**УНІВЕРСИТЕТ ІМЕНІ АЛЬФРЕДА НОБЕЛЯ**

**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

# КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

# на тему

# *“Розробка інформаційної системи автоматичного аналізу коментарів на наявність позитивної та негативної тональності”*

Виконав: здобувач 4 курсу, групи КН-18-2

Спеціальності

\_\_\_\_\_\_\_122 Комп’ютерні науки\_\_\_\_\_\_\_\_

*шифр і назва*

\_\_\_\_\_\_\_\_\_\_\_\_\_Кущ Д.Д.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

прізвище, ініціали

Керівник: \_\_\_\_\_Рижков І. В., Доктор технічних наук, доцент\_\_\_\_\_\_\_

прізвище, ініціали, науковий ступінь, вчене звання

### м. Дніпро

### 2022

**УНІВЕРСИТЕТ ІМЕНІ АЛЬФРЕДА НОБЕЛЯ**

Кафедра інформаційних технологій

Освітній рівень БАКАЛАВР

Спеціальність 122 «Комп’ютерні науки»

Затверджую:

Зав. кафедри\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барташевська Ю.М., к.е.н, доц. “\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_р.

### ЗАВДАННЯ

### на кваліфікаційну роботу здобувачу

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кущу Даніілу Дмитровичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*прізвище, ім’я та по батькові*

# 1. Тема роботи *“*Розробка інформаційної системи автоматичного аналізу коментарів на наявність позитивної та негативної тональності”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Керівник роботи роботи Рижков Ігор Вікторович, Доктор технічних наук, доцент\_\_\_\_\_\_\_\_\_\_

*(прізвище, ім’я та по батькові, науковий ступінь, вчене звання*

затверджені наказом від «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р., № \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Термін здачі здобувачем закінченої роботи \_\_\_\_\_5 червня\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Цільова установка та вихідні дані до роботи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Зміст роботи (перелік питань, які належить розробити)\_\_ВСТУП, ОСНОВНА ЧАСТИНА(РОЗДЛІ 1, РОЗДІЛ 2) ВИСНОВКИ, ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ, ДОДАТКИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Консультанти розділів роботи:

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Консультант (прізвище, ініціали, посада) | Підпис, дата | |
| Завдання видав | Завдання прийняв |
| 1 | Рижков І.В., Доктор технічних наук, доцент |  |  |
| 2 |  |  |
| 3 |  |  |

1. Дата видачі завдання\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Календарний план виконання роботи

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів кваліфікаційної роботи | Термін виконання етапів роботи | |
| За планом | Фактично |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Здобувач \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_Кущ Д.Д.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*підпис прізвище, ініціали*

Керівник кваліфікаційної роботи \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_Рижков І.В.\_\_\_\_\_\_

*підпис прізвище, ініціали*

А Н О Т А Ц І Я \*

Кущ Д.Д. Розробка інформаційної системи автоматичного аналізу коментарів на наявність позитивної та негативної тональності.

Метою дипломної роботи є створення та розробка програмного продукту, що автоматизує процес пошуку позитивної та негативної тональності в коментарях. В результаті виконання дипломної роботи було зроблено програмний продукт, який аналізує оцінку продукту через автоматизований пошук емоційно забарвленої лексики, що відображає ставлення коментаторів до певного продукту або явища.

Ключові слова: пошук, тональність, дані, стоп-слова, коментарі.

SUMMARY

Kushch D.D. Development of an information system for automatic analysis of comments on the presence of positive and negative tones.

The purpose of the thesis is to create and develop a software product that automates the process of finding positive and negative tones in the comments. As a result of the thesis, a software product was made that analyzes the evaluation of the product through an automated search for emotionally colored vocabulary that reflects the attitude of commentators to a particular product or phenomenon.

Key words: search, tonality, data, stop words, comments.

**ЗМІСТ**

1.Вступ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6

2. РОЗДІЛ 1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Сентимент аналіз: поняття та види\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

1.2 Проблеми роботи з текстовими даними\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11

3. РОЗДІЛ 2 ПРАКТИЧНА ЧАСТИНА

2.1 Створення проекту\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_13

2.2 Видалення символів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_15

2.3 Видалення стоп-слів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_18

2.4 Пошук слів з тональністю \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_23

2.5 Аналіз отриманих даних \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_30

5. Висновок \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_41

6. Джерела \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_42

**ВСТУП**

Internet - це глобальна комп'ютерна мережа, що охоплює весь світ. І ця мережа збільшується щодня, а якщо бути точніше, то на 7-10% щомісяця. Він став глобальним ядром, в якому багато установ передають та обробляють інформацію.

Кожен з нас використовує смартфони, планшети, ноутбуки та комп’ютери, щоб підтримувати зв’язок зі знайомими чи друзями. Ми зв’язуємося з ними, щоб поговорити про те, що ми бачимо чи відчуваємо, купуємо чи продаємо, чи хочемо від них почути. Усі ці відповідності об’єднує спільна риса – наявність емоцій. Це показує наше ставлення до чогось чи когось. Ми не помічаємо, як ці думки аналізуються і як ми формуємо своє ставлення до подій, речей чи людей. Хоча ми не помічали їх важливості, ця інформація може бути дуже корисною для медіа-ресурсів.

Величезна кількість даних, які люди щодня створюють у соціальних мережах, існує і буде тільки збільшуватися. Якби ви робили цей аналіз самостійно, це зайняло б багато часу, і нові дані з’являлися б набагато швидше, ніж ваша робота. Тому виникає нагальна потреба в автоматизації цих процесів з використанням великих центрів обробки даних, які зберігають та обробляють отриману інформацію. Процес вилучення та обробки даних називається Data Mining і в нашому випадку ми будемо використовувати текстовий аналіз даних. Text Data Mining — це обробка неструктурованих текстових даних з метою їх подальшої обробки в більш зручній формі.

З кожним роком необхідність аналізувати дані тільки збільшується. Більше за інших в цьому зацікавлені соціальні медіа або онлайн магазини. Вони хочуть краще розуміти що потрібно користувачам, дізнатися їх думки, реакцію щодо різноманітних подій, об'єктів та інше. Ви могли помітити як шукаєте щось в інтернеті і через деякий час бачите рекламу, яка пропонує вам послуги або товар які схожі на те, що вам щойно хотілося знайти.

Данні бувають найрізноманітніші. Для соціальної мережі як Instagram, головним типом контенту буде фото, для YouTube – відео, джерелом інформації для Twitter буде текст. Для цієї курсової роботи будуть розглянуті саме тексти.

Ціль цієї роботи показати як можна здійснити сентимент-аналіз думок тексту за допомогою Python як саме це працює.

**РОЗДІЛ 1 ТЕОРЕТИЧНА ЧАСТИНА**

**1.1 Сентимент аналіз: поняття та види**

Аналізом тексту на сентименти і думки називається - сентимент-аналіз. Необхідно перевірити зміст суб’єктивних висловлювань у тексті на тональність змісту. Для більшого розуміння і розкриття визначення сентимент-аналіз можна привести такі його різновиди:

- Сентиментометрія (sentiment metrics);

- Брендмоніторинг (brand monitoring);

- Соціальний медіа аналіз (social media analysis);

- Розвідка думок (opinion mining);

- «підслуховування» думок (opinion listening);

- Аналіз тональності тексту.

Для відгуків також присутня тональність. Тональність – це фундамент усіх сентимент-аналізів. Тому що вона найкраще відображає думки автора і його настрій на відміну від “сухих” даних із новин або наукових книжок, які в першу чергу повинні якомога швидше і як найбільше надати даних. Тональність прийнято поділяти на три основних параметра:

- автор тексту

- тональна оцінка

- тема тексту

Тональність в свою чергу поділяється на позитивну і негативну оцінки, іноді ще виділяють нейтральну тональність. Для більшого розуміння як виглядає позитивна і негативна тональності можна подивитися в табл. 1.

Таблиця 1

Приклади тональності в тексті

|  |  |
| --- | --- |
| **Позитивна тональність** | **Негативна тональність** |
| “Мені так **сподобалось** це морозиво!” | “Ця робота мене **виснажує**” |
| “Це було **цікаве** завдання” | “Цей колір **поганий**” |
| “В цьому місті є **гарний** парк, щоб **приємно** відпочити” | “Як вони могли **програти**?” |
| “Я вважаю, що він **добрий** та **щедрий**” | “Я це **ненавиджу**!?” |
| “В нас були **чудові** вихідні” | “Мій настрій **зіпсували** ще зранку” |

Як ви могли помітити, для колонки с позитивними відгуками можна виділити ключові слова-маркери:

- сподобалось

- гарно

- цікаво

- чудово

Ці слова краще дають нам зрозуміти ставлення автора до певного об'єкту чи людини. В розглянутому випадку реакція була позитивна.

