

Projekt zespołowy

Temat	<i>Projekt algorytmu do przewidywania cen akcji na giełdzie na podstawie danych z przeszłości</i>	
Zespół	<i>Imię i nazwisko</i>	<i>E-mail</i>
	1. <i>Mateusz Wiśniewski</i>	
	2. <i>Przemysław Sawoniuk</i>	
	3. <i>Adam Szymczyk</i>	

Kierunek studiów: *Modelowanie Matematyczne i Analiza Danych*

Gdańsk 2024/2025

Streszczenie

Spis treści

1	Założenia projektowe	5
1.1	Zakres	5
1.2	Charakterystyka projektu oraz użytkownicy	5
1.3	Wymagania funkcjonalne i нефункционалне	5
2	Analiza Swot	7
3	Reguły i mechanizmy zabezpieczeń	8
4	Techniczna architektura systemu	Error! Bookmark not defined.
5	Podsumowanie	Error! Bookmark not defined.

Wprowadzenie

Geneza projektu

Rynek finansowy stanowi jedno z najbardziej dynamicznych i złożonych środowisk, w którym miliony inwestorów na całym świecie podejmują decyzje inwestycyjne każdego dnia. Decyzje te są często oparte na danych historycznych, wskaźnikach technicznych, a także analizie fundamentalnej. Wzrost dostępności danych giełdowych oraz rozwój technologii analizy danych stwarza unikalną możliwość wykorzystania zaawansowanych algorytmów do przewidywania cen akcji.

Wraz z rosnącą popularnością inwestowania indywidualnego i automatycznego, wzrasta zapotrzebowanie na narzędzia, które mogą wspierać decyzje inwestycyjne poprzez dostarczanie dokładnych prognoz opartych na danych historycznych. Nasz projekt ma na celu stworzenie takiego narzędzia, które będzie pomocne zarówno dla inwestorów indywidualnych, jak i profesjonalnych analityków.

Motywacja

Motywacją do realizacji projektu jest potrzeba zrozumienia mechanizmów rządzących rynkiem giełdowym i wspierania użytkowników w podejmowaniu bardziej świadomych decyzji inwestycyjnych. Dzięki wykorzystaniu metod analizy danych, statystyki oraz uczenia maszynowego możliwe jest identyfikowanie wzorców w danych historycznych, które mogą wskazywać potencjalne zmiany cen w przyszłości.

Dodatkowo projekt odpowiada na wyzwania współczesnego rynku, takie jak:

- Przetwarzanie dużych zbiorów danych
- Szybkie i precyzyjne prognozowanie w dynamicznym środowisku.
- Wykorzystanie interdyscyplinarnego podejścia łączącego finanse, matematykę oraz informatykę.

Cele projektu

Stworzenie algorytmu przewidującego ceny akcji:

- Opracowanie modelu, który na podstawie danych historycznych prognozuje przyszłe ceny akcji z określonym poziomem dokładności.
- Integracja wskaźników technicznych oraz statystycznych w procesie analizy.

Zaprojektowanie i implementacja systemu analizy danych:

- Pobieranie danych ze źródeł, takich jak API giełdowe
- Przetwarzanie danych: czyszczenie, normalizacja i analiza eksploracyjna.

Zapewnienie potencjalnej skalowalności i elastyczności systemu w przyszłości:

- Możliwość dostosowania algorytmu do różnych typów aktywów finansowych (np. akcje, ETF, kryptowaluty).
- Rozszerzenie funkcjonalności o dodatkowe dane (np. dane makroekonomiczne, sentyment rynkowy).

Projekt ma na celu nie tylko opracowanie skutecznego narzędzia do prognozowania, ale również stworzenie platformy edukacyjnej, która umożliwi użytkownikom zrozumienie procesu analizy danych i algorytmów, na których opierają się prognozy.

1 Założenia projektowe

1.1 Zakres

Projekt obejmuje stworzenie prototypowego algorytmu do prognozowania cen akcji, który będzie testowany na wybranych danych giełdowych. Ze względu na ograniczenia czasowe i zasobowe projekt nie będzie obejmował wdrożenia komercyjnego ani integracji z rzeczywistymi systemami inwestycyjnymi. Zakres prac obejmuje:

- Analizę i przetwarzanie danych historycznych.
- Implementację prostego modelu uczenia maszynowego.
- Opracowanie podstawowych wizualizacji wyników.

