

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Навчально-науковий інститут бізнесу, економіки та
менеджменту

**Проблеми та перспективи розвитку
фінансово-кредитної системи
України**
**Problems and Prospects for the Development of the Financial
and Credit System**

Матеріали
Міжнародної науково-практичної конференції
(Україна, Суми, 4-5 грудня 2023 р.)

Суми
Сумський державний університет
2023

РОЛЬ ТА ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ НЕЙРОННОЇ МЕРЕЖІ У ВДОСКОНАЛЕННІ АНАЛІЗУ, КЛАСИФІКАЦІЇ ТА УПРАВЛІННЯ ВІДГУКАМИ НА ТОВАРИ

*Кільдей Анастасії Денисівни
студентка кафедри економічна кібернетика,
Сумський державний університет, м. Суми
Кушнерьов Олександр Сергійови,
асистент кафедри економічна кібернетика,
Сумський державний університет, м. Суми
Койбічук Віталія Василівна
доцентка, кандидатка економічних наук
кафедри економічна кібернетика,
Сумський державний університет, м. Суми*

У сучасному інформаційному суспільстві, насиченому великим обсягом текстової інформації, виникає необхідність в ефективних методах її аналізу та класифікації. Особливо це важливо для розпізнавання та систематизації текстів, які відіграють ключову роль у багатьох галузях, від бізнесу до науки. У цьому контексті, дослідження з питань класифікації та аналізу тексту за допомогою нейронних мереж стає важливим завданням, що визначає актуальність даної роботи.

Штучний інтелект (ШІ) активно впроваджується в різні сфери людського життя, переосмислюючи традиційні підходи до бізнес-процесів. Використання штучного інтелекту в бізнес-операціях має потенціал для збільшення продуктивності праці на організаційному рівні [1]. ШІ функціонує за допомогою алгоритмів, які представляють собою набори правил або інструкцій для обробки обширних обсягів даних та для навчання на їх основі. Штучний інтелект може використовувати різні методи, такі як обробка природної мови (NLP), машинне навчання (ML) та глибоке навчання (DL). Однією з областей, де ШІ виявляє значний вплив, є аналіз, класифікація та управління відгуками на товари.

Нейронні мережі також можуть сприяти автоматизації процесу аналізу відгуків клієнтів з метою контролю якості, зекономивши час і ресурси, і в той же час підвищуючи точність та послідовність отриманих результатів. Штучний інтелект здатний розкривати приховані шаблони, кореляції та деталі з відгуків клієнтів, які можуть залишитися непоміченими або пропущеними людьми. Крім того, ця технологія допомагає визначити потреби, очікування, уподобання та больові точки клієнтів з метою пропозиції дій чи рішень для вдосконалення продуктів та послуг.

Для ефективного використання таких технологій у процесі аналізу відгуків клієнтів для контролю якості необхідно збирати відповідні, чіткі та повні відгуки клієнтів з різних джерел, таких як опитування, огляди, рейтинги, соціальні мережі, електронні листи, чати, дзвінки та інші. Після цього необхідно провести очистку, організацію та форматування даних зворотного зв'язку для полегшення обробки алгоритмами штучного інтелекту.

На наступному етапі можна застосувати відповідну методологію, таку як обробка природної мови (NLP), машинне навчання (ML) або глибоке навчання (DL), для створення балів, міток, категорій, резюме чи рекомендацій. Нарешті, важливо розуміти та оцінювати результати алгоритмів штучного інтелекту і використовувати їх для удосконалення процесів контролю якості. Це може включати візуалізацію отриманих результатів або впровадження змін на основі зібраних відгуків клієнтів [2].

Наразі існує значна кількість готових датасетів, доступних на платформах, таких як Kaggle. Проте при зборі даних ми враховували такі параметри: чистота даних, оскільки допомогою генерації даних ми можемо згенерувати повністю чистий від шуму датасет, який буде містити найбільш емоційно забарвлені слова для ефективного та результативного навчання; можливість одразу класифікувати відгуки як позитивний та негативний, що полегшує подальший процес підготовки датасету; безумовно такий метод збору даних вважається легким, а дані можна генерувати нескінченно; генерування бази даних можливо на будь-якій мові; відсутність додаткових грошових витрат.

Після збору вихідних даних необхідно провести належну обробку тексту, оскільки цей етап вважається важливим у класифікації тексту. Підготовчий процес спрямований на зменшення вірогідності помилок та підвищення точності класифікації [3, 4]. Основною метою цього етапу є вилучення шумових та незначущих слів [3].

Кожен відгук проходив обробку за наступним процесом: спочатку здійснювалася очистка даних, що включала видалення дублікатів, цифр, розділових знаків і чисел, а також вилучення стоп-слів та непотрібних слів, таких як займенники та прийменники. Потім застосовувалася трансформація та токенізація, розбиваючи відгуки на токени. Наступним етапом було вилучення стоп-слів, які не мали відношення до концепції тексту. Після цього застосовувався стемінг для зведення словоформ до однієї форми шляхом вилучення префіксів, суфіксів та інфіксів. Послідовно проводився процес ваги термінів, де застосовувався алгоритм TF-IDF для визначення ваги термінів у кожному документі. На завершальному етапі відбувалася обрізка слів з метою відсічення менш часто вживаних ознак у наборі документів. Після виконання усіх цих етапів було сформовано датасет із 2

500 відгуків, включаючи такі колонки, як ПББ, контактні дані і сам відгук. Цей датасет може бути використаний для створення та навчання моделі.

