Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

“Інформаційна NLP-технологія класифікації україномовної інформації про стан водних ресурсів ”

Виконав: студент 2 курсу, групи 2ІСТ-21м
спеціальності 126 – «Інформаційні системи та технології»
_____ Пасічнюк Д.В.

Керівник: д.т.н., проф. каф. САІТ
_____ Мокін В.Б.
«01» 12 2022 р.

Опонент: к.т.н., доц. каф. АІІТ
_____ Сторчак В. Г.
«15» 12 2022 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«05» 12 2022 р.

Вінниця ВНТУ – 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – ІІ-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 126 Інформаційні системи та технології
Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«16» 09 2022 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Пасічнюку Дмитрові Васильовичу

1. Тема роботи: «Інформаційна NLP-технологія класифікації україномовної інформації про стан водних ресурсів»
керівник роботи: Мокін В.Б., д.т.н., проф. каф. САІТ,
затверджені наказом закладу вищої освіти від «14» 09 2022 року №203
2. Строк подання студентом роботи «01» 12 2022 року
3. Вихідні дані до роботи: Kaggle Dataset «NLP : Reports & News Classification» https://www.kaggle.com/datasets/vbmokin/nlp-reports-news-classification
4. Зміст розрахунково-пояснювальної записки:
   - аналіз проблеми автоматичної класифікації україномовної екологічної інформації;
   - вибір оптимальних налаштувань інформаційної технології для розв'язання поставленої задачі;
   - проєктування та створення інтелектуальної інформаційної технології класифікації україномовної текстової інформації про масиви вод;
   - економічна частина.
5. Перелік графічного матеріалу:
   - алгоритм розробки моделі класифікаторів;
   - алгоритм інформаційної технології;
   - сформований датасет для тестування моделей;
   - результат роботи відібраних моделей;
   - класифікація за допомогою моделі випадкових лісів (Random Forest).

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
4	Бурєннікова Н.В., д.е.н., проф. каф. ЕПВМ	01.11.22	30.11.22

7. Дата видачі завдання «11» 09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МКР	Строк виконання етапів роботи	Примітка
1	Характеристика проблеми класифікації україномовної інформації про стан водних ресурсів	09.2022	
2	Вибір оптимальних налаштувань інформаційної технології для розв'язання поставленої задачі	10.2022	
3	Створення інтелектуальної інформаційна технології класифікації україномовної інформації про масиви вод басейну річки південний Буг	11.2022	
4	Економічна частина	11.2022	
5	Оформлення матеріалів до захисту МКР	11.2022	

Студент _____ Пасічнюк Д. В.

Керівник роботи _____ Мокін В. Б.

АНОТАЦІЯ

УДК 004.8:338+332.628

Пасічнюк Д.В. Інформаційна NLP-технологія класифікації україномовної інформації про стан водних ресурсів. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2022. 87 с.

На укр. мові. Бібліогр.: 21 назв; рис.: 75; табл.: 5.

В магістерській кваліфікаційній роботі проведено аналіз предметної області класифікації текстової україномовної інформації. Здійснено огляд аналогічних рішень, запропоновано алгоритм створення ІТ класифікації текстової україномовної інформації, на основі якого проведено розроблення ІТ. Виконано пошук необхідних текстових даних, проведено попереднє очищення й сформовано датасет. Проведено розвідувальний аналіз даних, обрано регресійні моделі машинного навчання, здійснено їх тренування та визначено оптимальну модель серед них, точність передбачення якої – 0.95, що є більшим за 0.80, як у найкращого аналога.

Ілюстративна частина складається з 6 плакатів.

У розділі економічної частини розглянуто питання про доцільність розроблення та впровадження інформаційної NLP-технології класифікації україномовної інформації про стан водних ресурсів.

Ключові слова: інформаційна технологія, розвідувальний аналіз даних, класифікація тексту, водні ресурси, ознаки, моделі машинного навчання.

ABSTRACT

UDC 004.8:338+332.628

Pasichniuk D.V. Informational NLP technology for the classification of Ukrainian-language information about the state of water resources. Master's qualification work on specialty 126 - information systems and technologies, educational and professional program - information technologies of data and image analysis. Vinnytsia: VNTU, 2022. 87 p.

In Ukrainian languages Bibliography: 21 titles; Fig.: 75; tab.: 5.

In the master's qualification thesis, an analysis of the subject area of the classification of textual Ukrainian-language information was carried out. An overview of similar solutions was carried out, an algorithm for creating an IT classification of textual Ukrainian-language information was proposed, on the basis of which IT was developed.

The search for the necessary text data was performed, preliminary cleaning was carried out, and the dataset was formed. Exploratory data analysis was carried out, machine learning regression models were selected, their training was carried out, and the optimal model among them was determined, the prediction accuracy of which is 0.95, which is higher than 0.80, as the best analogue.

The illustrative part consists of 6 posters.

