Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: « **АНАЛІЗ ІНФОРМАЦІЇ ПРО СТАН ФІНАНСОВИХ РИНКІВ ЗА ДОПОМОГОЮ АЛГОРИТМІВ ОБРОБКИ ПРИРОДНОЇ МОВИ»**

Виконав: студент 4-го року навчання
освітньої програми «Прикладна математика»,
спеціальності 113 Прикладна математика
Мальцев Ілля Володимирович

Керівник: Олецький О. В. ,
кандидат тех. наук, доцент

Рецензент_____
(прізвище та ініціали)

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК _____

«____» _____ 20____ р.

Київ – 2021

ЗМІСТ

ВСТУП

*«Інновації завжди виникають на **перетині** ідей, дисциплін та культур»*

Франс Йохансон «Ефект Медичі»

Швидкий розвиток сучасних алгоритмів обробки природної мови (англ. Natural-language processing, NLP) зумовлює розширення меж їх застосування. А прагнення людей навчитися прогнозувати поведінку позиції цінних паперів на фондовому ринку буде актуальним, поки буде актуальною сучасна економічна система в світі. Найбільш точні моделі прогнозування дозволяють істотно зменшити ризики, що в свою чергу дає можливість максимізувати прибуток від інвестування у відповідні фінансові активи. Кожного дня у світі публікуються безліч текстів: новини на загальну тематику, спеціалізовані економічні збірники, пресрелізи, державні звіти компаній-учасників фондового ринку та інші. Навіть найдосвідченіші гравці на біржі фізично не можуть самостійно обробити таку кількість інформації. В силу цього виникає вагоме поле для досліджень з точки зору побудови систем для автоматизованої обробки текстів для подальшого прогнозування позицій відповідних активів.

Метою роботи є оглянути сучасні практики застосування обробки природної мови, реалізувати та протестувати декілька програм на різних наборах текстових даних на можливість класифікації для прогнозування у предметній області фондових ринків в межах запропонованих моделей машинного навчання.

Мета роботи зумовила наступне наукове завдання:

1. Здійснити огляд основних понять, методів та алгоритмів обробки природної мови.

2. Запропонувати методи машинного навчання, що будуть застосовані в дослідженні, для задачі класифікації.
3. Дослідити предметну область та її особливості.
4. Підібрати та описати природу наборів даних.
5. Розробити методику процесу прогнозування на основі текстів. Застосувати відповідні алгоритми для побудови моделей взявши за основу обрані набори даних.
6. Провести аналіз результатів та порівняльний аналіз застосування запропонованих алгоритмів на відповідних наборах даних.
7. Порівняти результати з результатами інших наукових робіт та робіт у відкритому доступі.

Робота складається з трьох розділів.

У першому розділі розглядаються основні поняття обробки природної мови: загальні поняття, попередня обробка тексту, векторизація документів та інші. Також, запропоновано кілька методів машинного навчання для задачі класифікації.

Другий розділ присвячений предметній області: описані важливі аспекти роботи фондового ринку, описана природа даних та розглянута специфічність підготовки даних до тренування та навчання моделей машинного навчання.

У третьому розділі проведено та описано застосування та проведено аналіз отриманих результатів. Також, проведено огляд результатів та критика зовнішніх публічних робіт у цій області.

Розділ 1. ВСТУП ДО ОБРОБКИ ПРИРОДНОЇ МОВИ

1.1 Визначення

Поняття обробки природної мови є дуже обширним, постійно розвивається та змінюється, тому дати точне визначення є вкрай складною задачею. Проте, вказана цитата дуже влучно окреслює границі визначення, описуючи основні аспекти та об’єднуючі поняття різних визначень.

«Обробка природної мови *(NLP, Natural Language Processing)* - це теоретично вмотивований комплекс обчислювальних методів для аналізу та представлення текстів, що зустрічаються у природі, на одному або декількох рівнях лінгвістичного аналізу з метою досягнення подібної до людини обробки мови для цілого ряду завдань або додатків.» [1]

Виходячи з вказаного визначення випливає, що NLP - це не один обчислювальний метод, а саме набір теоретично підкріплених методів, кожен з яких окремо або у комплексі, застосовують для розв’язання відповідних задач.

Також варто розтлумачити поняття *«текстів, що зустрічаються у природі»*. Загалом, це формулювання дозволяє охопити безліч варіантів форм текстів, що може створити людина у природі. Для прикладу, це може бути: стаття, пісня, книжка, коментарі у соціальній мережі та навіть переведений у текст діалог між людьми та аудіозапис.

«Лінгвістичний аналіз тексту передбачає виявити систему мовленнєвих естетичних функцій усіх мовних одиниць, усіх категорій (лексичних, фразеологічних, фонетико-орфоепічних, морфемних, словотворчих, частиномовних, синтаксичних), які беруть участь у створенні системи художніх образів тексту.» [2]. Все це дозволяє людині правильно перевести форму тексту у розуміння його суті. А методи NLP, в свою чергу, застосовують один або кілька рівнів лінгвістичного аналізу для розв’язання відповідних задач.

NLP вважається підрозділом штучного інтелекту *(AI, Artificial intelligence)* оскільки загальною метою є обробити текст подібно до того, як це робить людина, тобто якимось чином відтворити людську поведінку при обробці тексту.

Задач для застосування є безліч. Саме тому системи інформаційного пошуку *(IR, Informational Retrieval)*, машинний переклад *(MT, Machine Translation)*, питально-відповідальні системи *(QA, Question Answering)* застосовують відповідні методи NLP.

1.2 Базові поняття

Корпус (*Corpus*) – великий, структурований набір даних представлених у текстовому форматі. Приклад: всі випуски журналу Forbes за всі роки.

Документ (*Document*) – одиниця даних у корпусі, є безпосередньо текстом. Великий набір документів є корпусом. Приклад: один лист у великій базі електронної пошти.

Термін (*Term*) – умовна одиниця у документі. Зазвичай терміном називають слово у документі.

Словник (*Dictionary*) – набір термінів, що зустрічаються у документі або корпусі. Відповідно є словник кожного документу і словник всього корпусу.

1.3 Загальні методи попередньої обробки тексту

Перед початком обробки тексту відповідними алгоритмами, його потрібно підготувати для того, щоб в наступних кроках можна було представити одиницю тексту у векторному або матричному форматі. Для цього кожен документ в корпусі потрібно розбити на терміни та скласти словник документу. А для підвищення точності варто позбутися неінформативних та об’єднати однакові за значенням терміни. Для цього застосовуються наступні кроки:

Зведення тексту до нижнього регістру. Наприклад у тексті: *«I love mathematics. Mathematics is the queen of sciences.»* слово «*mathematics»* зустрічається двічі і несуть вони одну і ту ж суть, проте при поділі на терміни утворяться два терміни. Щоб уникнути цього, весь документ зводять до нижнього регістру.

Видалення символів, що не є словами. Обробляючи документ, необхідно витягнути найважливішу інформацію з тексту, а всього іншого позбутися. В цій роботі нам не важливий порядок речень, коми, наголоси та інше, тільки слова.

Токенізація. Токенізація – це поділ тексту на логічні частини (токени). У випадку підготовки документу це поділ на терміни. Для цього можна поділити документ за пробілами (попередньо очистивши подвійні та замінивши табуляцію та інші на пробіли) або застосувати регулярний вираз.

Видалення стоп-слів. Стоп-слова такі як: «I», «am», «the», «this» зустрічатимуться майже в кожному тексті і не один раз. В розрізі великого корпусу інформативність цих слів прямує до нуля, тому їх теж варто позбутися.

Стемінг або лематизація. Для прикладу слова: «dog», «dogs», «dog’s», «dogs’» мають можливо різні значення в тексті, проте всі ці терміни можуть містити одну і ту ж інформацію про те, що текст пов'язаний з собакою. І вважати це за 4 різних терміни буде, як мінімум затратно, а як максимум це «розмиє» значущість слова «dog» у тексті. Щоб правильно визначити термін для спільних по інформації термінів є два методи: стемінг та лематизація.

Стемінг – це процес скорочення слова до основи. У випадку стемінгу в слова просто відкидається частина, що не є коренем.

Лематизація – це процес зведення слова до своєї канонічної форми. Це складний процес в якому враховуються морфологічні правила та аналіз.

Як правило, якщо важлива точність слів, то обирається лематизація. Якщо ж нам важлива швидкість обробки тексту – обирається стемінг. Приклад порівняння на рис. 1.3.1.

