

POLITECHNIKA BIAŁOSTOCKA

WYDZIAŁ INFORMATYKI

KATEDRA KATEDRA OPROGRAMOWANIA

PRACA DYPLOMOWA INŻYNIERSKA

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

WYKONAWCA: MATEUSZ PERNAL

.....

podpis

PROMOTOR: DR INŻ. KRZYSZTOF JURCZUK

.....

podpis

BIAŁYSTOK 2019 r.

Karta dyplomowa

Politechnika Białostocka Wydział Informatyki Katedra Katedra Oprogramowania	Studia stacjonarne studia I stopnia	Numer albumu studenta: 101420
		Rok akademicki 2099/2020
		Kierunek studiów: informatyka Specjalność: Brak
Mateusz Pernal TEMAT PRACY DYPLOMOWEJ: Aplikacja internetowa do obsługi system gdt w technologii Python/Django. Zakres pracy: 1. Zakres 1 2. Zakres 2 3. Zakres 3		
<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: 35%;"> 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: 35%;"> Ocena recenzenta </div> <div style="width: 30%;"> Podpis recenzenta </div> </div>		

Thesis topic:

Temat po angielsku.

Summary

Abstrakt po angielsku

Spis treści

Streszczenie	3
Wprowadzenie	5
1 Przedstawienie problemu	6
1.1 Drzewa decyzyjne	6
1.2 Uczenie maszynowe	6
1.3 Drzewa decyzyjne w technikach uczenia maszynowego	7
1.4 Istniejące rozwiązania	8
2 Wizja aplikacji	9
2.1 Wymagania funkcjonalne	9
2.2 Wymagania нефункционалне	14
2.3 Wykorzystane technologie	14
Podsumowanie	15
Bibliografia	16
Spis tabel	17
Spis rysunków	18
Spis listingów	19
Spis algorytmów	20

Wprowadzenie

Tu będzie wstęp

Cel pracy

Celem pracy jest stworzenie aplikacji webowej umożliwiającej obsługę systemu GDT (*Global Decision trees*) służącego do generowania drzew decyzyjnych. Strona internetowa umożliwi tworzenie, uruchamianie oraz zarządzanie zadaniami uruchamianymi przy pomocy systemu. Jedną z ważniejszych cech programu powinno być danie użytkownikowi możliwość ustawienia parametrów konfiguracyjnych przed wystartowaniem zadania. Aplikacja internetowa powinna także udostępniać opcje związane z wyświetleniem drzewa w postaci graficznej oraz przedstawieniem wyników eksperymentu.

Zakres pracy

Zakres pracy obejmuje:

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

Organizacja pracy

Tu będzie organizacja pracy/ zawartość pracy

1. Przedstawienie problemu

1.1 Drzewa decyzyjne

Podjęcie decyzji jest procesem myślowym, który od początku istnienia ludzkości stwarza pewne trudności, a polega on na wybraniu najlepszego rozwiązania z dostępnych. 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 w postaci diagramu. Sam diagram może przybrać formę graficzną drzewa decyzyjnego.

Podstawowymi elementami drzewa są korzeń, gałęzie, węzły oraz liście. Korzeniem jest decyzja od którego rozpoczyna się budowa całej struktury, której poszczególne węzły, odpowiadające za sprawdzenie pewnego warunku, są połączone gałęziami.[2] Liście są końcowymi wierzchołkami drzewa i określają wybraną decyzję. Podczas próby określenia decyzji, należy poddać klasyfikacji posiadane dane. Aby to zrobić konieczne jest przejście całego drzewa od samego korzenia do wynikowego liścia. Rezultatem takiej operacji będzie klasa definiująca decyzję.

