

POLITECHNIKA BIAŁOSTOCKA

WYDZIAŁ INFORMATYKI

PRACA DYPLOMOWA INŻYNIERSKA

TEMAT: APLIKACJA INTERNETOWA DO OBSŁUGI  
SYSTEMU GDT W TECHNOLOGII  
PYTHON/DJANGO.

WYKONAWCA: MATEUSZ PERNAL

.....  
podpis

PROMOTOR: DR INŻ. KRZYSZTOF JURCZUK

.....  
podpis

BIAŁYSTOK 2021 r.

## Karta dyplomowa

Politechnika Białostocka  Wydział Informatyki   Katedra Oprogramowania	Studia stacjonarne  studia I stopnia	Numer albumu studenta: 101420
		Rok akademicki 2019/2020
		Kierunek studiów: informatyka Specjalność: Brak
<b>Mateusz Pernal</b>  <b>TEMAT PRACY DYPLOMOWEJ: Aplikacja internetowa do obsługi systemu GDT w technologii Python/Django.</b>  Zakres pracy: <ol style="list-style-type: none"> <li>1. Zapoznanie z systemem GDT</li> <li>2. Analiza wymagań aplikacji</li> <li>3. Projekt i implementacja aplikacji</li> <li>4. Testy oraz wdrożenie aplikacji.</li> </ol>		
<div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="width: 45%;">                 .....                  Imię i nazwisko promotora - podpis             </div> <div style="width: 45%;">                 .....                  Imię i nazwisko kierownika katedry -                  podpis             </div> </div>		
<div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="width: 30%;">                 .....                  Data wydania tematu pracy dyplomowej                  - podpis promotora             </div> <div style="width: 30%;">                 .....                  Regulaminowy termin złożenia                  pracy dyplomowej             </div> <div style="width: 30%;">                 .....                  Data złożenia pracy dyplomowej                  - potwierdzenie dziekanatu             </div> </div>		
<div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="width: 45%;">                 .....                  Ocena promotora             </div> <div style="width: 45%;">                 .....                  Podpis promotora             </div> </div>		
<div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="width: 30%;">                 .....                  Imię i nazwisko recenzenta             </div> <div style="width: 30%;">                 .....                  Ocena recenzenta             </div> <div style="width: 30%;">                 .....                  Podpis recenzenta             </div> </div>		

Subject of diploma thesis: Internet application to support the GDT system in Python/Django technology.

## Summary

The aim of this thesis was to design, implement and introduce a web application to support GDT (*Global Decision Trees*) system using Python and Django technologies. In the basic version, GDT is a console program for creating decision trees. The requirement of the project was to create a web application allowing to create, delegate and manage the tasks launched by using the GDT system. The developed tool will also provide graphical representation of the obtained results.

The thesis consists of four chapters. First chapter describes the problem and presents the decision trees. It also shows the most popular existing solutions. Second chapter contains an analysis of project requirements and a overview of the technologies that have been used. Third chapter is intended to present the system architecture. It illustrates most important mechanisms and solutions created for the application. It also shows the database schema with a short description of tables. Fourth chapter contains a overview of particular views of the application, but also an example how to use them. In addition, it presents the results of load and manual tests.

# Spis treści

## Streszczenie

<b>Wprowadzenie</b>	<b>6</b>
<b>1 Przedstawienie problemu</b>	<b>9</b>
1.1 Drzewa decyzyjne . . . . .	9
1.2 Uczenie maszynowe . . . . .	9
1.3 Drzewa decyzyjne w technikach uczenia maszynowego . . . . .	10
1.4 Istniejące rozwiązania . . . . .	11
<b>Podsumowanie</b>	<b>16</b>
<b>Bibliografia</b>	<b>18</b>
<b>Spis tabel</b>	<b>19</b>
<b>Spis rysunków</b>	<b>20</b>
<b>Spis listingów</b>	<b>21</b>

# Wprowadzenie

Proces myślowy człowieka jest w dużej mierze oparty o pewien schemat podejmowania decyzji. Podejmowane decyzje mają kluczowy wpływ na jego aktualne życie i przyszłość. Wybór najbardziej optymalnego rozwiązania danego problemu wymaga dokładnej analizy dostępnych informacji. Posiadając wystarczającą ilość danych możemy wykorzystać różne algorytmy, które mogą pomóc podjąć właściwy wybór. Mechanizm podejmowania decyzji bezpośrednio dotyczy nie tylko człowieka, a wszystkiego co znajduje się w jego otoczeniu.