Теж саме можна сказати і про слова з протилежним емоційним забарвленням, а саме з негативним:

- поганий

- програти

- виснажує

Наведені вище слова мають емоційний підтекст і дають негативну оцінку явища чи предмета.

Цей приклад проливає більше світла на те, як великі компанії можуть визначити, що ми хочемо передати у своєму відгуку.

**1.2 Проблеми роботи з текстовими даними**

Розглянемо основні проблеми роботи з текстовими даними

1. Очистка текстів від символів
2. Видалення стоп-слів
3. Побудова моделей

Розмітка текстів, очистка їх від непотрібних символів, видалення стоп-слів та виправлення помилок потрібні нам для більш зручної роботи з великими об’ємами даних та фінальний фільтрований вид даних значно легше сприймається машиною і сильно збільшує ефективність її роботи.

Стоп-слова (stop-words) – це слова які не мають ніякого значення під час сентимент-аналізу. Для прикладу, в українській мові, “а”, “в”, “та”, можуть бути визначеними як стоп-слова. Недоліком використання стоп-слів – це складність знайти усі стоп-слова в тексті. Якщо база даних постійно буде збільшуватися, то і набір стоп-слів теж потрібно буде постійно змінювати і додавати нові, проте використання їх зі статичними даними абсолютно можливе і досить популярне. В цій роботі буде розглянуто відбірку стоп-слів з використанням спеціальної бібліотеки призначеної саме для роботи з текстом і без необхідності створювати списки власноруч.

Також треба прибрати різні граматичні форми одного слова. Таким чином, після усіх цих змін мі отримаємо сухий текст без зайвих деталей, які могли б нас відволікати. Це буде мати вид великого набору слів, які розділені між собою за допомоги пробілів. Саме це нам і буде потрібно в подальшій роботі. Страшно уявити скільки часу та людей було б потрачено на обробку даних отриманих лише за одну добу з однієї соціальної мережі, якби це робилося власноруч.

Після усіх цих фільтрів, нам потрібно побудувати графіки або діаграми. Під час обробки отриманих нами даних, вони перетворюються в векторні форми. В графіках ми можемо задавати, що ми хочемо знайти. Графік може основуватися на кількості слів в коментарях та на самій кількості цих коментарів, що покаже нам яка довжина коментаря була найпопулярнішою. Таким чином можна побудувати величезну кількість подібних діаграм, які буду показувати потрібні нам залежності від тих чи інших даних. Головне під час таких розрахунків правильно налаштувати усі фільтри та передати саме ті дані, що нам потрібно, інакше ми хоч і отримаємо графік, проте він буде помилковим і не дасть нам правильного результату на сентимент-аналіз тексту.

**2.1 Створення проекту**

Роздивимось на практиці як проходить сентимент-аналіз тексту на основі 40000 відгуків з Amazon на англійській мові. Для початку складемо план дій для сентимент-аналізу тексту.

**План:**

1. Створюється новий проект на Python;
2. Додається база відгуків;
3. Звернення до бази даних з метою прочитати дані і зробити фільтрацію від символів і стоп-символів;
4. Пошук слів з тональністю;
5. Сентимент-аналіз на основі отриманих даних.

Запускаємо Python та створюємо новий проект див рис. №2.1.

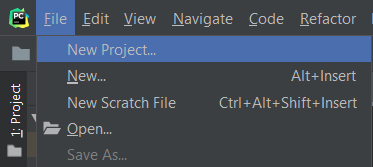


Рис. №2.1 Створення проекту на Python

Тепер нам потрібно додати дані з відгуками у форматі .txt до папки де зберігається щойно створений проект див. рис. №2.2. Нові файли з коментарями слід додавати саме таким чином, як показано на рисунку. Оскільки код налаштовано саме для обробки тих файлів, що знаходяться в одній папці з проектом і повинні задовольняти таким умовам:

* Бути на англійській мові
* Бути у форматі .txt

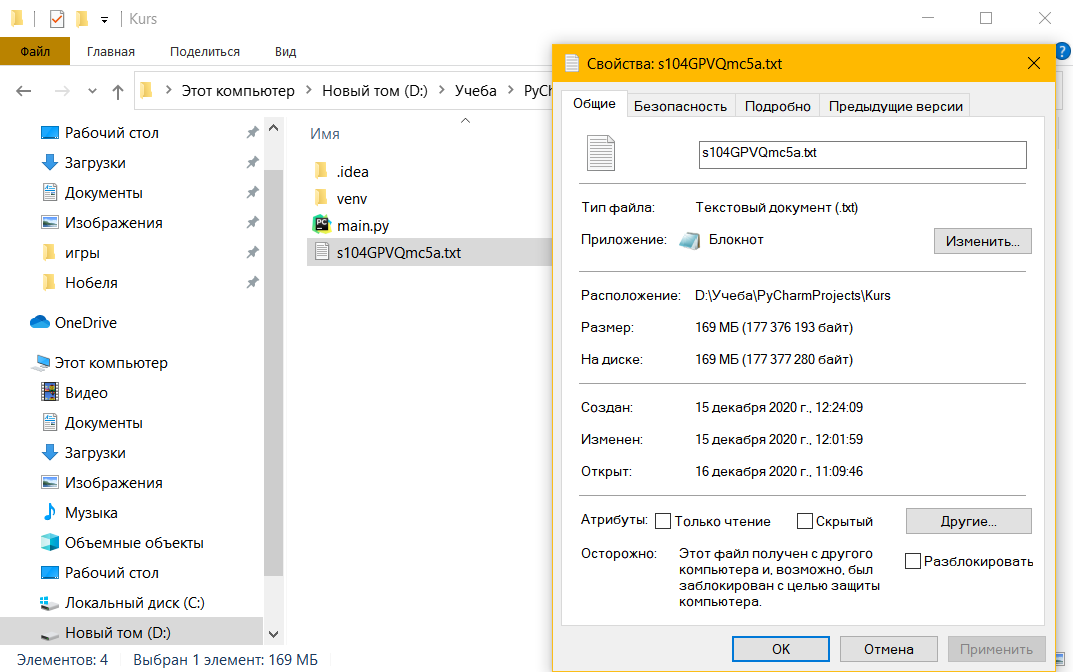


Рис. №2.2 Файл з відгуками у папці проекту.

**2.2 Видалення символів.**

Проект було створено і в ньому вже знаходиться файл з відгуками з Amazon. Даль потрібно видалити усі символи з тексту.

Імпортувавши “re” можна почати роботу з регулярними виразам. Регулярні вирази дають можливість звернутися до кожного слова в тексті за заданими користувачем особливостями. Такими особливостями можуть бути окремі символи. Наприклад в e-mail адресах присутній знак “@”, а номер телефону починається з “+”. А якщо потрібно відфільтрувати користувачів номерів однієї країни від користувачів іншої, то можна вказати перші декілька символів номеру, які характерні для тієї чи іншої країни. Вхідний файл з відгуками див. рис. №2.3.

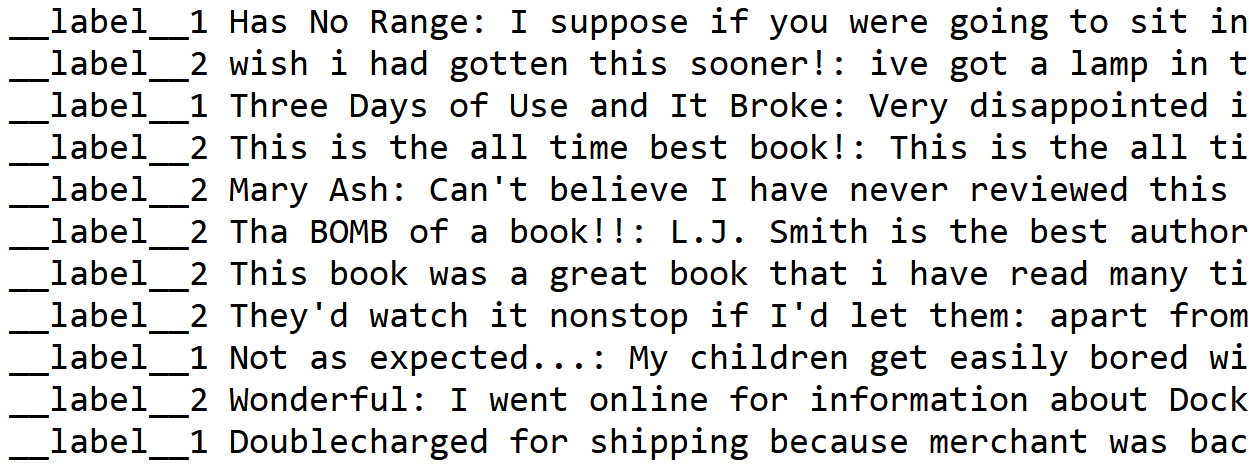


Рис. №2.3.Вхідні дані

Використовуючи регулярні вирази можна прибрати усі наявні в тексті символи. Повністю готовий пошук зайвих слів та символів для видалення їх з тексту виглядатиме наступним чином див. Рис. №2.4.