In the section of the economic part, the issue of the feasibility of developing and implementing the NLP information technology for the classification of Ukrainian-language information on the state of water resources is considered.

Keywords: information technology, data intelligence analysis, text classification, water resources, features, machine learning models.

ЗМІСТ

ВСТУП

Актуальність теми. У сучасному житті, коли все пов’язано з інформаційними технологіями, не складно знайти інформацію потрібної тематики, як наприклад про забруднення водойм, але пошук це лише половина проблеми, адже знайдена інформація часто містить неточності, дублюється чи навіть містить «фейкову» інформацію, також важко знайти необхідні корисні та чіткі дані про екологічний стан водойми.

Для того щоб визначити стан водойм із знайдених даних, потрібно проаналізувати їх та правильно класифікувати. Ця задача дуже об’ємна адже щоб зробити об’єктивні висновки потрібно опрацювати великі об’єми інформації, тому процес аналізу та класифікації потрібно автоматизувати за допомогою інтелектуальної інформаційної системи. Подібні технології вже існують, але – для англомовних текстів та інших задач. Зокрема, не має адаптацій до задачі класифікації інформації україномовних джерел про масиви вод басейну річки Південний Буг. Отже, розробка є актуальною.

Мета і задачі дослідження. Метою дослідження є підвищення точності класифікації текстової україномовної інформації про масиви вод басейну річки Південний Буг.

Розробка технології передбачає виконання наступних задач:

- огляд та вибір оптимальних інформаційних технологій;
- створення датасету та вибір моделей;
- вибір оптимальної моделі, створення інформаційної технології та її застосування для класифікації даних.

Об’єкт дослідження магістерської кваліфікаційної роботи є класифікація текстової україномовної інформації про масиви вод басейну р. Південний Буг.

Предмет дослідження магістерської кваліфікаційної роботи є інтелектуальна інформаційна технологія класифікації текстової україномовної інформації про масиви вод басейну р. Південний Буг.

На основі теоретичних результатів розроблено і реалізовано програмне забезпечення інтелектуальної інформаційної технології класифікації текстової україномовної інформації про масиви вод басейну р. Південний Буг. Результати впроваджено у діяльність басейнової ради Південного Бугу та у навчальний процес кафедри системного аналізу та інформаційних технологій ВНТУ.

Методи дослідження. У дослідженнях використано методи розвідувального аналізу даних, методи машинного навчання для побудови моделей та методи оброблення природномовних текстів на Python.

Новизна одержаних результатів. Наукова новизна полягає у тому, що дістала подальший розвиток інформаційна технологія класифікації текстової україномовної інформації про масиви вод басейну р. Південний Буг з використанням методів машинного навчання, яка дозволяє підвищити точність такої класифікації у порівнянні з аналогами.

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи доповідались на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2021-2022 рр.).

Публікація результатів магістерської кваліфікаційної роботи.

Опубліковано тези на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2021-2022 рр.) [1].

1 ХАРАКТЕРИСТИКА ПРОБЛЕМИ КЛАСИФІКАЦІЇ УКРАЇНОМОВНОЇ ІНФОРМАЦІЇ ПРО СТАН ВОДНИХ РЕСУРСІВ

1.1 Суть проблеми

З розвитком екологічної сфери та інформаційних технологій росте і кількість даних про екологічний стан важливих об’єктів, як наприклад водойми. На даний момент є чимало інформації про екологічний стан річок басейну Південного Бугу, в різного типу відкритих джерелах від монографій до веб сайтів. Для того, щоб мати уявлення про те, в якому стані знаходяться річки потрібно проаналізувати якомога більше джерел щоб висновки були максимально об’єктивними. Для того, щоб оптимізувати цей процес, а не перебирати масивні об’єми даних вручну, що буде менш ефективним через людський фактор, необхідно створити систему, за допомогою якої можна буде автомати процес аналізу та класифікації даних, відповідно до потреб.

Для того, щоб така технологія працювала, необхідно створити датасет із відповідним набором даних. Для цього також потрібно залучитись допомогою експертів, щоб визначити правильну розмітку, для отримання коректного тренувального датасету, яким потім можна буде навчати модель, щоб в подальшому вона могла розв’язувати необхідні задачі.

Датасет має складатись із конкретної кількості словосполучень, що вживаються в контексті речень, в яких ідеться про забруднення. Такі конструкції використовуються для того, щоб максимально точно розкрити і класифікувати довільну інформацію, як таку, яка характеризує забруднення вод Південного Бугу.

Для формування датасету можна використовувати різного роду джерела, в даному варіанті розглянемо такі, як веб джерела та файли формату docx. Далі детальніше про кожен із них:

– Збір даних із документів формату docx: в такому випадку текст потрібно заздалегідь очистити від зображень, таблиць чи деяких спеціальних символів (рис. 1.1) [2].

