**Porównanie metod wstępnego przetwarzania i klasyfikacji danych biomedycznych**

Piotr Tąkiel

Instytut Informatyki, Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska

Ul. Nowowiejska 15/19, 00-665 Warszawa, Polska

p.takiel@stud.elka.pw.edu.pl

**Streszczenie.** *Niniejszy artykuł składa się z dwóch części. W pierwszej z nich przedstawię cel mojej pracy oraz przypomnę czym są przetwarzanie wstępne danych, klasyfikacja danych oraz testy statystyczne. W części drugiej opiszę projekt mojej pracy, przed-stawię wstępne wyniki badań oraz, płynące z nich, wnioski.*

**1. Cel pracy magisterskiej.** Końcowym celem mojej pracy dyplomowej jest porów-nanie ocen grup metod wstępnego prze-twarzania i klasyfikacji danych biome-dycznych. W przypadku mojej pracy, dane te dotyczyć będą zapadalności na raka piersi u kobiet.

**1.1. Motywacja.** Podstawowym powodem, dla którego postanowiłem zająć się tym tematem, jest aktualność problemu raka piersi u kobiet. Zapadalność na raka piersi wzrasta z roku na rok w krajach wysoko rozwiniętych. W krajach tych, pomimo postępów w diagnozowaniu i leczeniu, obserwuje się wzrost corocznej liczby zgonów wywołanej tą chorobą. W Polsce kobiety chorują na raka piersi nieco rzadziej niż w reszcie świata. U co dwunastej kobiety na świecie diagnozuje się raka piersi, podczas gdy w Polsce jest to co czternasta kobieta. Wyleczalność wygląda w naszym kraju znacznie gorzej. Co druga Polka chora na raka piersi umiera z powodu tej choroby, podczas gdy np. w Holandii jedynie trzy na dziesięć kobiet.

Drugiem powodem, dla którego postano-wiłem zająć się tą pracą, jest popularność metod uczenia maszynowego. Od końca 2011 roku można zaobserwować rosnące zainteresowanie tym tematem w Internecie. Uczenie maszynowe znajduje bardzo wiele praktycznych zastosowań, w skład których wchodzą: rozpoznawanie obrazów, wyszu-kiwanie, detekcja spamu, wykrywanie fał-szerstw finansowych oraz rozpoznawanie chorób na podstawie symptomów. To ostatnie zastosowanie sprawia, że to właśnie uczenie maszynowe zostanie użyte w mojej pracy dyplomowej.

W celu zrozumienia celu pracy, w nas-tępnych sekcjach przybliżę czym są prze-twarzanie wstępne oraz klasyfikacja danych.

**2. Klasyfikacja.** Klasyfikacja statystyczna to rodzaj algorytmu statystycznego, który przydziela obserwacje statystyczne do klas, używając atrybutów tych obserwacji. Algorytm klasyfikacji tworzy model kla-syfikacji na podstawie danych trenujących zawierających znane klasy (kategorie). Gotowy model można następnie zastosować do predykcji kategorii nowych, nieznanych wcześniej, danych. Klasyfikacja jest zatem przykładem uczenia się z nadzorem. Nadzór polega w tym przypadku na obecności informacji o kategoriach danych, które mamy. Algorytm tworzenia modelu stara się nauczyć model docelowych kategorii, jednocześnie dbając o to, aby wiedza na temat problemu była jak najbardziej ogólna i niezależna od próby, którą dysponujemy.

**2.1. Ocena klasyfikacji.** W celu oceny gotowego modelu klasyfikacji, możemy po-służyć się miarami zdefiniowanymi na podstawie zawartości tzw. *macierzy pomyłek*. Dla klasyfikacji dwuwartościowej macierz pomyłek składa się z dwóch wierszy i dwóch kolumn. Kolumny odpowiadają faktycznym kategoriom danych, wiersze zaś predykcjom klasyfikatora. Na przecięciu kolumn i wierszy znajdują się cztery wartości liczbowe charakteryzujące po-prawność wyników modelu. Wartości *True Positive* i *True Negative* określają ile poprawnych wyników pozytywnych i negatywnych zostało zwróconych przez klasyfikator. Z kolei wartości *False Positive* i *False Negative* pokazują ile pozytywnych i negatywnych przypadków model zaklasy-fikował niewłaściwie.