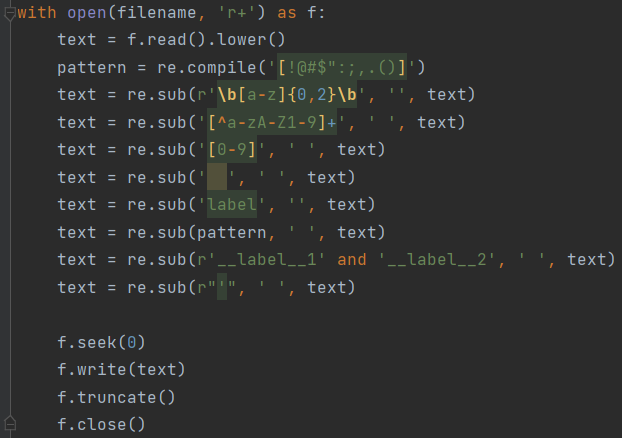


Рис. №2.4 Регулярні вирази.

filename – повна назва файлу з відгуками (завжди треба вказувати формат файлу).

with open – цей блок коду отримує назву файлу і потім нам потрібно визначитися для чого він викликається. Це також відбувається у дужках і в нашому випадку використовується ‘r+’ – це означає, що ми можемо прочитати наші дані і при необхідності записати в них щось, якби був просто ‘r’, то можна було б тільки читати файл і нічого не змінювати в ньому, а ‘w’ – тільки запис. Закінчується рядок тим, що було відкрито файл з відгуками як f.

Відкриваємо файл для читання, потім в команду re.sub вписуємо символи, які нам потрібно прибрати в тексті. Принцип роботи в тому, що за допомоги регулярного виразу знаходяться усі збіги з введеним символом і усі варіанти, що підходять під запит ми видаляємо і перезаписуємо отримані дані замість тих, що були раніше. Результатом роботи цього коду буде див. Рис. №2.5.

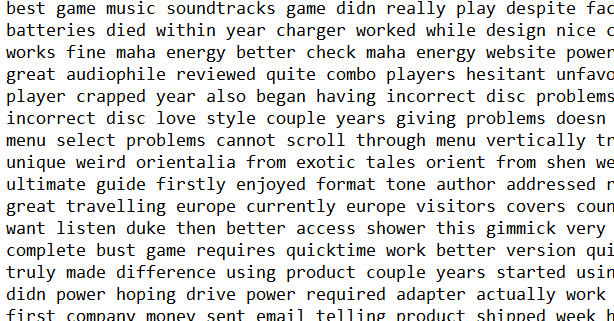


Рис. №2.5 Відгуки після видалення усіх символів.

На Рис. №2.4 відгуки до фільтрації, а на Рис. №2.5 вже повністю відфільтровані. Для більшої зручності регістр тексту коментарів було понижено заздалегідь. Таким чином пошук збігів зробити значно легше через порівняння однакових регістрів не дублюючи слова з тональністю для кожного регістру або робити зміну регістру з однієї бази слів, що сильно уповільнить роботу з даними.

**2.3 Видалення стоп-слів.**

Раніше було ознайомлення зі стоп-словами і для чого вони потрібні. Аналіз нашого тексту і будь-якого іншого починається з видалення цих слів. Це можна виконати по-різному і мною було обрано варіант з використанням бібліотеки nltk див. рис. №2.6.

Бібліотека nltk (Natural Language Toolkit) дає широкий функціонал для роботи з даними людської мови і включає в себе набір бібліотек обробки тексту для класифікації, виділення коренів, тегів, синтаксичного аналізу та семантичного аналізу, оболонки для промислових бібліотек NLP та активний дискусійний форум. Саме з нею і будемо працювати.

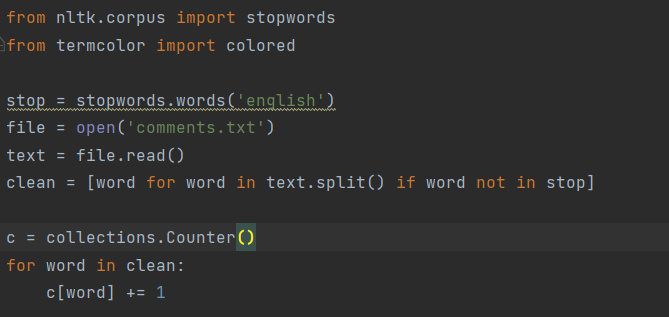


Рис. №2.6 Видалення стоп-слів.

Тут ми обираємо потрібну нам мову(в нашому прикладі це англійська), потім звертаємося до файлу з текстом і відкриваємо його для читання. Далі, використовуючи генератор, ми порівнюємо базу стоп-слів зі словами в тексті і якщо перемежувань не відбувається, то слова з тексту потрапляють у list(список) clean. Потім використовуємо модуль collections для підрахунку кількості повторів для кожного окремого слова в отриманому list clean. Результат усіх цих дій у консолі виглядатиме як показано на рис. №2.7.

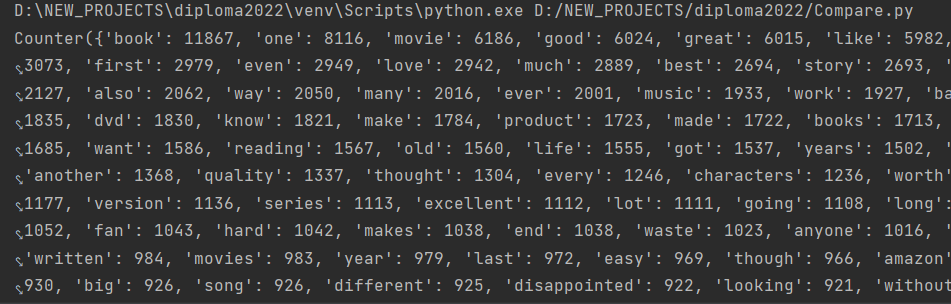


Рис. №2.7 Кількість повторів для кожного слова.

Усе працює як слід і ми маємо словник який містить пару слово/кількість повторів.

Зараз ще докладніше роздивимось чому прибирати стоп-слова необхідно, наскільки це вигідно і який має вплив на швидкість виконання.

По-перше для машини дуже важливо яка кількість даних поступає на обробку в центральний процесор, адже його швидкість розрахунків має якийсь ліміт. Тобто, чим менше об’єм інформацій з яким ми працюємо, тим менше часу на виконання потрібно комп’ютеру.

Великі бази даних, що було відфільтровано легше десь зберігати. В нашому випадку, файл до вилучення стоп-слів важив 8.67 мб, а після фільтрації розмір вдалося зменшити до 5.02 мб див. Рис. №2.8. В цих масштабах це мала різниця, але співвідношення дуже хороше, а це означає, що при розмірі файлу в 1гб його можна було б зменшити до 580мб, а це більше ніж 40%!