1.2 Charakterystyka projektu oraz użytkownicy

Projekt ma charakter edukacyjny. Priorytetem jest zrozumienie procesu tworzenia algorytmu analitycznego oraz zastosowanie wiedzy teoretycznej w praktyce. Projekt zakłada wykorzystanie darmowych lub otwartoźródłowych narzędzi, takich jak Python, R, oraz dostępnych publicznie API giełdowych.

Użytkownicy

Głównymi użytkownikami projektu są:

- **Studenci realizujący projekt:** Zdobywają doświadczenie w pracy z danymi finansowymi i algorytmami predykcyjnymi.
- **Prowadzący przedmiot:** Ocena projektu pod kątem zastosowania wiedzy i realizacji celów edukacyjnych.

Pośrednio wyniki projektu mogą być inspiracją dla innych studentów i osób zainteresowanych rynkiem finansowym do pogłębiania wiedzy na temat analizy danych i uczenia maszynowego.

1.3 Wymagania funkcjonalne i нефункционалне

Wymagania funkcjonalne

1. System powinien umożliwiać pobieranie danych giełdowych z wybranych
2. Powinien przetwarzać dane giełdowe, w tym ich czyszczenie, normalizację oraz generowanie podstawowych wskaźników technicznych.
3. Algorytm powinien prognozować ceny akcji na podstawie danych historycznych przy użyciu wybranego modelu uczenia maszynowego.
4. System musi generować podstawowe wizualizacje, takie jak wykresy cen akcji oraz wskaźników technicznych.
5. Powinien umożliwiać eksport wyników analizy w formacie CSV lub PDF.

Wymagania нефunkcjonalne

1. System powinien działać na komputerach osobistych bez potrzeby instalacji zaawansowanego sprzętu.
2. Oprogramowanie musi być zgodne z darmowymi bibliotekami i narzędziami open-source.
3. Kod źródłowy powinien być czytelny i dobrze udokumentowany, aby umożliwić jego rozbudowę przez innych użytkowników.
4. Całość powinna działać w czasie rzeczywistym lub w krótkim czasie przetwarzania (do kilku minut na jedno zadanie).
5. Interfejs systemu powinien być intuicyjny, nawet dla osób bez zaawansowanej wiedzy technicznej.

2 Analiza Swot

Mocne strony (Strengths)

- Wykorzystanie otwartoźródłowych narzędzi minimalizuje koszty projektu.
- Projekt edukacyjny pozwala na rozwój umiejętności analitycznych i programistycznych.
- Możliwość rozszerzenia systemu o dodatkowe funkcjonalności.

Słabe strony (Weaknesses)

- Ograniczone zasoby czasowe i ludzkie.
- Brak profesjonalnego wsparcia technicznego.
- Możliwości algorytmu mogą być ograniczone przez prostotę modelu.

Szanse (Opportunities)

- Możliwość wykorzystania projektu jako bazy do przyszłych, bardziej zaawansowanych prac.
- Zdobycie wiedzy i doświadczenia w analizie danych finansowych i uczeniu maszynowym.
- Inspiracja dla innych studentów do podjęcia podobnych inicjatyw.

Zagrożenia (Threats)

- Potencjalne problemy z dostępnością lub ograniczeniami w API giełdowych.
- Trudności z walidacją i weryfikacją wyników prognoz.
- Ograniczona dokumentacja techniczna w przypadku wybranych bibliotek lub narzędzi.

3 Reguły i mechanizmy zabezpieczeń

W projekcie należy uwzględnić podstawowe mechanizmy zabezpieczające dane i system przed nieautoryzowanym dostępem oraz zapewniające integralność danych. Do takich mechanizmów należą:

1. **Hashowanie kluczy API:** Klucze API używane do pobierania danych z zewnętrznych serwisów powinny być przechowywane w zaszyfrowanej formie, np. z wykorzystaniem bibliotek takich jak cryptography w Pythonie.
2. **Ograniczenie dostępu:** Pliki konfiguracyjne zawierające wrażliwe dane (np. klucze API) powinny być dodane do .gitignore w systemach kontroli wersji, aby uniknąć ich przypadkowego udostępnienia.
3. **Walidacja danych wejściowych:** System powinien sprawdzać poprawność i format danych wejściowych, aby zapobiec atakom typu injection.
4. **Bezpieczne logowanie:** W przypadku rozbudowy systemu o funkcjonalności logowania użytkowników, należy wykorzystać sprawdzone biblioteki do obsługi haseł, takie jak bcrypt.
5. **Monitorowanie i logowanie błędów:** Implementacja mechanizmów rejestracji błędów z możliwością ich analizy w celu wykrywania potencjalnych naruszeń bezpieczeństwa.