Szybki rozwój technologii w XX i XXI wieku prowadzi do produkowania i gromadzenia coraz większej ilości informacji. Firmy starają się wyciągnąć ze zgromadzonych danych możliwe jak najlepsze wnioski. Poddając analizie tak duże zbiory informacji wymagane jest zastosowanie narzędzi upraszczających i przyspieszających uzyskanie wyników. Prowadzi to do tworzenia algorytmów oraz mechanizmów zarówno obróbki danych, jak i ich analizy w celu osiągnięcia zadowalających rezultatów. W przeciągu ostatnich kilkunastu lat entuzjazm związany z technikami komputerowymi wzrósł gwałtownie i zdominował przemysł. Uczenie maszynowe wraz z analizą danych stanowi bardzo ważny element rozwiązań produkowanych przez firmy. Wspomaga takie technologie, jak rozpoznawanie mowy, pisma czy też autonomiczne samochody i roboty sprzątające. Wszystkie te rozwiązania wymagają przetwarzania ogromnych ilości informacji, w jak najkrótszym czasie oraz podjęcie wystarczająco dobrej decyzji.

W pracy tej rozwijany będzie system do uczenia maszynowego GDT (*Global Decision trees*)[1] tworzony przez pracowników Politechniki Białostockiej. System ten służy do generowania drzew decyzyjnych na podstawie zbioru uczącego. Drzewa generowane są z wykorzystaniem algorytmów ewolucyjnych (metoda alternatywna do algorytmów zachłanych typu *top-down*). Aplikacja GDT jest programem konsolowym. W celu ułatwienia dostępu do platformy GDT większemu gronu użytkowników w niniejszej pracy zostanie zaprojektowana, zaimplementowana oraz wdrożona aplikacja do obsługi systemu GDT z poziomu przeglądarki internetowej.

## **Cel pracy**

Celem pracy jest stworzenie aplikacji webowej umożliwiającej obsługę systemu GDT. Aplikacja ta będzie umożliwiać tworzenie, zlecanie oraz zarządzanie zadaniami uruchamianymi przy pomocy systemu. Podczas tworzenia zadań użytkownik powinien móc ustawić opcje dotyczące, np. wybranego algorytmu oraz jego parametrów. Aplikacja powinna także udostępniać opcje związane z wyświetleniem drzewa wynikowego w postaci graficznej, jego eksport do pliku oraz wgląd do pozostałych wyników uruchomianego algorytmu.

## **Zakres pracy**

Zakres pracy obejmuje:

- Zapoznanie z systemem GDT,
- Analiza wymagań aplikacji,
- Projekt i implementacja aplikacji,
- Testy oraz wdrożenie aplikacji.

## **Organizacja pracy**

Praca została podzielona na cztery główne części. Pierwsze dwa rozdziały zawierają przedstawienie problemu, analizę wymagań i wykorzystane technologie. W dalszej części pracy została omówiona architektura aplikacji wraz z zastosowanymi rozwiązaniami. Natomiast prezentacja stworzonej aplikacji oraz opis testów został przedstawiony w rozdziale 4.

Rozdział 1 zawiera opis mechanizmu tworzenia drzew decyzyjnych. Przedstawia również zagadnienia związane z systemem GDT. Porusza też temat podobnych aplikacji dostępnych w internecie.

Rozdział 2 przedstawia analizę wymagań funkcjonalnych i нефункциональных tworzonej aplikacji. Został w nim zamieszczony diagram przypadków użycia wraz z ich opisem oraz diagram czynności. Następnie zaprezentowany został schemat rozwiązania. Rozdział kończy przedstawienie użytych technologii podczas tworzenia aplikacji.

Rozdział 3 przedstawia architekturę tworzonej aplikacji. Prezentuje najważniejsze mechanizmy oraz rozwiązania stworzone na potrzeby aplikacji. Przedstawia także schemat bazy danych wraz z krótkim opisem najważniejszych tabel.

