

APLIKACJA DO PROGNOZOWANIA CEN ZŁOTA WZGLĘDEM DOLARA

Sławomir Kobyłko

Wojciech Czarnecki

Miłosz Furman



KLAUZULA INFORMACYJNA / DISCLAIMER

PREZENTOWANA APLIKACJA MA CHARAKTER WYŁĄCZNIE
EDUKACYJNY I DEMONSTRACYJNY.

PROGNOZY CEN ZŁOTA GENEROWANE PRZEZ MODEL NIE
STANOWIĄ REKOMENDACJI INWESTYCYJNYCH, FINANSOWYCH ANI
EKONOMICZNYCH.

AUTORZY NIE PONOSZĄ ODPOWIEDZIALNOŚCI ZA DECYZJE
PODJĘTE NA PODSTAWIE WYNIKÓW APLIKACJI.

Stworzenie aplikacji prognozującej ceny złota na podstawie danych historycznych

Zastosowanie modelu SARIMA do analizy szeregu czasowego

Udostępnienie prostego interfejsu użytkownika

Automatyczne pobieranie danych z Yahoo Finance po wybraniu odpowiedniej daty

CEL APLIKACJI

1. **Fragmenty Kodu**
2. **Chat GPT**
3. **Aplikacja do prognozowania**
4. **API – Yahoo Finance**
5. **SARIMA**
6. **Komunikacja aplikacji z API**
7. **Generowanie prognozy**
8. **Interfejs użytkownika**
9. **Technologia użyta w projekcie**
10. **Podsumowanie**

AGENDA



KOD APLIKACJI FRAGMENT

```
5 Model: SARIMA (statsmodels.SARIMAX bez zmiennych zewnętrznych, szybka wersja)
6 """
7
8 import warnings
9 import os
10 import os, certifi, tempfile, shutil
11
12
13 #kod wstawiony przez Sławomira Kobyłko, wymagane jest do działania na jego komputerze (problemy z certyfikatem)
14 # 1) skopiuj cacert.pem do ścieżki bez diakrytyków (temp)
15 _src = certifi.where()
16 _dst = os.path.join(tempfile.gettempdir(), "cacert.pem")
17 try:
18     if not os.path.exists(_dst) or os.path.getsize(_dst) != os.path.getsize(_src):
19         shutil.copyfile(_src, _dst)
20 except Exception:
21     # jeśli coś pójdzie nie tak - trudno, zostawimy oryginał
22     _dst = _src
23
24 # 2) ustaw wszystkie istotne zmienne
25 os.environ["SSL_CERT_FILE"] = _dst
26 os.environ["REQUESTS_CA_BUNDLE"] = _dst
27 os.environ["CURL_CA_BUNDLE"] = _dst # ważne dla curl / curl_cffi
28 os.environ["VF_USE_CURL_CFFI"] = "0" # wyłącz curl_cffi w yfinance (na wszelki wypadek)
29 os.environ["VFINANCE_USE_CURL_CFFI"] = "0" # alias - też ustawiamy
30 warnings.filterwarnings("ignore")
31
32
33 import datetime as dt
34 import numpy as np
35 import pandas as pd
36 import streamlit as st
37 import yfinance as yf
38 from dateutil.relativedelta import relativedelta
39 from statsmodels.tsa.statespace.sarimax import SARIMAX
40 import matplotlib.pyplot as plt
41
42
43 st.set_page_config(page_title="Prognoza ceny złota - Yahoo", page_icon="📈", layout="centered")
44
```

- ▶ Ten kod przygotowuje środowisko do prognozowania ceny złota z wykorzystaniem modelu SARIMA.
- ▶ Na początku rozwiązuje problem certyfikatów SSL (m.in. dla yfinance i requests), kopiując plik cacert.pem i ustawiając odpowiednie zmienne środowiskowe.
- ▶ Następnie importuje biblioteki do analizy danych, modelowania szeregów czasowych, wizualizacji oraz tworzenia aplikacji Streamlit.
- ▶ Na końcu konfiguruje stronę aplikacji Streamlit, która będzie służyć do prezentacji prognozy.