Original	Stemming	Lemmatization
New	New	New
York	York	York
is	is	be
the	the	the
most	most	most
densely	dens	densely
populated	popul	populated
city	citi	city
in	in	in
the	the	the
United	Unite	United
States	State	States

Рисунок 1.3.1 Приклад різниці результатів стемінгу та лематизації

Після проведення всіх кроків можна формувати словник термінів. Для наочності було обрано документ з одинадцяти тисяч термінів, та застосовано всі етапи попередньої обробки з відповідною проміжною статистикою по кількості слів в словнику на кожному з етапів використовуючи мову програмування Python [3] та додаткову програмну бібліотеку nltk [4] (див. рис. 1.3.2 та табл. 1.3.3).

```
import string

from nltk.corpus import brown
from nltk.corpus import stopwords

terms = brown.words()
print("all terms:", len(terms), "unique terms:", len(set(terms)))

print("1. To lower")
terms = [x.lower()for x in terms]
print("all terms:", len(terms), "unique terms:", len(set(terms)))

print("2. Delete punctuation")
terms = list(map(lambda x: "".join([c for c in x if c not in string.punctuation]), terms))
terms = list(filter(lambda x: x is not "", terms))
print("all terms:", len(terms), "unique terms:", len(set(terms)))

print("3. Delete stop words")
sw = stopwords.words('english')
terms = list(filter(lambda x: x not in stopwords.words('english'), terms))
print("all terms:", len(terms), "unique terms:", len(set(terms)))

print("4. Stemming")
stemmer = nltk.PorterStemmer()
terms = [stemmer.stem(x) for x in terms]
print("all terms:", len(terms), "unique terms:", len(set(terms)))
```

Рисунок 1.3.2 Код збору статистики всіх етапів попередньої обробки.

	Кількість термінів	Різниця	Кількість унікальних термінів	Різниця
Необроблений текст	1161192	0	56057	0
Зведено до нижнього регістру	1161192	0	49815	-6,242
Видалено пунктуацію	1013319	-147,873	48017	-1,798
Видалено стоп-слова	539921	-473,398	47880	-137
Застосовано стемінг	539921	0	31514	-16,366

Таблиця 1.3.3 Проміжна статистика всіх етапів попередньої обробки.

З таблиці 1.3.8 видно, що найбільший результат по зменшенню кількості термінів дало видалення стоп-слів (майже половина всіх слів), а найбільший по унікальним виявився стемінг (майже третина всіх унікальних).

1.4 Векторизація документів

Наступним кроком після попередньої обробки є переведення тексту у векторний формат. Робота з векторною формою відкриває можливість застосовувати весь математичний потенціал для досягнення цілей. У випадку нашої задачі, потрібно для кожного документу виразити вектор, який однозначно визначає його у векторному просторі. Векторною моделлю називається набір всіх векторів документів корпусу. Існує кілька методів векторизації документів, розглянемо два найпоширеніші.

1. Векторизація підрахунком термінів

Кожен документ містить терміни і можна за словником корпусу порахувати скільки разів кожен термін зустрівся в документі. Для прикладу візьмемо корпус, що складається з чотирьох документів (табл. 1.4.1).

Document id	**Document text**
1	*‘This is the first document.’*
2	*‘This document is the second document.’*
3	*‘And this is the third one.’*
4	*‘Is this the first document?’*

Таблиця 1.4.1 Корпус з чотирьох документів

Словником корпусу будуть такі терміни (табл. 1.4.2)

Term id	**Term text**
1	*and*
2	*document*
3	*first*
4	*is*
5	*one*
6	*second*
7	*third*
8	*this*

Таблиця 1.4.2 Словник корпусу

Відповідно до словника, кожен документ можна виразити вектором:

$$d_1 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, d_2 = \begin{pmatrix} 0 \\ 2 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, d_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, d_4 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Недоліком такого підходу є те, що слова які часто зустрічатимуться в корпусі, будуть приймати великі значення у векторах документів, тим самим переважуючи при обчисленнях різних методів. Цю проблему вирішує наступний метод.

2. TF-IDF [5]

TF (Term Frequency, частота терміну) – це відношення числа входжень терміну до загальної кількості термінів документа. Визначається формулою:

$$TF = \frac{n_i}{\sum_k n_k}$$

де n_i – кількість входжень терміну в документ, а $\sum_k n_k$ – кількість слів у всьому документі.

IDF (Inverted Document Frequency, інвертована частота документу) – це інверсія частоти, з якою термін зустрічається в документах корпусу. Визначається формулою:

$$IDF = \log\left(\frac{|D|}{|d_i \ni t_i|}\right)$$

де $|D|$ – кількість документів в корпусі, а $|d_i \ni t_i|$ - кількість документів, в яких зустрічається термін t_i.

$$TFIDF = TF \cdot IDF$$

Таким чином TF-IDF визначає не тільки кількість термінів, а визначає важливість терміну у контексті документа, що є частиною корпусу.

Для корпусу з таблиці 1.4.1 документи виражатимуться наступними векторами:

$$d_1 = \begin{pmatrix} 0 \\ 0.46979139 \\ 0.58028582 \\ 0.38408524 \\ 0 \\ 0 \\ 0.38408524 \\ 0 \\ 0.38408524 \end{pmatrix}, d_2 = \begin{pmatrix} 0 \\ 0.6876236 \\ 0.28108867 \\ 0.53864762 \\ 0 \\ 0 \\ 0.28108867 \\ 0 \\ 0.28108867 \end{pmatrix},$$

$$d_3 = \begin{pmatrix} 0.51184851 \\ 0 \\ 0 \\ 0.26710379 \\ 0.51184851 \\ 0 \\ 0.26710379 \\ 0.51184851 \\ 0.26710379 \end{pmatrix}, d_4 = \begin{pmatrix} 0 \\ 0.46979139 \\ 0.58028582 \\ 0.38408524 \\ 0 \\ 0 \\ 0.38408524 \\ 0 \\ 0.38408524 \end{pmatrix}$$

Така векторна модель буде більш чутливою до рідкісних слів, та менш чутливою до часто вживаних.

1.5 N-грамні словники.

N-грам – це послідовність елементів кількістю в n [6]. У випадку обробки тексту часто використовують як послідовності слів. Ідея полягає в тому, що деякі послідовності слів мають великий вплив на інформацію закладену в тексті. Наприклад розглянемо два документи (табл. 1.5.1).

Document id	Document text
1	*‘ That is not funny.’*
2	*‘Not bad, this is funny’*

Таблиця 1.5.1 Корпус з двох документів.

У векторному вигляді ці два речення матимуть майже однаковий вигляд, проте їх значення дуже різне. Проте якщо об’єднувати по два слова в терміні формуючи словник, він набуде вигляду (табл. 1.5.2).

Term id	Term text
1	*that is*
2	*is not*
3	*not funny*
4	*not bad*
5	*bad this*
6	*this is*
7	*is funny*

Таблиця 1.5.2 Бі-грамний словник корпусу.

Даний словник називається бі-грамним, бо містить по два слова в терміні. В такій моделі словосполучення на кшталт: «not bad», «not funny» розглядаються цільними та додають інформативності моделі.

У задачах NLP найчастіше розглядають бігамні та триграмні моделі, проте можна наповнювати словник відразу багатьма варіантами.

Загальна кількість термінів (не унікальних) у документі обчислюється за формулою:

$$C = T + 1 - N$$

, де T це кількість слів у документі, а N – кількість «грам».

1.6 Зменшення розмірності

В результаті векторизації, розмірність вектору документу буде дорівнювати кількості слів в словнику. А кількість векторів буде дорівнювати кількості документів. При обробці великих об’ємів даних подальша робота з такими великими векторами є тяжкою задачею для навіть сучасних обчислювальних машин. А оскільки при тренуванні моделі машинним навчанням таких операцій з векторами буде дуже багато, це стає великою проблемою по очікуванню результатів. Розв’язати цю проблему можна застосувавши методи факторного аналізу, а саме метод головних компонент.

Метод головних компонент [7] дозволяє в наборі даних зменшити розмірність колонок до вказаної кількості, об’єднуючи найбільш залежні колонки між собою в спільні (див. рис. 1.6.1).

1	2		299	300
12245	2	...	0	625
1345	2	53	5	4251
3654	13	352	1324	245
1029	22	24	2345	6254

=>

1	2
2134	4315
756453	4253
43	2
2452	6541

Рисунок 1.6.1 Приклад зменшення розмірності.

Наприклад візьмемо тривіальний набір даних X. І потрібно зменшити кількість стовпців до 1.

$$X = \begin{pmatrix} 1 & 2.99 \\ 2 & 3.72 \\ 3 & 7.29 \\ 4 & 11.04 \\ 5 & 9.53 \end{pmatrix}$$

Розглянемо алгоритм МГК.

1. Відцентрувати дані відносно середнього.

$$X_{ci} = X_i - E(X_i)$$

Отримаємо:

$$X_c = \begin{pmatrix} -2 & -3.92 \\ -1 & -3.19 \\ 0 & 0.37 \\ 1 & 4.12 \\ 2 & 2.61 \end{pmatrix}$$

2. Знайти коваріаційну матрицю відцентрованих даних. Тобто потрібно порахувати коваріацію для кожної пари стовпців *i* та *j*.

$$cov(X_i, X_j) = E\left[(X_i - EX_i)(X_j - EX_j)\right]$$

За прикладом отримаємо:

$$cov(X) = \begin{pmatrix} 2.5 & 5.09 \\ 5.09 & 12.4 \end{pmatrix}$$

3. Знайти всі власні числа та власні вектори коваріаційної матриці.

$$\lambda_i cov(X) = \lambda_i v_i$$

Де λ_i це власне число, а v_i - ненульовий власний вектор.

Отримаємо:

$$\lambda_1 = 0.34, \qquad \lambda_2 = 14.56$$

$$v_1 = \begin{pmatrix} -0.92 \\ 0.39 \end{pmatrix},\ v_2 = \begin{pmatrix} 0.39 \\ 0.92 \end{pmatrix}$$

4. Зменшити розмірність

Відсортувавши власні вектори за спаданням значень власних чисел, отримаємо набір векторів V. Залежно від кількості компонент – n, які потрібно отримати в результаті, знаходимо підпростір V_n простору V, що міститиме перші *n* векторів.