Rozdział 4 przedstawia stworzoną aplikację. Zawiera on opis poszczególnych widoków aplikacji, ale także przykładowy sposób ich użycia. Ponadto przedstawia wyniki testów obciążeniowych i manualnych przeprowadzonych przez studentów Wydziału Informatyki Politechniki Białostockiej.

# 1. Przedstawienie problemu

## 1.1 Drzewa decyzyjne

Podjęcie decyzji jest procesem nietrywialnym. Już od początku istnienia ludzkości dobrze podjęte decyzje pozwalały przeżyć wybranym grupom ludzi czy też zwierząt. Wpływ na optymalną decyzję mają informacje, które zostaną poddane analizie, ale także sama metoda analizy. Racjonalny wybór może być wspomagany różnymi algorytmami, czy też wizualną reprezentacją możliwych decyzji. Jedną z form graficznych jest drzewo decyzyjne.

Podstawowymi elementami drzewa są węzły oraz gałęzie. Korzeń drzewa to pierwszy węzeł od którego rozpoczyna się budowa całej struktury zawierającej poszczególne węzły odpowiadające za sprawdzenie pewnego warunku. Natomiast gałęzie pełnią rolę połączenia pomiędzy węzłami na kolejnych poziomach drzewa [2]. Liście są końcowymi wierzchołkami drzewa i zawierają decyzje. Aby otrzymać decyzję konieczne jest przejście całego drzewa od samego korzenia do wynikowego liścia. Rezultatem takiej operacji będzie klasa decyzyjna (w przypadku drzew klasyfikacyjna) lub np. model regresyjny (w przypadku drzew regresyjnych).

## 1.2 Uczenie maszynowe

W otaczającym nas świecie ilość generowanych oraz gromadzonych informacji nadal przewyższa ilość danych, które można przeanalizować z użyciem obecnych zasobów. Aby analizować tak duże ilości informacji wykorzystywane są najnowsze rozwiązania technologiczne, zarówno na poziomie sprzętu komputerowego oraz oprogramowania. Dzięki zastosowaniu różnych algorytmów przetwarzania danych, klasyfikacji oraz predykcji programy komputerowe posiadają możliwość uczenia się. Tzn. uczenie maszynowe (ang. *machine learning*) jest obszarem sztucznej inteligencji, który zajmuje się wykorzystywaniem komputerowego wspomaganie lub podejmowania decyzji. Uczenie maszynowe w przeciągu ostatniej dekady stało się tak popularne, iż w dużej mierze zdominowało przemysł sztucznej inteligencji oraz przyczyniło się do jej rozwoju [3]. Uczenie maszynowe stanowi trzon wielu usług, serwisów i aplikacji. Pod względem algorytmicznym odpowiada za wyniki wyszukiwania w przeglądarkach oraz za rozpoznawanie mowy przez nasze telefony. Jest



także wykorzystywane w dużo trudniejszych zadaniach, jak sterowanie autonomicznymi samochodami, czy też wspomaganie lotów kosmicznych i operacji chirurgicznych w medycynie.

### 1.3 Drzewa decyzyjne w technikach uczenia maszynowego

Drzewa decyzyjne stanowią jeden z najbardziej rozpowszechnianych mechanizmów w obszarze uczenia maszynowego. Z jednej strony mogą być wykorzystywane w zadaniach z zakresu klasyfikacji, a z drugiej strony również odgrywają ważną rolę w regresji [3]. Mechanizm drzew decyzyjnych pozwala na budowanie modeli na podstawie ogromnych zbiorów uczących. Dodatkowym atutem drzew jest możliwość wizualnego przedstawienia sposobu dojścia do rozwiązania, które będzie zrozumiałe dla osób nie mających do czynienia z uczeniem maszynowym i statystyką.