KOD APLIKACJI FRAGMENT

- ▶ Ten fragment kodu odpowiada za wygląd aplikacji Streamlit.
- ▶ Za pomocą wstrzykniętego CSS-a zmienia tło na gradient, kolory nagłówków, przycisków, pól wyboru, metryk i rozwijanych sekcji, nadając aplikacji tło „dark theme”.
- ▶ Ustawia styl wykresów Matplotlib na ciemny oraz definiuje tytuł aplikacji i podpis źródła danych (Yahoo Finance, model SARIMA).

```
45 #kod wstawiony przez Miłosz Furman
46 st.markdown("""
47 <style>
48 html, body, [data-testid="stApp"] {
49     background: radial-gradient(1100px 600px at 10% 0%, #0f172a 0%, #0b1220 45%, #0a0f1a 100%) !important;
50     color: #e5e7eb;
51 }
52 h1, h2, h3 { color: #f8fafc; }
53 .block-container { padding-top: 2rem; }
54 .stButton button {
55     background: linear-gradient(135deg, #334155, #0ea5e9);
56     color: #f8fafc; border: 0; border-radius: 12px; padding: .6rem 1rem; font-weight: 600;
57     box-shadow: 0 6px 18px rgba(14,165,233,.25);
58 }
59 .stButton button:hover { filter: brightness(1.08); transform: translateY(-1px); }
60 .stSelectbox > div, .stDateInput > div {
61     background: rgba(30,41,59,.6); border:1px solid rgba(148,163,184,.25); border-radius:12px;
62 }
63 [data-testid="stMetric"] {
64     background: rgba(2,6,23,.55); border:1px solid rgba(148,163,184,.2); border-radius:16px; padding: 16px;
65 }
66 [data-testid="stExpander"] {
67     background: rgba(2,6,23,.55); border:1px solid rgba(148,163,184,.2); border-radius:14px;
68 }
69 </style>
70 """ , unsafe_allow_html=True)
71
72 plt.style.use("dark_background")
73
74 st.title("📈 Prognoza ceny złota – GoldPredict")
75 st.caption("Źródło: Yahoo Finance. Model: **SARIMA**.")
76
```

KOD APLIKACJI FRAGMENT

- ▶ Ten fragment kodu tworzy interfejs użytkownika do sterowania prognozą w aplikacji Streamlit.
- ▶ Użytkownik wybiera datę prognozy, zakres danych historycznych, źródło ceny złota (kurs rynkowy lub ETF GLD) oraz jednostkę ceny.
- ▶ Na końcu znajduje się przycisk „Prognozuj”, który uruchamia proces pobrania danych i obliczenia prognozy modelem SARIMA.

```
77 #kod wstawiony przez Wojciech Czarnecki (zmodyfikowany potem przez chatGPT)
78 colA, colB = st.columns([2, 1])
79 with colA:
80     target_date = st.date_input("📅 Wybierz datę prognozy", value=dt.date.today() + relativedelta(years=1))
81 with colB:
82     history_start = st.date_input(
83         "🕒 Pobierz historię od",
84         value=(dt.date.today() - relativedelta(years=10)),
85         help="Początek zakresu danych historycznych"
86     )
87
88 col1, col2, col3 = st.columns([2, 1.5, 1])
89 with col1:
90     symbol_map = {
91         "GC=F": "Cena rynkowa złota (USD/oz)",
92         "GLD": "Fundusz inwestycyjny GLD (ETF w złoto)"
93     }
94     choice = st.selectbox("Źródło ceny", list(symbol_map.values()), index=0)
95     symbol = next(k for k, v in symbol_map.items() if v == choice)
96 with col2:
97     unit = st.selectbox(
98         "Jednostka ceny",
99         ["USD za 1 uncję (oz)", "USD za 1 gram"],
100         index=0 if symbol == "GC=F" else 0,
101         disabled=(symbol == "GLD"),
102         help="Dla GLD jednostka zawsze: USD za 1 udział ETF"
103     )
104 with col3:
105     btn = st.button("Prognozuj")
106
```

- ChatGPT znacząco usprawnił pracę nad projektem, przyspieszając proces tworzenia i poprawiania kodu.
- Pomagał w rozwiązywaniu błędów (m.in. związanych z certyfikatami), sugerował rozwiązania w Streamlit oraz wspierał przy implementacji modelu SARIMA.
- Dzięki niemu korzystanie z dokumentacji Pythona stało się prostsze, co pozwoliło skupić się na tym, aby aplikacja działała poprawnie i była logicznie zaprojektowana.



DLACZEGO CZAT GPT ?

Proгноза цены золота — GoldPredict

Źródło: Yahoo Finance. Model: SARIMA.

