**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО»**

**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Кафедра інформаційної безпеки

«На правах рукопису» «До захисту допущено»

УДК 004.056 В.о. завідувача кафедри

\_\_\_\_\_\_\_\_\_\_ М.В.Грайворонський

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2019 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

зі спеціальності: 125 Кібербезпека

на тему: Виявлення зловмисних повідомлень в соціальній мережі Twitter на основі URL-посилань

Виконала: студентка 2 курсу, групи ФБ-81мп

Дешуніна Дарія Сергіївна

(прізвище, ім’я, по батькові) (підпис)

Науковий керівник

к.ф.-м.н., доцент, Микола Владленович Грайворонський

(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО»**

**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (освітньо-професійна програма) – 125 Кібербезпека («Системи, технології та математичні методи кібербезпеки)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_\_\_\_\_\_ М.В.Грайворонський

(підпис)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2019 р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

Дешуніній Дарії Сергіївні

(прізвище, ім’я, по батькові)

1. Тема дисертації: Виявлення зловмисних повідомлень в соціальній мережі Twitter на основі URL-посилань

науковий керівник дисертації

к.ф.-м.н., доцент, Микола Владленович Грайворонський

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

затверджені наказом по університету від «\_\_\_» \_\_\_\_\_\_\_\_ 2019 р. № \_\_\_\_\_\_\_

2. Термін подання студентом дисертації 10.12.2019 р.

3. Об’єкт дослідження

4. Вихідні дані

5. Перелік завдань, які потрібно розробити

6. Орієнтовний перелік ілюстративного матеріалу

7. Орієнтовний перелік публікацій

8. Дата видачі завдання \_\_\_\_\_\_\_\_\_\_

Календарний план

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів виконання  магістерської дисертації | Термін виконання етапів магістерської дисертації | Примітка |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Студент \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (ініціали, прізвище)

Науковий керівник дисертації \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (ініціали, прізвище)

# РЕФЕРАТ

Робота обсягом 83 сторінки містить 12 ілюстрацій, 46 таблиць, 31 джерело за переліком посилань та 4 додатка.

Метою роботи є побудова і практична перевірка удосконаленої моделі виявлення зловмисних повідомлень в мережі Twitter для підвищення надійності детектування таких дій як розповсюдження спаму, фішингу та зловмисного ПЗ.

Об’єктом дослідження є виявлення зловмисних повідомлень в соціальних мережах.

Предметом дослідження є удосконалена системи виявлення зловмисних повідомлень в соціальній мережі Twitter використовуючи алгоритми машинного навчання.

Результатами роботи є аналіз сучасних підходів при вирішенні задач виявлення зловмисних повідомлень та посилань в соціальних мережах, побудований набір даних з використанням Twitter API та маркування за допомогою VirusTotal API, результатом чого було виявлено 3 класи зловмисних посилань, а саме спам, фішинг та розповсюдження зловмисного ПЗ. Тестування запропонованої модулі виявлення повідомлень на основі алгоритмів класифікації та порівняння результатів точності виявлення.

Методи дослідження: теоретичний аналіз поставленої задачі, аналіз набору даних, застосування алгоритмів класифікації та метрик оцінки якості роботи системи.

Ключові слова: Twitter, виявлення зловмисних повідомлень, URL- посилання, спам, фішинг, зловмисне ПЗ, машинне навчання, класифікація, аналіз даних, соціальній мережі, набір даних, VirusTotal API.

# ABSTRACT

The work volume 83 pages contains 12 illustrations, 46 tables, 31 source of literature and 4 appendices.

The purpose of the study is to build a software for detection of malicious messages on Twitter and detect such activities as spamming, phishing and malware.

The object of the research is detection of malicious messages on social networks.

The subject of the study is the possibility of implementing a system for detecting malicious messages on the social network Twitter using machine learning algorithms.

The results of the work consist from an analysis of existing approaches to solving problems of detection malicious messages and links on social networks, created dataset using the Twitter API and labelling using the VirusTotal API, which resulted in the detection of 3 classes of malicious links such as spam, phishing and malware. Testing the proposed message detection model based on classification algorithms and comparison of detection accuracy results.

Research methods: theoretical analysis of the task, analysis of the data set, application of classification algorithms and metrics for assessing the quality of the system.

Keywords: Twitter, detection of malicious messages, URL links, spam, phishing, malware, machine learning, classification, data analysis, social networks, dataset, VirusTotal API.

ЗМІСТ

[Перелік умовних позначень, символів, одиниць, скорочень і термінів 8](#_Toc27352240)

[Вступ 9](#_Toc27352241)

[1 Огляд сучасних рішень 11](#_Toc27352242)

[1.1 Виявлення зловмисних URL посилань на основі поведінкового аналізу 12](#_Toc27352243)

[1.2 Використання методів машинного навчання для виявлення зловмисних URL посилань 14](#_Toc27352244)

[1.3 Виявлення зловмисних повідомлень в соціальних мережах використовуючи аналіз тексту 17](#_Toc27352245)

[1.4 Аналіз зловмисної інформації в соціальних мережах 18](#_Toc27352246)

[1.4.1 Аналіз спаму на основі Blacklisted URL-посилань 18](#_Toc27352247)

[1.4.2 Аналіз підозрілих облікових записів в соціальній мережі Twitter 19](#_Toc27352248)

[1.4.3 Виявлення джерел розповсюдження спаму в соціальних мережах 19](#_Toc27352249)

[Висновки до розділу 1 20](#_Toc27352250)

[2 Обґрунтування запропонованого рішення при виявлення зловмисних повідомлень 22](#_Toc27352251)

[2.1 Twitter API 22](#_Toc27352252)

[2.2 VirusTotal API 26](#_Toc27352253)

[2.4 Методи машинного навчання 29](#_Toc27352254)

[2.5 Метрики оцінки якості роботи алгоритмів 35](#_Toc27352255)

[Висновки до розділу 2 40](#_Toc27352256)

[3 Реалізація запропонованого рішення та аналіз результатів 41](#_Toc27352257)

[3.1 Процес отримання набору даних 41](#_Toc27352258)

[3.2 Маркування даних з використанням VirusTotal 43](#_Toc27352259)

[3.3 Виділення важливих ознак даних 44](#_Toc27352260)

[3.4 Реалізація запропонованої моделі виявлення зловмисних повідомлень із застосуванням алгоритмів машинного навчання 46](#_Toc27352261)

[3.5 Аналіз результатів роботи системи 48](#_Toc27352262)

[3.5.1 Оцінка алгоритмів за метриками Precision, Recall та F1-score 48](#_Toc27352263)

[3.5.2 Визначення найбільш важливих характеристик зловмисних повідомлень 53](#_Toc27352264)

[Висновки до розділу 3 55](#_Toc27352265)

[4 Розробка стартап проекту 57](#_Toc27352266)

[4.1 Опис ідеї проекту 57](#_Toc27352267)

[4.2 Технологічний аудит ідеї проекту 59](#_Toc27352268)

[4.3 Аналіз ринкових можливостей запуску стартап-проекту 60](#_Toc27352269)

[4.4 Розроблення ринкової стратегії проекту 67](#_Toc27352270)

[4.5 Розроблення маркетингової програми стратап-проекту 69](#_Toc27352271)

[Висновки до розділу 4 72](#_Toc27352272)

[Висновки 73](#_Toc27352273)

[Перелік джерел посилання 74](#_Toc27352274)

[ДОДАТКИ 77](#_Toc27352275)

[ДОДАТОК А 78](#_Toc27352276)

[ДОДАТОК Б 80](#_Toc27352277)

[ДОДАТОК С 81](#_Toc27352278)

[ДОДАТОК Д 83](#_Toc27352279)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

**URL** (Uniform Resource Locator) – єдиний вказівник на ресурс

**ПЗ** – програмне забезпечення

**API** (Application Programming Interface) – прикладний програмний інтерфейс

**TP** (true positive) – істині значення позитивного класу

**TN** (true negative) – істині значення негативного класу

**FP** (false positive) – помилки першого роду

**FN** (false negative) – помилки другого роду

**KNN** (K-nearest Neighbors) – алгоритм класифікації k-найближчих сусідів

**GNB** (Gaussian Naïve Bayes) – алгоритм класифікації «Гауссів наївний класифікатор»

**RFC** (Random Forest Classifier) – алгоритм класифікації «Випадковий ліс»

**SVM** (Support Vector Machine) – алгоритм класифікації метод опорних векторів

# ВСТУП

У зв’язку з тим, що соціальні мережі на сьогодні є одним з основних засобів спілкування, а також поширення новин, зловмисники все частіше використовують їх для розповсюдження спаму, фішингу, зловмисного ПЗ та ін.

Соціальна мережа Twitter є і залишається одним з найчастіше використовуваних мереж на рівні з Facebook та Instagram. Таким чином, в мережі Twitter все частіше зустрічаються спам та фішинг, які з часом стає все складніше виявляти. Складність процесу виявлення зловмисних повідомлень такого виду полягає в тому, що швидкість розповсюдження URL- посилань за рахунок повідомлень Twitter є дуже високою на відміну від електронної пошти. Також зловмисники застосовують різні підходи для приховання їхніх дій, наприклад, за рахунок обфускації URL-посилань, тобто скорочення їх довжини та приховування кінцевого призначення посилання.

Таким чином в даній роботі досліджуються можливості виявлення зловмисних повідомлень з використанням методів машинного навчання.

**Актуальність роботи** зумовлена тим, що автоматизація процесу виявлення зловмисних повідомлень в соціальних мережах залишається невирішеним питанням та має місце для досліджень та покращень.

**Метою роботи** є побудова і практична перевірка удосконаленої моделі виявлення зловмисних повідомлень в мережі Twitter для підвищення надійності детектування таких дій як розповсюдження спаму, фішингу та зловмисного ПЗ.

**Завданням дослідження** є:

* аналіз існуючих рішень при вирішенні подібних задач;
* побудова удосконаленої моделі виявлення зловмисних повідомлень в мережі Twitter;
* програмна реалізація запропонованого рішення;
* аналіз отриманих результатів із використанням метрик машинного навчання.

**Об’єктом дослідження** є виявлення зловмисних повідомлень в соціальних мережах.

**Предметом дослідження** є удосконалена системи виявлення зловмисних повідомлень в соціальній мережі Twitter використовуючи алгоритми машинного навчання.

**Методами дослідження** є теоретичний аналіз поставленої задачі, аналіз набору даних, застосування алгоритмів класифікації та метрик оцінки якості роботи системи.

**Наукова новизна** полягає у власному методі побудови набору даних та їх маркування з використанням VirusTotal API; реалізована система виявлення зловмисних повідомлень в мережі Twitter з використанням класифікації, що виявляє не лише фішинг або спам, а й інші види зловмисних посилань.

**Практичним значенням** одержаних результатів є, за рахунок порівняльного аналізу, визначено найбільш ефективний алгоритм класифікації для даної задачі.

# ОГЛЯД СУЧАСНИХ РІШЕНЬ

На сьогоднішній день соціальні мережі є одним з основних засобів спілкування і найбільш відвідуваним ресурсом порівняно з іншими web-застосунками. Таким чином, соціальні мережі стають чудовою платформою для розповсюдження спаму, небезпечних програмних засобів, фішингу та ін.

Особливу увагу необхідно звернути на те, що небезпечним є те, які дії виконує користувач. Зазвичай така поведінка як, ввід інформації в поля web-сторінок або перехід на інші URL-адреси, призводять до викрадення інформації або завантаження зловмисного ПЗ на машину користувача. Але для того щоб користувач виконав необхідні дії для зловмисника необхідним є застосування психологічної маніпуляції, що примусить користувача перейти на певну сторінку. Такі дії називаються соціальною інженерією. Зловмисники використовують різноманіття методів та технологій для нанесення матеріальних та моральних збитків організаціям або фізичним особам. До соціальної інженерії відносять такі основні вектори атак, як вішинг (з англ. vishing), фішинг (з англ. phishing), смішинг (з англ. smishing), impersonation.

В даному розділі розглядаються роботи, які присвячені вирішенню проблеми виявлення зловмисних повідомлень в соціальних мережах. При аналізі обраних робіт було вирішено розділити їх за такими трьома категоріями: системи які працюють на основі поведінкового аналізу, моделі з використанням машинного навчання та системи на основі аналізу тексту.

Отже, в наступних підрозділах буде розглянуто сучасні дослідження та підходи які використовувались для виявлення зловмисних URL-посилань, спаму, фішингу та ін., що розповсюджуються в соціальних мережах.

## Виявлення зловмисних URL-посилань на основі поведінкового аналізу

Деякі роботи базуються на поведінковому аналізі, спираючись на той факт, що поведінкові характеристики складніше імітувати чи підроблювати, ніж традиційні.

Робота [1] зосереджена на дослідженні URL- адрес та виявленні спаму в соціальних мережах використовуючи поведінковий аналіз, на основі таких характеристик: публікування інформації (з англ. posting behavior) та перехід за посиланням (з англ. clicking behavior). Таким чином, дослідники виявили 15 пов’язаних функцій, таких як кількість постів (з англ. posting count), частота постів (з англ. posting intensity), динаміка кліків (з англ. click dynamics), загальна кількість кліків (з англ. total number of clicks) та інші. Після цього було застосовано алгоритм класифікації Random Forest на виявлених показниках та досягли точність 86%.

В [2] було запропоновано підхід для виявлення URL спаму, який має назву BEAN. Після аналізу повідомлень спамерів, поведінку надсилання повідомлень та їх характеристики в Twitter, вони виявили шість поведінкових характеристик. В результаті на зібраних даних та характеристиках застосували модель Ланцюги Маркова (Markov Chain Model) для виявлення спаму. Таким чином, даний підхід продемонстрував можливість виявляти спам який неможливо виявити за допомогою таких звичайних алгоритмів як Support Vector Machine або TrustRank.

Наступна робота, яка була розглянута, також присвячена виявленню спаму із застосуванням машинного навчання але в блогах, а саме Yahoo! Hong Kong blog [3]. В даній роботі було проведено широке дослідження поведінки користувачів і виявлено такі основні поведінкові патерни: медіана інтервалу постів, тривалість активності, ранкові пости, медіана довжини URL адрес, властивості хостів та інше. Після виявлення основних характеристик було застосовано алгоритми класифікації Naive Bayes, Logistic Regression, та DecisionTree. Результати показали, що обрані поведінкові характеристики є релевантними і на їх основі ці алгоритми досягли дуже високі значення true positive і низькі значення false positive. Після чого вони побудували онлайн детектор спаму який базується на класифікаторі Decission Tree, blacklists та whitelists. В результаті отримані результати досягали від 91% до 95% значення true positive та false positive від 15% до 13%.

Деякі користувачі під час перебування у соціальних мережах паідпадали під фішинг атаки або спам. Проте також є користувачі які ніколи не були жертвами таких атак. Під впливом такого феномену було зроблено дослідження [4], яке вивчає поведінки тих чи інших користувачів і причини чому одні є в зоні ризику, а інші ні. Для початку було виявлено такі 5 характеристик за кожним кліком за посиланням, для виявлення контекстної інформації: Domain Name, Bag-of-Words, generic Top-Level Domains (gTLD), IP Address, Port Number. Після чого використовується алгоритм Conditional Random Field (CRF), який є типом дискримінаційної моделі ймовірності, під час чого вивчається поведінка користувача в браузері для передбачення категорії доступу користувача який він здійснить наступний раз. В результаті алгоритм показав високі показники виявлення фішингу які змінюються від 91% до 97%.

Таблиця 1.1 – Загальний опис робіт на основі поведінкового аналізу

|  |  |  |  |
| --- | --- | --- | --- |
| *Назва роботи* | *Ознаки* | *Алгоритми* | *Платформа* |
| Detecting Spam URLs in Social Media via Behavioral Analysis | 15 URL-based характеристик публікування інформації (posting behavior) та перехід за посиланням (clicking behavior) | Random Forest | Twitter |
| BEAN: A BEhavior ANalysis Approach of URL Spam Filtering in Twitter | 6 URL поведінкових параметрів | a Markov Chain Model | Twitter |
| Online Spam Detection in Blogs: A Behavior-based Approach | медіана інтервалу постів, тривалість активності, ранкові пости, медіана довжини URL адрес, властивості хостів та інше | Naive Bayes, Logistic Regression, DecisionTree | Yahoo! Hong Kong blog |
| Users’ Behavioral Prediction for Phishing Detection | 5 характеристик на основі кліків на URL посилання | Conditional Random Field | на основі данних лабораторії Trend Micro |

## Використання методів машинного навчання для виявлення зловмисних URL посилань

Більшість робіт використовують в своїх дослідженнях алгоритми машинного навчання з учителем, так як вони є більш точними і, при правильному виборі параметрів характеризації даних, є дуже ефективними. Таким чином, правильний підбір описових характеристик є на стільки ж важливо як і вибір алгоритму.

В першій роботі яку ми розглянемо [5], запропоновано підхід виявлення спамерів в Twitter на основі алгоритму Баєса. За допомогою соціальної графової моделі було виявлено взаємозв’язки «followers» та «friends», також було застосовано content-based та graph-based характеристики. Таким чином, в зібраних даних серед 25 тисяч користувачів було виявлено 3% спамерів, після порівнянь результатів алгоритму та реальних даних, точність становила 81%.

Цікавим дослідженням є робота із використанням honeypots [6]. В даній роботі основними компонентами було: 1) Встановлення соціальних honeypots для збирання спам профілей; 2) Статичний аналіз цих профілей для виявлення характеристик спамерів для подальшої побудови класифікатора. Встановленні honeypots зібрали 1570 спамерів з соціальної мережі MySpace та 500 спамерів з Twitter. В ході аналізу виявилось, що Honeypots данні корелюють з характеристиками профілів (content, friend information, posting patterns, number of friends, text on the profile). Таким чином було виявлено 4 групи характеристик: демографічні дані користувачів (вік, стать, локації, та ін.), контент наданий користувачем (текстове поле «About me», блог пости, коментарі, твіти та ін.), активність користувача (частота твітів та постів) та підключення користувачів (кількість друзів, фоловерів). Оцінка виявлення спаму виконувалась за допомогою 10 класифікаторів з інструменту WEKA. В результаті точність для MySpace перевищувала 98% і коливалася від 82% до 88% для даних Twitter. Таким чином, результати доводять припущення, що вбудовані honeypots можуть привабити увагу спамерів чия поведінка сильно відрізняється від звичайних користувачів.

