy przetwarzania języków naturalnych (Natural language processing) nazywane w skrócie NLP, oznaczają systemy mogące zrozumieć mowę i pismo ludzkie w takiej formie jaką ludzie posługują się na co dzień. Programy takie mogą wykonywać zadania od zliczania częstotliwości występowania danego słowa w tekście do automatycznego pisania artykułów.

Głównym problemem w tworzeniu tego typu systemów jest to, że do komunikacji z komputerem zazwyczaj potrzebne jest posługiwanie się precyzyjnymi komendami danego języka programowania, mowa ludzka jednak nie zawsze jest precyzyjna i jej znaczenie może różnić się w zależności od kontekstu czy różnego rodzaju regionalnych dialektów. Systemy NLP są często wykorzystywane w takich celach jak:

- asystenci głosowi na przykład (Siri, Alexa, Cortana) - są to urządzenia, które wykonują komendy wypowiedziane w ich kierunku przez użytkownika,
- wyodrębnianie ważnych informacji z tekstu w celu późniejszej analizy,
- analiza sentymentu, czyli wnioskowanie na podstawie tekstu opinii użytkowników na dany temat,
- sprawdzanie błędów ortograficznych,
- tłumaczenie tekstu na inne języki,
- chatboty wykorzystywane przez wiele firm w celu posiadania całodobowej zautomatyzowanej obsługi klienta.

Rozwój NLP ma bardzo duże znaczenie dla osób niepełnosprawnych, które często tylko dzięki ich pomocy są w stanie nawiązać interakcję z technologią pozwalającą im na znaczne podniesienie jakości życia. [18]

3.1. Przygotowywanie danych tekstowych

Aby ułatwić analizę ogromnej ilości danych tekstowych potrzebnych do poprawnego nauczenia systemu NLP, wykonuje się na nich różnego rodzaju operacje. Operacje te powodują, że tekst nie traci swoich najważniejszych cech, natomiast znacznie zmniejsza się moc obliczeniowa potrzebna do nauczenia algorytmów wykonywanych na nim. Najczęściej wykorzystywanymi tego typu operacjami są [7]:

- zamiana wszystkich dużych liter na małe,
- usunięcie znaków specjalnych,
- usunięcie tak zwanych “Stop words” - są to bardzo często występujące w danym języku słowa, które zazwyczaj nie wnoszą istotnych dla analizy informacji,

- tokenizacja - polega na podziale tekstu na mniejsze części zwane tokenami. W przypadku dużych bloków tekstu może to być podział na zdania, a w przypadku zdań podział na słowa itd.,
- lematyzacja - oznacza ona sprowadzenie grupy wyrazów stanowiących odmianę danego zwrotu do wspólnej postaci, co pozwala na traktowanie ich jako to samo słowo,
- stemming - jest to proces usunięcia końcówki fleksyjnej pozostawiając tylko temat czyli nośnik znaczenia wyrazu.

Wykonanie wybranych operacji na tekście daje na wyjściu skrócone dane tekstowe, które można następnie przeanalizować lub wykonać na nich wektoryzację, co pozwala na wykorzystanie ich w różnych algorytmach uczenia maszynowego.

3.2. Wektoryzacja

Z uwagi na fakt, że algorytmy sztucznej inteligencji potrafią uczyć się tylko z danych przedstawionych w formie numerycznej, aby móc wykorzystać je w kombinacji z NLP potrzebny jest pewien sposób zamiany formy tekstowej na liczbową, tak by straciły one jak najmniej swoich najważniejszych cech, a zarazem były czytelne dla maszyny.

Operacje takie nazywa się wektoryzacją tekstu, ponieważ reprezentacja liczbowa będąca ich wynikiem to najczęściej wektory. Wybór poprawnego sposobu zamiany tekstu może mieć ogromne znaczenie dla efektywności nauczonego modelu.

Jedne z najpopularniejszych metod wektoryzacji to [13]:

- Bag of words jest to metoda wektoryzacji, w której każdemu unikalnemu symbolowi z tekstu przypisuje się liczbę odpowiadającej jego ilości w analizowanym tekście. Symbolami mogą być całe zdania, słowa bądź ngramy. Metoda ta w żaden sposób nie zachowuje informacji o porządku ani kontekście występujących symboli, a jedynie o częstotliwości ich występowania. Wektor wynikowy metody “Bag of words” ma wymiar równy liczbie unikalnych symboli w tekście, przez co przy analizie dużej ilości dokumentów ma dużą złożoność pamięciową aby naprawić ten problem najpierw na danych wykonuje się metody przygotowywania danych tekstowych.

Ala ma kota, a kot ma Alę

Ala	ma	kota	a	kot	Alę
1	2	1	1	1	1

Rys. 3.1: Przykład wektoryzacji zdania za pomocą metody Bag of words Źródło: Własne

- TFIDF (ang. Term frequency inverse document frequency) jest to metoda przypisująca każdemu symbolowi jego wagę w kontekście analizowanych dokumentów. Aby obliczyć tę wagę potrzebne są dwa elementy:
  1. częstość symboli, którą uzyskuje się dzieląc liczbę wystąpień symbolu przez liczbę symboli w całym dokumencie,
  2. odwrotną częstość w dokumentach.