Wybierz datę prognozy

2028/12/09

Pobierz historię od

2022/12/09

Źródło ceny

Cena rynkowa złota (USD/oz)

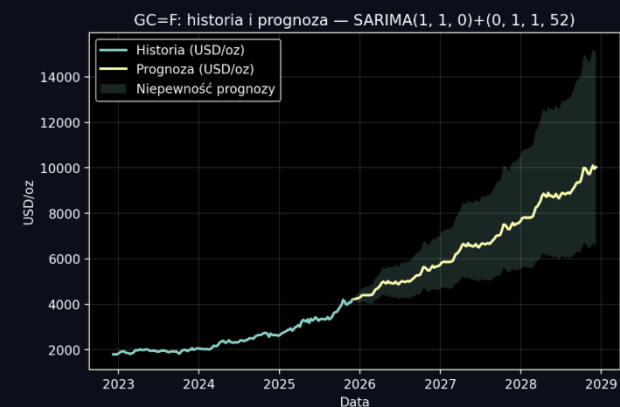
Jednostka ceny

USD za 1 uncję (oz)

Prognozuj

☒ Tryb szybki (zalecany)

☒ Dane tygodniowe (szybciej)



Wynik prognozy

Prognoza na 2028-12-09

10,030.22 USD/oz

> Szczegóły techniczne

APLIKACJA GOLDPREDICT

Użytkownik

Streamlit – interfejs użytkownika

Logika aplikacji - python, pandas/NumPy, statsmodels (SARIMA)

Yahoo Finance – źródło danych historycznych

Prognoza (obliczenia w SARIMA + wynik)

Przepływ danych

ARCHITEKTURA APLIKACJI

Aplikacja służy do **prognozowania przyszłych cen złota**.

Wykorzystuje dane z **Yahoo Finance**.

Model predykcyjny: **SARIMA**.