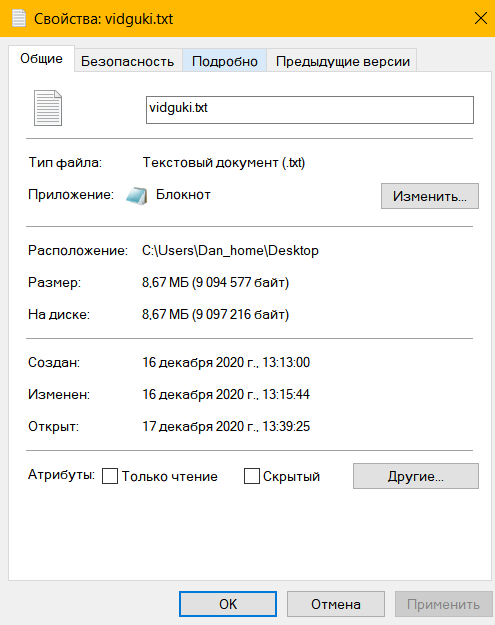
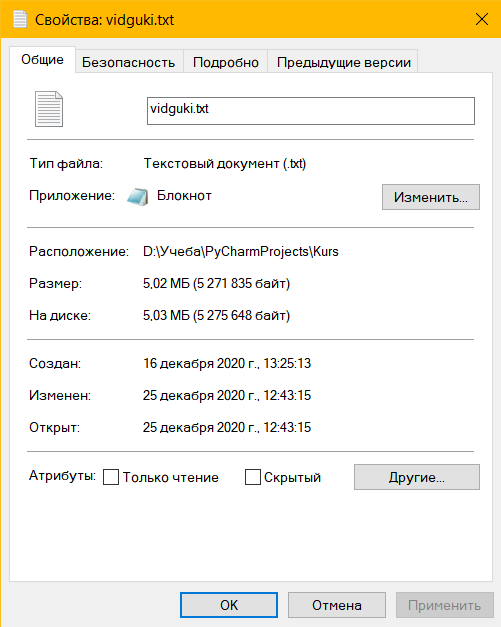
**** ****

Рис. №2.8 Порівняння розмірів файлів.

Іншим великих плюсом для нас буде швидкість виконання коду. І тут для замірів швидкості використовується раніше підключений модуль time. Відкриваємо проект на Python та імпортуємо два модулі див. рис. №2.9.

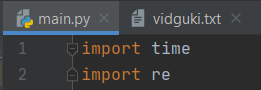


Рис. 2.9 Імпорт модулю time.

Модуль time потрібен для визначення швидкості роботи програми. Його принцип роботи можна порівняти з секундоміром. Він складається з двох частин:

- Перша частина розміщується в тому місці, з якого ми хочемо почати рахувати час виконання коду див. рис. №2.10.

- Друга частина розміщується в тому місці, де ми хочемо зупинити наш секундомір див. рис. №2.11.

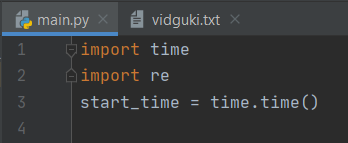


Рис. №2.10 Оголошення початку.

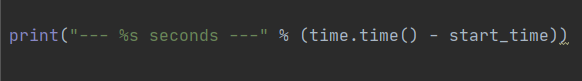
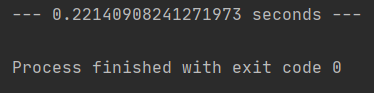


Рис. №2.11 Оголошення кінця.

В нашому випадку будемо заміряти швидкість усього коду, тому друга частина буде в кінці коду.

З неочищеними даними швидкість була 0.210 – 0.230 секунди, а після очистки швидкість проходу по даним стала 0.09 – 0.100, а це означає, шо швидкість стала більш ніж в два рази вищою ніж раніше! Див. Рис. №2.12.

****

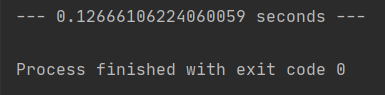
****

Рис. №2.12 Порівняння швидкості виконання коду

Тепер можна на прикладі виконаної роботи по видаленню зайвої інформації побачити яку важливу роль займає фільтрація тексту від зайвих слів, символів та стоп-слів.

**2.4 Пошук слів з тональністю.**

Дані відфільтровано. Наступним кроком буде пошук слів які мають тональність. В англійській мові також є однокореневі слова і їх також слід отримати. Мною було знайдено дві бази слів різних за тональністю, які включають в собі усі існуючі слова для кожної із тональностей див. Рис. №2.13 та Рис. №2.14.



Рис. №2.13. Слова з позитивною тональністю.

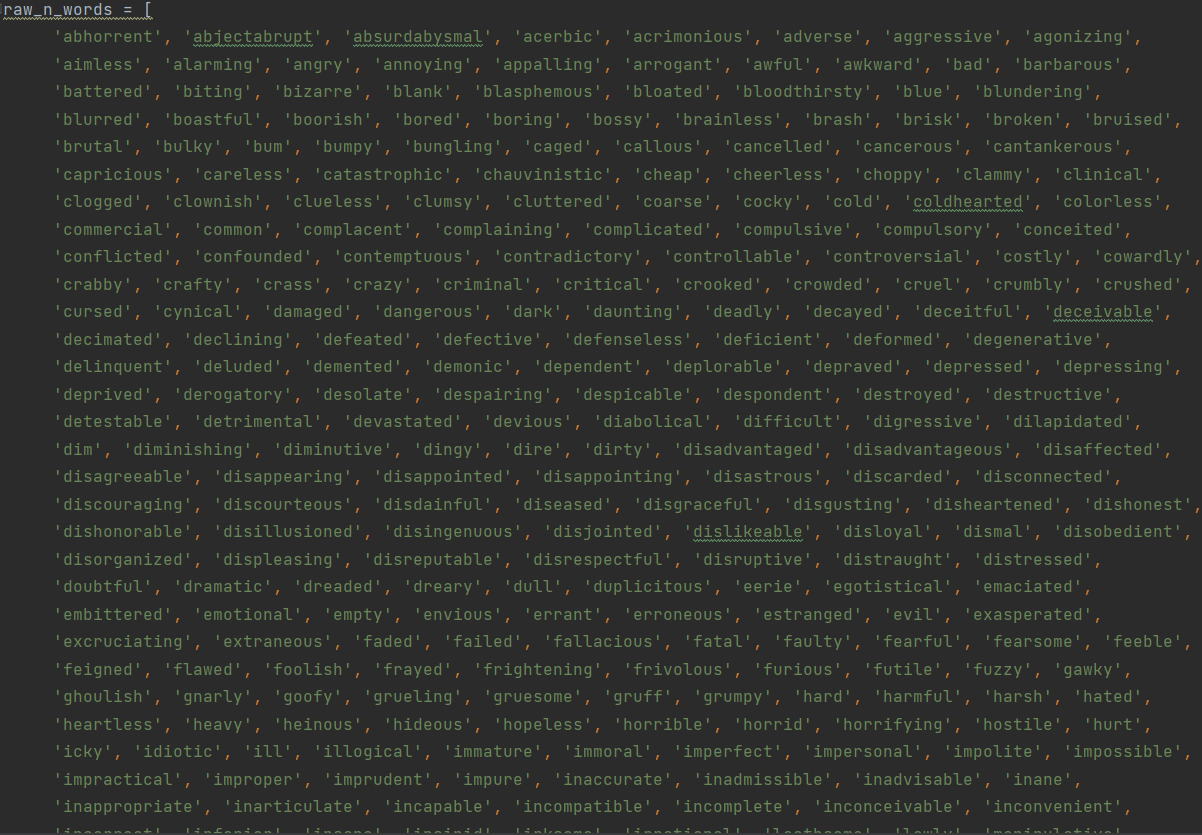


Рис.№2.14. Слова з негативною тональністю.

Для прикладу беремо слово “sad” і шукати треба не тільки саме слово “sad”, а й однокореневі слова як “sadness” або “sadly”. Для того, щоб виконати ці операції з пошуку слів знов було використано collections для підрахунку збігів і бібліотеку nltk, але вже для пошуку слів з однаковим коренем через PorterStemmer, який автоматично знаходить і зберігає корені слів див. Рис. №2.15. І створюємо для нього об’єкт.

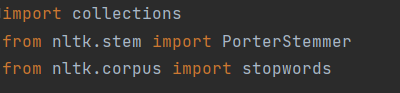




Рис. №2.15. Імпорт PorterStemmer та створення для нього об’єкту.

Тепер потрібно записати усі слова з останньої версії коментарів, але зберігати у словник будемо тільки корені кожного слова і кількість їх повторів див. Рис. №2.16.