Для тренування моделі був використаний сервіс Google Colab. Процес навчання моделі описаний у фрагменті коду, представленою на рисунку 1. Для тренування використовується бібліотека TensorFlow, яка дозволяє тренувати попередньо визначену нейронну мережу, використовуючи дані тренування та валідації. Крім того, реалізовано використання засобу ранньої зупинки (early stopping), який автоматично завершує процес навчання моделі, якщо протягом певної кількості епох (у цьому випадку 29 епох) не виявлено покращення на валідаційному наборі даних.

```
[ ] # Train the model
early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

history = model.fit(
    train_padded,
    train_labels,
    epochs=100,
    validation_data=(test_padded, test_labels),
    batch_size=64,
    callbacks=[early_stopping]
)

Epoch 22/100
31/31 [=====] - 1s 17ms/step - loss: 0.2077 - accuracy: 0.9460 - val_loss: 0.2890 - val_accuracy: 0.9133
Epoch 23/100
31/31 [=====] - 1s 27ms/step - loss: 0.1954 - accuracy: 0.9510 - val_loss: 0.2586 - val_accuracy: 0.9032
Epoch 24/100
31/31 [=====] - 1s 21ms/step - loss: 0.1834 - accuracy: 0.9500 - val_loss: 0.2463 - val_accuracy: 0.9153
Epoch 25/100
31/31 [=====] - 0s 12ms/step - loss: 0.1680 - accuracy: 0.9571 - val_loss: 0.2678 - val_accuracy: 0.9093
Epoch 26/100
31/31 [=====] - 1s 26ms/step - loss: 0.1618 - accuracy: 0.9611 - val_loss: 0.2308 - val_accuracy: 0.9194
Epoch 27/100
31/31 [=====] - 0s 12ms/step - loss: 0.1586 - accuracy: 0.9581 - val_loss: 0.2288 - val_accuracy: 0.9234
Epoch 28/100
31/31 [=====] - 1s 17ms/step - loss: 0.1466 - accuracy: 0.9616 - val_loss: 0.2295 - val_accuracy: 0.9254
Epoch 29/100
31/31 [=====] - 0s 11ms/step - loss: 0.1362 - accuracy: 0.9707 - val_loss: 0.2139 - val_accuracy: 0.9294
```

Рисунок 1 – Навчання моделі

Після завершення навчання моделі виведемо графік точності (Model accuracy) та втрат (Model loss). Графік показує динаміку точності та втрат моделі машинного навчання протягом процесу навчання.

З рисунку 2 бачимо, що точність моделі на тренувальному наборі (синя лінія) поступово зростає, досягаючи приблизно 96% після 20 епох. Точність на наборі для перевірки (помаранчева лінія) також збільшується, досягаючи приблизно 94% після 20 епох. Втрати моделі на обох наборах зменшуються протягом процесу навчання.

На графіку також видно, що втрати моделі на наборі для перевірки починають зростати після певної кількості епох, що означає можливість перенавчання моделі при збільшенні кількості епох. Перенавчання – це проблема, яка може виникнути в машинному навчанні, коли модель навчена на надмірно великому тренувальному наборі. У цьому випадку модель стає занадто точною для тренувального набору, але має низьку точність на новому наборі даних.

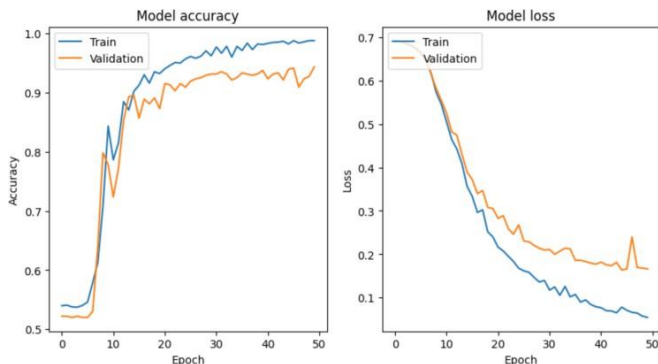


Рисунок 2 – Графік точності та втрат моделі

Таким чином, графік свідчить про те, що LSTM-модель вдається навчитися виконувати завдання бінарної класифікації тексту. Це означає, що модель може бути використана для прогнозування класу даних з високою точністю.

Список використаних джерел

1. Czarnitzki, D., Fernández, G. P., & Rammer, C. (Year). Artificial Intelligence and Firm-Level Productivity. *Journal of Economic Behavior & Organization*.
2. Quality Control. (2023, 16 квітня). How do you use AI to automate customer feedback analysis for quality control? LinkedIn: <https://www.linkedin.com/advice/0/how-do-you-use-ai-automate-customer-feedback-analysis>
3. Bahassine, S., Madani, A., Al-Sarem, M., & Kissi, M. (2018). Feature selection using an improved Chi-square for Arabic text classification. *Journal of King Saud University-Computer and Information Sciences*, 32(2), 225-231. <https://doi.org/10.1016/j.jksuci.2018.05.010>
4. Uysal, A., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, 50, 104–12. [doi:10.1016/j.ipm.2013.08.006](https://doi.org/10.1016/j.ipm.2013.08.006).