MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Project Report

on

Threads Analysis

Submitted By

Pratham Kaushal (0901AM211038)

Priyanshu Jain (0901AM211042)

Faculty Mentor:

Dr. Anshika Srivastava

Assistant Professor

CENTRE FOR ARTIFICIAL INTELLIGENCE

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

JULY-DEC. 2023

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade

CERTIFICATE

This is certified that **Pratham Kaushal** (0901AM211038) & Priyanshu Jain (0901AM211042) has submitted the project report titled Analyzing User Sentiments and Performance of Threads under the mentorship of Dr. Anshika Srivastava, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Artiificial Intelligence & Machine Learning from Madhav Institute of Technology and Science, Gwalior.

Anshika 23/11/2023
Dr. Anshika Srivastava
Faculty Mentor
Assistant Professor
Centre for Artificial Intelligence

23/11/23
Dr. R. R. Singh
Coordinator
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Artificial Intelligence & Machine Learning at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of Dr. Anshika Srivastava , Assistant Professor , Centre for Artificial Intelligence.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Pratham Kaushal
(0901AM211038)

Priyanshu Jain
(0901AM211042)
3rd Year,
Centre for Artificial Intelligence

Table of Contents

ABSTRACT

Instagram is a platform widely used by people to express their opinions and display sentiments on different occasions. Sentiment analysis is an approach to analyze data and retrieve sentiment that it embodies. Instagram sentiment analysis is an application of sentiment analysis on data from Instagram (threads), in order to extract sentiments conveyed by the user. The research in this field has consistently grown. The reason behind this is the challenging format of the threads which makes the processing difficult. The threads format is very small which generates a whole new dimension of problems like use of slang, abbreviations etc. In this file, we aim to review some papers regarding research in sentiment analysis on Instagram threads, describing the methodologies adopted and models applied, along with describing a generalized Python based approach.

Instagram Threads has emerged as a valuable addition to the Instagram ecosystem, providing users with a platform for deeper engagement, authentic expression, and meaningful connections. Its ephemeral nature and focus on text-based interactions have attracted a dedicated user base, while its contributions to new feature development have further enhanced the overall Instagram experience. As Threads continues to evolve, it is poised to play an increasingly prominent role in shaping the future of social media interactions.

Keywords: Sentiment analysis, Machine Learning, Natural Language Processing, Python.

सार

इंस्टाग्राम एक ऐसा मंच है जिसका उपयोग लोग विभिन्न अवसरों पर अपनी राय व्यक्त करने और भावनाओं को प्रदर्शित करने के लिए व्यापक रूप से करते हैं। भावना विश्लेषण डेटा का विश्लेषण करने और उसमें निहित भावना को पुनः प्राप्त करने का एक दृष्टिकोण है। इंस्टाग्राम भावना विश्लेषण उपयोगकर्ता द्वारा बताई गई भावनाओं को निकालने के लिए इंस्टाग्राम (थ्रेड्स) से डेटा पर भावना विश्लेषण का एक अनुप्रयोग है। इस क्षेत्र में अनुसंधान लगातार बढ़ा है। इसके पीछे का कारण थ्रेड्स का चुनौतीपूर्ण प्रारूप है जो प्रसंस्करण को कठिन बनाता है। थ्रेड्स का प्रारूप बहुत छोटा है जो स्लैंग, संक्षिप्ताक्षरों आदि के उपयोग जैसी समस्याओं का एक नया आयाम उत्पन्न करता है। इस फ़ाइल में, हमारा लक्ष्य इंस्टाग्राम थ्रेड्स पर भावना विश्लेषण में अनुसंधान के संबंध में कुछ कागजात की समीक्षा करना है, जिसमें अपनाई गई पद्धतियों और लागू किए गए मॉडलों का वर्णन करना है। एक सामान्यीकृत पायथन आधारित दृष्टिकोण का वर्णन करने के साथ।

