**Cel projektu**

Celem projektu jest opracowanie narzędzia do wspierania wyznaczania trendów na podstawie różnych danych dostępnych w internecie. Umożliwi to inwestorom podejmowanie bardziej trafnych i opartych na danych decyzji inwestycyjnych.

## Spis Treści

**Konfiguracja środowiska** **Wczytanie danych** **Analiza danych** **Przetwarzanie wstępne danych** **Stworzenie modeli** **Backtesting modeli i optymalizacja** **Ocena modeli** **Zastosowanie wyników w rzeczywistości** **Wnioski z uzyskanych danych** **Raport końcowy**

# visualization  
import matplotlib.pyplot as plt  
import seaborn as sns  
from wordcloud import WordCloud  
  
# linear algebra  
import numpy as np   
  
# data processing  
import pandas as pd  
  
# import data  
import yfinance as yf  
  
# data analysis  
from statsmodels.graphics.tsaplots import month\_plot  
from statistics import mean  
  
# text formatting  
from pprint import pprint  
  
# natural language processing  
import nltk  
nltk.download('stopwords')  
from nltk.corpus import stopwords  
from nltk.stem import PorterStemmer  
from transformers import pipeline  
from sklearn.feature\_extraction.text import CountVectorizer  
  
# machine learning

**Dane są wykorzystane i ich pochodzenie.**

Oto dane użyte w projekcie oraz ich pochodzenie:

* **Notowania S&P 500** - pobrane z użyciem biblioteki *yfinance*
* **Nagłówki artykułów z New York Times** - pobrane za pomocą API dostarczanego przez NYT - <https://developer.nytimes.com/>
* **Dane makroekonomiczne gospodarki USA i podstawowe wskaźniki dla S&P 500** - pobrane z:
  + <https://data.nasdaq.com/>
  + <https://fred.stlouisfed.org/>
  + <https://www.bls.gov/>
  + <https://stooq.pl/>
* **Wyszukiwanie frazy "S&P 500" w wyszukiwarce** - pobrano dane z wyszukiwarki google ze strony - <https://trends.google.com/trends/>

Powyższe dane są w dalszych etapach poddane analizie oraz wstępnie przetworzone w celu wykorzystania ich a modelach uczenia maszynowego.

W projekcie skupiono się na danych związanych z indeksem S&P 500, gospodarką Stanów Zjednoczonych, nagłówkami artykułów z NYT, trendami wyszukiwania frazy "S&P 500" w wyszukiwarce google. Na podstawie danych z notowań wyznaczono też wiele wskaźników pomocnych przy analizie technicznej, których także użyto do predykcji notowań. Dane pochodzą z okresu od 01/01/2004 do 08/31/2023.

**Notowania S&P 500**

# get S&P 500 financial data  
try:  
 # fetch from yfinance  
 sp500 = yf.Ticker("^GSPC")  
   
 #print all index info  
 #pprint("Info")  
 #pprint(sp500.info)  
  
 #get stock quotes  
 sp500 = sp500.history( period="max" )  
 start\_date = '2004-01-01'  
 end\_date = '2023-08-31'  
 sp500 = sp500[(sp500.index >= start\_date) & (sp500.index <= end\_date)]  
  
except Exception as e:  
 # catch any other unexpected exceptions  
 print(f"An unexpected error occurred: {e}")  
 # load the dataset  
 sp500 = pd.read\_csv('data/SP500\_Stock\_Quotes.csv', index\_col='date')  
 sp500.index = pd.DatetimeIndex(sp500.index, tz='America/New\_York')  
   
  
# make all column headers and index name lowercase  
sp500.columns = sp500.columns.str.lower()  
sp500.index.name = sp500.index.name.lower()  
  
sp500

open high low close \  
date   
2004-01-02 00:00:00-05:00 1111.920044 1118.849976 1105.079956 1108.479980   
2004-01-05 00:00:00-05:00 1108.479980 1122.219971 1108.479980 1122.219971   
2004-01-06 00:00:00-05:00 1122.219971 1124.459961 1118.439941 1123.670044   
2004-01-07 00:00:00-05:00 1123.670044 1126.329956 1116.449951 1126.329956   
2004-01-08 00:00:00-05:00 1126.329956 1131.920044 1124.910034 1131.920044   
... ... ... ... ...   
2023-08-25 00:00:00-04:00 4389.379883 4418.459961 4356.290039 4405.709961   
2023-08-28 00:00:00-04:00 4426.029785 4439.560059 4414.979980 4433.310059   
2023-08-29 00:00:00-04:00 4432.750000 4500.140137 4431.680176 4497.629883   
2023-08-30 00:00:00-04:00 4500.339844 4521.649902 4493.589844 4514.870117   
2023-08-31 00:00:00-04:00 4517.009766 4532.259766 4507.390137 4507.660156   
  
 volume dividends stock splits   
date   
2004-01-02 00:00:00-05:00 1153200000 0.0 0.0   
2004-01-05 00:00:00-05:00 1578200000 0.0 0.0   
2004-01-06 00:00:00-05:00 1494500000 0.0 0.0   
2004-01-07 00:00:00-05:00 1704900000 0.0 0.0   
2004-01-08 00:00:00-05:00 1868400000 0.0 0.0   
... ... ... ...   
2023-08-25 00:00:00-04:00 3296180000 0.0 0.0   
2023-08-28 00:00:00-04:00 2957230000 0.0 0.0   
2023-08-29 00:00:00-04:00 3354820000 0.0 0.0   
2023-08-30 00:00:00-04:00 3064110000 0.0 0.0   
2023-08-31 00:00:00-04:00 3946360000 0.0 0.0   
  
[4950 rows x 7 columns]

**Nagłówki New York Times**

# load the dataset  
nyt\_hl = pd.read\_csv('data/NYT\_Headlines.csv', index\_col='date', encoding='utf-8')  
nyt\_hl.index = pd.to\_datetime(nyt\_hl.index)  
nyt\_hl.index = nyt\_hl.index.tz\_localize('America/New\_York')  
nyt\_hl

headline  
date   
2004-01-01 00:00:00-05:00 FreeMarkets Agrees to Buy Auction Unit of Covi...  
2004-01-01 00:00:00-05:00 National Briefing | South: Arkansas: Assembly ...  
2004-01-01 00:00:00-05:00 Clark Courts Veterans in Swing Through South  
2004-01-01 00:00:00-05:00 Paid Notice: Deaths PAONE, NICOLA  
2004-01-01 00:00:00-05:00 The Time We Thought We Knew  
... ...  
2023-08-31 00:00:00-04:00 Yale’s President Announces He Will Step Down  
2023-08-31 00:00:00-04:00 7 Spectacular Space Images From August  
2023-08-31 00:00:00-04:00 Trump, Waiving Arraignment, Pleads Not Guilty ...  
2023-08-31 00:00:00-04:00 The End (of Summer) Is Nigh. Let’s Enjoy It!  
2023-08-31 00:00:00-04:00 Starfield’s 1,000 Planets May Be One Giant Lea...  
  
[1763372 rows x 1 columns]

**Dane makroekonomiczne gospodarki USA i podstawowe wskaźniki dla S&P 500**

# load the dataset  
macro\_data = pd.read\_csv('data/USA\_SP500\_Fundamentals\_2004\_2023\_Combined.csv', index\_col='date')  
macro\_data.index = pd.to\_datetime(macro\_data.index)  
# make all column headers lowercase  
macro\_data.columns = macro\_data.columns.str.lower()  
macro\_data.index = macro\_data.index.tz\_localize('America/New\_York')  
macro\_data

cape div\_growth div\_yield earn\_yield pbv \  
date   
2004-01-01 00:00:00-05:00 27.66 8.16 1.61 4.40 3.03   
2004-01-02 00:00:00-05:00 27.66 8.16 1.61 4.40 3.03   
2004-01-03 00:00:00-05:00 27.66 8.16 1.61 4.40 3.03   
2004-01-04 00:00:00-05:00 27.66 8.16 1.61 4.40 3.03   
2004-01-05 00:00:00-05:00 27.66 8.16 1.61 4.40 3.03   
... ... ... ... ... ...   
2023-08-27 00:00:00-04:00 31.36 7.33 1.50 3.85 4.47   
2023-08-28 00:00:00-04:00 31.36 7.33 1.50 3.85 4.47   
2023-08-29 00:00:00-04:00 31.36 7.33 1.50 3.85 4.47   
2023-08-30 00:00:00-04:00 31.36 7.33 1.50 3.85 4.47   
2023-08-31 00:00:00-04:00 31.09 7.33 1.54 3.89 4.31   
  
 pe price\_sales\_ratio 10usy\_bond\_yield \  
date   
2004-01-01 00:00:00-05:00 22.73 1.56 4.26   
2004-01-02 00:00:00-05:00 22.72 1.56 4.37   
2004-01-03 00:00:00-05:00 22.71 1.56 4.37   
2004-01-04 00:00:00-05:00 22.70 1.56 4.37   
2004-01-05 00:00:00-05:00 22.70 1.56 4.39   
... ... ... ...   
2023-08-27 00:00:00-04:00 25.76 2.61 4.23   
2023-08-28 00:00:00-04:00 25.75 2.61 4.21   
2023-08-29 00:00:00-04:00 25.74 2.61 4.12   
2023-08-30 00:00:00-04:00 25.74 2.61 4.11   
2023-08-31 00:00:00-04:00 25.73 2.52 4.11   
  
 cons\_sent cons\_sent\_percent\_change cpiaucsl\_pch \  
date   
2004-01-01 00:00:00-05:00 103.8 12.10 0.43   
2004-01-02 00:00:00-05:00 103.8 12.10 0.43   
2004-01-03 00:00:00-05:00 103.8 12.10 0.43   
2004-01-04 00:00:00-05:00 103.8 12.10 0.43   
2004-01-05 00:00:00-05:00 103.8 12.10 0.43   
... ... ... ...   
2023-08-27 00:00:00-04:00 69.5 -2.93 0.60   
2023-08-28 00:00:00-04:00 69.5 -2.93 0.60   
2023-08-29 00:00:00-04:00 69.5 -2.93 0.60   
2023-08-30 00:00:00-04:00 69.5 -2.93 0.60   
2023-08-31 00:00:00-04:00 69.5 -2.93 0.60   
  
 fed\_effective\_rate fed\_effect\_rate\_pch gdp \  
date   
2004-01-01 00:00:00-05:00 1.00 0.02 11923.45   
2004-01-02 00:00:00-05:00 1.00 0.02 11923.45   
2004-01-03 00:00:00-05:00 1.00 0.02 11923.45   
2004-01-04 00:00:00-05:00 1.00 0.02 11923.45   
2004-01-05 00:00:00-05:00 1.00 0.02 11923.45   
... ... ... ...   
2023-08-27 00:00:00-04:00 5.33 0.21 26798.60   
2023-08-28 00:00:00-04:00 5.33 0.21 26798.60   
2023-08-29 00:00:00-04:00 5.33 0.21 26798.60   
2023-08-30 00:00:00-04:00 5.33 0.21 26798.60   
2023-08-31 00:00:00-04:00 5.33 0.21 26798.60   
  
 gdp\_percent\_change pmi\_index unemploy\_rate   
date   
2004-01-01 00:00:00-05:00 1.28 60.8 5.7   
2004-01-02 00:00:00-05:00 1.28 60.8 5.7   
2004-01-03 00:00:00-05:00 1.28 60.8 5.7   
2004-01-04 00:00:00-05:00 1.28 60.8 5.7   
2004-01-05 00:00:00-05:00 1.28 60.8 5.7   
... ... ... ...   
2023-08-27 00:00:00-04:00 1.01 46.4 3.8   
2023-08-28 00:00:00-04:00 1.01 46.4 3.8   
2023-08-29 00:00:00-04:00 1.01 46.4 3.8   
2023-08-30 00:00:00-04:00 1.01 46.4 3.8   
2023-08-31 00:00:00-04:00 1.01 47.6 3.8   
  
[7183 rows x 17 columns]

# divide data to two data frames  
# data related with S&P 500   
sp500\_rates = macro\_data[['pe', 'cape', 'earn\_yield','pbv', 'price\_sales\_ratio','div\_yield', 'div\_growth']]  
sp500\_rates

pe cape earn\_yield pbv price\_sales\_ratio \  
date   
2004-01-01 00:00:00-05:00 22.73 27.66 4.40 3.03 1.56   
2004-01-02 00:00:00-05:00 22.72 27.66 4.40 3.03 1.56   
2004-01-03 00:00:00-05:00 22.71 27.66 4.40 3.03 1.56   
2004-01-04 00:00:00-05:00 22.70 27.66 4.40 3.03 1.56   
2004-01-05 00:00:00-05:00 22.70 27.66 4.40 3.03 1.56   
... ... ... ... ... ...   
2023-08-27 00:00:00-04:00 25.76 31.36 3.85 4.47 2.61   
2023-08-28 00:00:00-04:00 25.75 31.36 3.85 4.47 2.61   
2023-08-29 00:00:00-04:00 25.74 31.36 3.85 4.47 2.61   
2023-08-30 00:00:00-04:00 25.74 31.36 3.85 4.47 2.61   
2023-08-31 00:00:00-04:00 25.73 31.09 3.89 4.31 2.52   
  
 div\_yield div\_growth   
date   
2004-01-01 00:00:00-05:00 1.61 8.16   
2004-01-02 00:00:00-05:00 1.61 8.16   
2004-01-03 00:00:00-05:00 1.61 8.16   
2004-01-04 00:00:00-05:00 1.61 8.16   
2004-01-05 00:00:00-05:00 1.61 8.16   
... ... ...   
2023-08-27 00:00:00-04:00 1.50 7.33   
2023-08-28 00:00:00-04:00 1.50 7.33   
2023-08-29 00:00:00-04:00 1.50 7.33   
2023-08-30 00:00:00-04:00 1.50 7.33   
2023-08-31 00:00:00-04:00 1.54 7.33   
  
[7183 rows x 7 columns]

# data related with USA economy  
usa\_eco\_rates = macro\_data[['gdp', 'gdp\_percent\_change', 'unemploy\_rate','pmi\_index', 'cpiaucsl\_pch','fed\_effective\_rate', 'fed\_effect\_rate\_pch','10usy\_bond\_yield', 'cons\_sent',  
 'cons\_sent\_percent\_change']]  
usa\_eco\_rates

gdp gdp\_percent\_change unemploy\_rate \  
date   
2004-01-01 00:00:00-05:00 11923.45 1.28 5.7   
2004-01-02 00:00:00-05:00 11923.45 1.28 5.7   
2004-01-03 00:00:00-05:00 11923.45 1.28 5.7   
2004-01-04 00:00:00-05:00 11923.45 1.28 5.7   
2004-01-05 00:00:00-05:00 11923.45 1.28 5.7   
... ... ... ...   
2023-08-27 00:00:00-04:00 26798.60 1.01 3.8   
2023-08-28 00:00:00-04:00 26798.60 1.01 3.8   
2023-08-29 00:00:00-04:00 26798.60 1.01 3.8   
2023-08-30 00:00:00-04:00 26798.60 1.01 3.8   
2023-08-31 00:00:00-04:00 26798.60 1.01 3.8   
  
 pmi\_index cpiaucsl\_pch fed\_effective\_rate \  
date   
2004-01-01 00:00:00-05:00 60.8 0.43 1.00   
2004-01-02 00:00:00-05:00 60.8 0.43 1.00   
2004-01-03 00:00:00-05:00 60.8 0.43 1.00   
2004-01-04 00:00:00-05:00 60.8 0.43 1.00   
2004-01-05 00:00:00-05:00 60.8 0.43 1.00   
... ... ... ...   
2023-08-27 00:00:00-04:00 46.4 0.60 5.33   
2023-08-28 00:00:00-04:00 46.4 0.60 5.33   
2023-08-29 00:00:00-04:00 46.4 0.60 5.33   
2023-08-30 00:00:00-04:00 46.4 0.60 5.33   
2023-08-31 00:00:00-04:00 47.6 0.60 5.33   
  
 fed\_effect\_rate\_pch 10usy\_bond\_yield cons\_sent \  
date   
2004-01-01 00:00:00-05:00 0.02 4.26 103.8   
2004-01-02 00:00:00-05:00 0.02 4.37 103.8   
2004-01-03 00:00:00-05:00 0.02 4.37 103.8   
2004-01-04 00:00:00-05:00 0.02 4.37 103.8   
2004-01-05 00:00:00-05:00 0.02 4.39 103.8   
... ... ... ...   
2023-08-27 00:00:00-04:00 0.21 4.23 69.5   
2023-08-28 00:00:00-04:00 0.21 4.21 69.5   
2023-08-29 00:00:00-04:00 0.21 4.12 69.5   
2023-08-30 00:00:00-04:00 0.21 4.11 69.5   
2023-08-31 00:00:00-04:00 0.21 4.11 69.5   
  
 cons\_sent\_percent\_change   
date   
2004-01-01 00:00:00-05:00 12.10   
2004-01-02 00:00:00-05:00 12.10   
2004-01-03 00:00:00-05:00 12.10   
2004-01-04 00:00:00-05:00 12.10   
2004-01-05 00:00:00-05:00 12.10   
... ...   
2023-08-27 00:00:00-04:00 -2.93   
2023-08-28 00:00:00-04:00 -2.93   
2023-08-29 00:00:00-04:00 -2.93   
2023-08-30 00:00:00-04:00 -2.93   
2023-08-31 00:00:00-04:00 -2.93   
  
[7183 rows x 10 columns]

**Trendy wyszukiwania "S&P 500" w wyszukiwarce google**

# load the dataset  
goog\_trend = pd.read\_csv('data/SP500\_GoogleTrends.csv', index\_col='date')  
goog\_trend.index = pd.to\_datetime(goog\_trend.index)  
goog\_trend.index = goog\_trend.index.tz\_localize('America/New\_York')  
goog\_trend

month\_trend  
date   
2004-01-01 00:00:00-05:00 10  
2004-02-01 00:00:00-05:00 9  
2004-03-01 00:00:00-05:00 8  
2004-04-01 00:00:00-05:00 9  
2004-05-01 00:00:00-04:00 7  
... ...  
2023-04-01 00:00:00-04:00 35  
2023-05-01 00:00:00-04:00 35  
2023-06-01 00:00:00-04:00 38  
2023-07-01 00:00:00-04:00 34  
2023-08-01 00:00:00-04:00 40  
  
[236 rows x 1 columns]

**Dane z szeregów czasowych.**

W danym notatniku mamy do czynienia z danymi z szeregów czasowych. Dane z szeregów czasowych (ang. "time series data") to rodzaj danych, w którym informacje są zbierane, zapisywane lub pomiarowe w określonych momentach w czasie, lub w regularnych odstępach czasu. Te dane reprezentują zmienne, które ewoluują w czasie i pozwalają na analizę, modelowanie i prognozowanie przyszłych wartości na podstawie wcześniejszych obserwacji.

Dane z szeregów czasowych mają pewne charakterystyczne cechy, takie jak sezonowość, trendy, cykliczność i autokorelacja, które można wykorzystać do analizy i modelowania. Analiza danych szeregów czasowych może pomóc w identyfikacji wzorców, prognozowaniu przyszłych notowań, zarządzaniu ryzykiem i podejmowaniu właściwych decyzji opartych na danych.

**sp500**

Indeks **S&P 500**, pełna nazwa "Standard & Poor's 500 Index", jest jednym z najważniejszych indeksów giełdowych na świecie. Jest on używany do pomiaru ogólnej wydajności amerykańskiego rynku akcji. Obejmuje on 500 największych i najbardziej znaczących spółek notowanych na giełdzie amerykańskiej, reprezentujących różne sektory gospodarki.

S&P 500 jest szeroko stosowany jako wskaźnik, który pomaga inwestorom i analitykom śledzić ogólną kondycję rynku akcji w Stanach Zjednoczonych. W skład indeksu wchodzą firmy takie jak Apple, Microsoft, Amazon, Google (Alphabet), czy Berkshire Hathaway.

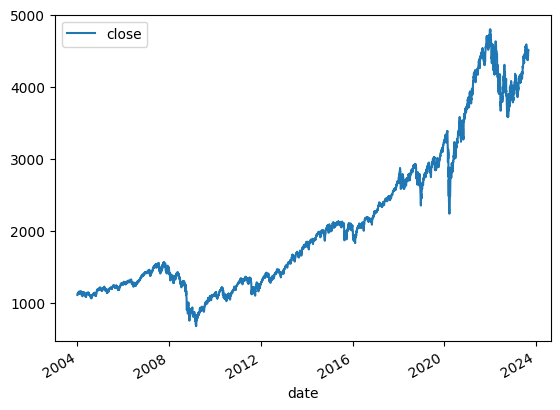
Indeks S&P 500 jest ważnym punktem odniesienia dla wielu inwestorów i profesjonalistów branży finansowej i jest używany do oceny trendów giełdowych, porównywania wyników portfeli inwestycyjnych oraz analizy ogólnego klimatu na rynku.

# check for missing values in each column  
missing\_columns = sp500.isnull().sum()  
pprint("Missing values in column")  
pprint(missing\_columns)  
# print all sp500 data info  
print()  
pprint(sp500.info())

'Missing values in column'  
open 0  
high 0  
low 0  
close 0  
volume 0  
dividends 0  
stock splits 0  
dtype: int64  
  
<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 4950 entries, 2004-01-02 00:00:00-05:00 to 2023-08-31 00:00:00-04:00  
Data columns (total 7 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 open 4950 non-null float64  
 1 high 4950 non-null float64  
 2 low 4950 non-null float64  
 3 close 4950 non-null float64  
 4 volume 4950 non-null int64   
 5 dividends 4950 non-null float64  
 6 stock splits 4950 non-null float64  
dtypes: float64(6), int64(1)  
memory usage: 309.4 KB  
None

# plot prices  
sp500.plot.line(y="close", use\_index = True)

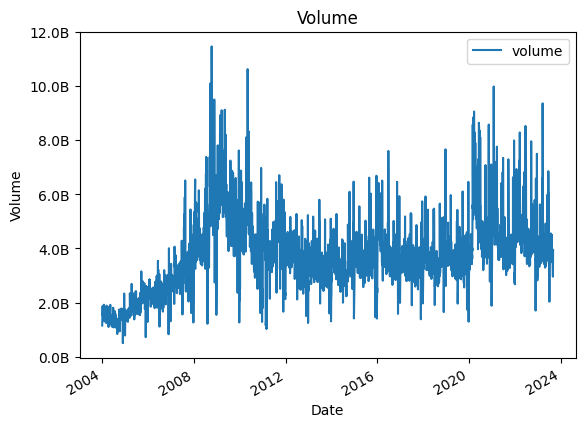
<Axes: xlabel='date'>



Nie wydaje się, aby dane wykazały jakąkolwiek zauważalną sezonowość w poszczególnych latach. Zamiast tego pokazuje trend wzrostowy, a nie zachowanie cykliczne. Można zauważyć też kilka większych spadków. W 2008 r. spadek spowodowany przez światowy kryzys. Na początku 2020 zauważalny spadek cen był związany z pandemią Covid-19. Ostatni widoczny spadek był widoczny w 2022 r.. Ogólnie widać bardzo duży wzrost w wartości indeksu w przeciągu niecałych 20 lat.`

# plot volume  
volume\_plot = sp500.plot.line(y="volume", use\_index = True)  
volume\_plot.yaxis.set\_major\_formatter(plt.FuncFormatter(lambda x, \_: '%1.1fB' % (x \* 1e-9)))  
volume\_plot.set\_xlabel("Date")  
volume\_plot.set\_ylabel("Volume")  
volume\_plot.set\_title('Volume')

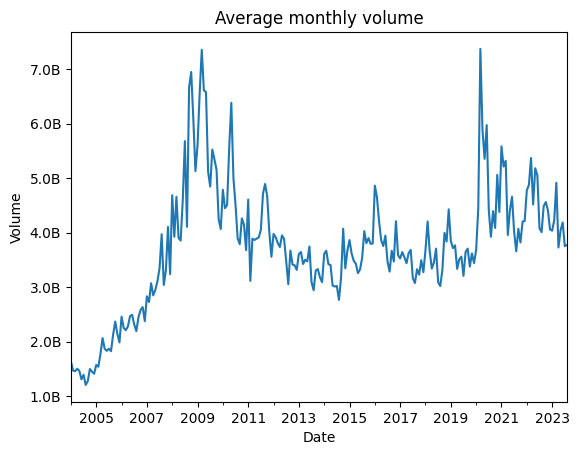
Text(0.5, 1.0, 'Volume')



Wykres wykazuje znaczny szum wynikający z zastosowania skali dziennej. Wybierając średni miesięczny wolumen obrotu, a nie dzienny wolumen obrotu, oczekuje się, że wynikowa wizualizacja będzie bardziej zrozumiała i atrakcyjna wizualnie.