4 Plan realizacji projektu

Aby projekt był realizowany w sposób uporządkowany, przyjęto następujący podział na etapy, uwzględniając techniczną architekturę systemu:

Przygotowanie środowiska pracy

- Konfiguracja wspólnego repozytorium kodu (np. na GitHubie lub GitLabie) do zarządzania wersjami i współpracy.
- Instalacja potrzebnych narzędzi i bibliotek:
 - Python: pandas, numpy, matplotlib, scikit-learn, requests.
 - R (jeśli używane): tidyverse, caret.
 - Dodatkowe narzędzia do pracy z API (np. yfinance, alpha_vantage).
- Ustalenie podziału ról w zespole (np. jedna osoba odpowiedzialna za pobieranie danych, druga za modele, trzecia za wizualizacje i raporty).

Pobieranie i przetwarzanie danych

- **Implementacja modułu do pobierania danych z API:**
 - Stworzenie funkcji do komunikacji z wybranymi API giełdowymi.
 - Implementacja mechanizmów obsługi błędów (np. limitów żądań, niepoprawnych odpowiedzi).

Przetwarzanie danych:

- Implementacja funkcji do czyszczenia danych (np. usuwanie braków, uzupełnianie wartości).
- Normalizacja danych (np. przekształcanie wartości na skalę 0–1).

Eksploracja danych i wizualizacja

- Przeprowadzenie analizy eksploracyjnej (EDA):
 - Identyfikacja trendów, zmienności i korelacji między zmiennymi.
- Przygotowanie wizualizacji:
 - Wykresy linii (np. ceny akcji w czasie), histogramy, scatter ploty.
 - Wizualizacje wskaźników technicznych (np. średnie ruchome).

Budowa modelu predykcyjnego

- **Wybór modelu uczenia maszynowego:**
 - Proste modele (np. regresja liniowa, drzewa decyzyjne) na początek.
 - W przypadku większego postępu: modele bardziej zaawansowane (np. LSTM do danych czasowych).

Trenowanie modelu:

- Podział danych na zestaw treningowy i testowy.
- Trenowanie modelu i mierzenie jego dokładności za pomocą odpowiednich metryk (np. RMSE, MAE).
- **Testowanie modelu:**
 - Badanie, jak model radzi sobie na danych, których wcześniej nie widział.

Generowanie raportów i wyników

- Implementacja modułu raportowania:
 - Automatyczne generowanie plików CSV lub PDF z wynikami analizy.
- Przygotowanie czytelnych wizualizacji wyników predykcji.

Testowanie systemu

- Przeprowadzenie testów jednostkowych i integracyjnych, aby upewnić się, że poszczególne moduły współpracują poprawnie.
- Identyfikacja i naprawa potencjalnych błędów.

Optymalizacja i zabezpieczenia

- **Optymalizacja wydajności:**
 - Analiza czasu wykonywania kluczowych operacji (np. pobierania danych, trenowania modelu).
 - Zastosowanie narzędzi do profilowania kodu (np. cProfile w Pythonie) w celu identyfikacji wąskich gardeł.
 - Refaktoryzacja kodu, w tym eliminacja niepotrzebnych operacji i poprawa efektywności algorytmów.
- **Wdrożenie mechanizmów zabezpieczeń:**
 - Implementacja hashowania kluczy API oraz przechowywanie ich w zaszyfrowanych plikach konfiguracyjnych.
 - Ustawienie odpowiednich uprawnień dostępu do plików i danych.
 - Dodanie logowania błędów i zdarzeń w systemie, co ułatwi identyfikację potencjalnych problemów.

Dokumentacja i prezentacja wyników

- **Dokumentacja techniczna:**
 - Opis architektury systemu, struktury kodu oraz wykorzystanych bibliotek.
 - Instrukcja użytkownika, w tym sposób uruchamiania systemu i konfiguracji API.
 - Wytyczne dotyczące rozwoju i potencjalnej rozbudowy projektu.
- **Przygotowanie raportu końcowego:**
 - Podsumowanie wyników analizy, osiągnięć i napotkanych wyzwań.
 - Omówienie dokładności modelu i potencjalnych obszarów do ulepszenia.
- **Prezentacja projektu:**
 - Przygotowanie krótkiej prezentacji (np. w PowerPoint) z wizualizacjami wyników.
 - Demonstracja działania systemu na rzeczywistych danych podczas zaliczenia.

5 Opis Architektury

6 Podsumowanie

harmonogram realizacji