

# Zakup Twittera przez Elona Muska

Łukasz Nowosielski

## 1 Dataset

W dzisiejszych czasach, media społecznościowe stały się głównym źródłem informacji dla wielu osób na całym świecie. Twitter, jako jedna z najpopularniejszych platform, jest miejscem, gdzie użytkownicy dzielą się swoimi opiniami na różne tematy. Jednym z takich tematów, który wywołał duże zainteresowanie, był zakup Twittera przez Elona Muska, założyciela i CEO firmy SpaceX i Tesla. Ta decyzja wywołała liczne reakcje wśród użytkowników Twittera, co czyni ją interesującym tematem do analizy.

Do projektu wykorzystałem gotową bazę danych zawierającą tweety z dnia, w którym Twitter zaakceptował ofertę Elona Muska na kwotę 44 miliardów dolarów. W ramach tego projektu skupimy się na analizie 100 tysięcy tweetów dotyczących tego wydarzenia. Wykorzystamy różne techniki przetwarzania języka naturalnego oraz analizy danych, aby zrozumieć główne tematy poruszane przez użytkowników, najczęściej używane słowa oraz wydźwięk tych tweetów i jego zmienność w czasie.

```
df = pd.read_csv("tweets.csv")
print(df.shape)
(100000, 39)
```

## 2 Preprocessing

Początkowo przeprowadziłem wstępne przetwarzanie danych. W tym procesie usunąłem duplikaty i zredukowałem liczbę kolumn do trzech najistotniejszych: "created at" reprezentującej czas publikacji tweeta, "tweet" zawierającej treść tweeta oraz "language" określającą język w którym został napisany dany tweet. Następnie wyfiltrowałem tweety napisane w innym języku niż angielski. Dodatkowo, przekształciłem wartości w kolumnie "created at" tak, aby reprezentowały jedynie godzinę publikacji, co będzie przydatne w dalszych etapach analizy.

```

def preprocess_data(df):
    columns_to_drop = [
        "id",
        "conversation_id",
        "user_id",
        "user_id_str",
        "date",
        "timezone",
        "place",
        "cashtags",
        "hashtags",
        "username",
        "link",
        "nlikes",
        "nreplies",
        "nretweets",
        "day",
        "hour",
        "urls",
        "photos",
        "video",
        "thumbnail",
        "retweet",
        "quote_url",
        "search",
        "name",
        "near",
        "geo",
        "source",
        "user_rt_id",
        "user_rt",
        "retweet_id",
        "reply_to",
        "retweet_date",
        "translate",
        "trans_src",
        "trans_dest",
    ]
    df.drop(columns_to_drop, axis=1, inplace=True)
    df = df.iloc[:, 1:]

    df = df[df["language"] == "en"]
    df["tweet"] = df["tweet"].str.lower()
    df.drop_duplicates(subset="tweet", keep="first", inplace=True)
    df["created_at"] = pd.to_datetime(df["created_at"], unit="ms")
    df["created_at"] = df["created_at"].dt.hour

    return df

```

```
print(df.head())
print(df.shape)
```

	created_at	tweet	language
0	23	now that free speech has finally been restored...	en
1	23	@govabbott let's bring twitter to texas possib...	en
2	23	@mmm_oranges if elon musk takes over twitter,...	en
3	23	@elonmusk now that twitter is yours can i post...	en
4	23	@derpfighter @cenkuygur @elonmusk @twitter ht...	en

```
(75261, 3)
```

### 3 Tokenizacja

Kolejnym krokiem jest tokenizacja. Jest to kluczowy krok w przetwarzaniu języka naturalnego. Polega na podziale tekstu na mniejsze jednostki, zwane tokenami. W naszym przypadku treść tweetów zostaje podzielona na pojedyncze wyrazy. Tokenizacja pomaga w analizie tekstu, ponieważ pozwala na łatwiejsze zrozumienie i analizę struktury tekstu.

```
def tokenize(df):
    df["tweet_words"] = df["tweet"].apply(word_tokenize)

    return df
```

```
df = tokenize(df)
print(df["tweet_words"].head())
```