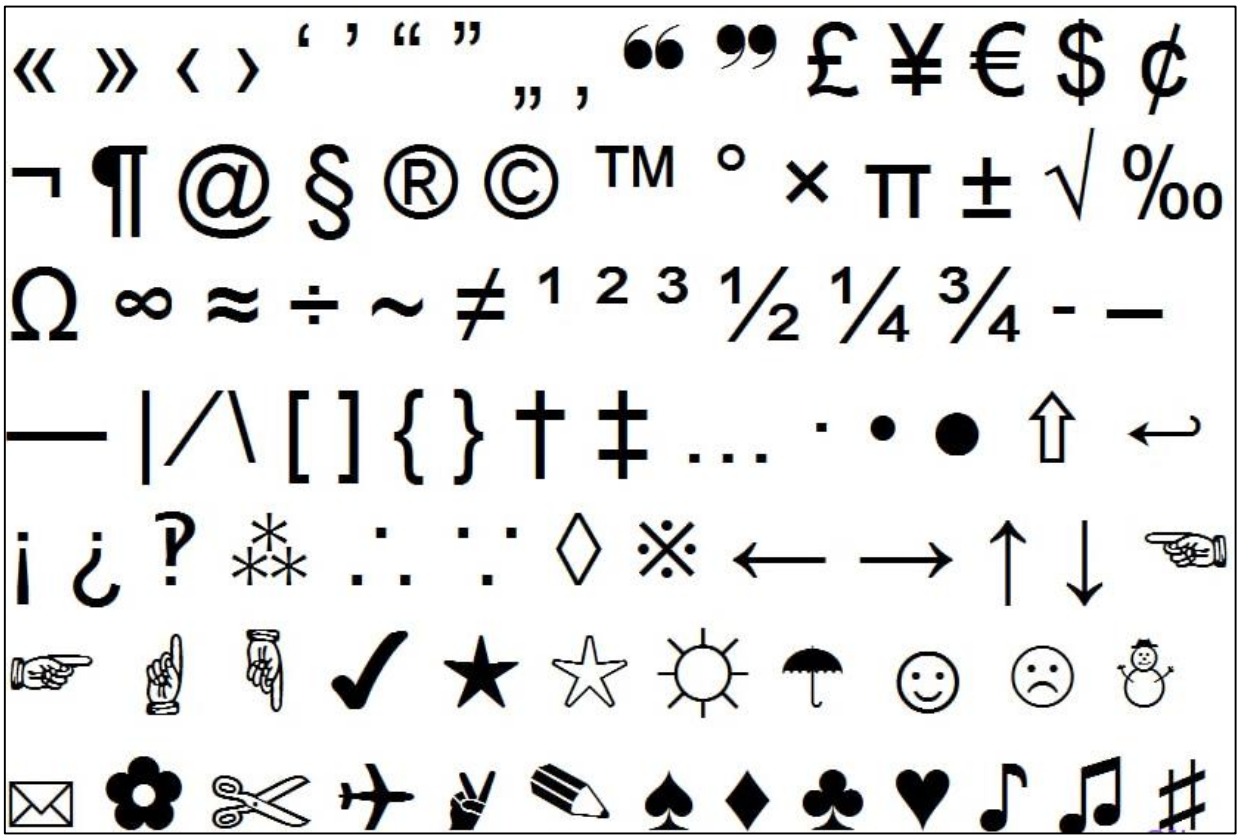
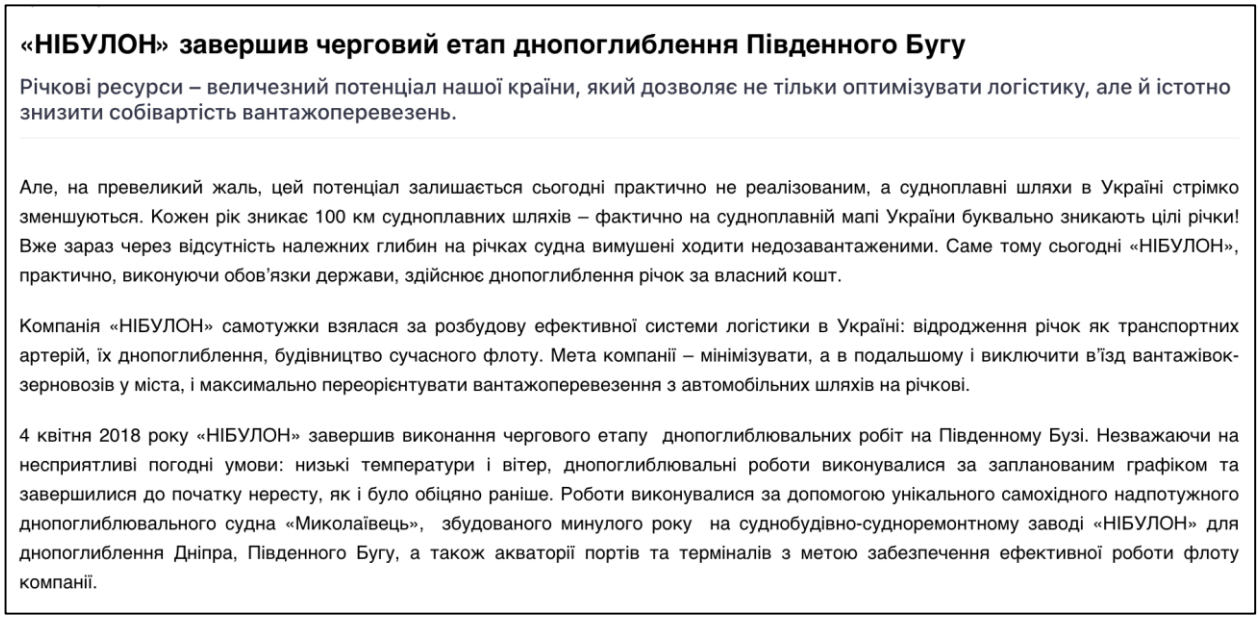