Далі за формулою отримаємо нову матрицю зі зменшеною розмірністю.

$$X_{new} = (dot(V_n^T, X_C^T))^T$$

, де $dot(V_n^T, X_C^T)$ – це поелементне множення матриць V_n^T та X_C^T

За прикладом отримаємо зменшені дані:

$$X_{new} = \begin{pmatrix} -4.39 \\ -3.33 \\ 0.34 \\ 4.19 \\ 3.18 \end{pmatrix}$$

Приклад коду мовою Python за вказаним алгоритмом з використанням бібліотеки numpy для матричних обчислень на рис.1.6.2.

```
def compute_pca(X, n_components=2):
    X_c = X - np.mean(X,axis=0)

    covariance_matrix = np.cov(X_demeaned, rowvar=False)

    eigen_vals, eigen_vecs = np.linalg.eigh(covariance_matrix, UPLO='L')

    idx_sorted = np.argsort(eigen_vals)
    eigen_vecs_sorted = eigen_vecs[:,idx_sorted[::-1]]
    eigen_vecs_subset = eigen_vecs_sorted[:,0:n_components]

    X_reduced = np.dot(eigen_vecs_subset.T,X_demeaned.T).T

    return X_reduced
```

Рисунок 1.6.2 Приклад коду реалізації алгоритму МГК.

Звичайно, метод головних компонент зумовлює втрату інформації, проте «видаляються» стовпці з найменшою значимістю. Ці «втрати» можна порахувати за формулою квадрату відносної помилки.

$$\delta_n^2 = \frac{\lambda_{1+n} + \lambda_{2+n} + \cdots + \lambda_{k+n}}{\lambda_1 + \lambda_2 + \cdots + \lambda_k}$$

, де n – кількість компонент.

Вважається, що втрати в менше ніж 0.1 є припустимими, проте для різних цілей та задач ця цифра може бути різною.

Для задачі роботи з текстами незначні помилки припустимі, оскільки в процесі валідації алгоритмів, тобто вибору найкращого методу тренування моделей, можна проводити експерименти з зменшеними даними, а потім натренувати модель цим методом вже на повних даних.

1.7 Побудова моделі

Задачею даної роботи є побудувати модель яка буде опрацьовувати вхідні дані та видаватиме результат у вигляді нуля або одиниці, що буде означати падіння або ріст активу відповідно. Вхідними даними якої буде вектор документу. Є різні методи машинного навчання для побудови таких моделей. В машинному навчанні ця задача розглядається як задача класифікації. В цій роботі запропоновано логістичну регресію, наївний Баєсів класифікатор та рекурентну нейронну мережу LSTM.

1. Логістична регресія [8].

Логістичну регресію було обрано як базовий метод класифікації в машинному навчанні. Розглянемо його реалізацію.

Нехай задана вибірка X, що складається з векторів незалежних змінних вигляду $\{x_1, x_2, \dots, x_n\}$, $n \in N$, та Y, що складається з залежних змінних $y \in \{1, 0\}$. Моделлю логістичної регресії є модель, що задається залежністю:

$$y = \frac{1}{1 + e^{-z}}$$

$$z = w_0 + \sum_{i=1}^{n} w_i x_i$$

, де $w_1 \dots w_i$ – це ваги моделі. А задачею є знайти такі ваги при яких значення S буде найменшим.

$$S = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

, де $\hat{y}_i$ – реальні дані, а y_i – прогнозовані.

Для мінімізації втрат зміною ваг застосовується градієнтний спуск. Це відбувається циклічно доки різниця втрат після зміни ваг не досягне заданого значення точності α.

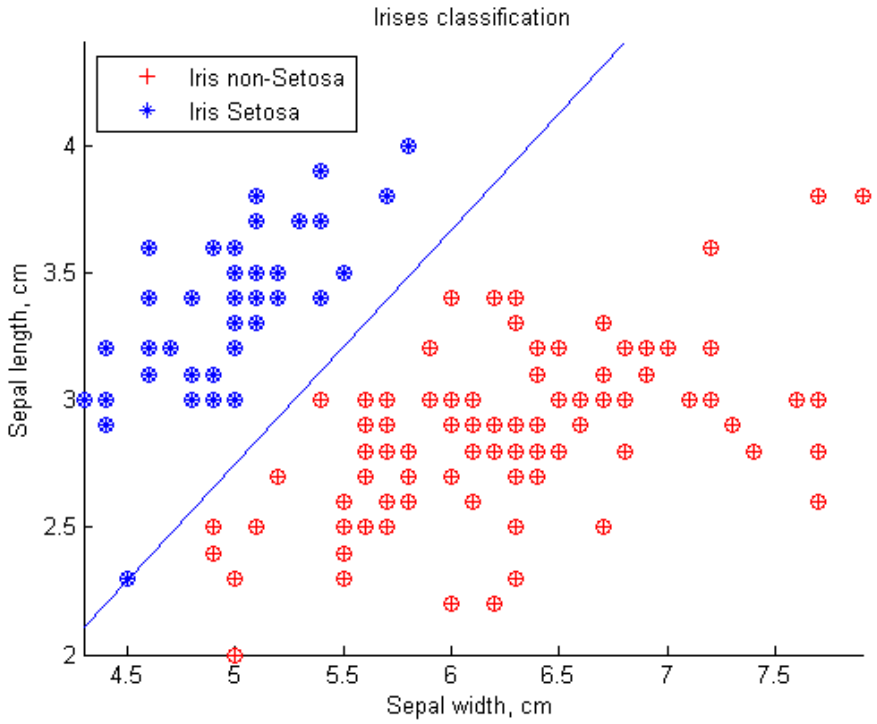

Рисунок 1.7 Візуалізація розділення даних на два класи логістичною регресією.

2. Наївний Баєсів класифікатор [9].

Розглянемо інтуїтивно зрозумілий приклад застосування класифікатора на основі векторизації документу за підрахунком входження слова. Для зручності вектори будуть розглядатися у вигляді таблиць з додатковими позначеннями.

Для початку потрібно розділити корпус на документи на відповідність до класу (нулю або одиниці). Далі просумувавши вектори документів в кожному з класів, отримаємо вектор класу.

Наприклад розглянемо корпус (табл. 1.7.1).

Class = 0	Class = 1
I am happy because I am learning NLP	I am sad, I am not learning NLP
I am happy, not sad	I am sad, not happy

Таблиця 1.7.1 Корпус розділений на класи

Тоді виконавши всі дії отримаємо (табл. 1.7.2)

	Class = 0	Class = 1
i	3	3
am	3	3
happy	2	1
because	1	0
learning	1	1
nlp	1	1
sad	1	2
not	1	2
SUMM	13	13

Таблиця 1.7.2 Словник розділений за класами та кількістю.

За формулою умовної ймовірності можна порахувати для кожного терміну ймовірність зустріти термін x_i за умови що $class = k$

$$P(x_i|class = k) = \frac{P(x_i \cap class = k)}{P(class = k)}$$

x_i	$P(x_i\|class = 0)$	$P(x_i\|class = 1)$
i	0.24	0.24
am	0.24	0.24
happy	0.15	0.08
because	0.08	0
learning	0.08	0.08
nlp	0.08	0.08
sad	0.08	0.15
not	0.08	0.15

Таблиця 1.7.3 Умовні ймовірності слів належати до класу.

За правилом Баєса, можна порахувати умовну ймовірність належності до класу за умови слів $x_1, x_2, .. x_n$.

$$P(class = k \mid x_1, x_2, .. x_n) = \frac{P(x_1, x_2, .. x_n \mid class = k)\, P(class = k)}{P(x_1, x_2, .. x_n)}$$

$$\approx \frac{\prod_{i=1}^{n} P(x_i, \mid class = k)\ P(class = k)}{\prod_{i=1}^{n} P(x_i)}$$

Формула 1.7.4

Знак приблизної рівності вказує на наївну інтерпритацію правила Баєса припускаючи, що значення незалежні.

Наприклад, ймовірність, що документ «I am happy» належить до першого класу можна порахувати за формулою

$$P(class = 1 \mid "i", "am", .. "happy") = \frac{0.24 * 0.24 * 0.15 * 0.5}{0.24 * 0.24 * 0.11} \approx 0.68$$

Проте, це не кінець, з таблиці видно, що якщо потрібно буде порахувати ймовірність відповідності до першого класу, а в документі буде слово «because», то відбудеться множення на 0 і ймовірність буде дорівнювати нулю. Таке стається коли у нас є слова які ні одного разу не зустрічалися у відповідному класі. Водночас, це речення може містити слово яке чітко вказує на цей клас, тому варто обробити такі випадки. Для того щоб позбутися нулів в умовних ймовірностях можна застосувати згладжування Лапласа.

$$P(x_i | class = k) = \frac{freq(x_i \cap class = k) + 1}{sum(class = k) + V}$$

, де V – це кількість унікальних слів (елементів у векторі), freq -це кількість, а sum – це сума.