0	[now, that, free, speech, has, finally, been, ...
1	[@, govabbott, let, ', s, bring, twitter, to, ...
2	[@, mmm_oranges, if, elon, musk, takes, over,...
3	[@, elonmusk, now, that, twitter, is, yours, c...
4	[@, derpfighter, @, cenkuygur, @, elonmusk, @,...

```
Name: tweet_words, dtype: object
```

## 4 Stopwords

Po tokenizacji, kolejnym krokiem jest usunięcie tzw. "stopwords". Stopwords to najczęściej występujące słowa w danym języku, które zazwyczaj nie niosą dużo znaczenia, takie jak "the", "is", "at", "which", i "on" w języku angielskim. Usunięcie tych słów pozwoli nam skupić się na tych, które są najbardziej istotne dla naszej analizy.

```
def stop_words(df):
    stop_words = set(stopwords.words("english"))
    additional_stopwords = ["http", "https"]
    stop_words.update(additional_stopwords)
    df["tweet_words"] = df["tweet_words"].apply(
        lambda x: [
            word
            for word in x
            if word.isalpha() and word not in stop_words and len(word) > 1
        ]
    )

    return df
```

W powyższym kodzie, oprócz standardowych angielskich stopwords, dodajemy również własne, specyficzne dla naszego przypadku - "http" i "https". Są to częste elementy w tweetach, które nie niosą istotnej informacji. Ponadto filtrujemy słowa krótsze niż dwa znaki, ponieważ zazwyczaj nie mają one dużego znaczenia i mogą zakłócać wyniki analizy.

```
print(df["tweet_words"].head())
df = stop_words(df)
print(df["tweet_words"].head())
0    [now, that, free, speech, has, finally, been, ...
1    [@, govabbott, let, ', s, bring, twitter, to, ...
2    [@, mmmm_oranges, if, elon, musk, takes, over,...
3    [@, elonmusk, now, that, twitter, is, yours, c...
4    [@, derpfighter, @, cenkuygur, @, elonmusk, @,...
Name: tweet_words, dtype: object
0    [free, speech, finally, restored, twitter, tha...
1    [govabbott, let, bring, twitter, texas, possib...
2    [elon, musk, takes, twitter, everyone, leave, ...
3    [elonmusk, twitter, post, words, like, fu, sh,...
4    [derpfighter, cenkuygur, elonmusk, twitter, wo...
Name: tweet_words, dtype: object
```

## 5 Lematyzacja

Po usunięciu stopwords przechodzimy do lematyzacji. Lematyzacja to proces przekształcania słów do ich formy podstawowej, czyli lematu. Pomaga ona w redukcji szumów i złożoności w danych tekstowych, co jest szczególnie przydatne w analizie sentymentu. Zastosujemy w tym przypadku *WordNetLemmatizer*, który bierze pod uwagę kontekst słowa, a nie tylko jego formę.

```
def lemmatize(df):
    lemmatizer = WordNetLemmatizer()
    df["tweet_words_lem"] = df["tweet_words"].apply(
        lambda x: [lemmatizer.lemmatize(word) for word in x]
    )

    return df
```