# plot of average monthly volume  
sp500\_volume = sp500['volume'].copy()  
sp500\_volume = sp500\_volume.resample('1M').mean()  
volume\_plot\_avg\_m= sp500\_volume.plot.line(y="volume", use\_index = True)  
volume\_plot\_avg\_m.set\_xlabel("Date")  
volume\_plot\_avg\_m.yaxis.set\_major\_formatter(plt.FuncFormatter(lambda x, \_: '%1.1fB' % (x \* 1e-9)))  
volume\_plot\_avg\_m.set\_ylabel("Volume")  
volume\_plot\_avg\_m.set\_title('Average monthly volume')

Text(0.5, 1.0, 'Average monthly volume')



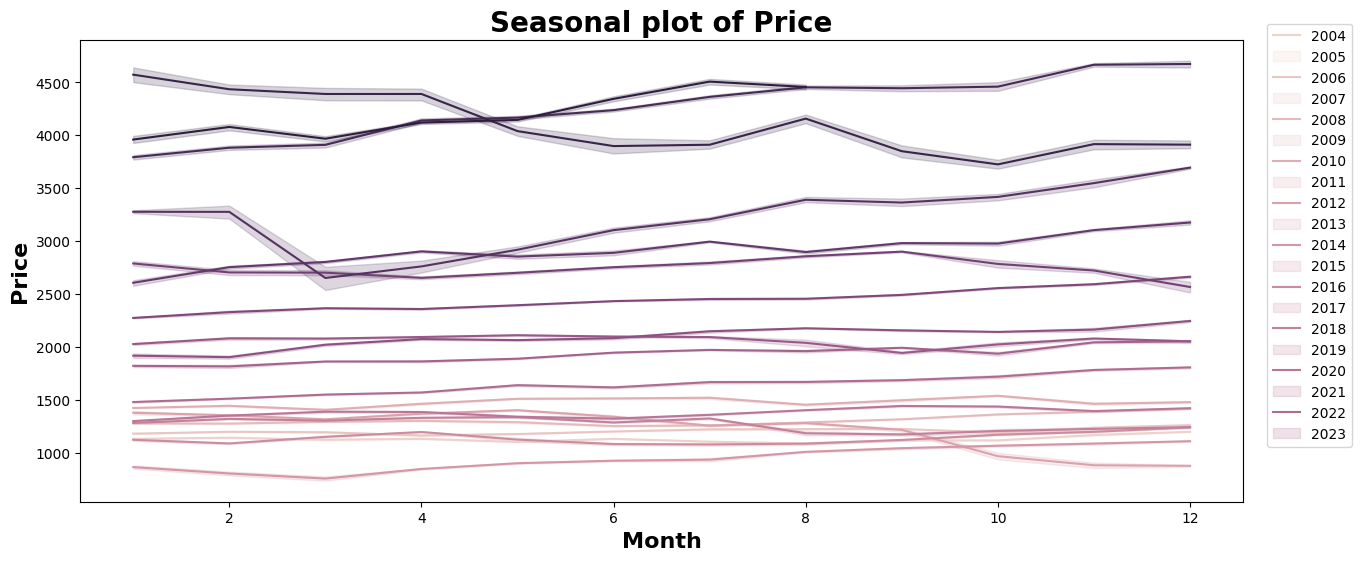
Wykres jest dużo bardziej czytelny dzięki zwiększeniu interwału pomiędzy poszczególnymi danymi do miesiąca. Widać na tym wykresie początkowy trend wzrostowy, a później ustabilizowanie ilości wolumenu w obrocie. Są dwa widoczne okresy większych obrotów w okolicy kryzysu z 2008 r. oraz początku pandemii Covid-19.

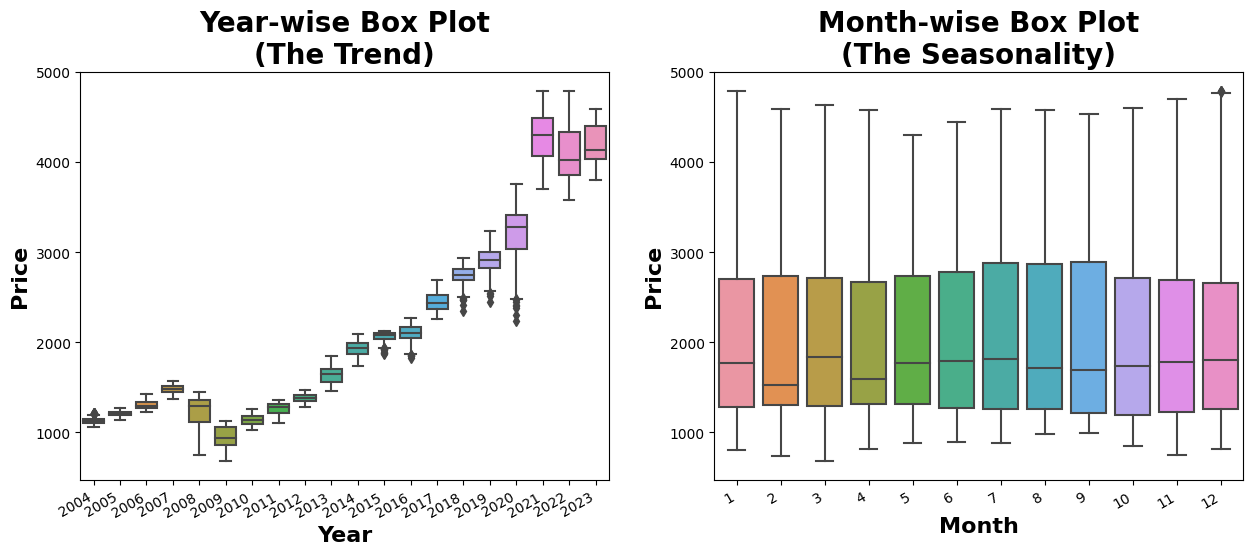
# seasonality  
sp500\_copy = sp500.copy()  
sp500\_copy['year'] = sp500\_copy.index.year  
sp500\_copy['month'] = sp500\_copy.index.month  
print(sp500\_copy['year'].unique())  
sp500\_copy

[2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017  
 2018 2019 2020 2021 2022 2023]

open high low close \  
date   
2004-01-02 00:00:00-05:00 1111.920044 1118.849976 1105.079956 1108.479980   
2004-01-05 00:00:00-05:00 1108.479980 1122.219971 1108.479980 1122.219971   
2004-01-06 00:00:00-05:00 1122.219971 1124.459961 1118.439941 1123.670044   
2004-01-07 00:00:00-05:00 1123.670044 1126.329956 1116.449951 1126.329956   
2004-01-08 00:00:00-05:00 1126.329956 1131.920044 1124.910034 1131.920044   
... ... ... ... ...   
2023-08-25 00:00:00-04:00 4389.379883 4418.459961 4356.290039 4405.709961   
2023-08-28 00:00:00-04:00 4426.029785 4439.560059 4414.979980 4433.310059   
2023-08-29 00:00:00-04:00 4432.750000 4500.140137 4431.680176 4497.629883   
2023-08-30 00:00:00-04:00 4500.339844 4521.649902 4493.589844 4514.870117   
2023-08-31 00:00:00-04:00 4517.009766 4532.259766 4507.390137 4507.660156   
  
 volume dividends stock splits year month   
date   
2004-01-02 00:00:00-05:00 1153200000 0.0 0.0 2004 1   
2004-01-05 00:00:00-05:00 1578200000 0.0 0.0 2004 1   
2004-01-06 00:00:00-05:00 1494500000 0.0 0.0 2004 1   
2004-01-07 00:00:00-05:00 1704900000 0.0 0.0 2004 1   
2004-01-08 00:00:00-05:00 1868400000 0.0 0.0 2004 1   
... ... ... ... ... ...   
2023-08-25 00:00:00-04:00 3296180000 0.0 0.0 2023 8   
2023-08-28 00:00:00-04:00 2957230000 0.0 0.0 2023 8   
2023-08-29 00:00:00-04:00 3354820000 0.0 0.0 2023 8   
2023-08-30 00:00:00-04:00 3064110000 0.0 0.0 2023 8   
2023-08-31 00:00:00-04:00 3946360000 0.0 0.0 2023 8   
  
[4950 rows x 9 columns]

# seasonality of price column  
column = "close"  
# create a figure with one plot  
fig\_seasonal, ax = plt.subplots(figsize=(15, 6))  
  
# plot the seasonal line plot  
sns.lineplot(x=sp500\_copy['month'], y=sp500\_copy[column], hue = sp500\_copy['year'])  
  
# format the plot  
ax.set\_title('Seasonal plot of Price', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax.set\_xlabel('Month', fontsize = 16, fontdict=dict(weight='bold'))  
ax.set\_ylabel('Price', fontsize = 16, fontdict=dict(weight='bold'))  
ax.legend(labels = [str(2004+i) for i in range(20)], bbox\_to\_anchor=(1.1, 1.05))  
  
# create a figure with two plots  
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(15, 6))  
  
# plot the year-wise box plot  
sns.boxplot(x=sp500\_copy['year'], y=sp500\_copy[column], ax=ax[0])  
  
# format the year-wise box plot  
ax[0].set\_title('Year-wise Box Plot\n(The Trend)', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax[0].set\_xlabel('Year', fontsize = 16, fontdict=dict(weight='bold'))  
ax[0].set\_ylabel('Price', fontsize = 16, fontdict=dict(weight='bold'))  
  
# plot the month-wise box plot  
sns.boxplot(x=sp500\_copy['month'], y=sp500\_copy[column], ax=ax[1])  
  
# format the month-wise box plot  
ax[1].set\_title('Month-wise Box Plot\n(The Seasonality)', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax[1].set\_xlabel('Month', fontsize = 16, fontdict=dict(weight='bold'))  
ax[1].set\_ylabel('Price', fontsize = 16, fontdict=dict(weight='bold'))  
  
# rotate the x-axis labels for better readability  
fig.autofmt\_xdate()





**Seasonal plot of Price**

Dla poszczególnych lat nie widać wyraźniej sezonowości. Chociaż można zauważyć większą zmienność notowań dla ostatnich lat.

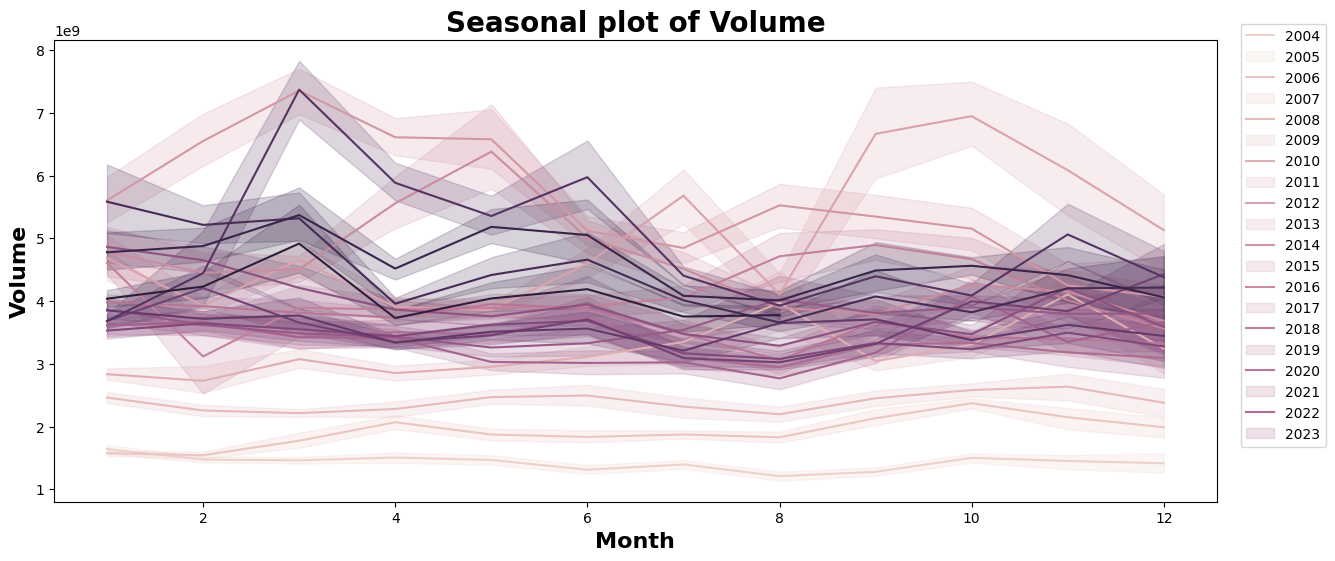
**Year wise Box Plot**

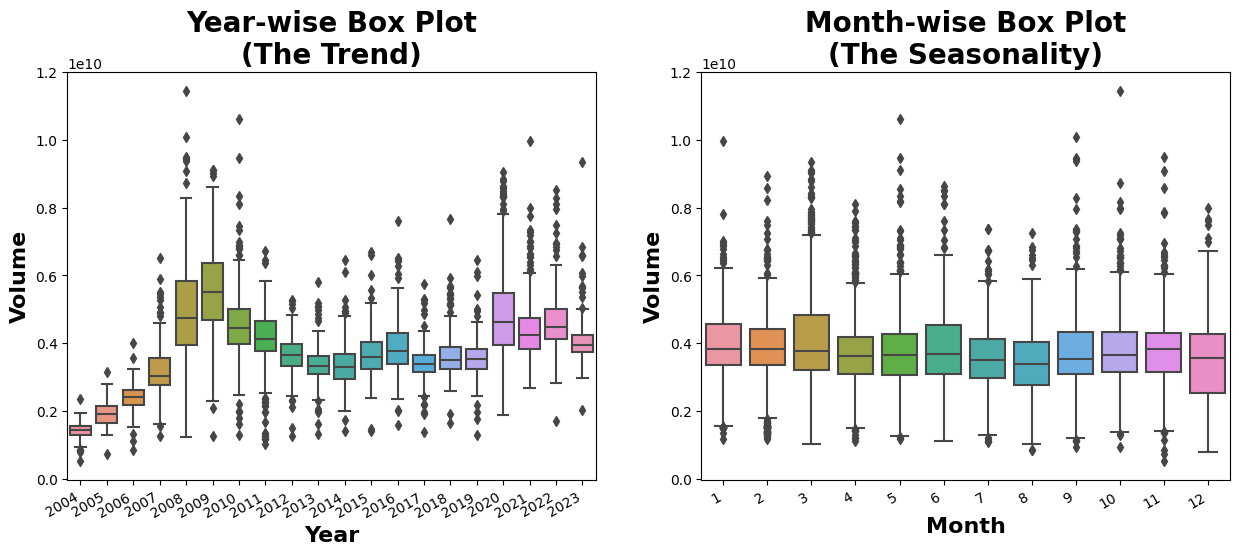
Na tym wykresie widać ogólny trend wzrostowy. Nie jest widoczna sezonowość. Natomiast można zauważyć znaczne wartości odstające dla 2008 r. oraz ostatnich lat. To sugeruje większą zmienność w tych okresach. Dla reszty notowań wartości są dosyć mocno ograniczone w wąskim zakresie.

**Month wise Box Plot**

Nie są widoczne dużej rozbieżności. Można jednak zauważyć, że mediana dla lutego oraz kwietnia jest troszkę niższa od pozostałych miesięcy.

# seasonality of volume column  
column = "volume"  
# create a figure with one plot  
\_, ax = plt.subplots(figsize=(15, 6))  
  
# plot the seasonal line plot  
sns.lineplot(x=sp500\_copy['month'], y=sp500\_copy[column], hue = sp500\_copy['year'])  
  
# format the plot  
ax.set\_title('Seasonal plot of Volume', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax.set\_xlabel('Month', fontsize = 16, fontdict=dict(weight='bold'))  
ax.set\_ylabel('Volume', fontsize = 16, fontdict=dict(weight='bold'))  
ax.legend(labels = [str(2004+i) for i in range(20)], bbox\_to\_anchor=(1.1, 1.05))  
  
# create a figure with two plots  
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(15, 6))  
  
# plot the year-wise box plot  
sns.boxplot(x=sp500\_copy['year'], y=sp500\_copy[column], ax=ax[0])  
  
# format the year-wise box plot  
ax[0].set\_title('Year-wise Box Plot\n(The Trend)', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax[0].set\_xlabel('Year', fontsize = 16, fontdict=dict(weight='bold'))  
ax[0].set\_ylabel('Volume', fontsize = 16, fontdict=dict(weight='bold'))  
  
# plot the month-wise box plot  
sns.boxplot(x=sp500\_copy['month'], y=sp500\_copy[column], ax=ax[1])  
  
# format the month-wise box plot  
ax[1].set\_title('Month-wise Box Plot\n(The Seasonality)', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax[1].set\_xlabel('Month', fontsize = 16, fontdict=dict(weight='bold'))  
ax[1].set\_ylabel('Volume', fontsize = 16, fontdict=dict(weight='bold'))  
  
# rotate the x-axis labels for better readability  
fig.autofmt\_xdate()





**Seasonal plot of Volume**

Dla poszczególnych lat nie widać wyraźniej sezonowości. Można zauważyć mniejszą ilość wolumenu dla miesięcy letnich, czyli lipca i sierpnia.

**Year wise Box Plot**

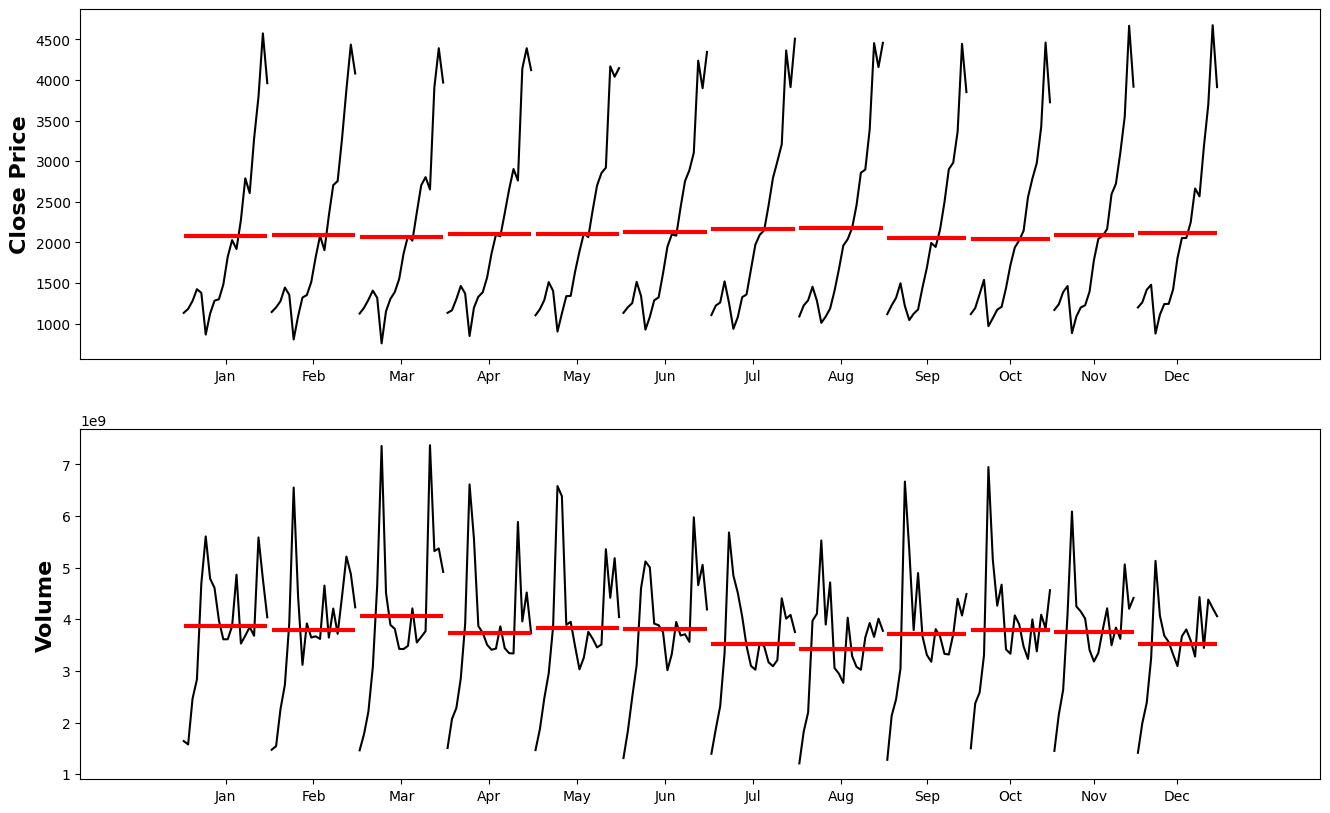
Na tym wykresie jest widoczny początkowy z największą medianą dla roku 2009. Następnie ilość wolumenu w obiegu spada i stabilizuje w ostatnich latach. Największa zmienność jest dla lat związanych z większymi spadkami notowań. To zapewne wynika z większej dynamiki cen oraz nerwowości inwestorów. Znaczne wartości odstające pojawiają się od roku 2007. Wskazują większą zmienność wolumenu, która stale się utrzymuje.

**Month wise Box Plot**

Nie są widoczne dużej rozbieżności. Nie można stwierdzić istniejącej sezonowości

# aggregating sp500 stock quotes to a monthly scaled index  
sp500\_month\_agg = sp500.resample('1M').mean()  
  
  
# # Create the seasonal subseries plots  
fig, ax = plt.subplots(nrows=2, ncols=1, figsize=(16, 10))  
  
month\_plot(sp500\_month\_agg['close'], ax=ax[0])  
ax[0].set\_ylabel('Close Price', fontsize = 16, fontdict=dict(weight='bold'))  
  
month\_plot(sp500\_month\_agg['volume'], ax=ax[1])  
ax[1].set\_ylabel('Volume', fontsize = 16, fontdict=dict(weight='bold'))

Text(0, 0.5, 'Volume')



Wykresy te agregują wartości ze wszystkich lat dla poszczególnych miesięcy. Dzięki temu można potencjalnie zauważyć wyraźniejszy obraz potencjalnej miesięcznej sezonowości.

Staje się oczywiste, że nie ma sezonowości dla notowań. Średnia miesięcznych cen otwarcia i zamknięcia S&P 500 pozostaje stosunkowo stała.

Dla wolumenu jest widoczna większa zmienność, ale nie jest duża. Średnia miesięcznych ilości obrotów na indeksie S&P 500 różni się nieznacznie dla poszczególnych miesięcy. Widoczne są mniejsze wartości dla miesięcy letnich.

**nyt\_hl**

New York Times jest jednym z najbardziej wpływowych i rozpoznawalnych dzienników w Stanach Zjednoczonych i na całym świecie. Jest to amerykańska gazeta z siedzibą w Nowym Jorku, znana ze wszechstronnego relacjonowania wiadomości krajowych i międzynarodowych, polityki, biznesu, kultury i wielu innych tematów. Dlatego analiza nagłówków może dawać wskazówki o sytuacji geopolitycznej i ekonomicznej na świecie, co może uda się z powodzeniem wykorzystać do predykcji notowań.

