Uniwersytet Warmińsko-Mazurski w Olsztynie

Wydział Matematyki i Informatyki

Kierunek: ISI

Internetowa strona do głosowania online

Borys Kutsenko 170271

Viktoria Onyshchenko

Olsztyn 2025 rok

Krótkie podsumowanie

Celem pracy jest stworzenie pełnoprawnej aplikacji internetowej do głosowania online. Aplikacja została zaprojektowana z myślą o prostocie użytkowania oraz bezpieczeństwie procesu głosowania. System działa lokalnie w środowisku kontenerowym Docker, co zapewnia łatwą konfigurację, izolację środowiska oraz możliwość szybkiego wdrażania na różnych systemach operacyjnych. Użytkownicy mogą oddawać głosy za pośrednictwem przeglądarki internetowej, łącząc się z aplikacją przez adres IP lub nazwę domeny.

Projekt został zrealizowany z użyciem frameworka Django w języku Python. Komunikacja asynchroniczna i zadania w tle obsługiwane są przy użyciu Celery i brokera Redis. W ramach aplikacji wyróżniono dwie główne części funkcjonalne: moduł głosowania dostępny dla wszystkich użytkowników oraz panel tworzenia nowych ankiet dostępny jedynie po zalogowaniu. Aplikacja obsługuje również statystyki głosowania, które prezentowane są w sposób czytelny i interaktywny.

W trakcie realizacji projektu zastosowano zasady bezpiecznego przetwarzania danych oraz dobrych praktyk w zakresie projektowania systemów webowych. Interfejs użytkownika został przygotowany z wykorzystaniem HTML, CSS oraz frameworka Bootstrap, co zapewnia responsywność i estetyczny wygląd.

Uzyskane wyniki potwierdzają, że aplikacja działa zgodnie z założeniami, pozwalając na skuteczne przeprowadzanie głosowań w zamkniętym środowisku lokalnym. System może być również łatwo dostosowany do potrzeb szerszego wdrożenia, w tym w środowisku produkcyjnym. Wnioskiem końcowym jest potwierdzenie, że wykorzystane technologie i podejście umożliwiają stworzenie niezawodnego i nowoczesnego narzędzia do głosowania online.

Spis treści:

1.

**Wstęp**

W dzisiejszych czasach coraz większe znaczenie zyskują rozwiązania cyfrowe umożliwiające przeprowadzanie różnego rodzaju procesów w sposób zdalny. Jednym z obszarów, który szczególnie korzysta z postępu technologicznego, jest organizacja głosowań – zarówno w celach społecznych, edukacyjnych, jak i firmowych. Tradycyjne metody głosowania, oparte na papierze lub bezpośrednim kontakcie, wiążą się z kosztami, czasochłonnością oraz ograniczoną dostępnością. W związku z tym celem niniejszej pracy jest stworzenie aplikacji internetowej do głosowania online, która umożliwia użytkownikom łatwe, szybkie i bezpieczne oddawanie głosów za pośrednictwem przeglądarki internetowej.

Motywacją do podjęcia tego tematu była rosnąca potrzeba wdrażania zautomatyzowanych systemów głosowania w środowiskach lokalnych – takich jak szkoły, uczelnie czy organizacje – które nie zawsze mają dostęp do gotowych, komercyjnych rozwiązań. Projekt aplikacji jest odpowiedzią na aktualne wyzwania związane z cyfryzacją codziennych procesów i dąży do udowodnienia, że nawet przy wykorzystaniu darmowych technologii open-source możliwe jest stworzenie w pełni funkcjonalnego, niezawodnego systemu. Temat ten jest szczególnie aktualny w kontekście coraz częstszych debat na temat bezpieczeństwa i transparentności cyfrowych form komunikacji i głosowania.

Zakres pracy obejmuje analizę potrzeb użytkowników, projektowanie oraz implementację aplikacji do głosowania w środowisku kontenerowym Docker. Skupiono się na stworzeniu rozwiązania lokalnego, z ograniczonym dostępem zewnętrznym, co pozwala lepiej kontrolować środowisko i zapewnić jego stabilność. Pominięto natomiast bardziej zaawansowane aspekty skalowania aplikacji na dużą liczbę użytkowników oraz integrację z zewnętrznymi bazami danych i systemami uwierzytelniania zewnętrznego, koncentrując się na wersji demonstracyjnej systemu, działającej w środowisku lokalnym.

