

```
In [1]: import pandas as pd
pd.set_option('display.max_rows', 1000)
pd.set_option('display.max_columns', 100)
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import spacy
import regex as re
from collections import Counter
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from textblob import TextBlob

# Wczytanie pliku tekstowego z listą stop words
def load_stop_words(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        stop_words = [word.strip() for word in file.readlines()]
    return stop_words

# Ścieżka do pliku tekstowego z listą stop words
stop_words_file = 'Stopwords.txt'

# Wczytanie listy stop words z pliku
my_stop_words = load_stop_words(stop_words_file)

# Wczytanie danych z pliku Excel
df = pd.read_excel('Tweet Tracker.xlsx')
```

```
In [2]: # df.head(500)
```

```
In [3]: # # Pobieranie modelu języka polskiego w Spacy
# !python -m spacy download pl_core_news_md

# # Załadowanie modelu języka polskiego
nlp = spacy.load('pl_core_news_md')
```

```
In [4]: def clean_tweet(tweet):

    tweet = re.sub(r"http\S+|www\S+|https\S+", '', tweet, flags=re.MULTILINE)
    tweet = re.sub(r'\@w+|\#', '', tweet)
    tweet = re.sub(r'^\w\s', '', tweet)
    tweet = re.sub(r'\bRT\b', '', tweet)

    # Tokenizacja tekstu na słowa
    tokens = word_tokenize(tweet)
    # Usunięcie znaków interpunkcyjnych i przekształcenie słów na małe litery
    words = [word.lower() for word in tokens if word.isalpha()]
    # Usunięcie stop words
    words = [word for word in words if word not in my_stop_words]
    # Analiza części mowy dla słów
    lemmas = []
    doc = nlp(' '.join(words))
    for token in doc:
        # Wybieramy tylko rzeczowniki (NOUN), czasowniki (VERB) i przymiotniki (ADJ)
        if token.pos_ in ['NOUN', 'VERB', 'ADJ']:
            lemmas.append(token.lemma_)

    # Połączenie lematów w jednozdaniowy tekst
    cleaned_tweet = ' '.join(lemmas)
    return cleaned_tweet

# Stworzenie nowej kolumny z wyczyszczonymi tweetami
```

```
df['CLEANED_TEXT'] = df['TEXT'].apply(clean_tweet)
```

```
# Wyświetlenie wyczyszczonych tweetów  
# df.head(500)
```

```
In [5]: # Tworzenie listy słów z oczyszczonych tweetów  
all_words = ' '.join(df['CLEANED_TEXT']).split()  
  
# Obliczanie częstości słów  
word_freq = Counter(all_words)  
  
# Wyświetlanie najczęściej występujących słów  
for word, frequency in word_freq.most_common(100):  
    print(f'{word}: {frequency}')
```

```
socjal: 139  
pracować: 123  
zasiłek: 108  
biedny: 85  
chcieć: 85  
mieć: 84  
polski: 83  
rząd: 82  
imigrant: 81  
pracownik: 80  
program: 78  
rodzina: 77  
sejm: 76  
zatrudniać: 74  
wydać: 73  
państwo: 73  
bochenek: 72  
biuro: 72  
poselski: 72  
wyjaśnić: 72  
niedopatrzanie: 72  
człowiek: 71  
społeczny: 71  
praca: 62  
dziecko: 61  
ubóstwo: 59  
świadczenie: 58  
być: 56  
pobierać: 49  
płacić: 47  
minister: 47  
wiedzieć: 46  
dostawać: 45  
podatek: 45  
polityka: 43  
rok: 43  
kraj: 40  
pogrzebowy: 36  
francja: 35  
waloryzacja: 35  
nowy: 34  
likwidacja: 33  
projekt: 33  
ustawa: 33  
powód: 32  
ostatni: 31  
duży: 31  
posiadać: 29  
mówić: 28  
obywatel: 28
```