# check for missing values in each column  
missing\_columns = nyt\_hl.isnull().sum()  
pprint("Missing values in column")  
pprint(missing\_columns)  
print()  
# print all problematic rows  
null\_mask = nyt\_hl.isnull().any(axis=1)  
null\_rows = nyt\_hl[null\_mask]  
pprint("Null rows")  
pprint(null\_rows)  
print()  
# delete null rows  
nyt\_hl = nyt\_hl.dropna()

'Missing values in column'  
headline 562  
dtype: int64  
  
'Null rows'  
 headline  
date   
2004-01-18 00:00:00-05:00 NaN  
2004-02-08 00:00:00-05:00 NaN  
2004-02-10 00:00:00-05:00 NaN  
2004-02-22 00:00:00-05:00 NaN  
2004-02-23 00:00:00-05:00 NaN  
... ...  
2022-01-20 00:00:00-05:00 NaN  
2022-01-20 00:00:00-05:00 NaN  
2022-10-14 00:00:00-04:00 NaN  
2022-10-28 00:00:00-04:00 NaN  
2022-10-28 00:00:00-04:00 NaN  
  
[562 rows x 1 columns]

# group all articles for dates to lists  
nyt\_hl= nyt\_hl.groupby(nyt\_hl.index).agg({'headline': list})  
nyt\_hl

headline  
date   
2004-01-01 00:00:00-05:00 [FreeMarkets Agrees to Buy Auction Unit of Cov...  
2004-01-02 00:00:00-05:00 [Homeless in America, Yes, Today, Tony Cragg, ...  
2004-01-03 00:00:00-05:00 [AIMING TO MAKE FASTER CHIPS, AGERE ACQUIRES T...  
2004-01-04 00:00:00-05:00 [Wanted: Young Delegates, Corrections, How to ...  
2004-01-05 00:00:00-05:00 [Cowboys Preparing To Take Next Step, Paid Not...  
... ...  
2023-08-27 00:00:00-04:00 [Flashback: Your Weekly History Quiz, August 2...  
2023-08-28 00:00:00-04:00 [Man Accused of Injecting Substance Into Neigh...  
2023-08-29 00:00:00-04:00 [President of Powerful Realtors’ Group Resigns...  
2023-08-30 00:00:00-04:00 [Last Defendant in Trump Election Interference...  
2023-08-31 00:00:00-04:00 [Man Gets 2 Years in Prison for Spending Pande...  
  
[7182 rows x 1 columns]

# print all nyt\_hl data info  
pprint(nyt\_hl.info())

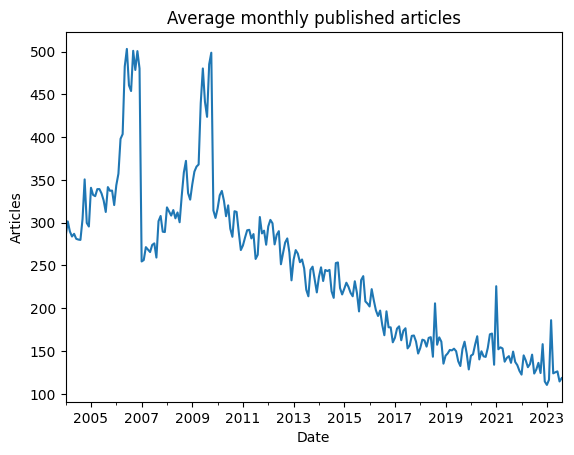
<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 7182 entries, 2004-01-01 00:00:00-05:00 to 2023-08-31 00:00:00-04:00  
Data columns (total 1 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 headline 7182 non-null object  
dtypes: object(1)  
memory usage: 112.2+ KB  
None

# sum articles for all dates  
nyt\_hl['art\_amount'] = nyt\_hl['headline'].str.len()  
nyt\_hl

headline \  
date   
2004-01-01 00:00:00-05:00 [FreeMarkets Agrees to Buy Auction Unit of Cov...   
2004-01-02 00:00:00-05:00 [Homeless in America, Yes, Today, Tony Cragg, ...   
2004-01-03 00:00:00-05:00 [AIMING TO MAKE FASTER CHIPS, AGERE ACQUIRES T...   
2004-01-04 00:00:00-05:00 [Wanted: Young Delegates, Corrections, How to ...   
2004-01-05 00:00:00-05:00 [Cowboys Preparing To Take Next Step, Paid Not...   
... ...   
2023-08-27 00:00:00-04:00 [Flashback: Your Weekly History Quiz, August 2...   
2023-08-28 00:00:00-04:00 [Man Accused of Injecting Substance Into Neigh...   
2023-08-29 00:00:00-04:00 [President of Powerful Realtors’ Group Resigns...   
2023-08-30 00:00:00-04:00 [Last Defendant in Trump Election Interference...   
2023-08-31 00:00:00-04:00 [Man Gets 2 Years in Prison for Spending Pande...   
  
 art\_amount   
date   
2004-01-01 00:00:00-05:00 224   
2004-01-02 00:00:00-05:00 70   
2004-01-03 00:00:00-05:00 158   
2004-01-04 00:00:00-05:00 516   
2004-01-05 00:00:00-05:00 182   
... ...   
2023-08-27 00:00:00-04:00 63   
2023-08-28 00:00:00-04:00 120   
2023-08-29 00:00:00-04:00 135   
2023-08-30 00:00:00-04:00 129   
2023-08-31 00:00:00-04:00 92   
  
[7182 rows x 2 columns]

# plot of average monthly amount of published articles   
nyt\_hl\_sum = nyt\_hl['art\_amount'].copy()  
nyt\_hl\_sum = nyt\_hl\_sum.resample('1M').mean()  
art\_avg\_m= nyt\_hl\_sum.plot.line( use\_index = True)  
art\_avg\_m.set\_xlabel("Date")  
art\_avg\_m.set\_ylabel("Articles")  
art\_avg\_m.set\_title('Average monthly published articles')

Text(0.5, 1.0, 'Average monthly published articles')



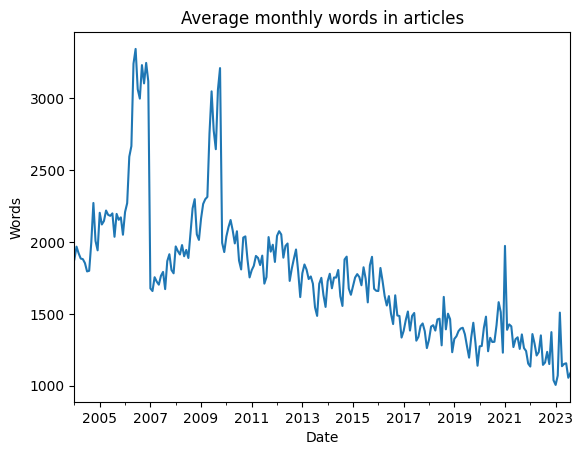
Z powyższego wykresu widać niestety duży trend spadkowy w ilości publikowanych artykułów przez NYT. Są też dwa okresy zwiększonej ilości publikacji. Niestety nie jest to dobrą wiadomością dla wyznaczenia predykcji inwestowania. Nierównomierne rozłożenie danych w czasie, czyli ich niezbalansowanie to istotny problem. Ma on wpływ na skuteczność modeli uczenia maszynowego i może prowadzić do niepożądanych efektów.

# count words for all dates  
def count\_words(text\_list):  
 amount = 0  
 for text in text\_list:  
 amount+= len(text.split())  
 return amount  
  
nyt\_hl['word\_count'] = nyt\_hl['headline'].apply(count\_words)  
nyt\_hl

headline \  
date   
2004-01-01 00:00:00-05:00 [FreeMarkets Agrees to Buy Auction Unit of Cov...   
2004-01-02 00:00:00-05:00 [Homeless in America, Yes, Today, Tony Cragg, ...   
2004-01-03 00:00:00-05:00 [AIMING TO MAKE FASTER CHIPS, AGERE ACQUIRES T...   
2004-01-04 00:00:00-05:00 [Wanted: Young Delegates, Corrections, How to ...   
2004-01-05 00:00:00-05:00 [Cowboys Preparing To Take Next Step, Paid Not...   
... ...   
2023-08-27 00:00:00-04:00 [Flashback: Your Weekly History Quiz, August 2...   
2023-08-28 00:00:00-04:00 [Man Accused of Injecting Substance Into Neigh...   
2023-08-29 00:00:00-04:00 [President of Powerful Realtors’ Group Resigns...   
2023-08-30 00:00:00-04:00 [Last Defendant in Trump Election Interference...   
2023-08-31 00:00:00-04:00 [Man Gets 2 Years in Prison for Spending Pande...   
  
 art\_amount word\_count   
date   
2004-01-01 00:00:00-05:00 224 1529   
2004-01-02 00:00:00-05:00 70 433   
2004-01-03 00:00:00-05:00 158 1000   
2004-01-04 00:00:00-05:00 516 3001   
2004-01-05 00:00:00-05:00 182 1244   
... ... ...   
2023-08-27 00:00:00-04:00 63 548   
2023-08-28 00:00:00-04:00 120 1142   
2023-08-29 00:00:00-04:00 135 1271   
2023-08-30 00:00:00-04:00 129 1194   
2023-08-31 00:00:00-04:00 92 808   
  
[7182 rows x 3 columns]

# plot of average monthly amount of published articles   
nyt\_hl\_sum = nyt\_hl['word\_count'].copy()  
nyt\_hl\_sum = nyt\_hl\_sum.resample('1M').mean()  
art\_avg\_m= nyt\_hl\_sum.plot.line( use\_index = True)  
art\_avg\_m.set\_xlabel("Date")  
art\_avg\_m.set\_ylabel("Words")  
art\_avg\_m.set\_title('Average monthly words in articles')

Text(0.5, 1.0, 'Average monthly words in articles')



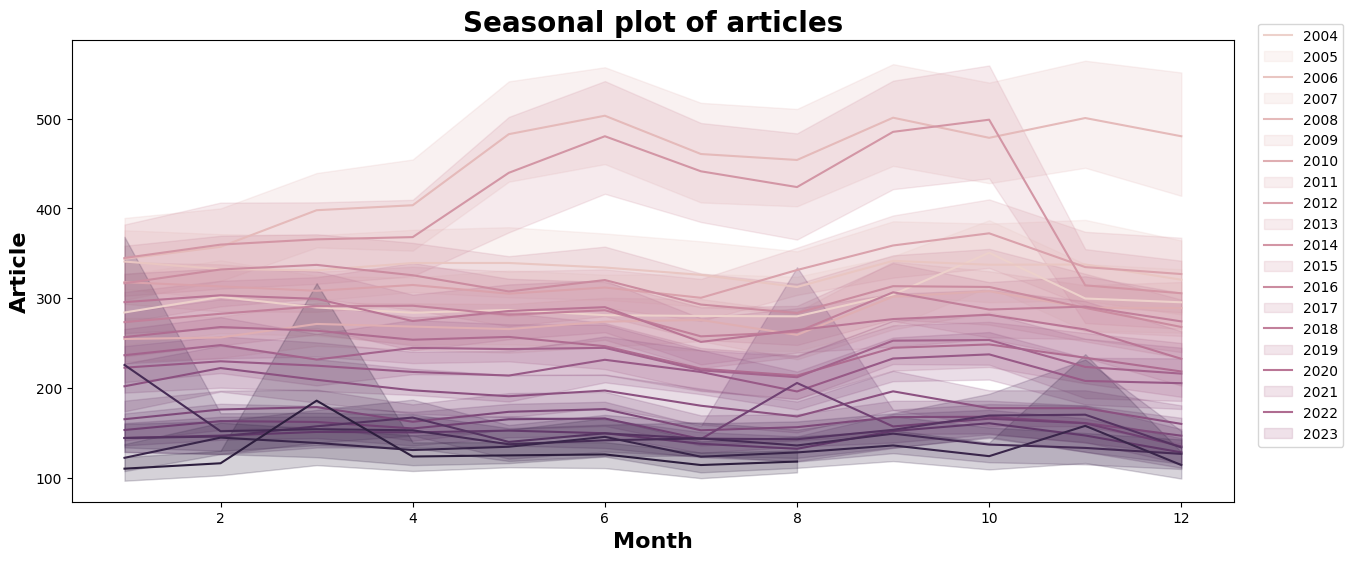
Powyższy wykres pokrywa się z poprzednim wykresem. Widać dużo większą ilość słów dla lat z pierwszych lat zebranych danych.

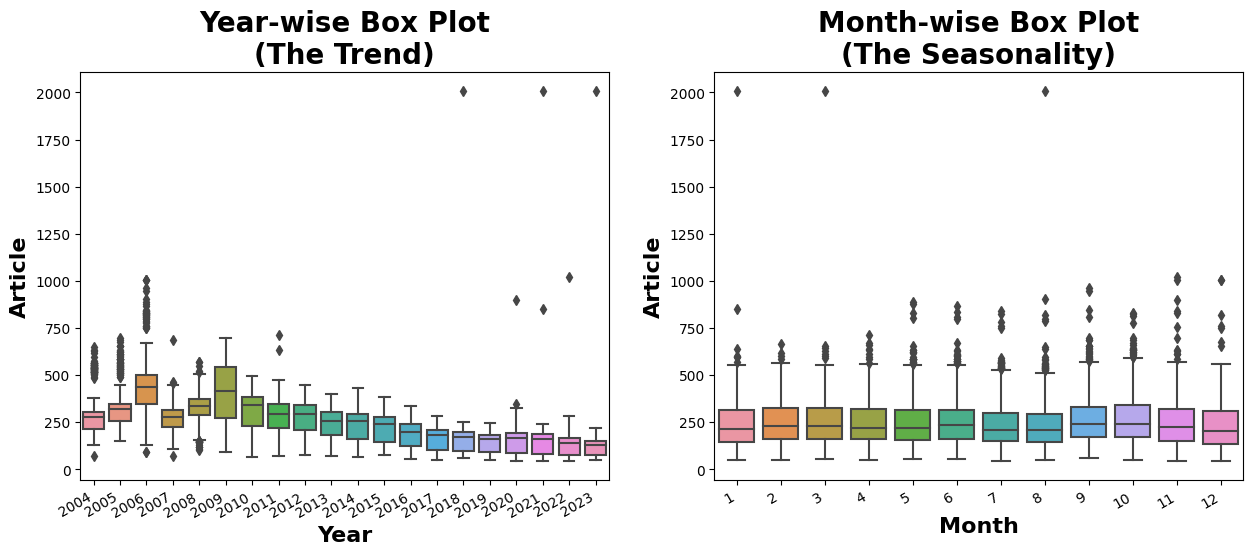
# seasonality  
nyt\_hl\_copy = nyt\_hl.copy()  
nyt\_hl\_copy['year'] = nyt\_hl\_copy.index.year  
nyt\_hl\_copy['month'] = nyt\_hl\_copy.index.month  
print(nyt\_hl\_copy['year'].unique())  
nyt\_hl\_copy

[2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017  
 2018 2019 2020 2021 2022 2023]

headline \  
date   
2004-01-01 00:00:00-05:00 [FreeMarkets Agrees to Buy Auction Unit of Cov...   
2004-01-02 00:00:00-05:00 [Homeless in America, Yes, Today, Tony Cragg, ...   
2004-01-03 00:00:00-05:00 [AIMING TO MAKE FASTER CHIPS, AGERE ACQUIRES T...   
2004-01-04 00:00:00-05:00 [Wanted: Young Delegates, Corrections, How to ...   
2004-01-05 00:00:00-05:00 [Cowboys Preparing To Take Next Step, Paid Not...   
... ...   
2023-08-27 00:00:00-04:00 [Flashback: Your Weekly History Quiz, August 2...   
2023-08-28 00:00:00-04:00 [Man Accused of Injecting Substance Into Neigh...   
2023-08-29 00:00:00-04:00 [President of Powerful Realtors’ Group Resigns...   
2023-08-30 00:00:00-04:00 [Last Defendant in Trump Election Interference...   
2023-08-31 00:00:00-04:00 [Man Gets 2 Years in Prison for Spending Pande...   
  
 art\_amount word\_count year month   
date   
2004-01-01 00:00:00-05:00 224 1529 2004 1   
2004-01-02 00:00:00-05:00 70 433 2004 1   
2004-01-03 00:00:00-05:00 158 1000 2004 1   
2004-01-04 00:00:00-05:00 516 3001 2004 1   
2004-01-05 00:00:00-05:00 182 1244 2004 1   
... ... ... ... ...   
2023-08-27 00:00:00-04:00 63 548 2023 8   
2023-08-28 00:00:00-04:00 120 1142 2023 8   
2023-08-29 00:00:00-04:00 135 1271 2023 8   
2023-08-30 00:00:00-04:00 129 1194 2023 8   
2023-08-31 00:00:00-04:00 92 808 2023 8   
  
[7182 rows x 5 columns]

# seasonality of art\_amount column  
column = "art\_amount"  
# create a figure with one plot  
\_, ax = plt.subplots(figsize=(15, 6))  
  
# plot the seasonal line plot  
sns.lineplot(x=nyt\_hl\_copy['month'], y=nyt\_hl\_copy[column], hue = nyt\_hl\_copy['year'])  
  
# format the plot  
ax.set\_title('Seasonal plot of articles', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax.set\_xlabel('Month', fontsize = 16, fontdict=dict(weight='bold'))  
ax.set\_ylabel('Article', fontsize = 16, fontdict=dict(weight='bold'))  
ax.legend(labels = [str(2004+i) for i in range(20)], bbox\_to\_anchor=(1.1, 1.05))  
  
# create a figure with two plots  
\_, ax = plt.subplots(nrows=1, ncols=2, figsize=(15, 6))  
  
# plot the year-wise box plot  
sns.boxplot(x=nyt\_hl\_copy['year'], y=nyt\_hl\_copy[column], ax=ax[0])  
  
# format the year-wise box plot  
ax[0].set\_title('Year-wise Box Plot\n(The Trend)', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax[0].set\_xlabel('Year', fontsize = 16, fontdict=dict(weight='bold'))  
ax[0].set\_ylabel('Article', fontsize = 16, fontdict=dict(weight='bold'))  
  
# plot the month-wise box plot  
sns.boxplot(x=nyt\_hl\_copy['month'], y=nyt\_hl\_copy[column], ax=ax[1])  
  
# format the month-wise box plot  
ax[1].set\_title('Month-wise Box Plot\n(The Seasonality)', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax[1].set\_xlabel('Month', fontsize = 16, fontdict=dict(weight='bold'))  
ax[1].set\_ylabel('Article', fontsize = 16, fontdict=dict(weight='bold'))  
  
# rotate the x-axis labels for better readability  
fig.autofmt\_xdate()





**Seasonal plot of Articles**

Dla poszczególnych lat nie widać wyraźniej sezonowości. Można zauważyć inne wartości dla poszczególnych lat oraz większą zmienność dla lat z dużą ilością artykułów

**Year wise Box Plot**

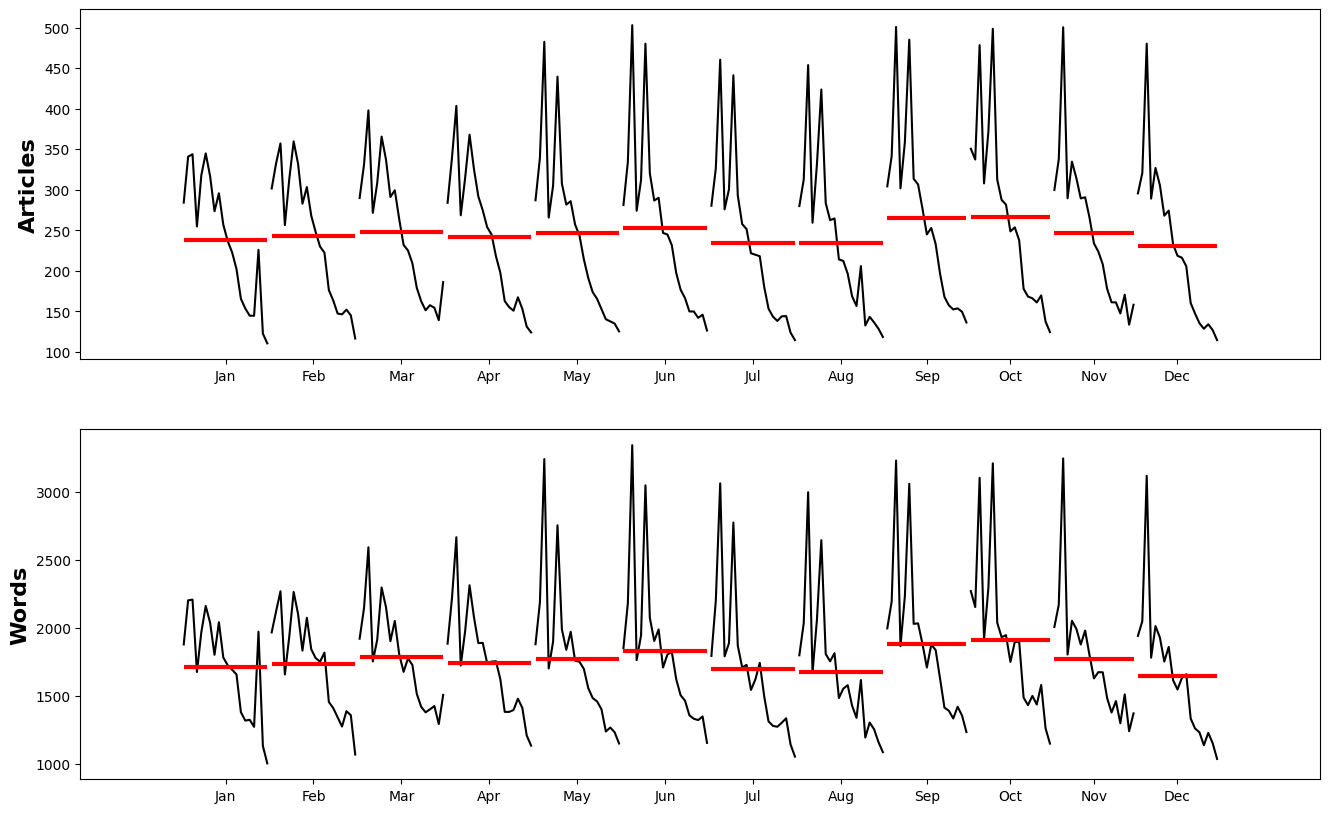
Na tym wykresie jest widoczny początkowy wzrost z największą medianą dla lat 2006 i 20009. Następnie ilość artykułów. Największa zmienność jest dla lat z największą ilością notowań. Znaczne wartości odstające pojawiają się różnych okresach. W ostatnich latach widać dni, w których nadzwyczajnie było dużo publikacji, co jest zastanawiające, jak rzetelne są te dane pobrane z API NYT.

**Month wise Box Plot**

Nie są widoczne dużej rozbieżności. Nie można stwierdzić istniejącej sezonowości

# aggregating nyt\_hl to a monthly scaled index  
nyt\_hl\_copy = nyt\_hl.copy()  
del nyt\_hl\_copy['headline']  
  
nyt\_hl\_month\_agg = nyt\_hl\_copy.resample('1M').mean()  
  
fig, ax = plt.subplots(nrows=2, ncols=1, figsize=(16, 10))  
  
month\_plot(nyt\_hl\_month\_agg['art\_amount'], ax=ax[0])  
ax[0].set\_ylabel('Articles', fontsize = 16, fontdict=dict(weight='bold'))  
  
month\_plot(nyt\_hl\_month\_agg['word\_count'], ax=ax[1])  
ax[1].set\_ylabel('Words', fontsize = 16, fontdict=dict(weight='bold'))

Text(0, 0.5, 'Words')



Nie widać dużej zmienności dla poszczególnych miesięcy dla tych danych. Powoduje to, że nie ma sezonowości w ilości publikacji.

**macro\_data - sp500\_rates**

Wskaźniki dla akcji, indeksów to liczby lub metryki, które pomagają inwestorom, analitykom finansowym i traderom ocenić wydajność i zdrowie rynku giełdowego, konkretnych akcji lub instrumentów finansowych. Istnieje wiele różnych wskaźników giełdowych, z których każdy koncentruje się na innych aspektach rynku.