Po obliczeniu obu tych wartości oblicza się tzw. TFIDF score, której wartość określa jak ważny jest dany symbol dla dokumentu w kontekście wykonywanej analizy. Wynik ten oblicza się na podstawie wzoru 3.1, gdzie

- x - symbol,
- y - dokument,
- TF - częstość symbolu,
- N - liczba wszystkich dokumentów,
- df - liczba dokumentów, w których pojawił się symbol x.

$$TFIDF_{x,y} = TF_{x,y} * \log \frac{N}{df_x} \tag{3.1}$$

Rozdział 4

Projekt i implementacja systemu

Do wykonania badań efektywności algorytmów uczenia maszynowego w rozpoznawaniu fałszywych informacji potrzebne było stworzenie systemu, który w prosty sposób dla każdego z badanych algorytmów wykonałby operację uczenia go na danych treningowych, a następnie sprawdził jak dobrze przewiduje on przypadki ze zbioru testowego. System musi w odpowiedni sposób przygotować dane tekstowe przy użyciu metod takich jak zamiana dużych liter na małe, usuwanie znaków interpunkcyjnych itd. Ważną funkcjonalnością oprogramowania jest też to by dzielił on dane tekstowe na różnej długości ngramy.

System pozwoli na zbadanie wyników algorytmów takich jak poprawność predykcji, odchylenie standardowe poprawności oraz czasy trwania poszczególnych faz.

4.1. Wykorzystane technologie

Najpopularniejszymi językami programowania wykorzystywanymi do tworzenia modeli uczenia maszynowego są:

- Python,
- R,
- Lisp,
- Prolog.

Do wykonania systemu został wybrany język Python w wersji 3.8.5, jest to interpretowalny język wysokiego poziomu powstały w roku 1991. Został on wybrany ponieważ posiada dużą ilość bibliotek takich jak:

- Scikit-learn,
- PySpark,
- PyTorch,
- TensorFlow,
- Keras.

Biblioteki te znacznie ułatwiają korzystanie z możliwości uczenia maszynowego. W pracy zostanie wykorzystana biblioteka scikit-learn w wersji 0.23, która zawiera wszystkie testowane algorytmy i pozwala na bardzo wygodną ich implementację. Posiada także obie badane metody wektoryzacji tekstu, czyli Bag of words oraz TFIDF. Jest ona udostępniana na zasadach licencji BSD [4]. Licencja ta pozwala na użytkowanie i redystrybucję kodu zarówno zmodyfikowanego, jak i oryginalnego.

Ostatnią biblioteką potrzebną do stworzenia systemu była Natural Language Toolkit nazywana w skrócie NLTK. Jest to biblioteka służąca do przetwarzania języków naturalnych, która pozwala na proste przygotowanie tekstu do jego analizy

4.2. Wymagania funkcjonalne

Aplikacja powinna spełniać następujące wymagania funkcjonalne:

- wczytywać dane z zewnętrznego pliku w formacie csv,
- w odpowiedni sposób przygotowywać dane,
- dzielić dane na treningowe oraz testowe,
- wykonywać uczenie badanych algorytmów na danych treningowych,
- testować nauczone modele na danych testowych,
- zapisywać wyniki badań do pliku w formacie csv.

Są to funkcjonalności kluczowe aby system pozwalał wykonywać ważne dla osiągnięcia celu badania.

4.3. Implementacja

Implementacja składa się z dwóch funkcji służących do przygotowywania danych tekstowych do późniejszej analizy oraz jednej pętli głównej, która wykonuje wszystkie badania, a następnie zapisuje je do pliku w formacie csv.

W systemie tworzony jest także obiekt wielowymiarowej mapy, który po zakończeniu wykonywania programu przechowuje wszystkie nauczone modele wektoryzacji oraz uczenia maszynowego. Zapisanie go za pomocą biblioteki takiej jak *Pickle* pozwala w prosty sposób użyć i przetestować modele na dowolnych danych testowych.

```
def basicPreparation(fileName):
    file = pd.read_csv(fileName)
    label_encoder = preprocessing.LabelEncoder()
    file['label'] = label_encoder.fit_transform
        (file['label'])
    file = file.applymap
        (lambda s: s.lower() if type(s) == str else s)
    return (file['text'], file['label'])
```

Listing 4.1: Funkcja przygotowywująca dane pobrane z pliku csv

Pierwsza funkcja *basicPreparation* przyjmuje tylko jeden argument, którym jest nazwa pliku. Metoda ta wczytuje do pamięci dane z pliku csv a następnie wykorzystuje tzw. LabelEncoder, który pozwala na wykonanie zamiany etykiet w zbiorze danych na formę numeryczną, w tym przypadku zamienia ona wartości REAL i FAKE na 0 i 1. Ostatnią akcją wykonywaną przez funkcję jest zamiana wszystkich liter zawartych w danych na małe.

Funkcja ta zwraca krotkę zawierającą dwa elementy:

- listę artykułów,
- listę etykiet.

```
def dataPreprocessing(articles, labels):
    deletionIndexes = []
    for articleIndex in range(len(articles)):
        articles[articleIndex] =
            ''.join(w+' ' for w in articles[articleIndex]
            .split(' ') if not w in stop_words and w != '')
        articles[articleIndex] =
            re.sub(r'[^a-zA-Z]+', ' ', articles[articleIndex])
        articleLength = len(articles[articleIndex])
```

```
        if(articleLength == 0
            or articleLength < 500
            or articleLength > 5000):
            deletionIndexes.append(articleIndex)
    articles = articles.drop(deletionIndexes, axis=0)
    labels = labels.drop(deletionIndexes, axis=0)
    return (articles, labels)
```

Listing 4.2: Funkcja przygotowywująca dane tekstowe

Drugą funkcją znajdującą się w programie jest *dataPreprocessing*. Posiada ona dwa argumenty:

- articles - lista artykułów,
- labels - lista etykiet.