Umożliwia wybór zakresu danych i parametrów ceny

APLIKACJA DO PROGNOZOWANIA



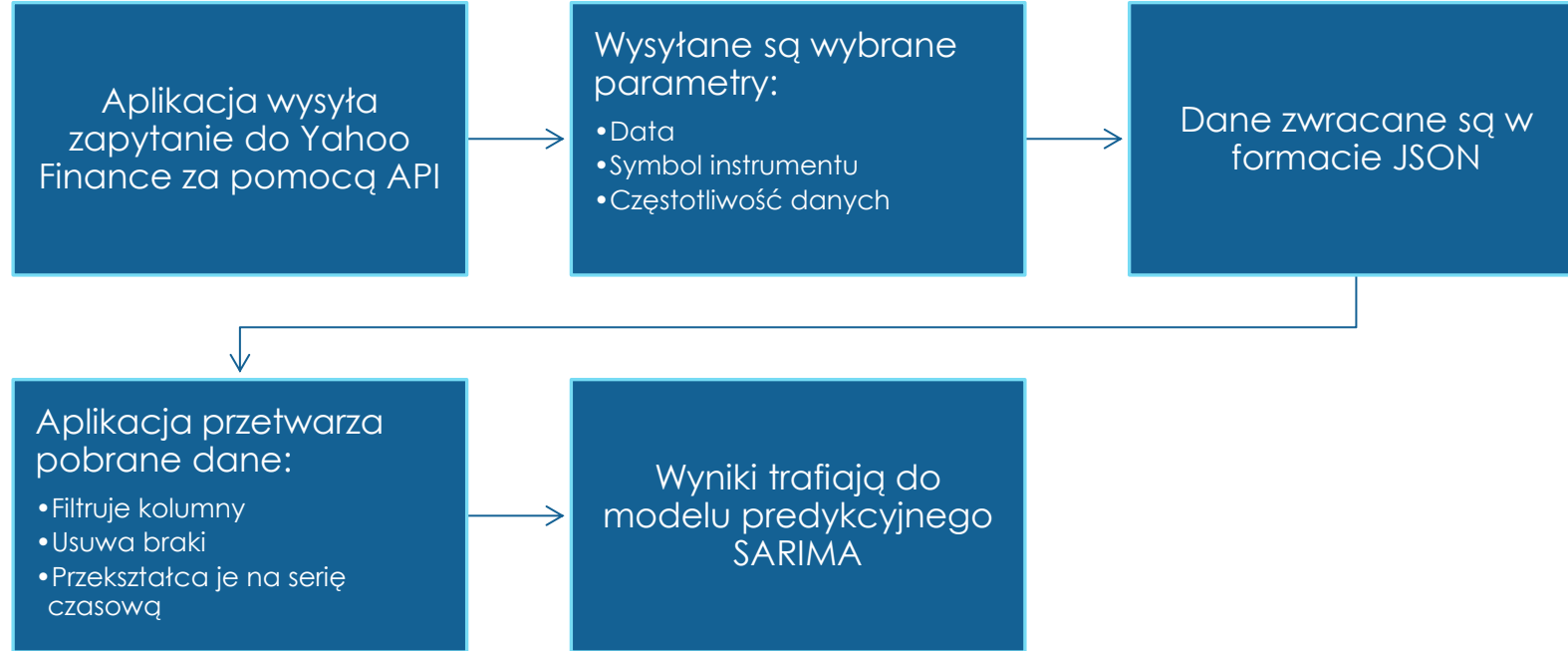
API – YAHOO FINANCE

OGRANICZENIA YAHOO FINANCE

- ▶ Dane są darmowe przez co mogą być niepełne
- ▶ Opóźnienia danych giełdowych, ETF i indeksów (zwykle o 15 minut). Dane historyczne bez opóźnień.
- ▶ Brak gwarancji ciągłości danych (przerwy świąteczne, itp.)
- ▶ Ryzyko niedostępności API
- ▶ Dane są informacyjne (nie są to oficjalne dane giełdowe)

ZALETY KORZYSTANIA Z YAHOO FINANCE

- ▶ Darmowy dostęp do danych
- ▶ Brak limitów zapytań
- ▶ Łatwe pobieranie przez API
- ▶ Wiele instrumentów finansowych (złoto, ETF-y, indeksy, waluty, akcje)
- ▶ Szybka integracja z modelami prognozującymi
- ▶ Dane historyczne nawet do 50 lat wstecz
- ▶ Łatwa implementacja API z pythonem (biblioteka *yfinance*)



KOMUNIKACJA API Z YAHOO FINANCE



SARIMA

Seasonal Auto-Regressive Integrated Moving Average



SARIMA

p – rząd autoregresji

- Ile poprzednich wartości pomaga przewidzieć kolejną

d – liczba zróżnicowań
(usuwanie trendu)

- Usuwa trend aby seria była stabilniejsza

q – rząd średniej ruchomej

- Modeluje wpływ losowych zakłóceń (szumów)

P, D, Q – część sezonowa

- Tak samo jak dla p, d, q , ale dla sezonowości (np. cykl roczny)

s – długość sezonu

- np. 12 miesięcy dla danych miesięcznych, 52 tygodnie dla danych tygodniowych

SARIMA

Parametry modelu

Jest to model prognostyczny
użyty do projektu

Rozszerzona wersja modelu
ARIMA

Dobry do danych finansowych
o charakterze czasowym

Daje prognozy o wysokiej
jakości przy niewielkich
wymaganiach obliczeniowych

SARIMA był najlepszym
wybozem, ponieważ:

Radzi sobie z trendem

Radzi sobie z sezonowością

SARIMA

Umożliwia
modelowanie
trendu i
sezonowości

dane finansowe
często mają trend
i sezonowość

Dobrze radzi sobie z
szumem rynkowym

Pokazuje to
dokładniejsze
prognozy

Stabilna i
sprawdzona
metoda w ekonomii

Pokazuje dobre
przewidywania
prognozy

Nie wymaga dużej
mocy obliczeniowej

stosunkowo szybki
do trenowania

Skuteczna w
krótkoterminowych
prognozach

Najlepiej sprawdza
się przy
przewidywaniu
najbliższych
wartości cen

SARIMA

Parametry modelu

Umożliwia modelowanie trendu i sezonowości
dane finansowe często mają trend + sezonowość