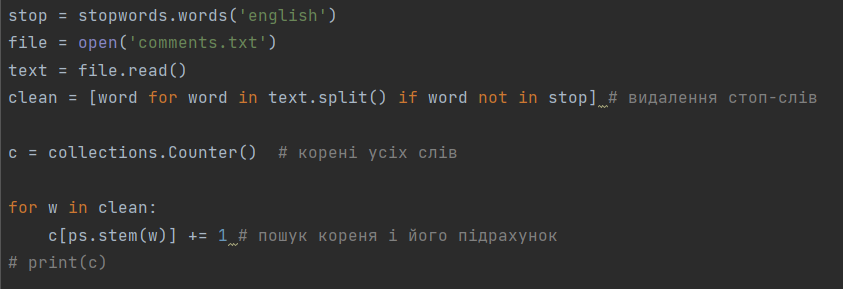


Рис. №2.16. Пошук коренів слів.

Таку саму операцію потрібно зробити і для слів з тональністю див. Рис. №2.17.

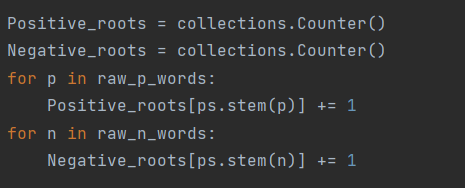


Рис. №2.17. Вилучення кореня в словах з тональністю.

Отримані результати для кожної з тональностей записуються у свої змінні у вигляді слово/кількість повторів.

Після цього коментарі і емоційно забарвлені слова знаходяться в однакових умовах для подальших операцій над ними.

Зробивши видалення символів, стоп-слів та приведення відгуків і слів тональністю до єдиного вигляду, звертаємося до відгуків для пошуку в них збігів див. Рис. №2.18.

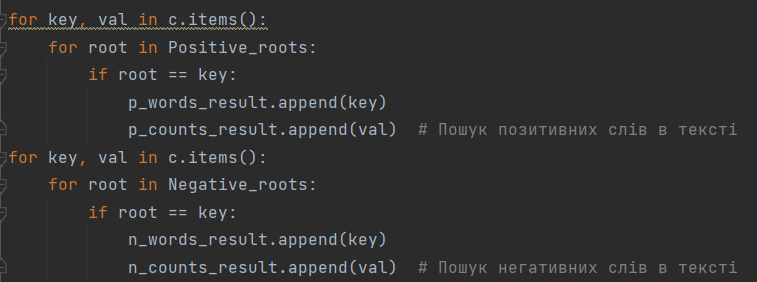


Рис. №2.18. Пошук слів з тональністю в коментарях.

Відбувається перевірка таким чином, що ми порівнюємо ключ(слово) серед усіх слів з елементом списку Positive\_roots(коренем слова з позитивною тональністю) і якщо знаходимо збіг, то додаємо лише це слово з його кількістю повторів до таких змінних:

* **p\_words\_result** - слова з позитивною тональністю;
* **p\_counts\_result** - кількість слів з позитивною тональністю;
* **n\_words\_result** - слова з негативною тональністю;
* **n\_counts\_result** - кількість слів з негативною тональністю.

Ті слова, які перевірку не пройшли – будуть ігноруватися і до попередніх змінних передаватися не будуть.

Для початку нам потрібно знову звернутися до даних. В нашому випадку ми звертаємося до сливнику “c” позначивши ключ(слово) та значення(кількість повторів цього слова в тексті). Наступним кроком ми звертаємося до бази слів з негативною тональністю і так само потім звертаємося до бази з позитивною тональністю див. Рис. №2.19.

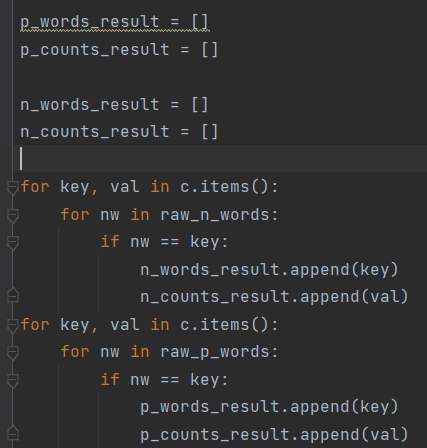


Рис. №2.19. Пошук слів з тональністю

Усі слова з тональністю було знайдено і тепер треба підготувати отримані дані для побудови графіків. Зараз слова знаходяться в хаотичному вигляді, а зручніше було б відсортувати їх по кількості збігів і тим самим створити плавний графік від найбільш вживаних слів до найменш вживаних. Для цього потрібно зв’язати списки p\_words\_result та p\_counts\_result і так само n\_words\_result та n\_counts\_result в два словники див. Рис. №2.20.

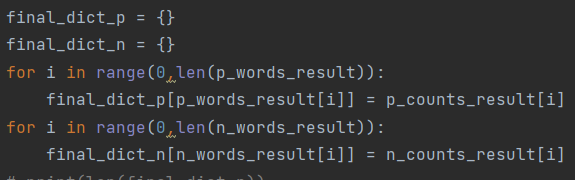


Рис. №2.20. Створення словників для кожної тональності.

Після цього потрібно зробити сортування словнику за значенням(кількості збігів) див. Рис. №2.21.

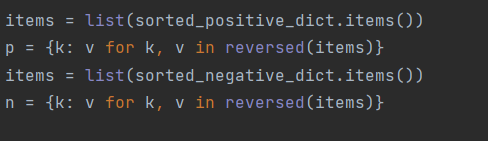


Рис. №2.21. Сортування словників.

Сортування відбувається у порядку від найменшого числа до найбільшого і тому потрібно розвернути його командою reversed як показано на рисунку. Таким чином і буде отримано правильний порядок. Перевіримо правильність виконання коду див. Рис. №2.22.



Рис. №2.22. Перевірка коректності роботи коду.

На рисунку можна побачити два рядки де першим числом(поза квадратними скобками) завжди виступає розмір отриманого словника. Перши рядок показує кількість позитивних слів(було знайдено 146 слів), а наступний рядок – кількість негативних(було знайдено 261 слово). Розміри списків було виведено для швидкої перевірки чи всі дані знаходяться у відповідних списках і чи нема розбіжностей, адже за наявності розбіжностей у розмірах буде неможливим створення графіків і діаграм у майбутньому. У самих квадратних дужках можна побачити як відсортувався кожний словник.

Після успішного пошуку і усіх розрахунків ми отримали достатню кількість даних і вже на основі цих даних можна буде зробити сентимент-аналіз.

**2.5 Аналіз отриманих даних.**

Після численних фільтрацій та пошуку слів, що мають тональну насиченість можна об’єднати отримані дані та побудувати діаграми або графіки, щоб провести сентимент-аналіз відгуків. Для початку потрібно імпортувати необхідні модулі і бібліотеки див. Рис. №2.23.

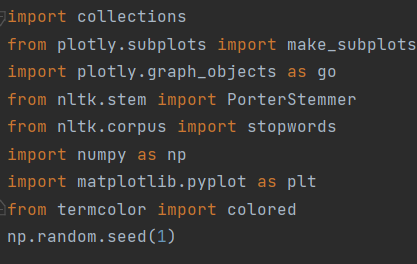


Рис. №2.23 Імпортовані бібліотек і модулів.

Тепер потрібно ввести дані в DataFrame, та перед цим вказати стиль відображення. Його можна імпортувати в готовому вигляді див. Рис. №2.24.

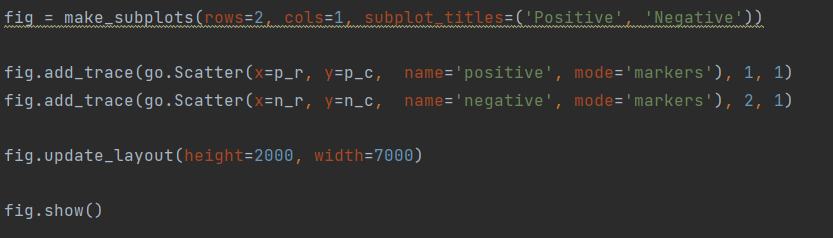


Рис. №2.24. Створення графіків.

В коді можна побачити як створюються два графіки, де в кожному з них знаходяться слова і їхня кількість збігів в тексті. Також в цих рядках вказано вид графіку “Scatter”, назва графіку та режим поміток “markers”, який буде відображатися у вигляді точок. Останні два числа показують як саме будуть розташовані майбутні графіки. Передостанній рядок потрібен для збільшення масштабу, яке за замовчуванням відображає будь-яку інформацію тільки у повноекранному режимі, а за наших обставин стандартних значень буде недостатньо і потребує збільшення. Індекси слів прив’язано до індексів чисел таким чином, щоб перше слово відповідало першому числу і так до кінця.