Funkcja ta wykonuje przygotowanie tekstu do późniejszej wektoryzacji. Wykonuje ona kolejno operacje:

- usunięcia Stop words z tekstu każdego artykułu,
- usunięcia wszystkich znaków nie będących literami,
- usunięcia ze zbioru danych artykułów o długości poniżej 500 znaków oraz powyżej 5000 znaków, ponieważ przekazują one niepotrzebną ilość informacji.

Funkcja ta zwraca dokładnie taką samą krotkę jak *basicPreparation*.

Główna pętla programu wykonuje kolejno następujące operacje:

- wytrenowanie modelu wektoryzatora na danych treningowych,
- wektoryzacja danych treningowych i testowych za pomocą nauczonego wcześniej modelu,
- normalizacja cech za pomocą StandardScaler,
- trening algorytmu na cechach treningowych funkcją *fit*,
- obliczenie efektywności funkcją *score* na danych testowych,
- zapisanie wyników do pliku csv o tytule odpowiadającym nazwie algorytmu uczenia maszynowego.

Aby obliczyć czas trwania faz uczenia i predykcji wykorzystano bibliotekę time w celu zapisania czasu przed i po wykonaniu każdej z faz, a następnie odjęciu od siebie tych wartości.

Rozdział 5

Ocena eksperymentalna

5.1. Cel Badań

Celem badań było sprawdzenie jak popularne algorytmy uczenia maszynowego radzą sobie z rozwiązaniem problemu klasyfikacji informacji jako prawdziwe lub nieprawdziwe. W celu sprawdzenia ich efektywności należało zbadać takie cechy jak:

- czas fazy uczenia,
- czas fazy predykcji,
- poprawność modelu (*accuracy*)
- odchylenie standardowe poprawności modelu.

Poprawność oraz odchylenie standardowe modelu została zbadana metodą K-krotnej walidacji krzyżowej. Metoda ta polega na podzieleniu zbioru danych na wybraną liczbę podzbiorów, a następnie wykorzystanie jednego z nich jako dane treningowe a resztę jako testowe. Operację tą wykonuje się tyle razy ile wynosi liczba podzbiorów, końcowe wyniki tej metody to średnia poprawność badań oraz odchylenie standardowe. Wartość odchylenia standardowego pozwala określić jak bardzo poprawność danej metody jest zależna od danych treningowych. W badaniach głównym sprawdzanym elementem było to, jak zmiana długości ngramów, na które podzielony został tekst przed wektoryzacją wpływa na wyniki algorytmów i odnalezienie takiego rozmiaru, który daje najlepsze wyniki.

Do badania wybrano pięć popularnych algorytmów uczenia maszynowego:

- *k*-NN - K najbliższych sąsiadów,
- SVM - maszyna wektorów nośnych,
- Naive Bayes,
- RF - las losowy,
- MLP - wielowarstwowy perceptron.

5.2. Warunki przeprowadzonego eksperymentu

Badania zostały wykonane na komputerze stacjonarnym o następującej specyfikacji:

- system operacyjny Windows 10 Pro w wersji 10.0.19041,
- procesor AMD Ryzen 5 3600,
- pamięć RAM 16 GB.

Wszystkie badania były wykonywane w jednakowych warunkach, aby można było je w prosty sposób porównywać.

Dane badawcze to zbiór pod nazwą "ISOT Fake News Dataset" przygotowany przez uczelnię w Kanadzie "Univerity of Victoria". Zawiera on 12600 artykułów prawdziwych pochodzących ze strony internetowej Reuters.com oraz 12600 artykułów nieprawdziwych zebranych z niewiarygodnych źródeł oznaczonych przez organizację do sprawdzania faktów Politifact. Tematyka artykułów to głównie polityka i wiadomości ze świata. Teksty zawarte w zbiorze zostały wstępnie przygotowane, jednak błędy znajdujące się w nieprawdziwych artykułach pozostały. Średnia długość artykułów ze zbioru wynosi 2469 znaków a najdłuższy z nich składa się z 51794. Ponieważ artykuł o takim rozmiarze niesie ze sobą zbyt dużą ilość informacji, przed wykonaniem badań zostają usunięte artykuły posiadające ponad 5000 znaków. Zbiór został pobrany w dniu 16 czerwca 2020 z witryny znajdującej się pod adresem `https://www.uvic.ca/`. [9]

Algorytmy *k*-NN, RF oraz Naive Bayes zostały zbadane na rozmiarach ngramów od 1 do 10, natomiast MLP i SVC od 1 do 5 z powodu dużej złożoności obliczeniowej przy większym rozmiarze. Podczas wykonywania walidacji krzyżowej zbiór danych był dzielony na 5 podzbiorów na każdym z nich wykonywana była kolejno wektoryzacja oraz redukcja wymiarowości obiektem StandardScaler z biblioteki scikit-learn.

5.3. Wyniki

W opisanych poniżej wynikach, każdy z algorytmów został zbadany pod kątem trzech cech.