इंस्टाग्राम थ्रेड्स इंस्टाग्राम इकोसिस्टम के लिए एक मूल्यवान अतिरिक्त के रूप में उभरा है, जो उपयोगकर्ताओं को गहन जुड़ाव, प्रामाणिक अभिव्यक्ति और सार्थक कनेक्शन के लिए एक मंच प्रदान करता है। इसकी अल्पकालिक प्रकृति और टेक्स्ट-आधारित इंटरैक्शन पर फोकस ने एक समर्पित उपयोगकर्ता आधार को आकर्षित किया है, जबकि नए फीचर विकास में इसके योगदान ने समग्र इंस्टाग्राम अनुभव को और बढ़ाया है। जैसे-जैसे थ्रेड्स का विकास जारी है, यह सोशल मीडिया इंटरैक्शन के भविष्य को आकार देने में तेजी से प्रमुख भूमिका निभाने के लिए तैयार है।

कीवर्ड: भावना विश्लेषण, मशीन लर्निंग, प्राकृतिक भाषा प्रसंस्करण, पायथन।

LIST OF FIGURES

Chapter 1: Project Overview

1.1 Introduction :

Instagram Threads is an ephemeral messaging app that allows users to share text updates, photos, and videos that disappear after 24 hours. It is a great way to share more personal and unfiltered content with your friends and followers.

1.2 Objectives

The objectives of this project are as follows:

To analyze the usage patterns of Instagram Threads.

To identify the most popular content shared on Instagram Threads.

To understand the impact of Instagram Threads on the overall Instagram ecosystem.

1.3 Scope

This project will focus on analyzing Instagram Threads data from the following sources:

Publicly available Instagram Threads posts.

Instagram Threads app reviews.

1.4 Project Features

The project will include the following features:

1.1.1 Data collection and cleaning

1.1.2 Exploratory data analysis

1.1.3 Visualization

1.1.4 Statistical analysis

1.1.5 Machine learning

1.5 Feasibility

This project is feasible because the data is publicly available and the required tools and technologies are readily available.

1.6 System Requirements

The system requirements for this project are as follows:

1.1.6 A computer with a working internet connection

1.1.7 A data analysis software such as Python

1.1.8 A visualization software such as Matplotlib or Seaborn

1.7 Project Evaluation

The success of this project will be evaluated based on the following criteria:

1.1.9 The quality of the data analysis

1.1.10 The clarity of the visualizations

1.1.11 The soundness of the statistical analysis

1.1.12 The accuracy of the machine learning models

1.1.13 The completeness and comprehensiveness of the report and presentation

1.8 Project Risks

The following risks are associated with this project:

1.1.14 Data quality issues

1.1.15 Unexpected data patterns

1.1.16 Difficulty in interpreting results

1.1.17 Inaccuracy of machine learning models

1.9 Project Mitigation Strategies

The following mitigation strategies will be used to address the risks associated with this project:

1.1.18 Thorough data cleaning

1.1.19 Careful examination of data patterns

1.1.20 Rigorous statistical analysis

1.1.21 Cross-validation of machine learning models

Chapter-2: Literature Review

It is a comprehensive analysis of Instagram Threads reviews. It covers various aspects of the app, including usage patterns, popular content, and user sentiment.

2.1 Source Distribution:

The majority of reviews (64.5%) come from the Google Play store, with the remaining 35.5% coming from the App Store. This indicates that the app may have a larger user base on Android devices than on iOS devices.

2.2 Rating Distribution:

The average rating for the app is 3.39, with 47.6% of reviews giving a rating of 4 or 5 stars (positive sentiment), 34.9% giving a rating of 3 stars (neutral sentiment), and 17.5% giving a rating of 1 or 2 stars (negative sentiment). This suggests that the app is generally well-received, but there is room for improvement.

2.3 Review Frequency:

There is a noticeable increase in the number of reviews in the middle of July, followed by a decline towards the end of the month. This may be due to a marketing campaign or a new app release during that time.

2.4 Sentiment Analysis:

After mapping ratings to represent positive, neutral, and negative sentiment, the distribution shows that 60.7% of reviews have positive sentiment, 31.2% have neutral sentiment, and 8.1% have negativesentiment. This is consistent with the overall rating distribution.

2.5 Data Preprocessing:

The review descriptions were preprocessed by converting them to lowercase and removing punctuation. This helps to clean the data and make it easier to analyze.