Тоді таблиця набуде вигляду.

x_i	$P(x_i \mid class = 0)$	$P(x_i \mid class = 1)$
i	0.19	0.19
am	0.19	0.19
happy	0.15	0.09
because	0.09	0.05
learning	0.09	0.09
nlp	0.09	0.09
sad	0.09	0.15
not	0.09	0.15

Таблиця 1.7.5 Умовні ймовірності слів належати до класу після застосування згладжування Лапласа.

Видно, що більші значення стали меншими, а менші навпаки, проте сума ймовірностей залишилася рівна 1.

Далі застосовуючи формулу Баєса, (1.7.4) можна розрахувати ймовірність, що документ належить відповідному класу.

3. Рекурентна нейронна мережа LSTM.

Традиційні штучні нейронні мережі, не запам’ятовують інформацію при обробці моделі, проте інколи це може бути недоліком. Тому існують рекурентні нейронні мережі, що місять цикли у моделі, що дозволяє зберігати додаткову інформацію.

Довго-короткострокова пам’ять (Long Short Term Memory, LSTM) – це особливий вид рекурентної нейронної мережі, що були запропоновані Хохрейтером та Шмідхубером у 1997 році. В основі моделі є блоки, що повторюються, а також через кожен блок проходить лінія стану, що дозволяє зберегти інформацію для наступних. Кожен блок складається з чотирьох нейронів (див. рис. 1.7.6).

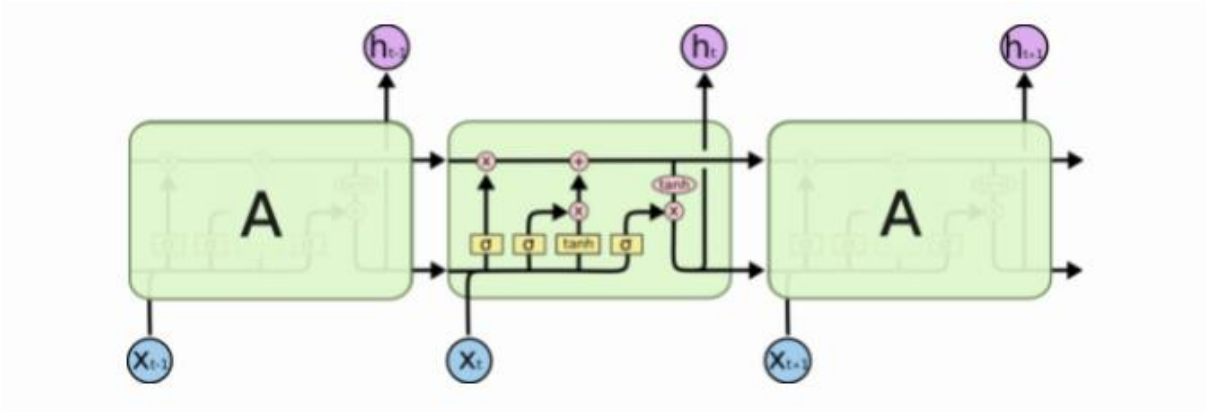

Рисунок 1.7.6 Архітектура блоку LSTM.

У статті Кристофа Олаха міститься опис всього алгоритму [10]. Це надскладна модель, розбір якої потребує окремої роботи, тому в цій роботі розглянуто її застосування використовуючи вже надані готові рішення.

Розділ 2. ПРЕДМЕТНА ОБЛАСТЬ

2.1 Вступ у предметну область

Ціллю цієї роботи є на основі великої кількості текстових даних зробити прогноз поведінки цінних паперів на фондовому ринку. Сам фондовий ринок є абстрактним поняттям, а саме, узагальненням усіх понять про дії та механізми, що роблять можливим торгівлю цінними паперами. Сама ж торгівля фактично відбувається на фондовій біржі.

«Фондова біржа – організаційно оформлений, постійно діючий ринок, на якому здійснюється торгівля цінними паперами; акціонерне товариство, яке зосереджує попит і пропозицію цінних паперів, сприяє формуванню їх біржового курсу та здійснює свою діяльність відповідно до чинного законодавства, статуту і правил фондової біржі» [11].

На фондовій біржі відбуваються торги по різним цінним паперам, у даній роботі розглядається прогнозування поведінки ринкової ціни акцій компаній та індексів.

«Акція – це вид цінних паперів, що являє собою свідоцтво про власність на визначену частку статутного капіталу акціонерного товариства і надає її власнику (акціонеру) певні права, зокрема: право на участь в управлінні товариством, право на частину прибутку товариства у випадку його розподілу (дивіденд), а у випадку ліквідації – на частину залишкової вартості підприємства» [12]

«Ринкова ціна або курс акцій – це фактична продажна ціна акцій на ринку цінних паперів. Тобто це ціна, за якою акція конвертується у грошовий вираз на вторинному ринку цінних паперів. Як правило, її розмір відрізняється від номінальної вартості, тому що прямо пропорційний отриманому дивідендові з урахуванням перспектив його зростання у майбутньому» [13]

Капіталізація фондового ринку США є найбільшою у світі та перевищує 30 трлн доларів. А найбільшими фондовими біржами по об'ємах торгів є NYSE (New York Stock Exchange) та NASDAQ (National Association of Securities Dealers Automated Quotation).

Основні фондові індекси на цих біржах:

- NYSE: NYSE Composite Index, The Dow Jones Composite Average (DJCA).
- Nasdaq: Nasdaq Composite (IXIC), Nasdaq 100.

Фондовий індекс – це показник зміни цін групи цінних паперів на ринку, наприклад, DJCA вказує на зміну цін топ-30 компаній на фондовій біржі NYSE, а його ціна визначається як середнє зважене всіх цін.

2.2 Дані про ціну акцій

Існує багато ресурсів де містяться дані про ціну акцій на фондовій біржі. Одним з таких є інтернет-ресурс Yahoo Finance [14]. Для зручності сайт надає відкрите API та відповідні пакети для зручної з ним роботи для різних мов програмування. Для роботи було застосовано пакет yfinance [15] для Python. З його допомогою можна вивантажити історичні дані про кожен день на ринку за весь період існування акцій. Проте, для кожного корпусу буде використано відповідний діапазон часу, що теж дозволяє вказати пакет. Приклад коду на рис. 2.2.1.

```
import yfinance as yf


dji = yf.Ticker('^DJI')
data = dji.history(period='1y', interval='1d')
```

Рисунок 2.2.1 Приклад коду вивантаження історичних даних.

Змінна history містить в собі об’єкт класу pandas.DataFrame. Pandas [16] – це програмна бібліотека, написана для мови програмування Python для зручної