Dobrze radzi sobie z szumem rynkowym
Pokazuje to dokładniejsze prognozy

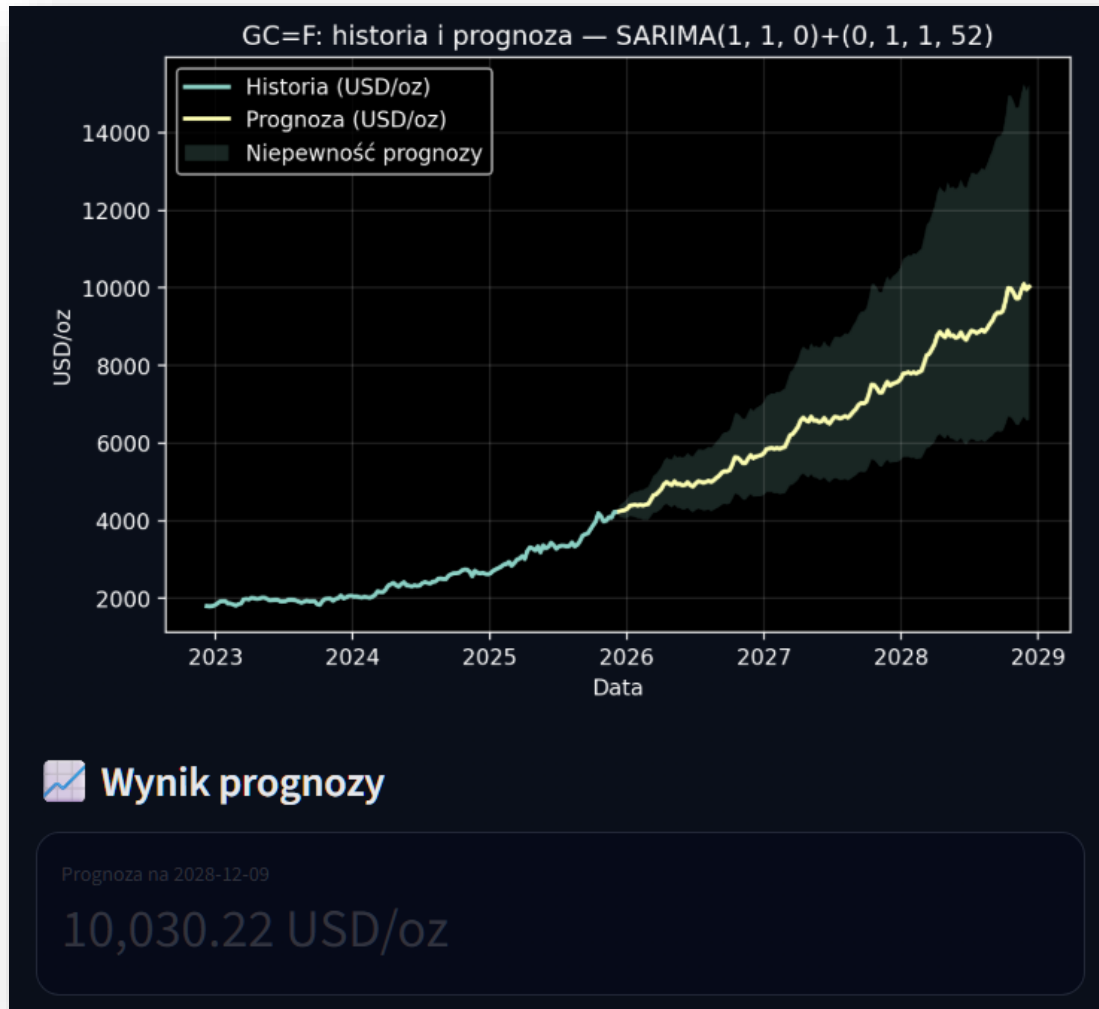
Stabilna i sprawdzona metoda w ekonomii
Pokazuje dobre przewidywania prognozy

Nie wymaga dużej mocy obliczeniowej
stosunkowo szybki do trenowania

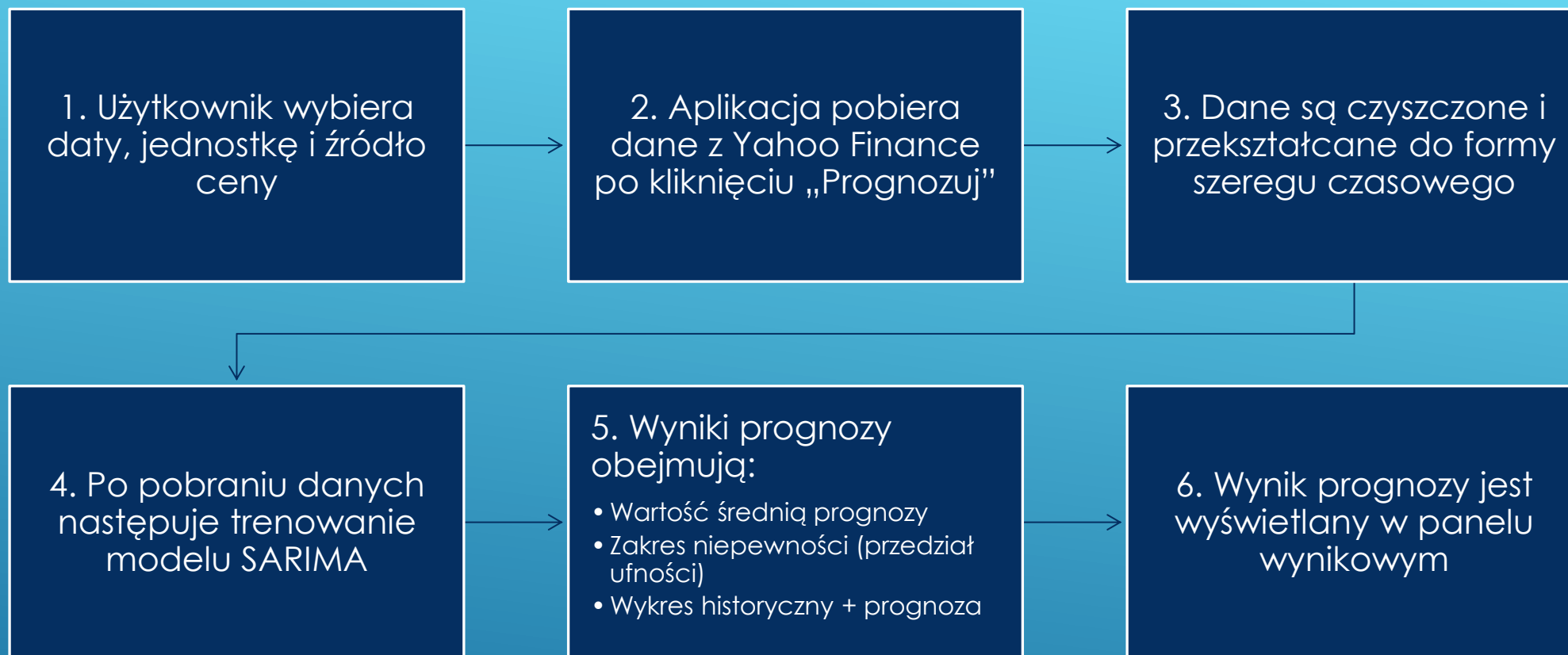
Skuteczna w krótkoterminowych prognozach
Najlepiej sprawdza się przy przewidywaniu najbliższych wartości cen