Ocenę jakości modelu klasyfikacji łatwo wyrazić wartościami liczbowymi policzo-nymi przy użyciu poszczególnych komórek macierzy pomyłek. *Celnością*  nazywamy stosunek liczby poprawnych wyników klasyfikacji do wszystkich rozpat-rywanych przypadków:

Miara ta pokazuje, jak często klasyfikator daje poprawny wynik. W medycynie częściej jednak stosuje się inne miary oceny, w szczególności *precyzję* i *wrażliwość*.

Precyzja odpowiada na pytanie, jaka część zdiagnozowanych przypadków była praw-dziwa. Precyzję wyrażamy jako stosunek liczby poprawnie zdiagnozowanych przy-padków do liczby wszystkich pozytywnych wyników predykcji modelu:

Druga miara - wrażliwość - pokazuje nam, jaka część chorych została wykryta przez klasyfikator. Wrażliwość wyraża się jako iloraz liczby słusznie zdiagnozowanych przypadków oraz liczby wszystkich zachorowań:

W praktyce trudo o model, który ma jednocześnie wysokie wartości precyzji i wrażliwości. Z tego powodu często stosuje się miarę *F1*, będącą średnią harmoniczną dwóch powyższych miar:

Oceniając jakość klasyfikacji nie należy zapominać o problemie *nadmiernego do-pasowania modelu*. Nadmierne dopaso-wanie to zjawisko zachodzące, gdy model klasyfikacji daje gorsze wyniki dla nowych danych niż dla znanych danych trenujących. Aby ocena modelu nie była obarczona błędem związanym z tym zjawiskiem, do ewaluacji klasyfikatora należy używać osobnego zbioru testowego rozłącznego ze zbiorem trenującym.

Dzieląc dostępne dane na zbiór trenujący i testowy należy pamiętać o konsekwencjach tego podziału. Im mniejszy jest zbiór uczący (a testowy większy) tym mniej informacji stosujemy do nauki modelu i końcowa ocena może być zaniżona. Z drugej strony, im mniejszy jest zbiór testowy, tym mniej jest reprezentatywny dla całej populacji i tym większa będzie wariancja oceny na tym zbiorze.

Z powyższym problemem można radzić sobie wykonując wiele testów. W literaturze wyróżnia się zwykle dwie metody z tym związane: *sprawdzian krzyżowy* (inaczej: walidacja krzyżowa) oraz *metoda bootstrap*.

*Sprawdzian krzyżowy* polega na podziale danych na równolicznych części, utwo-rzeniu danych uczących z z nich i przeprowadzniu testów na pozostałej częsci. Uczenie i ocenę modelu powtarza się *n*-krotnie, za każdym razem dla innych części. Otrzymane ocen można użyć np. do wyliczenia oceny końcowej, będącej średnią.

*Metoda bootstrap* tworzy zbiór uczący poprzez losowanie ze zwracaniem ze zbioru danych wejściowych. Przykłady niewy-losowane stają się zbiorem testowym.

**2.2. Algorytmy klasyfikacji.** Do najpopularniejszych klasyfikatorów stoso-wanych obecnie należą naiwny klasyfikator bayesowski, drzewo decyzyjne, losowy las oraz maszyna wektorów ciągłych. Dalsza część rozdziału skupi się na omówieniu tych algorytmów.

Naiwny klasyfikator bayesowski jest klasyfikatorem probabilistycznym. Oznacza to, że dla danego przykładu wynikiem zwracanym przez model są prawdo-podobieństwa poszczególnych kategorii. Klasyfikator ten znany jest od ponad 60 lat. Zasada działania tego klasyfikatora opiera się na twierdzeniu Bayesa, znanym z probabilistyki:

„Naiwność” tej metody klasyfikacji polega na założeniu, że wewnątrz kategorii, atrybuty są względem siebie niezależne. W wielu przypadkach założenie to nie jest spełnione, co odbija się na jakości oceny klasyfikatora. Mimo tej oczywistej wady, naiwny klasyfikator bayesowski znajduje praktyczne zastosowanie, np. przy diagno-zowaniu niektórych chorób.