Проте, характеристики на основі акаунту користувача мають два основних недоліки при виявленні спаму в соціальних мережах: 1) параметри можуть бути легко підробленні (fabricated); 2) існує затримка між розповсюдженням спаму та його виявленням, так як для виявлення спаму необхідні їх описові параметри, які можливо зібрати тоді коли велика кількість спаму уже була здійсненна. Для вирішення цих недоліків, було запропоновано фреймворк який базується на соціальних зв’язках [7], які можуть уникнути ухилення від функцій спамерів і виявити спам одночасно, коли одержувач отримує повідомлення. У цій роботі запропоновано дві нові риси співвідношення: «відстань» (з англ. distance) і «зв'язок» (з англ. connectivity). Під значенням distance розумілась довжина найкоротшого шляху, коли connectivity було обчислення за допомогою алгоритмів мінімального розрізу та випадкового блукання. З оцінки розподілу спамерів і неспамерів, вони виявили, що більшість спаму надходило від користувачів на відстані більше трьох. Крім того, зв'язок між спамерами та не спамерами відрізнявся від зв'язку між спамерами. Розглядаючи вимірювання цих двох характеристик співвідношень, спрямований підграф G, який був частиною всього соціального графа G, генерувався за чотирьох умов. Щоб оцінити цей метод, вони сканували набір даних, що містить 148 371 профіль, 267,551 твітів. Таким чином, вони отримали 308 спамерів і 10 000 спам-повідомлень. П'ять алгоритмів класифікації, вбудовані у WEKA, були перевірені на 10-кратну перехресну перевірку (10-fold cross validation). True Positive (TP) показник найкращого класифікатора, Bagging, коливався від 93,3% до 95,1% У попередніх роботах автори також імпортували 11 функцій, заснованих на користувачах, і виявили, що коефіцієнт True Positive був підвищений до 99,7%, а показник False Positive був знижений до 0,5% з приблизно 5%. Обмеження цієї роботи включають: 1) повідомлення від нового користувача буде класифікуватися як спам; 2) спам-повідомлення від компрометованих користувачів може бути позначено як доброякісне повідомлення.

Таблиця 1.2 – Загальний опис робіт на основі методів машинного навчання

|  |  |  |  |
| --- | --- | --- | --- |
| *Назва роботи* | *Ознаки* | *Алгоритми* | *Платформа* |
| Don't follow me: Spam detection in Twitter | content-based та graph-based характеристики | Алгоритм Баєса | Twitter |
| Don't follow me: Spam detection in Twitter | 4 групи характеристик: демографічні дані користувачів; контент наданий користувачем ;активність користувача; підключення користувачів | 10 алгоритмів класифікації, що вбудовані в WEKA | MySpace та Twitter |
| Spam filtering in twitter using sender-receiver relationship | 2 групи параметрів: відстань (distance) і зв'язок (connectivity) | п'ять алгоритмів класифікації, вбудовані у Weka, були перевірені на 10-кратну перехресну перевірку (10-fold cross validation) | Twitter |

## 1.3 Виявлення зловмисних повідомлень в соціальних мережах використовуючи аналіз тексту

Текст також може мати ознаки за якими класифікатори можуть виявляти аномалії. Таким чином, деякі дослідники застосовують аналіз текстової інформації для виявлення спаму.

В роботі [8] було запропоновано систему яка виявляє спам повідомлення на основі аналізу мови яка використовується в твітах. Основною метою роботи було освітити виконати наступні завдання: виявлення спам-твітів окремо і без попередньої інформації користувача; і застосування статистичного аналізу мови для виявлення спаму в тенденційних темах. Зокрема, мовними моделями, які було застосовано, є (1) мовна модель твітів, пов'язана з трендовою темою, (2) мовна модель твітів, і (3) мовна модель посилання на сторінку, яка присутня в твітах. Мовна модель є статистичною моделлю для аналізу тексту, яка ґрунтується на розподілі ймовірностей над фрагментами тексту, що вказує на ймовірність спостереження цих фрагментів в мові. Дані моделі використовувались для виявлення характеристик та відмінностей в тексті між спамом та не спамом. Загальна кількість зібраних даних: 34 тисячі трендових тем та 20 мільйонів твітів. Наступним кроком було застосування алгоритмів класифікації з використанням WEKA. Алгоритми які було обрано: Decision Tree (C4.5), Naive Bayes, Logistic Regression, Support Vector Machines (SVMs) Decorate, Random Forest. Підсумовуючи, система виявлення спаму здатна виявити 94,5% спам-твітів і отримати помилкову позитивну (FPR) 5,4%.

В наступній роботі [9] досліджували трендові теми з 6.5 мільйонів твітів та виявили, що число трендових тем має найбільшу розбіжність між спамерами та законними користувачами. Таким чином вони побудували модель класифікації на основі алгоритму Decission Tree, яка виявила правильно 73.5% спамерів з 0.25% FPR.

## 1.4 Аналіз зловмисної інформації в соціальних мережах

Перед тим як будувати модель виявлення зловмисних повідомлень необхідним є аналіз та розуміння природи таких повідомлень та акаунтів з яких вони розповсюджуються. Таким чином, далі ми розглянемо деякі роботи які проаналізувавши зловмисні повідомлення роблять висновки, щодо поведінки спамерів та особливості їх повідомлень.

Найбільш розповсюдженим типом зловмисної інформації в соціальних мережах є спам. Таким чином більшість робіт зосереджені на виявленні спаму та аналізу таких повідомлень та акаунтів з яких розповсюджується спам.

## 1.4.1 Аналіз спаму на основі Blacklisted URL-посилань

В роботі [10] було проаналізовано 26 мільйонів URL-адрес із 200 мільйонів публічних твітів, які були зібрані в межах одного місяця. Після маркування за допомогою чотирьох «чорних» списків (blacklists), 3 мільйони твітів було визначено як спам з яких 5% було віднесено до зловмисного програмного забезпечення, і решта 95% спрямовували жертви на афери. Однак 26% URL-посилань визначили як спам після того, як виконали перевірку набору даних вручну, що вказує на те, що ефективність чорного списку була не достатньо високою. Також було виявлено, що URL-адреси потрапляють у чорний список після їх існування від 4 до 20 днів. Тим не менше, 90% жертв відвідало зловмисні URL-посилання протягом перших двох днів їх розміщення. Отже, затримка в виявленні спаму за допомогою чорних списків занадто довга, щоб запобігти користувачам відвідування посилань. Крім цього, дослідники також вважали, що 16% облікових записів спаму це шахрайські акаунти (які створюються явно для розповсюдження спаму), і 84% є скомпрометованими акаунтами.

## 1.4.2 Аналіз підозрілих облікових записів в соціальній мережі Twitter

Так як результати попередньої роботи [10] показали, що чорні списки URL-посилань пропускають великі відсотки спаму (8% виявлено чорними списками, коли 26% спаму було знайдено за допомогою ручного аналізу), в роботі [11] пропонують алгоритм власного маркування спаму в Twitter. Також цікавим є те, що 77% спам акаунтів призупиняють свою активність протягом одного дня після їх перших повідомлень, а 92% спам-акаунтів тривають лише протягом 3 днів. Таким чином, під таким тиском, 89% облікових записів спамерів рідко встановлюють соціальні зв’язки з користувачам (їхні дії вимагають не більше 10 фоловерів). Також 17% спам акаунтів використовують в своїх повідомленнях популярні теми на даний момент для зацікавлення жертв, коли 52% спамерів згадують небажані теми в своїх повідомленнях. Дослідники вивчали спам групи та їх активність і виявили, що більшість з них направляють жертви на різні авторитетні інтернет-магазини, наприклад, таких компаній як Amazon.

## 1.4.3 Виявлення джерел розповсюдження спаму в соціальних мережах

На відміну від робіт [10] та [11], в роботі [12] дослідники акцентують увагу на розповсюдженні спаму в мережі Facebook. В процесі збору даних було отримано 187 мільйонів повідомлень (постів на стіні) від 3.5 мільйони користувачів.

По-перше, метою даної роботи було виявлення користувачів які розповсюджують спам в мережі. Кожен пост було змодельоване як пара: опис повідомлення та URL-посилання. Два повідомлення вважались подібними якщо вони: 1) мають подібний опис; 2) мають однакову URL-адресу. Таким чином подібні повідомлення формують кластери. Кластери які помічено як шкідливі повідомлення мають такі властивості: розподілена властивість (кількість користувачів та кількість постів у даному кластері) і властивість «busty» (абсолютний інтервал часу між послідовними постами). Тобто, якщо кілька облікових записів надсилають подібні повідомлення або багато повідомлень надсилається за короткий проміжок часу, такі кластери будуть ідентифікуватись як спам-кампанії. По-друге, завданням цієї роботи також було характеризувати спам-кампанії та виявити їх ознаки.

Таким чином, було виявлено три основні спам-кампанії: 1) спамери, які обіцяють безкоштовні подарунки; 2) спамери викликають інтерес за рахунок повідомлень, що хтось цікавиться ними; 3) спамери описують різні продукти. Вони також виявили, що 70,3% спам повідомлень спрямовують жертв на фішинг-сайти, а 35,1% спам повідомлень призводять до завантаження зловмисного програмного забезпечення. Також цікавим є те, що 97% акаунтів, які поширюють спам, є скомпрометованими, а не створюють нові унікальні облікові записи для спамерів.

## Висновки до розділу 1

В даному розділі розглянуто роботи та підходи які використовувались для вирішення подібних задач. Ці рішення ми розділили на три групи за методом аналізу та виявлення зловмисних посилань. Таким чином маємо системи які працюють на основі поведінкового аналізу, моделі з використанням машинного навчання та системи на основі аналізу тексту.

Також ми розглянули роботи, які зосереджено на аналізі зловмисних повідомлень та акаунтів користувачів, тобто виявленні ознак, які відрізняють звичайних користувачів від підозрілих.

# 2 ОБҐРУНТУВАННЯ ЗАПРОПОНОВАНОГО РІШЕННЯ ПРИ ВИЯВЛЕННЯ ЗЛОВМИСНИХ ПОВІДОМЛЕНЬ

Після аналізу існуючих робіт та рішень в даній галузі ми обрали власну стратегію отримання набору даних та побудови моделі. На етапі побудови набору даних ми використовуємо Twitter API для збору повідомлень які містять URL-посилання, для маркування даних використовується VirusTotal API, який сканує URL-посилання і на основі результатів повідомлення відноситься до визначеного класу. Наступним етапом є виділення ознак за якими побудована модель буде класифікувати дані. Цей етап є дуже важливим, так як від правильного підбору ознак, при навчанні, алгоритм виявить взаємозв’язки і чим більше вони будуть виражені тим більш точні результати класифікації отримаємо.

Усі програмні реалізацію було написано мовою програмування Python та проаналізовано отримані результати за допомогою метрик машинного навчання.

В наступних підрозділах ми розглянемо більш детально теоретичну складову кожного етапу даного дослідження.

## 2.1 Twitter API

Основною відмінною рисою мережі Twitter від інших соціальної мереж є те, що довжина повідомлення обмежується 280 символами, таким чином Twitter є платформою для мікроблогінгу, де повідомлення обновлюються дуже швидко і через декілька годин можуть бути неактуальними. Тобто «твіт» має набагато коротший термін служби, ніж оновлення у Facebook. Згідно з деякими цікавими відомостями компанії Sysomos, приблизно 92% взаємодії з «твітом» відбувається протягом першої години після того, як опубліковано повідомлення. З іншого боку, оновлення Facebook набагато довше присутнє у часових межах людей. Він часто може розтягнутися на кілька днів, поки вся взаємодія з оновленням не вичерпається. Таким чином Twitter стає чудовим джерелом для відслідковування новин та подій, що відбуваються у світі в даний момент часу.

Розробники Twitter також пропонують своїм користувачам більше можливостей за допомогою Twitter API. Twitter надає доступ до деяких служб із підтримкою API-інтерфейсу, що дає можливість програмістам розробляти програмне забезпечення та тісно взаємодіяти з Twitter, наприклад, додаток який автоматично відповідає на відгуки користувачів про компанію Twitter. [14] API-платформа Twitter надає широкий доступ до відкритих даних, якими користувачі готові поділитись з усім світом. Проте за допомогою API-інтерфейсів користувачі можуть керувати своєю закритою інформацією (наприклад, особистими повідомленнями) і надавати цю інформацію розробникам, яким вони дали відповідний дозвіл.

Для можливості користування даним сервісом необхідно мати акаунт в Twitter та подати заявку на власний додаток на сайті для розробників. Під час подання заявки Twitter збирає детальну інформацію про мету даного проекту. Через декілька днів після обробки даної інформації на пошту приходить повідомлення про успішну реєстрацію або відмову в використанні даного сервісу. Основою для взаємодію розробника та Twitter API є ключі автентифікації які надаються після успішної реєстрації.

Для взаємодії з API було вирішено використовувати мову програмування Python та вбудовану бібліотеку Tweepy. Клас API надає доступ до всіх методів API RESTful. Сервіс працює як класичний API додаток за принципом запит-відповідь, таким чином, кожен метод приймає параметри та повертає екземпляр класу Tweepy, що буде містити дані, повернені з Twitter, які ми зможемо використовувати всередині додатку.

Twitter API потребує автентифікації користувача. Необхідним є використання процесу OAuth для автентифікації, що вимагає два ключі автентифікації та два токени – це: Consumer key, Consumer secret key, Access token та Access token secret. Таким чином код для отримання доступу до взаємодії з Twitter буде мати наступний вигляд (Рисунок 2.1):

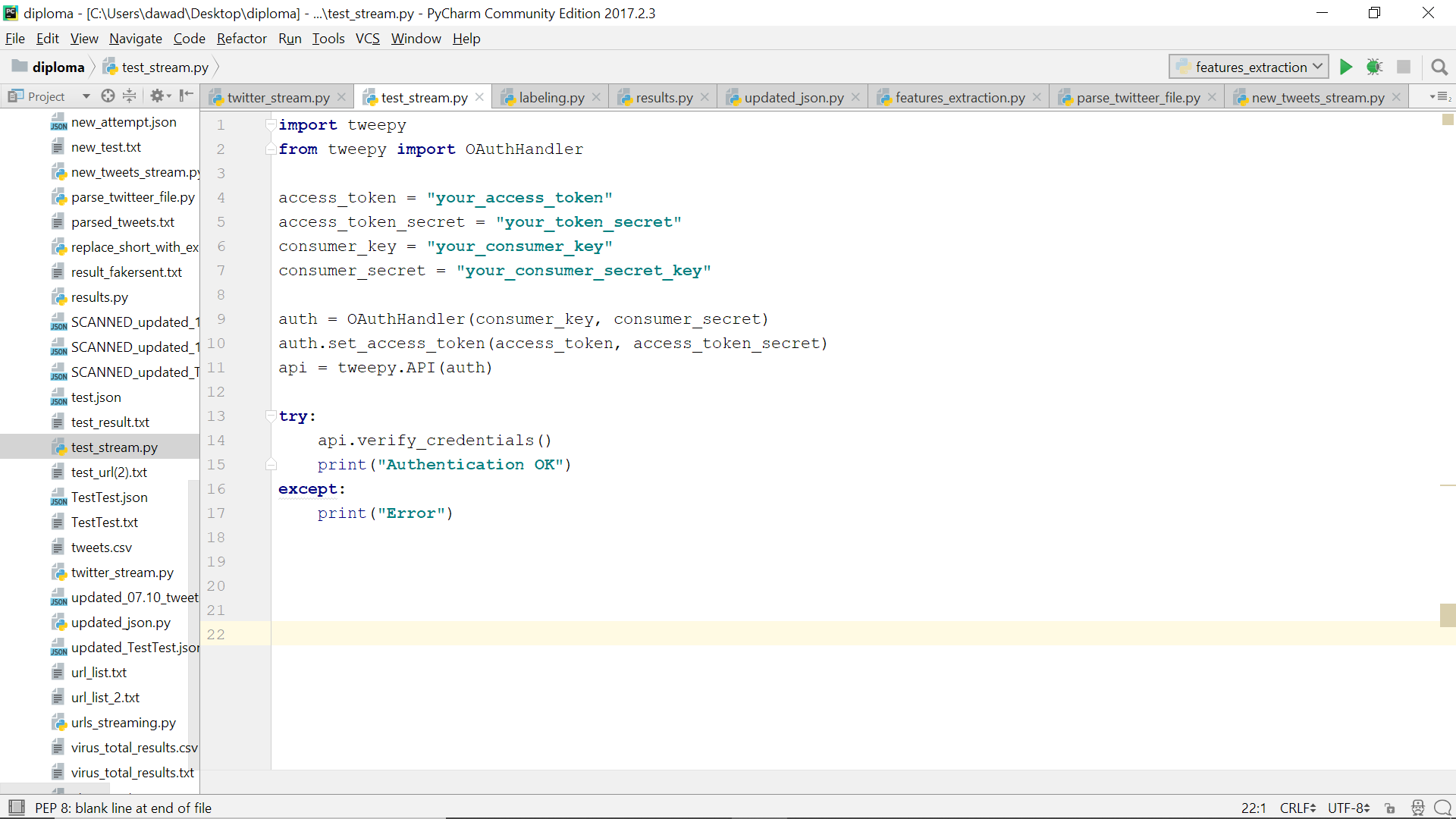


Рисунок 2.1 – Приклад програмного коду для доступу до Twitter API

Twitter API надає наступні можливості за рахунок RESTful методів [15]:

* GET- метод отримання даних
* POST – метод надсилання запиту
* PUT – метод оновлення даних
* DELETE – метод видалення даних

Кожний твіт складається з 28 характеристик (Рисунок 2.2) з яких «user», «extended\_tweet» та «entities» є складними об’єктами які надають: додаткову інформацію про користувача який надіслав даний твіт, розширену інформацію про сам твіт (наприклад, повний текст твіта) та додаткові елементи такі як URL-посилання, хештеги тощо.