Struktura pracy została podzielona na kilka głównych rozdziałów. Pierwszy rozdział zawiera przegląd literatury oraz istniejących rozwiązań w zakresie głosowania online. Drugi rozdział przedstawia użyte technologie, takie jak Django, Docker, Celery, Redis oraz sposób ich wzajemnej integracji. W trzecim rozdziale opisano proces projektowania aplikacji – od planowania interfejsu użytkownika po organizację bazy danych. Czwarty rozdział koncentruje się na implementacji systemu oraz testach funkcjonalnych. W ostatnim rozdziale omówiono wyniki przeprowadzonych testów, wnioski z pracy oraz możliwe kierunki dalszego rozwoju projektu.

Tło teoretyczne – uczenie maszynowe

Uczenie maszynowe (ang. *machine learning*) to dziedzina sztucznej inteligencji zajmująca się tworzeniem algorytmów, które potrafią uczyć się na podstawie danych i podejmować decyzje bez konieczności programowania każdej reguły. W ostatnich latach uczenie maszynowe znalazło szerokie zastosowanie w różnych dziedzinach – od medycyny, przez finanse, aż po analizę danych użytkowników w aplikacjach internetowych.

Jednym z najczęściej wykorzystywanych podejść w kontekście klasyfikacji i analizy danych są **drzewa decyzyjne (ang. Decision Trees)** oraz **losowe lasy (ang. Random Forests)**. Drzewa decyzyjne to struktury hierarchiczne, które dzielą dane wejściowe na podstawie zadanych kryteriów, prowadząc do decyzji końcowej. Ich zaletą jest przejrzystość i łatwa interpretacja wyników. Z kolei losowe lasy to metoda zespołowa (ang. ensemble learning), która łączy wiele drzew decyzyjnych w jeden model, zwiększając tym samym dokładność i odporność na przeuczenie (*overfitting*).

W literaturze naukowej, m.in. w pracach Breimana (2001) oraz Quinlana (1986), podkreślono skuteczność tych algorytmów w analizie danych kategorii, co czyni je doskonałym narzędziem do analizy wyników głosowań, gdzie klasyfikacja może dotyczyć np. wykrywania anomalii, przewidywania trendów lub grupowania użytkowników według preferencji.

Analiza istniejących rozwiązań

Istnieje wiele aplikacji i systemów głosowania online, takich jak **Google Forms**, **surveymonkey, Typeform,** **StrawPoll** czy platformy open-source, jak **Helios Voting**. Większość z nich skupia się na aspekcie użytkowym i interfejsie graficznym, pomijając zaawansowaną analizę wyników po zakończeniu głosowania. Część rozwiązań komercyjnych wykorzystuje statystykę do prezentacji wyników, ale tylko nieliczne oferują jakąkolwiek formę predykcji lub klasyfikacji wyników z użyciem sztucznej inteligencji.

Nasza aplikacja wyróżnia się na tle istniejących rozwiązań dzięki zastosowaniu mechanizmów uczenia maszynowego, które pozwalają na pogłębioną analizę oddanych głosów. W ramach projektu zaimplementowano model drzewa decyzyjnego oraz losowego lasu przy użyciu biblioteki scikit-learn, co umożliwia m.in. wykrywanie nietypowych zachowań użytkowników, predykcję przyszłych głosów na podstawie wzorców oraz segmentację danych.

Podsumowując, integracja metod klasyfikacyjnych z systemem głosowania online to innowacyjne podejście, które może otworzyć nowe możliwości analizy i interpretacji wyników, a także podnieść poziom bezpieczeństwa i zaufania do systemu.

Cel główny

Celem głównym niniejszej pracy dyplomowej jest zaprojektowanie i implementacja kompletnej, nowoczesnej aplikacji internetowej przeznaczonej do głosowania online, która będzie działać lokalnie w środowisku kontenerowym Docker. Aplikacja ta ma umożliwiać użytkownikom bezproblemowe tworzenie, udostępnianie i udział w głosowaniach poprzez prosty interfejs webowy dostępny z poziomu przeglądarki internetowej. Głosowania mogą być przeprowadzane bez konieczności logowania się, jednak proces ich tworzenia oraz analiza wyników będą dostępne jedynie dla zalogowanych użytkowników. Jednym z kluczowych aspektów systemu jest jego rozszerzenie o elementy uczenia maszynowego — konkretnie algorytmy drzew decyzyjnych oraz losowych lasów — których zadaniem jest analiza zebranych danych z głosowania i prezentacja wniosków w sposób zrozumiały dla użytkownika. Celem jest zatem nie tylko umożliwienie samego głosowania, ale także pogłębiona interpretacja wyników, co pozwala na bardziej świadome decyzje i rozumienie preferencji uczestników.