SARIMA

Dlaczego jest odpowiednia do danych finansowych



GENEROWANIE PROGNOZY



GENEROWANIE PROGNOZY

Niepewność
prognozy


- ▶ Prognoza wygenerowana na podstawie danych historycznych od 09.12.2022
- ▶ Wykres przedstawia prognozę cen złota względem dolara do dnia 9.12.2028 oraz zakres niepewności (półprzezroczyste pole)
- ▶ Jednostka ceny: USD za 1 uncję (oz)
- ▶ Tendencja prognozy wzrostowa: pod wykresem prezentowany jest końcowy wynik prognozy




PRZYKŁADOWY
WYNIK PROGNOZY

Prognoza ceny złota — GoldPredict

Źródło: Yahoo Finance. Model: SARIMA.

 Wybierz datę prognozy

2028/12/09

 Pobierz historię od

2022/12/09


Źródło ceny


Cena rynkowa złota (USD/oz) ▼

Jednostka ceny

USD za 1 uncję (oz) ▼

Prognozuj

☒  Tryb szybki (zalecany) ?

☒  Dane tygodniowe (szybciej) ?

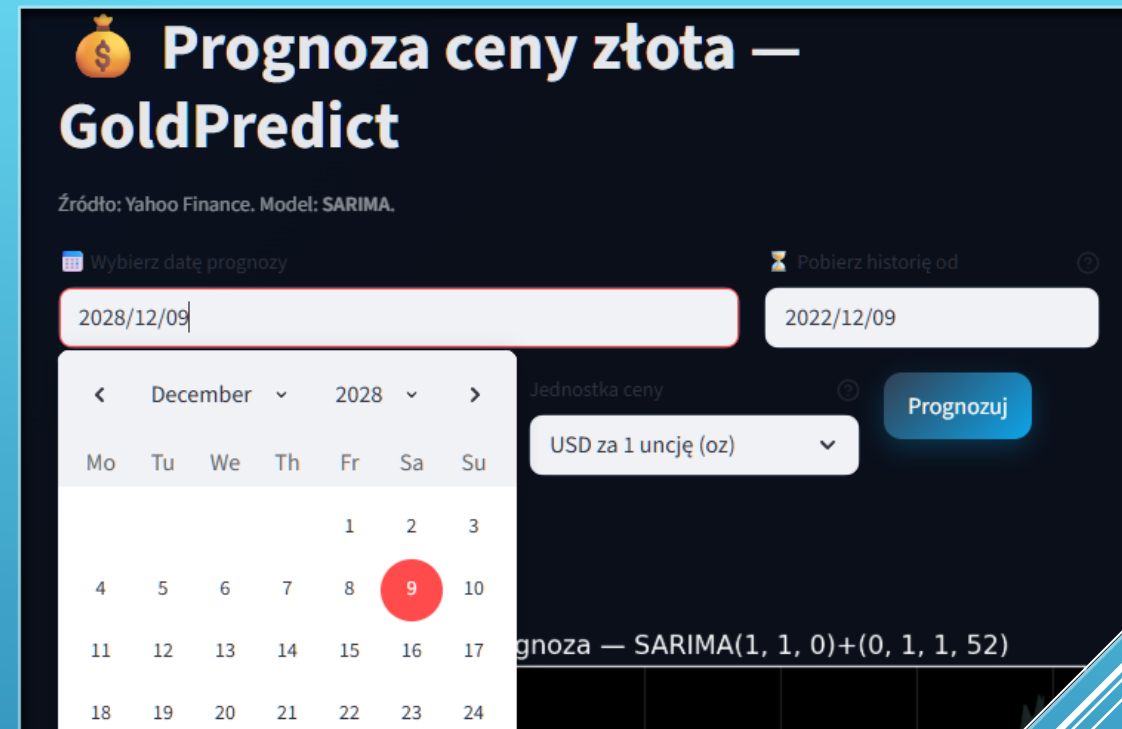
INTERFEJS UŻYTKOWNIKA

Celem interfejsu jest maksymalne uproszczenie procesu generowania prognozy



Wybór jednostki ceny:

USD za 1 uncję
USD za 1 gram



Wybór zakresu dat:
po kliknięciu pojawia się kalendarz

FUNKCJONALNOŚCI INTERFEJSU

 **Prognoza ceny złota — GoldPredict**

Źródło: Yahoo Finance. Model: SARIMA.

Wybierz datę prognozy: 2028/12/09 Pobierz historię od: 2022/12/09

Źródło ceny: Cena rynkowa złota (USD/oz) Jednostka ceny: USD za 1 uncję (oz) **Prognozuj**

☒ ⚡ Tryb szybki (zalecany)

☐ 📅 Dane tygodniowe (szybciej)

Wady: minimalnie mniejsza dokładność

Trenowanie modelu

Tryb szybki (zalecany):

Nie przeszukuje pełnej siatki parametrów
Wybiera tylko kilka najlepszych wariantów
Znacznie skraca czas trenowania modelu

Zalety: Znacznie szybsze działanie

FUNKCJONALNOŚCI INTERFEJSU

 **Prognoza ceny złota — GoldPredict**

Źródło: Yahoo Finance. Model: SARIMA.