1.2 Uczenie maszynowe

W otaczającym nas świecie ilość informacji produkowanych przez otoczenie oraz zbieranych przez firmy czy instytucje nadal przewyższa ilość danych, które można przeanalizować z użyciem obecnych zasobów. Chcąc choć w jakimś stopniu poddać analizie duże ilości danych, aby na ich podstawie wyciągnąć wnioski używa się licznych rozwiązań technologicznych. Dzięki zastosowaniu różnych algorytmów przetwarzania danych, klasyfikacji oraz predykcji istnieje możliwość umożliwienia uczenia się programowi komputerowemu. Kierunek nauki, który zajmuje się tą dziedziną nazywamy uczeniem maszynowym. W ciągu ostatnich dziesięciu lat entuzjazm związany z wykorzystywaniem tej technologii wzrósł gwałtownie i w dużej mierze zdominował przemysł, ale również przyczynił się do jej rozwoju.[1] Uczenie maszynowe stanowi trzon wielu usług, serwisów i aplikacji. Pod względem technologicznym odpowiada za wyniki wyszukiwania w przeglądarkach, za rozpoznawanie mowy przez nasze telefony, ale także jest odpowiedzialne za prowadzenie autonomicznych samochodów.

1.3 Drzewa decyzyjne w technikach uczenia maszynowego

Drzewa decyzyjne stanowią jedno z bardziej wszechstronnych algorytmów w dziedzinie uczenia maszynowego. Z jednej strony mogą być wykorzystywane w zadaniach z zakresu klasyfikacji, a z drugiej strony również odgrywają ważną rolę w regresji.[1] Z ich pomocą możemy uzyskać potężne modele i narzędzia zdolne do uczenia się ze złożonych zbiorów danych. Dodatkowym atutem drzew jest możliwość wizualnego przedstawienia rozwiązania, które będzie zrozumiałe dla osób nie mających do czynienia z uczeniem maszynowym lub ze statystyką. Z racji wzrostu popularności tej technologii zwiększyły się nakłady pracy naukowej w celu osiągnięcia coraz to lepszych i bardziej optymalnych algorytmów pod względem wydajnościowym.

1.3.1 System GDT

Pracownicy Politechniki Białostockiej również mają wkład w budowę takich rozwiązań. Autorski system GDT (*Global Decision Trees*), który jest wykorzystywany w aplikacji inżynierskiej, służy do generowania modelu drzewa decyzyjnego na podstawie zbiorów wejściowych. Ten system jest zaimplementowany w języku c++ oraz jest skompilowany do pliku wykonywalnego, aby umożliwić jego uruchomienie z poziomu konsoli systemu operacyjnego. Całe rozwiązanie jest unikalne i nie ma takiego drugiego identycznego, a głównym założeniem jest wykorzystanie algorytmów genetycznych. Z ich pomocą przestrzeń rozwiązań danego problemu jest większa niż w klasycznym podejściu, co skutkuje możliwością osiągnięcia dokładniejszych i lepszych wyników. Metody pracy algorytmów genetycznych w dużej mierze odwzorowują działania samej natury.[4] Podczas definiowania pracy algorytmu należy podać takie parametry jak wielkość populacji, prawdopodobieństwo mutacji czy też krzyżowania się danych osobników. Wartości tych parametrów i innych są określane w pliku konfiguracyjnym opartym o strukturę XML, który jest zarazem plikiem wejściowym do aplikacji GDT. System oprócz tego pliku wykorzystuje pliki z konkretnymi rozszerzeniami:

- *.data - plik zawierający dane treningowe,
- *.test - plik zawierający dane testowymi,
- *.names - plik określających nazwy klas oraz rodzaj zmiennych.

Na podstawie tych danych aplikacja GDT może stworzyć model drzewa decyzyjnego, którego przedstawienie jest zapisywane w pliku tekstowym.