#### 1.3.1 System GDT

Pracownicy Wydziału Informatyki Politechniki Białostockiej od ponad 20 lat tworzą i rozwijają narzędzie do indukowania drzew decyzyjnych, nazwane GDT (*Global Decision Trees*). Narzędzie te zostało wykorzystane w pracy inżynierskiej. GDT służy do generowania drzew decyzyjnych na podstawie zbiorów wejściowych [1]. System ten jest zaimplementowany w języku C++. Podstawowa jego wersja jest programem konsolowym. Całe rozwiązanie jest unikalne, a głównym założeniem jest wykorzystanie algorytmów ewolucyjnych w procesie indukcji drzew decyzyjnych. Używając algorytmów ewolucyjnych budowane drzewa są bardziej globalne (trudniej wpaść w minimum lokalne) niż w klasycznym podejściu. Skutkuje to możliwością osiągnięcia dokładniejszych i lepszych wyników [4]. Algorytmy ewolucyjne wzorują się na ewolucji biologicznej [5]. Podczas inicjalizacji parametrów algorytmu należy podać takie parametry jak wielkość populacji, prawdopodobieństwo mutacji czy też krzyżowania się danych osobników. Wartości parametrów algorytmu są określane w pliku konfiguracyjnym opartym o strukturę XML (ang. *Extensible Markup Language*), który jest zarazem plikiem wejściowym do aplikacji GDT. Oprócz pliku z opcjami należy także określić pliki ze zbiorem danych:

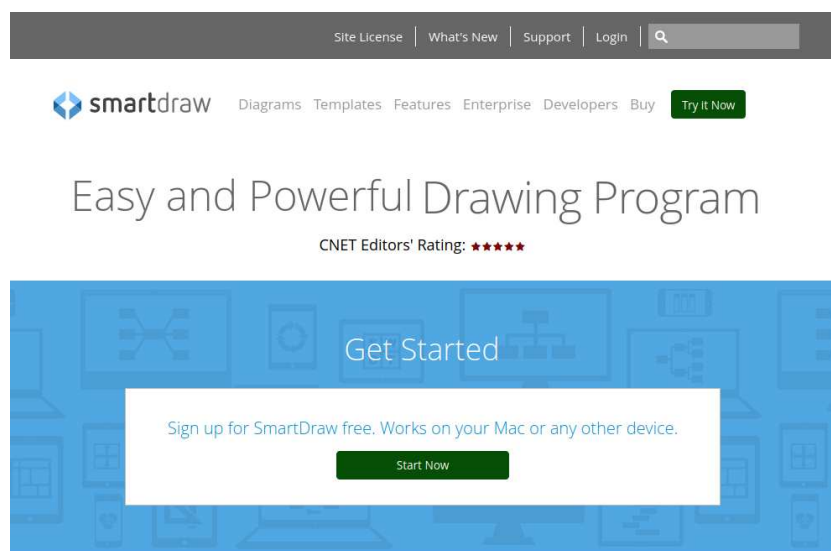
- \*.data - plik zawierający dane treningowe,
- \*.test - plik zawierający dane testowe,

- \*.names - plik określających nazwy klas oraz rodzaj zmiennych.

Wykorzystując określony zbiór trenujący oraz wartości parametrów algorytmu w pliku XML, system GDT indukuje drzewa decyzyjne. Aplikacja zapisuje do plików tekstowych statystyki wynikowe drzewa oraz użyte ustawienia parametrów.