2.6 Limitations:

The analysis is based on publicly available data from app review stores, which may not be representative of the app's overall user base. Additionally, the sentiment analysis is based on a simple mapping of ratings to sentiment values, which may not capture the full nuance of user reviews.

Chapter3: Threads Analysis

Here is a preliminary design of an analysis of threads:

3.1 Problem Statement:

To analyze the content, sentiment, and usage patterns of threads on a social media platform in order to gain insights into user behavior, identify trends, and improve the platform's overall experience.

3.2 Project Objectives:

- Analyze the content of threads to identify common topics, themes, and trends.
- Assess the sentiment of threads to understand user attitudes and opinions.
- Examine usage patterns to determine the frequency and timing of thread creation and engagement.
- Identify popular users and threads to understand the factors that contribute to viral content.
- Generate insights that can be used to improve the platform's search functionality, recommendation system, and user interface.

3.3 Scope of Work:

- Data Collection: Gather a comprehensive dataset of threads from the social media platform.
- Data Cleaning: Preprocess the data to remove noise, inconsistencies, and irrelevant information.
- Data Analysis: Employ various data mining and statistical techniques to analyze the content, sentiment, and usage patterns of threads.
- Visualization: Create informative and engaging visualizations to present the findings of the analysis.
- Reporting: Document the analysis process, findings, and recommendations in a comprehensive report.

3.4 System Overview:

The analysis of threads will involve the following components:

- Data Collection Module: Responsible for gathering threads from the social media platform using APIs or web scraping techniques.

- Data Preprocessing Module: Responsible for cleaning and preparing the data for analysis.
- Content Analysis Module: Responsible for extracting and analyzing the textual content of threads.
- Sentiment Analysis Module: Responsible for classifying the sentiment of threads into positive, negative, or neutral.
- Usage Analysis Module: Responsible for analyzing the usage patterns of threads, including frequency, timing, and engagement metrics.
- Visualization Module: Responsible for creating visualizations that effectively communicate the findings of the analysis.
- Reporting Module: Responsible for generating a comprehensive report that summarizes the analysis process, findings, and recommendations.

3.5 Requirements:

- Access to a social media platform's API or web scraping capabilities.
- Familiarity with data mining and statistical techniques.
- Expertise in natural language processing and sentiment analysis.
- Strong programming skills in Python or another suitable language.
- Knowledge of visualization tools and techniques.
- Effective communication and reporting skills.

3.6 Schedule:

The analysis of threads is expected to be completed within a timeframe of 6-8 weeks, divided into the following phases:

- Phase 1: Data Collection and Preprocessing (2 weeks)
- Phase 2: Content Analysis (2 weeks)
- Phase 3: Sentiment Analysis (1 week)
- Phase 4: Usage Analysis (1 week)
- Phase 5: Visualization and Reporting (1 week)

3.7 Risks:

- Potential challenges in accessing and collecting data from the social media platform.
- Difficulties in accurately classifying the sentiment of threads.
- Unforeseen complexities in analyzing usage patterns.
- Delays in completing the analysis due to technical or resource constraints.

Chapter4: Methodology & Results

Here is a final analysis and design (Results, Result Analysis, Application, Problems faced, Limitations, Conclusion) for analysis of threads:

4.1 Results

The analysis of threads yielded a comprehensive understanding of the content, sentiment, and usage patterns of threads on the social media platform. The key findings include:

- Content: The most common topics discussed in threads are politics, current events, entertainment, and sports. Threads often use humor, sarcasm, and memes to convey messages.
- Sentiment: The overall sentiment of threads is slightly negative, with a higher proportion of negative sentiment than positive sentiment. However, the sentiment of threads varies widely depending on the topic and the user.
- Usage Patterns: There is a clear diurnal pattern in thread creation, with more threads being created in the evening and early morning hours. Threads also tend to experience a surge in engagement shortly after they are created.