# check for missing values in each column  
missing\_columns = sp500\_rates.isnull().sum()  
pprint("Missing values in column")  
pprint(missing\_columns)  
  
# print all sp500\_rates data info  
print()  
pprint(sp500\_rates.info())

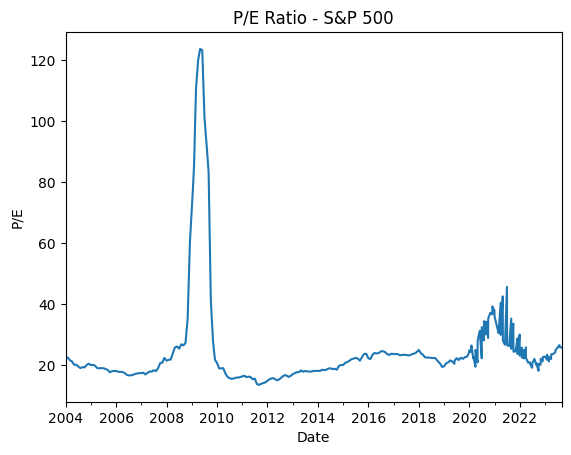
'Missing values in column'  
pe 0  
cape 0  
earn\_yield 0  
pbv 0  
price\_sales\_ratio 0  
div\_yield 0  
div\_growth 0  
dtype: int64  
  
<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 7183 entries, 2004-01-01 00:00:00-05:00 to 2023-08-31 00:00:00-04:00  
Data columns (total 7 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 pe 7183 non-null float64  
 1 cape 7183 non-null float64  
 2 earn\_yield 7183 non-null float64  
 3 pbv 7183 non-null float64  
 4 price\_sales\_ratio 7183 non-null float64  
 5 div\_yield 7183 non-null float64  
 6 div\_growth 7183 non-null float64  
dtypes: float64(7)  
memory usage: 448.9 KB  
None

**P/E Ratio** - (kolumna - 'pe')

P/E (Price-to-Earnings ratio) to stosunek ceny akcji do zysku na akcje. Pomaga ocenić, czy akcje są przecenione (niski P/E) lub nadwycenione (wysoki P/E).

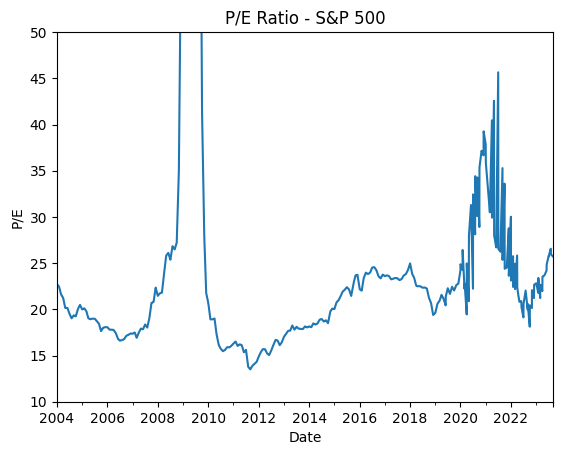
# plot P/E  
sp500\_pe\_plot = sp500\_rates['pe'].plot.line( use\_index = True)  
sp500\_pe\_plot.set\_ylabel("P/E")  
sp500\_pe\_plot.set\_xlabel("Date")  
sp500\_pe\_plot.set\_title('P/E Ratio - S&P 500')

Text(0.5, 1.0, 'P/E Ratio - S&P 500')



# # plot P/E, set\_ylim  
sp500\_pe\_plot = sp500\_rates['pe'].plot.line( use\_index = True)  
sp500\_pe\_plot.set\_ylabel("P/E")  
sp500\_pe\_plot.set\_xlabel("Date")  
sp500\_pe\_plot.set\_title('P/E Ratio - S&P 500')  
sp500\_pe\_plot.set\_ylim(10, 50)

(10.0, 50.0)



Od razu rzuca się w oczy na wykresie Cena/Zysk duży pik w 2009 roku. Ten wystrzał w górę jest związany z osiąganiem niskich zysków przez firmy spowodowany kryzysem z 2008 r. Dodatkowo 2009r sytuacja się stabilizowała i wyceny spółek poszły do góry. Co spowodowało, aż taką wielkość tego współczynnika. Wysokie wskaźniki P/E obserwowane w 2009 roku były odzwierciedleniem skrajnych warunków rynkowych w tym okresie. W miarę jak gospodarka wracała do wzrostu, a zyski spółek się poprawiały, wskaźniki P/E stopniowo się normalizowały.

W okresie 2020 do 2022 widać dużą zmienność. Wskaźnik w tym okresie także osiągał duże wartości. Co także jest spowodowane wieloma czynnikami takimi jak pandemia czy polityka fiskalna i monetarna państw w tym szczególnie USA.

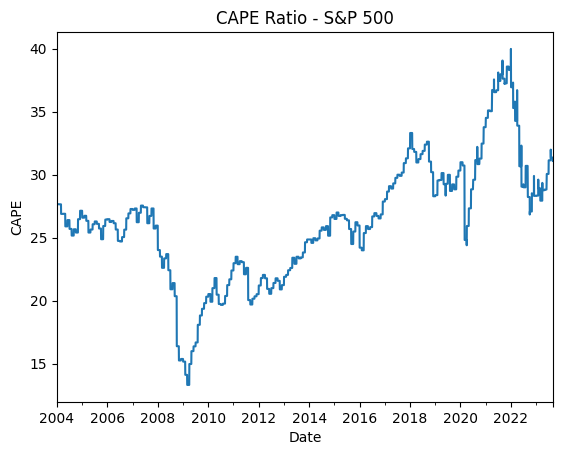
**CAPE Ratio** - (kolumna - 'cape')

CAPE (Cyclically Adjusted Price-to-Earnings) to wskaźnik oceny wyceny akcji, który został opracowany przez laureata Nagrody Nobla, Roberta Schiller. Jest to miara, która uwzględnia inflację i cykl gospodarczy przy ocenie, czy rynki akcji są przeceniane lub nadwyceniane.

Wskaźnik ten porównuje aktualną cenę akcji do średniego zysku firmy na jedną akcję, skorygowanego o inflację, w ciągu ostatnich 10 lat. Główną ideą CAPE jest to, że przy użyciu wieloletniego przeciętnego zysku eliminuje się krótkoterminowe wahania cyklu gospodarczego. Wskaźnik ten pomaga inwestorom oceniać, czy rynki akcji są wyceniane na podstawie długoterminowych fundamentów, czy może występuje przecena lub nadwyżka wyceny.

# plot CAPE  
sp500\_pe\_plot = sp500\_rates['cape'].plot.line( use\_index = True)  
sp500\_pe\_plot.set\_ylabel("CAPE")  
sp500\_pe\_plot.set\_xlabel("Date")  
sp500\_pe\_plot.set\_title('CAPE Ratio - S&P 500')

Text(0.5, 1.0, 'CAPE Ratio - S&P 500')



Nie ma tutaj jak w P/E bardzo gwałtownych zmian wartości. Dzięki wzięciu wyników finansowych z ostatnich 10 lat wykres jest bardziej odporny na krótkoterminowe wahania notowań. Minimum lokalne i globalne CAPE osiągał w czasach większych spadków indeksu. Obecnie jego wartość wynosi ponad 30 i jest to dosyć znacznie powyżej średniej wartości.

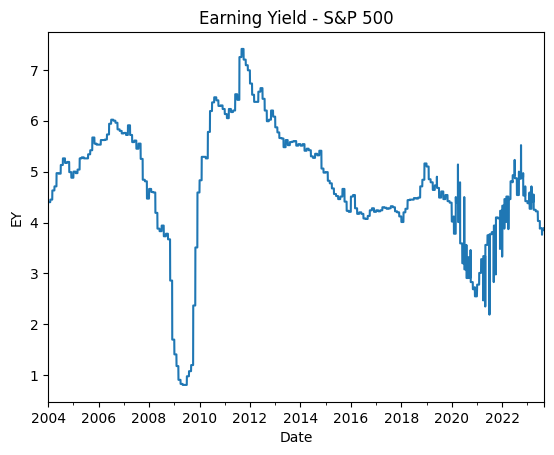
**Earning Yield** - (kolumna - 'earn\_yield')

Earnings Yield to wskaźnik finansowy używany do oceny rentowności inwestycji w akcje lub spółki. Jest to stosunek zysku netto firmy do ceny jej akcji lub kapitalizacji rynkowej (wartości rynkowej spółki na giełdzie). Earnings Yield mierzy, ile procentowo zysku netto generowanego przez spółkę przypada na jedną jednostkę jej akcji. Wartość tego wskaźnika można interpretować podobnie jak wskaźnik P/E (Price-to-Earnings Ratio), ale w odwrotny sposób. Wskaźnik P/E mierzy, ile razy cena akcji zawiera się w zysku netto, podczas gdy Earnings Yield wyraża zysk netto jako procent ceny akcji.

Earnings Yield może być przydatny w porównywaniu różnych spółek lub rynków giełdowych pod kątem potencjalnej rentowności inwestycji. Wyższy Earnings Yield oznacza, że inwestorzy mogą spodziewać się wyższej rentowności z inwestycji w akcje danej spółki.

# plot Earning Yield  
sp500\_pe\_plot = sp500\_rates['earn\_yield'].plot.line( use\_index = True)  
sp500\_pe\_plot.set\_ylabel("EY")  
sp500\_pe\_plot.set\_xlabel("Date")  
sp500\_pe\_plot.set\_title('Earning Yield - S&P 500')

Text(0.5, 1.0, 'Earning Yield - S&P 500')



Ten wskaźnik jest odwrotnością P/E, więc można wysunąć podobne wnioski. Widzimy, że obecnie EY nie jest zbyt wysoko, co oznacza, że rentowność nie jest duża w porównaniu do notowań historycznych.

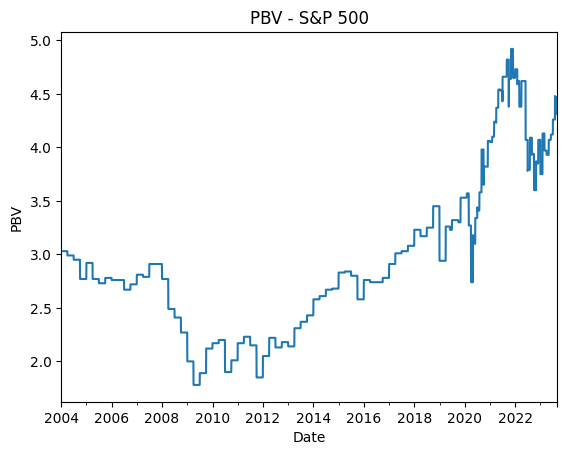
**PBV** - (kolumna - 'pbv')

Wskaźnik PBV (Price-to-Book Value) jest wskaźnikiem finansowym, który służy do oceny wyceny spółki na podstawie jej wartości księgowej. Jest to stosunek ceny rynkowej akcji spółki do jej wartości księgowej na jedną akcję.

Niższy wskaźnik PBV może sugerować, że akcje są bardziej przystępne cenowo, podczas gdy wyższy wskaźnik może wskazywać na przecenę. Wartości PBV poniżej 1 mogą sugerować, że akcje są wyceniane poniżej wartości księgowej.

# plot Earning Yield  
sp500\_pe\_plot = sp500\_rates['pbv'].plot.line( use\_index = True)  
sp500\_pe\_plot.set\_ylabel("PBV")  
sp500\_pe\_plot.set\_xlabel("Date")  
sp500\_pe\_plot.set\_title('PBV - S&P 500')

Text(0.5, 1.0, 'PBV - S&P 500')



Na wykresie widać, że w ostatnich 20 latach indeks dla 500 największych spółek USA miał wartość PBV dużo powyżej 1. Nie tak dawno osiągał swoje maksima, osiągając wartość prawie 5.

Jednak wskaźnik PBV powinien być interpretowany w kontekście zmian, jakie zachodzą w gospodarce. Ponieważ różne branże mogą mieć różne typowe zakresy wartości PBV. Ostatnio coraz większe znaczenia mają firmy z branży technologicznej, a nie przemysłowej, które zazwyczaj posiadają mniej aktywów. Może to powodować większą wartość PBV.

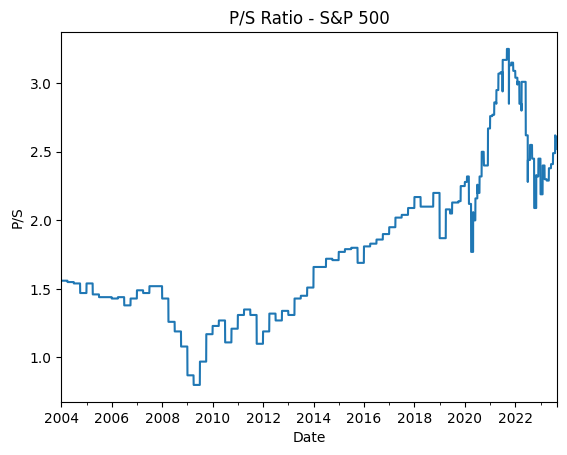
**P/S Ratio** - (kolumna - 'prices\_sales\_ratio')

Wskaźnik P/S (Price-to-Sales Ratio), nazywany również wskaźnikiem wyceny na podstawie przychodów, to wskaźnik finansowy wykorzystywany do oceny wyceny spółek. Jest to stosunek ceny akcji danej spółki do jej przychodów ze sprzedaży na jedną akcję lub na jednostkę udziałową.

Niższy wskaźnik P/S może sugerować, że spółka jest bardziej przystępnie wyceniana w porównaniu do swoich przychodów, podczas gdy wyższy wskaźnik P/S może wskazywać na przecenę lub oczekiwania rynku dotyczące przyszłego wzrostu przychodów. Wskaźnik P/S jest szczególnie przydatny w analizie spółek, które generują duże przychody, ale niekoniecznie osiągają znaczne zyski netto. Jest on również często stosowany w analizie firm technologicznych oraz spółek z sektora wzrostowego, gdzie perspektywa wzrostu przychodów jest kluczowa.

# plot P/S Ratio  
sp500\_pe\_plot = sp500\_rates['price\_sales\_ratio'].plot.line( use\_index = True)  
sp500\_pe\_plot.set\_ylabel("P/S")  
sp500\_pe\_plot.set\_xlabel("Date")  
sp500\_pe\_plot.set\_title('P/S Ratio - S&P 500')

Text(0.5, 1.0, 'P/S Ratio - S&P 500')



Obecnie wartość P/S Ratio jest dosyć wysoka. Widać wyraźny trend wzrostowy z maksimum w 2021 r. Minimum w 2009 r..

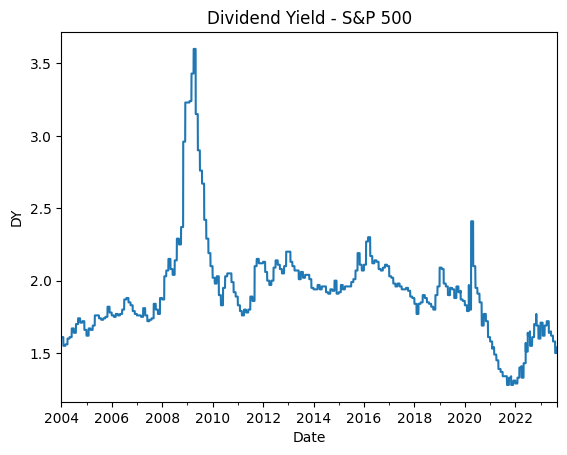
**Dividend Yield** - (kolumna - 'div\_yield')

Wskaźnik Dividend Yield (Wydajność Dywidendy) to wskaźnik finansowy, który pomaga inwestorom ocenić stopę zwrotu z dywidend, jaką mogą uzyskać, inwestując w akcje danej spółki. Wyraża on procentowy stosunek rocznej dywidendy wypłacanej przez spółkę do jej aktualnej ceny akcji.

Wskaźnik Dividend Yield informuje inwestorów o tym, ile procentowo dywidendy stanowią w stosunku do aktualnej wartości rynkowej akcji. Wyższy Dividend Yield oznacza, że inwestorzy mogą oczekiwać wyższej stopy zwrotu z dywidend w porównaniu do ceny akcji. Jest to szczególnie atrakcyjne dla inwestorów poszukujących stałego i stabilnego dochodu z inwestycji. Jednak warto pamiętać, że wyższy Dividend Yield może być spowodowany przez niższą cenę akcji, co może wynikać z różnych czynników, w tym z niedocenienia spółki lub problemów finansowych. Dlatego inwestorzy powinni oceniać Dividend Yield w kontekście innych czynników, takich jak stabilność i wzrost dywidendy, zdolność spółki do generowania dochodu oraz ogólna sytuacja na rynku.

# plot Dividend Yield  
sp500\_pe\_plot = sp500\_rates['div\_yield'].plot.line( use\_index = True)  
sp500\_pe\_plot.set\_ylabel("DY")  
sp500\_pe\_plot.set\_xlabel("Date")  
sp500\_pe\_plot.set\_title('Dividend Yield - S&P 500')

Text(0.5, 1.0, 'Dividend Yield - S&P 500')



Największe wartości zwrot z dywidendy miał w czasie kryzysu z 2008 r. Najniższe wartości na przełomie 2021/2022 r. Obecnie wartość nie jest za wysoka i wynosi około 1.5%.

**Dividend Growth** - (kolumna - 'div\_growth')

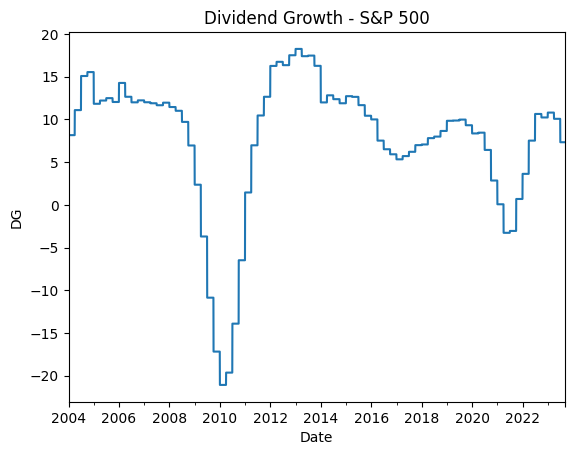
Wskaźnik Dividend Growth (Wzrost Dywidendy) to wskaźnik finansowy mierzący tempo wzrostu dywidend wypłacanych przez spółkę, indeks w ciągu określonego okresu. Jest to wskaźnik często analizowany przez inwestorów, zwłaszcza tych, którzy poszukują inwestycji, które regularnie zwiększają swoje wypłaty dywidendowe. Wzrost dywidend dla spółki jest uważany za wskaźnik zdrowego i stabilnego zarządzania finansami spółki oraz może sugerować, że spółka ma perspektywy wzrostu.

Wskaźnik Dividend Growth wyraża względny wzrost dywidendy jako procent. Wskaźnik Dividend Growth jest ważnym narzędziem dla inwestorów, ponieważ może świadczyć o zdolności spółki do generowania rosnącego dochodu dla swoich akcjonariuszy. Jednak warto zwrócić uwagę, że wzrost dywidendy może być wynikiem wielu czynników, w tym poprawy wyników finansowych, zysków spółki, a także polityki dywidendowej zarządu. Jednak w tym indeksie mamy 500 spółek więc pojedyncze specyficzne zachowania spółek nie mają tak dużej wagi i widzimy jak ogólnie cały rynek, prowadzi politykę dywidend.

Wskaźnik tutaj jest obliczany kwartalnie. Czyli zmiana procentowa jest obliczana na podstawie wysokości dywidend z dwóch ostatnich kwartałów. Dzięki czemu będzie można zauważyć szybciej zmiany na rynku, ale może być więcej szumu, nagłych zmian.

# plot Dividend Growth  
sp500\_pe\_plot = sp500\_rates['div\_growth'].plot.line( use\_index = True)  
sp500\_pe\_plot.set\_ylabel("DG")  
sp500\_pe\_plot.set\_xlabel("Date")  
sp500\_pe\_plot.set\_title('Dividend Growth - S&P 500')

Text(0.5, 1.0, 'Dividend Growth - S&P 500')



Na wykresie widać duży spadek dywidend dla okresów związanych z pogorszeniem sytuacji gospodarczej. Co wiąże się z mniejszymi zyskami w tym okresie. Obecnie wartość Dividend Growth wynosi 7,33%.

**macro\_data - usa\_eco\_rates**

Wskaźniki makroekonomiczne to liczby i statystyki używane do pomiaru i analizy zdrowia i wydajności gospodarki na poziomie makroekonomicznym, czyli ogólnym. Te wskaźniki dostarczają informacji na temat kluczowych aspektów gospodarki danego kraju lub regionu i pozwalają na ocenę jej kondycji, trendów oraz zdolności do wzrostu. Wskaźniki makroekonomiczne są szeroko stosowane przez rządy, banki centralne, inwestorów i analityków do podejmowania decyzji i oceny polityki gospodarczej.

# check for missing values in each column  
missing\_columns = usa\_eco\_rates.isnull().sum()  
pprint("Missing values in column")  
pprint(missing\_columns)  
  
# print all usa\_eco\_rates data info  
print()  
pprint(usa\_eco\_rates.info())

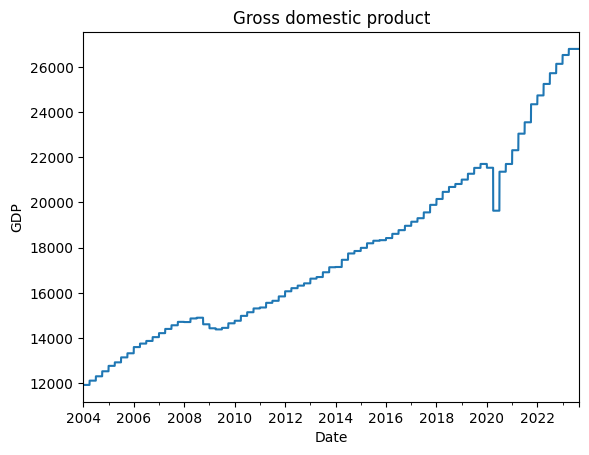
'Missing values in column'  
gdp 0  
gdp\_percent\_change 0  
unemploy\_rate 0  
pmi\_index 0  
cpiaucsl\_pch 0  
fed\_effective\_rate 0  
fed\_effect\_rate\_pch 0  
10usy\_bond\_yield 0  
cons\_sent 0  
cons\_sent\_percent\_change 0  
dtype: int64  
  
<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 7183 entries, 2004-01-01 00:00:00-05:00 to 2023-08-31 00:00:00-04:00  
Data columns (total 10 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 gdp 7183 non-null float64  
 1 gdp\_percent\_change 7183 non-null float64  
 2 unemploy\_rate 7183 non-null float64  
 3 pmi\_index 7183 non-null float64  
 4 cpiaucsl\_pch 7183 non-null float64  
 5 fed\_effective\_rate 7183 non-null float64  
 6 fed\_effect\_rate\_pch 7183 non-null float64  
 7 10usy\_bond\_yield 7183 non-null float64  
 8 cons\_sent 7183 non-null float64  
 9 cons\_sent\_percent\_change 7183 non-null float64  
dtypes: float64(10)  
memory usage: 617.3 KB  
None

**Gross domestic product (GDP)** - (kolumna - 'gdp')

GDP, czyli PKB, Produkt Krajowy Brutto, to kluczowy wskaźnik makroekonomiczny, który mierzy ogólną wartość wszystkich dóbr i usług wytworzonych w danym kraju w określonym okresie, zwykle w ciągu roku. PKB jest używany do pomiaru wielkości i wydajności gospodarki danego kraju i stanowi ważny wskaźnik do analizy i oceny jej zdrowia ekonomicznego.