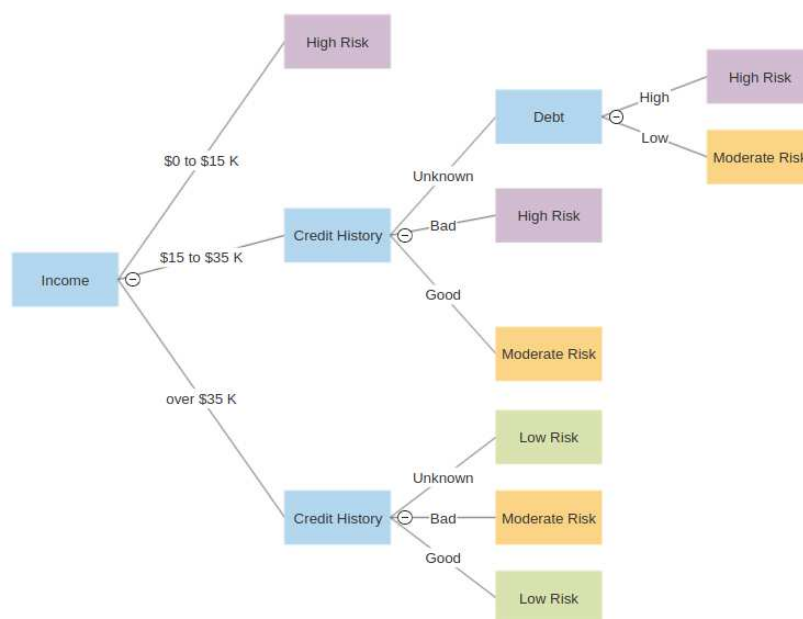
## 1.4 Istniejące rozwiązania

Aktualnie na rynku można znaleźć różne aplikacje pozwalające na budowanie drzew decyzyjnych. Aplikacje te różnią się między sobą zakresem funkcjonalności. Rozwiązania internetowe głównie są nastawione na zarobek, ale oferują też darmowe wersje z pewnymi ograniczeniami. Istnieją też liczne rozwiązania dla programistów w postaci np. bibliotek dla wielu popularnych języków programowania. Takie biblioteki umożliwiają poprzez wykorzystanie modułów stworzenie podstawowych modeli uczenia maszynowego w tym drzew decyzyjnych. Niestety ich wykorzystanie wymaga przynajmniej podstawowej wiedzy z zakresu programowania. Dodatkowo chcąc osiągnąć bardzo wydajne rozwiązania zazwyczaj należy znać szczegóły implementacji biblioteki. Możemy także spotkać aplikacje desktopowe. Wiele takich programów jest rozwijanych przez zespoły naukowe na uniwersytetach. Do wykonania obliczeń wymagają dobrej jakości sprzętu komputerowego, który zapewni odpowiednią moc obliczeniową.



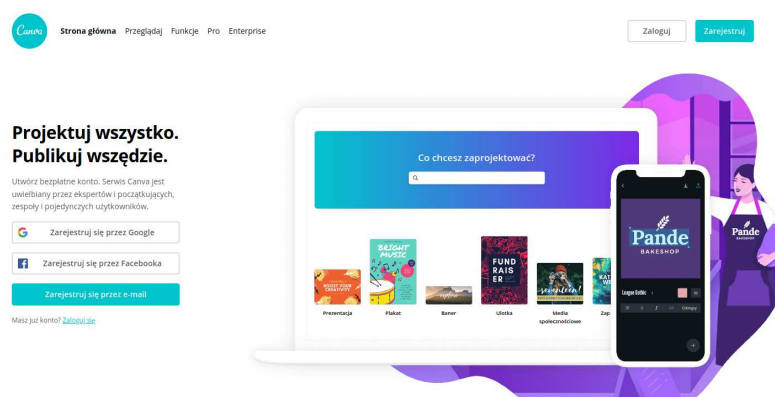
Rysunek 1.1: Strona główna aplikacji *SmartDraw* [6].

Aplikacja internetowa *SmartDraw* jest według autora pracy jedną z wygodniejszych platform do tworzenia drzew decyzyjnych [6]. Użytkownik może za darmo założyć