роботи з даними. Одним з найголовніших елементів цієї бібліотеки є об’єкт pandas.Dataframe, що містить в собі самі дані та спеціальні методи для роботи з ними. Наприклад, метод tail() дозволяє вивести таблицю останніх п’яти днів цін на біржі (див. рис. 2.2.2).

```
data.tail()
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2021-03-19	32858.359375	32858.359375	32505.070312	32627.970703	8118900	0	0
2021-03-22	32601.820312	32810.351562	32512.529297	32731.199219	3835000	0	0
2021-03-23	32691.500000	32753.769531	32356.279297	32423.150391	3858400	0	0
2021-03-24	32470.880859	32787.988281	32418.150391	32420.060547	3993900	0	0
2021-03-25	32346.810547	32672.689453	32071.410156	32619.480469	4119900	0	0

Рисунок 2.2.2 Таблиця перших п’яти днів у історичних даних.

В одному рядку вказані дані про один день. В колонках Open та Close вказана ціна акції на початку та кінці торгів відповідно. В колонках Low та Hight вказана найменша та найбільша ціна під час торгів за день. Volume – це міра того, яка частина фінансового активу торгувалася за день. Dividends відповідає за суму виплат акціонерам у цей день. Stock Splits – це дроблення акцій. Всі стовпчики є невідкладною частиною технічного аналізу ринку, проте дана робота зосереджується на впливі текстів різного виду на позицію, що скорше відноситься до фундаментального аналізу ринку. Тому, з цієї таблиці для роботи потрібно тільки стовпець Open, а точніше різниця значень одного дня до наступного. З цієї інформації можна визначити чи виросла ціна за день чи впала.

2.3 Природа текстових даних

Існує безліч ресурсів та текстів, що можуть якимось чином впливати на поведінку позицій. На цю тему є дуже багато думок та суперечок і складно визначити який з них важливіший, тому в цій роботі розглядається декілька різновидів текстових даних:

1. Загальні новини.
2. Фінансові новини.
3. Новини пов’язані з назвою компанії.

Для кожного різновиду використовуються різні набори даних та підходи до обробки, тому надалі в дослідженнях варто розглядати кожен різновид окремо.

Всі дані базуються на ресурсах, що є визнаними світовою спільнотою та налічують сотні тисяч читачів. Також всі дані взяті з відкритих джерел, тому результати дослідження легко перевірити використовуючи вказані методи.

2.4 Специфіка підготовки до тренування та тестування моделей

Важливою частиною підготовки даних до тренування є процес розділу даних для навчання, валідації та тестування. Для прикладу розглянемо графік (рис. 2.4.1) позицій акцій Tesla за останній рік.

Рисунок 2.4.1 Графік ціни акції Tesla з позначенням поділу даних на тренувальні та тестувальні.

Якщо стандартно поділити дані за принципом 70% для тренування , а 30% для тестування як на графіку, то відразу видно, що модель тренуватиметься на цінах, що постійно зростає, а тестуватиметься на спадаючих – це помилка. Для того, щоб її уникнути потрібно поділити дані рівномірно. Для цього, наприклад, можна застосувати випадкове визначення тренувальної частини з ймовірністю 70% і так само тестувальні та валідаційні. Це дозволить тренувальній частині взяти максимум інформації з всього періоду, а не лише якоїсь частини.

Щодо тестування, то до всіх наборів даних буде побудована модель класифікації, що повертає значення 1 або 0. За показник успішності моделі було обрано точність (accuracy) на тестових даних.

$$\text{точність} = \frac{\text{кількість вірно визначених класів}}{\text{кількість всіх визначень}}$$

Найбільшою перевагою точності є інтерпритованість та реальна оцінка збалансованих даних. Тому, якщо відношення кількості класів в даних близьке до одиниці, така метрика добре пояснює результат.

Розділ 3. ЗАСТОСУВАННЯ

3.1 Загальні новини

Для роботи з загальними новинами було обрано ресурс Reddit [17] як джерело найважливіших новин. На відміну від стандартних світових новин у Reddit є функція «підняття» новини в гору користувачами, таким чином можна вивантажувати найгарячіші новини за день за думкою користувачів. А оскільки ця платформа налічує понад 1 млрд користувачів, це дає підстави вважати їх думку репрезентативною.

На відкритому ресурсі Kaggle [18] є набір даних «Daily News for Stock Market Prediction» [19], що містить набір топ-25 заголовків найважливіших новин у Reddit кожного дня з 2008 по 2016 рік. Цей набір дуже влучно підходить для нашої задачі. Також він одразу містить позначення (label) росту чи падіння індексу Доу Джонса.

Розглянемо колонки набору (табл. 3.1.1)

Date	Label	Top1	Top2	…	Top25

Таблиця 3.1.1 Назви стовпців у наборі даних.

У колонці Date міститься дата відповідного дня. Label – це значення 0 або 1. Значення 0, якщо позиції індексу по завершенню дня впадуть, 1 - якщо залишаться незмінними або виростуть. У комірках з колонкою Top1..Top25 містяться заголовки, що відповідають своїй позиції в рейтингу у день Date. На рис. 3.1.2 зображена таблиця перших десяти стовпців та п’яти рядків набору даних.

	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8
0	2008-08-08	0	b"Georgia 'downs two Russian warplanes' as cou...	b'BREAKING: Musharraf to be impeached.'	b'Russia Today: Columns of troops roll into So...	b'Russian tanks are moving towards the capital...	b"Afghan children raped with 'impunity,' U.N. ...	b'150 Russian tanks have entered South Ossetia...	b"Breaking: Georgia invades South Ossetia, Rus...	b"The 'enemy combatent' trials are nothing but...
1	2008-08-11	1	b'Why wont America and Nato help us? If they w...	b'Bush puts foot down on Georgian conflict'	b"Jewish Georgian minister: Thanks to Israeli ...	b'Georgian army flees in disarray as Russians ...	b"Olympic opening ceremony fireworks 'faked'"	b'What were the Mossad with fraudulent New Zea...	b'Russia angered by Israeli military sale to G...	b'An American citizen living in S.Ossetia blam...
2	2008-08-12	0	b'Remember that adorable 9-year-old who sang a...	b"Russia 'ends Georgia operation'"	b"If we had no sexual harassment we would hav...	b"Al-Qa'eda is losing support in Iraq because ...	b'Ceasefire in Georgia: Putin Outmaneuvers the...	b'Why Microsoft and Intel tried to kill the XO...	b'Stratfor: The Russo-Georgian War and the Bal...	b"I'm Trying to Get a Sense of This Whole Geor...
3	2008-08-13	0	b' U.S. refuses Israel weapons to attack Iran:...	b"When the president ordered to attack Tskhinv...	b' Israel clears troops who killed Reuters cam...	b'Britain\'s policy of being tough on drugs is...	b'Body of 14 year old found in trunk; Latest (...	b'China has moved 10 *million* quake survivors...	b"Bush announces Operation Get All Up In Russi...	b'Russian forces sink Georgian ships '