Wybierz datę prognozy Pobierz historię od

Źródło ceny Jednostka ceny

☒ Tryb szybki (zalecany) ?

☒ Dane tygodniowe (szybciej)

Wady: mniejsza szczegółowość danych


Trenowanie modelu

Dane tygodniowe (szybciej):

Dane dzienne przekształcane na tygodniowe
Zmniejsza liczbę danych o około 5x → szybsze trenowanie kosztem szczegółowości

Zalety: dużo szybsze trenowanie modelu

FUNKCJONALNOŚCI INTERFEJSU

 **Prognoza ceny złota — GoldPredict**

Źródło: Yahoo Finance, Model: SARIMA.

Wybierz datę prognozy: 2028/12/09

Pobierz historię od: 2022/12/09

Źródło ceny: Cena rynkowa złota (USD/oz) ▼

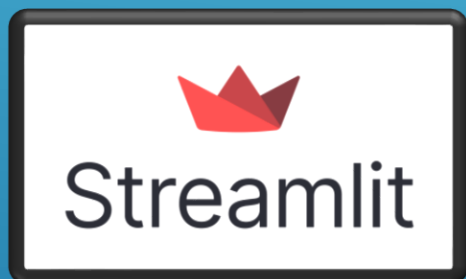
Jednostka ceny: USD za 1 uncję (oz) ▼

Tryb szybki (zalecany) ☒ Dane tygodniowe (szybciej) ☒

Prognozuj

- ▶ Wybór źródła ceny:
- ▶ Cena rynkowa złota (USD)
- ▶ pobiera cenę rynkową złota notowaną na giełdzie w USA
- ▶ Fundusz inwestycyjny ETF w złoto
- ▶ Pobiera cenę ETF

FUNKCJONALNOŚCI INTERFEJSU



**BILBIOTEKI I NARZĘDZIA UŻYTE
W APLIKACJI**

- ▶ Odpowiada za cały backend aplikacji (logikę, przetwarzanie danych)
- ▶ Zarządza obróbką danych przy użyciu NumPy i pandas
- ▶ Łatwa integracja z API (Yahoo Finance)
- ▶ Implementacja modelu prognozującego SARIMA (biblioteka statsmodels)
- ▶ Pozwala tworzyć prosty interfejs użytkownika (Streamlit)
- ▶ Idealny do szybkiego prototypowania aplikacji analitycznych



PYTHON



- ▶ Biblioteka do analizy i prezentowania danych w języku python
- ▶ Łatwa integracja z kodem w pythonie
- ▶ Ułatwia pracę z tabelami oraz szeregami czasowymi (DataFrame)
- ▶ Przygotowuje dane z API (czyszczenie, filtrowanie, konwersja dat)
- ▶ Łatwe tworzenie danych wejściowych dla modeli prognozujących
- ▶ Pozwala łatwo tworzyć dane wejściowe dla modelu SARIMA