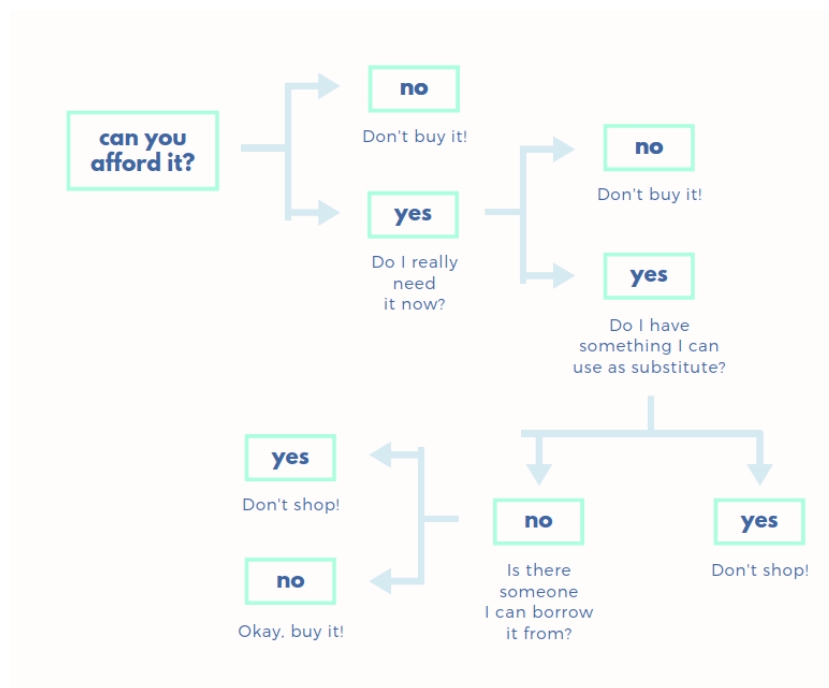


Rysunek 1.2: Drzewo stworzone w aplikacji *SmartDraw*, źródło: opracowanie własne.

konto oraz używać aplikacji bez abonamentu przez okres próbny. Widok strony głównej został przedstawiony na Rys. 1.1. Platforma ponadto udostępnia funkcjonalność tworzenia innych rodzajów diagramów. Proces budowy diagramu (np. w postaci drzewa) polega na przeciąganiu i łączeniu bloków. Istnieje również możliwość wczytania struktury drzewa z pliku o rozszerzeniu \*.csv (ang. *comma-separated values*). Diagramy są prezentowane w czytelny i przejrzysty sposób (Rys. 1.2). Ukończony diagram można wyeksportować do pliku graficznego lub dokumentu programu MS Word. Aplikacja niestety nie umożliwia budowy drzewa używając metody uczenia, jedynie pozwala na graficzne przedstawienie wcześniej przygotowanej struktury.



Rysunek 1.3: Strona główna aplikacji *Canva* [7].



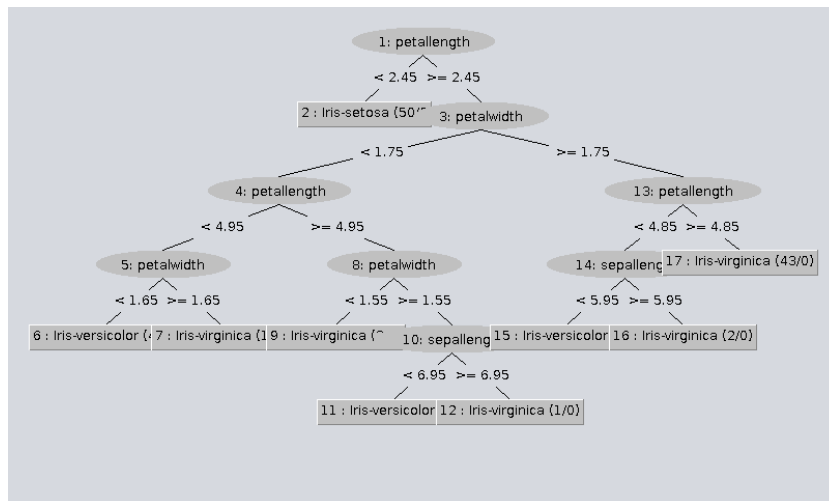
Rysunek 1.4: Drzewo stworzone w aplikacji *Canva*, źródło: opracowanie własne.

*Canva* to kolejne narzędzie umożliwiające tworzenie drzew decyzyjnych przy pomocy przeglądarki internetowej [7]. Dostęp do platformy wymaga założenia konta. Aplikacja także posiada rozszerzony, płatny pakiet funkcjonalności dla firm oraz osób prywatnych. Strona główna aplikacji została przedstawiona na Rys. 1.3. Użytkownik ma możliwość wizualnego przedstawienia metodą przeciągnij i upuść (ang. *drag'n drop*). W aplikacji jednak brakuje opcji indukcji drzewa decyzyjnego z wykorzystaniem zbioru uczącego. Tworzone drzewa można wzbogacić o liczne walory wizualne i dostępne gotowe motywy (Rys. 1.4). Głównymi odbiorcami aplikacji są reklamodawcy oraz osoby prowadzące rozbudowaną działalność w serwisach społecznościowych.