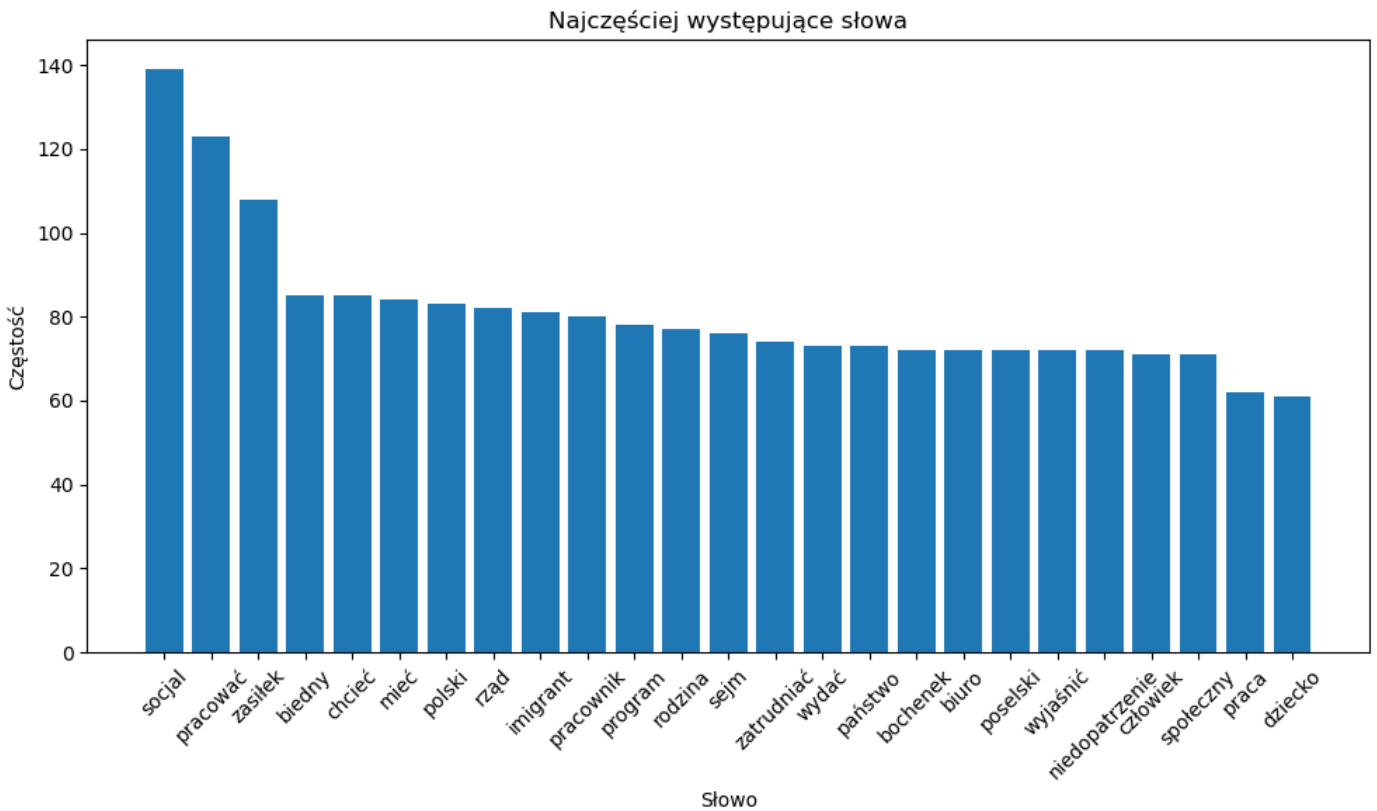
biedn: 28
rodzic: 28
sam: 27
wysokość: 27
deficyt: 27
świadczyć: 27
budżet: 27
głosowanie: 27
życie: 26
styczeń: 26
procedura: 26
nadmierny: 26
edp: 26
bezrobotny: 25
problem: 25
wypłacać: 24
móc: 24
podnieść: 24
intel: 24
uwaga: 23
chodzić: 22
pieniądz: 22
należeć: 21
wicemarszałek: 21
współpracownik: 21
asys: 21
pomoc: 21
zmiana: 21
eurokolchóz: 21
robić: 20
zrobić: 20
pamiętać: 20
sondapo: 20
szokujący: 20
wypowiedziach: 20
zaprzeczać: 20
sobieistniej: 20
ryzyko: 20
dobry: 19
propozycja: 19
prawica: 19
raz: 19
wystąpien: 19
by: 18
rządowy: 18
polak: 18
świadczec: 17
czas: 17
osoba: 17
inny: 17

```
In [6]: # Wybranie 25 najczęściej występujących słów
top_words = word_freq.most_common(25)

# Przygotowanie danych do wykresu
words = [word for word, _ in top_words]
frequencies = [freq for _, freq in top_words]

# Wygenerowanie wykresu słupkowego
plt.figure(figsize=(10, 6))
plt.bar(words, frequencies)
plt.xlabel('Słowo')
plt.ylabel('Częstość')
plt.title('Najczęściej występujące słowa')
plt.xticks(rotation=45)
plt.tight_layout()
```

```
# Wyświetlenie wykresu
plt.show()
```



```
In [24]: # Lista słów do usunięcia
words_to_remove = ['biedny', 'socjal', 'zasiłek', 'mieć', 'chcieć', "bochenek", "zatrudniać"]

# Usunięcie słów z word_freq
word_freq_cleaned = {word: freq for word, freq in word_freq.items() if word not in words_to_remove}

# Generowanie chmury słów z oczyszczonymi częstościami słów
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(word_freq_cleaned)

# Wyświetlanie chmury słów
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```


praca: 62
dziecko: 61
ubóstwo: 59
świadczenie: 58
być: 56
pobierać: 49
płacić: 47
minister: 47
wiedzieć: 46
dostawać: 45
podatek: 45
polityka: 43
rok: 43
kraj: 40