4	2008-08-14	1	b'All the experts admit that we should legalis...	b'War in South Osetia - 89 pictures made by a ...	b'Swedish wrestler Ara Abrahamian throws away ...	b'Russia exaggerated the death toll in South O...	b'Missile That Killed 9 Inside Pakistan May Ha...	b"Rushdie Condemns Random House's Refusal to P...	b'Poland and US agree to missle defense deal. ...	b'Will the Russians conquer Tblisi? Bet on it,...

Рисунок 3.1.2 Перші десять стовпців та п’ять рядків набору даних.

Всі топ-25 заголовків статей є документом у день Date, що впливає на значення колонки Label, тому їх було об’єднано конкатенацією через знак пробілу в один текст для кожного дня. Також, для кожного такого тексту було застосовано всі етапи попередньої обробки для підготовки до векторизації. На рис. 3.1.3 зображена таблиця вже з підготовленими текстами.

	Date	Label	Combined Texts
0	2008-08-08	0	georgia down two russian warplan countri move ...
1	2008-08-11	1	wont america nato help us wont help us help ir...
2	2008-08-12	0	rememb ador yearold sang open ceremoni fake ru...
3	2008-08-13	0	us refus israel weapon attack iran report pres...
4	2008-08-14	1	expert admit legalis drug war south osetia pic...

Рисунок 3.1.3 Дані після обробки.

Для прикладу, частина тексту у день 08.08.2008 має вигляд:

«georgia down two russian warplan countri move brink war break musharraf impeach russia today column troop roll south ossetia footag fight youtub russian tank move toward capit south ossetia reportedli complet destroy georgian artilleri fire afghan children rape impun un offici say sick three year old rape noth russian tank enter south ossetia whilst georgia shoot two russian jet break georgia invad south ossetia russia warn would interven so side enemi combat trial noth sham salim haman sentenc year kept longer anyway feel like georgian troop retreat osettain capit presum leav sever hundr peopl kill video…»

Задачею моделі є прогнозування позицій індексу Доу Джонса у період між 08.08.2008 та 01.07.2016, тому варто дослідити поведінку позицій за цей період. На рис. 3.1.4 зображено графік ціни індексу за вказаним періодом.

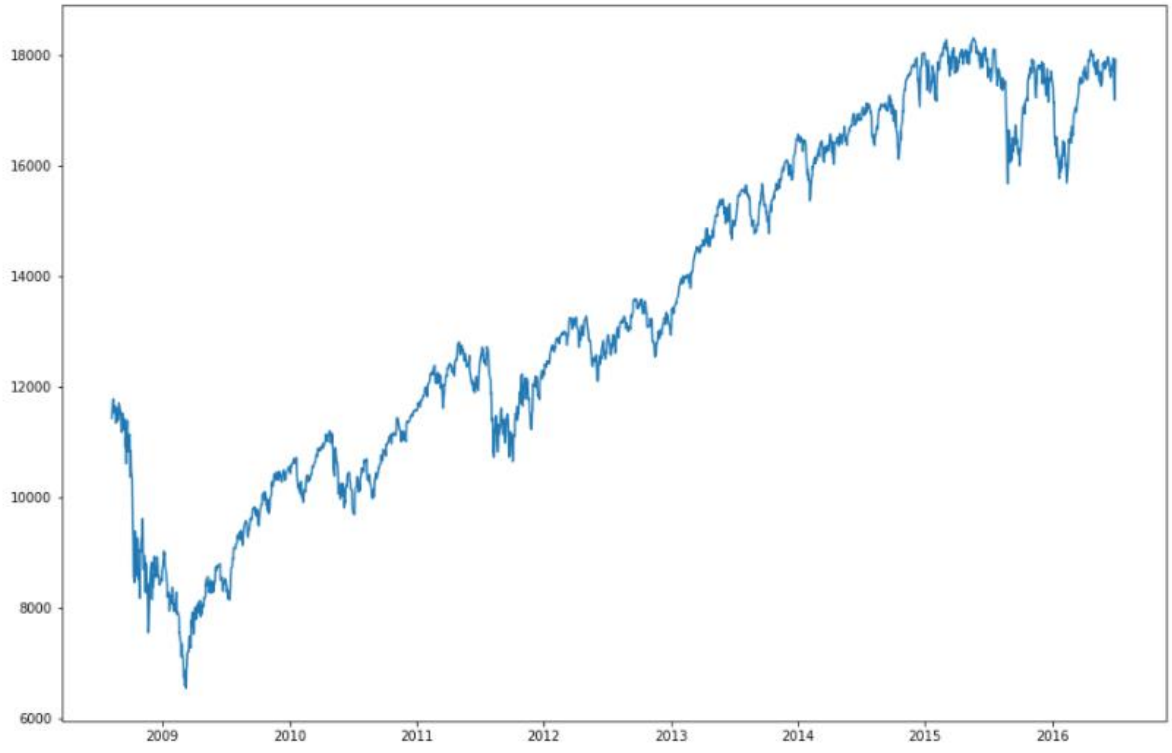

Рисунок 3.1.4 Графік позицій індексу Доу Джонса між 08.08.2008 та 01.07.2016

З графіку видно, що у цей час ціна більше росла ніж падала, проте ріст був не лінійним і часто на графіку можна побачити падіння, особливо до 2010 року. Щоб краще дослідити поведінку ціни було побудовано таблицю різниць ціни в

якій кожного дня є значення на скільки ціна виросте або впаде (від'ємне значення). За цими значеннями на рис. 3.1.6 зображено гістограму частот, а на рис. 3.1.5 коробку з вусами.

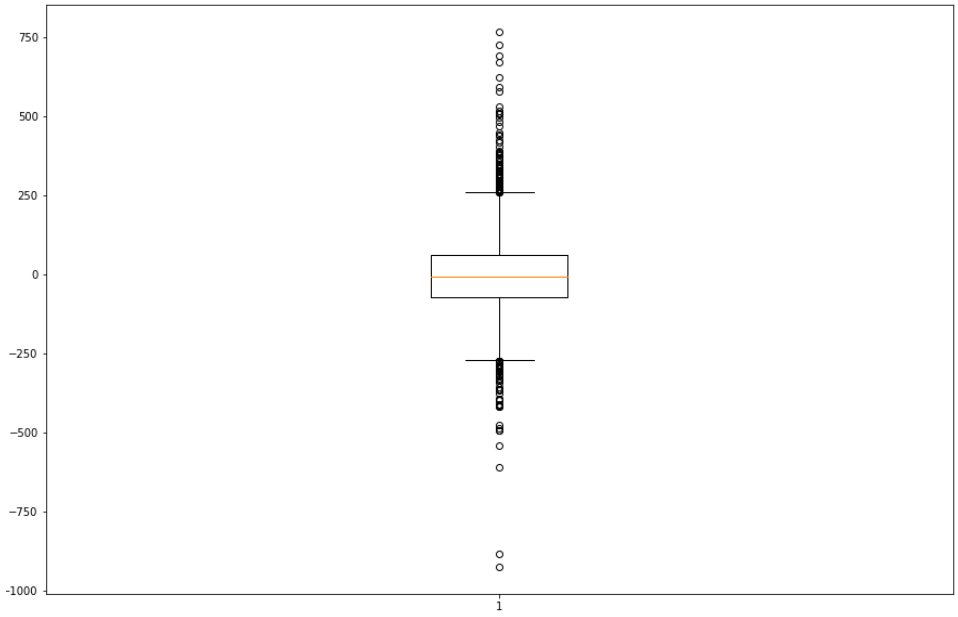

Рисунок 3.1.5 Коробка з вусами різниць ціни.

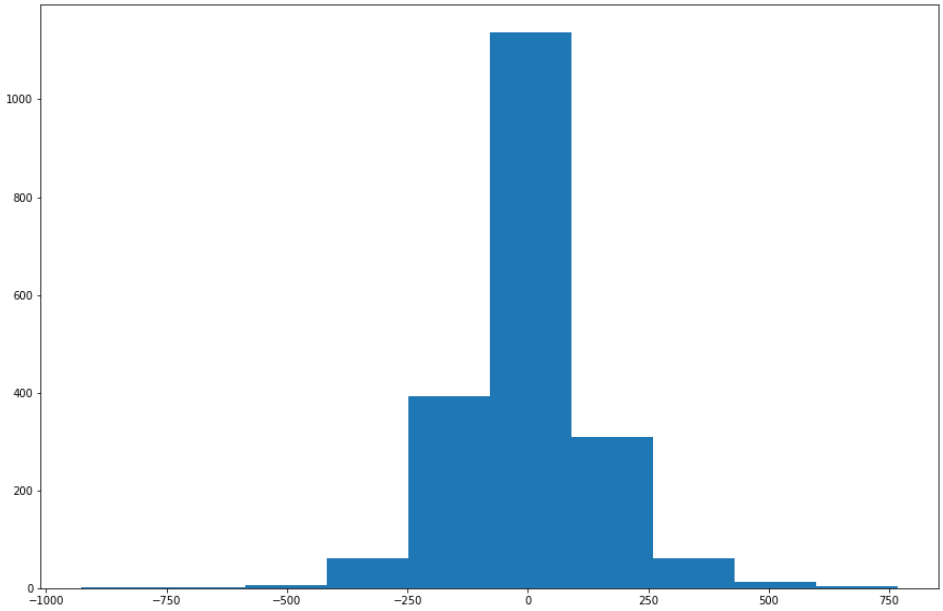

Рисунок 3.1.6 Гістограма різниць ціни.

Коробки з вусами на рис. 3.1.5 вказує на те, що 95% даних знаходяться в діапазоні (-250, 250), а 50% в діапазоні (-100, 100). Також видно, що середнім та медіаною приблизно є 0, про що свідчить і гістограма на рис. 3.1.6. Також

розподіл на гістограмі схожий на нормальний. Виходячи з цього можна сказати, що при переході до бінарних даних, кількість елементів класу буде приблизно однакова. Це можна побачити і на гістограмі класів на рис. 3.1.7.

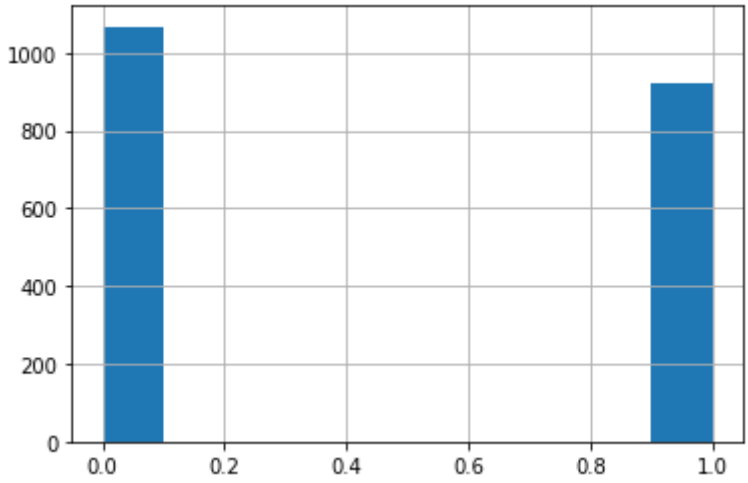

Рисунок 3.1.7 Гістограма класів.

Рівномірність кількості елементів у класі дозволяє краще натренувати модель, яка буде готова до різних випадків однаково. Далі, текстові дані та позначення класу були розділені випадковим способом зі збереженням класового розподілу в пропорції 70% – тренувальні, 30% тестові дані (як зазначено у підрозділі 2.5)

Далі, було проведено ряд експериментів. Для всіх варіацій векторизаторів, моделей, розмірів n-gram діапазону та кількості топ заголовків в документі було запущено тренування моделі та замір її ефективності, щоб знайти найкращий набір інструментів та вхідних даних для побудови моделі. Фрагмент таблиці результатів на рис. 3.1.8.

	accuracy	vectorizer	model	ngram_range	pred_Y_1	pred_Y_0	article count
0	0.517588	TfidfVectorizer	Naive Baies	(1, 1)	0.556114	0.443886	9
1	0.549414	TfidfVectorizer	Naive Baies	(1, 2)	0.618090	0.381910	9
2	0.556114	TfidfVectorizer	Naive Baies	(1, 3)	0.618090	0.381910	9
3	0.519263	CountVectorizer	Naive Baies	(1, 1)	0.510888	0.489112	9
4	0.546064	CountVectorizer	Naive Baies	(1, 2)	0.587940	0.412060	9
5	0.549414	CountVectorizer	Naive Baies	(1, 3)	0.597990	0.402010	9
6	0.537688	TfidfVectorizer	LogisticRegression	(1, 1)	0.710218	0.289782	9
7	0.546064	TfidfVectorizer	LogisticRegression	(1, 2)	0.862647	0.137353	9
8	0.547739	TfidfVectorizer	LogisticRegression	(1, 3)	0.907873	0.092127	9

Рисунок 3.1.8 Фрагмент результатів.

В результаті найкращим набором за метрикою accuracy вийшли наївний Баєсів класифікатор з попередньою векторизацією TF-IDF, n-gram діапазоном в (1, 3) та кількістю заголовків розміром в 9.

Проте навіть найкращий результат у майже 56% є вкрай малим для бінарної класифікації. Це може свідчити про погано підібрані моделі, або невірний процес підготовки даних, або дані дійсно не можна класифікувати за поведінкою індексу.

Рекурентна нейронна мережа на базі LSTM теж показала низький результат максимум в 60%. Тому, для того щоб краще зрозуміти дані, можна зменшити розмірність векторного простору та побудувати матрицю діаграм розсіювання, щоб візуально побачити різницю між класами. Методом головних компонент була зменшена розмірність до п’яти та на рис. 3.1.9 можна побачити матрицю з діаграмами розсіювання по кожній парі стовпців, де кольори вказують на клас. Зелений це ріст ціни, а червоний – падіння.

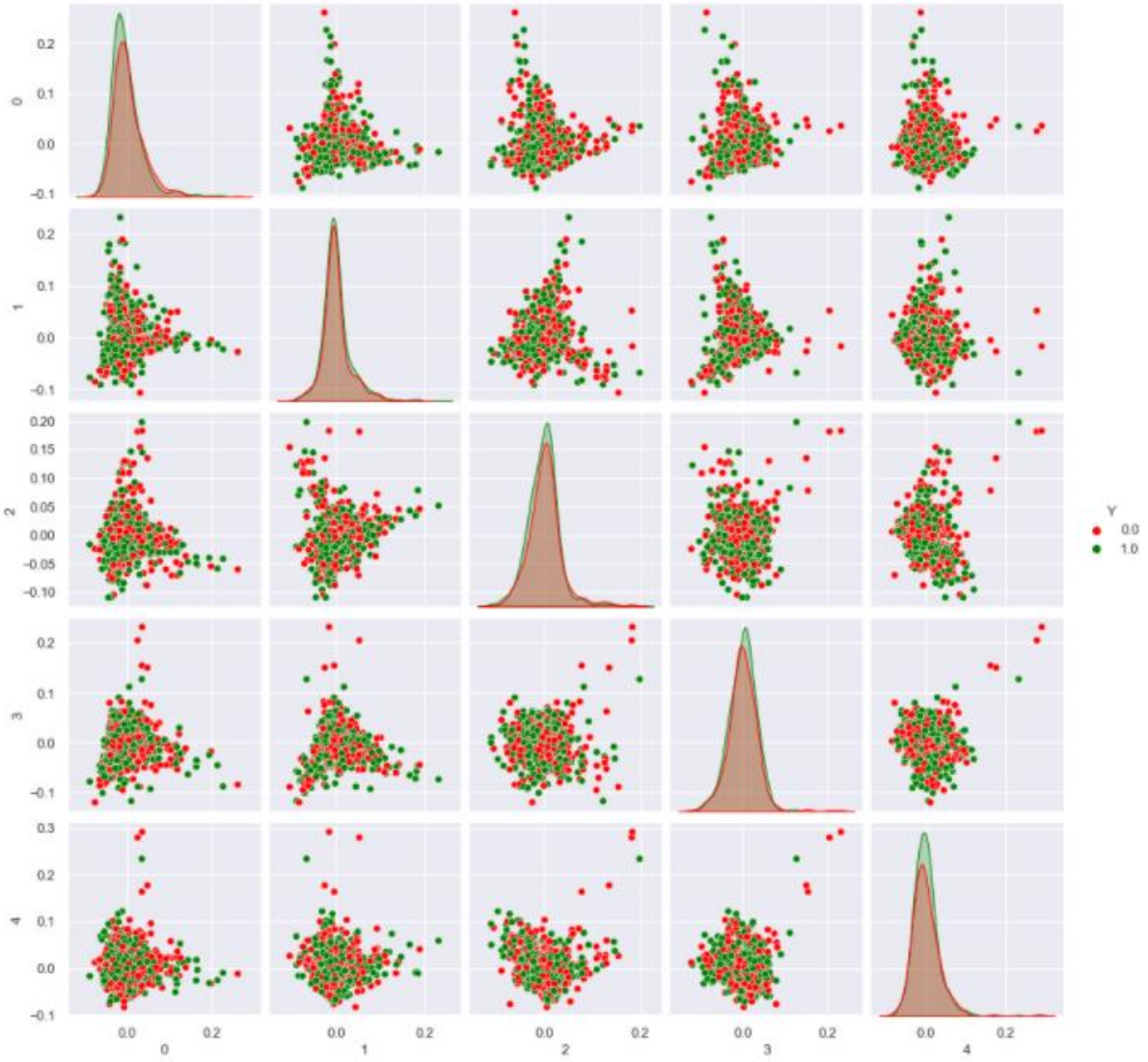

Рисунок 3.1.9 Матриця діаграм розсіювання.

З рисунка видно, що позиції класів дуже перемішані та складно виділити якісь зони окремих класів (враховуючи, що на малюнку кольори перекривають один одного по даті). Це свідчить про те, що проблема не в виборі моделі, а в самих вхідних даних. Їх складно класифікувати за поведінкою індексу на фондовій біржі. Проблема може бути в попередній обробці тексту або в методі векторизації документів, проте, враховуючи дуже низькі показники результатів моделей, найімовірніше загальні новини з обраного ресурсу ніяким чином не відображають та на їх основі, на жаль, неможливо класифікувати поведінку індексу Доу Джонса.

3.2 Фінансові новини