Tab. 5.1: Wyniki algorytmu *k*-NN wektoryzacja metodą Bag of words

Rozmiar ngramu	1	2	3	4	5	6	7	8	9	10
Poprawność	73.54%	80.07%	60.5%	48.63%	46.81%	46.61%	46.6%	55.38%	52.38%	52.36%
Odchylenie standardowe	5.12%	5.13%	4.83%	1.71%	0.39%	0.47%	0.49%	19.39%	5.56%	5.54%
Czas fazy uczenia	0.003s	0.022s	0.059s	0.099s	0.102s	0.121s	0.126s	0.129s	0.122s	0.129s
Czas fazy predykcji	27.221s	180.565s	312.157s	221.672s	137.788s	91.407s	67.833s	54.803s	45.88s	40.02s

Tab. 5.2: Wyniki algorytmu *k*-NN wektoryzacja metodą TFIDF

Rozmiar ngramu	1	2	3	4	5	6	7	8	9	10
Poprawność	76.0%	69.91%	60.38%	53.64%	46.74%	53.47%	52.37%	52.37%	52.37%	55.72%
Odchylenie standardowe	1.92%	5.34%	6.14%	9.74%	0.6%	6.82%	5.48%	5.46%	5.46%	7.92%
Czas fazy uczenia	0.003s	0.024s	0.067s	0.092s	0.112s	0.111s	0.122s	0.123s	0.127s	0.129s
Czas fazy predykcji	27.168s	186.84s	321.7s	220.716s	141.115s	93.327s	68.613s	46.536s	47.467s	37.644s

Tab. 5.3: Wyniki algorytmu RF wektoryzacja metodą Bag of words

Rozmiar ngramu	1	2	3	4	5	6	7	8	9	10
Poprawność	83.41%	97.72%	99.29%	99.5%	99.22%	98.87%	98.4%	98.12%	97.88%	97.52%
Odchylenie standardowe	2.6%	0.77%	0.8%	0.68%	1.22%	1.67%	2.25%	2.82%	2.7%	2.96%
Czas fazy uczenia	23.56s	47.484s	48.267s	45.867s	52.639s	73.57s	108.591s	143.566s	187.709s	225.265s
Czas fazy predykcji	0.234s	0.764s	2.587s	5.095s	6.324s	8.315s	15.628s	19.665s	23.49s	25.823s

Tab. 5.4: Wyniki algorytmu RF wektoryzacja metodą TFIDF

Rozmiar ngramu	1	2	3	4	5	6	7	8	9	10
Poprawność	83.24%	97.76%	99.29%	99.51%	99.11%	98.77%	98.47%	98.17%	97.87%	97.32%
Odchylenie standardowe	2.76%	0.66%	0.44%	0.65%	1.15%	1.84%	2.26%	2.61%	3.01%	3.52%
Czas fazy uczenia	26.536s	57.014s	53.246s	47.681s	48.82s	63.392s	92.896s	132.436s	175.463s	217.02s
Czas fazy predykcji	0.24s	0.729s	3.027s	5.021s	6.554s	7.893s	14.047s	20.172s	24.588s	26.493s

Tab. 5.5: Wyniki algorytmu Naive Bayes wektoryzacja metodą Bag of words

Rozmiar ngramu	1	2	3	4	5	6	7	8	9	10
Poprawność	67.23%	77.99%	81.75%	88.32%	91.05%	92.79%	93.91%	94.22%	94.38%	94.25%
Odchylenie standardowe	7.65%	10.74%	10.58%	2.43%	3.15%	3.33%	2.65%	2.5%	2.0%	1.54%
Czas fazy uczenia	0.005s	0.031s	0.102s	0.147s	0.201s	0.377s	0.551s	0.692s	0.837s	1.027s
Czas fazy predykcji	0.003s	0.016s	0.047s	0.072s	0.095s	0.182s	0.299s	0.362s	0.382s	0.41s

Tab. 5.6: Wyniki algorytmu Naive Bayes wektoryzacja metodą TFIDF

Rozmiar ngramu	1	2	3	4	5	6	7	8	9	10
Poprawność	73.96%	80.52%	83.36%	87.9%	90.54%	92.53%	93.68%	94.04%	94.15%	93.95%
Odchylenie standardowe	7.8%	10.69%	10.28%	1.97%	3.79%	3.46%	2.23%	2.05%	1.91%	1.67%
Czas fazy uczenia	0.005s	0.033s	0.105s	0.159s	0.193s	0.293s	0.504s	0.675s	0.853s	1.02s
Czas fazy predykcji	0.002s	0.017s	0.05s	0.077s	0.09s	0.145s	0.319s	0.343s	0.406s	0.407s

Tab. 5.7: Wyniki algorytmu SVC wektoryzacja metodą Bag of words

Rozmiar ngramu	1	2	3	4	5
Poprawność	79.48%	95.08%	94.88%	95.43%	96.11%
Odchylenie standardowe	7.06%	2.02%	3.71%	2.11%	1.56%
Czas fazy uczenia	14.175s	319.6s	2779.462s	7282.055s	10144.174s
Czas fazy predykcji	8.18s	127.618s	716.666s	1502.585s	2534.03s

Tab. 5.8: Wyniki algorytmu SVC wektoryzacja metodą TFIDF

Rozmiar ngramu	1	2	3	4	5
Poprawność	77.32%	95.35%	96.54%	96.81%	97.14%
Odchylenie standardowe	6.62%	1.77%	2.46%	1.85%	1.41%
Czas fazy uczenia	14.046s	220.009s	3068.672s	8322.766s	10384.898s
Czas fazy predykcji	8.008s	106.538s	751.689s	1594.27s	2286.608s

Tab. 5.9: Wyniki algorytmu MLP wektoryzacja metodą Bag of words

Rozmiar ngramu	1	2	3	4	5
Poprawność	79.6%	96.69%	97.91%	98.18%	97.76%
Odchylenie standardowe	4.71%	0.71%	0.5%	0.63%	0.75%
Czas fazy uczenia	32.655s	22.982s	115.108s	544.161s	2038.845s
Czas fazy predykcji	0.039s	0.11s	0.373s	1.085s	2.444s