*Weka* jest aplikacją desktopową stworzoną przez naukowców zajmujących się tematami uczenia maszynowego na Uniwersytecie Waikato w Nowej Zelandii. Aplikacja została zaimplementowana w technologii JAVA SE. Pozwala to na jej uruchomienie na różnych systemach operacyjnych. Główne okno aplikacji zostało zaprezentowane na Rys. 1.5. Ponadto naukowcy zaimplementowali liczne tzw. nakładki, umożliwiające korzystanie ze stworzonych mechanizmów za pośrednictwem innych języków programowania. W aplikacji użytkownik ma dostęp do dużej ilości gotowych algorytmów uczenia maszynowego, łącznie z algorytmami indukcji drzew decyzyjnych. Tworzenie nowego eksperymentu zaczyna się od wybrania zestawu danych. Użytkownik początkujący może skorzystać z przykładowych



Rysunek 1.5: Okno główne aplikacji *Weka*, źródło: opracowanie własne.



Rysunek 1.6: Drzewo stworzone w aplikacji *Weka*, źródło: opracowanie własne.

plików. Po wczytaniu zbioru wejściowego istnieje możliwość wizualizacji danych oraz ich wstępnej obróbki. W kolejnym kroku użytkownik wybiera algorytm budowy klasyfikatora. Do wyboru jest kilka różnych algorytmów związanych z tworzeniem drzew decyzyjnych. Czas obliczeń zależy od ilości danych, wybranego algorytmu oraz posiadanych zasobów obliczeniowych. Rezultaty eksperymentu są przedstawiane w postaci tekstowej. Istnieje także możliwość wyświetlenia drzewa graficznie. Przykładowe drzewo decyzyjne stworzone za pośrednictwem programu zostało przedstawione na Rys. 1.6.

Aplikacja tworzona w ramach pracy dyplomowej jest aplikacją oryginalną. Po pierwsze ze względu na wykorzystany system GDT. Po drugie, będzie to aplikacja internetowa, która umożliwi indukowanie drzew decyzyjnych z wykorzystaniem algorytmów

ewolucyjnych. To co będzie ją odróżniać to moduł do zarządzania zadaniami, które będą uruchamiane się zdalnie, przez co użytkownik końcowy nie będzie musiał posiadać znaczących zasobów pamięciowych i obliczeniowych. Budowane narzędzie pozwoli także na graficzną i interaktywną reprezentację otrzymanych wyników.

## Podsumowanie

Założeniem pracy dyplomowej było stworzenie aplikacji internetowej do obsługi systemu GDT. W tym celu został wykorzystany język Python wraz z frameworkiem Django i Django REST Framework. Stworzona aplikacja umożliwia w łatwy sposób zarządzania eksperymentami i danymi. Wyniki eksperymentu są przedstawione w formie graficznej oraz tekstowej. Dodatkowo został stworzony specjalny zestaw ról możliwych do przypisania dla użytkownika. Dzięki temu aplikacja może mieć większe grono użytkowników o różnym stopniu zaawansowania. Zarządzanie uprawnieniami użytkowników jest możliwe poprzez panel administratora. Ponadto administrator ma możliwość edytowania modeli zapisanych w bazie danych na przykład zmiany praw dostępu do akcji eksperymentu.

Stworzona aplikacja internetowa cechuje się wysoką intuicyjnością korzystania, a zarazem łatwością zarządzania. Stosując logiczne rozdzielanie poszczególnych elementów aplikacji w celu zwiększenia stabilności zastosowano konteneryzację z użyciem oprogramowania Docker. Cały system został podzielony na pięć instancji tworzących wspólnie całość. Wykorzystanie takiego rozwiązania umożliwia w dowolnym momencie wymiany kontenerów aplikacji na przykład na ich nowszą wersję. Podział na elementy pozwalana wdrożyć aplikację na kilka serwerów, gdzie każdy będzie odpowiadał za jedną instancję.