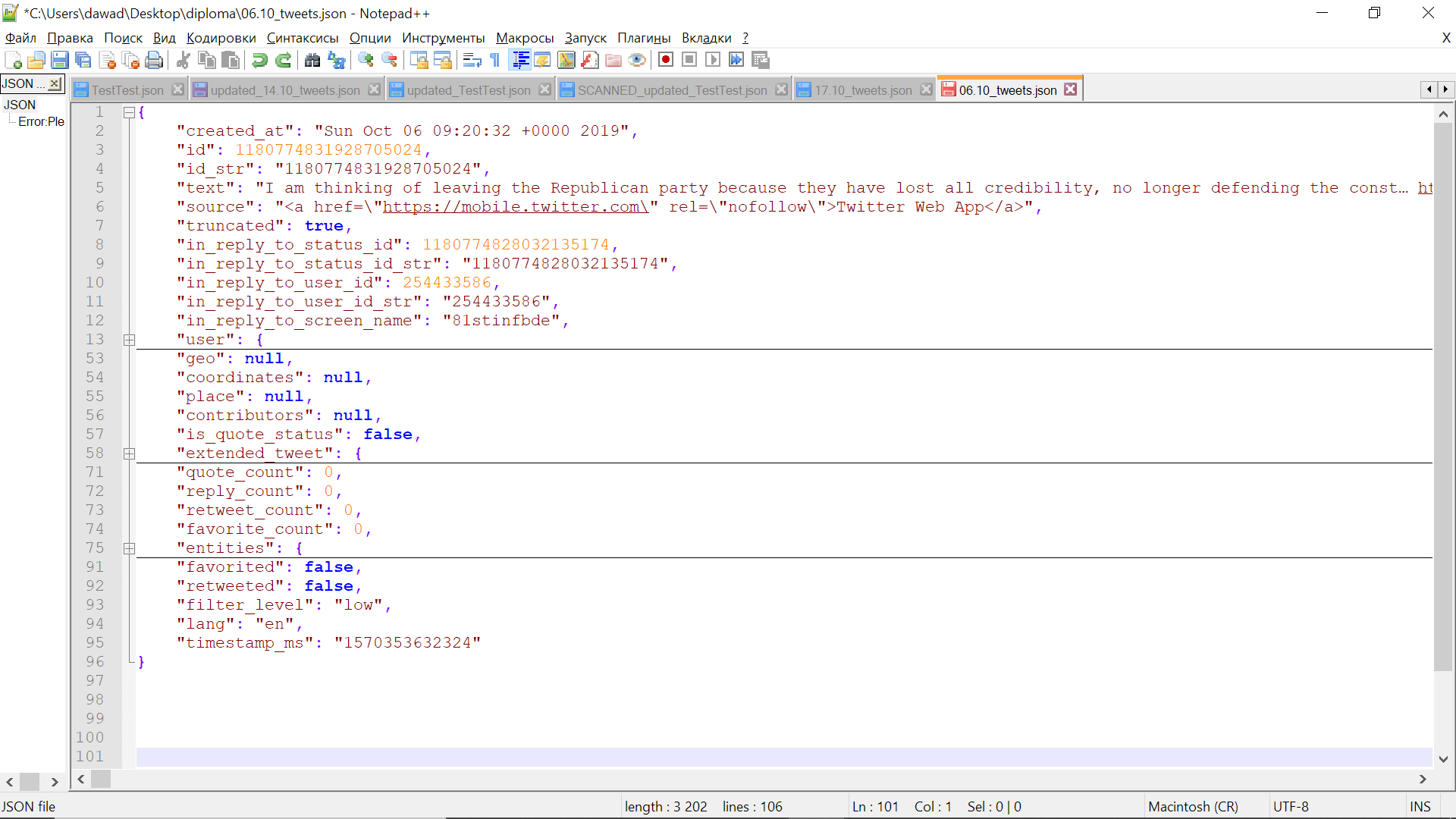


Рисунок 2.2 – Приклад твіта у вигляді JSON-об'єкта

На рисунку 2.3 зображено інформацію усередині обєкта «entities». В цьому полі зберігається інформація про такі складові повідомлення, як хештеги, URL-посилання, згадування інших користувачів в повідомленнях та ін.

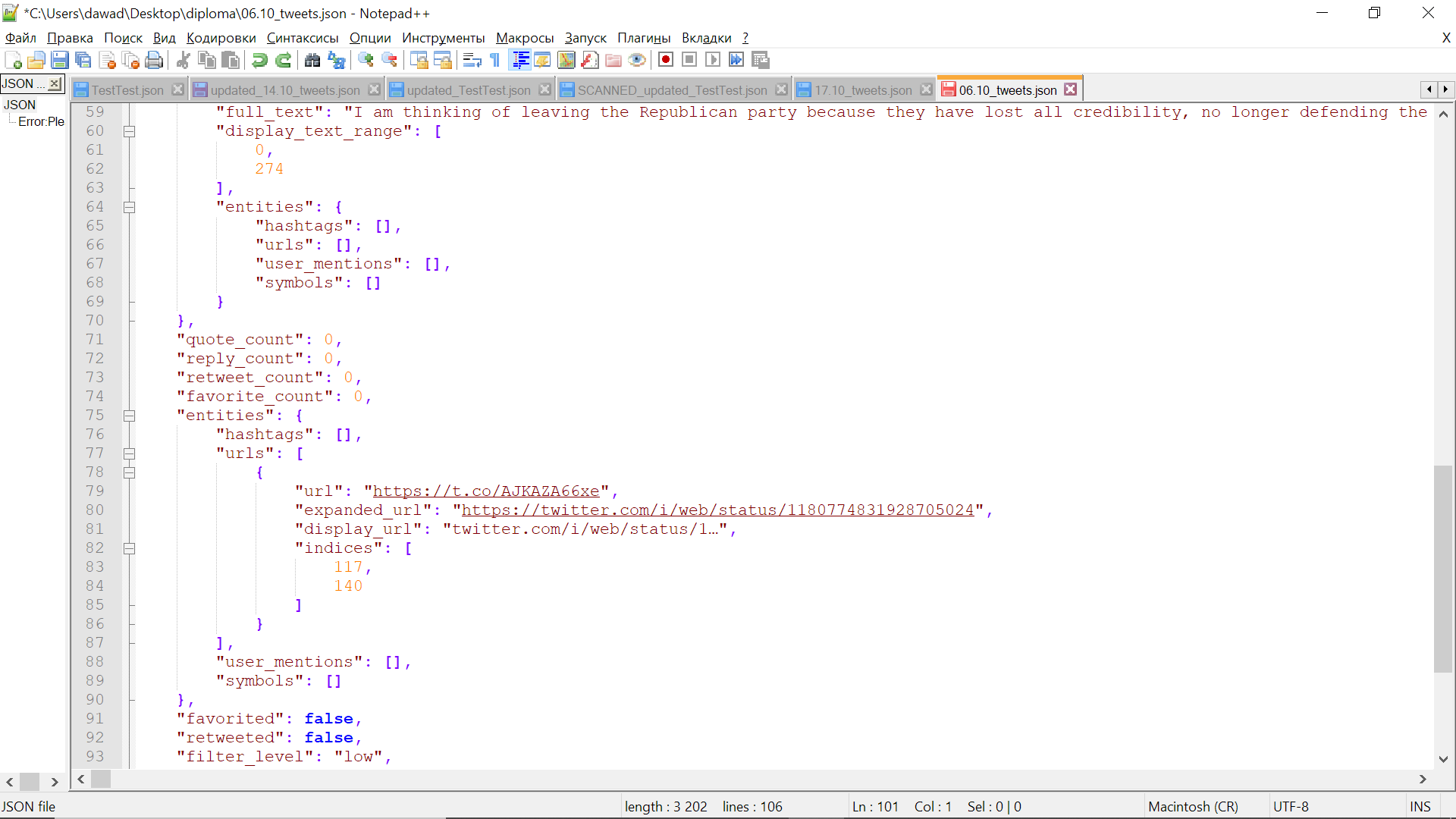


Рисунок 2.3 – Приклад поля ‘entities’ в JSON-об’єкті твіта

Наступною важливою можливістю даного сервісу є отримання даних за певними встановленими характеристиками або ключовими словами. Таким чином відбувається не збір усіх даних, а фільтрація за певними категоріями які є необхідними для розробника [15].

В даній роботі ми зосередимось тільки на тих твітах, які містять URL-посилання та написані англійською мовою для того, щоб спростити подальший аналіз даних.

## 2.2 VirusTotal API

Наступним етапом в формуванні набору даних є «маркування» або «мічення» даних (з англ. labeling). Помічені данні (з англ. Labeled data) – це група зразків даних які помічені одною або декількома мітками Після отриманяя таких даних можна застосовувати моделі машинного навчання, таким чином побудована модель буде мати можливість передбачувати мітки для немічені екземплярів даних.[16]

В даній роботі застосовується VirusTotal API для мічення повідомлень за рахунок аналізу URL-посилань, які були зібрані разом із твітами. VirusTotal є важливим інструментом для досліджень в сфері інформаційної безпеки, threat hunting та ін. Даний інструмент також має власний API інтерфейс, що дає багато можливостей для взаємодії та автоматизації пошуку та аналізу шкідливих програм та посилань. VirusTotal API дозволяє завантажувати та сканувати файли чи URL-адреси, отримувати доступ до готових звітів про сканування та робити автоматичні коментарі без необхідності використання інтерфейсу веб-сайту. Іншими словами, це дозволяє створювати прості сценарії для доступу до інформації, що генерується VirusTotal. [17]

Щоб користуватись сервісом необхідно зареєструватись у VirusTotal Community. Після реєстрації користувачам надається Public API key, цей ключ надає велику кількість можливостей. [17] Проте він також має деякі обмеження, що ускладнюють роботу в розробці більш масштабних проектів. Одним з таких обмежень є ліміт у запитах – 4 запити в одну хвилину. Для того, щоб уникнути цих обмежень користувачам пропонується придбати Private API key. При використанні даного ключа можна отримати більш широкий аналіз та відгук про досліджуваний екземпляр і самостійно контролювати ліміти доступу. На рисунку 2.4 зображено приклад застосування сервісу.

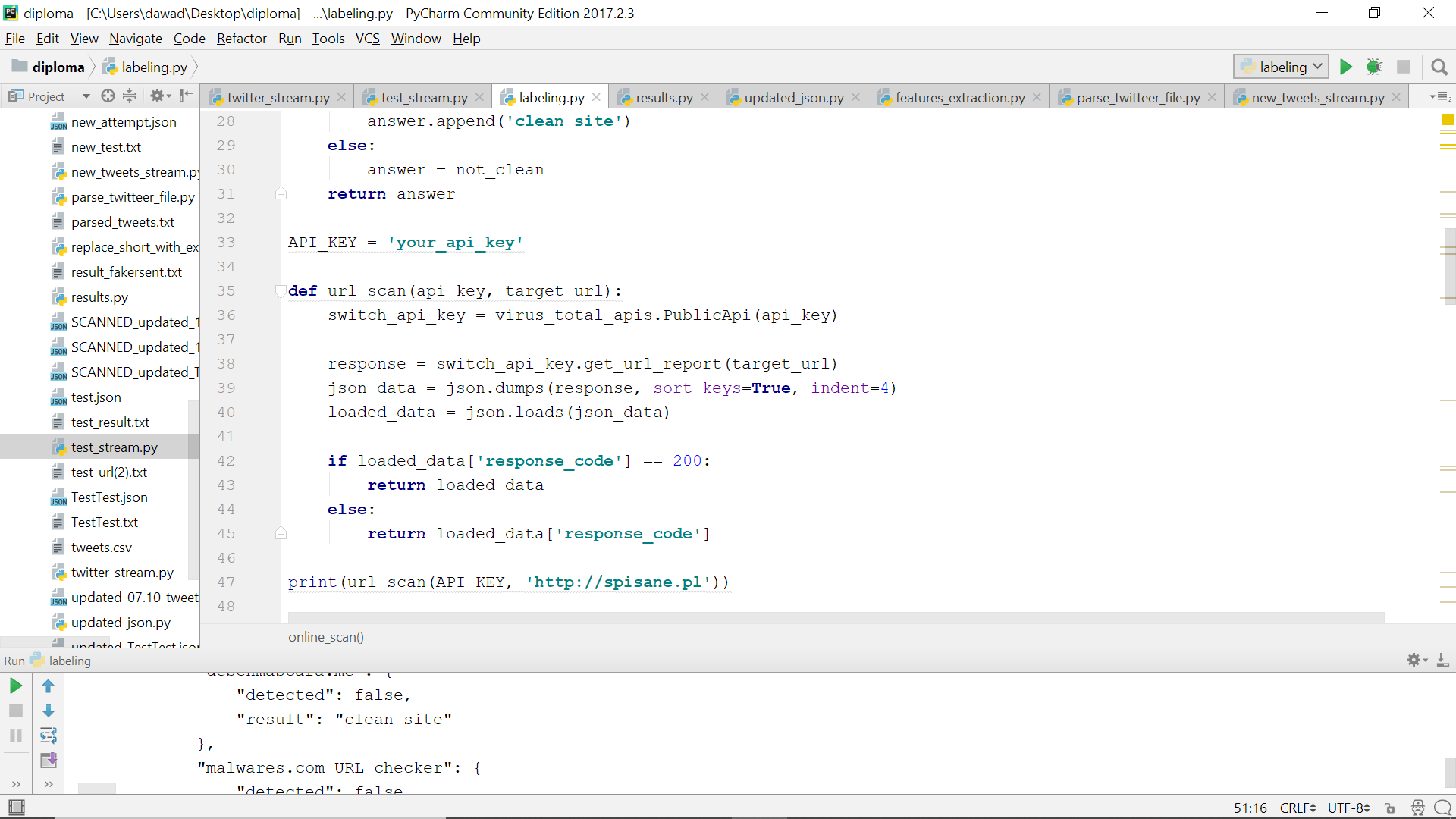


Рисунок 2.4 – Реалізація методу сканування URL-посилання

Даний API сервіс також працює за принципом REST [18], як і Twitter API, але запити обмежуються лише GET та POST методами. Проте нам немає необхідності використовувати усі можливості даного API інтерфейсу для вирішення поставлених завдань. Тому основним запитом буде GET який позначається як /url/report в документації [19]. Відповідь від сервера приходить у форматі JSON в якому зберігається результати кожного з антивірусів. Антивіруси URL-адрес розділяють сайти на різні класи. Окрім «clean site» та «malicious» анивіруси розрізняють фішинг-сайти, спам, сайти із зловмисним програмним забезпеченням тощо. Деякі сканери навіть додають додаткову інформацію, чітко вказуючи, чи належить дана URL-адреса певному ботнету, на яку марку орієнтований фішинг-сайт тощо. Далі показано приклад відповіді VirusTotal API, де ми бачимо як два антивіруси показують такі результати: «phishing site» та «malicious site».

{

"response\_code": 200,

"results": {

"filescan\_id": null,

"permalink": "https://www.virustotal.com/url/91d6c3e53c34593cd19c8b83329df0f84ed01c797005364c1da1fd602ab068bd/analysis/1572970494/",

"positives": 2,

"resource": "http://spisane.pl/",

"response\_code": 1,

"scan\_date": "2019-11-05 16:14:54",

"scan\_id": "91d6c3e53c34593cd19c8b83329df0f84ed01c797005364c1da1fd602ab068bd-1572970494",

"scans": {

"ADMINUSLabs": {

"detected": false,

"result": "clean site"

},

"AegisLab WebGuard": {

"detected": false,

"result": "clean site"

},

. . .

"BitDefender": {

"detected": true,

"result": "phishing site"

},

"Blueliv": {

"detected": false,

"result": "clean site"

},

"CRDF": {

"detected": true,

"result": "malicious site"

},

"DNS8": {

"detected": false,

"result": "clean site"

},

"Dr.Web": {

"detected": false,

"result": "clean site"

},

"ESET": {

"detected": false,

"result": "clean site"

},

. . .

"total": 71,

"url": "http://spisane.pl/",

"verbose\_msg": "Scan finished, scan information embedded in this object"

}

}

Отже, в даній роботі ми пропонуємо використовувати VirusTotal API для маркування набору даних, який ми отримали з Twitter API та побудови набору даних з «ground truth», про що більш докладно буде описано в наступному розділі.

## 2.4 Методи машинного навчання

Використання машинного навчання для вирішення такого типу задач є дуже ефективним. В даному підрозділі ми коротко опишемо теоретичну основу даного підходу та алгоритми які будуть використовувати при вирішенні поставленої задачі.

Вперше термін «машинне навчання» використав Артур Семюель e 1959 році. Тоді він дав таке означення терміну машинне навчання - «Область навчання, яка дає комп'ютерам можливість навчатись, не будучи явно запрограмованими.»

На сьогодні дана галузь є дуже розвиненою і включає велику кількість математичних обґрунтувань та використовується в багатьох сферах життя. Таким чином машинне навчання є більшою мірою розвиненою формою штучного інтелекту, яка широко використовує статистику, розпізнавання шаблонів та виявлення знань та обмін даними. Акцент машинного навчання робиться на автоматичних методах. Іншими словами, мета полягає у розробці алгоритмів навчання, які здійснюють навчання автоматично без втручання людини.

