



DOKUMENTACJA PROJEKTU

I. Charakterystyka oprogramowania

Nazwa skrócona:	ANALIZA PORÓWNAWCZA
Nazwa pełna:	Analiza porównawcza metod analitycznych dla wybranych danych giełdowych
Krótki opis:	<p>Celem projektu jest analiza dostępnych narzędzi analitycznych przeznaczonych do prognozowania szeregów czasowych. Istnieje wiele metod podejścia do tego zagadnienia, wywodzących się z wielu dziedzin takich jak statystyka, ekonometria czy też sztuczna inteligencja. Projekt koncentruje się na porównaniu wybranych technik analizy danych, przy użyciu darmowych ogólnodostępnych rozwiązań.</p> <p>Ze względu na dużą dostępność różnorodnych bibliotek i frameworków wspierających uczenie maszynowe, sztuczną inteligencję, analizę statystyczną oraz ekonometrię, do realizacji projektu wybrano język programowania Python. Wybrany język programowania oferuje bogaty ekosystem narzędzi, które pozwalają na efektywne przetwarzanie, transformację i modelowanie danych, co czyni go idealnym środowiskiem do pracy nad analizą szeregów czasowych.</p>

II. Prawa autorskie

- Autorzy
 - Mateusz Ortmann
 - Iga Kozłowska

Warunki licencyjne do oprogramowania - Licencja MIT

III. Specyfikacja wymagań

Identyfikator	Opis	Nazwa	Priorytet	Kategoria
R1	Pozyskanie danych	Implementacja rozwiązania które pobierze dane. Rozwiązanie ręczne (maualne wejście na stronę) lub pobranie danych przez API.	Wymagane	Funkcjonalne
R1 - alternatywnie	Utworzenie danych	Utworzenie Benchamrku czyli prostego zbioru danych na potrzeby porównania modeli	Opcjonalne	Funkcjonalne
R2	Załadowanie danych	W przypadku użycia metody manualnego pobrania danych, załadowanie danych do pamięci wykorzystując odpowiednie biblioteki (pandas)	Wymagane	Funkcjonalne
R3	Wizualizacja danych	W celu lepszego zrozumienia problemu jak i danych warto dokonać wizualizacji szeregów	Opcjonalne	Pozafunkcjonalne
R3	Transfórmacja danych	Po załadowaniu danych nalerzy dokonać odpowiednich transformacji przed rozpoczęciem proescu trenowania. Przykładowo (usunięcie szumu: różnicowanie), zamiana na odpowiedni typ danych	Wymagane	Funkcjonalne
R4	Wybór zależności	Wybór odpowiednich bibliotek oraz modeli w celu ich przetestowania	Wymagane	Funkcjonalne
R4	Trenowanie	Napisanie skryptu / funkcji której celem jest wytrenowanie modelu na danych. Ustalenie parametrów trenowania: batch size, random seeds	Wymagane	Funkcjonalne
R5	Testowanie	Porównanie wyników predykcji modeli z danymi rzeczywistymi, obliczanie miar błędów i kluczowych statystyk do oceny jakości modelu.	Wymagane	Funkcjonalne

Identyfikator	Opis	Nazwa	Priorytet	Kategoria
R6	Zmiana parametrów modeli	Po uzyskaniu wyników możliwa będzie zmiana konfiguracji modeli w celu uzyskania lepszych rezultatów. Badania można powtarzać wielokrotnie a wyniki zapisywać i porównywać	Opcjonalne	Funkcjonalne

IV. Architektura oprogramowania

<i>Architektura rozwoju</i>	<ul style="list-style-type: none"> ● Google colab ● Linux 22.04 – system operacyjny ● Python 3.12 – język programowania ● Windows 11 – system operacyjny ● pre-commit 4.0.1 – Zarządzanie hookami ● poetry 1.8.5 – zarządzanie zależnościami ● cuda 12.1 – akceleracja GPU ● ipykernel 6.29.5 – kernel dla jupytera
<i>Architektura uruchomieniowa</i>	<ul style="list-style-type: none"> ● Transformacja danych: <ul style="list-style-type: none"> ○ Numpy 2.2.1 – BSD License ○ Pandas 2.2.3 – BSD License ○ sklearn 1.6.0 – MIT License ○ copy – Python Software Foundation License (BSD-style) ○ math – Python Software Foundation License (BSD-style) ● Wizualizacja danych: <ul style="list-style-type: none"> ○ Matplotlib 3.10.0 – Python Software Foundation License (BSD-style) ○ Seaborn 0.13.2 – BSD License ○ pylab ● Modele ekonometryczne <ul style="list-style-type: none"> ○ Statsmodels 0.14.4 – BSD License ● Modele ML/AI: <ul style="list-style-type: none"> ○ torch 2.5.1+cu121 – BSD License ○ lightning 2.5.0 – Apache Software License ○ pytorch_forecasting 1.2.0 – MIT License ● Dane finansowe <ul style="list-style-type: none"> ○ yfinance 0.2.51 – Apache Software License ● Testy: <ul style="list-style-type: none"> ○ pytest 8.3.4

V. Testy

Testowanie funkcji generującej dane Benchmarku	<p>create_benchmark_data_test.py</p> <p>Plik zawiera funkcję test_create_benchmark ze statementami assert które sprawdzają czy wygenerowane zostały odpowiedniej długości szeregi czasowe.</p> <p>Testy przebiegły pomyślnie</p>
Testowanie funkcji tworzącej sekwencje treningowe	<p>create_sequence_test.py</p> <p>Plik zawiera funkcję test_create_sequences ze statementami assert które sprawdzają czy wygenerowane zostały sekwencje treningowe i testowe o odpowiednim kształcie</p> <p>Testy przebiegły pomyślnie</p>
Testowanie funkcji transformującej dane	<p>data_transformation_test.py</p> <p>Plik zawiera funkcję test_data_transformation ze statementami assert które sprawdzają typy danych wygenerowanych ramek danych.</p> <p>Test przebiegły pomyślnie</p>