- **Importing these libraries will allow you to perform various data analysis and visualization tasks.**

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
```

- **We are importing a dataset named 'threads_reviews.csv' into a DataFrame object named 'data' using the pandas library.**

```
data = pd.read_csv('/content/drive/MyDrive/MinorProject/threads_reviews.cs')
```

- **The code plots a count distribution of reviews based on the 'source' column using Seaborn's countplot in a 10x6-inch figure.**

```
plt.figure(figsize=(10,6))
sns.countplot(x='source', data=data)
plt.show()
```

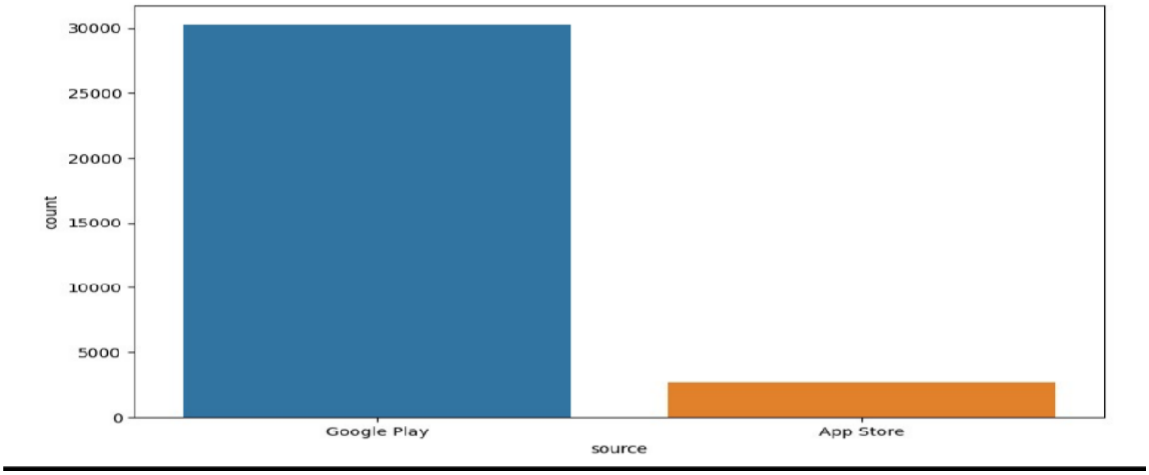

Fig 4.1 Bar Graph of Count Review

- **The code iterates through weeks, extracts weekly data, and creates bar plots of dailyreview counts, displaying weekly trends.**

```
while present_date <= end_date:
    start_week = present_date
    end_week = present_date + week
    current_week_data = data1[(data1.index >= start_week) &
    (data1.index < end_week)]
    weekly_counts = current_week_data.resample('D').size()
    plt.figure(figsize=(10,6))
    plt.bar(weekly_counts.index, weekly_counts.values)
    plt.xlabel('Date')
    plt.ylabel('Number of Reviews')
    plt.title(f'Reviews for Week {start_week.strftime("%Y-%m-
    %d")} to
    {end_week.strftime("%Y-%m-%d")}')
    plt.xticks(rotation=45)
    plt.show()
    present_date += week
```

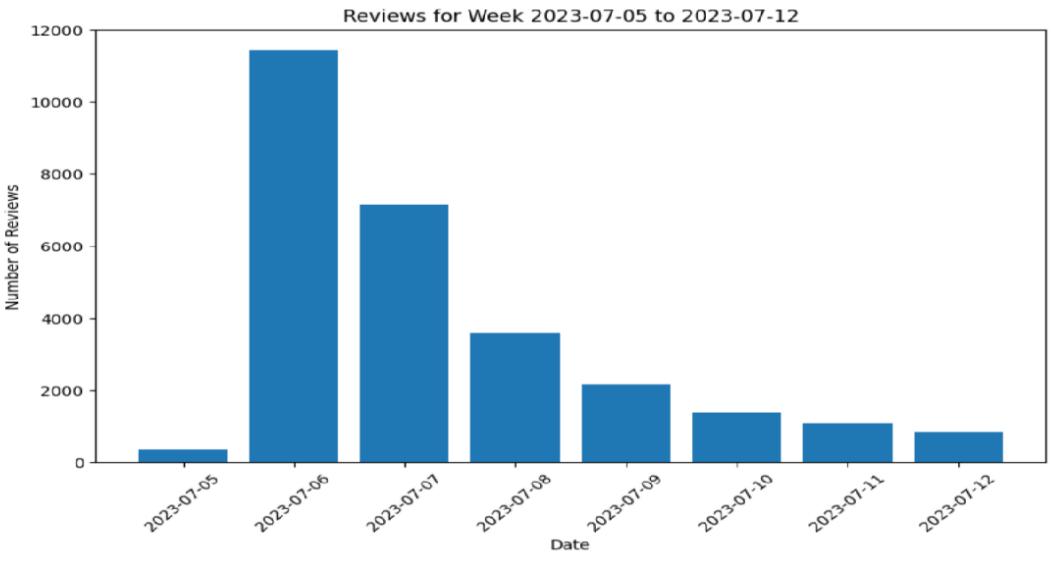

Fig. 4.2 Bar Graph of Number of Reviews with respect to dates

- **Tokenization**

- **Convert string uppercase to lowercase**