pogrzebowy: 36
francja: 35
waloryzacja: 35
nowy: 34
likwidacja: 33
projekt: 33
ustawa: 33
powód: 32
ostatni: 31
duży: 31
posiadać: 29
mówić: 28
obywatel: 28
biedn: 28
rodzic: 28
sam: 27
wysokość: 27
deficyt: 27
świadczyc: 27
budżet: 27
głosowanie: 27
życie: 26
styczeń: 26
procedura: 26
nadmierny: 26
edp: 26
bezrobotny: 25
problem: 25
wypłacać: 24
móc: 24
podnieść: 24
intel: 24
uwaga: 23
chodzić: 22
pieniądz: 22
należeć: 21
wicemarszałek: 21
współpracownik: 21
asys: 21
pomoc: 21
zmiana: 21
eurokolchóz: 21
robić: 20
zrobić: 20
pamiętać: 20
sondapo: 20
szokujący: 20
wypowiedziach: 20
zaprzeczać: 20
sobieistniej: 20
ryzyko: 20
dobry: 19

propozycja: 19
prawica: 19
raz: 19
wystąpien: 19
rządowy: 18
polak: 18
świadczec: 17
czas: 17
osoba: 17
inny: 17
jakość: 17
poprawic: 17
nowelizacja: 17
rzecznik: 17
kolejny: 16
sytuacja: 16
pacjent: 16
krótki: 16
usługa: 16
brać: 16
powiedzenie: 16
elegancja: 16
syf: 16
brud: 16
metr: 16
muł: 16
popierać: 15
dać: 15
powinien: 15
imigracyjny: 15
ręka: 15
inflacja: 15
musiec: 15
wstawac: 15
wybory: 15
przyjac: 14
prezydent: 14
sensowny: 14
zabrac: 14
gnojom: 14
dewastujacym: 14
ograć: 14
dzień: 14
dostać: 14
nielegalny: 14
wychowywać: 14
wykluczyć: 14
kolan: 14
doskonały: 14
stan: 14
biegunk: 14
stomia: 14
zespół: 14
temat: 13
miliard: 13
granica: 13
spotkanie: 13
premier: 13
niemiecki: 13
widać: 13
fabryka: 13
brak: 12
podatnik: 12
fundować: 12
działanie: 12
prawo: 12

socjalny: 12
dawać: 12
wyborca: 12
lewica: 12
zachód: 12
stać: 12
opozycyjny: 12
czołówka: 12
zainwestować: 12
wybudować: 12
skandal: 12
fundowaliać: 12
sasin: 12
miejsce: 11
wielki: 11
przygotować: 11
komentarz: 11
koniec: 11
kobieta: 11
kasa: 11
wypowiedź: 11
jedyne: 11
firma: 11
dziękować: 11
komisja: 11
trup: 11
skrzydło: 11
przypominać: 11
klub: 10