# plot Gross domestic product  
usa\_eco\_rates\_plot = usa\_eco\_rates['gdp'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("GDP")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('Gross domestic product')

Text(0.5, 1.0, 'Gross domestic product')



Widać duży trend wzrostowy, jeśli chodzi o PKB USA. Są zauważalne dwa spadki PKB związane z kryzysem w 2008 r. oraz z pandemią Covid-19. Jednak te spadki zostały z dużą nawiązką szybko odrobione.

**GDP Percent Change** - (kolumna - 'gdp\_percent\_change')

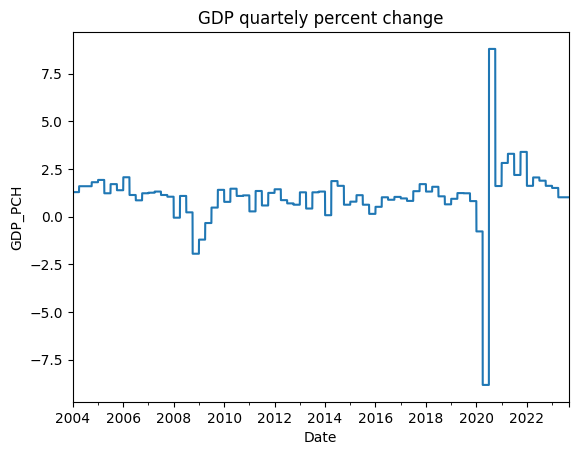
Zmiana procentowa PKB, zwana także wzrostem lub spadkiem procentowym PKB, to wskaźnik używany do określenia zmiany ogólnej wartości Produktu Krajowego Brutto (PKB) w określonym okresie, wyrażonej jako procent. Ten wskaźnik jest ważnym narzędziem do monitorowania tempa wzrostu lub spadku gospodarki. Pozytywna zmiana procentowa oznacza wzrost gospodarczy, podczas gdy ujemna zmiana procentowa oznacza spadek gospodarczy.

Zmiana procentowa PKB jest kluczowym wskaźnikiem do oceny tempa wzrostu gospodarczego lub recesji w danym kraju, lub regionie. Jest często używana przez analityków, inwestorów i ekonomistów do analizy kondycji gospodarczej i prognozowania przyszłych trendów gospodarczych.

Wskaźnik tutaj jest obliczany kwartalnie. Czyli zmiana procentowa jest obliczana na podstawie wysokości PKB z dwóch ostatnich kwartałów. Dzięki czemu będzie można zauważyć szybciej zmiany na rynku, ale podobnie jak dla Dividend Growth może być więcej szumu i nagłych zmian.

# plot GDP quarterly percent change  
usa\_eco\_rates\_plot = usa\_eco\_rates['gdp\_percent\_change'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("GDP\_PCH")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('GDP quarterly percent change')

Text(0.5, 1.0, 'GDP quartely percent change')



Wykres prezentuje dość stabilny kilkuprocentowy kwartalny wzrost PKB gospodarki USA. W czasie kryzysu z 2008 r. widoczna jest recesja gospodarki. W czasie pandemii i spowodowanej przez nie lockdownu widoczna jest bardzo duża recesja, ale jest ona krótkotrwała i szybko odbudowana.

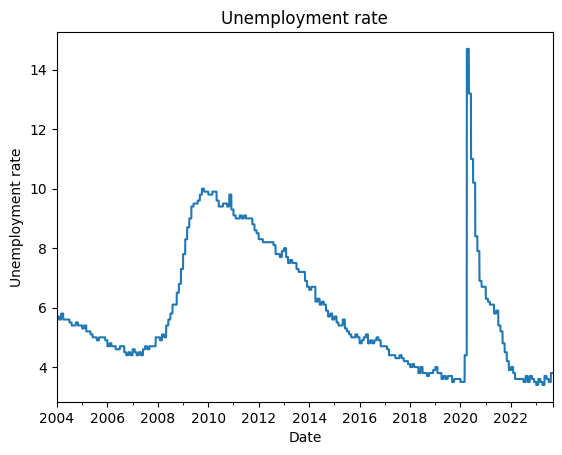
**Unemployment rate** - (kolumna - 'unemploy\_rate')

Krótka stopa bezrobocia, często nazywana również "stopą bezrobocia" lub "stopą bezrobocia ogólnokrajową", to wskaźnik, który mierzy odsetek ludności aktywnej zawodowo, która jest bez pracy i aktywnie jej poszukuje w danym kraju w określonym okresie. Jest to jeden z kluczowych wskaźników rynku pracy i jest często używany do oceny zdrowia gospodarki oraz efektywności polityki rynku pracy.

Im niższa jest stopa bezrobocia, tym bardziej korzystna jest sytuacja na rynku pracy. Ponieważ oznacza to, że większość osób jest zatrudniona. Natomiast wyższa stopa bezrobocia może świadczyć o trudnościach na rynku pracy i słabszym stanie gospodarki.

# plot Unemployment rate  
usa\_eco\_rates\_plot = usa\_eco\_rates['unemploy\_rate'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("Unemployment rate")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('Unemployment rate')

Text(0.5, 1.0, 'Unemployment rate')



Wykres stopy bezrobocia w USA jest dość zmienny i prezentuje dwa wzrosty spowodowane recesją gospodarki. Widać, że w 2008 roku bezrobocie wzrastało nawet po kryzysie i miało dużą bezwładność. Zajęło kilka lat do powrotu do 5% bezrobocia po kryzysie. Natomiast dla pandemii, bardzo dynamiczny wzrost bezrobocia, także ma swoją bezwładność, ale został w miarę szybko zniwelowany. Obecnie amerykańska ma bardzo niskie bezrobocie wynoszące koło 3%.

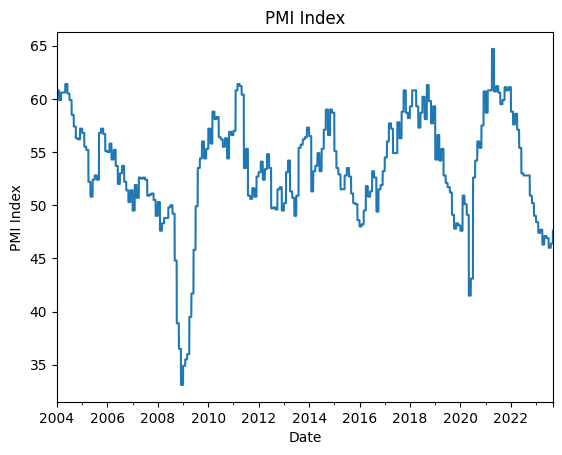
**Purchasing Managers' Index** - (kolumna - 'pmi\_index')

PMI (Purchasing Managers' Index) to indeks zakupów, który mierzy aktywność w sektorze produkcyjnym i usługowym gospodarki. Jest to ważny wskaźnik ekonomiczny używany do monitorowania wydajności gospodarki oraz oceny kierunku, w jakim zmierza. Indeks PMI jest szeroko stosowany na całym świecie i dostarcza informacji na temat kondycji gospodarki oraz tendencji koniunkturalnych.

Ogólna idea indeksu PMI polega na zbieraniu informacji od menedżerów odpowiedzialnych za zakupy (purchasing managers) w firmach. Menedżerowie ci są pytani o różne aspekty działalności swoich firm, takie jak liczba nowych zamówień, produkcja, zatrudnienie, poziom cen surowców itp.. Odpowiedzi te są następnie agregowane i przeliczane na wskaźnik PMI. Indeks PMI jest publikowany regularnie, zwykle co miesiąc, i stanowi ważne narzędzie analizy rynku. Jest używany w wielu krajach na całym świecie i pomaga w zrozumieniu zmian w aktywności gospodarczej oraz w podejmowaniu decyzji zarówno na rynkach finansowych, jak i w biznesie.

# plot PMI Index  
usa\_eco\_rates\_plot = usa\_eco\_rates['pmi\_index'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("PMI Index")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('PMI Index')

Text(0.5, 1.0, 'PMI Index')



Duża zmienność jest widoczna na tym wykresie. Najniższe wartości notowane są w czasach recesji gospodarki USA. Obecnie wskaźnik w stosunku do ostatnich 20 lat jest nieznacznie poniżej 50, co jest niską wartością.

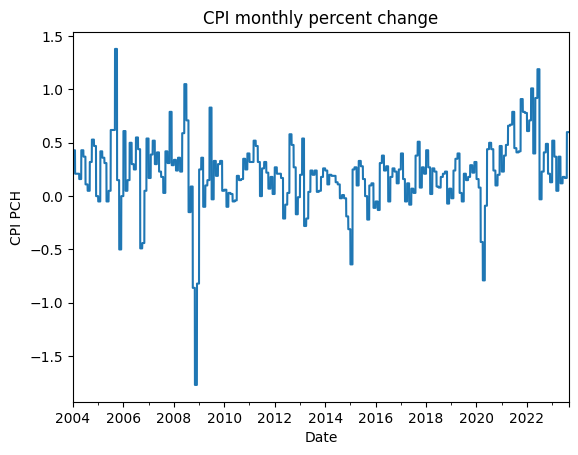
**Consumer Price Index for All Urban Consumers** - (kolumna - 'cpiaucsl\_pch')

CPIAUCSL to skrót od "Consumer Price Index for All Urban Consumers". Jest to indeks cen konsumenckich obliczany w Stanach Zjednoczonych przez Biuro Statystyki Pracy (Bureau of Labor Statistics-BLS). Indeks CPIAUCSL jest używany do pomiaru zmian cen towarów i usług zakupowanych przez gospodarstwa domowe w obszarach miejskich w USA. Jest to jeden z głównych wskaźników inflacji i pomaga w monitorowaniu wzrostu cen na rynku konsumenckim.

W kontekście indeksu CPI-U "percent change" (zmiana procentowa) odnosi się do procentowej zmiany wartości indeksu w określonym okresie. W tych danych wskaźnik jest liczony w każdy miesiąc do okresu wcześniejszego. Indeks PMI jest publikowany regularnie, zwykle co miesiąc, i stanowi ważne narzędzie analizy rynku. Jest to kluczowy wskaźnik ekonomiczny używany przez ekonomistów, banki centralne i rządy do analizy zdrowia gospodarki i podejmowania decyzji politycznych.

# plot Consumer Price Index for All Urban Consumers  
usa\_eco\_rates\_plot = usa\_eco\_rates['cpiaucsl\_pch'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("CPI PCH")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('CPI monthly percent change')

Text(0.5, 1.0, 'CPI monthly percent change')



Z powodu częstego mierzenia tego indeksu widzimy dużą zmienność. W czasach recesji mamy deflację, szczególnie widoczną dla okresu z kryzysu 2008 r. Dla lat wzrostu gospodarczego indeks jest nieznacznie powyżej zera. W 2021 roku widać duży wzrost inflacji, która obecnie wydaje się ustabilizowana i wynosi 0,17% w stosunku do poprzedniego miesiąca.

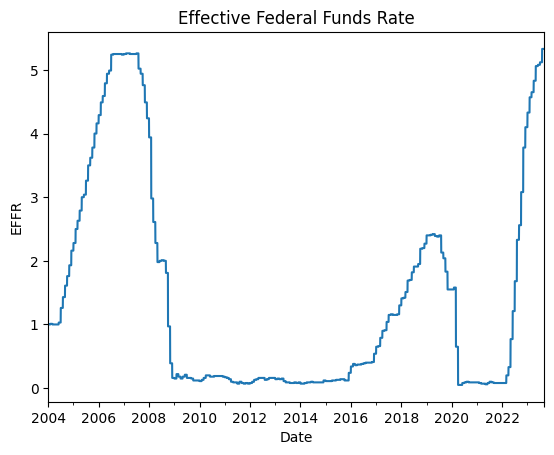
**Effective Federal Funds Rate (EFFR)** - (kolumna - 'fed\_effective\_rate')

"Effective Federal Funds Rate" (EFFR), znana także jako "federal funds effective rate", to jedna z kluczowych stóp procentowych w Stanach Zjednoczonych. Jest to stopa procentowa, po której banki komercyjne udzielają sobie pożyczek na rynku międzybankowym, zwanych "federal funds market". EFFR jest jednym z głównych narzędzi, które Federal Reserve (Fed), amerykański bank centralny, używa do kontrolowania polityki monetarnej.

EFFR jest kluczowym narzędziem, które pozwala Fed regulować poziom aktywności gospodarczej, kontrolować inflację i stabilizować rynki finansowe. Jest to ważny wskaźnik, który jest śledzony przez ekonomistów, inwestorów i instytucje finansowe, ponieważ jego zmiany mają wpływ na szeroki zakres aspektów gospodarki i rynków finansowych.

# plot Effective Federal Funds Rate  
usa\_eco\_rates\_plot = usa\_eco\_rates['fed\_effective\_rate'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("EFFR")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('Effective Federal Funds Rate')

Text(0.5, 1.0, 'Effective Federal Funds Rate')



Na wykresie widać dwa duże wzrosty efektywnych stóp procentowych do wartości powyżej 5%. Obecnie na takim poziomie są stopy procentowe, co jest spowodowane niedawną wysoką inflacją w USA. Można zauważyć, że przez długi okres stopy procentowe wynosiło koło 0%.

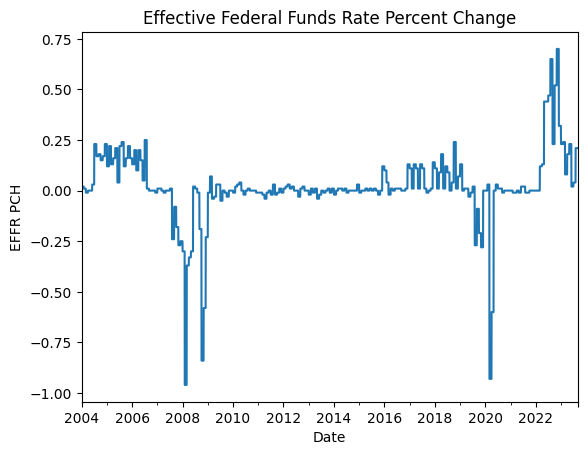
**Effective Federal Funds Rate Percent Change** - (kolumna - 'fed\_effective\_rate\_pch')

Procentowa zmiana stopy "Effective Federal Funds Rate" (EFFR) odnosi się do procentowej zmiany tej stopy procentowej w określonym okresie. Procentowa zmiana w EFFR jest obliczana przez porównanie obecnej stopy do wcześniejszej stopy, zazwyczaj z miesiąca, kwartału lub roku wcześniej. Pomaga to ocenić, o ile procent stopa procentowa wzrosła lub spadła w tym okresie. W tych danych jest obliczana co miesiąc.

Procentowa zmiana stopy "Effective Federal Funds Rate" jest istotnym wskaźnikiem służącym do analizowania tendencji w krótkoterminowych stopach procentowych w Stanach Zjednoczonych. Jest śledzona uważnie przez ekonomistów, decydentów politycznych i inwestorów, ponieważ może dostarczać informacji na temat stanowiska Federalnej Rezerwy w kwestii polityki pieniężnej oraz jej działań mających na celu kontrolowanie inflacji i wspieranie wzrostu gospodarczego.

# plot Effective Federal Funds Rate Monthly Percent Change  
usa\_eco\_rates\_plot = usa\_eco\_rates['fed\_effect\_rate\_pch'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("EFFR PCH")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('Effective Federal Funds Rate Percent Change')

Text(0.5, 1.0, 'Effective Federal Funds Rate Percent Change')



Na wykresie widać duże okresy stabilizacji efektywnych stóp procentowych. Jednak są widoczne trzy okresy-kryzys 2008, pandemia 2020, inflacja 2021, w których nastąpiły duże i częste zmiany stóp procentowych.

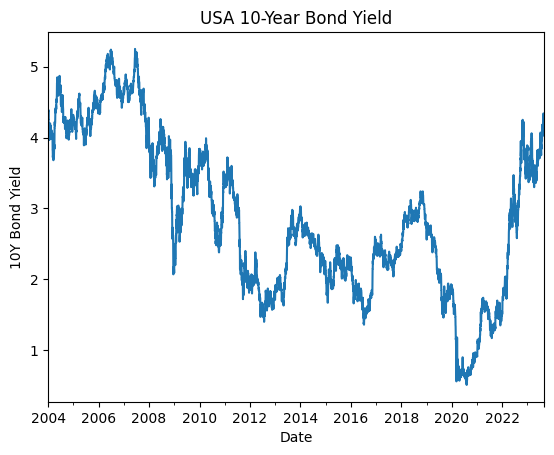
**USA 10-Year Bond Yield** - (kolumna - '10usy\_bond\_yield')

"USA 10-year bond yield" oznacza rentowność lub stopę procentową na 10-letnich obligacjach skarbowych Stanów Zjednoczonych. Reprezentuje to roczny zwrot, jaki inwestor może oczekiwać, trzymając 10-letnią obligację rządową Stanów Zjednoczonych do daty wykupu.

Rentowność 10-letnich obligacji skarbowych USA to kluczowy wskaźnik finansowy, który odzwierciedla stopę procentową na 10-letnich obligacjach rządowych Stanów Zjednoczonych. Gra on istotną rolę na rynkach finansowych i pełni funkcję wskaźnika ekonomicznego, który odzwierciedla warunki gospodarcze i nastroje inwestorów.

# plot USA 10-Year Bond Yield  
usa\_eco\_rates\_plot = usa\_eco\_rates['10usy\_bond\_yield'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("10Y Bond Yield")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('USA 10-Year Bond Yield')

Text(0.5, 1.0, 'USA 10-Year Bond Yield')



Ogólnie na wykresie jest widoczny trend spadkowy przez większość okresu. Od wartości powyżej 4%, rentowność 10-letnich obligacji spadła prawie do zera w 2020 r.. W ostatnich 2 latach widać odwrócenie trendu i duży szybki wzrost do wartości powyżej 4%.

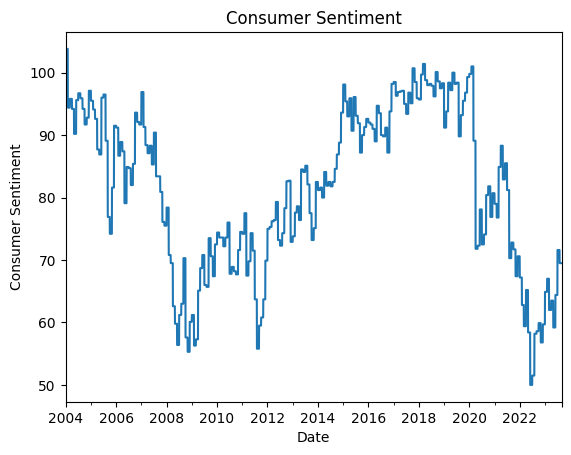
**Consumer Sentiment** - (kolumna - 'cons\_sent')

Jeden z popularnych wskaźników mierzących nastroje konsumentów w Stanach Zjednoczonych. Mierzony przez "University of Michigan Consumer Sentiment Index" (Indeks Nastrojów Konsumentów Uniwersytetu Michigan). "Consumer sentiment" odnosi się do ogólnego nastroju i postaw konsumentów wobec gospodarki, finansów osobistych oraz przyszłości. Jest to wskaźnik mierzący poziom pewności i optymizmu konsumentów, który może wpływać na ich decyzje zakupowe i wydatki.

Indeks Nastrojów Konsumentów jest używany jako wskaźnik do analizy klimatu konsumenckiego i może dostarczać wskazówek dotyczących przyszłego zachowania rynku. Jest to również ważny element analizy ekonomicznej, który pomaga monitorować zmiany nastrojów konsumentów w odpowiedzi na zmieniające się warunki gospodarcze i wydarzenia na rynku.

# plot Consumer Sentiment  
usa\_eco\_rates\_plot = usa\_eco\_rates['cons\_sent'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("Consumer Sentiment")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('Consumer Sentiment')

Text(0.5, 1.0, 'Consumer Sentiment')



Im niższa wartość, tym gorsze nastroje konsumentów. W recesjach gospodarki widoczne są najniższe wartości tego indeksu. Obecnie indeks nastoje poprawiają się po pandemii Covid-19, ale patrząc na cały wykres dalej, nie są na wysokim poziomie.

**Consumer Sentiment Percent Change** - (kolumna - 'cons\_sent\_percent\_change')

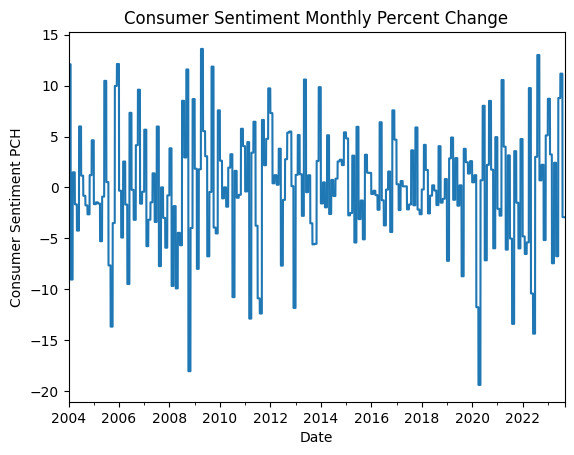
"Consumer sentiment percent change" dla Stanów Zjednoczonych odnosi się do procentowej zmiany wskaźnika nastrojów konsumentów (consumer sentiment) w danym okresie w porównaniu do poprzedniego okresu. Jest to miara, która informuje o tym, o ile procent wskaźnik nastrojów konsumentów wzrósł lub spadł w określonym czasie.

Ten wskaźnik jest istotny, ponieważ pomaga analizować, czy nastroje konsumentów stają się bardziej optymistyczne (jeśli wartość jest dodatnia) lub bardziej pesymistyczne (jeśli wartość jest ujemna) w stosunku do wcześniejszego okresu. To może dostarczać informacji na temat zmian w postrzeganiu gospodarki i wpływać na decyzje konsumentów dotyczące zakupów i wydatków.

Zmiany w nastrojach konsumentów są istotne dla ekonomistów, inwestorów i polityków, ponieważ mogą wpłynąć na zachowanie gospodarki, rynek pracy i inne aspekty ekonomiczne. Dlatego "Consumer sentiment percent change" jest śledzony i analizowany jako część badania nastrojów konsumentów.

# plot Consumer Sentiment Monthly Percent Change  
usa\_eco\_rates\_plot = usa\_eco\_rates['cons\_sent\_percent\_change'].plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("Consumer Sentiment PCH")  
usa\_eco\_rates\_plot.set\_xlabel("Date")  
usa\_eco\_rates\_plot.set\_title('Consumer Sentiment Monthly Percent Change')

Text(0.5, 1.0, 'Consumer Sentiment Monthly Percent Change')



Widać dużą zmienność tego wykresu. Wykres jest bardzo trudny do odczytu. Ilość zmian powoduje, że trudno jest wyczytać coś z tego wykresu. Możliwe, że zwiększenie okresu porównywania wskaźnika do roku spowodowałoby większą czytelność i możliwość wysnucia ciekawych wniosków.

**goog\_trend**

Google Trends to narzędzie dostarczane przez Google, które umożliwia analizowanie popularności i wyszukiwań określonych fraz lub słów kluczowych w wyszukiwarce Google w określonym czasie i w określonych regionach geograficznych. To narzędzie pozwala zrozumieć, jakie tematy są popularne w danym momencie oraz jak zmienia się ich popularność w czasie.

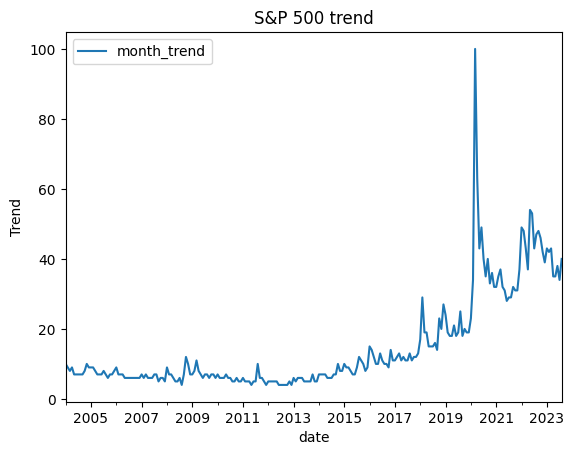
Google Trends jest używane w różnych celach, takich jak analiza rynku, badanie trendów marketingowych, badanie zainteresowań społeczności online, śledzenie popularności produktów i wiele innych. Jest to przydatne narzędzie zarówno dla profesjonalistów w dziedzinie marketingu, jak i dla osób śledzących trendy w danym obszarze tematycznym.