За ресурс економічних та фінансових новин було обрано видавництво Reuters [20]. Цей ресурс має велике визнання у світі та спеціалізується на новинах пов’язаних з фінансами та економікою.

На ресурсі Kaggle було обрано набір даних «Financial News Headlines Data» [21], що містить більше ніж 32 тисячі заголовків статей Reuters у період між 20.03.2018 та 18.07.2020. На рис. 3.2.1 зображена таблиця перших та останніх п’яти елементів даних.

	Headlines	Time
0	TikTok considers London and other locations fo...	Jul 18 2020
1	Disney cuts ad spending on Facebook amid growi...	Jul 18 2020
2	Trail of missing Wirecard executive leads to B...	Jul 18 2020
3	Twitter says attackers downloaded data from up...	Jul 18 2020
4	U.S. Republicans seek liability protections as...	Jul 17 2020
...	...	...
32765	Malaysia says never hired British data firm at...	Mar 20 2018
32766	Prosecutors search Volkswagen headquarters in ...	Mar 20 2018
32767	McDonald's sets greenhouse gas reduction targets	Mar 20 2018
32768	Pratt & Whitney to deliver spare A320neo engin...	Mar 20 2018
32769	UK will always consider ways to improve data l...	Mar 20 2018

Рисунок 3.2.1 Таблиця перших та останніх п’яти елементів даних.

Далі, згрупувавши дані за днем (конкатенацією тексту через пробіл), можна отримати документ для кожної дати. А провівши всі етапи попередньої обробки тексту – отримати готовий до тренування набір даних.

Ціллю цього набору є також прогнозування класу росту чи падіння індексу Доу Джонса. Тому, теж варто розглянути графіку позицій у цей період на рис. 3.2.2.

Рисунок 3.2.2 Графік позицій індексу Доу Джонса.

З графіку видно, що дані розбавлені як падінням, так і ростом, що є добрим для тренування моделі. Також це підтверджує діаграма класів на рис. 3.2.3.

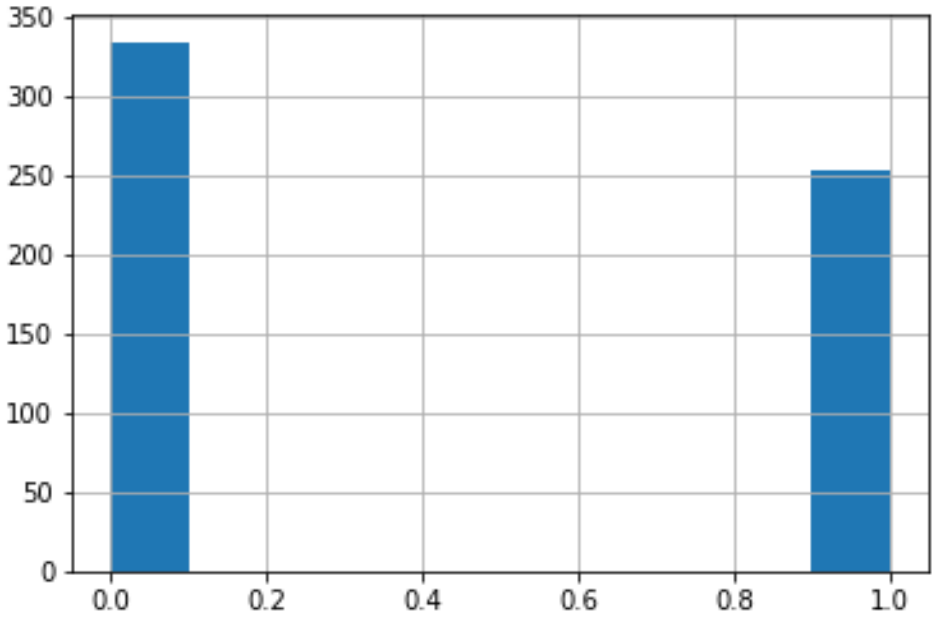

Рисунок 3.2.3 Діаграма класів.

В результаті, проведення циклу експериментів з підбору моделі та додаткових елементів, найкращий результат показала логістична регресія з векторизатором на базі підрахунку слів з діапазоном n-gram в (1, 2). Фрагмент підсумкового результату в таблиці на рис.3.2.4.

	accuracy	vectorizer	model	ngram_range
0	0.627119	TfidfVectorizer	LogisticRegression	(1, 1)
1	0.615819	TfidfVectorizer	LogisticRegression	(1, 2)
2	0.610169	TfidfVectorizer	LogisticRegression	(1, 3)
3	0.610169	TfidfVectorizer	LogisticRegression	(1, 4)
4	0.649718	CountVectorizer	LogisticRegression	(1, 1)
5	0.683616	CountVectorizer	LogisticRegression	(1, 2)
6	0.672316	CountVectorizer	LogisticRegression	(1, 3)
7	0.666667	CountVectorizer	LogisticRegression	(1, 4)
8	0.570621	TfidfVectorizer	Naive Baies	(1, 1)
9	0.610169	TfidfVectorizer	Naive Baies	(1, 2)

Рисунок 3.2.4 Фрагмент результатів тестування моделей.

Загалом, метрики всіх моделей є кращими ніж у попередньому наборі з загальними новинами. І у цьому є логіка, бо економічні новини більш точно вказують на якісь проблемні або позитивні точки. Проте, навіть показник у приблизно 70% є дуже ризиковим для такої задачі. Також найкращий результат у теоретично не найкращої моделі для цієї задачі на практиці (LSTM показав 61% accuracy) може свідчити про те, що це специфічний набір даних або дійсно дані не пов’язані з індексом.

3.3 Новини пов’язані з назвою компанії

Для роботи з новина пов’язаними з назвою компанії було обрано три компанії: Tesla, Amazon та Apple. Набір даних було взято з ресурсу Kaggle з назвою «Multipurpose World News Dataset» [22], що містить назви статей з 17 новинних ресурсів за кожен день з 21 березня 2020 року та налічує майже 200 тисяч текстів.

Далі тексти було відфільтровано за ключовими словами, пов’язаними з назвою компанії. Для прикладу для компанії Amazon були застосовані ключові слова: «Amazon», «AMZN» (назва акції), «Bezos» (прізвище голови компанії).

Для отриманих текстів були вирахувані класові відповідники з допомогою таблиці цін з Yahho Finance як і в попередніх підчастинах. Фрагмент результату можна побачити на рис. 3.3.1.

	company	accuracy	vectorizer	model	ngram_range
86	Tesla	0.642857	CountVectorizer	Naive Baies	(1, 3)
28	Apple	0.600000	CountVectorizer	Naive Baies	(1, 1)
63	Amazon	0.580000	CountVectorizer	Naive Baies	(1, 4)
22	Apple	0.580000	CountVectorizer	Naive Baies	(1, 3)
61	Amazon	0.580000	CountVectorizer	Naive Baies	(1, 2)
60	Amazon	0.580000	CountVectorizer	Naive Baies	(1, 1)
30	Apple	0.580000	CountVectorizer	Naive Baies	(1, 3)

Рисунок 3.3.1 Фрагмент результатів тестування моделей.

В результаті, тексти пов’язані з компанією Tesla показали найвищу точність в 64%. Проте, цей результат також можна вважати не значущим.

Також, варто зазначити, що в результаті фільтрування набору даних за ключовими словами кількість документів становила від 130 до 170 елементів залежно від обраної компанії. Такий набір даних може бути занадто малим для тренування моделей для такого типу задач.

3.4 Аналіз

Провівши тестування на різних наборах даних та застосовуючи різні алгоритми, методики та моделі, вважаю, що не вийшло досягнути бажаного результату. Проте, по всім показникам було показано, що фінансові новини дійсно мають більший вплив на поведінку індексу Доу Джонса ніж інші, а точність моделей побудованих за фільтрами по деталям компанії вища за точність прогнозу, що базується на загальних новинах для глобального індексу.

Також, варто зазначити, що не кожна стаття в новинах може мати вплив на фінансовий ринок, тому результат розглянутих моделей навіть при впливі деяких текстів є пояснюваним.

3.5 Аналіз та критика суміжних робіт

В мережі інтернет існують кілька робіт, що вказують на високі показники точності (понад 80%) схожих за методами та природою даних моделей. Проте, отримані результати кваліфікаційної роботи ставлять під сумнів результати даних робіт.

По перше, в більшості таких робіт не вказуються набори даних що використовуються, лише вказується, що це особисто накоплені дані з вказаного ресурсу. Наприклад, в роботі «Stock Trend Prediction Using News Sentiment Analysis»[23]. Це є проблемою, оскільки такі результати майже неможливо перевірити, а все ще не можна повторити або якось перевірити – не можна вважати за правду.

По друге, було знайдено кілька робіт в яких навмисно або помилково підібрані дані, що на етапі тестування дають хорошу точність. Наприклад в роботі «Stock Market Prediction using Natural Language Processing»[24], в публічному коді дані діляться так, що частина текстів з тренувального набору

попадають в тестувальний, що зумовлює високу точність. Фрагмент коду на рис.3.3.4.