Термін «машинне навчання» можна також описати більш формально, як підгалузь штучного інтелекту яка вивчає алгоритми та статистичні моделі які використовують комп’ютерні системи для вирішення деякої задачі спираючись на закономірності та без використання чітких інструкцій. Алгоритми машинного навчання будують математичну модель на основі вибіркових даних, відомих як "дані тренувань", для того, щоб приймати прогнози або рішення, не будучи явно запрограмованими для виконання завдання.

Спектр задач які можна вирішувати за допомогою методів машинного навчання дуже великий. Враховуючи це, дослідники визначили постійно зростаючу кількість шаблонів, які можна використовувати для вирішення багатьох ситуацій. Такі шаблони поділяються на такі задачі як бінарна класифікація, регресія, кластеризація, зменшення розмірності та ін. Далі ми розглянемо основні моделі машинного навчання в контексті задач які вони вирішують.

Відрізняють такі три основні моделі машинного навчання: контрольоване навчання або «навчання з учителем» (supervised learning), неконтрольоване навчання або «навчання без учителя» (unsupervised learning) (зокрема кластеризація), навчання з підкріпленням (reinforcement learning) [20]. Але зазвичай на практиці використовують перші дві моделі.

Модель навчання з учителем вирішує задачу класифікації, яка полягає в тому, щоб навчити алгоритм відносити екземпляри даних до заздалегідь відомих класів. Тобто, алгоритм має класифікувати об’єкт із заданої множини класів.

Кажучи більш формально, в випадку бінарної класифікації кожна мітка Y приймає одне з двох значень {0,1} (якщо мітки мають інші числові або текстові значення ми завжди можемо відобразити як 0 і 1). Тоді проблема класифікації полягає у використанні таких тренувальних даних, що (*X1, Y1*) , …, (*Xn, Yn*) Є *ꭓ* × {0, 1} для оцінки функції Ŷ, яка з урахуванням вектора ознак *x* Є *X*, виводить передбачувані мітки Ŷ(x) Є {0, 1} (тобто функція Ŷ «класифікує» x як клас 0 або клас 1). Функція Ŷ називається класифікатор.

В свою чергу, алгоритм здатний видати відповідь для об’єкту, якого він до цього не бачив, без допомоги людини. Найбільш відомий приклад роботи такого алгоритму це класифікація спаму, користувач пред’являє алгоритму велику кількість листів (об’єктів) разом з інформацією про те, є лист спамом або ні (відповідь). Для нового електронного листа алгоритм розрахує ймовірність з якою цей лист можна віднести до спаму.

Окрім задач класифікації, до машинного навчання з учителем відносять задачу регресії. Якщо множина значень відповіді кінцева, то це задача класифікації, якщо ж відповіді є дійсними числами або векторами, то це задача регресії [21].

Перейдемо до опису наступної моделі машинного навчання – «навчання без учителя». Якщо «навчання з учителем» ми можемо розглядати як задачу в якій присутні множини об’єктів та відповідей, то «навчання без учителя» це лише – об’єкти. Тобто «навчання без учителя» можна сформулювати, як задачу знаходження взаємозв’язків, залежностей, закономірностей які існують між об’єктами.

Існує два основних підходи до «навчання без учителя»:

* Кластеризація (clustering). Розподіл екземплярів даних за схожими рисами та ознаками і групування цих екземплярів в кластери. До цієї задачі відносяться метод K-means clustering, Gaussian mixture model clustering та Density-based clustering.
* Скорочення розмірності (dimensionality reduction). Вхідні ознаки перетворюються в невелику кількість координат із зйбереженням існуючої інформації про об’єкти. В цю групу відносяться: Principal component analysis, Graph-based kernel PCA, Linear discriminant analysis та інші [22].

Ці задачі є схожими між собою, що навіть можна виділити те, що задача кластеризації є логічним продовженням задачі класифікації, проте вони мають значну відмінність. На відміну від задачі класифікації, в задачі кластеризації класи попередньо не зумовлені і відношення об’єктів до кластерів відбувається відносно даних та ознак об’єктів. Тобто, ціллю кластеризації є пошук закономірностей в даних і розбиття сукупності об’єктів на групи (кластери або класи), а в кожній групі знаходяться об’єкти які схожі за певними ознаками і відрізняються від інших об’єктів даних [24].

Для різниці між навчанням з учителем та без учителя подивимось на рисунок 2.5, на якому зображено порівняння між задачами класифікації та кластеризації.

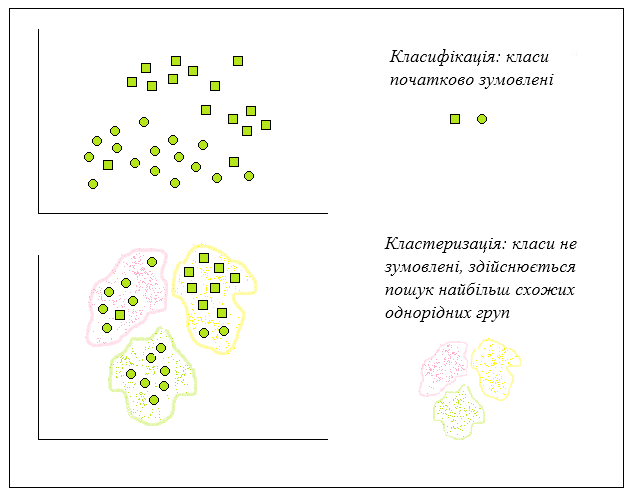


Рисунок 2.5 – Схематичне зображення задач класифікації та кластеризації

Для оцінки найефективнішої методики виявлення зловмисних повідомлень в Twitter ми досліджуємо використання декількох алгоритмів класифікації. Далі ми розглянемо чотири алгоритми класифікації та опишемо модель роботи кожного із них.

Першим алгоритмом який ми розглянемо буде Naive Bayes: Це імовірнісний класифікатор який ґрунтується на теоремі Баєса. Він працює ефективно, коли розмірність вхідного вектора ознак висока і кожна з характеристик об’єкта не залежить одна від одної. На основі кожної функції класифікатор Naive Bayes обчислює ймовірність класифікації точки даних у кожній можливій категорії. Точка даних класифікується в категорію, для якої ймовірність є найбільшою.

Наступний алгоритм це Random Forest Classifier (RFC). RFC є одним із найточніших класифікаторів, і він працює ефективно для великих баз даних. Для кожної точки даних, що підлягає класифікації, ця методика випадковим чином вибирає підмножину функцій, які використовуються для класифікації. Вона вибирає найважливіші особливості точки даних, завдяки чому підвищується точність прогнозування та контролюється надмірне встановлення.

Алгоритм k – nearest neighbor (KNN) – це третій метод який ми будемо використовувати в роботі. Принцип який закладено в даний алгоритм дуже простий, проте при вирішенні багатьох задач може бути дуже ефективним. Алгоритм присвоює об’єкт до того класу, кількість сусідів якого є найбільшою серед усіх інших класів. Сучасні програми алгоритму thek-NN включають системи рекомендацій, категоризацію тексту, класифікацію захворювань серця та прогнозування фінансового ринку.

Ефективність роботи цього класифікатора визначається за рахунок вибору параметру k та відстані між об’єктами. [23]

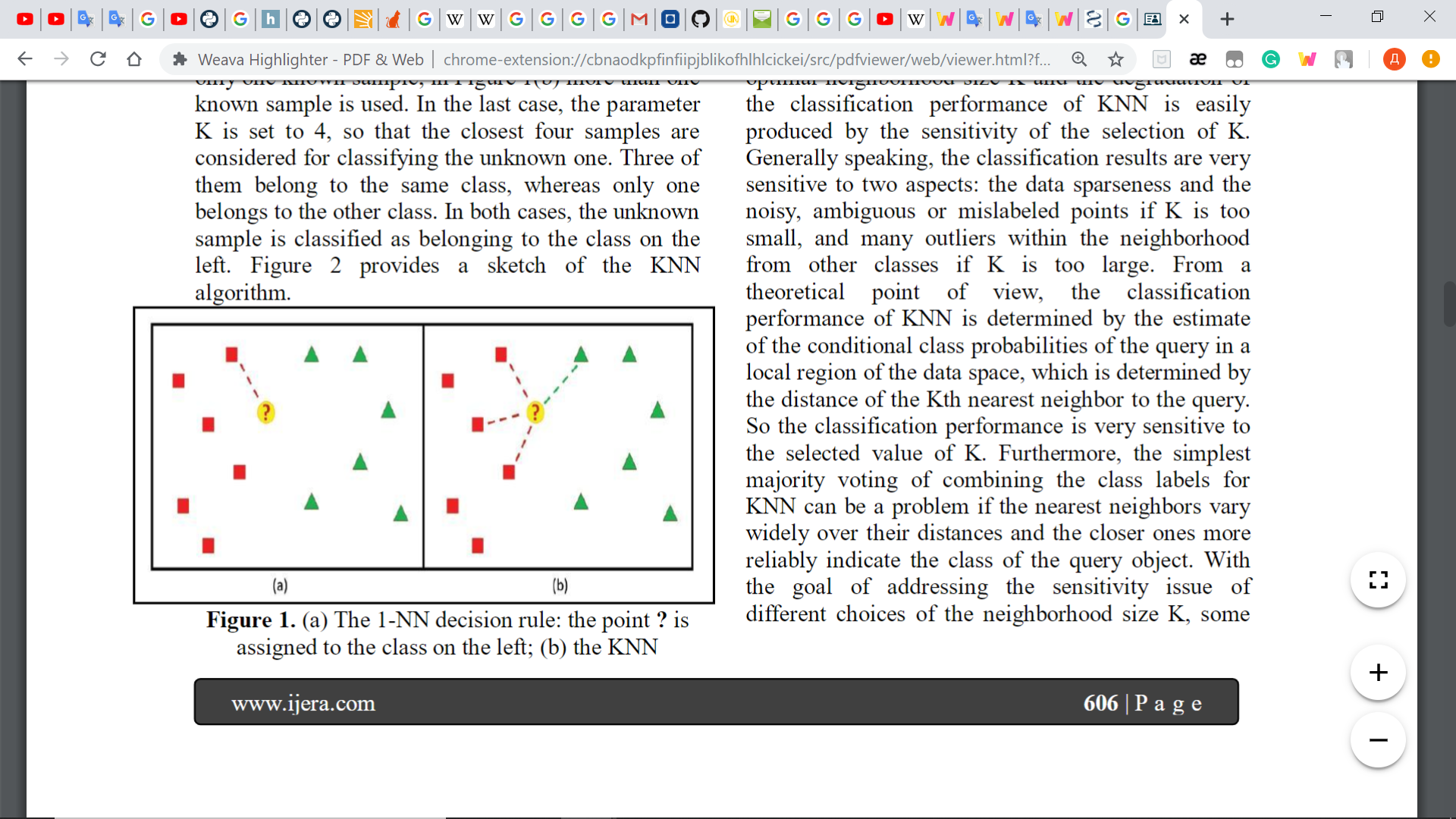


Рисунок 2.6 – Різниця між підходом роботи алгоритма KNN з k=1 та k=4

Одним із шляхів обчислення відстані є формула обчислення Евклідової відстані, яка має наступний вигляд (2.1):

Відповідний вибір k має істотний вплив на діагностичну ефективність алгоритму kNN. Великий k зменшує вплив дисперсії, спричиненої випадковою помилкою, але ризикує ігнорувати невеликий, але важливий зразок. Ключовим моментом для вибору відповідного значення k є встановлення балансу між надмірним і недостатнім. Деякі автори пропонують встановити k рівним квадратному кореню кількості спостережень у навчальному наборі даних. [25]

Останній алгоритм який ми будемо використовувати в роботі має назву Supported Vector Machine (SVM). Дану модель можна використовувати як для вирішення задач класифікації, так і для регресії. Кажучи більш формально, алгоритм конструює гіперплощину або набір гіперплощин у просторі з високим або нескінченним розміром, де екземпляр класу розглядається як вектор розмірності p. Таким чином, задача алгоритму полягає в тому, щоб дізнатись, чи можливо розділити дані точки, об’єкти даних, гіперплощиною розмірності p-1. Гіперплощина вважається оптимальною, якщо розподіл та різниця між класами максимальна.

Таким чином пошук гіперплощини, яка б оптимально розділяла класи між собою, і є основною задачею даного алгоритму. В випадку бінарної класифікації гіперплощиною є лінія, яка розділяє площину на два класи. Проте алгоритм не рахує відстань до всіх точок (так як це є дуже трудомістким процесом), алгоритм намагається знайти максимальну відстань виключно між точками, які розташовані близько до границі поділу класів [25].

На рисунку 2.7 зображено різні варіанти розташування гіперплощин. І можемо бачити, що серед трьох варіантів лінія H3 буде найоптимальнішою.

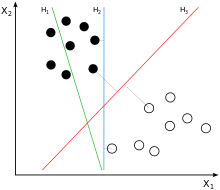


Рисунок 2.7 – Гіперплощини моделі SVM

## 2.5 Метрики оцінки якості роботи алгоритмів

Після побудови класифікатора та отримання результатів, не менш важливим етапом є оцінка ефективності роботи системи, що власне і буде результатом роботи.

При вирішені задачі класифікації даних метрика оцінки використовується в два етапи, які поділяються на навчальний етап (процес навчання) та етап тестування. На етапі навчання оцінювання використовується для оптимізації алгоритмів класифікації. Іншими словами, метрика оцінювання використовується для вибору оптимального рішення, яке може дати більш точний прогноз майбутнього оцінювання певного класифікатора. Тим часом, на етапі тестування, оцінка виконується на основі метрик які вимірюють ефективність класифікатора при тестуванні нових даних, які не були промарковані. [26]

Для проблем бінарної класифікації оцінку оптимального рішення можна визначити за матрицею помилок («confusion matrix»), яка показана в таблиці 2.1. Рядки матриці помилок відповідають класу який булло передбачено завдяки роботі класифікатора, а стовпці відповідають класу із фактичними даними. З матриці можна сформувати наступні значення для TP, TN, FP та FN.

Значення TP та TN позначають кількість позитивних і негативних випадків які було правильно класифіковано. Тим часом, FP та FN відносять до кількості неправильно класифікованих негативних та позитивних випадків, відповідно. З матриці помилок формуються основні метрики за якими оцінюють продуктивність класифікатора з різними фокусами оцінки. Тобто, для розуміння повноти інформації необхідно отримати цілу низку оцінок і робити висновки на різних метриках, щодо роботи класифікатора.

Опишемо більш докладно значення, що описують матрицю помилок. Перш за все «true positive» (TP) – кількість вірно передбачених об’єктів позитивного класу, коли «true negative» (TN) – кількість вірно передбачених об’єктів негативного класу. Далі «false positive» (FP) – кількість об’єктів позитивного класу помилково віднесених до негативного і «false negative» (FN) – кількість об’єктів негативного класу помилково віднесених до позитивного. Помилки false negative та false positive при виявленні зловмисних повідомлень відіграють дуже важливу роль. Таким чином, такі системи вимагають мінімізацію відсотка таких помилок. Ці значення також мають назву помилки першого та другого роду, відповідно.

Таблиця 2.1 – Матриця помилок

|  |  |  |
| --- | --- | --- |
| Class | Actual Positive Class | Actual Negative Class |
| Predicted Positive Class | TP | FN |
| Predicted Negative Class | FP | TN |

Спершу розглянемо одну з найчастіше використовуваних метрик – Accuracy. Ця метрика є найбільш використовуваною на практиці метрикою як для бінарних, так і для багато класових класифікаторів. Якість роботи алгоритмів оцінюється через Accuracy за рахунок відсотку правильних передбачень відносно загального числа екземплярів класів. Таким чином, з формули 2.2 ми бачимо як співвідносяться значення матриці помилок між собою.

Перевагами цього показника є те, що він дуже легко обчислюється, може бути застосована як для задач з двома класами, так і для задач з більшим числом класів, також вона є інтуїтивно зрозумілою та простою для людей. Проте, недоліком метрики Accuracy є те, що її результат дуже залежить від збалансованості даних. Якщо дані незбалансовані, тобто кількість екземплярів одного класу набагато менша від кількості екземплярів іншого класу, то значення метрики буде все одно достатньо високим.

В випадку незбалансованих даних хорошими метриками для обчислення точності роботи класифікатора будуть Precision та Recall. Ці метрики обчислюють точність виявлення класів не залежно один від одного.

Таким чином Precision та Recall можна формально описати наступними формулами 2.3 та 2.3, відповідно:

Якщо описати дані метрики неформально, то можна сказати, що Precision – це частка всіх виявлених аномалій, які є справжніми аномаліями, тоді як, Recall –це частка усіх реальних аномалій, які успішно було виявлено системою. В такому разу, ці метрики являються додатковими, і вони стають корисними при їх поєднанні, наприклад за допомогою . Такі комбінації допомагають оцінити якість прогнозів аномалій. Таким чином, наступною важливою метрикою і буде . Можна сказати, що дана метрика є середнім гармонічним між двома останніми значеннями, що ми власне і можемо побачити з формули 2.5.

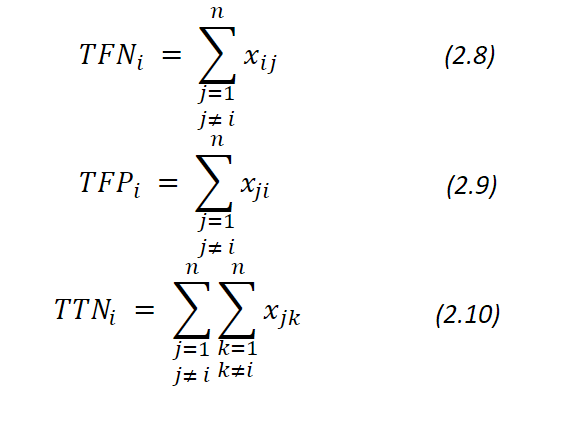
Для систем виявлення спаму або систем виявлення вторгнень важливим є обчислення рівня помилок першого та другого роду, що серед метрик машинного навчання носить назви False Positive Rate та False Negative Rate, відповідно.