System ma działać w sposób niezależny, w pełni lokalnie, co oznacza, że nie wymaga zewnętrznych serwerów ani usług w chmurze. Taki model działania zwiększa bezpieczeństwo danych, daje pełną kontrolę nad środowiskiem oraz czyni aplikację użyteczną w warunkach ograniczonego dostępu do internetu. Aplikacja ma być również modularna i łatwa do dalszego rozwijania, co stanowi dodatkowy cel projektu – przygotowanie solidnej bazy do potencjalnej rozbudowy funkcjonalnej w przyszłości.

Zakres zastosowania

Zaprojektowany system głosowania może być wykorzystywany w bardzo szerokim zakresie kontekstów – zarówno edukacyjnych, jak i zawodowych czy społecznych. Przykładowo, aplikacja może służyć do przeprowadzania anonimowych głosowań w szkołach i na uczelniach, np. w celu wyboru samorządu studenckiego, zbierania opinii o przedmiotach dydaktycznych lub organizacji nieformalnych plebiscytów. W środowiskach firmowych może być wykorzystana do podejmowania decyzji zespołowych, ankietowania pracowników lub organizowania szybkich konsultacji bez konieczności organizowania spotkań. System nadaje się również do użytku w organizacjach non-profit, stowarzyszeniach lokalnych, wspólnotach mieszkaniowych czy grupach hobbystycznych, gdzie może wspierać podejmowanie decyzji kolektywnych w sposób demokratyczny i przejrzysty.

Warto podkreślić, że lokalna natura systemu czyni go szczególnie użytecznym w środowiskach, w których zachowanie pełnej kontroli nad danymi jest priorytetem – np. w instytucjach administracji publicznej, zamkniętych sieciach korporacyjnych, czy tam, gdzie występują ograniczenia związane z przetwarzaniem danych w chmurze. Dodatkowo, zastosowanie uczenia maszynowego pozwala na wykorzystanie systemu również w celach edukacyjnych, badawczych lub eksperymentalnych – umożliwiając testowanie wpływu różnych czynników na wyniki głosowania.

Technologie i narzędzia

W realizacji projektu zastosowano szereg nowoczesnych, sprawdzonych technologii, narzędzi oraz bibliotek, które wspólnie tworzą stabilne i elastyczne środowisko aplikacyjne. Językiem programowania wykorzystanym w warstwie backendowej jest Python, który w połączeniu z frameworkiem Django zapewnia szybkie tworzenie logicznej struktury aplikacji, opierającej się na architekturze Model-View-Template (MVT). Django zostało wybrane ze względu na swoją prostotę, bezpieczeństwo oraz bogaty ekosystem rozszerzeń i dokumentacji.

W celu obsługi zadań asynchronicznych (np. wysyłania e-maili czy przeliczania wyników) użyto biblioteki Celery, współpracującej z systemem kolejkowania wiadomości Redis. To rozwiązanie pozwala na wydajne przetwarzanie operacji w tle bez obciążania głównego wątku aplikacji. Dane użytkowników, głosów oraz statystyk przechowywane są w relacyjnej bazie danych PostgreSQL, która zapewnia stabilność i dobre wsparcie dla operacji analitycznych.

Do analizy danych zastosowano bibliotekę scikit-learn, znaną z wydajnej implementacji algorytmów uczenia maszynowego. W pracy skupiono się na dwóch metodach: drzewach decyzyjnych (Decision Trees) oraz losowych lasach (Random Forests). Algorytmy te pozwalają zidentyfikować najważniejsze cechy wpływające na wybory użytkowników oraz stworzyć model predykcyjny na podstawie zebranych danych.

Cała aplikacja została umieszczona w kontenerach Docker, co ułatwia jej uruchomienie w dowolnym środowisku bez konieczności instalowania zależności ręcznie. Docker umożliwia również łatwe aktualizacje, testowanie oraz wdrażanie aplikacji na serwerze lokalnym lub zdalnym. Interfejs użytkownika został przygotowany z użyciem HTML5, CSS3, JavaScript oraz biblioteki Tailwind CSS, co pozwala na tworzenie nowoczesnych, responsywnych widoków zgodnych z aktualnymi trendami UI/UX.

Tak dobrany zestaw technologii zapewnia nie tylko sprawne działanie aplikacji, ale także jej łatwość rozbudowy, bezpieczeństwo i niezależność od platformy systemowej.