Рисунок 1.1 – Символи, яких потрібно позбутись перед автоматичним парсингом

Також при створенні датасету потрібно пам’ятати, що кількість нулів та одиниць має бути однаковою, задля того, щоб була можливість забезпечити нормальне навчання та тестування роботи моделей. Ще одним важливим нюансом є те, що речення не мають бути надто довгими, в такому випадку, їх варто обрізати.

– Пошук даних на веб сайтах: в такому випадку гарною практикою буде пошук даних на сайтах, де багато статей одної тематики (рис.1.2, 1.3), адже текстові дані зазвичай лежать в однакових ідентифікаторах доступу «class» (рис.1.4, 1.5), що в свою чергу значно зекономить час, бо зникне

необхідність шукати ідентифікатори доступу до кожного посилання окремо [3].



«НІБУЛОН» завершив черговий етап днопоглиблення Південного Бугу

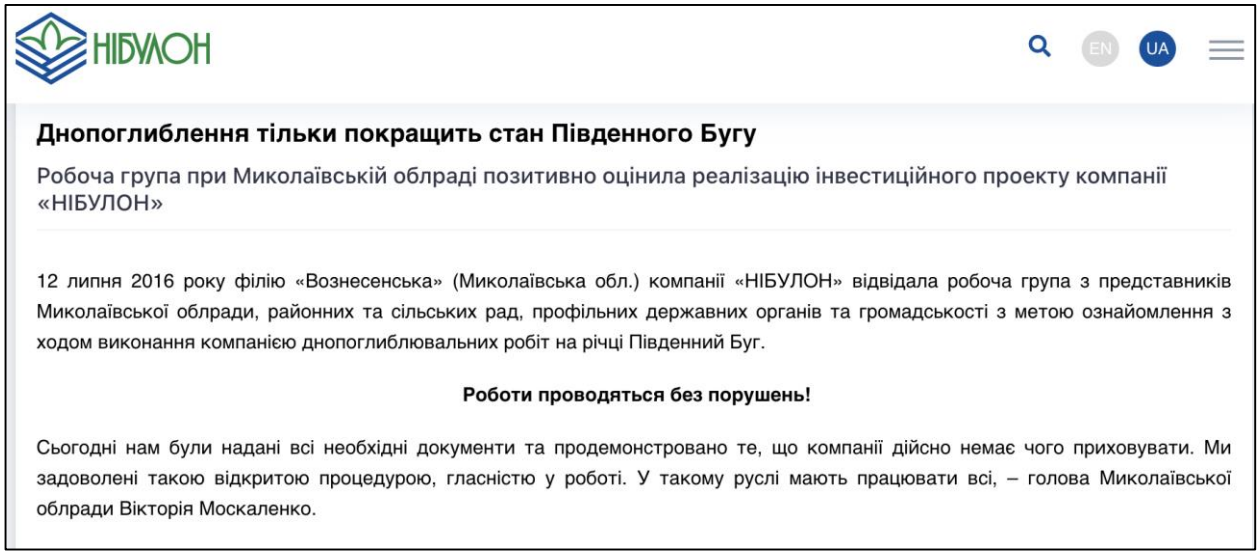
Річкові ресурси – величезний потенціал нашої країни, який дозволяє не тільки оптимізувати логістику, але й істотно знизити собівартість вантажоперевезень.

Але, на превеликий жаль, цей потенціал залишається сьогодні практично не реалізованим, а судноплавні шляхи в Україні стрімко зменшуються. Кожен рік зникає 100 км судноплавних шляхів – фактично на судноплавній мапі України буквально зникають цілі річки! Вже зараз через відсутність належних глибин на річках судна вимушені ходити недозавантаженими. Саме тому сьогодні «НІБУЛОН», практично, виконуючи обов'язки держави, здійснює днопоглиблення річок за власний кошт.