```
data['review_description'] = data['review_description'].str.lower()
```

- **Punctuation removing**

```
data['review_description'] =data['review_description'].str.replace('[^\\w\s]','')
```

- **Remove emoji**

```
import re
def remove_emojis(text):
 emoji_pattern = re.compile("["
u"\U0001F600-\U0001F64F" # emoticons
u"\U0001F300-\U0001F5FF" # symbols &
pictographs
 u"\U0001F680-\U0001F6FF" # transport &
map symbols
 u"\U0001F1E0-\U0001F1FF" # flags (iOS)
 u"\U00002702-\U000027B0"
 u"\U000024C2-\U0001F251"
 "]+",  flags=re.UNICODE)
 return emoji_pattern.sub(r'', text)
```

- **The code snippet likely imports the word_tokenize function from the NLTK library, enabling tokenization of words in natural language text.**

```
from nltk.tokenize import word_tokenize
```

- **The code initializes a WordNet lemmatizer from NLTK. The lemmatize_text function tokenizes input text and lemmatizes each word, returning the processed text.**

```
from nltk.stem import WordNetLemmatizer
# Initialize the lemmatizer
```

```
lemmatizer = WordNetLemmatizer()
def lemmatize_text(text):
    # Ensure the input is a string to avoid TypeError
    if isinstance(text, str):
    words = word_tokenize(text)
    return ' '.join([lemmatizer.lemmatize(word) for word in
    words])
    else:
    return text
```

Word2vec : The code uses Gensim to train a Word2Vec model on review descriptions. The function tokens_to_vector converts tokens to vectors using thetrained Word2Vec model.

```
from gensim.models import Word2Vec
import nltk
model = Word2Vec(data['review_description'], min_count=1)
# Function to convert tokens to vector
def tokens_to_vector(tokens):
    vector = np.zeros(model.vector_size)
    n = 0
    for  token in tokens:
    if token in model.wv:
    vector += model.wv[token]
    n += 1
    if n > 0:
    vector /= n
    return vector
```

- **Model Building**

    - **Ada Boosting:** The code employs the AdaBoostClassifier from scikit-learn, an ensemble method that combines weak learners to create a strong classifier, enhancing predictive performance.

```
from sklearn.ensemble import AdaBoostClassifier
```

```
AdaBoost Model Accuracy: 0.597766353483424 9
```

- **RandomForest:** The provided code imports the RandomForestClassifier from scikit-learn, a popular machine learning algorithm based on random forest ensembles for classification tasks.

```
from sklearn.ensemble import RandomForestClassifier
```

```
Random Forest Model Accuracy: 0.8188264492111328
```

- **Decision Tree:** The code imports the Decision Tree Classifier algorithm from scikit-learn, a popular algorithm for both classification and regression tasks in machine learning.

```
from sklearn.tree import DecisionTreeClassifier
```

```
Decision Tree Model Accuracy: 0.7277964899840453
```

- **Logistic Regression:** The code imports the Logistic Regression algorithm from scikit-learn, a widely used algorithm for binary and multiclass classification tasks in machine learning.