Після запуску можемо спостерігати діаграму складену на основі слів з позитивною та негативною тональністю див. Рис. №2.25.

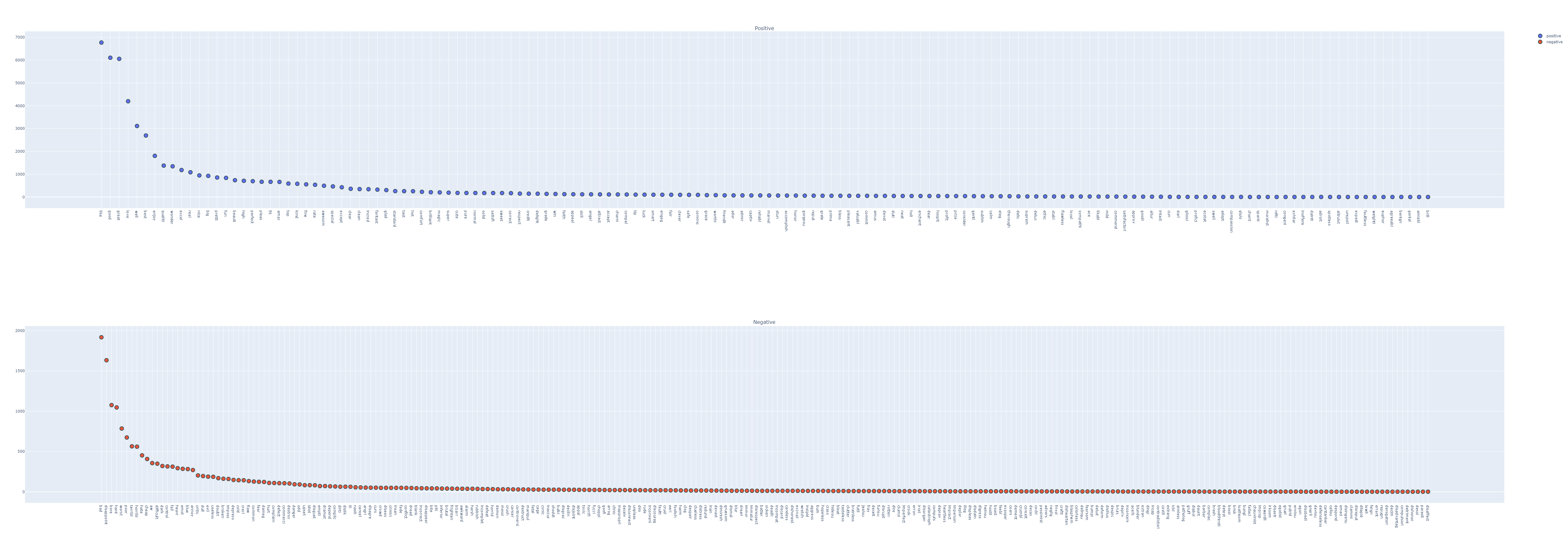


Рис. №2.25 Графіки слів з тональністю.

На рисунку діаграма з позитивною тональністю знаходиться зверху, а з негативною – знизу. Бачимо, що попереднє сортування працює правильно і графік виглядає плавним і плавно зменшується до самого кінця.

Після створення точкової діаграми, точки ставляться за замовчуванням малими і майже не помітними див. Рис. 2.26.

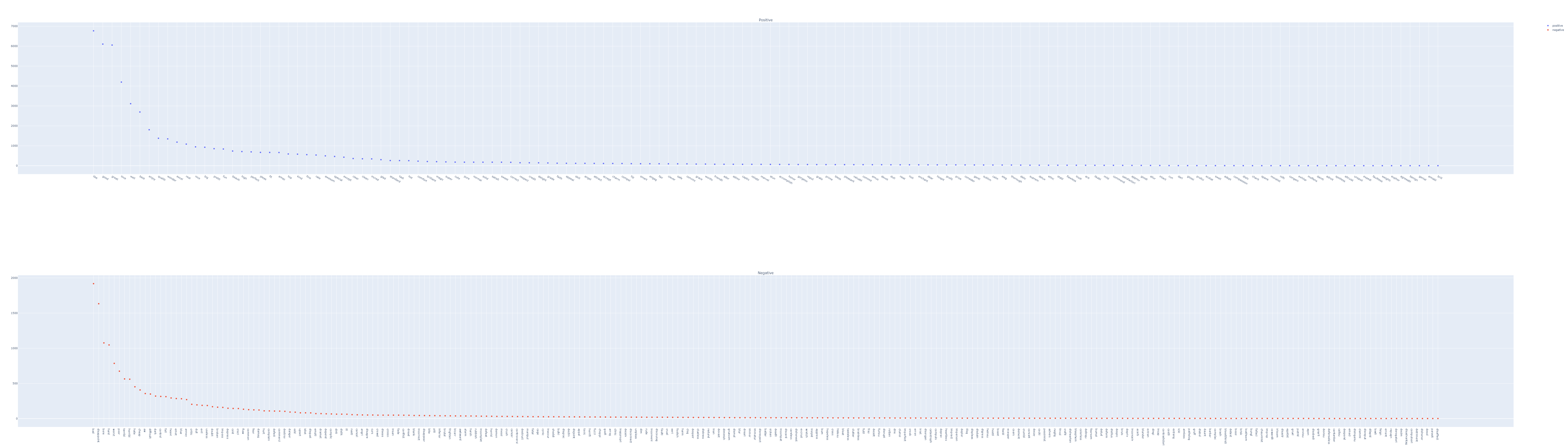


Рис. №2.26. Базовий вигляд діаграми.

Такий вигляд графіку дуже складний для аналізу і точки(маркери) майже неможливо помітити на великому віддалені, або коли потрібно зробити загальний знімок обох графіків.

Для того, щоб маркери були більш помітними на віддаленні, потрібно зробити їх більше і бажано ще зробити темне обведення для підвищення контрастності на світлому фоні. Такі зміни значно покращать загальний вигляд і буде значно легше зробити аналіз використовуючи такі графіки. Код див. Рис. №2.27.

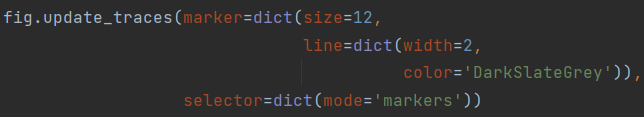


Рис. №2.27. Зміна маркерів для графіку.

Першим рядком ми звертаємося до самої конструкції маркеру, де змінюємо розмір. Наступним рядком додаємо обведення для маркеру вказуючи товщину, та на наступному рядку – колір. Порівняти якими маркери були і якими стали можна на Рис. №2.28.