Компанія «НІБУЛОН» самотужки взялася за розбудову ефективної системи логістики в Україні: відродження річок як транспортних артерій, їх днопоглиблення, будівництво сучасного флоту. Мета компанії – мінімізувати, а в подальшому і виключити в'їзд вантажівок-зерновозів у міста, і максимально переорієнтувати вантажоперевезення з автомобільних шляхів на річкові.

4 квітня 2018 року «НІБУЛОН» завершив виконання чергового етапу днопоглиблювальних робіт на Південному Бузі. Незважаючи на несприятливі погодні умови: низькі температури і вітер, днопоглиблювальні роботи виконувалися за запланованим графіком та завершилися до початку нересту, як і було обіцяно раніше. Роботи виконувалися за допомогою унікального самохідного надпотужного днопоглиблювального судна «Миколаївець», збудованого минулого року на суднобудівно-судноремонтному заводі «НІБУЛОН» для днопоглиблення Дніпра, Південного Бугу, а також акваторії портів та терміналів з метою забезпечення ефективної роботи флоту компанії.

Рисунок 1.2 – Стаття в якій ідеться про покращення екологічного стану р. Південний Буг



НІБУЛОН EN UA

Днопоглиблення тільки покращить стан Південного Бугу

Робоча група при Миколаївській облраді позитивно оцінила реалізацію інвестиційного проекту компанії «НІБУЛОН»

12 липня 2016 року філію «Вознесенська» (Миколаївська обл.) компанії «НІБУЛОН» відвідала робоча група з представників Миколаївської облради, районних та сільських рад, профільних державних органів та громадськості з метою ознайомлення з ходом виконання компанією днопоглиблювальних робіт на річці Південний Буг.

Роботи проводяться без порушень!

Сьогодні нам були надані всі необхідні документи та продемонстровано те, що компанії дійсно немає чого приховувати. Ми задоволені такою відкритою процедурою, гласністю у роботі. У такому руслі мають працювати всі, – голова Миколаївської облради Вікторія Москаленко.

Рисунок 1.3 – Ще одна стаття з того ж джерела, про екологічний стан Південного Бугу

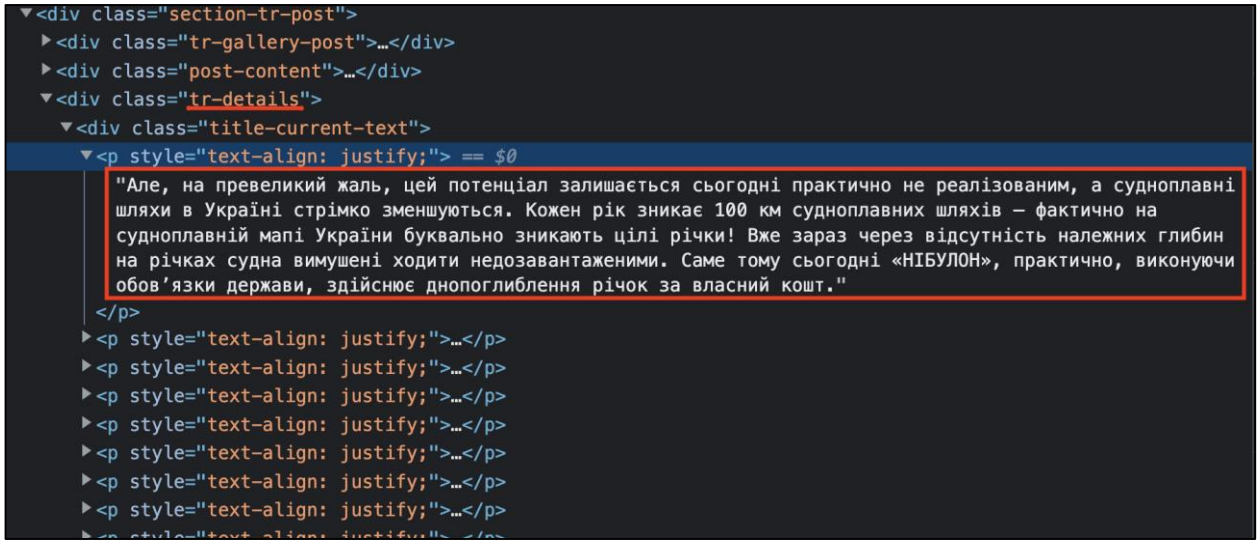


Рисунок 1.4 – Текст із першої статті знаходиться в ідентифікаторі доступу «class=tr-detail»