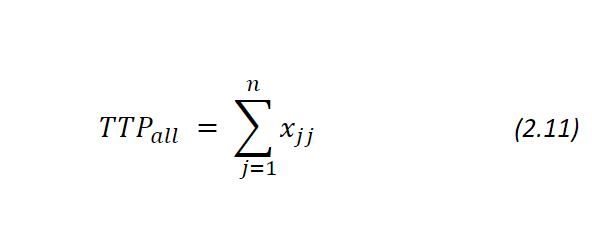
В випадку багатокласової класифікації, принцип побудови матриці помилок та обчислення помилок першого та другого роду залишається таким же. Проте так як при вирішенні поставленої задачі класифікатор працює на виявлення більше ніж двох класів, опишемо в узагальненому вигляді матрицю помилок та метрики Precision та Recall.

Таблиця 2.2 – Матриця помилок для багатокласової класифікації

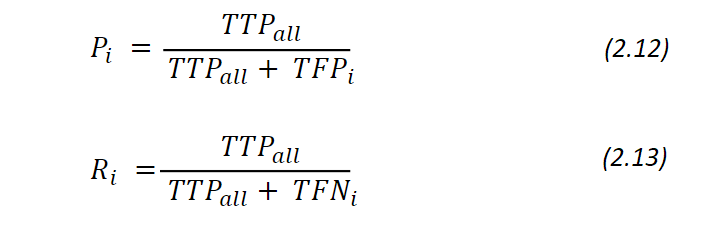
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | *Передбачені значення* | | | |
| *Клас 1* | *Клас 2* | *…* | *Клас N* |
| *Дійсні значення* | *Клас 1* | *X11* | *X12* | *…* | *X1n* |
| *Клас 2* | *X21* | *X22* | *…* | *X2n* |
| *…* | *…* | *…* | *…* | *…* |
| *Клас N* | *Xn1* | *Xn2* | *…* | *Xnn* |

Для обчислення метрик необхідно обрахувати усі загальні значення TN, TP, FN та FP. З таблиці 2.2 легко вивести наступні формули [27]:





Таким чином, формули для метрик Precision, Recall можна формально описати таким чином (2.12 та 2.13, відповідно).

**

З формул 2.12 та 2.13 (Pi відповідає Precision для кожного класу, Ri відповідає Recall також для кожного з класів) бачимо, що вони розраховуються для кожного класу про що говорять значення TFPi та TFNi.

## Висновки до розділу 2

В другому розділі даної роботи було описано теоретичні відомості, щодо методів які будуть використовуватись для вирішення поставлених завдань. Описано програмний інтерфейс Twitter API за допомогою якого виконуєтьс процес збору дані, програмний інтерфейс VirusTotal API, що приймає участь в маркуванні даних на основі аналізу URL-посилань які присутні в повідомленнях.

Коротко розглянуто основині принципи роботи моделей машинного навчання «з учителем» та «без учителя» та описано алгоритми класифікації, які будуть застосовуватись у дослідженні. Також формально описано метрики за якими оцінюється якість роботи системи виявлення повідомлень.

# 3 РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОГО РІШЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Практична частина даної роботи складається з декількох частин. Після аналізу відкритих даних та поставлення завдань роботи було вирішено будувати власний набір даних, що і є першим кроком. Таким чином, дані збираються за допомогою прикладного програмного інтерфейсу Twitter. Так як одним із завдань системи є класифікація, то набір даних має бути з «ground truth», тобто наступним етапом буде маркування даних, для чого використовується VirusTotal API. Перед застосуванням алгоритмів класифікації та тестування системи необхідно визначити ознаки даних за якими буде працювати класифікатор. Після цього до даних застосовуємо 4 алгоритми класифікації: K-nearest neighbors, Support Vector Machine, Naïve Bayes та Random Forest Classifier. Останнім кроком буде аналіз отриманих результатів за допомогою метрик машинного навчання.

## 3.1 Процес отримання набору даних

Для збору даних було вирішено використовувати Twitter Streaming API та вбудовану бібліотеку мови програмування Python – Tweepy. Даний сервіс забезпечує доступ до 1% усіх публічних твітів, але не надає доступ до твітів, надісланих захищеними акаунтами або персональними повідомленнями.

Клас API надає доступ до всіх методів API RESTful. Сервіс працює як класичний API додаток за принципом запит-відповідь, таким чином, кожен метод приймає параметри та повертає екземпляр класу Tweepy, що буде містити дані повернені з Twitter, які ми зможемо використовувати всередині додатку.

Кожен твіт повертається у форматі JSON-об’єкт (рис. 3.1), що є дуже зручним для доступу до даних, так як кожен рядок цього формату являє собою об’єкт.

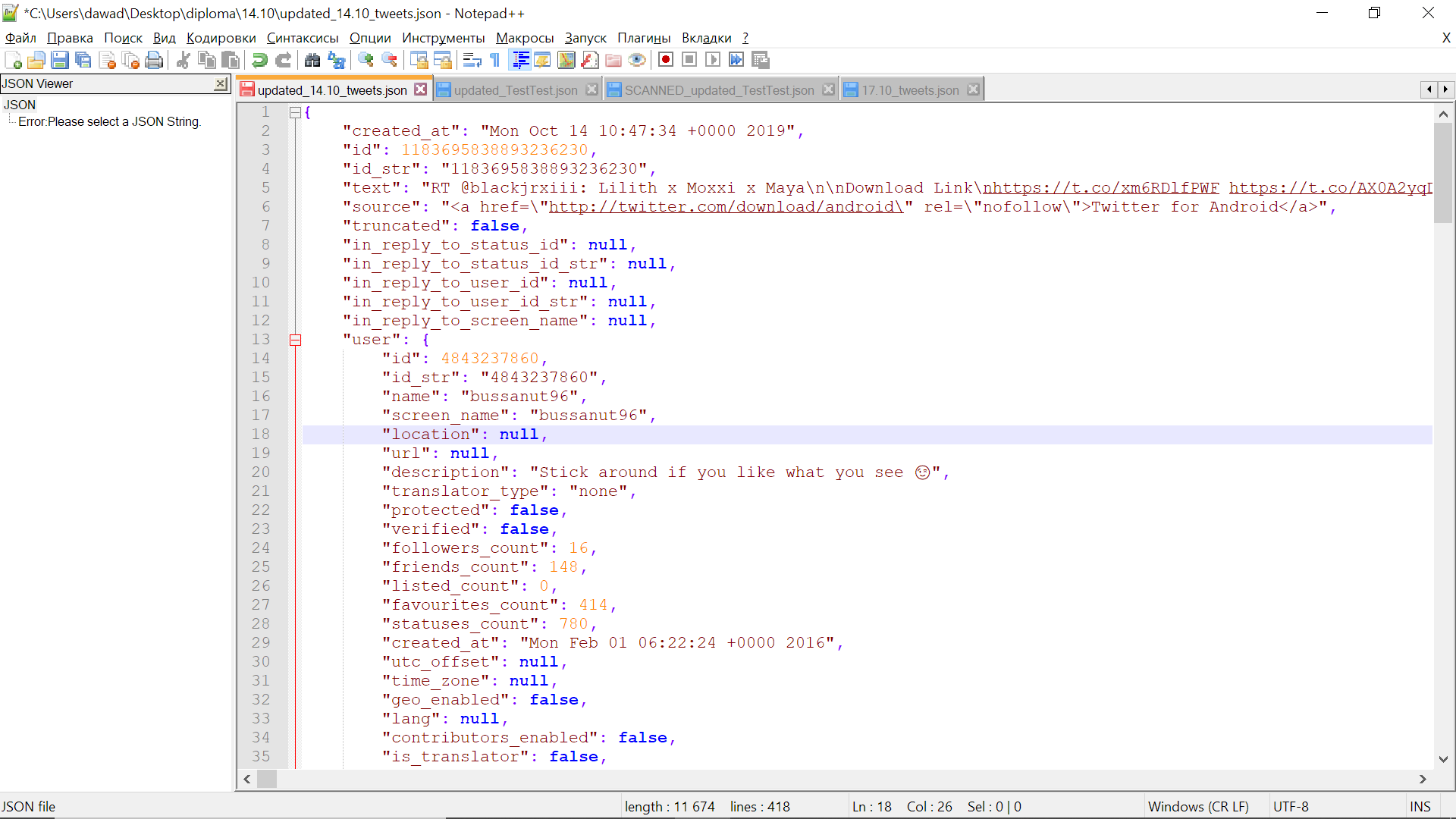
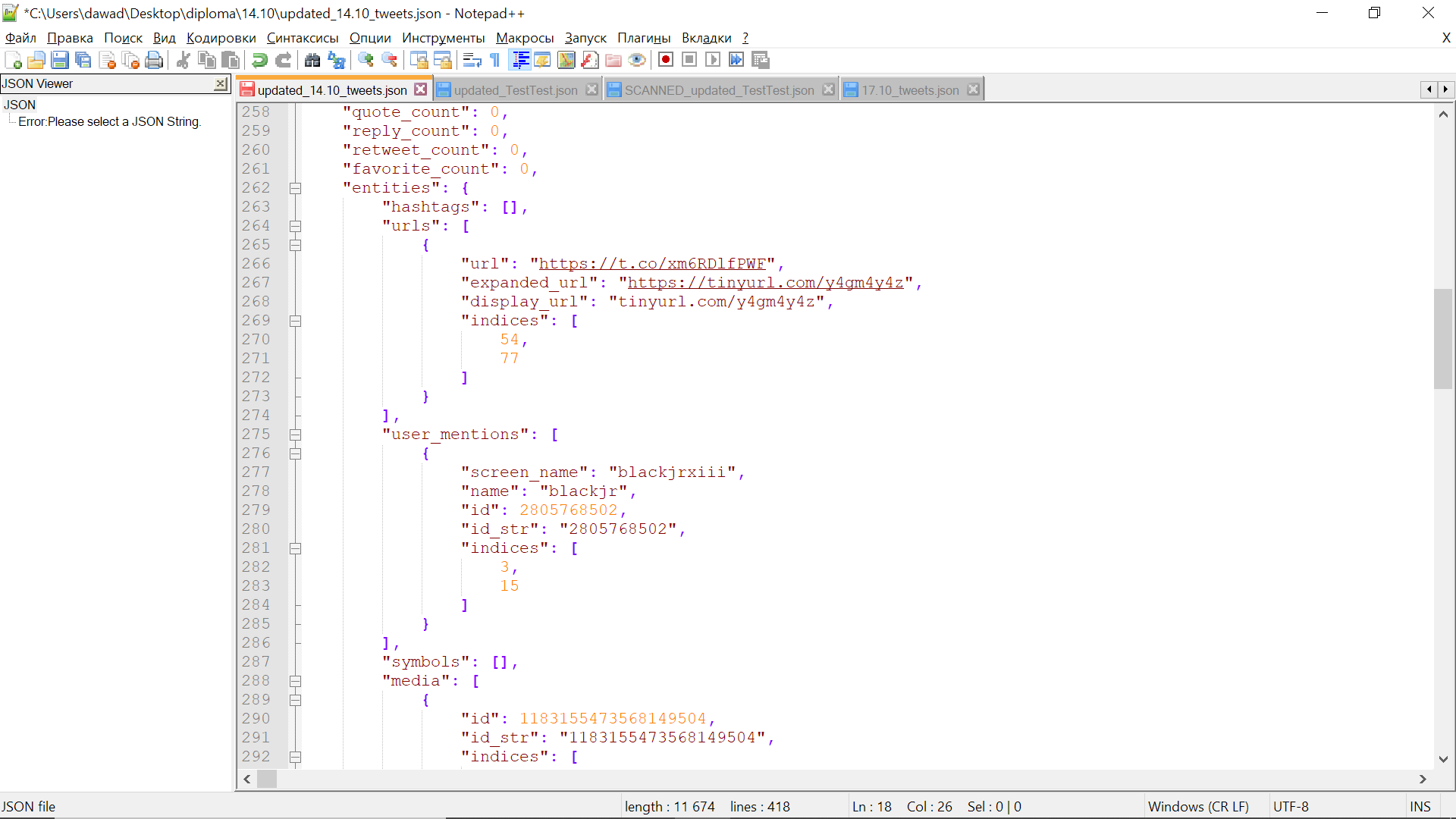
 

Рисунок 3.1 – Приклад твіта в форматі JSON-об’єкта

Проте надсилання спаму та інших зловмисних повідомлень є можливим і без використання URL посилань. Проте, дослідження [28] показало, що більшість зловмисних повідомлень містять URL-адреси. Таким чином було вирішено збирати тільки ті твіти, які містять URL-посилання. Реалізацію процесу збору даних винесено в Додаток А.

Проте після зібраних даних необхідно було попередньо обробити їх перед застосуванням маркування.

Таким чином, було виявлено велику кількість скорочених URL-посилань, які також можуть приховувати адресу зловмисного сайту. Проте, так як користувачі використовують різні сервіси для зменшення посилань, а бібліотек Python для розширення будь-яких адрес ми не знайшли було вирішено використовувати таку техніку як Web-scraping. Було написано скрипт який надсилає POST запит на web-сервіс checkshorturl.com та у відповідь отримує html код з відповіддю на запит, тобто розширеним посиланням. Далі ми розпарсили html-сторінку та отримали лише необхідну нам адресу. Таким чином усі екземпляри даних, які містили скорочені адреси було замінено і ми додали новий флаг до твіту – was\_expanded, який приймає два значення 1 та 0 і відповідно означає було розширено адресу чи ні. Надалі ми побачимо, що цей флаг також корелює із екземплярами які були віднесені до класів із зловмисними повідомленнями. Реалізацію даного процесу можна побачити в Додатку Б.

## 3.2 Маркування даних з використанням VirusTotal

В даній роботі нам необхідно мати набір даних з ground truth. Ground truth – це анотовані екземпляри даних з виділенням двох або більше класів, що також можна назвати маркуванням даних. Даний процес необхідний для побудови класифікатора.

На сьогодні дослідники використовують два методи для генерування таких даних це «manual inspectation» та «blacklisting». Проте виконувати ручну анотацію даних не є ефективним для нашої задачі, так як є дуже ресурсомістким та часомістким процесом для нашого об’єму даних. Таким чином, для маркування URL адрес ми вирішили використовувати VirusTotal API.

Даний API сервіс також працює за принципом REST, як і Twitter API, але запити обмежуються лише GET та POST методами. Проте нам немає необхідності використовувати усі можливості даного API інтерфейсу на даному етапі роботи. Тому основним запитом буде GET який позначається як /url/report в документації. Відповідь від сервера приходить у форматі JSON в якому зберігається результати кожного з антивірусів. VirusTotal складається з більше ніж 60 сканерів url-адрес, що збільшує вірогідність виявлення зловмисних послань. Також, сканери розмежують посилання на сайти із зловмисним програмним забезпеченням, фішинг-сайти, підозрілі сайти тощо. Деякі антивіруси надають додаткову інформацію, чітко вказуючи, чи належать дана URL-адреса певному ботнету, на яку марку орієнтований даний фішинг-сайт та інше.

Таким чином ми виявили 3 основних класи зловмисних URL-адрес це spam, phishing та malware, які можна спостерігати в таблиці 3.1 та їх відсотковий розподіл між позитивними url-адресами. Код із використанням VirusTotal для маркування даних можна побачити в Додатку С.

Таблиця 3.1 – Розподіл даних на класи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| All data | Normal | Spam | Phishing | Malware |
| 182400 | 180073 | 1854 | 460 | 13 |
| 100% | 98.7% | 1.03% | 0.26% | 0.01% |

Можемо бачити, що найбільш розповсюдженим видом зловмисних URL-адрес в соціальній мережі Twitter є спам. Також, хоча й кількість посилань які антивіруси віднесли до посилань, що містять зловмисне ПЗ, не так й багато, все ж такий відсоток є достатньо високим для такого невеликого об’єму даних.

## 3.3 Виділення важливих ознак даних

Наступним кроком після маркування даних було виділення важливих ознак даних. Виділяння ознак – це процес формування та визначення характеристик початкового набору даних, які є інформативними для поставленої задачі та покликані бути ненадлишковими. Оскільки Twitter Streaming API лише повертає випадкові твіти які не пов’язані між собою, ми не змогли створити соціальний графік із даних. Як наслідок, нам неможливо виділити такі соціальні ознаки, як коефіцієнт локальної кластеризації (Local Clustering Coefficient), центральність між ними (Betweenness Centrality) та відстань (distance). Такі вигідні функції не підходять для використання в режимі реального часу виявлення, незважаючи на те, що вони мають більш дискримінаційну силу при розділенні зловмисних повідомлень від позитивних. Більше того, ми спеціально зосереджуємось на ознаках які легко виявити та обчислити з самого твіта, щоб модель могла працювати в реальному часі. Таким чином, з нашого набору даних ми виявили такі 9 ознак, які перелічені в Таблиці 3.2.

Таблиця 3.2 – Назви та опис ознак твітів

|  |  |
| --- | --- |
| *Назва ознаки* | *Опис ознаки* |
| account\_age | час від створення акаунту до часу публікації останнього твіту |
| no\_follower | кількість фоловерів даного користувача |
| no\_following (no\_friends) | кількість друзів у даного користувача |
| user\_fav\_count | кількість усіх вподобань користувача |
| no\_tweets | кількість надісланих твітів |
| retweet\_count | кількість перепостів даного твіта |
| favourite\_count | кількість вподобань даного твіта |
| no\_hashtags | кількість хештегів |
| was\_expanded | 1 – посилання було скорочено; 0 – посилання не було скорочено |

Реалізацію процесу виділення ознак даних також можна побачити в Додатку Е.