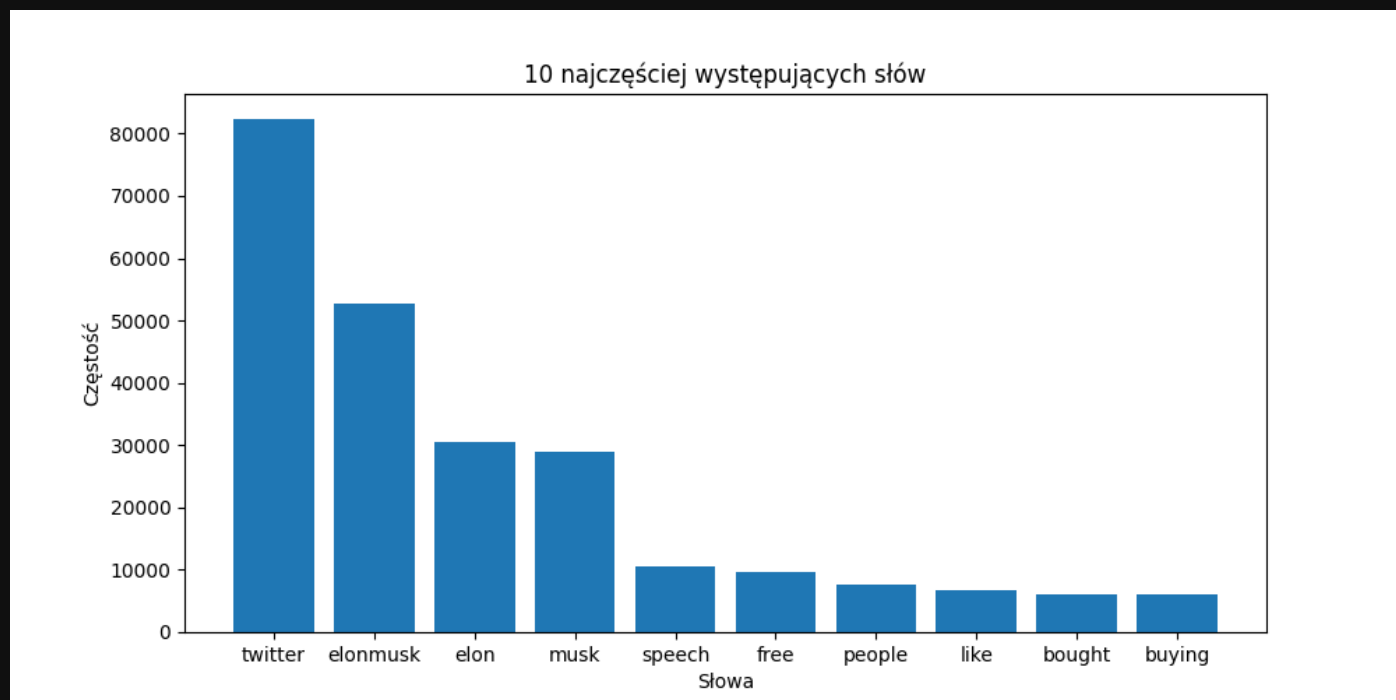
Sprawdźmy jeszcze ile tokenów zostało rzeczywiście zmienionych przez ten proces. Robimy to poprzez porównanie oryginalnych tokenów z ich lematyzowanymi wersjami i zliczanie, ile z nich jest różnych.

```
df = lemmatize(df)
df["equals"] = df["tweet_words"] == df["tweet_words_lem"]
print(df["equals"].value_counts())
equals
False      43665
True       31596
Name: count, dtype: int64
```