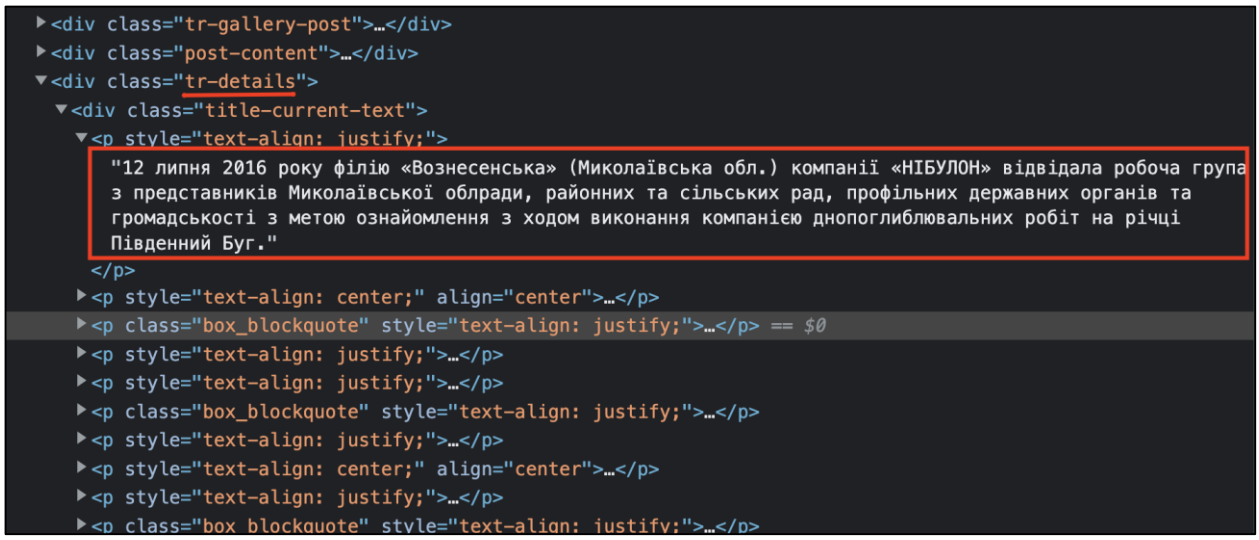


Рисунок 1.5 – Текст із другої статті, з того ж джерела, але ідентифікатор доступу ідентичний

1.2 Огляд технології NLP як методу вирішення проблеми класифікації

Люди для спілкування між собою використовують слова та текст. Такий спосіб передачі інформації один одному, називається природною мовою.

Проте комп'ютери не можуть інтерпретувати ці дані , які є природною мовою. Отримані дані є цінними та можуть дати цінну інформацію. Отже, вам

потрібні комп’ютери, щоб мати можливість розуміти, емулювати та розумно реагувати на людську мову.

Обробка природної мови або NLP відноситься до галузі штучного інтелекту, яка дає машинам можливість читати, розуміти та отримувати значення з людської мови.

NLP об’єднує галузі лінгвістики та інформатики, щоб розшифровувати мовну структуру та вказівки, а також створювати моделі, які можуть зрозуміти, розбити та відокремити важливі деталі від тексту та мови (рис. 1.6) [4].