```
In [3]: train = data[data['Date'] < '20150101']
        test = data[data['Date'] > '20141231']
```

Рисунок 3.5.4 Фрагмент коду поділу набору даних.

Якщо ж поділити дані випадковим чином, як вказано у підрозділі 2.4 , результат точності впаде до 54%.

Проте, це тільки маленька частина всіх робіт на цю тематику. Загалом, більшість досліджень вказують на точність в 45-65%. Зазвичай це зовсім різні методи та підходи, проте на схожих наборах даних. Це додатково свідчить про те, що всі новини в загальних підходах до аналізу тексту не показують великий вплив на поведінку цін на фондовому ринку.

ВИСНОВОК

В роботі оглянуто сучасні практики застосування обробки природної мови, реалізовано та протестовано декілька програм на різних наборах текстових даних на можливість класифікації для прогнозування у предметній області фондових ринків в межах запропонованих моделей машинного навчання.

З результатів дослідження випливає, що за підібраними наборами даних, розглянутими методами обробки та визначенню класів текстів по відношенню до поведінки позицій акцій та індексів на фондових ринках точність класифікації є в діапазоні 55-70%, що є незадовільним результатом для обраної предметної області.

Також, такі результати можуть вказувати як на слабкий зв’язок текстів такого типу з поведінкою обраних активів, так і на те, що може існувати кращий спосіб: обробки тексту такого типу, спосіб представлення документів у векторному вигляді та застосування моделей для класифікації.

Це, в свою чергу, створює поле для додаткових досліджень. Тому, продовженням цієї роботи можуть стати додаткові експерименти з вказаними частинами роботи. Для прикладу, варто розглянути моделі тексту, в яких порядок слів є важливим, або розглядати тільки ті тексти, вплив на акції яких є більшим середнього, таким чином розглядаючи тільки ті тексти, які дійсно могли мати вплив на обраний актив.

На перший погляд може здаватися, що вже всі методи обробки природної мови винайдені та застосовані в сучасному технологічному розвитку, проте досі існує безліч невирішених задач, тому застосування та розробка таких методів в різних сферах застосування розвиватиме як цю область, так і всю науку загалом.

Список Літератури

1. Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc
2. "Лінгвістичний аналіз тексту" як навчальна дисципліна у ВНЗ України / Н.П. Миронюк // Культура народов Причерноморья. — 2002. — № 32.
3. About Python. Python.org. URL: https://www.python.org/about/.
4. Natural Language Toolkit – NLTK. nltk.org. URL: https://www.nltk.org/.
5. Jones K. S. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation : журнал. — MCB University : MCB University Press, 2004. — Т. 60, № 5. — С. 493-502.
6. Proceedings of the 7th Annual Conference ZNALOSTI 2008, Bratislava, Slovakia, pp. 54-65, February 2008.
7. Pearson K., On lines and planes of closest fit to systems of points in space, Philosophical Magazine, (1901) 2, 559—572;
8. David A. Freedman (2009). Statistical Models: Theory and Practice. Cambridge University Press. с. 128.
9. Domingos, Pedro & Michael Pazzani (1997) «On the optimality of the simple Bayesian classifier under zero-one loss». Machine Learning, 29:103-137.
10. Understanding LSTM Networks. Colah's blog. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.
11. Юридична енциклопедія : [у 6 т.] / ред. кол. Ю. С. Шемшученко(відп. ред.) [та ін.] — К. : Українська енциклопедія ім. М. П.Бажана, 1998. — Т. 1 : А — Г. — 672 с.)
12. Слюсарчук Л. І. Акція (цінний папір) // Велика українська енциклопедія. URL: https://vue.gov.ua/Акція_(цінний_папір
13. Демківський А.В. Гроші та кредит Навчальний посібник / К.:Дакор, 2007.– 528 с. URL: http://www.info–library.com.ua/books–book–152.html
14. Yahoo Finance - Stock Market Live, Quotes, Business & Finance News. Yahoo Finance. URL: https://finance.yahoo.com/.

15. yfinance. PyPI. URL: https://pypi.org/project/yfinance/.
16. Pandas - Python Data Analysis Library. pandas.pydata. URL: https://pandas.pydata.org/.
17. Reddit. URL: https://www.reddit.com/.
18. Kaggle: Your Machine Learning and Data Science Community. Kaggle. URL: https://www.kaggle.com/.
19. Daily News for Stock Market Prediction, Kaggle, URL: https://www.kaggle.com/aaron7sun/stocknews.
20. Reuters, URL: https://www.reuters.com/
21. Financial News Headlines Data, Kaggle, URL: https://www.kaggle.com/notlucasp/financial-news-headlines
22. Multipurpose World News Dataset, Kaggle, URL: https://www.kaggle.com/gabrielmilan/multipurpose-world-news-dataset
23. Stock Trend Prediction Using News Sentiment Analysis, Kalyani Joshi, Bharathi H. N, Jyothi Rao, June 2016, DOI:10.5121/ijcsit.2016.8306, URL:https://www.researchgate.net/publication/304995235_Stock_Trend_Prediction_Using_News_Sentiment_Analysis
24. Stock Market Prediction using Natural Language Processing, URL: https://github.com/SATHVIKRAJU/Stock-Market-Prediction-using-Natural-Language-Processing