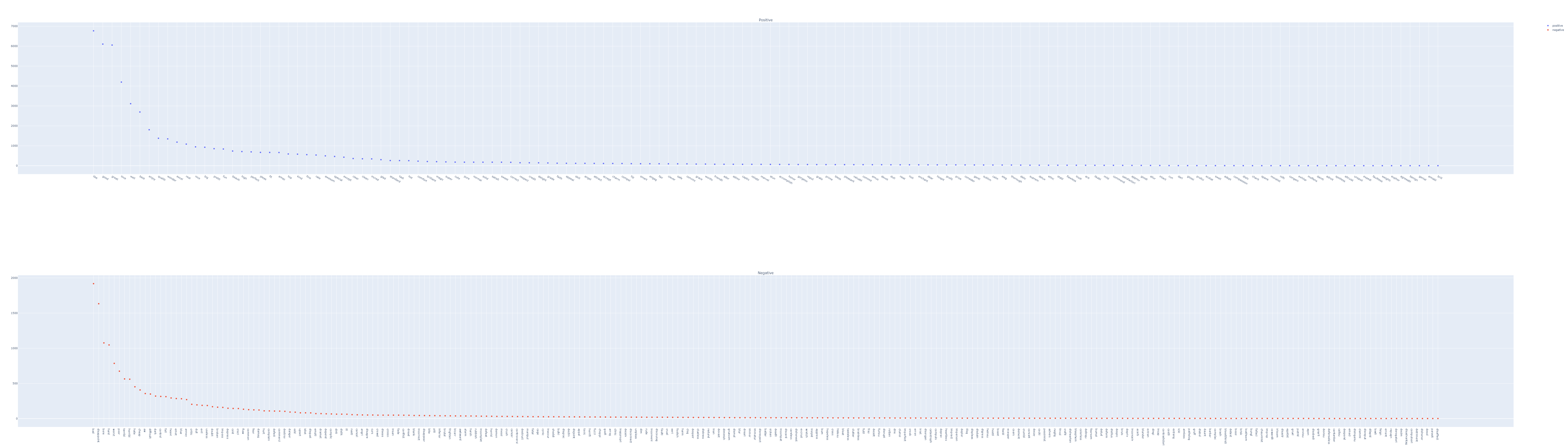
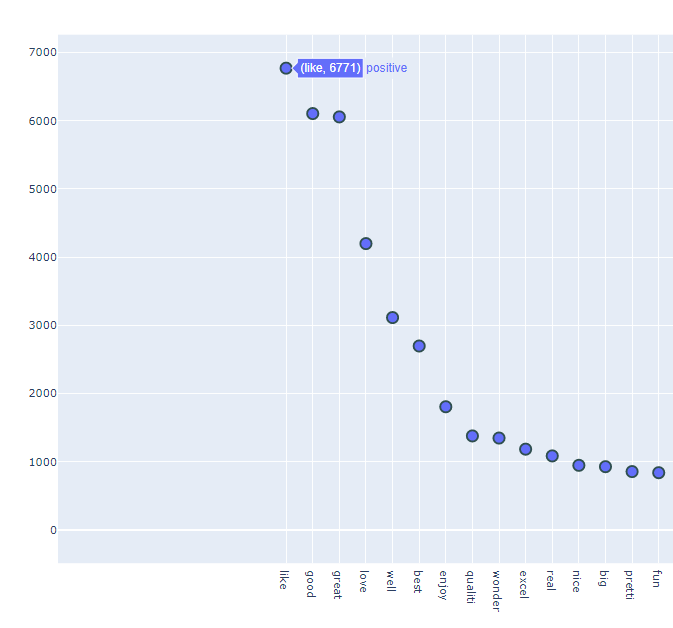
 

Рис. №.2.28. Порівняння маркерів.

Після зміни зовнішнього вигляду можна більш детально роздивитися отримані результати першої діаграми слів з позитивною тональністю див. Рис. №2.29.

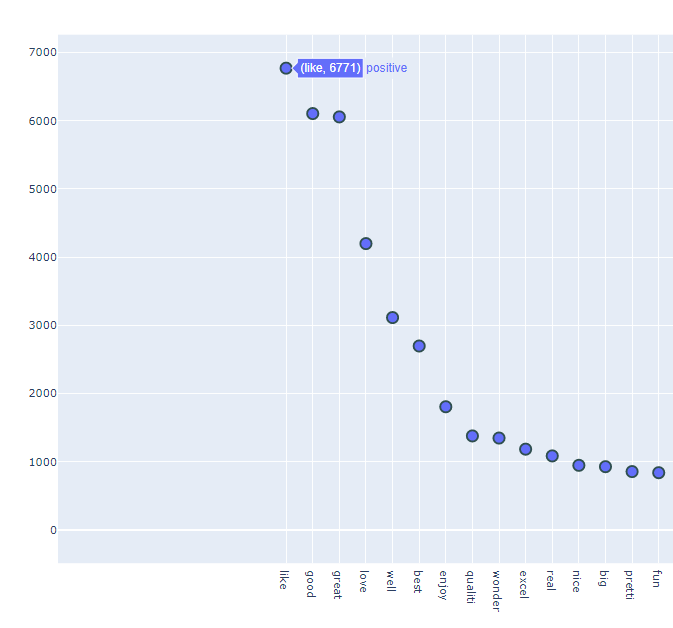
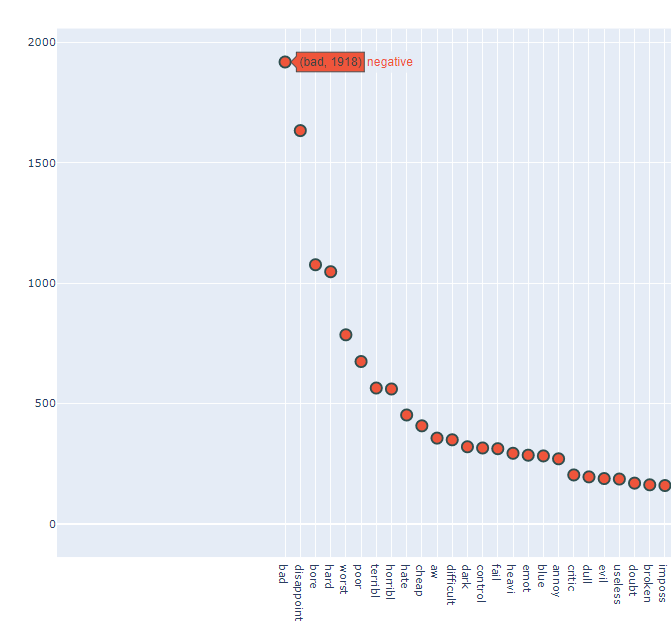


Рис. №2.29. Графік слів з позитивною тональністю.

На основі отриманих даних можна побачити які слова використовуються частіше за інші, а які навпаки. Можна зробити перший висновок, що найпопулярніші слова короткі, а тому їх можна швидко написати, а слово “like” дуже часто використовується як міра схвалення іншими людьми чого завгодно в соціальних мережах і саме тому власне вона і займає своє місце найбільш вживаного слова в даній діаграмі.

Тепер роздивимось діаграму складену на основі слів з негативною тональністю див. Табличка №2.30.



Табличка №2.30. Графік слів з негативною тональністю.

На основі вже цієї діаграми виводи дещо інші. Помітно, що слово “bad” займає перше місце за вживаністю серед усіх слів з негативною емоційною забарвленістю також коротке. І звісно ж воно добре підходить для того, щоб показати своє ставлення до певних явищ, товарів або процесів завдяки своїй універсальності як і слово “like”. Майже усі інші слова помітно програють і знаходяться на майже одному рівні з іншими словами з негативною тональністю.

Для роботи з графіком, Plotly надає такий набір інструментів для кожного графіку див. Рис. №2.31.



Рис. №2.31. Інструменти для роботи з графіками.

Згідно порядку розташування на рисунку зверху набір включає такі інструменти:

* Знімок екрану - Screenshot
* Збільшення окремої ділянки - Zoom
* Переміщення - Pan
* Виділення квадратом – Box select
* Виділення ласо – Lasso select
* Приближення – Zoom in
* Віддалення – Zoom out
* Автоматичне вирівнювання – Autoscale
* Відміна змін – Reset axes

Завдяки цим інструментам можна вивчати графік більш докладно, виділяти потрібні елементи і зберігати знімки для подальшого аналізу. Також завдяки цьому функціоналу дуже зручно виступати перед аудиторією і спрямовувати фокус глядачів на найбільш важливі моменти в графіку, щоб усі залучалися до роботи і допомагали аналізувати отримані дані.

На даному етапі аналізу слів можна зробити перший висновок, шо короткі, універсальні і зрозумілі майже кожному з дитинства слова набирають найбільше повторів в нашій базі коментарів. Найменша кількість повторів була у більш довгих слів, також ці слова призначені більше для опису певних явищ, товарів або процесів і не можуть бути універсальними через своє призначення, що значною мірою призводить до отриманого результату. На усю базу з 40000 коментарів подібні слова зустрічаються лише 1 один раз, але не брати їх до уваги під час аналізу буде недоречним, оскільки таких слів десятки і це може сильно вплинути на результат аналізу і призведе до хибної оцінки у найвідповідальніший момент.

Тепер треба порівняти кількість слів з якою тональністю більше для завершення сентимент-аналізу тексту. Для цього було написано наступний код див. рис. №2.32.