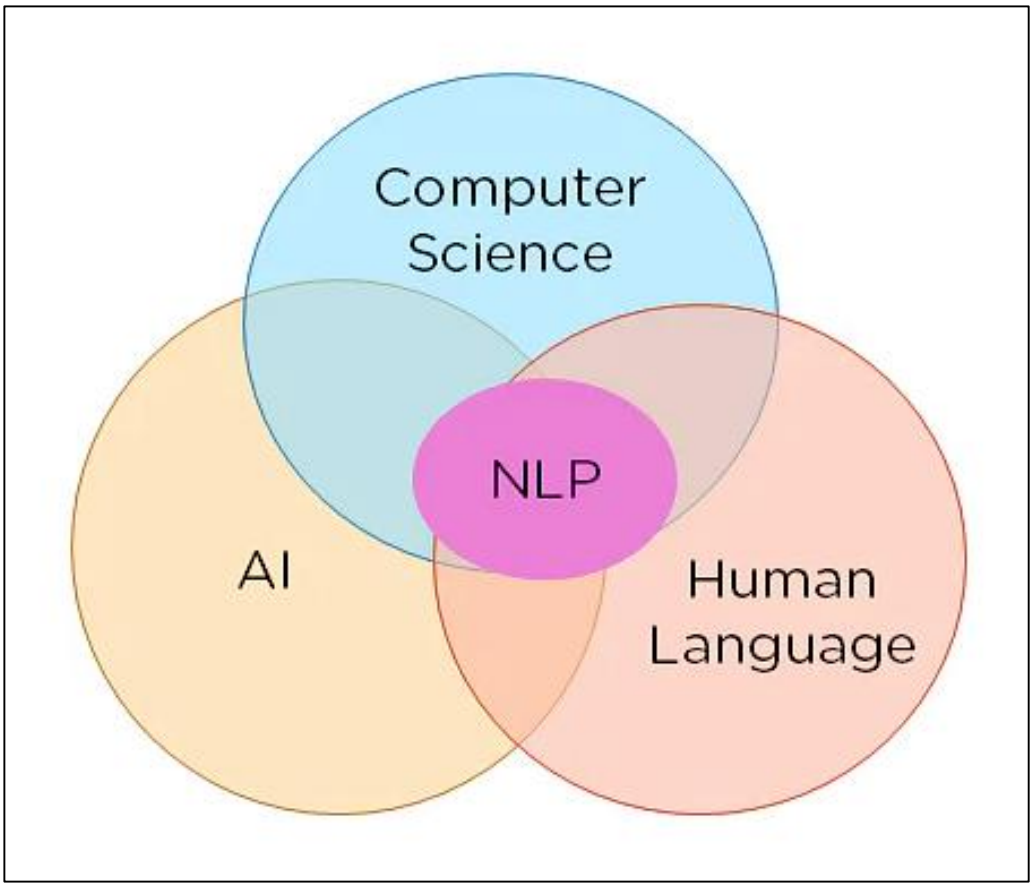


Рисунок 1.6 – Складові NLP

Етапи попередньої обробки даних у NLP включають:

– сегментацію;
– токенізацію;
– видалення стоп-слів;

– витікання;
– лематизація;
– позначення частини мови;
– позначення іменованих сутностей.

Далі трошки детальніше про кожен із них.

Сегментація. Для початку потрібно розбити весь документ на складові речення. Це можна реалізувати розділивши статтю разом із розділовими знаками, такими як крапки та коми (рис. 1.7) [4].

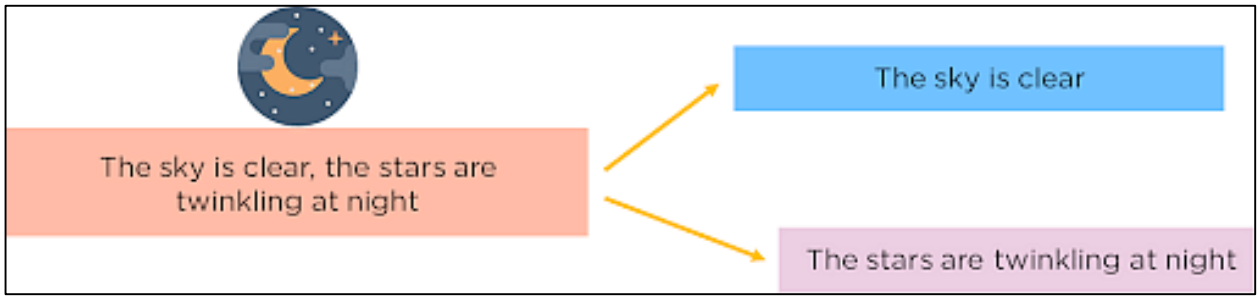


Рисунок 1.7 – Приклад сегментації

Токенізація. Щоб алгоритм зрозумів ці речення, вам потрібно отримати слова в реченні та окремо пояснити їх нашому алгоритму. Отже, ви розбиваєте своє речення на складові слова і зберігаєте їх. Це називається токенізацією, а кожне слово називається токеном (рис. 1.8) [4].

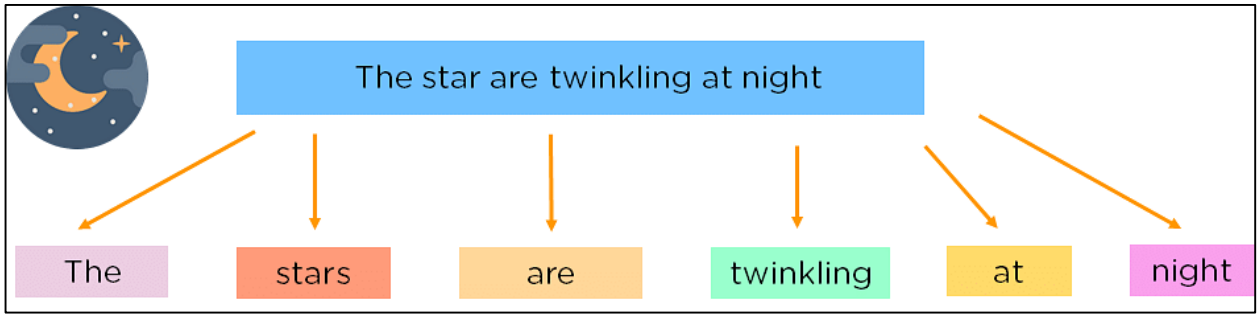


Рисунок 1.8 – Приклад токенізації речення

Видалення стоп-слів. Можна пришвидшити процес навчання, позбувшись несуттєвих слів, які додають мало сенсу нашому висловлюванню

та служать лише для того, щоб наше висловлювання звучало зв’язніше. Такі слова, як was, in, is і the, називаються стоп-словами, і їх можна видалити (рис. 1.9).