polecać: 10
raport: 10
robota: 10
dzisiejszy: 10
ukraina: 10
doświadczyć: 10
średni: 10
zlikwidować: 10
system: 10
milion: 10
wzgląd: 10
emerytura: 10
rosja: 10
rozwijać: 10
żywy: 10
migrant: 9
finansowy: 9
żał: 9
wyciągnięcie: 9
wolicie: 9
sob: 9
znajdziecie: 9
demograficzny: 9
ukryć: 9
większość: 9
rozumieć: 9
żyć: 9
cel: 9
drugi: 9
bezrobocie: 9
bezrobotni: 9
Polska: 9
wypadać: 9
unijny: 9
jeden: 9
bogaty: 9
zagrozić: 9

całość: 9
oznajmiać: 9
podwyższyć: 9
dodatek: 8
przyjeżdżać: 8
głosować: 8
rzeczywisty: 8
prognoz: 8
znać: 8
wojna: 8
społ: 8
obniżyć: 8
wzmacniać: 8
przemoc: 8
strona: 8
ziemia: 8
widzieć: 8
mały: 8
złoty: 8
przyjechać: 8
dokonać: 8
rozmawiać: 8
zakładać: 8
proponować: 8
wykluczenie: 8
radzić: 8
afer: 8
duklanowski: 8
nadzieja: 8
poprzeć: 8
patologia: 8
spowodować: 8
działać: 7
ubezpieczenie: 7
spadać: 7
społeczeństwo: 7
dom: 7
publiczny: 7
prosty: 7
zmienić: 7
rozpocząć: 7
posiedzenie: 7
mężczyzna: 7
pomysł: 7
euro: 7
polityk: 7
wystarczyć: 7
biedni: 7
rewolucja: 7
godnościowy: 7
zły: 7
przewidzieć: 7
oszczędność: 7
resorta: 7
obronność: 7
realizować: 7
republikapopołud: 7
wieszwięcej: 7
wiceminister: 7
europejski: 7
radzących: 7
unia: 7
relokacja: 7
zwracać: 7
skandaliczny: 7
wypowiedz: 7

deprecjonować: 7
członek: 7
pomóc: 6
korporacja: 6
zasada: 6
składka: 6
powiedzieć: 6
opinia: 6
rozdawać: 6
emerytalny: 6
punkt: 6
ograniczyć: 6
wysoki: 6
chwila: 6
bieda: 6
opieka: 6
korzystać: 6
pierwszy: 6
wyglądać: 6
konfederacja: 6
obywatelstwo: 6
pis: 6
świat: 6
niski: 6
opiekuńczy: 6
urząd: 6
wiadomo: 6
choroba: 6
zarabiać: 6
różnica: 6
imigracja: 6
legalny: 6
zmniejszyć: 6
sprawa: 6
tysiąc: 6
jechać: 6
zaskakujący: 6
czytać: 6
nasłuchać: 6
ściana: 6
wspierać: 6
prosić: 6
wrócić: 6
część: 6
ht: 6
beneficjent: 6
przysługiwać: 6
wprowadzać: 6