# check for missing values in each column  
missing\_columns = goog\_trend.isnull().sum()  
pprint("Missing values in column")  
pprint(missing\_columns)  
# print all goog\_trend data info  
print()  
pprint(goog\_trend.info())

'Missing values in column'  
month\_trend 0  
dtype: int64  
  
<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 236 entries, 2004-01-01 00:00:00-05:00 to 2023-08-01 00:00:00-04:00  
Data columns (total 1 columns):  
 # Column Non-Null Count Dtype  
--- ------ -------------- -----  
 0 month\_trend 236 non-null int64  
dtypes: int64(1)  
memory usage: 3.7 KB  
None

# plot trend  
goog\_trend\_plot = goog\_trend.plot.line( use\_index = True)  
goog\_trend\_plot.set\_ylabel("Trend")  
goog\_trend\_plot.set\_title('S&P 500 trend')

Text(0.5, 1.0, 'S&P 500 trend')



Z wykresu widać początkowo stabilną popularność wyszukiwań indeksu, ale od około roku 2015 następuje większa popularność. Szczyt równy 100 (trend jest mierzony w zakresie od 0 do 100) przypada w okresie wybuchu pandemii Covid-19.

# print date with max popularity  
pprint(goog\_trend.idxmax().max())  
pprint(goog\_trend.max().max())

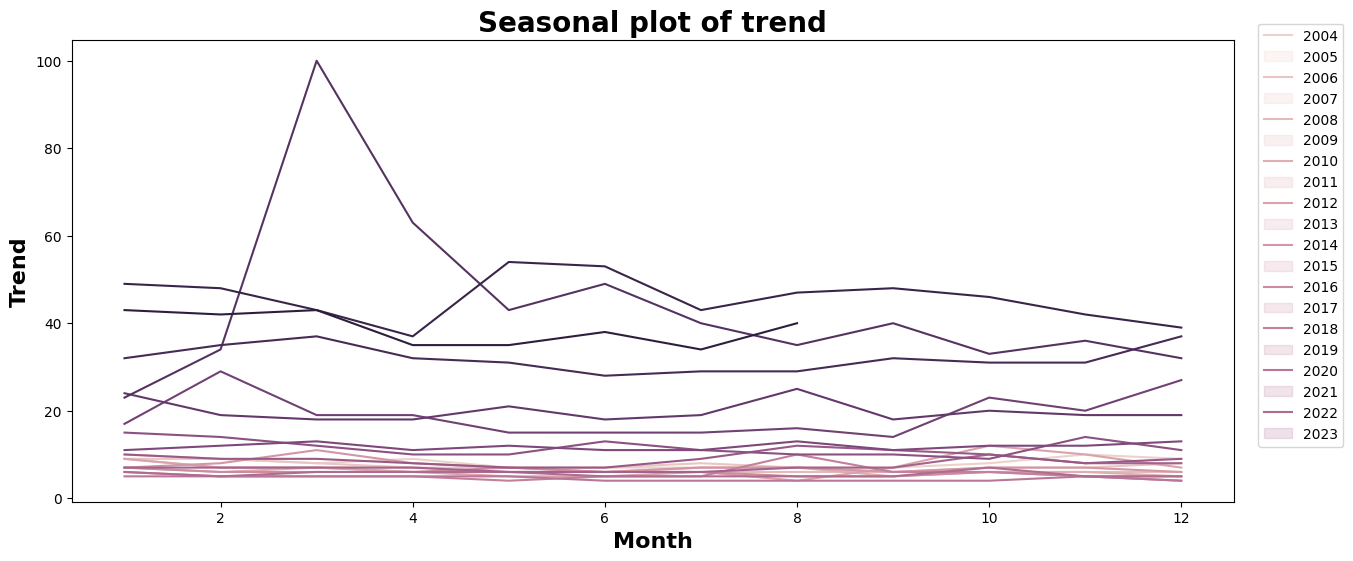
Timestamp('2020-03-01 00:00:00-0500', tz='America/New\_York')  
100

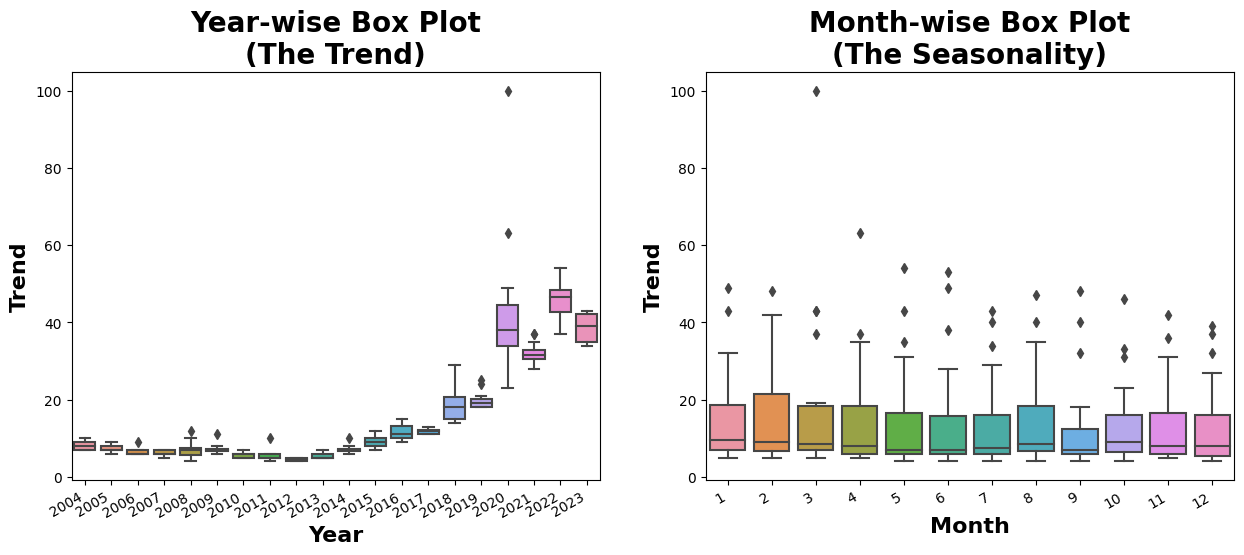
# seasonality  
goog\_trend\_copy = goog\_trend.copy()  
goog\_trend\_copy['year'] = goog\_trend\_copy.index.year  
goog\_trend\_copy['month'] = goog\_trend\_copy.index.month  
print(goog\_trend\_copy['year'].unique())  
goog\_trend\_copy.columns

[2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017  
 2018 2019 2020 2021 2022 2023]

Index(['month\_trend', 'year', 'month'], dtype='object')

# seasonality of trend column  
column = "month\_trend"  
# create a figure with one plot  
\_, ax = plt.subplots(figsize=(15, 6))  
  
# plot the seasonal line plot  
sns.lineplot(x=goog\_trend\_copy['month'], y=goog\_trend\_copy[column], hue = goog\_trend\_copy['year'])  
  
# format the plot  
ax.set\_title('Seasonal plot of trend', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax.set\_xlabel('Month', fontsize = 16, fontdict=dict(weight='bold'))  
ax.set\_ylabel('Trend', fontsize = 16, fontdict=dict(weight='bold'))  
ax.legend(labels = [str(2004+i) for i in range(20)], bbox\_to\_anchor=(1.1, 1.05))  
  
# create a figure with two plots  
\_, ax = plt.subplots(nrows=1, ncols=2, figsize=(15, 6))  
  
# plot the year-wise box plot  
sns.boxplot(x=goog\_trend\_copy['year'], y=goog\_trend\_copy[column], ax=ax[0])  
  
# format the year-wise box plot  
ax[0].set\_title('Year-wise Box Plot\n(The Trend)', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax[0].set\_xlabel('Year', fontsize = 16, fontdict=dict(weight='bold'))  
ax[0].set\_ylabel('Trend', fontsize = 16, fontdict=dict(weight='bold'))  
  
# plot the month-wise box plot  
sns.boxplot(x=goog\_trend\_copy['month'], y=goog\_trend\_copy[column], ax=ax[1])  
  
# format the month-wise box plot  
ax[1].set\_title('Month-wise Box Plot\n(The Seasonality)', fontsize = 20, loc='center', fontdict=dict(weight='bold'))  
ax[1].set\_xlabel('Month', fontsize = 16, fontdict=dict(weight='bold'))  
ax[1].set\_ylabel('Trend', fontsize = 16, fontdict=dict(weight='bold'))  
  
# rotate the x-axis labels for better readability  
fig.autofmt\_xdate()





**Seasonal plot of Volume**

Dla poszczególnych lat nie widać wyraźnej sezonowości. Widoczny jest duży wzrost popularności frazy "S&P 500" w marcu 2020 r..

**Year wise Box Plot**

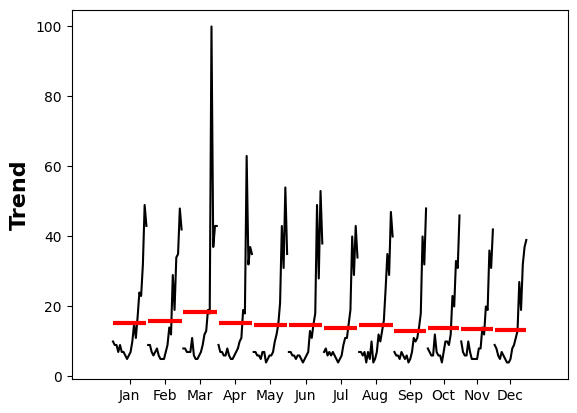
Na tym wykresie trend jest przez pierwsze 10 lat i stabilny na równym poziomie. Następnie widać coraz większą popularność wyszukiwań, co może świadczyć o większej liczbie osób zainteresowanych inwestowaniem. Największą mediana wyniosła dla roku 2022. Największa zmienność jest dla roku 2020, co jest to związane z pandemią. Znaczne wartości odstające pojawiają się także w tym roku.

**Month wise Box Plot**

Nie są widoczne dużej rozbieżności. Nie można stwierdzić istniejącej sezonowości.

# aggregating trend to monthly scaled index  
fig, ax = plt.subplots()  
  
month\_plot(goog\_trend['month\_trend'], ax=ax)  
ax.set\_ylabel('Trend', fontsize = 16, fontdict=dict(weight='bold'))

Text(0, 0.5, 'Trend')



Nie widać dużej zmienności dla poszczególnych miesięcy dla tych danych. Jedynie marzec ma większą średnią popularności. Co jest związane z marcem 2020 r. i osiągnięciem największej popularności. Znacznie większej niż dla innych okresów.

**sp500**

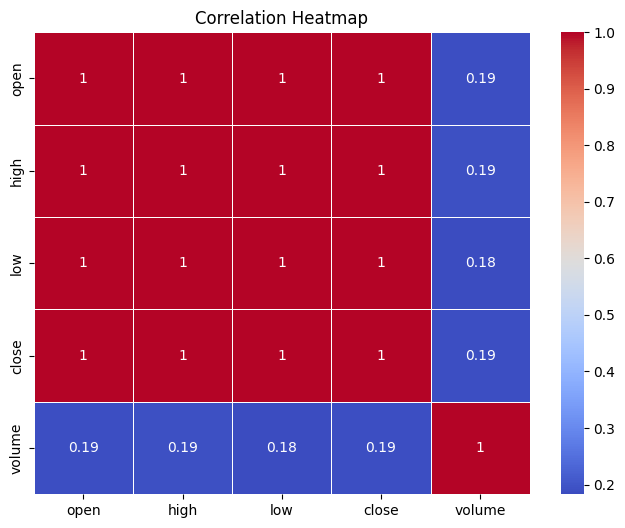
# basic stats about data values  
sp500.describe()

open high low close volume \  
count 4950.000000 4950.000000 4950.000000 4950.000000 4.950000e+03   
mean 2099.723726 2111.689146 2086.960859 2100.110286 3.735805e+09   
std 1051.776543 1057.550996 1045.629946 1051.937899 1.311274e+09   
min 679.280029 695.270020 666.789978 676.530029 5.045800e+08   
25% 1264.152527 1271.929993 1257.459961 1264.190033 3.030322e+09   
50% 1754.695007 1765.330017 1747.054993 1755.869995 3.663470e+09   
75% 2752.684937 2764.349915 2735.652527 2748.897461 4.334042e+09   
max 4804.509766 4818.620117 4780.040039 4796.560059 1.145623e+10   
  
 dividends stock splits   
count 4950.0 4950.0   
mean 0.0 0.0   
std 0.0 0.0   
min 0.0 0.0   
25% 0.0 0.0   
50% 0.0 0.0   
75% 0.0 0.0   
max 0.0 0.0

# delete unneeded columns  
del sp500['dividends']  
del sp500['stock splits']  
sp500

open high low close \  
date   
2004-01-02 00:00:00-05:00 1111.920044 1118.849976 1105.079956 1108.479980   
2004-01-05 00:00:00-05:00 1108.479980 1122.219971 1108.479980 1122.219971   
2004-01-06 00:00:00-05:00 1122.219971 1124.459961 1118.439941 1123.670044   
2004-01-07 00:00:00-05:00 1123.670044 1126.329956 1116.449951 1126.329956   
2004-01-08 00:00:00-05:00 1126.329956 1131.920044 1124.910034 1131.920044   
... ... ... ... ...   
2023-08-25 00:00:00-04:00 4389.379883 4418.459961 4356.290039 4405.709961   
2023-08-28 00:00:00-04:00 4426.029785 4439.560059 4414.979980 4433.310059   
2023-08-29 00:00:00-04:00 4432.750000 4500.140137 4431.680176 4497.629883   
2023-08-30 00:00:00-04:00 4500.339844 4521.649902 4493.589844 4514.870117   
2023-08-31 00:00:00-04:00 4517.009766 4532.259766 4507.390137 4507.660156   
  
 volume   
date   
2004-01-02 00:00:00-05:00 1153200000   
2004-01-05 00:00:00-05:00 1578200000   
2004-01-06 00:00:00-05:00 1494500000   
2004-01-07 00:00:00-05:00 1704900000   
2004-01-08 00:00:00-05:00 1868400000   
... ...   
2023-08-25 00:00:00-04:00 3296180000   
2023-08-28 00:00:00-04:00 2957230000   
2023-08-29 00:00:00-04:00 3354820000   
2023-08-30 00:00:00-04:00 3064110000   
2023-08-31 00:00:00-04:00 3946360000   
  
[4950 rows x 5 columns]

# calculate the correlation matrix  
correlation\_matrix = sp500.corr()  
plt.figure(figsize=(8, 6))  
sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)  
plt.title('Correlation Heatmap')  
plt.show()



Jak można było się domyślić, korelacja wynosi 1 dla czterech zmiennych odnoszących się do ceny. Natomiast pomiędzy notowaniami indeksu a ilością wolumenu jest korelacja niewielka wynosząca 0,19.

# leave only close price column  
del sp500['open']  
del sp500['high']  
del sp500['low']  
sp500

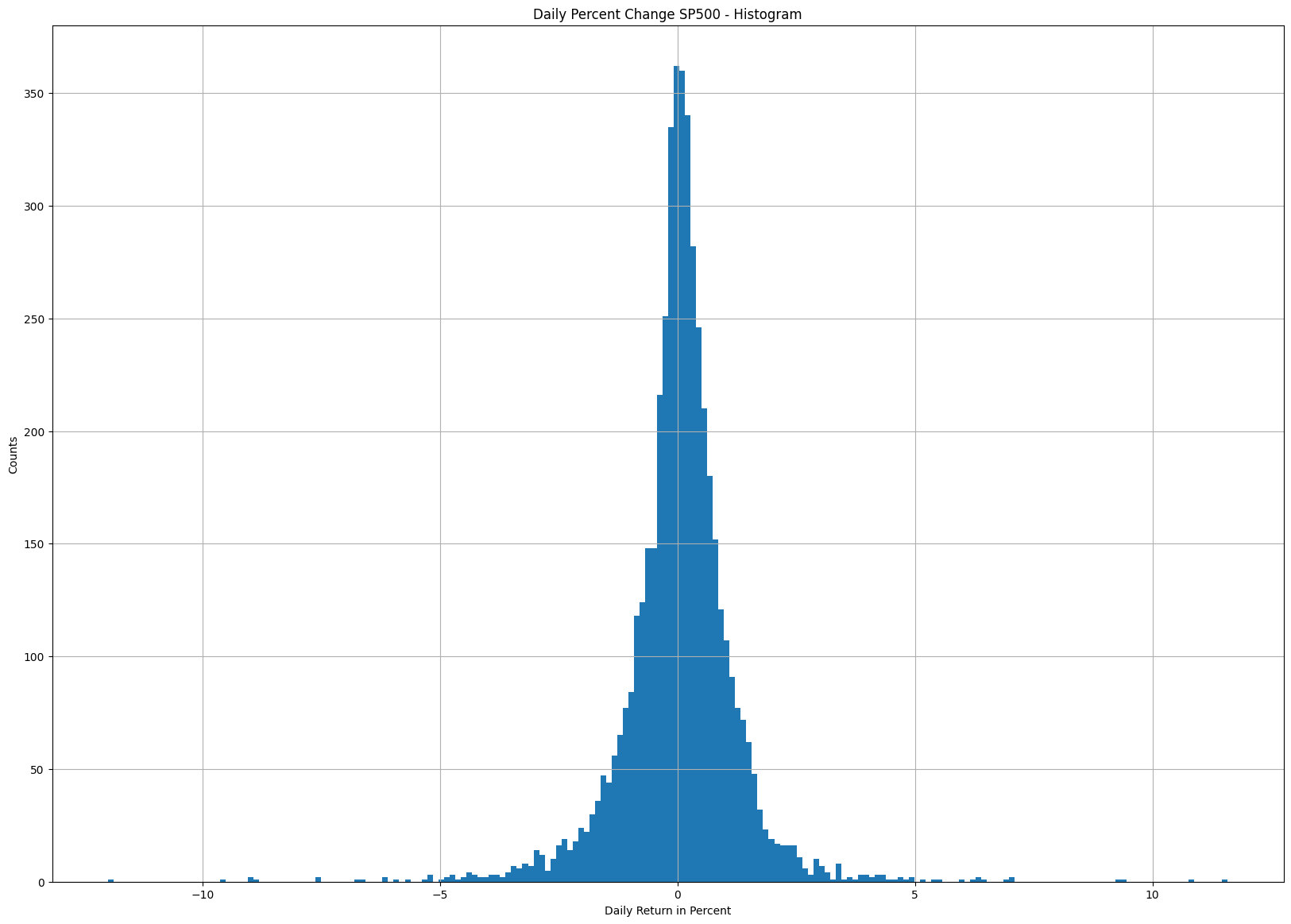
close volume  
date   
2004-01-02 00:00:00-05:00 1108.479980 1153200000  
2004-01-05 00:00:00-05:00 1122.219971 1578200000  
2004-01-06 00:00:00-05:00 1123.670044 1494500000  
2004-01-07 00:00:00-05:00 1126.329956 1704900000  
2004-01-08 00:00:00-05:00 1131.920044 1868400000  
... ... ...  
2023-08-25 00:00:00-04:00 4405.709961 3296180000  
2023-08-28 00:00:00-04:00 4433.310059 2957230000  
2023-08-29 00:00:00-04:00 4497.629883 3354820000  
2023-08-30 00:00:00-04:00 4514.870117 3064110000  
2023-08-31 00:00:00-04:00 4507.660156 3946360000  
  
[4950 rows x 2 columns]

# make a new column with daily price percent change  
sp500['daily\_pch'] = sp500['close'].pct\_change() \* 100  
sp500

close volume daily\_pch  
date   
2004-01-02 00:00:00-05:00 1108.479980 1153200000 NaN  
2004-01-05 00:00:00-05:00 1122.219971 1578200000 1.239534  
2004-01-06 00:00:00-05:00 1123.670044 1494500000 0.129215  
2004-01-07 00:00:00-05:00 1126.329956 1704900000 0.236716  
2004-01-08 00:00:00-05:00 1131.920044 1868400000 0.496310  
... ... ... ...  
2023-08-25 00:00:00-04:00 4405.709961 3296180000 0.671797  
2023-08-28 00:00:00-04:00 4433.310059 2957230000 0.626462  
2023-08-29 00:00:00-04:00 4497.629883 3354820000 1.450831  
2023-08-30 00:00:00-04:00 4514.870117 3064110000 0.383318  
2023-08-31 00:00:00-04:00 4507.660156 3946360000 -0.159694  
  
[4950 rows x 3 columns]

plt.figure(figsize=(20, 14))  
sp500['daily\_pch'].hist(bins=200)  
plt.xlabel('Daily Return in Percent')  
plt.ylabel('Counts')  
plt.title('Daily Percent Change SP500 - Histogram')

Text(0.5, 1.0, 'Daily Percent Change SP500 - Histogram')



# calculate the number of daily return above zero, equal to zero, and below zero  
above\_zero\_count = (sp500['daily\_pch'] > 0).sum()  
equal\_to\_zero\_count = (sp500['daily\_pch'] == 0).sum()  
below\_zero\_count = (sp500['daily\_pch'] < 0).sum()  
  
# Print the counts  
print(f'Number of daily returns above zero: {above\_zero\_count}')  
print(f'Number of daily returns equal to zero: {equal\_to\_zero\_count}')  
print(f'Number of daily returns below zero: {below\_zero\_count}')

Number of daily returns above zero: 2689  
Number of daily returns equal to zero: 2  
Number of daily returns below zero: 2258

# calculate the percent daily returns above zero, equal to zero, and below zero  
num\_rows = sp500.shape[0]  
  
# Print the counts  
print(f'Percent of daily returns above zero: {above\_zero\_count/num\_rows\*100:.2f}%')  
print(f'Percent of daily returns equal to zero: {equal\_to\_zero\_count/num\_rows\*100:.2f}%')  
print(f'Percent of daily returns below zero: {below\_zero\_count/num\_rows\*100:.2f}%')

Percent of daily returns above zero: 54.32%  
Percent of daily returns equal to zero: 0.04%  
Percent of daily returns below zero: 45.62%

# calculate the percent change between the first and last day for index, total return fow whole period  
first\_row\_value = sp500['close'].iloc[0]  
last\_row\_value = sp500['close'].iloc[-1]  
percent\_change = ((last\_row\_value - first\_row\_value) / first\_row\_value) \* 100  
  
print(f'Percent change S&P 500 for index: {percent\_change:.2f}%')

Percent change S&P 500 for index: 306.65%

Z powyższych danych można zauważyć, że większość dni kończyła się pozytywnie dla inwestorów. Około 54.32% notowań zakończyło się na plusie dla indeksu S&P 500. W czasie okresu badań wynoszącego prawie 20 lat indeks wzrósł znacznie, ponieważ aż o 306.65%

# create column "target" for prediction for next day - True if the index increases the next day, or False if it decreases  
sp500["tomorrow"] = sp500["close"].shift(-1)  
sp500.dropna(subset=['tomorrow'], inplace=True)  
sp500["target"] = (sp500["tomorrow"]> sp500["close"]).astype(int)  
del sp500['tomorrow']  
sp500.head(10)

close volume daily\_pch target  
date   
2004-01-02 00:00:00-05:00 1108.479980 1153200000 NaN 1  
2004-01-05 00:00:00-05:00 1122.219971 1578200000 1.239534 1  
2004-01-06 00:00:00-05:00 1123.670044 1494500000 0.129215 1  
2004-01-07 00:00:00-05:00 1126.329956 1704900000 0.236716 1  
2004-01-08 00:00:00-05:00 1131.920044 1868400000 0.496310 0  
2004-01-09 00:00:00-05:00 1121.859985 1720700000 -0.888761 1  
2004-01-12 00:00:00-05:00 1127.229980 1510200000 0.478669 0  
2004-01-13 00:00:00-05:00 1121.219971 1595900000 -0.533166 1  
2004-01-14 00:00:00-05:00 1130.520020 1514600000 0.829458 1  
2004-01-15 00:00:00-05:00 1132.050049 1695000000 0.135339 1