Отже, дані які будуть подаватись на вхід до класифікатора будуть мати наступний вигляд (Рис 3.2).

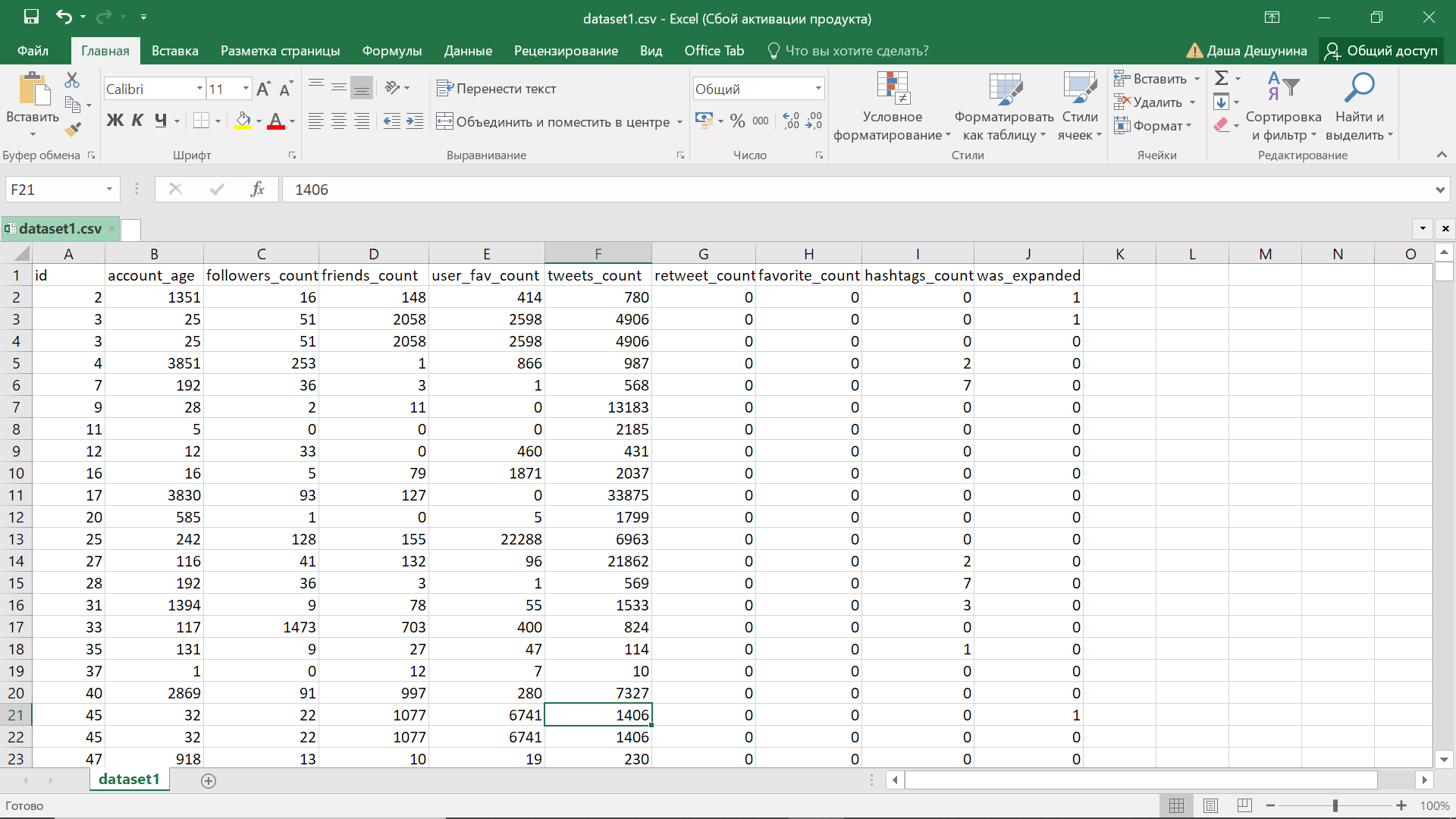


Рисунок 3.2 – Вигляд опрацьованого набору даних в csv форматі

## 3.4 Реалізація запропонованої моделі виявлення зловмисних повідомлень із застосуванням алгоритмів машинного навчання

Після збору даних, їх обробки, вибору важливих характеристик та маркування, дані набули вигляду та формату, щоб можна було застосовувати алгоритми машинного навчання та оцінювати якість та можливість моделі виявляти зловмисні повідомлення. Таким чином на наступному рисунку (Рис. 3.3) можна побачити послідовність дії від збору даних до формування готового набору даних який готовий до роботи із класифікатором.

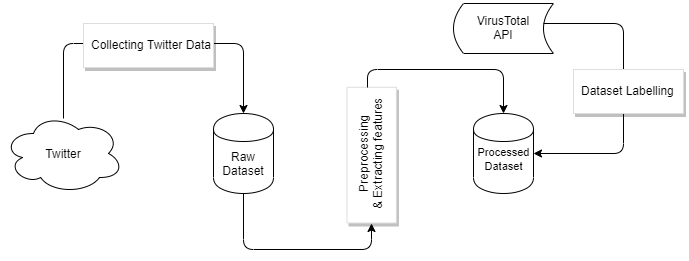


Рисунок 3.3 – Процес збору та опрацювання даних

Після отримання набору даних з «ground truth» та після виділення ознак повідомлень, ми побудували класифікатор та обчислили якість його роботи. На наступному рисунку 3.4 зображено принцип навчання та тестування роботи класифікатора.

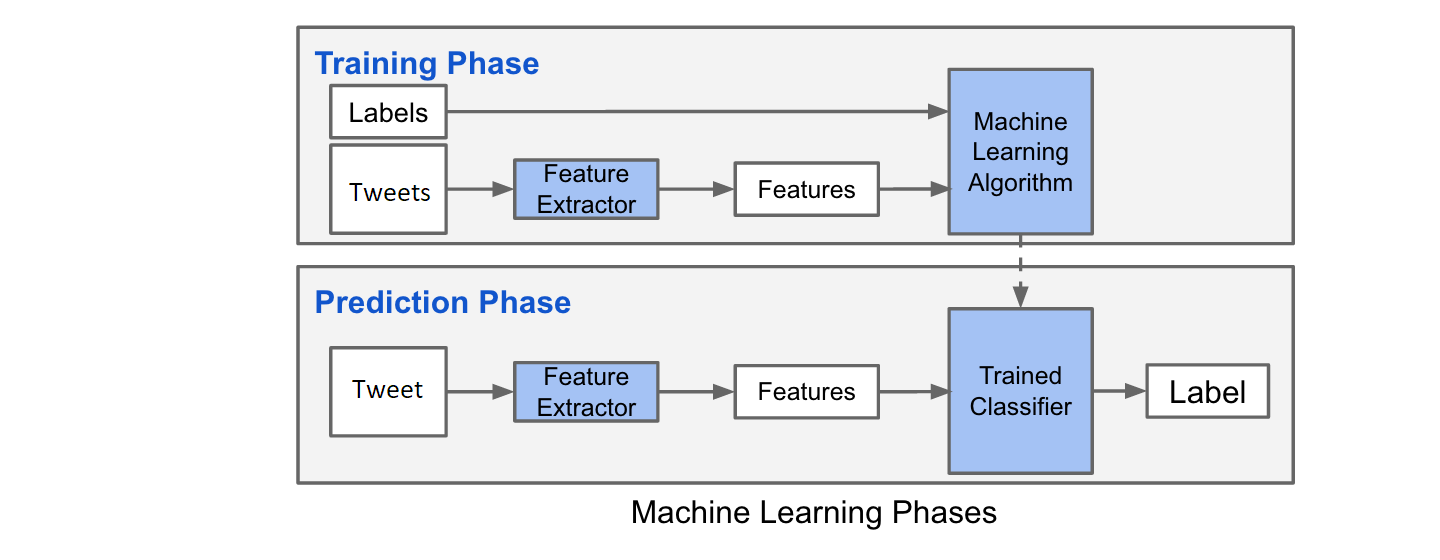


Рисунок 3.4 – Процес навчання та класифікації твіта

Щоб обрати оптимальний алгоритм, ми провели порівняльний аналіз між такими алгоритмами, як: k-NN, SVM, Naive Bayes та Random Forest.

Усі алгоритми, які ми використовуємо у нашому дослідження були взяті з бібліотеки Scikit-learn. Цей інструмент є одним із основних, для дослідників які працюють з методами машинного навчання. Бібліотека включає в себе реалізацію багатьох алгоритмів як для навчання з викладачем (Supervised Learning), так і для навчання без викладача (Unsupervised Learning).

Окрім алгоритмів машинного навчання, в scikit-learn присутні алгоритми обчислення метрик для визначення якості та точності алгоритмів, які ми також використовували, для отримання результатів та висновків.

## 3.5 Аналіз результатів роботи системи

В даному підрозділі ми опишемо результати метрик які оцінюють точність роботи системи виявлення зловмисних повідомлень. Також ми опишемо статистичні дані та ознаки за якими поведінка зловмисників відрізняється від звичайних користувачів.

## 3.5.1 Оцінка алгоритмів за метриками Precision, Recall та F1-score

Як вже було зазначено раніше для порівняння алгоритмів класифікації та роботи системи виявлення було обрано чотири алгоритми: K-nearest neighbors, Support Vector Machine, Naïve Bayes та Random Forest Classifier. Аналіз ефективності алгоритму буде визначатись відносно значень таких метрик, як Precision, Recall та F-1 score.

Усі метрики які ми обрали для обчислень можна визначити за допомогою матриці помилок. Таким чином далі ми покажемо роботу класифікаторів та розподіл значень між класами на основі матриці помилок.

Після побудови та аналізу матриць помилок усіх обраних алгоритмів, нам необхідно обчислити обрані метрики для більш ґрунтовного та формального висновку, щодо роботи системи.

Так як наші дані розподілені на більше ніж два класи, тобто в нашому випадку ми маємо багатокласову класифікацію, то і матриці помилок будуть розподілені на 4 класи.

Перша таблиця відповідає значенням які були отримані після роботи класифікатора KNN (Таблиця 3.1).

Таблиця 3.1 – Матриця помилок для алгоритму KNN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Actual | | | |  |
|  |  | Normal | Spam | Phishing | Malware |  |
| Predicted | Normal | 44163 | 129 | 9 | 0 | 44301 |
| Spam | 398 | 231 | 22 | 2 | 653 |
| Phishing | 381 | 76 | 56 | 1 | 514 |
| Malware | 77 | 27 | 28 | 0 | 132 |
|  |  | 45019 | 463 | 115 | 3 |  |

Бачимо, що через те, що класи є дуже незбалансованими то точність виявлення об’єктів нормального класу буде завжди високою, проте виявлення класів які відносяться до зловмисних є більш складною задачею для алгоритмів. Так алгоритм KNN виявив 132 об’єкти які відніс до Malware причому жоден з об’єктів не виявився TP, що говорить про дуже високий відсоток помилок першого роду (false positive).

Таким чином, за допомогою матриці помилок ми визначили наступні значення метрик (табл. 3.2):

Таблиця 3.2 – Значення метрик для алгоритму KNN

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-score |
| Normal | 0.99 | 0.98 | 0.985 |
| Spam | 0.35 | 0.49 | 0.42 |
| Phishing | 0.10 | 0.48 | 0.29 |
| Malware | 0.0 | 0.0 | 0.0 |

Отже, через великий відсоток помилок другого роду значення метрики Precision для усіх класів набагато менш.

Далі розглянемо результати роботи алгоритму Support Vector Machine на основі матриці помилок (табл. 3.3)

Таблиця 3.3 – Матриця помилок для алгоримту SVM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Actual | | | |  |
|  |  | Normal | Spam | Phishing | Malware |  |
| Predicted | Normal | 44568 | 181 | 17 | 0 | 44766 |
| Spam | 321 | 178 | 23 | 0 | 522 |
| Phishing | 112 | 89 | 52 | 0 | 253 |
| Malware | 18 | 15 | 23 | 3 | 59 |
|  |  | 45019 | 463 | 115 | 3 |  |

Аналізуючи отримані результати, цікавим є те, що для класу Malware, алгоритм виявив усі значення TN, що далі в метриках ми побачимо. Проте, не дивлячись на це, кількість помилок другого роду достатньо висока, що вплинуло на значення метрики Precision. З цього можна зробити висновок, що даний алгоритм здатний мінімізувати помилки першого роду для класу Malware, але так як цей клас заздалегідь має дуже низьку кількість об’єктів, а рівень помилок другого роду все ж є високим, складно зробити висновок щодо точності роботи алгоритму та якості виявлення тих чи інших ознак.

Таблиця 3.4 – Метрики Precision, Recall, F1-score для SVM

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-score |
| Normal | 0.99 | 0.98 | 0.985 |
| Spam | 0.34 | 0.38 | 0.36 |
| Phishing | 0.21 | 0.45 | 0.33 |
| Malware | 0.05 | 1.0 | 0.53 |

Наступна матриця помилок відноситься до алгоритму Naïve Bayes (табл. 3.5).

З таблиці 3.5 ми бачимо набагато кращі результати відносно попередніх двох алгоритмів, значення помилок першого та другого роду менші.

Таблиця 3.5 – Матриця помилок алгоритму Naïve Bayes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Actual | | | |  |
|  |  | Normal | Spam | Phishing | Malware |  |
| Predicted | Normal | 44846 | 63 | 15 | 2 | 44926 |
| Spam | 138 | 258 | 18 | 0 | 414 |
| Phishing | 27 | 131 | 76 | 0 | 234 |
| Malware | 8 | 11 | 6 | 1 | 26 |
|  |  | 45019 | 463 | 115 | 3 |  |

Таким чином, значення метрик для алгоритму Naïve Bayes можна бачити в таблиці 3.6.

Таблиця 3.6 – Значення метрик Precision, Recall, F1-score для Naïve Bayes

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-score |
| Normal | 0.99 | 0.99 | 0.99 |
| Spam | 0.62 | 0.55 | 0.58 |
| Phishing | 0.32 | 0.66 | 0.49 |
| Malware | 0.038 | 0.33 | 0.18 |

З наступної таблиці, де зображено результати роботи моделі Radom Forest Classifier, ми чітко бачимо значну різницю між попередніми моделями. Цей алгоритм показує найкращі результати з найменшими показниками помилок першого та другого роду і є найефективнішим при вирішені поставленої задачі.

Таблиця 3.6 – Матриця помилок для алгоритму Random Forest Classifier

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Actual | | | |  |
|  |  | Normal | Spam | Phishing | Malware |  |
| Predicted | Normal | 44978 | 116 | 10 | 0 | 45104 |
| Spam | 36 | 326 | 38 | 1 | 401 |
| Phishing | 4 | 19 | 63 | 0 | 86 |
| Malware | 1 | 2 | 4 | 2 | 9 |
|  |  | 45019 | 463 | 115 | 3 |  |

Таблиця 3.7 – Результати метрик алгоритму Random Forest Classifier

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-score |
| Normal | 0.99 | 0.99 | 0.99 |
| Spam | 0.81 | 0.70 | 0.75 |
| Phishing | 0.73 | 0.54 | 0.63 |
| Malware | 0.22 | 0.66 | 0.44 |

Таким чином, проаналізувавши значення матриць помилок та метрик до кожного алгоритму, можна зробити висновок, що Random Forest Classifier є найбільш ефективним та точним алгоритмом. Отже, для побудови системи виявлення зловмисних повідомлень.

## 3.5.2 Визначення найбільш важливих характеристик зловмисних повідомлень

В даному підрозділі ми визначаємо які з обраних ознак в нашому наборі даних мають найбільшу вагу при визначення зловмисних повідомлень в Twiter на основі результатів алгоритму Random Forest Classifier.

Знання важливих ознак даних може грати важливу роль як при навчанні класифікатора та оптимізації його роботи, так і при аналізі результатів для розуміння взаємозв’язків між ознаками об’єктів та їх відношенням до класів.

При роботі алгоритму RFC, після того, як кожне випадкове дерево побудовано за допомогою набору функцій, обчислюється його ефективність (коефіцієнт неправильної класифікації). Потім значення кожної ознаки випадковим чином перестановляються (для кожної функції) і оцінюється новий показник неправильної класифікації. Потім вибираються найкращі характеристики як найбільш інформативні особливості. Найбільш інформативні особливості, які ми знайшли для виявлення зловмисних повідомлень за RFC, описані в таблиці 3.8.

Таблиця 3.8 – Найбільш інформативні ознаки на основі алгоритму Random Forest Classifier

|  |  |
| --- | --- |
| Rank | Feature |
| 1 | Account\_age |
| 2 | Following\_Follower\_Ratio |
| 3 | No\_following |
| 4 | No\_follower |
| 5 | No\_tweets |
| 6 | No\_hashtags |

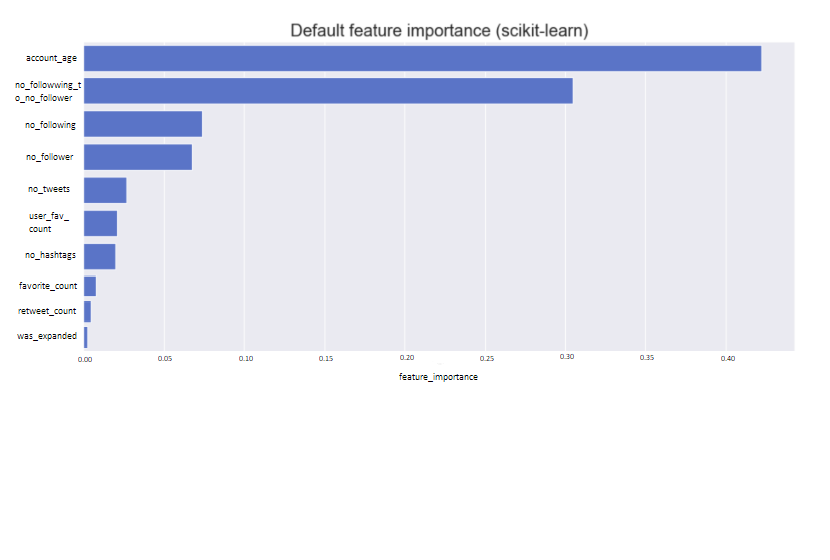


Рисунок 3.5 – Результат алгоритму RFC по визначенню найважливіших ознак даних

Одним з важливіших ознак виявилась час життя акаунту, тобто різниця між часом створення акаунту та останнім твітом користувача. Зазвичай зловмисники створюють велику кількість акаунтів і використовують їх для надсилання великої кількості повідомлень зі спамом або фішингом за короткий роміжок часу, до тих пір поки акаунт не заблокують. Таким чином, ця ознака сильно впливає на класифікатор при віявленні зловмисних повідомлень.