odsłona: 6
przyszłość: 6
zapowiedź: 6
programowyul: 6
zwiększyć: 6
apelem: 6
zaprzestać: 6
religia: 5
dobro: 5
fakt: 5
małąg: 5
mieszkanie: 5
wynikać: 5
utożsamiać: 5
odebrać: 5
islamski: 5
pozwolenie: 5
szukać: 5

możliwość: 5
udać: 5
rosnąć: 5
opłacać: 5
pisać: 5
dwunasty: 5
ciekawy: 5
wola: 5
gospodarka: 5
wyspa: 5
myśleć: 5
podjąć: 5
dokładać: 5
wydatek: 5
zagraniczny: 5
młody: 5
wniosek: 5
wychowawczy: 5
składać: 5
zacząć: 5
napisać: 5
tani: 5
otrzymywać: 5
ciężki: 5
woleć: 5
zielony: 5
liczyć: 5
salon: 5
miesiąc: 5
liczba: 5
zajmować: 5
przeciwdziałać: 5
społecznemu: 5
analiza: 5
partia: 5
całkowity: 5
dany: 5
obawiać: 5
największy: 5
referendum: 5
pochodzenie: 4
pkb: 4
socjać: 4
wsparcie: 4
lider: 4
stabilny: 4
wpływie: 4
stomatologiczny: 4
kontynuacja: 4
badanie: 4
eme: 4
zarejestrować: 4
służba: 4
prawda: 4
cena: 4
gówny: 4
ciąg: 4
wprowadzić: 4
traktować: 4
sondaż: 4
grupa: 4
konfa: 4
kwestia: 4
rynek: 4
warunki: 4
zgadzać: 4

bierny: 4
wydarzenie: 4
zamknąć: 4
wysłać: 4
twierdzić: 4
utrzymać: 4
wy tłumaczyć: 4
wyjechać: 4
siła: 4
mieszkać: 4
zatrudnić: 4
zwolnić: 4
dlatego że: 4
bezbронny: 4
matka: 4
poszukiwać: 4
szczęście: 4
dochód: 4
język: 4
zasilek: 4
obradę: 4
otrzymać: 4
szpital: 4
potrafić: 4
droga: 4
zamykać: 4
finansować: 4
wybór: 4
babciowy: 4
podwyżka: 4
kwota: 4
głośny: 4
odmówić: 4
przywilej: 4
renta: 4
potrzebny: 4
udzielać: 4
zapewnić: 4
grać: 4
nfz: 4
procedować: 4
inwestować: 4
najmłodszy: 4
debatadnie: 4
przerażony: 4
ofiara: 3
główny: 3
inżynier: 3
rządzić: 3
zapowiadać: 3
najlepszą: 3
migracyjny: 3
niemądry: 3
wiedzi: 3
wspólny: 3
prywatny: 3
wschodni: 3
głupi: 3
wpuszczać: 3
wjeżdżać: 3
pokomunistyczny: 3
kwa: 3
zostawiliście: 3
wiek: 3
środek: 3
zależać: 3

kultura: 3
potrzebować: 3
miasto: 3
inwestycja: 3
iść: 3