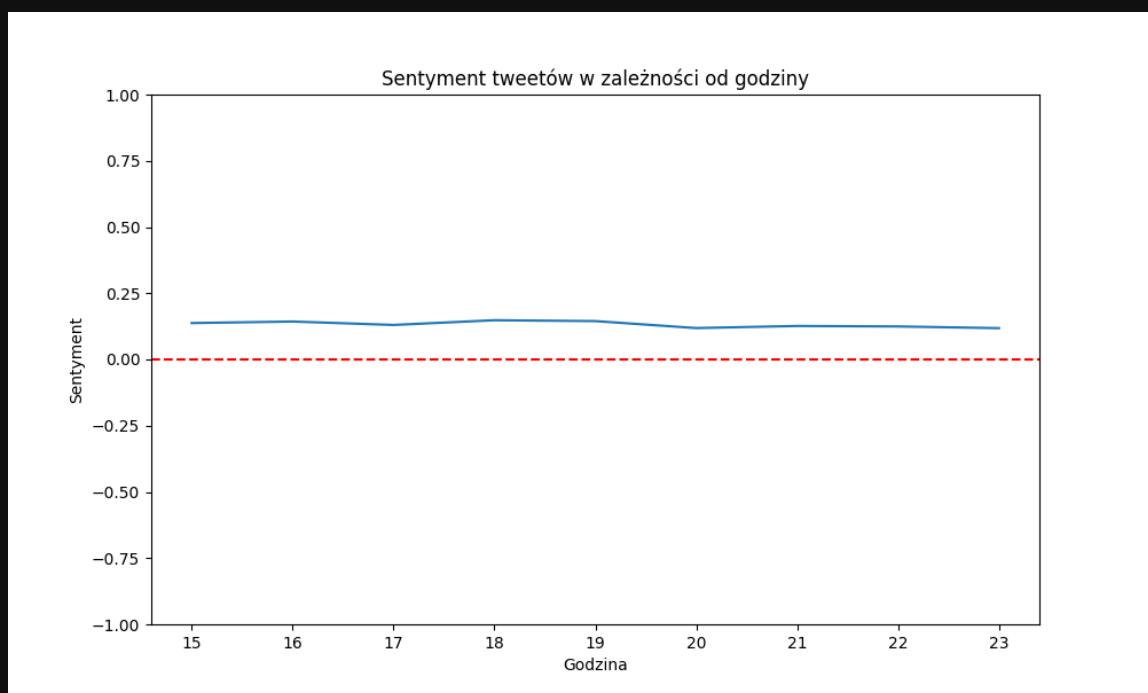
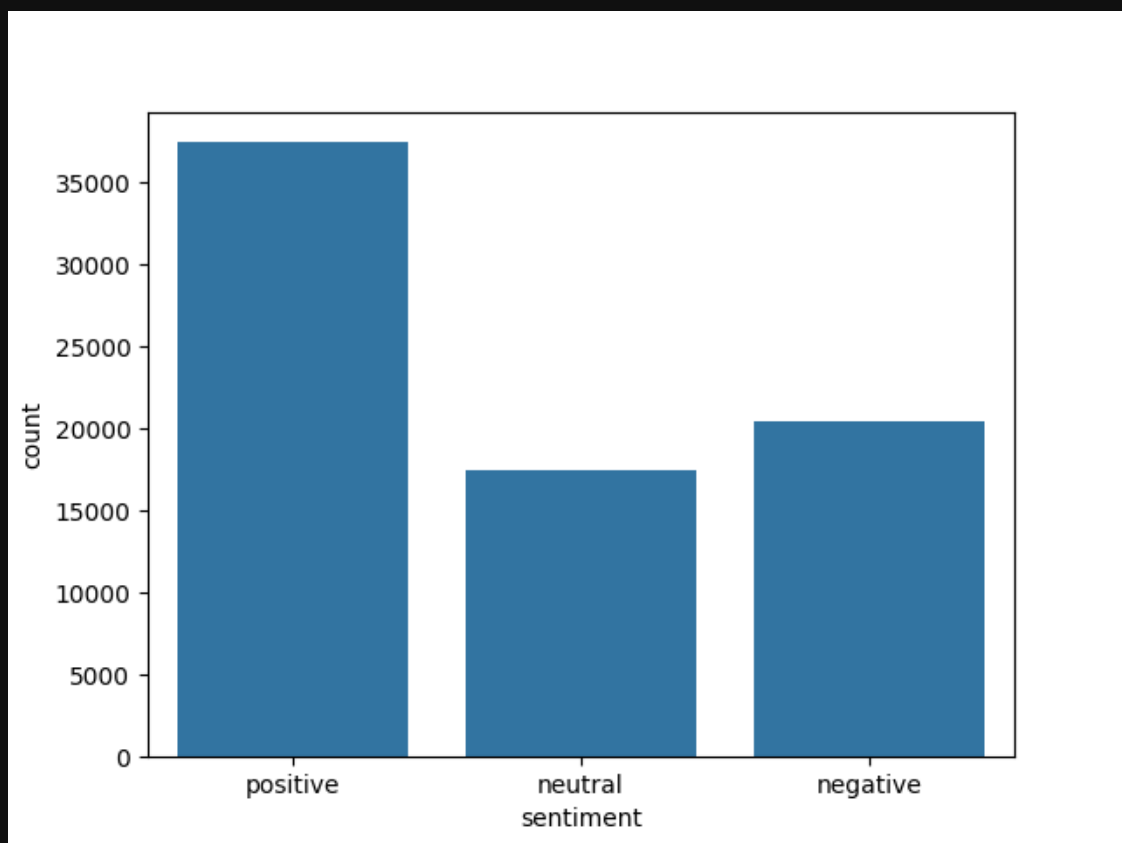
Jak widzimy proces lematyzacji zmienił 43665 tokenów, a 31596 pozostało niezmienionych.