1.4 Istniejące rozwiązania

2. Wizja aplikacji

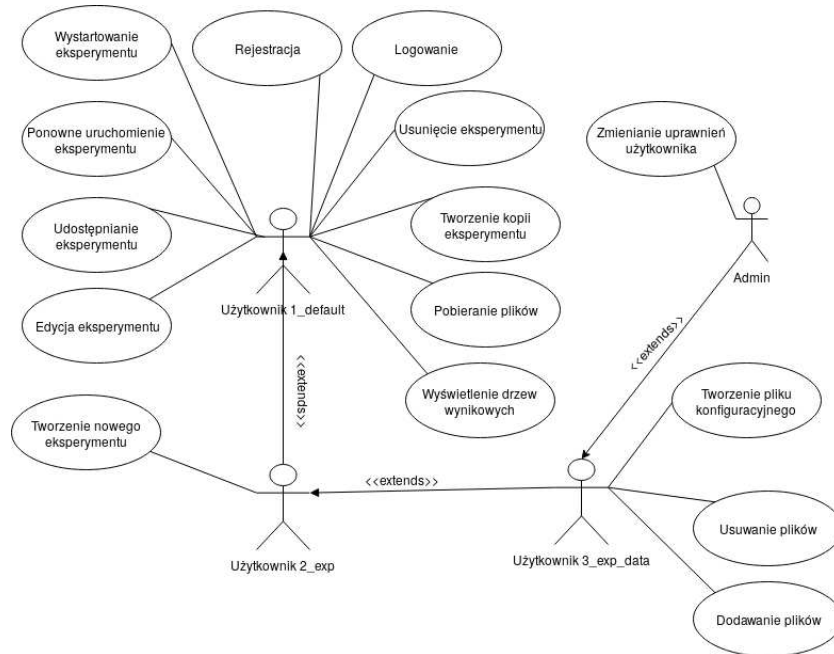
2.1 Wymagania funkcjonalne

Tworzenie aplikacji należało zacząć od nakreślenia zakresu funkcjonalności, które aplikacja będzie udostępniać użytkownikom. Podstawowym zadaniem, które musi spełniać jest możliwość przeprowadzania eksperymentów przy pomocy systemu GDT. Kolejnym ważnym aspektem jest możliwość zarządzania, wyświetlania, udostępniania oraz edycji poszczególnych eksperymentów. Każdy z użytkowników powinien konkretnie widzieć, które z przeprowadzanych przez niego doświadczeń zostały już ukończone, a które jeszcze są w trakcie lub czekają w kolejce do obliczeń. Aplikacja powinna również przede wszystkim w przejrzysty sposób wyświetlać wyniki doświadczenia w postaci wygenerowanego drzewa decyzyjnego i poszczególnych statystyk. Podczas uruchomienia nowego zadania do obliczenia, użytkownikowi zostanie wyświetlony pasek postępu oraz oszacowana długość trwania całego zadania, przy czym w dowolnym momencie będzie mógł anulować polecenie. Wraz z możliwością tworzenia eksperymentu nie odłącznym elementem będzie funkcjonalność zarządzania plikami wejściowymi oraz wyjściowymi. Dla użytkowników początkujących zostanie przedstawiona opcja tworzenia podstawowych plików konfiguracyjnych, bez wgłębiania się w bardziej zaawansowane parametry eksperymentu. Dostęp do całości funkcjonalności powinien być tylko dostępny dla zarejestrowanych użytkowników. Natomiast możliwość rejestracji oraz logowania będzie ogólnodostępna.

System kont użytkowników powinien wyróżniać różne role, które definiowałyby dostęp do poszczególnych funkcjonalności aplikacji. Zarządzanie tymi uprawnieniami będzie się odbywać po przez panel administratora. Administrator aplikacji dodatkowo może modyfikować oraz usuwać konta użytkowników. Co więcej z interfejsu admina będzie istniała możliwość edycji rekordów z bazy danych oraz edycja uprawnień do poszczególnych eksperymentów.

Biorąc pod uwagę możliwość udostępniania przez użytkownika doświadczenia innemu użytkownikowi, ważnym aspektem będzie umożliwienie ograniczenia części akcji możliwych do wykonania na eksperymencie. Podstawowym uprawnieniem, którego nie da się zablokować będzie możliwość wyświetlenia wyników eksperymentu. Natomiast reszta funkcjonalności możliwych do wykonania na doświadczeniu, takich jak uruchamianie,

kopiowanie, edycja, usuwanie czy też pobieranie plików wejściowych lub wyjściowych może zostać ograniczona. Użytkownik posiadający od kogoś udostępniony eksperyment z pewnymi ograniczeniami, może udostępnić go dalej tylko jeśli posiada nadane prawa do udostępniania, przy czym nie może znieść już nadanych wcześniej ograniczeń.