przyjmować: 3
chleb: 3
zapraszać: 3
rozwiązanie: 3
podnosić: 3
dotyczyć: 3
przyznawać: 3
związek: 3
polityczny: 3
energetyczny: 3
twarz: 3
energ: 3
zwalczać: 3
dziwny: 3
imigranci: 3
demokratyczny: 3
wolność: 3
zamożny: 3
moralny: 3
aborcja: 3
potrzeba: 3
rejestrować: 3
cieszyć: 3
bezdomny: 3
jeździć: 3
racja: 3
nauczyć: 3
zmieniać: 3
dzielić: 3
obcy: 3
centralny: 3
jasny: 3
limit: 3
czytać: 3
koszt: 3
rodzinny: 3
turysta: 3
obecny: 3
urodzić: 3
chęć: 3
setka: 3
proc: 3
postulat: 3
równość: 3
zdrowy: 3
super: 3
odpowiedzialny: 3
zarobek: 3
kryterium: 3
zło: 3
pozbawić: 3
wyrównywania: 3
spóźnić: 3
potwierdzać: 3
zaszczyt: 3
imię: 3
istnieć: 3
powszechny: 3
republikapopołudniu: 3
zapropionować: 3
przypadek: 3

podniesienie: 3
kadencja: 3
dochodowy: 3
wynosić: 3
czekać: 3
tytuł: 3
przyczyna: 3
ukraińca: 3
oko: 3
ukraińcy: 3
władza: 3
udział: 3
pełny: 3
bóg: 3
pięknietylko: 3
uciec: 3
słowacja: 3
rumuniczy: 3
tydzień: 3
zdrowie: 3
wziąć: 3
głowa: 3
konfy: 3
przestańcie: 3
nierob: 3
dupa: 3
nazista: 3
wyrok: 3
podlegać: 3
członkowski: 3
kupować: 3
trzeci: 3
wiedza: 3
ukończyć: 3
efekt: 3
wybrać: 3
rada: 3
naród: 3
nacja: 3
armia: 3
migranta: 3
decydować: 3
własny: 3
dane: 3
dostęp: 3
bronić: 3
kredyt: 3
myśl: 3
tusk: 3
strach: 3
umrzeć: 3
wolsce: 3
zabierać: 3
ubog: 3
społecznogospodarczy: 3
bojkotować: 3
legislacja: 3
zmuszeć: 3
zysk: 3
plac: 3
placówka: 3
endoprotezoplastyka: 3
staw: 3
kolanowy: 3

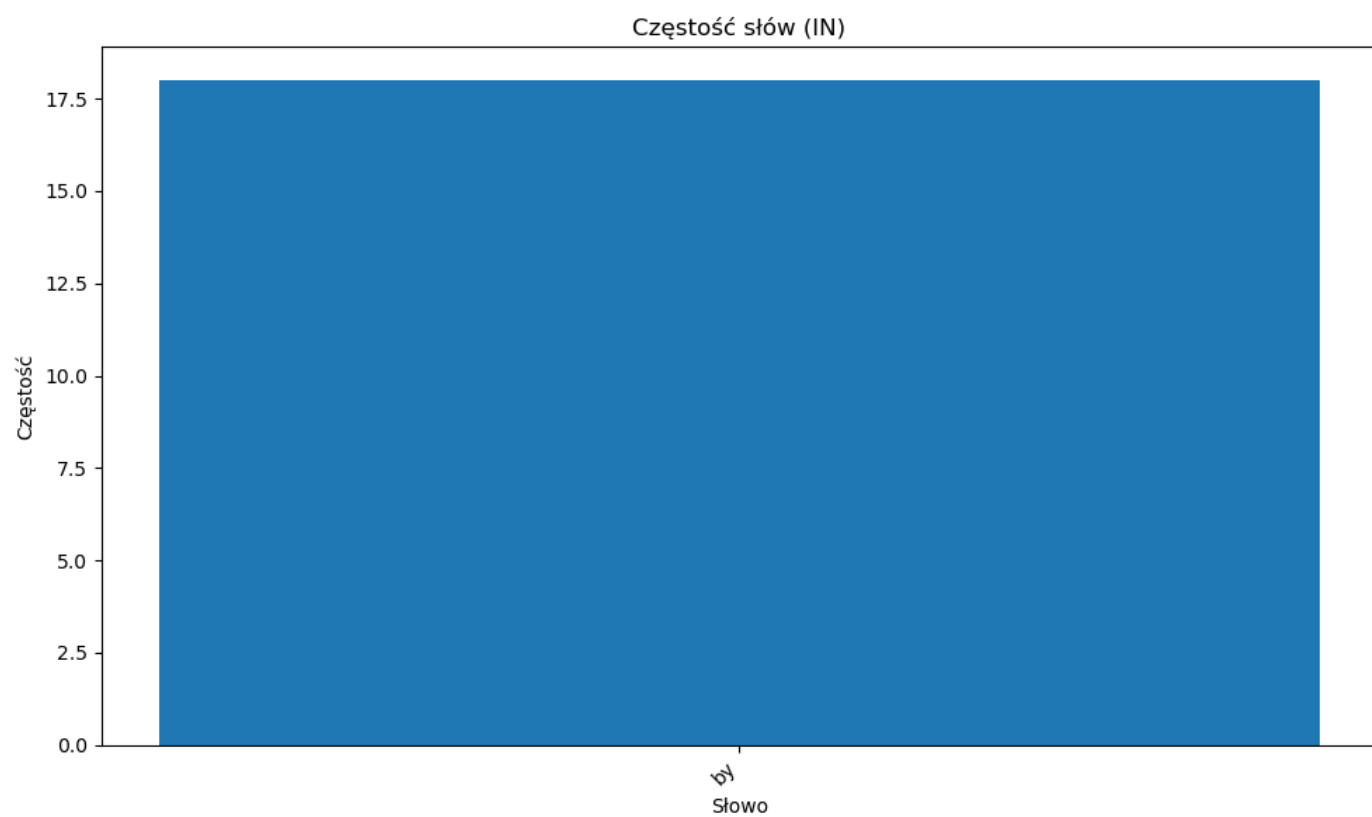
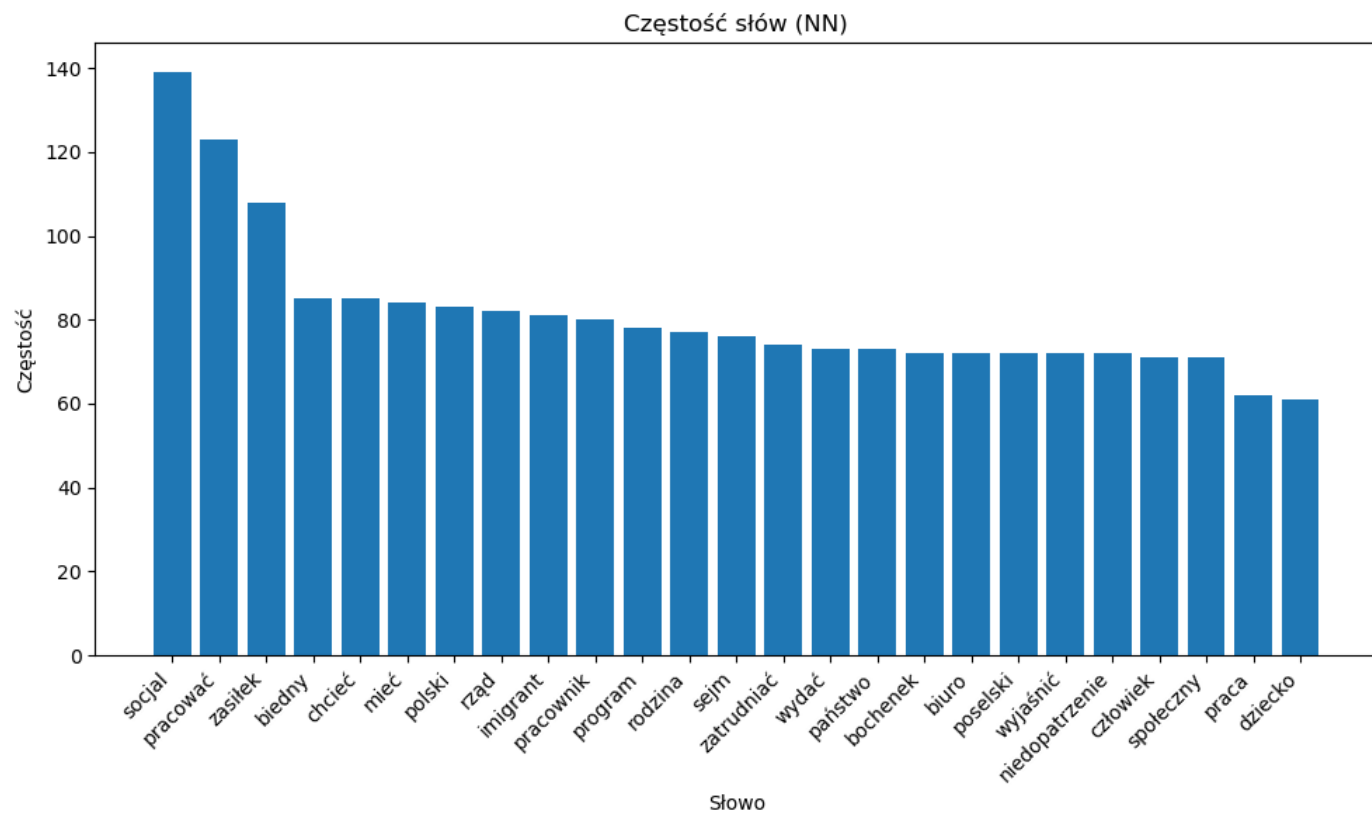
```
--- IN ---  
by: 18
```

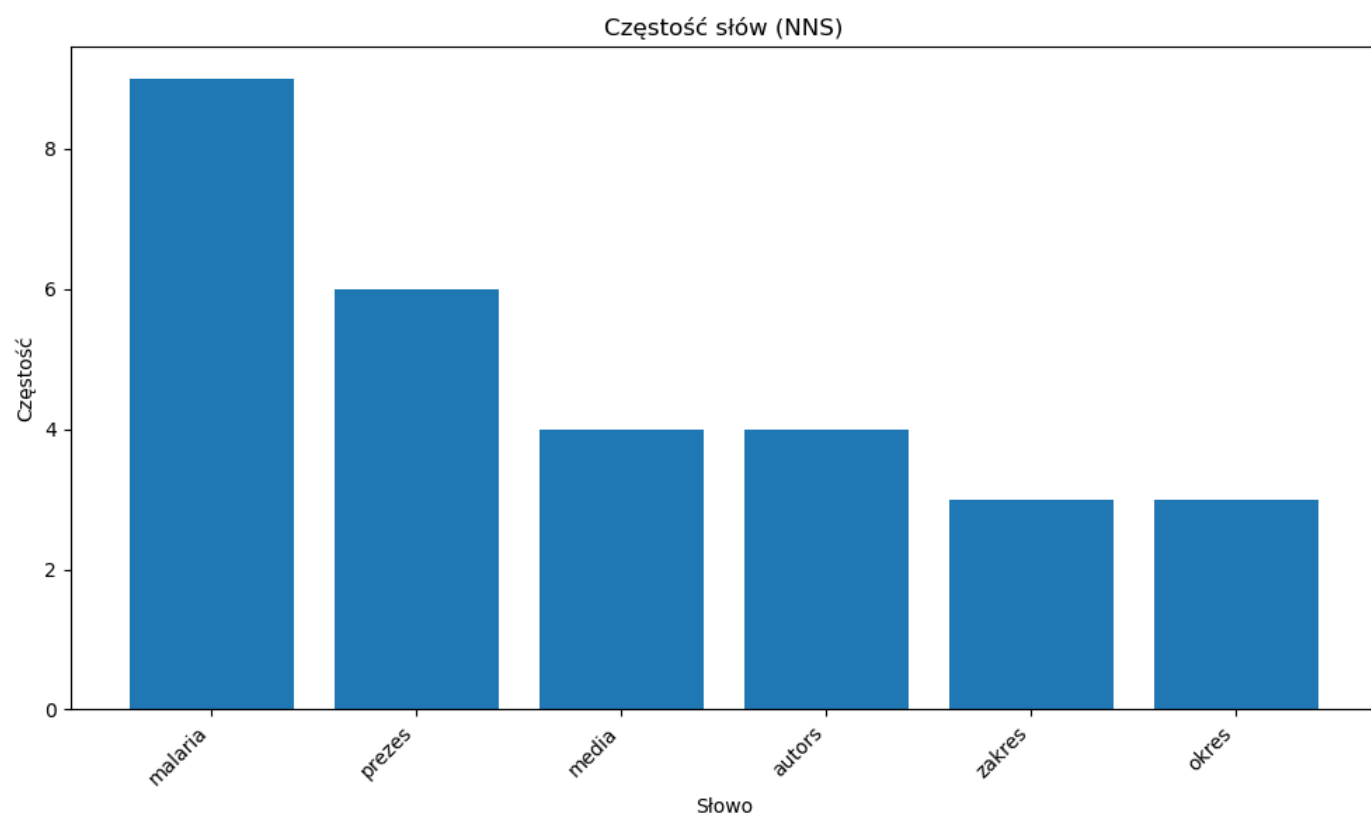
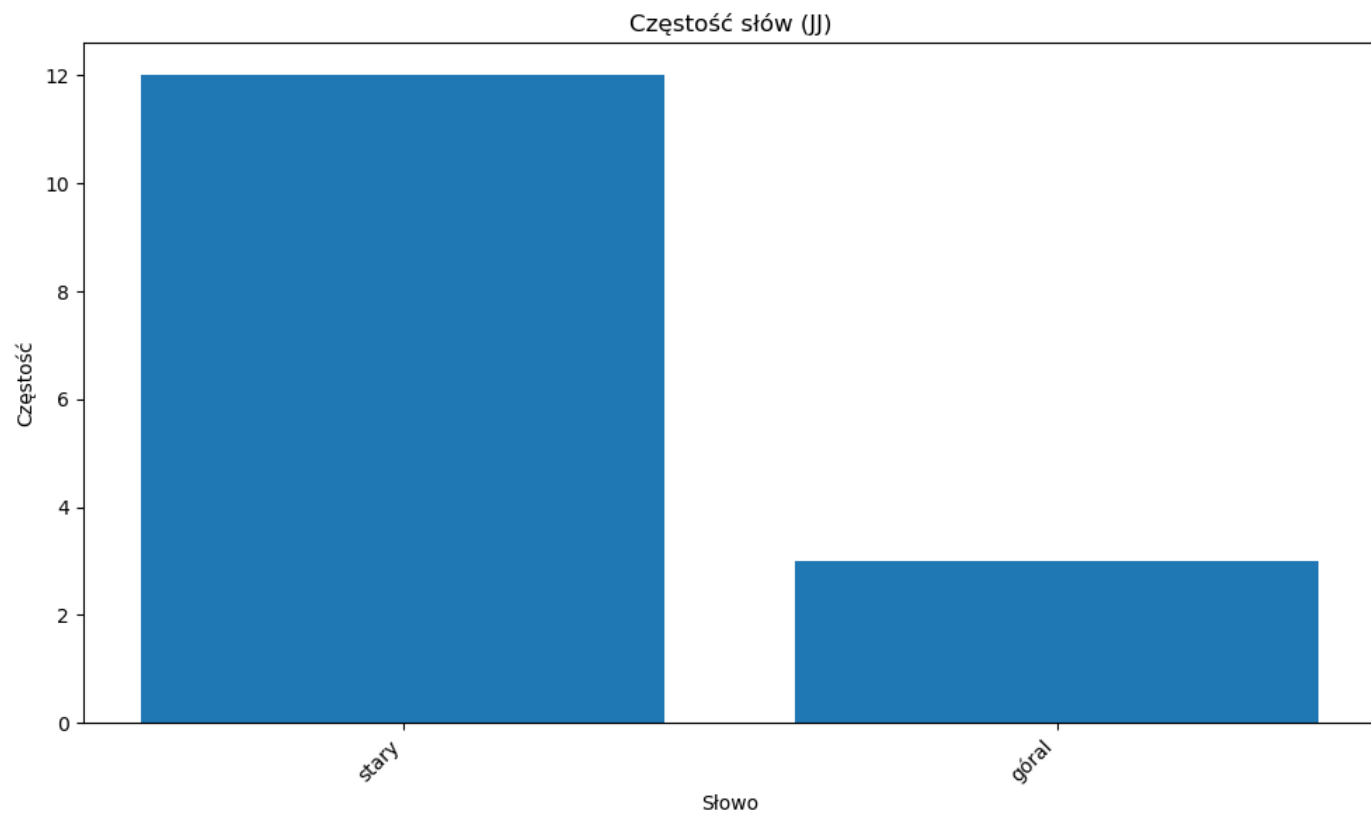
```
--- JJ ---  
stary: 12  
góral: 3
```

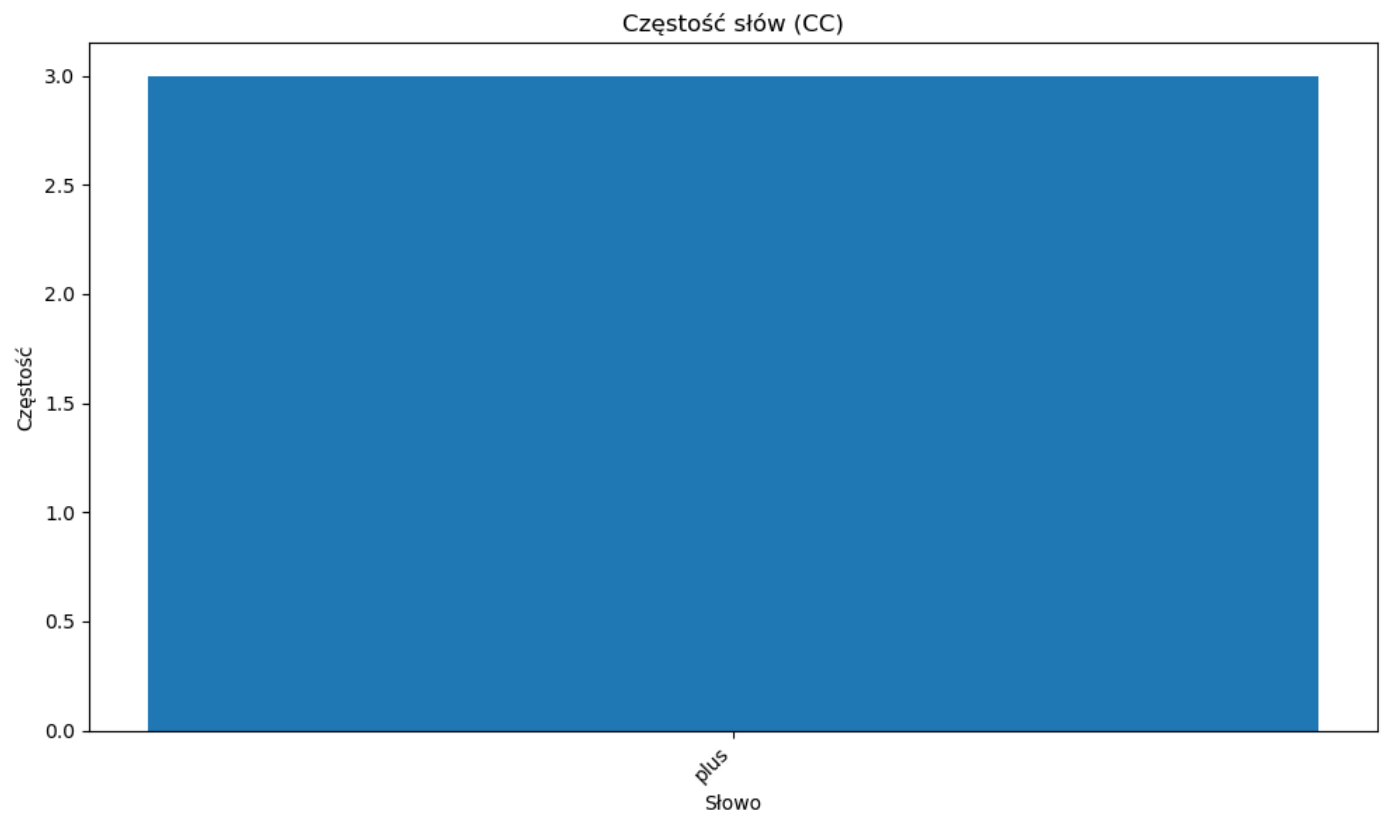
```
--- NNS ---  
malaria: 9  
prezes: 6  
media: 4  
autors: 4  
zakres: 3  
okres: 3
```