Kolejny popularnym klasyfikatorem jest drzewo decyzyjne. W trakcie uczenia modelu budowane jest drzewo, w którego węzłach znajdują się testy na wartości atrybutów, a w liściach kategorie lub prawdopodobieństwa kategorii. W trakcie predykcji przykład wędruje od korzenia drzewa do liścia, przechodząc konkretnymi gałęziami w zależności od wyników testów w węzłach. Popularność drzewa decyzyjnego bierze się m.in. z tego, że bardzo często daje ono zadowalające wyniki klasyfikacji. Często obserwowanym problemem jest ich nad-mierny rozrost wgłąb i, towarzyszące temu, nadmierne dopasowanie do danych uczą-cych. Z problemem tym można walczyć, przycinając drzewo, tj. zastępując pod-drzewa liśćmi. Postępowanie takie wynika z przekonania, że modele mniej skom-plikowane lepiej generalizują wiedzę. Wśród algorytmów przycinania wyróżnia się m.in.: *Reduced Error Pruning*, *Minimum Error Pruning* lub *Cost Complexity Pruning*.

Rozwinięciem pomysłu związanego z drzewami decyzyjnymi jest klasyfikator znany jako *losowy las*. Klasyfikator ten to zbiór drzew decyzyjnych, z których każde nauczone jest na oddzielnym zbiorze trenującym uzyskanym metodą *boostrap*. Podczas predykcji przykład przechodzi przez każde z drzew, generując przy tym serię wyników. Rezultaty tych pomniejszych klasyfikacji grupowane są w zbiorowy ranking, na podstawie którego klasyfikator losowy las podejmuje ostateczną decyzję co do kategorii przykładu.

Ostatnim prezentowanym przeze mnie klasyfikatorem jest *Maszyna Wektorów Nośnych*. Jest to klasyfikator binarny. Nauka modelu klasyfikatora ma na celu wyzna-czenie hiperpłaszczyzny rozdzielającej z maksymalnym marginesem przykłady nale-żące do dwóch klas. W odróżnieniu od naiw-nego klasyfikatora Bayesa, metoda maszyny wektorów nośnych jest względnie nowa – dokument opisujący tę metodę został opublikowany w latach 90 XX w.

**3. Przetwarzanie wstępne.** Celem przetwa-rzania wstępnego, w kontekście uczenia maszynowego i klasyfikacji, jest zamiana oryginalnej postaci danych wejściowych na taką, która umożliwia uzyskiwanie lepszych wyników późniejszej klasyfikacji. W mojej pracy dyplomowej korzystam z pewnych metod wstępnego przetwarzania danych i badam ich wpływ na końcową ocenę klasyfikacji. W rozdziale tym omawiam użyte przeze mnie metody, jak również kilka innych.

**3.1. Skalowanie.** Wiele algorytmów klasyfikacji oczekuje, aby wartości atrybutów należały do ustalonego przedziału liczb, np. [0,1]. Aby zapewnić dobre wyniki klasyfikacji, często stosuje się *skalowanie*. Metoda ta polega na pro-porcjonalnym przeskalowaniu wartości atrybutu tak, aby znalazły się w ustalonym przedziale. Częstymi docelowymi prze-działami są [0, 1] i [-1, 1].

**3.2. Brakujące wartości atrybutów.** Wszystkie cztery metody zakładają, że wśród danych nie ma brakujących wartości. W realnych zastosowaniach często niestety jest inaczej – dane, którymi dysponujemy, mają pewien odsetek braków. Przykładowo, dane medyczne użyte w mojej pracy mają 8% brakujących wartości. Nawet, jeżeli przyj-miemy, że wyżej wymienionych metod nie będzie się stosować, to wciąż niektóre algorytmy klasyfikacji wymagają komplet-nych danych. Wyróżniamy zwykle dwie metody rozwiązywania tego problemu.

Pierwszą z metod jest usuwanie przy-kładów, w których występują braki, i dalsza praca na pełnych danych. Jest to najprostsza metoda, ale również posiadająca pewne nieporządane konsekwencje. Po pierwsze, eliminując przykłady, pozbywamy się części dostępnej informacji. W rezultacie spodzie-wamy się, że wytworzony model klasyfikacji będzie działał gorzej. Po drugie, metoda ta sprawia, że późniejsza predykcja przy-kładów z brakami jest niemożliwa.

Drugą metodą radzenia sobie z bra-kującymi wartościami jest *imputacja*. Polega ona na zastępowaniu braków wartościami, stosując do tego pozostałe, dostępne dane. W przypadku atrybutów ciągłych braki można zastępować średnią lub medianą atrybutu. Braki w atrybutach nominalnych możemy „łatać” najczęstszą wartością. Podejściem nieco bardziej zaawansowanym jest użycie klasyfikatora lub algorytmu regresji w celu przewidywania wartości w pustych miej-scach.