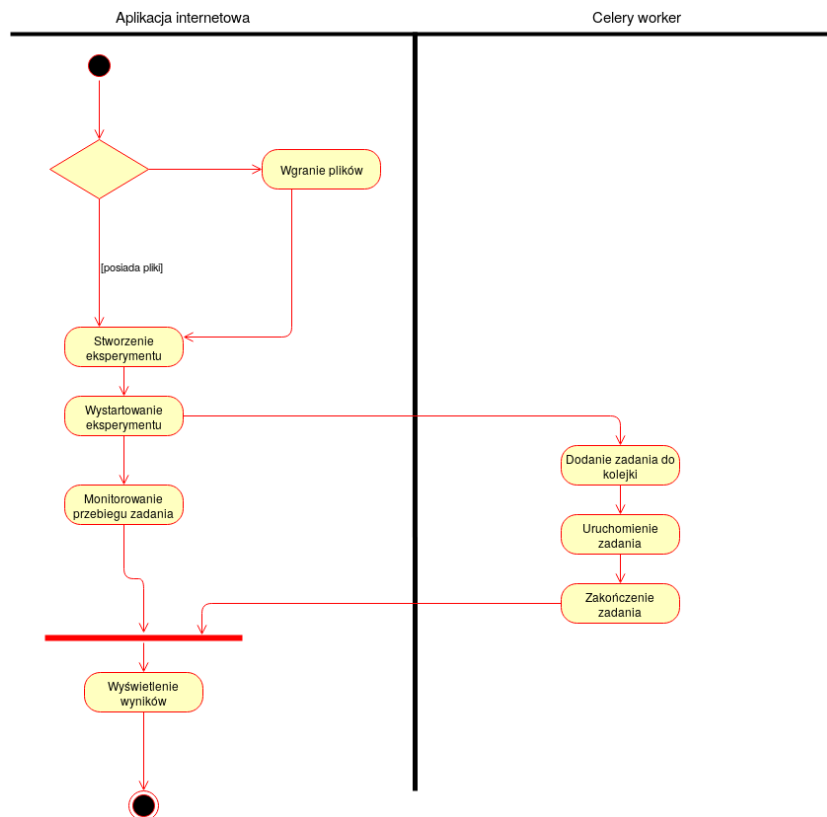


Rysunek 2.1: Diagram przypadków użycia, źródło: opracowanie własne

Na rysunku 2.1 przedstawiono wszystkie funkcjonalności w postaci diagramu przypadków użycia. W aplikacji zostały wyszczególnione trzy role dostępne do uzyskania dla każdego użytkownika oraz rola administratora całego systemu. Wszystkie przypadki użycia oprócz logowania i rejestracji są dostępne tylko dla użytkowników zalogowanych. Każdy nowy użytkownik musi założyć konto, aby mieć dostęp do aplikacji. Nowo powstałe konta otrzymują uprawnienia na domyślnym poziomie "1_default", a wyższe poziomy uprawnień mogą zostać nadane po przez administratora. Kolejne role rozszerzają możliwości użytkownika pod względem ilości akcji możliwych do wykonania. Poziom "2_exp" umożliwia tworzenie nowych eksperymentów, przy czym tylko najwyższy poziom uprawnień "3_exp_data" pozwala na wgrywanie plików do aplikacji. Przebieg czynności związanych ze stworzeniem nowego eksperymentu oraz wyświetleniem wyników został przedstawiony na rysunku 2.2.

Opis przypadku użycia "Tworzenie nowego eksperymentu":

1. Aktor



Rysunek 2.2: Diagram czynności tworzenia i uruchamiania eksperymentu, źródło: opracowanie własne

- Użytkownik.

2. Warunki początkowe

- Aktor jest zalogowany oraz posiada uprawnienia przynajmniej na poziomie "2_exp".

3. Zdarzenie inicjujące

- Naciśnięcie przycisku **New experiment** nad listą wszystkich eksperymentów użytkownika.

4. Przebieg w krokach

- Aplikacja przechodzi do formularza tworzenia nowego eksperymentu,
- Użytkownik wypełnia i zatwierdza formularz.