```
--- CC ---  
plus: 3
```

```
In [9]: # Wygenerowanie wykresów dla każdej części mowy  
for pos, freq in pos_freq.items():  
    sorted_freq = sorted(freq.items(), key=lambda x: x[1], reverse=True) # Sortowanie w  
    words = [word for word, _ in sorted_freq[:25]] # Wybór najczęstszych 25 słów  
    frequencies = [frequency for _, frequency in sorted_freq[:25]]  
  
    # Wygenerowanie wykresu słupkowego dla danej części mowy  
    plt.figure(figsize=(10, 6))  
    plt.bar(words, frequencies)  
    plt.xlabel('Słowo')  
    plt.ylabel('Częstość')  
    plt.title(f'Częstość słów ({pos})')  
    plt.xticks(rotation=45, ha='right') # Skrócenie etykiet i obrót o 45 stopni  
    plt.tight_layout()  
  
    # Wyświetlenie wykresu  
    plt.show()
```







```
In [10]: def getSubjectivity(tweet):
          return TextBlob(tweet).sentiment.subjectivity

          def getPolarity(tweet):
              return TextBlob(tweet).sentiment.polarity

          df['Subjectivity'] = df['CLEANED_TEXT'].apply(getSubjectivity)
          df['Polarity'] = df['CLEANED_TEXT'].apply(getPolarity)

          df.head(5)
```

Out[10]:

	URL	USER	TEXT	CLEANED_TEXT	Subjectivity
0	https://twitter.com/AnnaBurnosN1/status/167654...	mazureusz	@rocnat Popierają socjal?	popierać	0.0
1	https://twitter.com/Maciej81813761/status/1676...	Maciej81813761	@PEmeryt @marsza_jacek Raczej to Pan błądzi ni...	błądzić odróżniać sprawca ofiara migrant ofiar...	0.0
2	https://twitter.com/Maciej81813761/status/1676...	Cezary_Dalek	RT @winatuska1: Jeżeli Wicemarszałek Sejmu, Ry...	wicemarszałek sejm posiadać współpracownik asy...	0.0
3	https://twitter.com/Maciej81813761/status/1676...	Zyjel	Jaki to ma sens? \\nGłówny argument o tworzeniu...	sens główny argument tworzyć miejsce praca bez...	0.0
4	https://twitter.com/Maciej81813761/status/1676...	MalgoPiasecka	@KacperBomba2 @Andrzej31263190 No i tutaj poru...	poruszyć ważny temat kraj rządzić religia kons...	0.0

```
In [11]: def getAnalysis(score):
          if score < 0:
```

```

    return "Negative"
elif score == 0:
    return "Neutral"
else:
    return "Positive"

```

```

df["Analysis"] = df['Polarity'].apply(getAnalysis)
df.head(5)

```

Out[11]:

	URL	USER	TEXT	CLEANED_TEXT	Subjectivity
0	https://twitter.com/AnnaBurnosN1/status/167654...	mazureusz	@rocnat Popierają socjal?	popierać	0.0
1	https://twitter.com/Maciej81813761/status/1676...	Maciej81813761	@PEmeryt @marsza_jacek Raczej to Pan błądzi ni...	błądzić odróżniać sprawca ofiara migrant ofiar...	0.0
2	https://twitter.com/Maciej81813761/status/1676...	Cezary_Dalek	RT @winatuska1: Jeżeli Wicemarszałek Sejmu, Ry...	wicemarszałek sejm posiadać współpracownik asy...	0.0
3	https://twitter.com/Maciej81813761/status/1676...	Zyjel	Jaki to ma sens? \nGłówny argument o tworzeniu...	sens główny argument tworzyć miejsce praca bez...	0.0
4	https://twitter.com/Maciej81813761/status/1676...	MalgoPiasecka	@KacperBomba2 @Andrzej31263190 No i tutaj poru...	poruszyć ważny temat kraj rządzić religia kons...	0.0

In [12]:

```

j=1
sortedDF = df.sort_values(by=['Polarity'])
for i in range(0, sortedDF.shape[0]):
    if(sortedDF['Analysis'][i] == 'Positive'):
        print(str(j) + ' ' + sortedDF['CLEANED_TEXT'][i])
        print()
        j=j+1

```

- 1) przyjmować imigrant pracować płacić podatek brać sociale mieć zapierdalać super pomys
ł dom oddać biedn
- 2) obcokrajowiec stanowić bezrobotny zatrudnić mieć socjal praca zdechać głód wylądować
most
- 3) potężny raek krytykować super klub wiatr oko trenować biedn grać blaszak chcieć znisz
czyć
- 4) wyzywanie kogos sam dosłownie kazdy gówny konfy dawać konfa super problem socjal akce
ptujecie robić spreczny zapowiadać popierać konfy

In [13]:

```

j=1
sortedDF = df.sort_values(by=['Polarity'], ascending = False)
for i in range(0, sortedDF.shape[0]):
    if(sortedDF['Analysis'][i] == 'Negative'):
        print(str(j) + ' ' + sortedDF['CLEANED_TEXT'][i])
        print()
        j=j+1

```

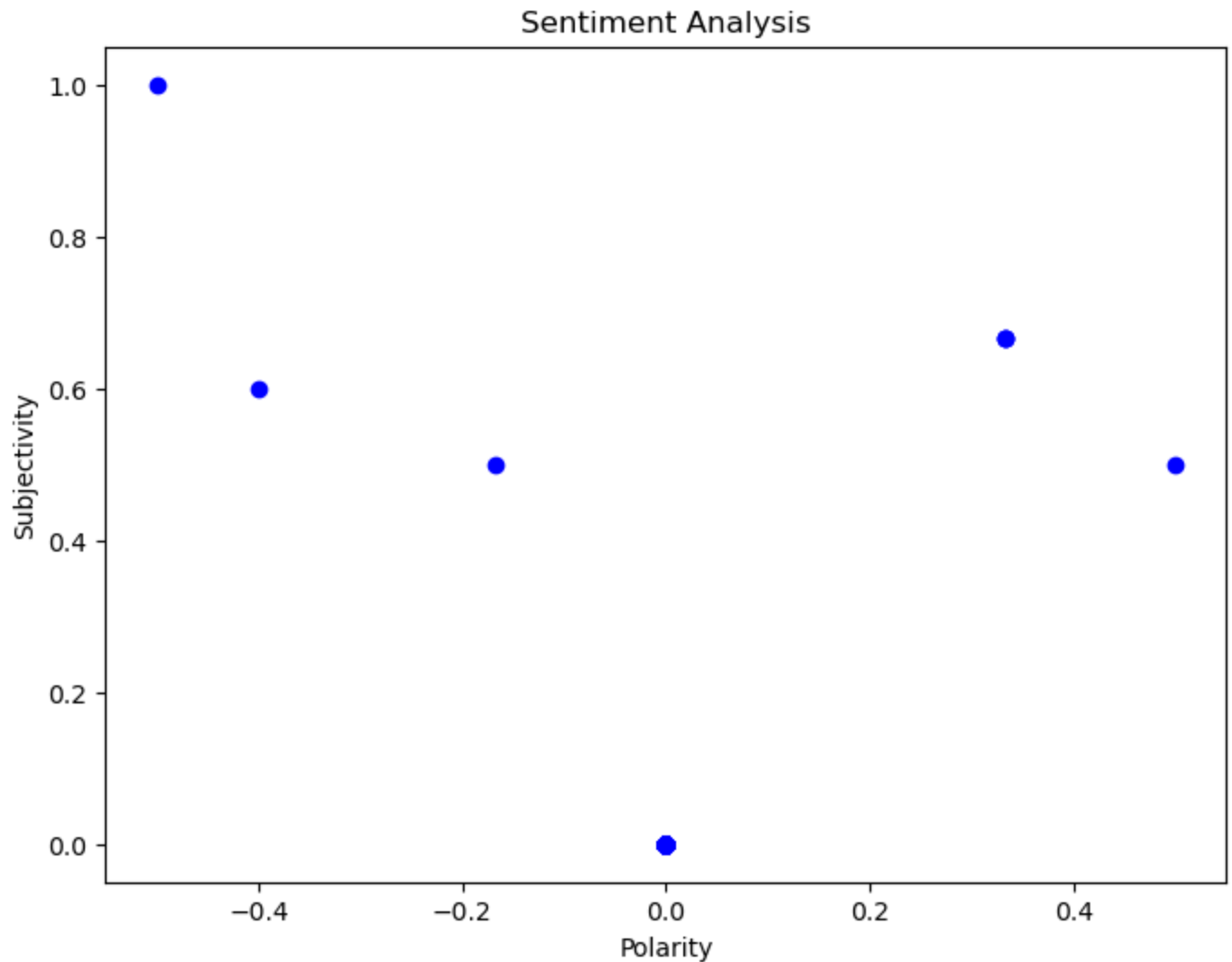
- 1) zwać stary dekada rządzić socjalista rozbudować socjal granica absurd gwarantować ogr
aniczyć socjal osoba potrzebować napływ imigrant spadnąć połowa połowa

2) turyngia spuszczony stary zjebow popierdolć człowiek chuj faszystowski stary dziadow w kurwiony podział biedny pamiętać jedynymiejsce widzieć być nalejek plakatow fuck afd

3) typical ukrainiec wypierdalać kraj bronić Polska