**3.3. Redukcja wymiarów.** Ostatnim z rodzajów metod przetwarzania wstępnego jest redukcja wymiarów, czyli zmniejszenie liczby atrybutów. Często bywa tak, że kategoria przykładu jest uzależniona od podzbioru jego atrybytów. Pozostałe atrybuty, nie dość, że nie powinny wnosić nic do procesu klasyfikacji, to jeszcze swoją obecnością wydłużają czas obliczeń i mogą negatywnie wpływać na proces uczenia. Usuwając te niepotrzebne atrybuty możemy zmniejszyć czas obliczeń oraz wpłynąć na ocenę klasyfikacji.

W celu wybrania właściwych atrybutów do usunięcia możemy usuwać kolejne kombinacje atrybutów, a następnie porów-nywać oceny klasyfikacji dokonanej na reszcie danych. Metoda ta ma oczywistą wadę – jest bardzo czasochłonna. Inne podejście prezentuje metoda *PCA* – poszu-kuje ona takich kombinacji atrybutów, które wiernie oddają wariancję całego zestawu danych.

Ostatnim podejściem do redukcji wymia-rów jest grupowanie atrybutów. Metoda ta traktuje wartości *n*-elementowego pod-zbioru atrybutów jako punkty w przestrzeni *n*-wymiarowej. Punkty te są grupowane za pomocą odpowiednich algorytmów uczenia maszynowego (np. K-średnich, DBSCAN) a następnie zamieniane na etykiety tych grup. W ten sposób z atrbutów powstaje jeden atrybut etykiet, a zostaje usuniętych.

**4. Testy statystyczne.** Celem mojej pracy dyplomowej jest porównanie jakości klasy-fikacji grup pewnych algorytmów. Aby porównanie to było wiarygodne i, aby mogło posłużyć dalszym badaniom, powinno ono używać ocen mierzonych na całej populacji, a nie na losowej próbie. Oczywistym jest, że nie dysponujemy informacjami na temat wszystkich pacjentek i możemy użyć jedynie niewielkiego zbioru danych (ok. 1000 przykładów w próbie). W związku z po-wyższym, aby porównanie było możliwie dokładne, wykonane zostanie w oparciu na teście statystycznym.

Testem statystycznym nazywamy formu-łę matematyczną pozwalającą oszacować prawdopodobieństwo spełnienia pewnej hipotezy statystycznej w populacji na pod-stawie próby losowej z tej populacji. W przypadku mojej pracy, hipotezą zerową, którą pragniemy obalić, jest jednakowy rozkład ocen klasyfikacji dla wybranych dwóch grup metod.

Testy statystyczne dzielimy na dwa ro-dzaje: parametryczne i nieparametryczne. Testy pierwszego rodzaju zakładają pewien ustalony rozkład danych oraz parametry tego rozkładu. Przykładami testów statys-tycznych są testy *t-Studenta* oraz korelacja Pearsona.

Testy nieparametryczne nie wymagają za-łożeń odnośnie populacji, z której losowana jest próba. Dzięki mniejszej liczbie założeń do spełnienia mogą być szerzej stosowane niż testy parametryczne. Wadą testów nieparametrycznych jest mniej dokładny pomiar oraz mniejsza moc. Mocą testu nazywamy prawdopodobieństwo odrzu-cenia fałszywej hipotezy zerowej. W przy-padku mojej pracy, testy nieparametryczne rzadziej będą wskazywać rzeczywiste różnice w rozkładach ocen pomiędzy grupami. Przykładami testów niepara-metrycznych są test Manna-Whitneya-Wilcoxona oraz test *chi kwadrat*.

Biorąc pod uwagę brak wiedzy a priori na temat rozkładu ocen w grupach metod klasyfikacji, do porównywania użyty został test nieparametryczny Manna-Whitneya-Wilcoxona. Test ten polega na policzeniu statystyki *U*, związanej z liczbą przypadków, gdzie elementy jednej z grup mają wartości wyższe niż elementy z drugiej grupy. Następnie wartość *U* porównywana jest z wartością graniczną dla ustalonych liczności grup oraz poziomu istotności testu. W przypadku bardzo licznych grup, statystyka *U* jest zmienną losową o rozkładzie nor-malnym. Pozwala to w prosty sposób obliczyć *p-wartość* dla danej obserwacji, a następnie porównać ją z ustalonym po-ziomem istotności testu.