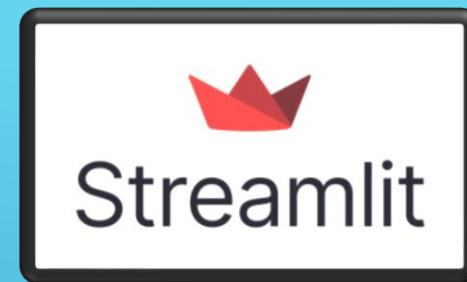
PANDAS

- ▶ Biblioteka do statystyki i ekonometrii w pythonie
- ▶ Zawiera implementację modelu SARIMA użytego w projekcie
- ▶ Umożliwia trenowanie modeli prognozujących na danych czasowych
- ▶ Automatycznie dopasowanie parametrów modelu
- ▶ Integruje się z NumPy i pandas co ułatwia pracę z danymi
- ▶ Posiada narzędzia diagnostyczne do oceny jakości modelu



STATSMODELS

- ▶ Framework do budowy prostych aplikacji webowych w pythonie
- ▶ Umożliwia łatwe tworzenie prostego interfejsu użytkownika
- ▶ Łatwa integracja z pyndas i matplotlib
- ▶ W projekcie odpowiada za widok aplikacji GoldPredict:
 - ▶ Pola wyboru dat i jednostek
 - ▶ Przycisk „Prognozuj”
 - ▶ Panel wyniku



STREAMLIT

- ▶ Podstawowa biblioteka w języku python do obliczeń numerycznych
- ▶ W projekcie przyspiesza obliczenia modelu SARIMA
- ▶ Umożliwia szybkie operacje na dużych zbiorach danych
- ▶ Dobrze współpracuje z Pandas i Statsmodels
- ▶ Przyspiesza działanie aplikacji dzięki optymalizacji obliczeń

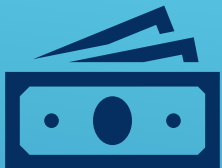


NUMPY



- ▶ Biblioteka do tworzenia wykresów w Pythonie
- ▶ Umożliwia wizualizację danych finansowych pobranych z API
- ▶ Pozwala przedstawić wynik modelu SARIMA w formie czytelnych wykresów
- ▶ Łatwa integracja ze Streamlit
- ▶ Pomaga analizować trendy, sezonowość oraz jakość prognoz

MATPLOTLIB



Co robi aplikacja

Pobiera dane finansowe z Yahoo Finance

Oczyszcza je i przekształca na szeregi czasowe

Generuje prognozę cen złota przy użyciu modelu SARIMA

Prezentuje wynik w czytelnej formie (wykres + wartości liczbowe)



Jakie narzędzia wykorzystuje

Python – logika aplikacji

Pandas / NumPy – przetwarzanie danych

Statsmodels (SARIMA) – modelowanie i prognoza

Matplotlib – wizualizacja

Streamlit – interfejs użytkownika



Główne ograniczenia

Prognoza opiera się na danych historycznych (brak gwarancji trafności)

Dane z Yahoo Finance mogą być opóźnione

Model SARIMA nie uwzględnia czynników makroekonomicznych

Aplikacja wymaga stałego dostępu do Internetu

PODSUMOWANIE

Minimalne:

- ▶ CPU: 2 rdzenie (x64), np. i3 / Ryzen 3
- ▶ RAM: 4 GB
- ▶ Dysk: min. 1 GB wolnego miejsca (Python + biblioteki + cache)
- ▶ System: Windows 10/11 / Linux / macOS
- ▶ Sieć: dostęp do internetu (pobieranie danych z Yahoo Finance)

Zalecane:

- ▶ CPU: 4+ rdzenie, 2.5 GHz+
- ▶ RAM: 8 GB
- ▶ Dysk: 2 GB wolnego miejsca
- ▶ Sieć: stabilne łącze (mniej błędów pobierania)

Uwagi wydajnościowe:

- ▶ Trenowanie SARIMA na dziennych danych 10+ lat może trwać długo (minuty), dlatego w aplikacji jest:
- ▶ Tryb tygodniowy (mniej punktów),
- ▶ Tryb szybki (mało konfiguracji do przetestowania),
- ▶ cache danych (przyspiesza kolejne uruchomienia).

WYMAGANIA SPRZĘTOWE

DZIĘKUJEMY

Sławomir Kobyłko
Wojciech Czarnecki
Miłosz Furman



N52-11
Rok 2025/2026
Inżynieria Oprogramowania