**nyt\_hl**

Analiza sentymentu nyt\_hl

nyt\_hl\_copy = nyt\_hl.copy()  
nyt\_hl = nyt\_hl\_copy

nyt\_hl['headline'].iloc[0]

['FreeMarkets Agrees to Buy Auction Unit of Covisint',  
 'National Briefing | South: Arkansas: Assembly Fails To Meet Court-Mandated Deadline',  
 'Clark Courts Veterans in Swing Through South',  
 'Paid Notice: Deaths PAONE, NICOLA',  
 'The Time We Thought We Knew',  
 'Mad Cow Disease: Lessons Unlearned (4 Letters)',  
 'Giants Want a Teacher Who Is Adept at Winning',  
 'An Odd Sight in Iran as Aid Team Tents Go Up: The U.S. Flag',  
 'Mad Cow Disease: Lessons Unlearned',  
 'Art Meets Artichoke',  
 'Corrections',  
 'Paid Notice: Deaths BECKER, SCOTT',  
 'A Place to Store Your Pictures, and to Show Them Off, Too',  
 'Iranians Appear to Warm to Easing Tensions With U.S.',  
 'So You Want to Be a Star? Help for Your Audition Tape',  
 'Blocks; Capturing the Spirit of 1776, but With a Different Number',  
 'Man Behind Sept. 11 Fund Describes Effort as a Success, With Reservations',  
 'Paid Notice: Deaths SIEGEL, ETHEL BEYERSKY',  
 'CONGOLEUM FILES BANKRUPTCY PLAN WITH ASBESTOS LIABILITY',  
 'For the Ex-Buccaneer, A Pillage-Free Playlist',  
 "BANKRUPTCY JUDGE REJECTS EXIDE'S REORGANIZATION PLAN",  
 'A Wary Eye On Sites For Music Sharing',  
 "Year's Big Rally Helps Investors Regain Ground",  
 "Washington Memo; Election? Mad Cow? Who Cares? The 'Skins Need a New Coach",  
 'Paid Notice: Memorials DEIN, GITTIE',  
 'National Briefing | South: Georgia: 6 Men Indicted In Cross Burning',  
 'A Small Company, a Global Approach',  
 'Corrections',  
 'Reimagining the Sofabed, And a Table Bed Is Born',  
 'Economic Scene; What separates rich nations from poor nations? History can provide some answers.',  
 'Corrections',  
 'Fireworks and Sharpshooters Welcome 2004',  
 'Curb Corporate Rights',  
 "With Glut at Point Guard, Knicks Wonder, Who's Next?",  
 'Rowland Helped Children Of a Partner Get State Jobs',  
 'Outdated Drug Laws',  
 'Paid Notice: Deaths MANGANO, AMERICO',  
 'Calming the Mentally Ill',  
 'Boompf, Boompf to Infinity And Beyond With Hancock',  
 'What We Will Do in 2004',  
 'Paid Notice: Deaths GOLDMUNTZ, RITA (NEE WACHSMAN)',  
 'John Gregory Dunne, Novelist, Screenwriter and Observer of Hollywood, Is Dead at 71',  
 'Unemployment Claims Reach Lowest Level in Three Years',  
 'Adulation for an Ascending Merry Diva',  
 'Chief Justice Attacks a Law As Infringing On Judges',  
 'World Business Briefing | Asia: India: Economic Growth Surges',  
 'FLIGHT SENT BACK ON TERROR FEAR, U.S. OFFICIALS SAY',  
 'The Writing on the Wall',  
 'Paid Notice: Deaths STEVENSON, ISABELLE',  
 'ABN AMRO AGREES TO SELL UNIT TO MERRILL LYNCH',  
 'Paid Notice: Deaths KLAR, HENRY A.',  
 "New Math Adds Up to an Extended Run for Ukraine's President",  
 'RAYOVAC TO CLOSE REMINGTON RAZOR SERVICE STORES',  
 'Sun Screen in a Back Alley',  
 "No 'Auld Lang Syne' for Jailed Russian Oilman",  
 'Historic Savannah Takes to Hermès',  
 'Corrections',  
 'Empties, Wilted Flowers: The Cleanup Begins',  
 'Metro Briefing | New York: Bronx: Fatal Fire Was Deliberately Set',  
 'Paid Notice: Deaths HALPRIN, NATALIE',  
 'Paid Notice: Deaths COTTON, ROBERT HENRY',  
 'Radio Begins to Catch the Eyes as Well as the Ears of Listeners',  
 'In Packed Square, Fear Is Pushed Out By the Big Party',  
 'Paid Notice: Deaths KHEEL, ANN SUNSTEIN',  
 'Paid Notice: Deaths PRESSMAN, WILLIAM B.',  
 'Rebuilding a Life Shattered by War (6 Letters)',  
 'Study Clears Way For 900 New Cabs; Fares May Increase',  
 'Rebuilding a Life Shattered by War',  
 'World Business Briefing | Europe: Britain: Cloning Technology Sold',  
 'Paid Notice: Deaths KIOK, HELEN',  
 'U.S. Acts to Ban Tax Shelters Using Roth I.R.A.',  
 'World Business Briefing | Europe: France: Jobless Rate Declines',  
 'Following Lead of Kidd, Nets Keep Party Going',  
 'A Seated Tour Of Europe',  
 'Paid Notice: Deaths KELSON, SELMA (NEE LURIE)',  
 'Corrections',  
 'Residential Sales',  
 'The Neediest Cases; A Mother and Daughter Regain Some Peace of Mind',  
 'The Latest Box Score: 27 Points, 13 Rebounds, 3 Tattoos',  
 'London Journal; The Sceptered Isle Is Appraised: $8 Trillion and Change',  
 'GARDEN Q.&A.',  
 'Paid Notice: Deaths ZIMERING, CHARLES',  
 'Bush Out of Sight Since Arrival at Ranch',  
 'Preventing Blackouts',  
 "Sports of The Times; Spurrier's Exit Is a Blow to All College Coaches With N.F.L. Aspirations",  
 '7 Detained As Parmalat Investigation Is Widened',  
 'Between the Years',  
 'Paid Notice: Deaths SMITH, VINCENT',  
 "Pakistani Leader's New Tactic: Persuasion",  
 'Wells Receives Better Offer And Leaves Yanks for Padres',  
 'COMCAST SEEKS TO SELL STAKE IN CABLE UNIT',  
 '$88 Billion in Sales of Emerging-Market Bonds',  
 'Titans Are N.F.L. Bully, Except Against Ravens',  
 'Rebuilding a Life Shattered by War',  
 'THE MARKETS: CURRENCIES',  
 'Paid Notice: Memorials GRIST, KEITH PAUL',  
 'COMPANY BRIEFS',  
 'Giving Up Those Weapons: After Libya, Who Is Next?',  
 "New Year's Wishes, Revised",  
 'Paid Notice: Deaths SHER, ROSALIE L.',  
 'ONLINE DIARY',  
 '10 Years Later, Italy Assesses Change In How Its Fabled Museums Are Run',  
 'Mad Cow Disease: Lessons Unlearned',  
 'In Your Hand, a Gallery to Go',  
 "India's Soybean Farmers Join the Global Village",  
 'Journeys That Defined A Century',  
 'World Business Briefing | Americas: Canada: Energy Company Plans To Split Up',  
 'Vestal Goodman, 74, Matriarch Of the World of Gospel Music',  
 "Democrats' Plan for Early Nominee May Be Costly",  
 'Corrections',  
 'QUOTATION OF THE DAY',  
 'Key Rates',  
 'Paid Notice: Deaths HEARD, DRAYTON JR.',  
 'In Rout, Islanders Win Sixth In a Row',  
 'Two Flights to U.S. Canceled in Wake of Terror Concerns',  
 'World Business Briefing | Europe: Britain: Bank Sells Unit',  
 'Fire Deaths In New York Jumped 23% Last Year',  
 'Pulp Fiction by Women With Protofeminist Roots',  
 'Yonkers City Council Raises Eyebrows by Giving Raises to Itself and Mayor',  
 'A Storm Of Patterns To Stencil',  
 'Paid Notice: Deaths GUTIERREZ, GERALD',  
 'Where Are the Tips?',  
 "Iran's Earthquake",  
 'Corrections',  
 'Afghanistan\x92s Constitutional Council Adjourns in Disarray',  
 'Play Button; A Crowded Bandwagon Yields Music Without Worries',  
 'Face on a Milk Carton? Amateur Sleuths Dig Deeper',  
 'Rebuilding a Life Shattered by War',  
 'Bank Money Accounts',  
 'Paid Notice: Deaths DWORMAN, MENACHEM',  
 'Mad Cow Disease: Lessons Unlearned',  
 'Resolve to Have Your Best Year on Skis',  
 'Documents Show U.S. Considered Using Force During Oil Embargo',  
 'TRANSACTIONS',  
 'Lenders Seek Debt Deferral For Korean Credit Card Issuer',  
 'Killed in Iraq',  
 'Florida State and Miami Know Just What to Expect',  
 'Large File Of Evidence Is Available In Leak Case',  
 'Paid Notice: Deaths TULLY, MARY JEAN',  
 'Dean and Kerry Issue Fund-Raising Totals',  
 'Division II Holds Arms Out for a Fallen Star',  
 'World Business Briefing | Asia: Singapore: Telephone Stake Sold',  
 'Clutter Combat: Containing the Enemy',  
 'U.S. Hunts Terror Clues in Case of 2 Brothers',  
 'Paid Notice: Memorials BARISON, DAVID ANDREW',  
 'College Football | Kickoff',  
 'Baghdad Bomb Kills at Least 5 At a Restaurant',  
 'Corrections',  
 'Corrections',  
 'EL PASO SELLS INTERESTS IN ARKANSAS AND THE PHILIPPINES',  
 'Corrections',  
 'NEWS SUMMARY',  
 'Police Officer Killed by Train At Fire Scene',  
 'Israel Plans 25% Expansion Of Its Settlements on Golan',  
 'Rebuilding a Life Shattered by War',  
 'Paid Notice: Deaths FRIEDMAN, LEE',  
 'Pakistan\x92s President Wins Key Vote of Confidence',  
 'Corrections',  
 "New Year's Day",  
 'Cuban Master Mines Local Talent',  
 'The Very Rich, It Now Appears, Give Their Share And Even More',  
 'Assistance for the Tongue-Tied In New Reference Software',  
 'INSIDE',  
 'No Groveling, Please, for Doorman to the Stars',  
 'Swift Response to News of Mad Cow, Years in the Making',  
 'Protesters Demand Free Elections in Hong Kong',  
 'Mad Cow Disease: Lessons Unlearned',  
 "For the Quake's Orphans, the Ordeal Has Only Begun",  
 'Shaky Past but Steady Confidence for the Colts',  
 'As Coach Dies Of Meningitis, People Seek Antibiotics',  
 'Paid Notice: Deaths SHAW, ROSALYN',  
 'From Vinyl to Digital, Hold the Crackle',  
 "Calling's Retro Touch",  
 'Drip, Drip, Zap: Electrical Current From Flowing Water',  
 'Paid Notice: Deaths SHANKMAN, STEVEN',  
 'CALENDAR',  
 'LAMSON & SESSIONS CUTS PROFIT FORECAST IN PRICE WAR',  
 'Opening Word Files On a PC With Linux',  
 'Merger Renaissance A Good Possibility In the Coming Year',  
 'Paid Notice: Deaths CONNOLLY, THOMAS HENRY',  
 'Thousands Rally in Hong Kong to Demand Free Elections',  
 'Rebuilding a Life Shattered by War',  
 'Paid Notice: Memorials SILVA, FABIO',  
 "Predecessor's Demonstration Of How to Play the Game Well",  
 'Paid Notice: Deaths SCHLOSS, ELIZABETH (LILY) VAN DAELEN',  
 'The Old Block of Ice And a Few New Chips',  
 'Paid Notice: Deaths FULLER, HELEN',  
 'Two Former City Detectives Discuss Pleas in Theft of Drug Cash',  
 'LeWitt the Collector, Filling Up a Warehouse',  
 'Handicapping the Race Ahead',  
 'River View, Mountain High, Price Low',  
 'A Continent Divided by Water, Now United by Air',  
 'Sports of The Times; In Spite of the B.C.S., Ambiguity Survives',  
 'Coaches Receive Both Big Salaries And Big Questions',  
 'Paid Notice: Deaths FRIEDMAN, MINNIE',  
 'Need Sounding Board? Cue the Bellyman; Finally, a Steinway Makes Some Music',  
 'The Rose Has Found New Strength in Its Roots',  
 'Paid Notice: Deaths SMERLING, SYLVIA',  
 'Paid Notice: Deaths BUDASOFF, DR. ISRAEL',  
 'To Soften the Din of Traffic, Walls of Lizards and Teepees',  
 'Bringing the Historical Confucius to Life; A Show in Paris Traces the Origins of the Sage, Who Has Influenced Countless Millions for 2,500 Years',  
 'A Knobless Solution',  
 'Letters to the Editor',  
 '9 Cows Linked to Disease Investigation Have Been Found',  
 'A West Side Stadium?',  
 'Paid Notice: Deaths SELIGMAN, AARON',  
 'ARTS BRIEFING',  
 'Young Florida Killer Accepts a Plea Deal',  
 'BUSINESS DIGEST',  
 'Ethnic Morass Bogs Down Afghan Talks On Charter',  
 'Paid Notice: Deaths SPRINTZ, MAE',  
 'A Time for Finesse: Marketing Beef After a Mad Cow Discovery',  
 'HealthSouth Trial Reset',  
 'For the Discerning Eye, Crisper Images on the Screen',  
 'David Bale, 62, Activist and Businessman',  
 'Rebuilding a Life Shattered by War',  
 'Poti Works Overtimes to Get Milestone',  
 'Punk For Posterity',  
 "Metro Briefing | Connecticut: Hartford: Boy's Abuse Had Been Reported",  
 'National Briefing | Plains: Oklahoma: Murder Trial In Oklahoma City Bombing Set For March',  
 'Paid Notice: Memorials SCHEINMAN, ALAN',  
 'A Party in Times Square',  
 'Shopping List: Snow Days',  
 'U.S.C. Wins Rose Bowl']

model\_id = "distilbert-base-uncased-finetuned-sst-2-english"  
sentiment\_pipeline = pipeline("sentiment-analysis", model = model\_id)  
  
def find\_sentiment(text):  
 sent = sentiment\_pipeline(text)[0]  
 score = sent["score"]  
 if sent["label"] == "NEGATIVE":  
 score \*= -1  
 return score

{"model\_id":"69f009049d764166bde03614ab277b12","version\_major":2,"version\_minor":0}

C:\Users\jimiolo\AppData\Local\Programs\Python\Python39\lib\site-packages\huggingface\_hub\file\_download.py:137: UserWarning: `huggingface\_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\jimiolo\.cache\huggingface\hub. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF\_HUB\_DISABLE\_SYMLINKS\_WARNING` environment variable. For more details, see https://huggingface.co/docs/huggingface\_hub/how-to-cache#limitations.  
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to see activate developer mode, see this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development  
 warnings.warn(message)

{"model\_id":"3c951293721e4fe482647004eb7d646f","version\_major":2,"version\_minor":0}

{"model\_id":"e0dd7087112f4360a0c64cbcee1523fc","version\_major":2,"version\_minor":0}

nyt\_hl

headline \  
date   
2004-01-01 00:00:00-05:00 [FreeMarkets Agrees to Buy Auction Unit of Cov...   
2004-01-02 00:00:00-05:00 [Homeless in America, Yes, Today, Tony Cragg, ...   
2004-01-03 00:00:00-05:00 [AIMING TO MAKE FASTER CHIPS, AGERE ACQUIRES T...   
2004-01-04 00:00:00-05:00 [Wanted: Young Delegates, Corrections, How to ...   
2004-01-05 00:00:00-05:00 [Cowboys Preparing To Take Next Step, Paid Not...   
... ...   
2023-08-27 00:00:00-04:00 [Flashback: Your Weekly History Quiz, August 2...   
2023-08-28 00:00:00-04:00 [Man Accused of Injecting Substance Into Neigh...   
2023-08-29 00:00:00-04:00 [President of Powerful Realtors’ Group Resigns...   
2023-08-30 00:00:00-04:00 [Last Defendant in Trump Election Interference...   
2023-08-31 00:00:00-04:00 [Man Gets 2 Years in Prison for Spending Pande...   
  
 art\_amount word\_count   
date   
2004-01-01 00:00:00-05:00 224 1529   
2004-01-02 00:00:00-05:00 70 433   
2004-01-03 00:00:00-05:00 158 1000   
2004-01-04 00:00:00-05:00 516 3001   
2004-01-05 00:00:00-05:00 182 1244   
... ... ...   
2023-08-27 00:00:00-04:00 63 548   
2023-08-28 00:00:00-04:00 120 1142   
2023-08-29 00:00:00-04:00 135 1271   
2023-08-30 00:00:00-04:00 129 1194   
2023-08-31 00:00:00-04:00 92 808   
  
[7182 rows x 3 columns]

nyt\_hl['sentiments'] = np.empty((len(nyt\_hl), 0)).tolist()  
nyt\_hl['total\_sent'] = 0  
nyt\_hl["sentiment"] = 0  
nyt\_hl["neg\_sentiment"] = 0  
for idx, headlines in enumerate (nyt\_hl['headline'][0:1]):   
 total\_sent = 0  
 for headline in headlines:  
 sent\_result = find\_sentiment(headline)  
 total\_sent += sent\_result  
 nyt\_hl['sentiments'].iloc[idx].append(sent\_result)  
 nyt\_hl['total\_sent'].iloc[idx] = total\_sent  
 nyt\_hl["sentiment"].iloc[idx] = mean(nyt\_hl['sentiments'][idx])  
 nyt\_hl["neg\_sentiment"].iloc[idx] = len([s for s in nyt\_hl['sentiments'][idx] if s < 0]) / len(nyt\_hl['sentiments'][idx])  
   
 #del nyt\_hl['sentiments'][idx]

C:\Users\jimiolo\AppData\Local\Temp\ipykernel\_30968\2557955322.py:11: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
 nyt\_hl['total\_sent'].iloc[idx] = total\_sent  
C:\Users\jimiolo\AppData\Local\Temp\ipykernel\_30968\2557955322.py:12: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
 nyt\_hl["sentiment"].iloc[idx] = mean(nyt\_hl['sentiments'][idx])  
C:\Users\jimiolo\AppData\Local\Temp\ipykernel\_30968\2557955322.py:13: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
 nyt\_hl["neg\_sentiment"].iloc[idx] = len([s for s in nyt\_hl['sentiments'][idx] if s < 0]) / len(nyt\_hl['sentiments'][idx])

nyt\_hl  
# Te obliczenia raz zrobić i zapisać je gdzieś. Później gotowe wczytywać by było szybciej.

headline \  
date   
2004-01-01 00:00:00-05:00 [FreeMarkets Agrees to Buy Auction Unit of Cov...   
2004-01-02 00:00:00-05:00 [Homeless in America, Yes, Today, Tony Cragg, ...   
2004-01-03 00:00:00-05:00 [AIMING TO MAKE FASTER CHIPS, AGERE ACQUIRES T...   
2004-01-04 00:00:00-05:00 [Wanted: Young Delegates, Corrections, How to ...   
2004-01-05 00:00:00-05:00 [Cowboys Preparing To Take Next Step, Paid Not...   
... ...   
2023-08-27 00:00:00-04:00 [Flashback: Your Weekly History Quiz, August 2...   
2023-08-28 00:00:00-04:00 [Man Accused of Injecting Substance Into Neigh...   
2023-08-29 00:00:00-04:00 [President of Powerful Realtors’ Group Resigns...   
2023-08-30 00:00:00-04:00 [Last Defendant in Trump Election Interference...   
2023-08-31 00:00:00-04:00 [Man Gets 2 Years in Prison for Spending Pande...   
  
 art\_amount word\_count \  
date   
2004-01-01 00:00:00-05:00 224 1529   
2004-01-02 00:00:00-05:00 70 433   
2004-01-03 00:00:00-05:00 158 1000   
2004-01-04 00:00:00-05:00 516 3001   
2004-01-05 00:00:00-05:00 182 1244   
... ... ...   
2023-08-27 00:00:00-04:00 63 548   
2023-08-28 00:00:00-04:00 120 1142   
2023-08-29 00:00:00-04:00 135 1271   
2023-08-30 00:00:00-04:00 129 1194   
2023-08-31 00:00:00-04:00 92 808   
  
 sentiments \  
date   
2004-01-01 00:00:00-05:00 [-0.605070948600769, -0.9997730851173401, 0.98...   
2004-01-02 00:00:00-05:00 []   
2004-01-03 00:00:00-05:00 []   
2004-01-04 00:00:00-05:00 []   
2004-01-05 00:00:00-05:00 []   
... ...   
2023-08-27 00:00:00-04:00 []   
2023-08-28 00:00:00-04:00 []   
2023-08-29 00:00:00-04:00 []   
2023-08-30 00:00:00-04:00 []   
2023-08-31 00:00:00-04:00 []   
  
 total\_sent sentiment neg\_sentiment   
date   
2004-01-01 00:00:00-05:00 -22.091328 -0.098622 0.553571   
2004-01-02 00:00:00-05:00 0.000000 0.000000 0.000000   
2004-01-03 00:00:00-05:00 0.000000 0.000000 0.000000   
2004-01-04 00:00:00-05:00 0.000000 0.000000 0.000000   
2004-01-05 00:00:00-05:00 0.000000 0.000000 0.000000   
... ... ... ...   
2023-08-27 00:00:00-04:00 0.000000 0.000000 0.000000   
2023-08-28 00:00:00-04:00 0.000000 0.000000 0.000000   
2023-08-29 00:00:00-04:00 0.000000 0.000000 0.000000   
2023-08-30 00:00:00-04:00 0.000000 0.000000 0.000000   
2023-08-31 00:00:00-04:00 0.000000 0.000000 0.000000   
  
[7182 rows x 7 columns]

To raczej powinno być przeniesione do stworzenia modeli, bo tutaj są dane, ale można przygotować te dane, by były gotowe do użycia.

# crete new data frame to NLP process headlines for using further for investing prediction  
nlp\_data = pd.merge(sp500['target'], nyt\_hl['headline'], on='date')  
nlp\_data

target \  
date   
2004-01-02 00:00:00-05:00 1   
2004-01-05 00:00:00-05:00 1   
2004-01-06 00:00:00-05:00 1   
2004-01-07 00:00:00-05:00 1   
2004-01-08 00:00:00-05:00 0   
... ...   
2023-08-24 00:00:00-04:00 1   
2023-08-25 00:00:00-04:00 1   
2023-08-28 00:00:00-04:00 1   
2023-08-29 00:00:00-04:00 1   
2023-08-30 00:00:00-04:00 0   
  
 headline   
date   
2004-01-02 00:00:00-05:00 [Homeless in America, Yes, Today, Tony Cragg, ...   
2004-01-05 00:00:00-05:00 [Cowboys Preparing To Take Next Step, Paid Not...   
2004-01-06 00:00:00-05:00 [Paid Notice: Deaths LUSK, RUTH, Metro Brief...   
2004-01-07 00:00:00-05:00 [Rice and Ice Cream: An Oddly Appealing Couple...   
2004-01-08 00:00:00-05:00 [Sued by Estate Of Ex-Beatle, Doctor Offers Ac...   
... ...   
2023-08-24 00:00:00-04:00 [Terry Funk, Hard-Core Hall of Fame Wrestler, ...   
2023-08-25 00:00:00-04:00 [Spin. Pilates. Prigozhin? Presidential Busine...   
2023-08-28 00:00:00-04:00 [Man Accused of Injecting Substance Into Neigh...   
2023-08-29 00:00:00-04:00 [President of Powerful Realtors’ Group Resigns...   
2023-08-30 00:00:00-04:00 [Last Defendant in Trump Election Interference...   
  