## 6 Najczęściej występujące słowa

Po wszystkich etapach przetwarzania tekstu, takich jak usuwanie stopwords i lematyzacja, narysujmy dwa wizualne przedstawienia najczęściej występujących słów w naszym zbiorze tweetów.





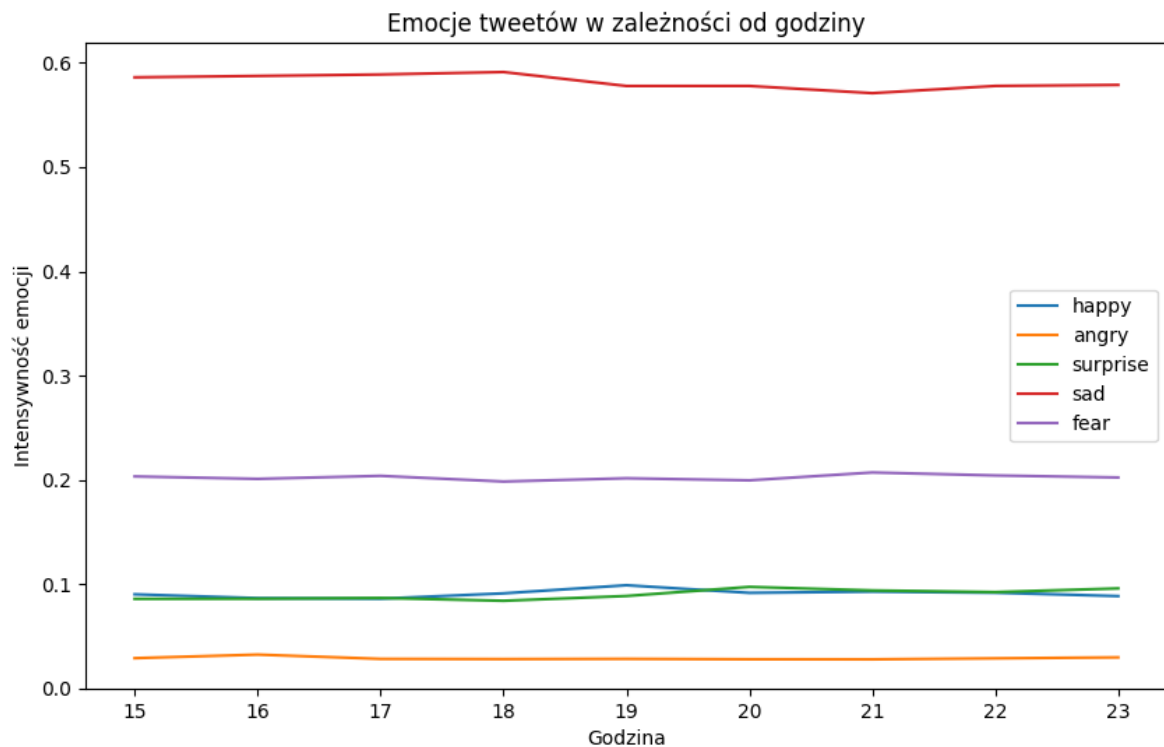


Analizując drugi wykres, zauważamy, że wszystkie punkty znajdują się powyżej zera, co wskazuje na ogólnie pozytywny sentyment tweetów w badanym okresie. Bez względu na porę dnia, średnia wartość sentymentu pozostawała powyżej neutralnej, co sugeruje, że użytkownicy pozytywnie ocenili zakup Twittera przez Elona Muska.