Також цікавим виявилось те, що зловмисники часто мають більшу кількість followings ніж followers. Оскільки взаємозв’язки між користувачами в Twitter є односпрямованими, тобто люди можуть слідкувати за вашим акаунтом, проте ви можете не слідкувати за ними і навпаки, коли ви слідкуєте за сторінкою це не означає, що ця людина слідкує за вашою. Так як спамери та фішери зазвичай публікують зловмисний контент та посилання, дуже мало користувачів, як правило, слідкують за такими акаунтами. Проте кількість акаунтів за якими слідкують зловмисники набагато більша. Таким чином звідси випливає співвідношення Followings до Followers.

Також ми виявили, що велика кількість хештегів (#hashtag) в повідомленнях може свідчити про наявність спаму або фішингу в повідомленнях. Це зумовлено тим, що зловмисники «викрадають» трендові теми та назви для використання в своїх повідомленнях, так як це збільшує видимість повідомлення з новітніми темами, які завжди відображаються раніше на домашній сторінці користувача Twitter.

## Висновки до розділу 3

В третьому розділі описано етапи реалізації програмного рішення поставлених задач. Першим етапом було створення набору даних з використанням Twitter API. Наступним кроком після отримання даних є маркування їх за допомогою VirusTotal API. За період трьох місяців було зібрано 182400 екземплярів даних з яких більше ніж один відсоток повідомлення, які містять зловмисні посилання. Далі, після побудови datast з ground truth було виконано обробку даних, розширено скорочені URL-посилання та виділено ознаки об’єктів за якими буде класифікувати система виявлення.

До отриманого набору даних було застосовано чотири алгоритми класифікації. Результати їх роботи було відображено в матрицях помилок і для обчислення якості роботи алгоритмів обраховано метрики машинного навчання: Precision, Recall та F1-score. За цими результатами ми визначили, що для системи виявлення зловмисних повідомлень більш ефективним алгоритмом є Random Forest Classifier.

На основі алгоритму RFC, що має найкращі результати класифікації серед трьох інших, було виявлено найбільш важливі характеристики або ознаки даних за якими зловмисні повідомлення виділяються серед повідомлень звичайних користувачів.

# 4 РОЗРОБКА СТАРТАП ПРОЕКТУ

В даному розділі проведено маркетинговий аналіз стартап-проекту, який показав можливості ринкового впровадження ідеї стартапу та можливих напрямів реалізації проекту. Метою даного старатп-проекту є розширення для соціальної мережі Twitter, яке класифікує повідомлення в Twitter на основі змісту повідомлення та присутнього в ньому URL-посилання та попереджає користувача про можливу небезпеку якщо перейти по даному посиланню. За результатами проведеного аналізу описано доцільність впровадження проекту та кроки для виходу на ринок.

## 4.1 Опис ідеї проекту

Першим кроком маркетингового аналізу проекту є опис ідеї. Необхідно визначитись зі змістом ідеї, напрямками застосування продукту, цільова аудиторія стартапу та можливі вигоди для користувачів за кожним напрямком застосування. Також потрібно провести аналіз технічних та економічних переваг у порівнянні з продуктами конкурентів для визначення сильних та слабких сторін проекту за рахунок чого оцінити конкурентоспроможність.

В табл. 4.1 представлено зміст ідеї проекту, напрямки її застосування та можливі вигоди користувачів.

Таблиця 4.1 – Опис ідеї стартап-проекту

|  |  |  |
| --- | --- | --- |
| *Зміст ідеї* | *Напрямки застосування* | *Вигоди для користувача* |
| Розробка розширення для соціальної мережі Twitter, яке класифікує повідомлення в реальному часі на основі присутніх в них URL-посилань. | 1. Використання окремими користувачами | Забезпечення безпечного спілкування в мережі та запобігання переходу по небажаним адресам. |

Після проведеного аналізу ринку, маємо таких конкурентів: PhishAri, Twitter Follower Blocker.

Для оцінки техніко-економічних характеристики було обрано такі характеристики ідеї: економічні, кросплатформеність, можливість віддавленого керування технічні, потребують модифікації, надійність. В таблиці наведено оцінку техніко-економічних характеристик відносно продукту та його конкурентів.

Таблиця 4.2 – Визначення техніко-економічних характеристик проекту та конкурентів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | *Техніко-економічні характеристики ідеї* | *(потенційні) товари/ концепції конкурентів* | | |
| *Мій проект* | *Конкурент 1* | *Конкурент 2* |
| 1 | Економічні | Витрати на утримання офісу та співробітників. | Витрати на утримання офісу та співробітників. | Витрати на утримання офісу та співробітників. |
| 2 | Кросплатформенність | Так | | |
| 3 | Можливість віддаленого керування | Так | Так | Так |
| 4 | Потребують модифікацій | Так | Ні | Так |
| 5 | Потребування глибоких знань | Так | Так | Так |

В таблиці 4.3 представлено сильні та слабкі сторони які залежать від визначення оцінки запропонованої ідеї та обох конкурентів продукту.

Таблиця 4.3 – Визначення сильних, слабких та нейтральних характеристик проекту

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | *Техніко-економічні характеристики ідеї* | *W (слабка сторона)* | *N (нейтральна сторона)* | *S (сильна сторона)* |
| 1 | Економічні |  | + |  |
| 2 | Кросплатформенність |  |  | + |
| 3 | Можливість віддаленого керування |  |  | + |
| 4 | Потребують модифікацій | + |  |  |
| 5 | Потребування глибоких знань | + |  |  |

## 4.2 Технологічний аудит ідеї проекту

В даному підрозділі будемо розглядати технології які необхідні для реалізації проекту. В результаті аналізу було виявлено, що усі запропоновані технології є доступними та вільними у користуванні. Проте безкоштовні версії VirusTotal API та Twitter API мають свої обмеження, що створюють деякі складності, проте на етапі тестування та проектування цього достатньо. Проте з отриманням більших можливостей з боку даних сервісів можна отримати більш точні та додаткові результати.

В табл. 4.4 показано необхідні кроки при реалізації проекту та запропоновані технології для їх вирішення.

Таблиця 4.4 – Технологічна здійсненність ідеї проекту

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | *Ідея проекту* | *Технології її реалізації* | *Наявність технологій* | *Доступність технологій* |
| 1 | Збір даних | Twitter API та Python | Наявна | Доступна |
| 2 | Маркування даних | VirusTotal API та Python | Наявна | Доступна |
| 3 | Обробка даних та побудова системи виявлення | Мова програмування Python та вбудовані бібліотеки | Наявна | Доступна |
| 4 | Аналіз даних | Мова програмування Python та вбудовані бібліотеки | Наявна | Доступна |

## 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Першим кроком визначення ринкових можливостей є аналіз та попередня характеристика ринку стартап проекту, що зображено в Таблиці 4.5

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проекту

|  |  |  |
| --- | --- | --- |
| № п/п | *Показники стану ринку (найменування)* | *Характеристика* |
| 1 | Кількість головних гравців, од. | 3 |
| 2 | Загальний обсяг продаж, грн/ум.од | 200 тис. ум.од. |
| 3 | Динаміка ринку (якісна оцінка) | Зростає |
| 4 | Наявність обмежень для входу (вказати характер обмежень) | Немає, необхідно зацікавити спільноту продуктом |
| 5 | Специфічні вимоги до стандартизації та сертифікації | Немає |
| 6 | Середня норма рентабельності в галузі (або по ринку), % | 75% |

Також необхідно розуміти цільову групу продукту і потреби які буде задовольняти продукт. Таким чином в таблиці 4.6 розглядаються характеристики потенційних клієнтів стартап-проекту.

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | *Потреба, що формує ринок* | *Цільова аудиторія (цільові сегменти ринку)* | *Відмінності у поведінці різних потенційних цільових груп клієнтів* | *Вимоги споживачів до товару* |
| 1. | Безпечне користування соціальними мережами та зменшення ризику бути жертвами фішингу, спаму або завантаженню зловмисного ПЗ через URL-посилання з повідомлень Twitter. | Користувачі соціальної мережі Twitter | – | Зручність використання, простий функціонал, практичність, надійність, не заважати основному контексту повідомлень |

Після визначення цільової аудиторії проекту необхідно проаналізувати можливі фактори загроз, що можуть перешкоджати розвитку проекту, та фактори можливостей, що сприяють ринковому впровадженню (Табл. 4.7 та 4.8).

Таблиця 4.7 – Фактори загроз

|  |  |  |  |
| --- | --- | --- | --- |
| № | *Фактор* | *Зміст загрози* | *Можлива реакція компанії* |
| 1. | Відсутність репутації | Немає статусу компанії, невідома якість продукту | Розробка документації та підготовка маркетингових рішень |

Таблиця 4.8 – Фактори можливостей

|  |  |  |  |
| --- | --- | --- | --- |
| № | *Фактор* | *Зміст можливості* | *Можлива реакція компанії* |
| 1. | Ефективність системи виявлення зловмисних повідомлень призведе до більш широкого розвитку продукту | Ефективність роботи продукту призведе до використання його не лише як розширення, а й стане доповненням та системою захисту користувачів в мережі Twitter або можливість застосування схожого підходу в інших соціальних мережах | Інвестиції які допоможуть розвитку продукту та виходу на широкий ринок |

Таблиця 4.9 - Ступеневий аналіз конкуренції на ринку

|  |  |  |
| --- | --- | --- |
| *Особливості конкурентного середовища* | *В чому проявляється дана характеристика* | *Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)* |
| 1. Конкуренція – чиста | Так як на даному ринку немає великих корпорацій та фаворитів, та ринок є досі незаповненим, типом конкуренції буде чиста | Зацікавлення аудиторії новими можливостями продукту |
| 2. Рівень конкурентної боротьби – світовий | Не залежить від розташування, користуються всі | Важлива присутність локалізаційних версій |
| 3. За галузевою ознакою  - міжгалузева | Не стосується певної галузі |  |
| 4. Конкуренція за видами товарів - товарно-видова | Тільки подібні товари є конкурентами |  |
| 5. За характером конкурентних переваг - нецінова | Ціна - не є засобом привабливості для клієнтів |  |
| 6. За інтенсивністю - не марочна | Ціни відрізняються для різних пакетів послуг | Виділення унікальних цін |

Після аналізу конкуренції проведено детальний аналіз умов конкуренції в галузі – за моделлю 5 сил М. Портера (таб. 4.10).

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Складові аналізу* | | *Прямі конкуренти в галузі* | | *Потенційні конкуренти* | | *Клієнти* | | *Товари-замінники* | |
| PhishAri, Twitter Follower Blocker | | Невідомий раніше продукт, може викликати незацікавленість | | За рахунок зворотнього звязку, відгуків, можна впливати на розвиток продукту та його вдосконалень | | Впровадження схожих систем | |
| Висновки: | Конкурентна боротьба не є дуже високою | | Строки виходу на ринок – півроку;  Конкуренти є, але даний продукт має свої переваги | | Так, клієнт впливає на розвиток та вдосконалення продукту | | Обмежень немає | |

З огляду на конкурентну ситуацію, з готовим продуктом можна успішно вийти на ринок за півроку після початку розробки.

На основі аналізу конкуренції, проведеного в Таблиці 4.10, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.6) та факторів маркетингового середовища (табл. 4.7-4.8) визначається та обґрунтовується перелік факторів конкурентоспроможності (табл. 4.11). [31]

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

|  |  |  |
| --- | --- | --- |
| № | *Фактор конкурентоспроможності* | *Обґрунтування (чинники, що роблять фактор значущим)* |
| 1 | Репутаційний | Значний вплив імені компанії, що надає послугу. Тобто, чи відома вона або її продукт. |
| 2 | Технологічний | Застосування нових підходів та ефективності їх роботи має великий вплив на розвиток продукту. Також зручність використання та точність роботи системи приваблює користувачів. |

За визначеними факторами конкурентоспроможності (табл. 4.11) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 4.12).

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін проекту

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | *Фактор конкурентоспроможності* | *Бали 1-20* | *Рейтинг товарів-конкурентів (в порівнянні)* | | | | | | |
| *-3* | *-2* | *-1* | *0* | *+1* | *+2* | *+3* |
| 1 | Ціновий | 15 |  |  |  |  | 1 | 1 | 3 |
| 2 | Репутаційний | 12 |  |  |  |  | 3 | 2 |  |
| 3 | Технологічний | 18 |  |  | 1 | 4 |  |  |  |

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.13). [31]

Таблиця 4.13 – SWOT-аналіз стартап-проекту

|  |  |
| --- | --- |
| **Сильні сторони:**  Технологічна новизна та простота взаємодії з продуктом | **Слабкі сторони:**  Відсутність хорошої репутації, маловідомість |
| **Можливості:**  Ефективність системи виявлення зловмисних повідомлень призведе до більш широкого розвитку продукту | **Загрози:**  Незацікавленість продуктом |

Після отримання результатів SWOT-аналізу розроблено альтернативи ринкової поведінки при виведенні стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 4.14). [31]

Таблиця 4.14 – Альтернативи ринкового впровадження стартап-проекту

|  |  |  |  |
| --- | --- | --- | --- |
| № | *Альтернатива (орієнтовний комплекс заходів) ринкової поведінки* | *Ймовірність отримання ресурсів* | *Строки реалізації* |
| 1 | Участь в стартап-акселераторах та більший зв’язок зі спільнотою заради подальшого маркетингу | 25% | Півроку |
| 2 | Прямий пошук початкових інвестицій | Досить імовірно | 3 місяці |
| 3 | Вихід з мінімальними початковими вкладеннями | 100% | Півроку |

## 4.4 Розроблення ринкової стратегії проекту

Після аналізу ринкових можливостей та конкурентоспроможності продукту необхідно розробити стратегію виходу продукту на ринок. В таблиці 4.15 виявлено цільові групи потенційних споживачів даного стартап-проекту.

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | *Опис профілю цільової групи потенційних клієнтів* | *Готовність споживачів сприйняти продукт* | *Орієнтовний попит в межах цільової групи* | *Інтенсивність конкуренції в сегменті* | *Простота входу у сегмент* |
|
| 1 | Користувачі мережі Twitter | Так | Середній | Чиста | Середньо |

В таблиці 4.16 та 4.17 визначаються базові стратегії розвитку та конкурентної поведінки продукту результати яких також впливають на можливості та складності при виході на ринок продукту.

Таблиця 4.16 – Визначення базової стратегії розвитку

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | *Обрана альтернатива розвитку* | *Стратегія охоплення ринку* | *Ключові конкурентноспроможні позиції відповідно до обраної альтернативи* | *Базова стратегія розвитку* |
| 1 | Реалізація самостійного програмного засобу | Стратегія диференційованого маркетингу | Незалежність від технічних рішень Twitter | Стратегія диференціації |
| 2 | Розробка інтеграції з соціальною мережею Twitter | Стратегія диференційованого маркетингу | Зручність у використанні | Стратегія спеціалізації |

Таблиця 4.17 – Визначення базової стратегії конкурентної поведінки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | *Чи є проект «першопрохідцем» на ринку?* | *Компанія шукатиме нових споживачів, або забиратиме існуючих у конкурентів?* | *Чи буде компанія*  *копіювати основні*  *характеристики товару конкурентів* | *Стратегія конкурентної поведінки* |
| 1 | Ні | Буде зацікавлювати нових споживачів | Ні, не копіюватиме. Застосовані технології та підхід відрізняється від конкурентів | Стратегія диференціації |

Таблиця 4.18 – Визначення стратегії позиціонування

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | *Вимоги до товару цільової аудиторії* | *Базова стратегія розвитку* | *Ключові конкурентноспроможні позиції власного стартап-проекту* | *Вибір асоціацій, які мають сформувати комплексну позицію власного проекту* |
| 1 | Виявлення небажаних (зловмисних) повідомлень та детектування їх видів | Стратегія диференціації | Застосування новго підходу з використанням новітніх технологій та висока ефективність при виявленні повідомлень | Безпечне спілкування мережею Twitter; Виявлення зловмисних повідомлень;  Розширення для мережі Twitter |

## 4.5 Розроблення маркетингової програми стратап-проекту

Важливою частиною запуску стартап-проекту є розробка маркетингової програми.

Першим кроком є формування *маркетингової концепції товару*, який отримає споживач. Для цього у табл. 4.19 підсумовуються результати попереднього аналізу конкурентоспроможності товару. [31]

Таблиця 4.19 – Визначення ключових переваг концепції потенційного товару

|  |  |  |  |
| --- | --- | --- | --- |
| № | *Потреба* | *Вигода, яку пропонує товар* | *Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити* |
| 1 | Безпечне спілкування в соціальній мережі Twitter | Виявлення зловмисних повідомлень додатково до систем виявлення Twitter | Виявлення різних видів загроз |

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.20). [31]

Таблиця 4.20 будується для того, щоб виявити реальний етап розробки продукту та висвітлюються особливості які мають бути освітлені в ході побудови маркетингових рішень.