[4948 rows x 2 columns]

# calculate the index for the split point  
split\_point = int(0.8 \* len(nlp\_data)) # 80% for training, 20% for testing  
split\_point

3958

nlp\_data.reset\_index(drop = True, inplace = True)  
# split the data into training and testing sets  
train\_data = nlp\_data.iloc[:split\_point]  
test\_data = nlp\_data.iloc[split\_point:]  
test\_data.reset\_index(drop = True, inplace = True)

train\_data

target headline  
0 1 [Homeless in America, Yes, Today, Tony Cragg, ...  
1 1 [Cowboys Preparing To Take Next Step, Paid Not...  
2 1 [Paid Notice: Deaths LUSK, RUTH, Metro Brief...  
3 1 [Rice and Ice Cream: An Oddly Appealing Couple...  
4 0 [Sued by Estate Of Ex-Beatle, Doctor Offers Ac...  
... ... ...  
3953 1 [After Tight Israeli Election, Netanyahu’s Ten...  
3954 0 [We’re Not the Saudis’ Mercenaries, As Trump W...  
3955 0 [Deportation Exemptions to Resume for Immigran...  
3956 0 [Don’t Mess With Modi in Texas, Emmy Winners: ...  
3957 1 [The Math Says the Mets Are Alive. Common Sens...  
  
[3958 rows x 2 columns]

test\_data

target headline  
0 0 [Group Behind California Privacy Law Aims to S...  
1 0 [Person of Interest Is Named in Killing of Tra...  
2 1 [Just How Corrupt Is Bill Barr?, Ryan Murphy, ...  
3 0 [Hong Kong’s Status as Neutral Ground at Risk ...  
4 0 [Trump’s Claims About Biden Aren’t ‘Unsupporte...  
.. ... ...  
985 1 [Terry Funk, Hard-Core Hall of Fame Wrestler, ...  
986 1 [Spin. Pilates. Prigozhin? Presidential Busine...  
987 1 [Man Accused of Injecting Substance Into Neigh...  
988 1 [President of Powerful Realtors’ Group Resigns...  
989 0 [Last Defendant in Trump Election Interference...  
  
[990 rows x 2 columns]

# splitting the dataset  
y\_train = train\_data['target']  
train = train\_data['headline']  
y\_test = test\_data['target']  
test = test\_data['headline']

train.info()  
train[0]

<class 'pandas.core.series.Series'>  
RangeIndex: 3958 entries, 0 to 3957  
Series name: headline  
Non-Null Count Dtype   
-------------- -----   
3958 non-null object  
dtypes: object(1)  
memory usage: 31.0+ KB

['Homeless in America, Yes, Today',  
 'Tony Cragg',  
 "'Home Sewn' -- 'Three Centuries of Stitching History'",  
 'Cabbies Fear Fare Increase Will Hurt More Than It Helps',  
 'Sports Briefing',  
 '36 Hours | Ottawa',  
 'Homeless in America, Yes, Today',  
 'Paid Notice: Deaths KELLY, JOSEPHINE (NEE MULLANE)',  
 'Corrections',  
 "A New Year's Tradition Lives, But the 4-Legged Star Doesn't",  
 'When Steuben Glass Went Clear, Clean and Modern',  
 "'108'",  
 'Homeless in America, Yes, Today',  
 'How Wild Was the Party? Ask the Army With the Brooms',  
 '\x92Home Sewn\x92; Joe Deal; Don Celender',  
 'POP AND JAZZ GUIDE',  
 'Condos in Ski Areas: Your Own Place After a Day on the Slopes',  
 'Egyptian Urges Arafat to Subdue Factions',  
 "In Iraq's Murky Battle, Snipers Offer U.S. a Precision Weapon",  
 "Some School Districts Challenge Bush's Signature Education Law",  
 "Holiday Movies' Somber Embrace",  
 'Paid Notice: Deaths SMERLING, SYLVIA',  
 'Killed in Iraq',  
 'Residential Real Estate; Developer Sees Dream Farm; Others See Ruin of Wetlands',  
 'Bush Says Iran Must Do More Before Relations Can Improve',  
 'Markets & Investing; China Demand and Weak Dollar May Support High Oil Price',  
 'British Flight Called Off After \x92Security Concerns\x92',  
 'Staying in School',  
 'The Confessions of Pete Rose',  
 "'Shop Windows'",  
 'Suspected Chinese SARS Case Baffles Experts',  
 'After Halting Start, Clark Seems to Be Finding Legs',  
 'Paid Notice: Deaths GREENBERG, RUTH OSTROFF',  
 'Justice Takes Its Time',  
 'Corrections',  
 "Peter Pan's Flying Adventures, Now Grounded in Literary Roots",  
 'Beyond Singles and Concept Albums, Pop Yearns for a Long Form',  
 'Markets & Investing; Industrial Commodities, Gold Climb',  
 'U.S. Overture to Iran',  
 "'Ndebele'",  
 'Woman and Bedridden Son Die in House Fire',  
 'Corrections',  
 'Officials Destroying Calves, Including That of Sick Cow',  
 "Editors' Note; SPECIAL TODAY -- Outlook",  
 'Paid Notice: Deaths MURRAY, A. BREAN',  
 'The Inside Art Column on Hold',  
 'What a Memorial Needs',  
 "Paid Notice: Deaths DAVIS, ELLSWORTH ''BINKY''",  
 'Paid Notice: Deaths SHANKMAN, STEVEN',  
 'Paid Notice: Deaths THURNAUER, WARREN, SR',  
 'Books as Art Objects (Reading Is Optional)',  
 'The Rush-Hour Revelations Of an Underground Museum',  
 'Knicks Try To Remain Focused',  
 'Hollywood and The Civil War',  
 'Markets & Investing; Global Investors Expect Still More Gains',  
 "Parody Can Become a 'Fledermaus' Trap",  
 'Markets & Investing; The Bounceback Year',  
 "Assessing Mayor Bloomberg's Year",  
 'Paid Notice: Deaths ANGELL, MARY ANNE K. (NEE KENNEDY)',  
 'Corrections',  
 'Markets & Investing; On the Street, Optimism Prevails for Stocks Through 2004',  
 'Paid Notice: Deaths MCMANUS, ELIZABETH',  
 'Paid Notice: Deaths ROSCOE, JEROME VERITY',  
 "Britain Says U.S. Planned To Seize Oil In '73 Crisis",  
 'Editorial Observer; A Community of Ex-Cons Shows How to Bring Prisoners Back Into Society',  
 'Quick Escapes',  
 'As Others Struggle',  
 'Markets & Investing; Measuring The Market',  
 'TRANSACTIONS',  
 'A Lonely Voice in China Is Critical on Rights and Reform']

train = train.apply(lambda x: '. '.join(x))  
test = test.apply(lambda x: '. '.join(x))

train[0]

"Homeless in America, Yes, Today. Tony Cragg. 'Home Sewn' -- 'Three Centuries of Stitching History'. Cabbies Fear Fare Increase Will Hurt More Than It Helps. Sports Briefing. 36 Hours | Ottawa. Homeless in America, Yes, Today. Paid Notice: Deaths KELLY, JOSEPHINE (NEE MULLANE). Corrections. A New Year's Tradition Lives, But the 4-Legged Star Doesn't. When Steuben Glass Went Clear, Clean and Modern. '108'. Homeless in America, Yes, Today. How Wild Was the Party? Ask the Army With the Brooms. \x92Home Sewn\x92; Joe Deal; Don Celender. POP AND JAZZ GUIDE. Condos in Ski Areas: Your Own Place After a Day on the Slopes. Egyptian Urges Arafat to Subdue Factions. In Iraq's Murky Battle, Snipers Offer U.S. a Precision Weapon. Some School Districts Challenge Bush's Signature Education Law. Holiday Movies' Somber Embrace. Paid Notice: Deaths SMERLING, SYLVIA. Killed in Iraq. Residential Real Estate; Developer Sees Dream Farm; Others See Ruin of Wetlands. Bush Says Iran Must Do More Before Relations Can Improve. Markets & Investing; China Demand and Weak Dollar May Support High Oil Price. British Flight Called Off After \x92Security Concerns\x92. Staying in School. The Confessions of Pete Rose. 'Shop Windows'. Suspected Chinese SARS Case Baffles Experts. After Halting Start, Clark Seems to Be Finding Legs. Paid Notice: Deaths GREENBERG, RUTH OSTROFF. Justice Takes Its Time. Corrections. Peter Pan's Flying Adventures, Now Grounded in Literary Roots. Beyond Singles and Concept Albums, Pop Yearns for a Long Form. Markets & Investing; Industrial Commodities, Gold Climb. U.S. Overture to Iran. 'Ndebele'. Woman and Bedridden Son Die in House Fire. Corrections. Officials Destroying Calves, Including That of Sick Cow. Editors' Note; SPECIAL TODAY -- Outlook. Paid Notice: Deaths MURRAY, A. BREAN. The Inside Art Column on Hold. What a Memorial Needs. Paid Notice: Deaths DAVIS, ELLSWORTH ''BINKY''. Paid Notice: Deaths SHANKMAN, STEVEN. Paid Notice: Deaths THURNAUER, WARREN, SR. Books as Art Objects (Reading Is Optional). The Rush-Hour Revelations Of an Underground Museum. Knicks Try To Remain Focused. Hollywood and The Civil War. Markets & Investing; Global Investors Expect Still More Gains. Parody Can Become a 'Fledermaus' Trap. Markets & Investing; The Bounceback Year. Assessing Mayor Bloomberg's Year. Paid Notice: Deaths ANGELL, MARY ANNE K. (NEE KENNEDY). Corrections. Markets & Investing; On the Street, Optimism Prevails for Stocks Through 2004. Paid Notice: Deaths MCMANUS, ELIZABETH. Paid Notice: Deaths ROSCOE, JEROME VERITY. Britain Says U.S. Planned To Seize Oil In '73 Crisis. Editorial Observer; A Community of Ex-Cons Shows How to Bring Prisoners Back Into Society. Quick Escapes. As Others Struggle. Markets & Investing; Measuring The Market. TRANSACTIONS. A Lonely Voice in China Is Critical on Rights and Reform"

# removing punctuation and special character from the text  
train.replace(to\_replace='[^a-zA-Z]', value=' ', regex=True, inplace=True)  
test.replace(to\_replace='[^a-zA-Z]', value=' ', regex=True, inplace=True)

train[0]

'Homeless in America Yes Today Tony Cragg Home Sewn Three Centuries of Stitching History Cabbies Fear Fare Increase Will Hurt More Than It Helps Sports Briefing Hours Ottawa Homeless in America Yes Today Paid Notice Deaths KELLY JOSEPHINE NEE MULLANE Corrections A New Year s Tradition Lives But the Legged Star Doesn t When Steuben Glass Went Clear Clean and Modern Homeless in America Yes Today How Wild Was the Party Ask the Army With the Brooms Home Sewn Joe Deal Don Celender POP AND JAZZ GUIDE Condos in Ski Areas Your Own Place After a Day on the Slopes Egyptian Urges Arafat to Subdue Factions In Iraq s Murky Battle Snipers Offer U S a Precision Weapon Some School Districts Challenge Bush s Signature Education Law Holiday Movies Somber Embrace Paid Notice Deaths SMERLING SYLVIA Killed in Iraq Residential Real Estate Developer Sees Dream Farm Others See Ruin of Wetlands Bush Says Iran Must Do More Before Relations Can Improve Markets Investing China Demand and Weak Dollar May Support High Oil Price British Flight Called Off After Security Concerns Staying in School The Confessions of Pete Rose Shop Windows Suspected Chinese SARS Case Baffles Experts After Halting Start Clark Seems to Be Finding Legs Paid Notice Deaths GREENBERG RUTH OSTROFF Justice Takes Its Time Corrections Peter Pan s Flying Adventures Now Grounded in Literary Roots Beyond Singles and Concept Albums Pop Yearns for a Long Form Markets Investing Industrial Commodities Gold Climb U S Overture to Iran Ndebele Woman and Bedridden Son Die in House Fire Corrections Officials Destroying Calves Including That of Sick Cow Editors Note SPECIAL TODAY Outlook Paid Notice Deaths MURRAY A BREAN The Inside Art Column on Hold What a Memorial Needs Paid Notice Deaths DAVIS ELLSWORTH BINKY Paid Notice Deaths SHANKMAN STEVEN Paid Notice Deaths THURNAUER WARREN SR Books as Art Objects Reading Is Optional The Rush Hour Revelations Of an Underground Museum Knicks Try To Remain Focused Hollywood and The Civil War Markets Investing Global Investors Expect Still More Gains Parody Can Become a Fledermaus Trap Markets Investing The Bounceback Year Assessing Mayor Bloomberg s Year Paid Notice Deaths ANGELL MARY ANNE K NEE KENNEDY Corrections Markets Investing On the Street Optimism Prevails for Stocks Through Paid Notice Deaths MCMANUS ELIZABETH Paid Notice Deaths ROSCOE JEROME VERITY Britain Says U S Planned To Seize Oil In Crisis Editorial Observer A Community of Ex Cons Shows How to Bring Prisoners Back Into Society Quick Escapes As Others Struggle Markets Investing Measuring The Market TRANSACTIONS A Lonely Voice in China Is Critical on Rights and Reform'

# converting the entire text to lower case  
train = train.str.lower()  
test = test.str.lower()

# creating corpus of train dataset  
ps = PorterStemmer()  
train\_corpus = []  
  
for i in range(0, len(train)):  
   
 # tokenizing the news-title by words  
 words = train[i].split()  
  
 # removing the stopwords  
 words = [word for word in words if word not in set(stopwords.words('english'))]  
  
 # stemming the words  
 words = [ps.stem(word) for word in words]  
  
 # joining the stemmed words  
 headline = ' '.join(words)  
  
 # building a corpus of news-title  
 train\_corpus.append(headline)

train\_corpus[0:1]

# creating corpus of test dataset  
test\_corpus = []  
  
for i in range(0, len(test)):  
   
 # tokenizing the news-title by words  
 words = test[i].split()  
  
 # removing the stopwords  
 words = [word for word in words if word not in set(stopwords.words('english'))]  
  
 # stemming the words  
 words = [ps.stem(word) for word in words]  
  
 # joining the stemmed words  
 headline = ' '.join(words)  
  
 # building a corpus of news-title  
 test\_corpus.append(headline)

test\_corpus[0:1]

down\_words = []  
for i in list(y\_train[y\_train==0].index):  
 down\_words.append(train\_corpus[i])  
  
up\_words = []  
for i in list(y\_train[y\_train==1].index):  
 up\_words.append(train\_corpus[i])

down\_words\_all = " ".join(down\_words)  
up\_words\_all = " ".join(up\_words)

# creating wordcloud for down\_words  
wordcloud\_down = WordCloud(background\_color='white', width=3000, height=2500).generate(down\_words\_all)  
plt.figure(figsize=(8,8))  
plt.imshow(wordcloud\_down)  
plt.axis('off')  
plt.title("Words which indicate a fall in S&P 500")  
plt.show()

# creating wordcloud for up\_words  
wordcloud\_up = WordCloud(background\_color='white', width=3000, height=2500).generate(up\_words\_all)  
plt.figure(figsize=(8,8))  
plt.imshow(wordcloud\_up)  
plt.axis('off')  
plt.title("Words which indicate a rise in S&P 500")  
plt.show()

# creating the Bag of Words model  
cv = CountVectorizer(max\_features=10000, ngram\_range=(2,2))  
X\_train = cv.fit\_transform(train\_corpus).toarray()

#test model to append to others features  
X\_train

X\_test = cv.transform(test\_corpus).toarray()

X\_test

**macro\_data - sp500\_rates**

# to delete later copy to retrieve data if something went wrong  
sp500\_rates\_copy = sp500\_rates.copy()  
sp500\_rates\_copy

# plot all data  
sp500\_rates\_plot = sp500\_rates.plot.line( use\_index = True)  
sp500\_rates\_plot.set\_ylabel("All rates for S&P 500")  
sp500\_rates\_plot.set\_title('S&P 500 Rates')  
sp500\_rates\_plot.set\_ylim(-10, 50)

# calculate the correlation matrix  
correlation\_matrix = sp500\_rates.corr()  
plt.figure(figsize=(8, 6))  
sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)  
plt.title('Correlation Heatmap')  
plt.show()

Na matrycy korelacji można zauważyć, że współczynniki P/E oraz Earning Yield są przeciwstawne do siebie, co jest zgodne ze sposobem ich obliczania. Widać także dużą pozytywną korelację pomiędzy współczynnikami CAPE, P/B, P/S, oraz dużą ujemną korelację pomiędzy tymi wskaźnikami a Dividend Yield.

Ogólnie można zauważyć, że dane nie są bardzo skorelowane ze sobą, co jest korzystne do wykorzystania danych w modelach uczenia maszynowego. Niskie lub brak skorelowania między cechami może pomóc w zrozumieniu, jakie cechy wpływają na predykcje modelu Warto jednak eksperymentować z różnymi użytymi danymi i monitorować wydajność modelu oraz interpretować wyniki, aby dostosować podejście do optymalnego rozwiązania problemu.

**macro\_data - usa\_eco\_rates**

# to delete later copy to retrieve data if something went wrong  
usa\_eco\_rates\_copy = usa\_eco\_rates.copy()  
usa\_eco\_rates\_copy

# plot all data without GDP  
usa\_eco\_rates\_wo\_gdp = usa\_eco\_rates.copy()  
del usa\_eco\_rates\_wo\_gdp['gdp']  
usa\_eco\_rates\_plot = usa\_eco\_rates\_wo\_gdp.plot.line( use\_index = True)  
usa\_eco\_rates\_plot.set\_ylabel("All rates for USA market without GDP")  
usa\_eco\_rates\_plot.set\_title('USA Economy Rates')

# calculate the correlation matrix  
correlation\_matrix = usa\_eco\_rates.corr()  
plt.figure(figsize=(8, 6))  
sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)  
plt.title('Correlation Heatmap')  
plt.show()

Dane dla amerykańskiej gospodarki nie są bardzo skorelowane ze sobą. Można wyróżnić pozytywną korelację pomiędzy 10 US Bond Yield a EFFR, co jest zgodne z logiką, że rentowności obligacji skarbowych dążą do wartości efektywnych stóp procentowych. Drugą parą o dodatniej korelacji jest PMI Index i GDP Percent Change. Widać tutaj powiązanie, wzrostu PKB USA w danym okresie jest skorelowane z pozytywnym nastawieniem managerów do nabywania różnego rodzaju dóbr i usług.

Największa ujemną korelację jest dla pary EFFR, Unemployment Rate. Co może być związane z polityką banków centralnych, którzy odpowiednio sterują stopami procentowymi i częstym kryterium do przyjęcia odpowiedniej strategii jest wartość bezrobocia w kraju. Drugą parą są wskaźniki GDP oraz 10 US Bond Yield. Co może być powiązane szczególnie z czasami kryzysu, w którym to PKB kraju często maleje, a inwestorzy inwestują w bezpieczniejsze aktywa jak obligacje.

**goog\_trend**

# load the dataset  
goog\_trend = pd.read\_csv('/kaggle/input/s-and-p500-google-trends/SP500\_GoogleTrends.csv', index\_col='date')  
goog\_trend.index = pd.to\_datetime(goog\_trend.index)  
goog\_trend.index = goog\_trend.index.tz\_localize('America/New\_York')  
goog\_trend

# calculate the difference between the current month and the previous month  
goog\_trend['month\_diff'] = goog\_trend['month\_trend'].diff()  
goog\_trend

# plot trend  
#goog\_trend\_plot = goog\_trend['month\_diff'].plot.line( use\_index = True)  
goog\_trend\_plot = goog\_trend.plot.line( use\_index = True)  
goog\_trend\_plot.set\_ylabel("Month change in Trend")  
goog\_trend\_plot.set\_title('S&P 500 Changes in trend')

Należy rozważyć, w projekcie czy warto używać wartości bezwzględnych, czy wartości względnych (zmian względem poprzedniego okresu). Oba podejścia mogą być cenne w różnych sytuacjach, dlatego należy rozważyć kilka kwestii, kiedy używać każdego z tych podejść. Oto krótkie charakterystyki podejść:

**1. Wartości Bezwzględne:**

* **Stabilne Trendy:** Jeśli Twoje dane wykazują stałe i spójne trendy w czasie, używanie wartości bezwzględnych może być dobrym wyborem. Na przykład, jeśli przewidujesz ceny akcji, sama absolutna cena może być bardziej informatywna niż zmiana w stosunku do poprzedniego dnia.
* **Interpretowalność:** Wartości bezwzględne są często bardziej interpretowane, ponieważ reprezentują rzeczywiste pomiary lub ilości. Może to być istotne w przypadkach, gdy musisz wyjaśnić lub uzasadnić przewidywania przed interesariuszami.

**2. Wartości Względne (Zmiany):**

* **Sezonowość:** Gdy Twoje dane wykazują silne wzorce sezonowe lub cykliczne zachowanie, może być korzystne pracować z wartościami względnymi (zmianami). Na przykład w prognozowaniu sprzedaży może być bardziej informatywne modelowanie zmian miesiąc po miesiącu lub rok po roku niż same liczby sprzedaży.
* **Stacjonarność:** Niektóre algorytmy uczenia maszynowego zakładają, że dane są stacjonarne, co oznacza, że właściwości statystyczne nie zmieniają się w czasie. W takich przypadkach modelowanie różnic między okresami może pomóc uczynić dane bardziej stacjonarnymi.
* **Redukcja Szumu:** Modelowanie zmian może pomóc usunąć szum lub nieregularności w danych, co ułatwia modelowi skupienie się na ukrytych wzorcach.

**3. Łączenie obu podjeść:**

* W wielu przypadkach warto pracować zarówno z wartościami bezwzględnymi, jak i wartościami względnymi jako cechy. Na przykład możesz uwzględnić zarówno absolutną wartość zmiennej, jak i jej zmianę względem poprzedniego okresu jako osobne cechy w modelu. To może uwzględnić zarówno długoterminowe trendy, jak i krótkoterminową dynamikę w danych.

# calculate the correlation matrix  
correlation\_matrix = goog\_trend.corr()  
  
# create a heatmap  
plt.figure(figsize=(10, 8))  
sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f")  
plt.title("Correlation Heatmap")  
plt.show()

Można zauważyć, że jest nieduża korelacja pomiędzy absolutnymi wartościami trendów wyszukiwania a relatywnymi miesięcznymi zmianami dla trendu.

**analiza techniczna**

Może tutaj merge wszystkich danych i zrobienie korelacji i innych rzeczy na tym podstawie wybranie ostatecznych danych do modeli.

**nyt\_hl**

**final data**

# create column for prediction for next day - True if the index increases the next day, or False if it decreases  
sp500\_predict = sp500.copy()  
sp500\_predict["tomorrow"] = sp500\_predict["close"].shift(-1)  
sp500\_predict = sp500\_predict.dropna()  
sp500\_predict["target"] = (sp500\_predict["tomorrow"]> sp500\_predict["close"]).astype(int)  
sp500\_predict

# create data frame for storing all preprocessed data  
#final\_data = pd.DataFrame()  
# copy all needed data from sp500 data to final\_data  
final\_data = sp500\_predict[['target', 'volume', 'close']]  
final\_data

Wnioski

Przewidywanie cen akcji na giełdzie jest od dawna przedmiotem zainteresowania zarówno analityków, jak i badaczy. Ceny akcji są trudne do przewidzenia ze względu na ich dużą zmienność, która zależy od różnorodnych czynników politycznych i ekonomicznych, zmiany przywództwa, nastrojów inwestorów i wielu innych czynników. Przewidywanie cen akcji na podstawie samych danych historycznych lub informacji tekstowych okazało się niewystarczające.