## 8 Analiza emocji

Analiza emocji służy do identyfikacji i wydobycia emocji wyrażonych w tekście. Korzystając z biblioteki *text2emotion*, możemy analizować zawarte w nim emocje, takie jak radość, złość, zaskoczenie, smutek i strach.

```
def emotion_analysis(df):  
    df_emotions = df.copy()  
    df_emotions["emotions"] = df["tweet_words_lem"].apply(  
        lambda x: te.get_emotion(" ".join(x))  
    )  
    df_emotions["happy"] = df_emotions["emotions"].apply(lambda x: x["Happy"])  
    df_emotions["angry"] = df_emotions["emotions"].apply(lambda x: x["Angry"])  
    df_emotions["surprise"] = df_emotions["emotions"].apply(lambda x: x["Surprise"])  
    df_emotions["sad"] = df_emotions["emotions"].apply(lambda x: x["Sad"])  
    df_emotions["fear"] = df_emotions["emotions"].apply(lambda x: x["Fear"])  
  
    return df_emotions
```



## 9 Analiza tematyki opinii i klasteryzacja

Analiza tematyki opinii jest używana do identyfikacji i ekstrakcji tematów z tekstu. Użyjemy jej do zbadania tematów, które dominują wśród opinii użytkowników Twittera na temat przejęcia platformy przez Elona Muska.

Klasteryzacja natomiast jest techniką uczenia maszynowego, która jest używana do grupowania podobnych elementów. Wykorzystamy więc ją do pogrupowania podobnych opinii.



```
def analyze_and_cluster(df):
    df_copy = df[df["language"] == "en"].copy()

    vectorizer = TfidfVectorizer(max_df=0.95, min_df=2, stop_words="english")
    tfidf = vectorizer.fit_transform(
        df_copy["tweet_words_lem"].apply(lambda x: " ".join(x))
    )

    lda = LatentDirichletAllocation(n_components=5, random_state=0)
    lda.fit(tfidf)

    feature_names = vectorizer.get_feature_names_out()
    for i, topic in enumerate(lda.components_):
        print(f"{i+1}:")
        print(" ".join([feature_names[i] for i in topic.argsort()[::-11:-1]]))

    kmeans = KMeans(n_clusters=5, random_state=0)
    kmeans.fit(tfidf)

    df_copy["cluster"] = kmeans.labels_

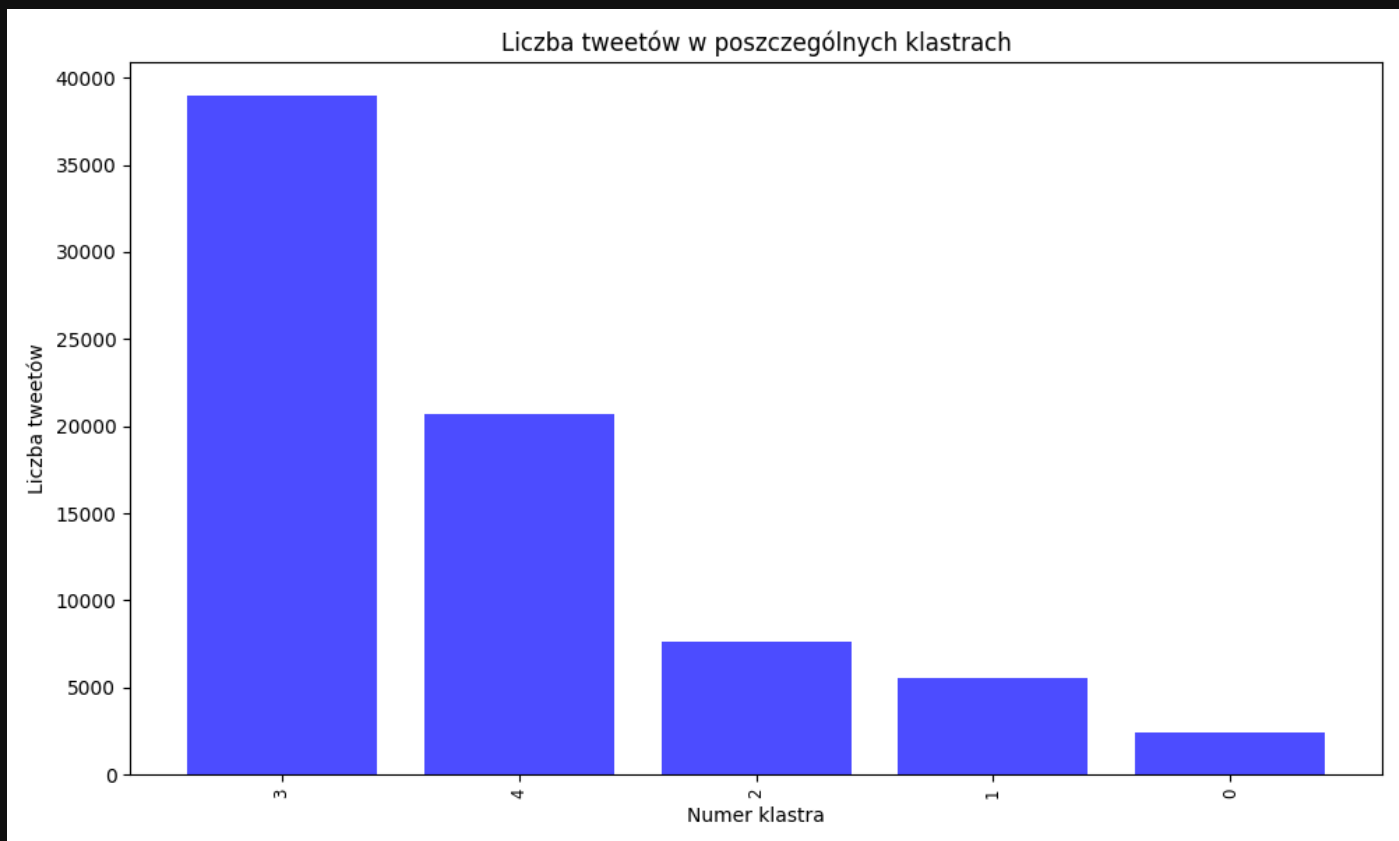
    return df_copy
```