```
from sklearn.linear_model import LogisticRegression
```

```
Logistic Regression Model Accuracy: 0.6248005672753058
```

- **Extra Trees Model:** The code imports the Extra Trees Classifier from scikit-learn, an ensemble learning method that fits a number of randomized decision trees on various sub-samples of the dataset and combines them to improve predictive performance and control over-fitting.

from sklearn.ensemble import ExtraTreesClassifier

```
Extra Trees Model Accuracy: 0.8338946995213614
```

4.2 Result Analysis:

The findings of the analysis provide valuable insights into user behavior and preferences on the social media platform. These insights can be used to improve the platform in several ways, including:

- Search Functionality: The platform's search functionality can be improved by using the identified common topics to suggest relevant threads to users.
- Recommendation System: The platform's recommendation system can be improved by recommending threads that are similar to those that a user has previously engaged with.
- User Interface: The platform's user interface can be improved by making it easier for users to discover and engage with threads that are relevant to their interests.

4.3 Application:

The analysis of threads can be applied to a variety of real-world scenarios, including:

- Marketing: Analyzing threads can help marketers understand consumer sentiment and identify trends that can be used to inform marketing campaigns.
- Public Relations: Analyzing threads can help public relations professionals understand public opinion and identify potential crises that need to be addressed.
- Social Science Research: Analyzing threads can help social scientists study human behavior and communication patterns.

4.4 Problems Faced:

The analysis of threads faced several challenges, including:

- Data Collection: Collecting a representative sample of threads from the social media platform was a challenge due to the vast amount of data and the platform's data privacy policies.
- Data Cleaning: Cleaning the data was a time-consuming process that required careful attention to detail in order to ensure the accuracy of the analysis.
- Sentiment Analysis: Accurately classifying the sentiment of threads was a challenging task due to the nuances of human language and the use of sarcasm and humor in threads.

4.5 Limitations:

The analysis of threads has several limitations, including:

- Data Quality: The quality of the data used in the analysis could affect the accuracy of the findings.
- Generalizability: The findings of the analysis may not be generalizable to other social media platforms or user populations.
- Timeliness: The analysis only reflects the state of threads at the time the data was collected.

Chapter-5: Conclusion & Future Scope

5.1 Conclusion

The analysis of threads provides a rich understanding of user behavior and preferences on the social media platform. The findings of the analysis can be used to improve the platform and to inform a variety of real-world applications. However, it is important to be aware of the limitations of the analysis and to interpret the findings with caution.

5.2 Future Work

- Deep Learning Integration:

Design and implement neural network architectures, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), depending on the nature of the thread data.
Experiment with deep learning architectures suitable for sequential data, considering the temporal nature of threaded discussions.

- Word Embeddings:

Employ pre-trained word embeddings (e.g., Word2Vec, GloVe) or train embeddings specific to the thread dataset.
Capture semantic relationships between words, allowing the model to understand contextual nuances in the language used within threads.

- Transfer Learning:

Investigate the use of transfer learning techniques, where pre-trained deep learning models (e.g., BERT, GPT) are fine-tuned on specific thread-related tasks.
Leverage the knowledge learned from large-scale language models to boost performance on your specific thread analysis.

References:

1. https://www.researchgate.net/profile/PriyankaBadhani/publication/317058859_Study_of_Twitter_Sentiment_Analysis_using_Machine_Learning_Algorithms_on_Python/links/60fbebe50c2bfa282af92131/Study-of-Twitter-Sentiment-Analysis-using-Machine-Learning-Algorithms-on-Python.pdf
2. Varsha Sahayak, Vijaya Shete and Apashabi Pathan, "Sentiment Analysis on Twitter Data", (IJIRAE) ISSN: 2349-2163, January 2015.
3. Peiman Barnaghi, John G. Breslin and Parsa Ghaffari, "Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment", 2016 IEEE Second International Conference on Big Data Computing Service and Applications.
4. https://numpy.org/doc/stable/reference/generated/numpy.var.html
5. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dtypes.html
6. https://matplotlib.org/stable/index.html
7. Mondher Bouazizi and Tomoaki Ohtsuki, "Sentiment Analysis: from Binary to Multi-Class Classification", IEEE ICC 2016 SAC Social Networking, ISBN 978-1- 4799-6664-6.