Таблиця 4.20 – Опис трьох рівнів моделі товару

|  |  |  |  |
| --- | --- | --- | --- |
| **Рівні товару** | **Сутність та складові** | | |
| **І. Товар за задумом** | Забезпечення виявлення зловмисних повідомлень в мережі Twitter | | |
| **ІІ. Товар у реальному виконанні** | **Властивості/характеристики** | **М/Нм** | **Вр/Тх /Тл/Е/Ор** |
| 1. Система виявляє зловмисні повідомлення та поділяє їх на різні класи загроз | М | Тх, Е |
| 1. Оцінка точності системи виявлення | М | Тх, Е |
| 1. Інтеграція з мережею Twitter | Нм | Тх, Е |
| 1. Робота на усіх платформах | Нм | Тх, Тл, Е |
| Якість: якість оцінюється метриками машинного навчання | | |
| **ІІІ. Товар із підкріпленням** | До продажу: запуск бета версій для тестування та подальшого покращення продукту | | |
| Після продажу: підтримка ПЗ та виявлення і вирішення недоліків | | |
| **За рахунок чого потенційний товар буде захищено від копіювання:** | Захист інтелектуальної власності | | |

Таблиця 4.21 – Визначення меж встановлення ціни

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | *Рівень цін на товари-замінники* | *Рівень цін на товари-аналоги* | *Рівень доходів цільової групи споживачів* | *Верхня та нижня межі встановлення ціни на товар/послугу* |
|  | Відсутній | Відсутній | Будь-який | Для усіх користувачів ціна може становити від 1-5 $ на місяць |

Таблиця 4.22 – Формування системи збуту

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | *Специфіка закупівельної*  *поведінки цільових клієнтів* | *Функції збуту, які має виконувати постачальник товару* | *Глибина каналу*  *збуту* | *Оптимальна*  *система збуту* |
| 1 | Пошук засобів захисту від зловмисних повідомлень в Twitter | Виконувати основну ціль та оновлювати ПЗ | Розширення та додатки в браузерах | Продаж через магазини додатків |

Таблиця 4.23 – Концепція маркетингових комунікацій

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | *Специфіка поведінки цільових клієнтів* | *Канали комунікацій, якими*  *користуються*  *цільові клієнти* | *Ключові позиції, обрані для позиціонування* | *Завдання рекламного повідомлення* | *Концепція рекламного звернення* |
| 1 | Тестування продукту | Соціальні мережі | Виявлення зловмисних повідомлень в мережі Twitter | Вказати на небезпеку на яку може потрапити користувач | Демонстрація зловмисних повідомлень і наслідки які стоять за ними; статистика розповсюдження спаму, фішингу та зловмисного ПЗ |

## Висновки до розділу 4

В даному розділі проаналізовано можливі перспективи виходу запропонованого продукту на ринок із огляду на маркетингові особливості, конкурентоспроможність, бар’єри входу на ринок та ін.

Запропоноване рішення маж перспективи впровадження з огляду на актуальність ідеї та потребу потенційних клієнтів, тобто користувачів соціальної мережі Twitter.

Для впровадження даного рішення було обрано розробити розширення для браузера, що є оптимальною можливістю з точки зору техніко-економічних можливостей, та допоможе виявити реальну потребу в даному продукті. В перспективі, при ефективності роботи застосунку та його доцільності, розробити окрему незалежну систему.

# ВИСНОВКИ

Результатом виконання дипломної роботи є модель виявлення зловмисних повідомлень в мережі Twitter та її програмна реалізація.

В процесі роботи було проаналізовано існуючі підходи при вирішенні подібних задач та на основі цього запропоновано удосконалену модель виявлення зловмисних повідомлень. Побудовано власний набір даних, який складається з повідомлень Twitter, які містять у собі URL-посилання. Маркування даних виконувалось за результатом аналізу URL-посилань із використанням VirusTotal API.

Після побудови та обробки набору даних було виявлено ознаки за якими будувались класифікатори. Для виявлення ефективного алгоритму класифікації було порівняно 4 моделі: K-nearest neighbors, Support Vector Machine, Naïve Bayes та Random Forest Classifier. В результаті чого, за метриками оцінки якості роботи алгоритмів, було обрано Random Forest Classifier як найбільш ефективний при вирішенні даної задачі.

На основі алгоритму RFC, що має найкращі результати класифікації серед трьох інших, було виявлено найбільш важливі характеристики або ознаки даних за якими зловмисні повідомлення виділяються серед повідомлень звичайних користувачів.

Запропоновано ідею для стартап-проекту, метою якої є розробка розширення для соціальної мережі Twitter, яке класифікує повідомлення в реальному часі на основі присутніх в них URL-посилань. Проаналізовано техніко-економічні та маркетингові можливості для запуску даного продукту на ринок.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Cheng Cao, James Caverlee. Detecting Spam URLs in Social Media via Behavioral Analysis. ECIR 2015: Advances in Information Retrieval, pp 703-714, 2015
2. D. Wang and C. Pu. BEAN: A BEhavior ANalysis Approach of URL Spam Filtering in Twitter. IEEE International Conference on Information Reuse and Integration, 2015.
3. Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang and Yihong (Eric) Zhao. Online Spam Detection in Blogs: A Behavior-based Approach, 2010.
4. Lung-Hao Lee, Kuei-Ching Lee,Yen-Cheng Juan, Hsin-Hsi Chen, Yuen-Hsien Tseng. Users’ Behavioral Prediction for Phishing Detection. WWW '14 Companion Proceedings of the 23rd International Conference on World Wide Web, pp 337-338, 2014
5. Alex Hai Wang. Don't follow me: Spam detection in Twitter. International Conference on Security and Cryptography (SECRYPT), 2010
6. Kyumin Lee, James Caverlee, Steve Webb. Uncovering Social Spammers: Social Honeypots + Machine Learning. SIGIR, 2010
7. Jonghyuk, SongSangho, LeeJong Kim. Spam filtering in twitter using sender-receiver relationship. Recent Advances in Intrusion Detection pp 301-317, 2011
8. Juan Martinez-Romo, Lourdes Araujo. Detecting malicious tweets in trending topics using a statistical analysis of language.  Expert Systems with Applications, 2013
9. Despoina Antonakaki, Iasonas Polakis, Elias Athanasopoulos, Sotiris Ioannidis, Paraskevi Fragopoulou. Exploiting abused trending topics to identify spam campaigns in twitter. Social Network Analysis and Mining, 2016
10. Chris Grier, Kurt Thomas, Vern Paxson, Michael Zhang. @spam: The Underground on 140 Characters or Less. Proceedings of the 17th ACM Conference on Computer and Communications Security, 2010
11. K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, 2011
12. W. Wei, K. Joseph, H. Liu, and K. M. Carley. Exploring characteristics of suspended users and network stability on twitter. Social Network Analysis and Mining, 2016
13. H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, 2010
14. Офіційна сторінка допомоги при роботі з Twitter API [Електронний ресурс] – Режим доступу: <https://help.twitter.com/ru/rules-and-policies/twitter-api>
15. Документація по роботі з Python-бібліотекою Tweepy [Електронний ресурс] – Режим доступу: <https://tweepy.readthedocs.io/>
16. Эндрю Гласснер. Глубокое обучение без математики. Том 1. Основы. М.: ДМК Пресс, 2019. – с. 27
17. Офіційна сторінка прикладного програмного інтерфейсу VirusTotal [Електронний ресурс] – Режим доступу: <https://developers.virustotal.com/reference>
18. Можливості роботи прикладного програмного інтерфейсу VirusTotal [Електронний ресурс] – Режим доступу: <https://www.programmableweb.com/api/virustotal-private>
19. Офіційна сторінка прикладного програмного інтерфейсу VirusTotal, запит на аналіз URL-посилання [Електронний ресурс] – Режим доступу:

<https://developers.virustotal.com/reference#url-report>

1. Вьюгин В.В. Математические основы теории машинного обучения и прогнозирования – М.: – 2013. - 387 с.
2. Обучение с учителем [Електронний ресурс ] – Ресурс доступу: <http://www.machinelearning.ru/wiki/index.php?title=Обучение_с_учителем>
3. Х. Бринк, Д. Ричардс, М. Феверолф Машинное обучение СПб.: Питер – 2017. – с. 100
4. А. Мюллер, С. Гвидо Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными – М.: – 2016-2017.
5. Курс лекций по Data Mining. Задачи Data Mining. Классификация и кластеризация [Електронний ресурс] – Режим доступу: https://studfiles.net/preview/5554364/page:17/
6. Х. Бринк, Д. Ричардс, М. Феверолф Машинное обучение СПб.: Питер – 2017. – с. 113
7. Yogendra Kumar Jain, Vivek Suryawanshi. A Review on Evaluation Metrics for Data Classification Evaluations. SATI, India, 2018.
8. Generalized Confusion Matrix for Multiple Classes [Електронний ресурс} – Режим доступу: <https://www.researchgate.net/publication/310799885_Generalized_Confusion_Matrix_for_Multiple_Classes>
9. M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Compa: Detecting compromised accounts on social networks. In Annual Network and Distributed System Security Symposium, 2013
10. Х. Бринк, Д. Ричардс, М. Феверолф Машинное обучение СПб.: Питер – 2017. – с. 113
11. Naïve Bayes Classifier [Електронний ресурс] – Режим доступу: <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>
12. **Розроблення стартап-проекту** [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.

# ДОДАТКИ

# ДОДАТОК А

*#Import the necessary methods from different libraries***import** os  
**from** tweepy.streaming **import** StreamListener  
**from** tweepy **import** OAuthHandler  
**from** tweepy **import** Stream  
**import** json  
  
  
*# Variables that contains the user credentials to access Twitter API*access\_token = **"your\_access\_token"**access\_token\_secret = **"your\_access\_token\_secret"**consumer\_key = **"your\_consumer\_key"**consumer\_secret = **"your\_consumer\_secret"**count = 2800  
num = 0  
list\_urls = []  
all\_data\_urls = []  
path = os.getcwd()  
subdir = **'17.10'  
  
class** StdOutListener(StreamListener):  
 **def** on\_data(self, data):  
  
 decoded = json.loads(data)  
  
 **global** num  
 **global** count *# get the count* **if** count <= 0:  
 **import** sys  
 sys.exit()  
 **else**:  
 **try**:  
  
 **for** url **in** decoded[**"entities"**][**"urls"**]:  
 **if** decoded[**"lang"**] != **'en'**:  
 **continue  
 else**:  
 count -= 1  
 jsonf\_name = **'17.10\_tweets.json'  
 with** open(os.path.join(path, subdir, jsonf\_name), **'a'**) **as** tf:  
 num += 1  
 decoded[**'num'**] = num  
 tf.write(json.dumps(decoded) + **'\n'**)  
 txt\_name = **'17.10\_urls.txt'  
 with** open(os.path.join(path, subdir, txt\_name), **'a'**) **as** file2:  
 new\_url = {**'num'** : num, **'url'** : url[**"expanded\_url"**]}  
 file2.write(str(new\_url) + **'\n'**)  
 **except** KeyError:  
 print(decoded.keys())  
  
 **return True  
  
 def** on\_error(self, status):  
 print(status)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
  
 l = StdOutListener()  
 auth = OAuthHandler(consumer\_key, consumer\_secret)  
 auth.set\_access\_token(access\_token, access\_token\_secret)  
 stream = Stream(auth, l)  
  
 stream.sample()

# ДОДАТОК Б

**import** scrapy  
**from** scrapy.crawler **import** CrawlerProcess  
**import** re  
**from** ast **import** literal\_eval  
**import** os  
  
**class** SpiderExpandURL(scrapy.Spider):  
 name = **'expanded\_url'  
  
 def** start\_requests(self):  
 short\_urls = [**'bit.ly'**, **'bitly.com'**, **'cur.lv'**, **'goo.gl'**, **'t.co'**, **'tinyurl.com'**, **'mol.im'**]  
 url = **'http://checkshorturl.com/expand.php'** dir = **r'directory'** subdir = **'15.10'  
 with** open(os.path.join(dir, subdir, **'clean\_15.10\_urls.txt'**), **'r'**) **as** f:  
 **for** line **in** f:  
 line = literal\_eval(line)  
 **if** any(x **in** line[**'url'**] **for** x **in** short\_urls):  
 params = {**'u'**: line[**'url'**].rstrip()}  
 **yield** scrapy.FormRequest(url, callback=self.parse, method=**'POST'**, formdata=params)  
 **else**:  
 **continue  
  
 def** parse(self, response):  
 dir = **r'directory'** subdir = **'15.10'  
 with** open(os.path.join(dir, subdir, **'expanded\_clean\_15.10\_urls.txt'**), **'a'**) **as** f:  
 result = response.xpath(**'(//body/div[2]/div[4]/table/tr[1]/td[2]/a/text())[1]'**).extract()  
 short\_url = response.xpath(**'//body/div[2]/div[4]/table/tr[3]/td[2]/text()'**).extract()  
 *#print(result)* pattern = re.compile(**r'\r+'**)  
 **if** len(result) != 0:  
 res2 = []  
 sh\_url = []  
 **for** i **in** result:  
 res2.append((re.sub(pattern, **''**, i.strip())).replace(**'\n'**, **''**))  
 *#print(res2)* **for** j **in** short\_url:  
 sh\_url.append((re.sub(pattern, **''**, j.strip())).replace(**'\n'**, **''**))  
 *#print(sh\_url)* new\_line = {**'short'** : **''**.join(sh\_url), **'expanded'** : **''**.join(res2)}  
 f.write(str(new\_line) + **'\n'**)  
  
  
process = CrawlerProcess()  
process.crawl(SpiderExpandURL)  
process.start()

# ДОДАТОК С

**import** requests, json  
**import** virus\_total\_apis  
**import** time  
**import** pickle  
**import** csv  
**import** ast  
  
API\_KEY = **your\_key**  
**def** parse\_json(json\_instance):  
 answer = []  
 not\_clean = []  
 result = json\_instance[**"results"**][**"scans"**]  
  
 **for** i **in** result:  
 *#print(i)* **if** (result[i][**'detected'**] == **True**):  
 not\_clean.append(result[i][**'result'**])  
 **if not** not\_clean:  
 answer.append(**'clean site'**)  
 **else**:  
 answer = not\_clean  
 **return** answer  
  
  
**def** online\_scan(api\_key, target\_url):  
 result = []  
 switch\_api\_key = virus\_total\_apis.PublicApi(api\_key)  
  
 response = switch\_api\_key.get\_url\_report(target\_url)  
 raw\_data\_json = json.dumps(response, sort\_keys=**True**, indent=4)  
 raw\_json\_dict = json.loads(raw\_data\_json)  
  
 **if** raw\_json\_dict[**"results"**][**"verbose\_msg"**] != **'Scan finished, scan information embedded in this object'**:  
 print(raw\_json\_dict[**"results"**][**"verbose\_msg"**])  
 result.append(raw\_json\_dict[**"results"**][**"verbose\_msg"**])  
 **else**:  
 result = parse\_json(raw\_json\_dict)  
  
 **return** result  
  
**def** add\_label\_to\_json(file\_name, api\_key):  
 data = create\_list\_of\_json(file\_name)  
 new\_file = open(**'SCANNED\_'**+file\_name.split(**'/'**)[2], **'a'**)  
 count = 0  
 num = 0  
 **for** i **in** data:  
 **if** i[**'was\_expanded'**] == 1:  
 num+=1  
 *#print(num)* print(**"expanded url : "** + str(num))  
 **if** count == 4:  
 time.sleep(69)  
 count = 0  
 scan\_res = online\_scan(api\_key, i[**'long\_url'**])  
 count+=1  
 i[**'label'**] = scan\_res  
 json\_string = json.dumps(i)  
 new\_file.write(json\_string)  
 new\_file.write(**'\n'**)  
 **elif** i[**'was\_expanded'**] == 0:   
 **for j in i['entities']['urls']:  
 if count == 4:  
 time.sleep(60)  
 count = 0  
 scan\_res2 = online\_scan(api\_key, j['expanded\_url'])  
 count+=1  
 i['label'] = scan\_res2  
 json\_string = json.dumps(i)  
 new\_file.write(json\_string)  
 new\_file.write('\n')  
  
 return** 0

# ДОДАТОК Д

**import** json  
**import** pandas **as** pd  
**from** dateutil **import** parser  
  
  
**def** account\_age(json\_inst):  
 tweet\_created\_at = json\_inst[**'created\_at'**]  
 account\_age = json\_inst[**'user'**][**'created\_at'**]  
  
 parsed\_date\_account = parser.parse(account\_age)  
 parsed\_tweet\_age = parser.parse(tweet\_created\_at)  
  
 day = str(parsed\_tweet\_age - parsed\_date\_account)  
 age = day.split(**' '**)[0]  
  
 **return** age  
  
  
**def** features\_extraction(json\_file):  
 names = [**'id'**,**'account\_age'**, **'followers\_count'**, **'friends\_count'**, **'user\_fav\_count'**, **'tweets\_count'**,  
 **'retweet\_count'**, **'favorite\_count'**, **'hashtags\_count'**, **'was\_expanded'**]  
  
  
 results = []  
 **with** open(json\_file, **'r'**) **as** fin:  
 **for** i **in** fin:  
 data = json.loads(i)  
 val = (data[**'num'**],account\_age(data), data[**'user'**][**'followers\_count'**], data[**'user'**][**'friends\_count'**],  
 data[**'user'**][**'favourites\_count'**], data[**'user'**][**'statuses\_count'**], data[**'retweet\_count'**],  
 data[**'favorite\_count'**], len(list(data[**'entities'**][**'hashtags'**])), data[**'was\_expanded'**])  
 results.append(val)df = pd.DataFrame(results, columns=names)  
 df.to\_csv(json\_file.split(**'/'**)[0] + **'/'** + json\_file.split(**'/'**)[1] + **'/'** + **'dataset1.csv'**, sep=**';'**,header=**True**, index=**False**)  
  
 **return** 0