Funkcja ta przeprowadza analizę tematyczną i klasteryzację na zbiorze danych tweetów. Wykorzystuje ona *TfidfVectorizer* do przekształcenia tweetów w reprezentację *TF-IDF*, a następnie stosuje algorytm *Latent Dirichlet Allocation (LDA)* do identyfikacji pięciu głównych tematów. Dodatkowo, korzysta z *K-means* do klasteryzacji tweetów na podstawie tych tematów.

```
0:
elonmusk elon musk bought like people free buying buy billion
1:
elon musk elonmusk bought buy buying elonmusktwitter elonmuskbuytwitter deal owns
2:
elonmusk elon musk thank tesla bought like new buy joined
3:
elonmusk elon musk sassyrightwing marcorubio takeover eu welcome new like
4:
speech elonmusk free musk elon law people platform right want
```

Wyniki możemy interpretować w następujący sposób:

- Temat 0: Finansowe aspekty
- Temat 1: Szczegóły transakcji i własność
- Temat 2: Zakup w kontekście Tesli i nowych zmian
- Temat 3: Polityczne i społeczne reakcje na przejęcie platformy
- Temat 4: Wolność słowa i prawo



Jak widać, znaczna większość tweetów zaliczona została do *tematu 3*, co z kolei sugeruje, że większość z nich koncentruje się na politycznych i społecznych reakcjach na przejęcie Twittera przez Elona Muska.

## 10 Podsumowanie

Reakcja użytkowników na przejęcie Twittera przez Elona Muska była przeważnie pozytywna, co potwierdza analiza sentymentu. Niezależnie od pory dnia, średnia wartość sentymentu nie spadła poniżej neutralnej, co świadczy o ogólnie przychylniej postawie społeczności. Co więcej, analiza emocji pokazała, że poziom złości był stosunkowo niski. Zaskakująco, najbardziej dominującą emocją okazał się być smutek, który konsekwentnie utrzymywał się na poziomie około 0.6. Może to sugerować pewien rodzaj melancholii wśród użytkowników, być może związanej ze zmianami, które mogą nastąpić w wyniku przejęcia platformy.

## 11 Źródła

Baza danych: <https://www.kaggle.com/datasets/aliraza48/elon-musk-tweets?select=elonmusktweet.csv>

Wykłady

ChatGPT