5. Przebiegi alternatywne

- Użytkownik nie uzupełnia wszystkich pól formularza, aplikacja wyświetla powiadomienie o pustych polach.

6. Sytuacje wyjątkowe

- Użytkownik nie posiada żadnych plików wgranych do aplikacji. Powoduje to, że pola formularza zawierające pliki są puste. Uniemożliwia to stworzenie nowego eksperymentu, a aplikacja wyświetla powiadomienie o pustych polach przy podjętej próbie zatwierdzenia.

7. Warunki końcowe

- System przekierowuje użytkownika do listy z eksperymentami, a na liście znajduje się nowo utworzony eksperyment.

8. Zależności czasowe

- Częstotliwość wykonywania: Około 20 razy dziennie na każdego użytkownika,
- Typowy czas realizacji: 8 sekund.

Opis przypadku użycia "Wystartowanie eksperymentu":

1. Aktor

- Użytkownik.

2. Warunki początkowe

- Aktor jest zalogowany oraz posiada stworzony eksperyment.

3. Zdarzenie inicjujące

- Naciśnięcie przycisku "Show" w liście eksperymentów na elemencie, którego status to "Created".

4. Przebieg w krokach

- Aplikacja przechodzi do podglądu szczegółów wybranego eksperymentu,
- Użytkownik klika przycisk "Start" znajdujący się na pasku możliwych czynności,
- Aplikacja przekierowuje użytkownika do listy eksperymentów.

5. Przebiegi alternatywne

- Po wystartowaniu eksperymentu nastąpił błąd i jest to sygnalizowane zmianą statusu na "Error", a w szczegółach eksperymentu można podejrzewać wiadomość z błędem.

6. Sytuacje wyjątkowe

- Użytkownikowi nie posiada praw do wystartowania konkretnego eksperymentu i w panelu akcji nie wyświetla się przycisk "Start".

7. Warunki końcowe

- Eksperyment zmienił swój status na "In queue" lub "Running", a po przejściu do szczegółów wyświetla się pasek postępu oraz szacowany czas oczekiwania na zakończenie.

8. Zależności czasowe

- Częstotliwość wykonywania: Około 20 razy dziennie na każdego użytkownika,
- Typowy czas realizacji: 8 sekund.

Opis przypadku użycia "Wyświetlenie drzewa wynikowego":

1. Aktor

- Użytkownik.

2. Warunki początkowe

- Aktor jest zalogowany oraz posiada ukończony eksperyment.

3. Zdarzenie inicjujące

- Naciśnięcie przycisku "Show" w liście eksperymentów na elemencie, którego status to "Finished".

4. Przebieg w krokach

- Aplikacja przechodzi do podglądu szczegółów wybranego eksperymentu, a na samym dole karty wyświetlają się linki do drzew decyzyjnych,

- Użytkownik klika w link do drzewa decyzyjnego.

5. Przebiegi alternatywne

- Brak.

6. Sytuacje wyjątkowe

- Brak.

7. Warunki końcowe

- Aplikacja wyświetliła drzewo decyzyjne wraz ze statystykami.

8. Zależności czasowe

- Częstotliwość wykonywania: Około 30 razy dziennie na każdego użytkownika,
- Typowy czas realizacji: 10 sekund.

2.2 Wymagania niefunkcjonalne

asdasd.

2.3 Wykorzystane technologie

asdasd.

Podsumowanie

Tutaj będzie podsumowanie.

Bibliografia

- [1] Aurélien Geron. *Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow*. Helion S.A., 2018.
- [2] Lucid Software Inc. Lucidchart. <https://www.lucidchart.com/pages/decision-tree>, stan z 25.11.2019 r.
- [3] B. Inny. Tytuł³ publikacji. In *Tytuł książki*, pages 5–32, Feb 2011.
- [4] Prateek Joshi. *Artificial Intelligence with Python*. Packt Publishing, 2017.

Spis tabel

Spis rysunków

Rysunek 2.1 Diagram przypadków użycia, źródło: opracowanie własne 10

Rysunek 2.2 Diagram czynności tworzenia i uruchomiania eksperymentu, źródło: opracowanie wł

Spis listingów

Spis algorytmów