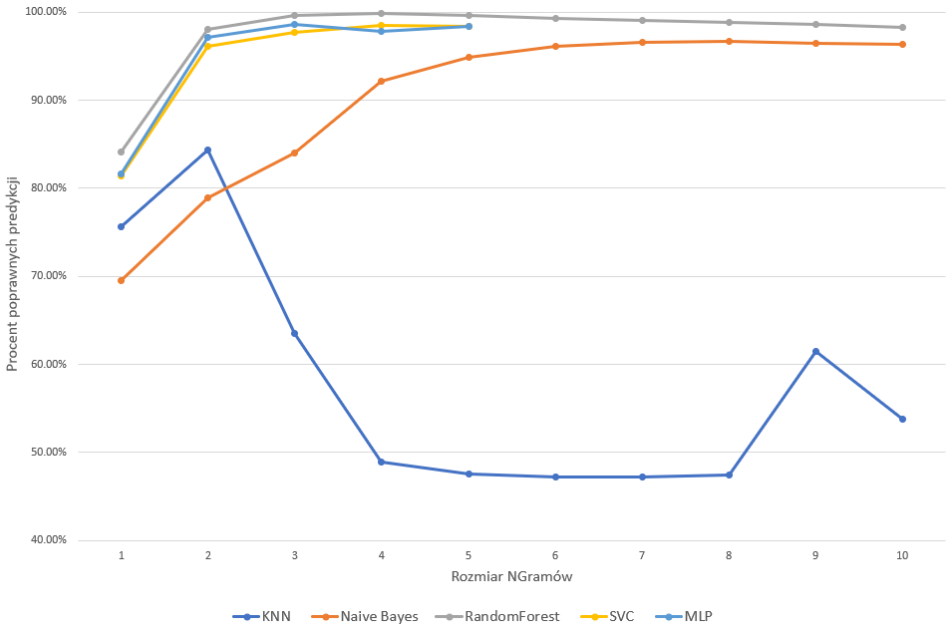
Tab. 5.10: Wyniki algorytmu MLP wektoryzacja metodą TFIDF

Rozmiar ngramu	1	2	3	4	5
Poprawność	79.8%	97.43%	98.6%	98.69%	98.09%
Odchylenie standardowe	4.54%	0.67%	0.62%	0.58%	0.7%
Czas fazy uczenia	16.176s	22.159s	111.424s	520.082s	1926.648s
Czas fazy predykcji	0.034s	0.113s	0.38s	0.911s	2.133s

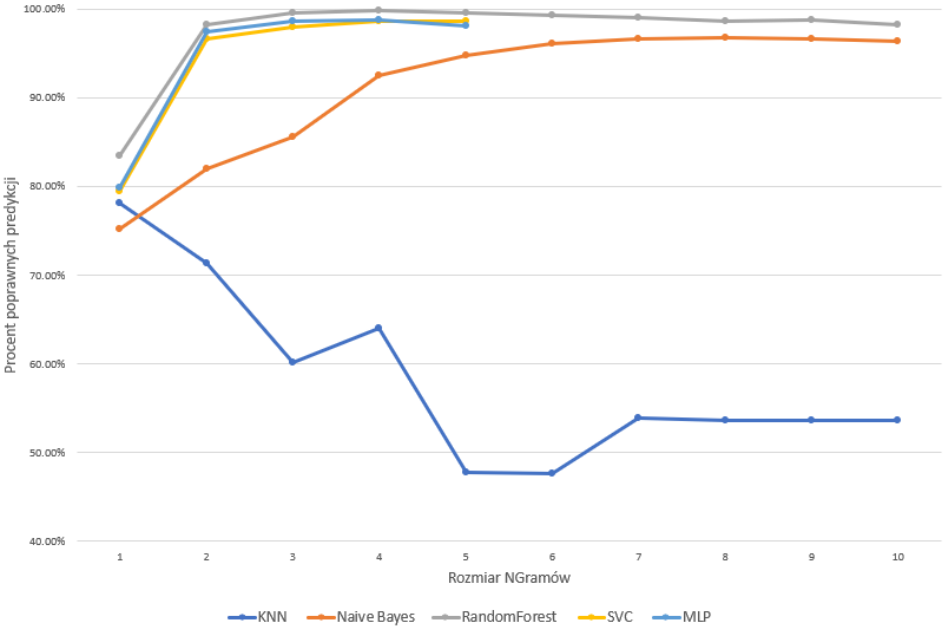
5.4. Analiza wyników wraz z oceną statystyczną

Wyniki zostały przeanalizowane pod kątem trzech badanych cech:

- Poprawność



Rys. 5.1: Poprawność algorytmów wektoryzacja metodą Bag of words



Rys. 5.2: Poprawność algorytmów wektoryzacja metodą TFIDF

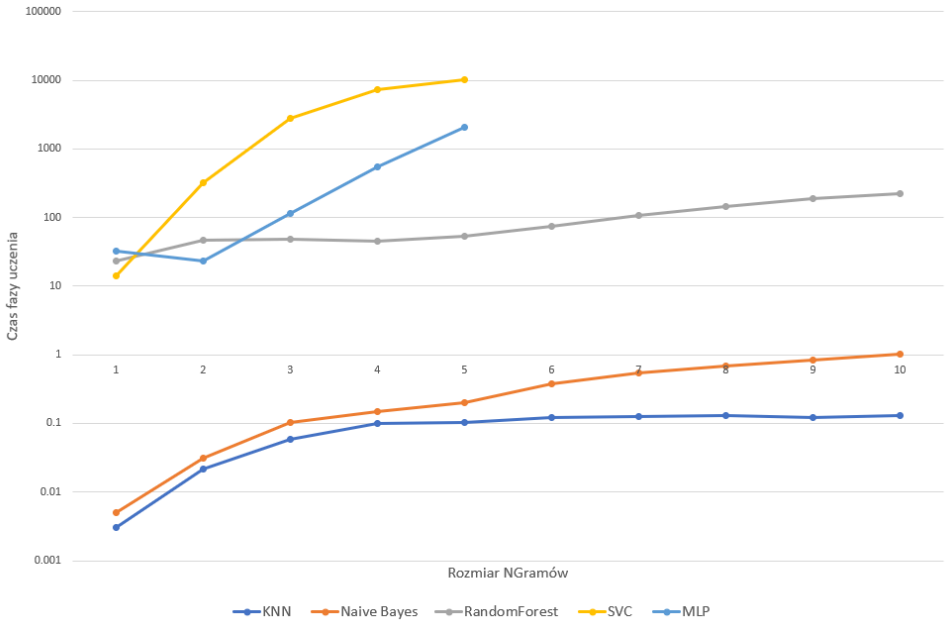
Pod względem poprawności algorytmy sprawdziły się następująco:

– *k*-NN - dla długości równej 1 algorytm osiąga wyniki większe od 70%, jednak wraz z podnoszeniem się jej zmniejsza się także poprawność. W przypadku obu metod wektoryzacji rozmiary 5 i 6 dają wyniki poniżej 50%, co oznacza, że są one gorsze od predykcji wykonanej na podstawie rzutu monetą. Jest to związane z overfittingiem będącym skutkiem

podnoszenia się ilości cech. Najlepszym wynikiem w przypadku k-NN jest 80.07% przy wykorzystaniu metody wektoryzacji Bag of words i rozmiarze równym 2, a najgorszym 46.60% przy wykorzystaniu metody Bag of words i rozmiaru 7,

- RF, SVC, MLP - algorytmy te zachowują się w bardzo podobny sposób. Zwiększanie długości ngramów do wartości 4 powoduje bardzo szybki wzrost efektywności do nawet 99.51%, natomiast zmiana długości ponad 4 prowadzi do powolnego spadku możliwości predykcyjnych, co może być związane podobnie jak w przypadku algorytmu k-NN z overfittingiem. Algorytmy te osiągają najlepsze wyniki pod względem efektywności ze wszystkich badanych,
- Naive Bayes - Algorytm Naive Bayes zwiększa swoją poprawność w taki sposób jak RF, SVC oraz MLP jednak robi to dużo wolniej i osiąga swoją szczytową poprawność dla długości równej 8 przy której jest ona równa 96.56%. Dalsze wydłużanie ngramów prowadzi do powolnego spadku możliwości predykcyjnych. Algorytm Naive Bayes osiąga najniższe wyniki w przypadku podziału na ngramy o długości równej 1, gdzie poprawnie klasyfikuje tylko 67.23% danych testowych.

- Czasy fazy uczenia



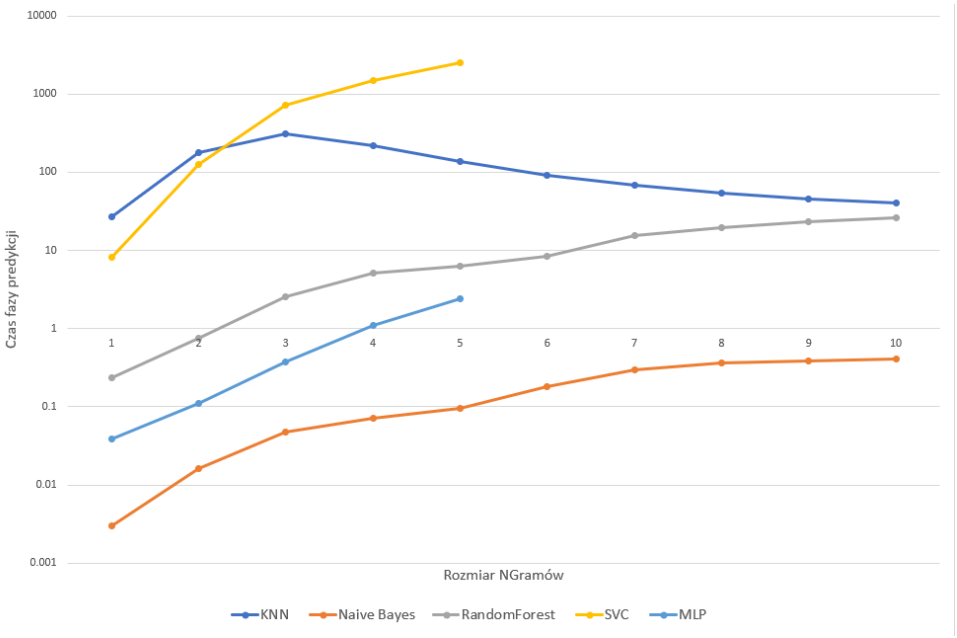
Rys. 5.3: Czasy uczenia algorytmów

Pod względem czasu fazy uczenia algorytmy sprawdziły się następująco:

- k-NN, Naive Bayes - Algorytmy k-NN oraz Naive Bayes osiągnęły najlepsze wyniki czasu trwania fazy uczenia, która podczas trwania całego badania nie była dłuższa niż jedna sekunda. Jest to spowodowane względnie prostymi operacjami wykonywanymi przez nie podczas tej fazy. Zadaniem algorytmu k-NN jest jedynie zapisanie wszystkich danych, natomiast Naive Bayes oblicza prawdopodobieństwa na podstawie prostych kalkulacji wykonywanych podczas jednej iteracji,
- RF - czas trwania fazy uczenia dla algorytmu RF ulega niewielkiemu wydłużaniu podczas zmiany rozmiaru ngramów, jednak podczas całego badania jest on przeciętny, przez co nie trwa zbyt długo, ale też nie jest szybki. Faza ta w tym przypadku polega na stworzeniu drzewa decyzyjnego jak najlepiej podejmującego decyzje. Czas ten nie ulega znacznemu wydłużeniu przy większej liczbie cech, ponieważ algorytm wybiera tylko najważniejsze z nich do stworzenia drzewa,

– SVC, MLP - w przypadku obu tych algorytmów można zauważyć znacznie zwiększanie się czasu uczenia wraz ze zwiększaniem długości ngramów. Doprowadziło to w ich przypadku do ograniczenia badań tylko do długości równej 5, ponieważ czas tej fazy wynosił niemal 3 godziny. Takie wyniki powodują, że nieprawdopodobne jest ich efektywne wykorzystanie w systemach rozpoznawania *fake newsów*.

- Czasy fazy predykcji



Rys. 5.4: Czasy predykcji algorytmów

Pod względem czasu fazy predykcji algorytmy sprawdziły się następująco:

– *k*-NN - predykcja tego algorytmu jest znacznie dłuższa niż faza uczenia i dla mniejszych długości ngramów trwa najdłużej. Początkowo zwiększanie długości powoduje wzrost czasu, jednak po osiągnięciu maksimum dla rozmiaru 3 następuje powolna stabilizacja w kierunku około 40 sekund,
– RF - czas trwania fazy predykcji zwiększa się powoli do wartości 26.49 sekund. Spowodowane jest to zwiększaniem się drzewa decyzyjnego który jest trawersowany w celu podjęcia decyzji. Czas ten podobnie jak w przypadku fazy uczenia jest przeciętny w porównaniu z innymi algorytmami,
– Naive Bayes - Jest to najszybszy algorytm zarówno pod względem fazy uczenia, jak i predykcji, której wykonanie opiera się na wyliczeniu odpowiednich prawdopodobieństw będących dla maszyny bardzo prostym zadaniem,
– SVC - czas trwania fazy predykcji jest bardzo podobny do czasu uczenia. Użycie SVC trwa zbyt długo by mógł on być zaimplementowany w jakichkolwiek systemach,
– MLP - w porównaniu z fazą uczenia predykcja następuje bardzo szybko, ponieważ opiera się tylko na aktywacji odpowiednich neuronów w stworzonym wcześniej modelu. Jest on na drugim miejscu pod względem prędkości wykonywania predykcji.

5.5. Wnioski z badań

Przeprowadzone badania pozwoliły na odnalezienie odpowiedzi na pytanie będące celem niniejszej pracy. Wykorzystanie algorytmów uczenia maszynowego w rozpoznawaniu fałszywych informacji skutkuje osiągnięciem bardzo optymistycznych wyników. Z badanych w algorytmów jedynie algorytm k-NN swoją efektywnością dla każdej długości ngramów nie pozwoliłby na pewne podjęcie decyzji, czy dany artykuł jest prawdziwy, czy też nie. Może to być związane z bardzo dużą liczbą cech w tekście, co bardzo szybko prowadzi do tak zwanego overfittingu.

Z powodu eksponencjalnego zwiększania się ilości cech wraz ze zwiększaniem długości ngramów, algorytmy takie jak SVC oraz MLP osiągają już dla długości równej 5 zbyt długie czasy trwania faz uczenia sięgające do 3 godzin. Czas taki nie pozwala na wykorzystanie ich w systemach detekcji *fake newsów*, ponieważ musiałyby one analizować dużą ilość artykułów, dla których bardzo duże znaczenie ma to jak szybko pojawią się na docelowej stronie internetowej bądź mediach społecznościowych.

Najlepszym z algorytmów okazał się RF, który dla ngramów o długości równej 4 osiągnął poprawność równą 99.82%. Pomimo iż poprawność ta została prawie osiągnięta przez algorytmy SVC oraz MLP to były one znacznie wolniejsze podczas fazy uczenia, a także podczas fazy predykcji w przypadku SVC. Szybszy od RF algorytm Naive Bayes również osiągnął zadowalające wyniki poprawności, jednak wysokie wartości odchylenia standardowego pozwalają stwierdzić, że wynik jest mocno zależny od danych, na których się on uczy w przeciwieństwie do algorytmu RF.

Zauważono także, że wybór metody wektoryzacji ma niewielki wpływ na to jak dobrze wykorzystany algorytm sprawdzi się w wykonywaniu swojego zadania. Jeżeli chodzi o czas trwania poszczególnych faz to różnica w czasie wynosiła zazwyczaj poniżej paru sekund w przypadku dłuższych algorytmów lub poniżej sekundy w przypadku krótszych. Pod względem efektywności można także zauważyć niewielkie różnice, które przeplatają się wraz ze zmianą długości, z tego powodu nie jest możliwe określenie, która metoda jest lepsza.