```
In [14]: plt.figure(figsize=(8,6))
for i in range(0, df.shape[0]):
    plt.scatter(df['Polarity'][i], df['Subjectivity'][i], color = 'Blue')

plt.title('Sentiment Analysis')
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
plt.show()
```



```
In [15]: ptweets = df[df.Analysis == 'Positive']
ptweets = ptweets['CLEANED_TEXT']

round(ptweets.shape[0] / df.shape[0] *100 , 1)
```

Out[15]: 0.4

```
In [16]: ptweets = df[df.Analysis == 'Negative']
ptweets = ptweets['CLEANED_TEXT']

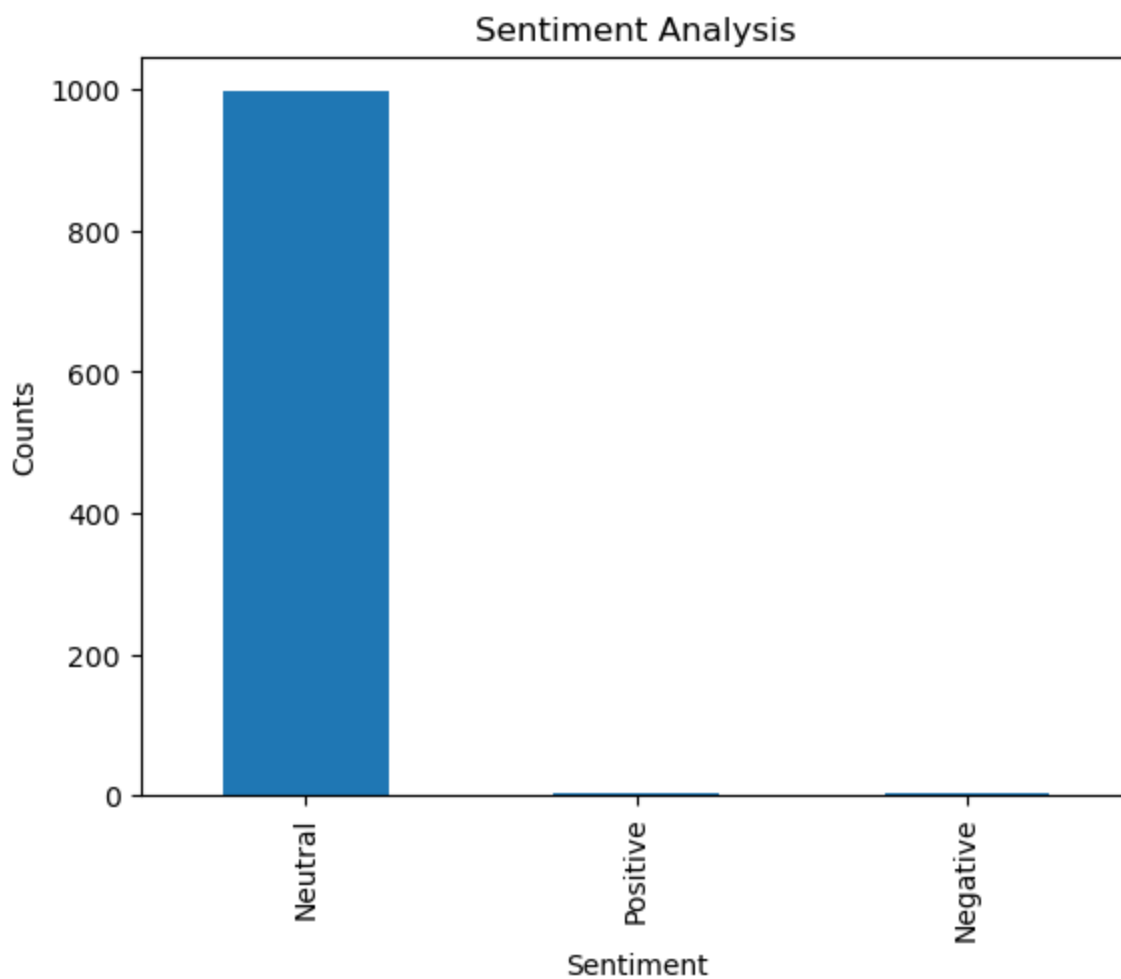
round(ptweets.shape[0] / df.shape[0] *100 , 1)
```

Out[16]: 0.3

```
In [17]: ptweets = df[df.Analysis == 'Neutral']  
ptweets = ptweets['CLEANED_TEXT']  
  
round(ptweets.shape[0] / df.shape[0] *100 , 1)
```

Out[17]: 99.3

```
In [18]: df['Analysis'].value_counts()  
  
plt.title("Sentiment Analysis")  
plt.xlabel('Sentiment')  
plt.ylabel('Counts')  
df['Analysis'].value_counts().plot(kind='bar')  
plt.show()
```



```
In [19]: # Połączenie oczyszczonych tweetów w jednozdaniowy tekst  
combined_text = ' '.join(df['CLEANED_TEXT'])  
  
# Analiza sentymentu dla całego tekstu tweetów naraz  
sentiment = TextBlob(combined_text).sentiment.polarity  
  
# Wyświetlenie sentymentu  
print('Sentiment:', sentiment)
```

Sentiment: 0.06190476190476191