Rozwiązanie stworzone w ramach pracy dyplomowej zostało wdrożone i udostępnione użytkownikom pod adresem „<http://decisiontree.pl>”. Dostęp do systemu jest możliwy za pośrednictwem dowolnej przeglądarki internetowej. Mechanizm zakładania kont i logowania jest udostępniony dla każdego. Bardziej zaawansowane zagadnienia wymagają uprzedniej autoryzacji. Wraz ze wzrostem zainteresowania istnieje możliwość rozbudowania zaimplementowanego oprogramowania o dodatkowe metody uczenia maszynowego, czy też mechanizm predykcji oraz bardziej zaawansowane drzewa decyzyjne (jak np. modelowe).

## Bibliografia

- [1] Kretowski M. *Evolutionary Decision Trees in Large-Scale Data Mining*. Studies in Big Data, vol. 59, Springer.
- [2] Lucid Software Inc. Lucidchart. <https://www.lucidchart.com/pages/decision-tree>, stan z 25.11.2019 r.
- [3] Aurélien Géron. *Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow*. Helion S.A., 2018.
- [4] Kretowski M. Jurczuk K., Czajkowski M. *Evolutionary Induction of Decision Tree for Large Scale Data*. Soft Computing, vol. 21: 7363-79.
- [5] Prateek Joshi. *Artificial Intelligence with Python*. Packt Publishing, 2017.
- [6] SmartDraw. Smartdraw. <https://www.smartdraw.com/>, stan z 17.12.2019 r.
- [7] Canva. Canva. <https://www.canva.com/>, stan z 17.12.2019 r.
- [8] Python Software Foundation. Python. <https://www.python.org/>, stan z 08.12.2019 r.
- [9] Django Software Foundation. Django. <https://www.djangoproject.com/>, stan z 10.12.2019 r.
- [10] Collaboratively funded project. Django rest framework. <https://www.django-rest-framework.org/>, stan z 10.12.2019 r.
- [11] The PostgreSQL Global Development Group. Postgresql. <https://www.postgresql.org/>, stan z 10.12.2019 r.
- [12] Mozilla and individual contributors. Javascript. <https://developer.mozilla.org/pl/docs/Web/JavaScript>, stan z 10.12.2019 r.
- [13] Facebook Inc. Reactjs. <https://reactjs.org/>, stan z 10.12.2019 r.
- [14] Inc. GitHub. Github. <https://github.com/search?o=desc&q=stars%3A%3E1&s=stars&type=Repositories>, stan z 10.12.2019 r.



- [15] Axios. Axios. <https://github.com/axios/axios>, stan z 10.12.2019 r.
- [16] Ask Solem and contributors. Celery project. <http://www.celeryproject.org/>, stan z 10.12.2019 r.
- [17] Inc. Pivotal Software. Rabbitmq. <https://www.rabbitmq.com/>, stan z 10.12.2019 r.
- [18] Benoit Chesneau and contributors. Gunicorn. <https://gunicorn.org/>, stan z 10.12.2019 r.
- [19] Inc. NGINX. Nginx. <https://www.nginx.com/>, stan z 10.12.2019 r.
- [20] Docker Inc. Docker. <https://www.docker.com/>, stan z 10.12.2019 r.
- [21] Locustio. Locust. <https://locust.io/>, stan z 17.12.2019 r.
- [22] Machine Learning Group at the University of Waikato. Weka. <https://www.cs.waikato.ac.nz/ml/weka/>, stan z 17.12.2019 r.
- [23] Mark Lutz. *Python. Wprowadzenie. Wydanie IV*. Helion S.A., 2010.

## **Spis tabel**

## Spis rysunków

Rysunek 1.1	Strona główna aplikacji <i>SmartDraw</i> [6]. . . . .	11
Rysunek 1.2	Drzewo stworzone w aplikacji <i>SmartDraw</i> , źródło: opracowanie własne. . . . .	12
Rysunek 1.3	Strona główna aplikacji <i>Canva</i> [7]. . . . .	12
Rysunek 1.4	Drzewo stworzone w aplikacji <i>Canva</i> , źródło: opracowanie własne.	13
Rysunek 1.5	Okno główne aplikacji <i>Weka</i> , źródło: opracowanie własne. . . . .	14
Rysunek 1.6	Drzewo stworzone w aplikacji <i>Weka</i> , źródło: opracowanie własne. .	14

## **Spis listingów**