Wykorzystanie technologii machine learningu do zadania detekcji *fake newsów* może stanowić rozwiązanie problemów jakie niosą ze sobą fałszywe informacje. Ich automatyczna detekcja uchroniłaby miliony ludzi przed byciem oszukanym.

Rozdział 6

Podsumowanie

Celem pracy było sprawdzenie czy algorytmy uczenia maszynowego mogą zostać wykorzystane w detekcji tak zwanych *fake newsów* i jak dobrze wykonają to zadanie. Cel pracy został osiągnięty a otrzymane wyniki jednoznacznie wskazują, że tego typu algorytmy bez większych problemów dobrze radzą sobie z klasyfikacją nieprawdziwych informacji.

W baniach postanowiono sprawdzić metodę podziału tekstu na rożnej długości ngramy i zwrócić uwagę jak zmiana ich długości wpływa na poszczególne cechy algorytmów. Wyniki badań wykonanych na algorytmach: k-NN, RF, Naive Bayes, SVC oraz MLP pokazują, że algorytmy uczenia jak najbardziej mogą znaleźć swoje miejsce w systemach automatycznego rozpoznawania fałszywych informacji. Wiele z nich nie miało problemu z osiągnieciem efektywności powyżej 90%. Najlepszym z badanych algorytmów okazał się RF osiągając wynik 99.82%. Najgorzej z zadaniem poradziła sobie metoda k-NN, jej najwyższa poprawność wynosiła 78.13%.

Jak przypuszczano ogromne znaczenie dla efektywności miała zmiana długości ngramów, które nie mogą być ani za krótkie, ani za długie. Dla wszystkich badanych algorytmów poza k-NN ngramy o długości 1 dały najgorsze wyniki, a wartości pośrednie najlepsze.

Istotnym problemem z tego typu rozwiązaniami jest brak zbiorów treningowych posiadających artykuły o zróżnicowanej tematyce. Większość istniejących danych zawiera głównie artykuły związane z polityką, przez co z taką tematyką algorytmy poradziłyby sobie najlepiej. Mogłyby mieć one jednak problem z artykułami naukowymi, sportowymi itp.

Jednym ze sposobów na dalszy rozwój pracy mogłoby być opracowanie systemu rozpoznającego przerobione obrazy. System taki w kombinacji z wykrywaniem fałszywych tekstów zawartym w niniejszej pracy jeszcze bardziej zwiększyłby poprawność detekcji *fake newsów* i stanowiłby duży krok w wyeliminowaniu z internetu zagrożenia jakim są nieprawdziwe informacje.

Spis rysunków

Spis tabel

Literatura

[1] Fake news. it's complicated. https://firstdraftnews.org/latest/fake-news-complicated/. Dostęp dnia: 15-06-2020.

[2] Krótki przewodnik po fake newsach o koronawirusie. https://www.cyberdefence24.pl/krotki-przewodnik-po-aktualnych-fake-newsach-o-koronawirusie. Dostęp dnia: 15-06-2020.

[3] Obrona przestrzeni informacyjnej na przykładzie litwy, Łotwy i estonii. https://warsawinstitute.org/pl/obrona-przestrzeni-informacyjnej-na-przykladzie-litwy-lotwy-estonii/. Dostęp dnia: 15-06-2020.

[4] Scikit learn. https://scikit-learn.org/. Dostęp dnia: 15-06-2020.

[5] Słownik języka polskiego. https://sjp.pwn.pl/. Dostęp dnia: 15-06-2020.

[6] Wstrzymaniem oddechu nie sprawdzisz, czy masz koronawirusa. https://demagog.org.pl/analizy_i_raporty/wstrzymaniem-oddechu-nie-sprawdzisz-czy-masz-koronawirusa/. Dostęp dnia: 15-06-2020.

[7] Effectively pre-processing the text data part 1: Text cleaning. https://towardsdatascience.com, Dostęp dnia: 16-06-2020.

[8] Elon musk: 'mark my words — a.i. is far more dangerous than nukes'. https://www.cnbc.com/2018/03/13/elon-musk-at-sxsw-a-i-is-more-dangerous-than-nuclear-weapons.html, Dostęp dnia: 16-06-2020.

[9] Isot fake news dataset. https://www.uvic.ca/engineering/ece/isot/datasets/fake-news/index.php, Dostęp dnia: 16-06-2020.

[10] 10 examples of fake news from history. https://www.thesocialhistorian.com/fake-news/, Dostęp dnia: 23-08-2020.

[11] Measuring the reach of "fake news" and online disinformation in europe. https://www.press.is/static/files/frettamyndir/reuterfake.pdf, Dostęp dnia: 23-08-2020.

[12] Random forest – random forest. https://easyai.tech/en/ai-definition/random-forest/, Dostęp dnia: 23-08-2020.

[13] C. Albon. *Machine Learning with Python Cookbook*. O'Reilly Media, 2018.

[14] M. Dice. *The True Story of Fake News: How Mainstream Media Manipulates Millions*. The Resistance Manifesto, 2017.

[15] A. Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2017.

[16] G. Hackeling. *Mastering Machine Learning with scikit-learn - Second Edition: Apply effective learning algorithms to real-world problems using scikit-learn*. Packt Publishing, 2017.

[17] F. L. Mott. *American Journalism*. Macmillan, 1941.

[18] E. L. Steven Bird, Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media, 2009.

Dodatek A

Opis załączonej płyty CD/DVD

Na załączonej płycie znajduje się niniejsza praca w formacie PDF oraz pliki z kodem źródłowym aplikacji wykorzystanej do wykonania badań.