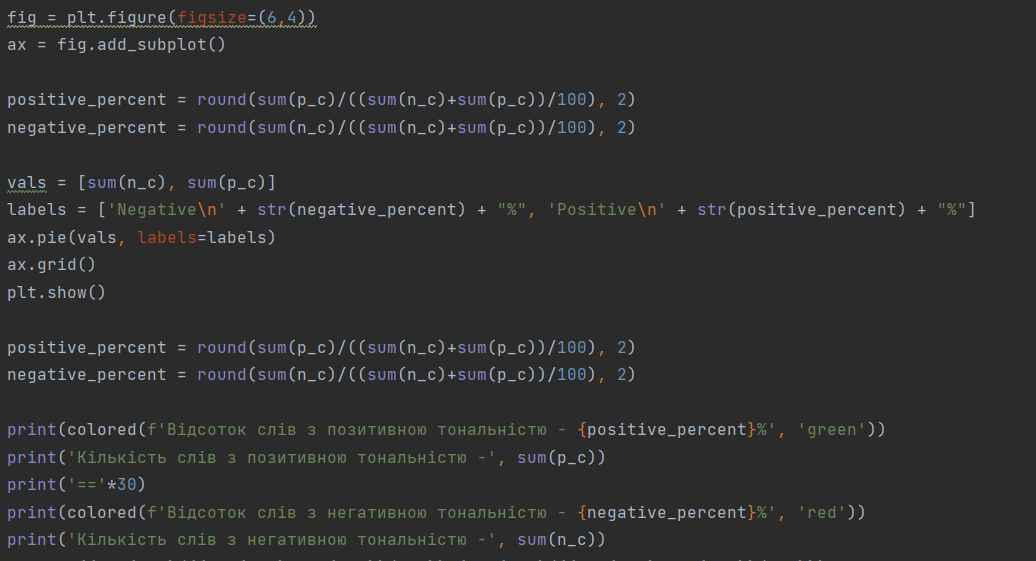
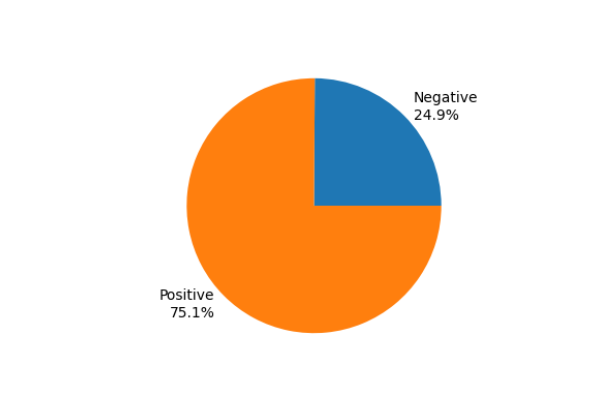


Рис. №2.32. Створення кругової діаграми.

Це буде круговий графік, який буде створено без використання plotly, саме для прикладу, що графіки можна будувати навіть не маючи браузеру на пристрої де виконується код і іноді це може бути корисним див. Рис.№2.33. Такий підхід також має свої недоліки, оскільки інструменти для роботи з діаграмою потрібно буде створювати власноруч і це займе більше часу і ресурсів у розробника. Але розглянути і цей варіант розробки не буде зайвим для більшого ознайомлення з варіантами реалізації. Особливо корисним такий варіант буде коли замовник потребує побудову діаграм без залучення інших програм.



Табличка №2.33 Кругова діаграма з відсотковим співвідношенням.

Неозброєним оком помітно, що слів з позитивною тональність значно більше ніж з негативною. На основі цих даних вже можна зробити повноцінний висновок, що велика кількість людей, шо мали спільну тему для обговорювання в більшості своїй зустріли її позитивно. Більше докладної інформації за отриманим графіком можна побачити у консолі програми див. Рис. №2.34.

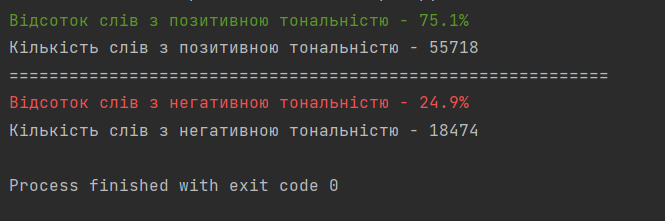


Рис. №2.34. Інформація для кругової діаграми.

Тут так само можна побачити процентне співвідношення, а також саму кількість слів з емоційним забарвленням.

Усі необхідні розрахунки і візуалізацію даних було виконано як і планувалося. На основі отриманих даних та графіків можна зробити аналіз для якоїсь компанії.

Можна уявити, що ці дані стосувалися молодої компанії яка запустила свій перший проект і вирішила перевірити які в неї відгуки щодо продукту. Вони бачать, що ¼ від усіх відгуків являється негативною оцінкою. Для молодої компанії це прийнятний показник, адже ще нема достатнього досвіду, щоб робити одразу потужний і відсутній на недоліки продукт. В такому разі їм слід дізнатися про свої слабкі сторони, зробити роботу над помилками і знову перевірити реакцію своїх клієнтів на наявність недоліків і робити так доки процент не буде задовольняти як компанію так і абсолютну більшість користувачів.

Можна роздивитися інший варіант коли мова йдеться про компанію з великим та іноді багаторічним досвідом роботи в своїй галузі і вже заробивши певний статус і прихильників серед інших компаній. Отримавши таку статистику, де ¼ клієнтів негативно зустріла продукт, буде означати, що все дуже погано з їхнім продуктом і потрібно діяти якомога швидше, щоб не втратити своє місце на ринку і прихильників які мають запас довіри до цієї компанії і можуть почекати поки рівень якості повернеться до нормальних значень.

Є ще один спосіб використання сентимент-аналізу, але вже не для свого продукту або послуги, а для перевірки конкурентів. Таким чином можна дізнатися, що зараз в тренді або тільки набуває популярності і швидко пристосуватися до змін і отримати хороший прибуток.

Також існує такий варіант, коли конкурент сильніший за вас різко почав втрачати позиції і це не тільки гарна можливість зайняти його місце, а й дізнатися причини його падіння в рейтингу, щоб навчитися на чужих помилках і ліквідувати їх у себе до істотного впливу цих помилок у власній компанії. Так можна переймати досвід нескінченну кількість разів і завжди залишатися на якісному рівні і завойовувати визнання на своєму ринку.

Вищеописані технології реалізовані в десятках доступних на ринку продуктів. В Інтернеті можна знайти велику кількість інструментів і платформ, які, як стверджують їхні автори, в змозі допомогти клієнтам відстежувати і оцінювати згадки компанії або брендів в соціальних медіа-каналах. Ці програми мають величезний розкид по функціональності та ціні. Найбільш прості надаються безкоштовно, більш просунуті доступні в складі комерційних продуктів або платних веб-сервісів.

Ми розглянули один з аспектів аналізу збору інформації на базі соціальних медіа. Інформація від клієнтів і потенційних і наявних споживачів продуктів компанії, інтегрована з іншими корпоративними системами, є основою для підтримки прийняття рішень в корпорації на різних рівнях управління та маркетингової стратегії компанії.

**Висновок**

Під час виконання цієї курсової роботи ми могли наглядно побачити, що збільшення росту технологій буде приводити до зростання кількості інформацій в інтернеті. Це означає, що необхідність в обробці неструктурованих даних з часом буде тільки збільшуватися.

Робота з даними була детально розглянута в курсовій на прикладі відгуків звичайних користувачів з Amazon на англійській мові у розмірі 40000 відгуків. Для роботи використовувався язик програмування Python, а для збору і візуалізації отриманих даних було використано регулярні вирази, nltk, matplotlib та plotly. Було виконано описання всіх процедур очистки і підготовки тексту до побудови графіку. Було побудовано два точкових графіка і кругова діаграма. Докладно розібрано код самої програми.

Даний спосіб сентимент-аналізу даних може використовуватися для досліджень відгуків з різноманітних джерел для виявлення думок авторів щодо конкретних об’єктів. Таким чином можна швидко і дешево проводити аналіз на тональність відгуків в соціальних медіа для покращення бізнесу або продукту.

**Джерела**

1. Сайт kaggle з відгуками користувачів Amazon для перевірки на сентимент-аналіз <https://www.kaggle.com/bittlingmayer/amazonreviews>
2. Документація про роботу з модулем Collections <https://pythonworld.ru/moduli/modul-collections.html>
3. Сайт для побудови графіків <https://dash.gallery/Portal/>
4. Документація для роботи з графіками <https://plotly.com/python-api-reference/generated/plotly.subplots.make_subplots.html>
5. Набір слів з негативною тональністю <https://www.englishdom.com/blog/vse-sinonimy-slova-bad-negativnye-prilagatelnye-v-anglijskom/#h2_0>
6. Набір слів з позитивною тональністю <https://www.englishdom.com/blog/vse-sinonimy-slova-good/>
7. Модуль nltk <https://www.nltk.org/>