**5. Zarys projektowy.** Przeprowadzone przeze mnie badania polegały na określeniu grup sparametryzowanych algorytmów przetwarzania wstępnego i klasyfikacji oraz na późniejszym porównaniu wyników kla-syfikacji dla tych grup. Każdy algorytm skła-dał się ze ściśle ustalonych etapów prze-twarzania i klasyfikacji.

**5.1. Schemat badań.** W pierwszej kolejności ustalone zostały grupy algorytmów *Ai ∈ A*. Następnie dla każdego algorytmu *aj ∈ Ai* obliczane były miary oceny klasyfikacji *Maj*. (np. precyzja algorytmu *ax* testowana była 10-krotnym sprawdzianem krzyżowym, dając w rezultacie zbiór *Max* zawierający 10 miar precyzji). W kolejnym kroku wszystkie miary z grupy łączone były we wspólny zbiór *Yi = ⋃Maj*. Po wyznaczeniu wszystkich zbiorów ocen, zostały one porównane ze sobą testem Manna-Whitneya-Wilcoxona. Jeżeli grupy wyników wy-kazywały istotną różnicę w wartościach, o wyższości jednej grupy nad drugą decy-dowała średnia. Po zakończeniu porównania, grupy zostały uszeregowane zgodnie z liczbą „wygranych” porównań z innymi grupami.

**5.2. Parametryzacja grup algorytmów.** Każdy algorytm składał się czterech etapów wstępnego przetwarzania oraz z poje-dynczego etapu końcowej klasyfikacji. W skład przetwarzania wchodziło: usuwanie braków, usuwanie kombinacji atrybutów, skalowanie kombinacji atrybutów oraz gru-powanie atrybutów. Każdy etap posiadał pewne konkretne parametry, np. zbiór atry-butów do przeskalowania lub parametry klasyfikacji.

Projektując metodę opisu grup algo-rytmów należało zachować równowagę pomiędzy skrótowością a ekspresywnością. Krótki, nieprecyzyjny opis nie pozwalałby na wyrażanie konkretnych rodzajów algo-rytmów w grupie. Opis rozwlekły byłby kło-potliwy w użyciu, gdyż wymagałby dużego nakładu pracy, aby opisać grupę metod.

Zastosowałem następujące rozwiąznie opisujące każdą z grup metod. Etap eli-minowania braków określony był zbiorem metod (np. imputacja średnią, imputacja medianą), których algorytmy w grupie powinny użyć. Usuwanie oraz skalowanie atrybutów sparametryzowane zostało zbio-rem atrybutów *X* oraz zbiorem liczności elemntów *Nx*. Algorytmy należące do grupy wykonywały właściwe operacje na każdych *ni* atrybutach z *X*, gdzie *ni ∈ Nx*. Etap grupowania został opisany zbiorem algo-rytmów grupowania *A*, zbiorem atrybutów *X*, zbiorem liczności *NA*, liczbą *nmax*oraz granularnością *g*. Algorytmy grupowania *A* uruchamiane były *ni*-krotnie (*ni* *∈* *NA*) we wszystkich możliwych kombinacjach na atrybutach ze zbioru *X*, z czego każdy algorytm na maksymalnie *nmax* atrybutach. Ostatni etap, klasyfikacja, został sparamet-ryzowany zbiorem algorytmów klasyfikacji *A* oraz granularnością *g*.

*Granularnością* nazywam liczbę, która określa, ile równo oddalonych od siebie wartości należy wybrać z danego przedziału. Parametr ten pozwala próbkować dany przedział liczb z określoną częstością. Umożliwia to mniej lub bardziej dokładne przeglądanie parametrów liczbowych dla algorytmów grupowania i klasyfikacji. W zależności od zastosowania, różnica po-między kolejnymi wartościami z przedziału może być równa na skali liniowej lub logarytmicznej.

**5.3. Skrócenie czasu obliczeń.** Jak łatwo zauważyć, możliwe jest zdefiniowanie grup algorytmów, których liczność jest bardzo duża (zależna kombinatorycznie od nie-których parametrów). Oznacza to, że badania przeprowadzanie na takich grupach będą bardzo czasochłonne. W celu skrócenia cza-su, zastosowałem pewną istotną opty-malizację oraz zrównolegliłem obliczenia na wielu maszynach.

Analizując złożoność czasową każdego etapu algorytmu, można zauważyć, że najwięcej czasu poświęca się na obliczenia związane z grupowaniem oraz klasyfikowaniem danych. O ile klasyfikacja musi być przeprowadzana za każdym razem, o tyle grupowanie niekoniecznie. Dane po grupowaniu (czyli po przetworzeniu wstę-pnym) można zapamiętać oraz natych-miastowo odtworzyć w następnej iteracji przebiegu badań.

Rozpraszając obliczenia na wiele maszyn, należy mieć na uwadze wpływ opóźnień w sieci na wydajność przebiegu badań. Komunikacja sieciowa jest wolniejszą meto-dą wymiany informacji niż np. komunikacja międzywątkowa lub międzyprocesowa. Na-leży również zdawać sobie sprawę z prob-lemu związanego z zawodnością zdalnych komputerów. Z losowych przyczyn mogą się one zawiesić, wyłączyć, wprowadzając za-kłócenia do całego procesu obliczeń. Do-datkowo, dobrze byłoby tak zaprojektować rozwiązanie rozproszone, aby można było zastosować w nim optymalizację opisaną w poprzednim akapicie.

Zrealizowany przeze mnie system do pewnego stopnia spełnia powyższe wymogi. System oparty jest na architekturze klient-serwer korzystającej z systemu kolejkowego. Każdy algorytm w danej grupie opisywany jest obiektem przechowującym parametry algorytmu. Proces koordynujący obliczenia wysyła bloki obiektów do maszyn, które zajmują się oceną każdego odebranego algorytmu. Przesyłanie metod w dużych grupach ma dwie duże zalety: minimalizuje narzut związany z komunikacją sieciową oraz pozwala na użycie, przedstawionej przeze mnie, optymalizacji. Jeżeli wyniki obliczeń nie wrócą do koordynatora w ustalonym czasie, blok uznaje się za stracony i wysyła ponownie. Rozwiązanie to pozwala uniknąć sytuacji, w której awaria na dowolnej maszynie mogłaby doprowadzić do niekompletnych obliczeń.

**6. Wstępne wyniki badań oraz wnioski**. Przeprowadzone przeze mnie badania objęły porównanie następujących grup metod. Dla każdego algorytmu klasyfikacji oraz każdej metody usuwania braków, badałem grupy bez przetwarzania wstępnego, z usuwaniem wszystkich kombinacji atrybutów, ze ska-lowaniem tychże atrybutów oraz z usu-waniem i ze skalowaniem.

Pod kątem miary F1, najlepsza okazała się grupa imputująca wartością średnią, uży-wająca klasyfikatora losowy las oraz ska-lująca wartości atrybutów. Ta sama grupa zajęła również pierwsze miejsce w rankingu precyzji. W rankingu wrażliwości zajęła jednak dopiero 10 miejsce.

Analizując otrzymany ranking stwier­dziłem, że do klasyfikacji danych najlepiej nadaje się klasyfikator losowy las – grupy używające go zajęły pierwsze 12 miejsc rankingu F1.

Następnie, przyglądając się rozkładowi ocen w najlepszych grupach, stwierdziłem, że możliwe jest zapewnienie jednoczesnej średniej precyzji na poziomie powyżej 84% oraz średniej wrażliwości powyżej 80%.

Ostatni wniosek dotyczy zastosowania klasyfikatora o nazwie maszyna wektorów nośnych. Pomimo bardzo wysokiej wra-żliwości (średnio ok. 93%) grupy używające go posiadały niewielką precyzję (średnio ok. 55%), co czyni go w badanym problemie bezużytecznym.

**6. Podsumowanie.** W niniejszym artykule przedstawiłem cel mojej pracy dyplomowej. Następnie krótko przypomniałem na czym polega klasyfikacja danych oraz ich przetwarzanie wstępne, pokazałem również dlaczego użyłem w mojej pracy testu statystycznego. Następnie opisałem plan badań oraz szkic architektury rozwiązania, a na samym końcu zaprezentowałem wstępne wyniki oraz wnioski, które wyciągnąłem na ich podstawie.

**7. Odnośniki**

1. Cichosz P.: Systemy Uczące Się. Wydawnictwa Naukowo-Techniczne, War-szawa, 2000
2. Internetowy podręcznik użytkownika biblioteki *scikit-learn* języka Python
3. Portal abczdrowie.pl, 20.04.2015
4. Wikipedia