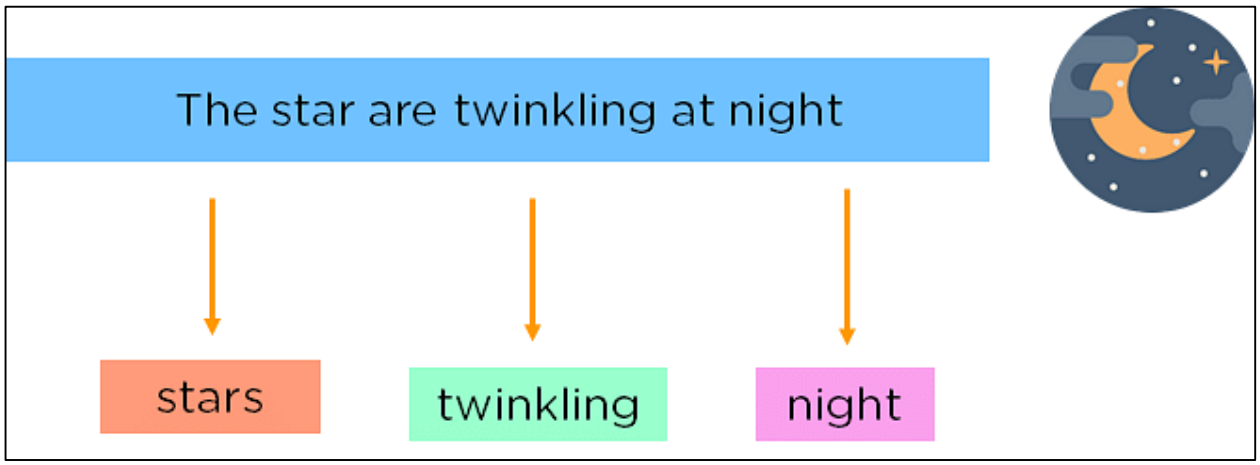


Рисунок 1.9 – Приклад видалення стоп-слів із тексту

Витікання. Це процес отримання основи слова. Основа слова дає нові слова після додавання до них афіксів (рис. 1.10) [4].

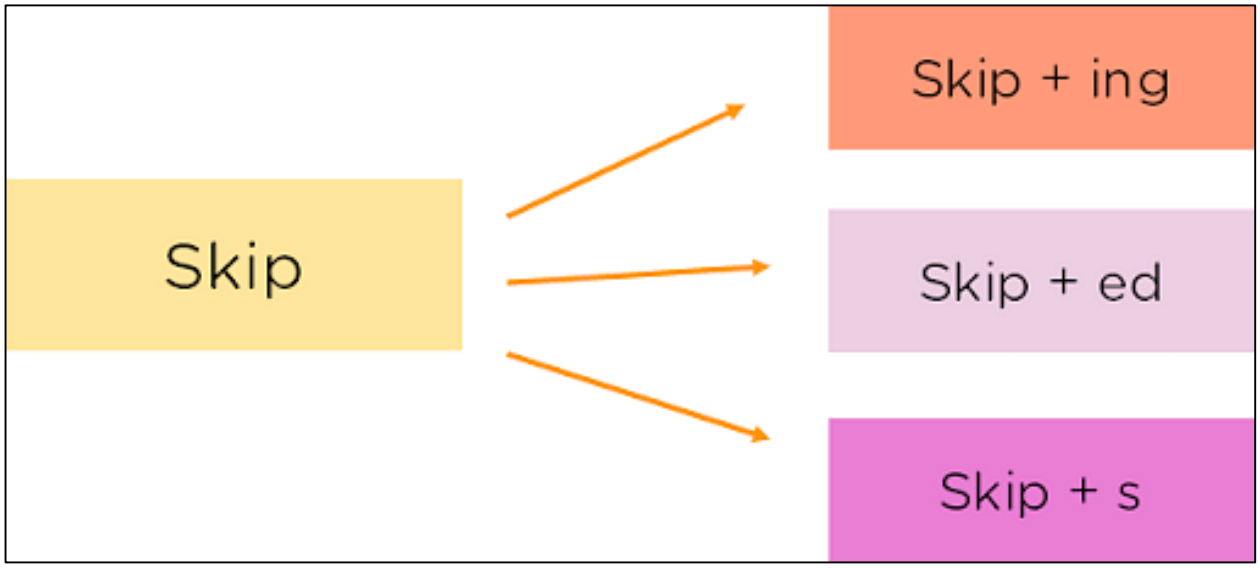


Рисунок 1.10 – Процес витікання

Лематизація. Процес утворення кореневої основи слова. Корінь Основа дає нову базову форму слова, яке присутнє в словнику і від якого це слово

походить. Ви також можете визначити базові слова для різних слів на основі часу, настрою, статі тощо (рис. 1.11).

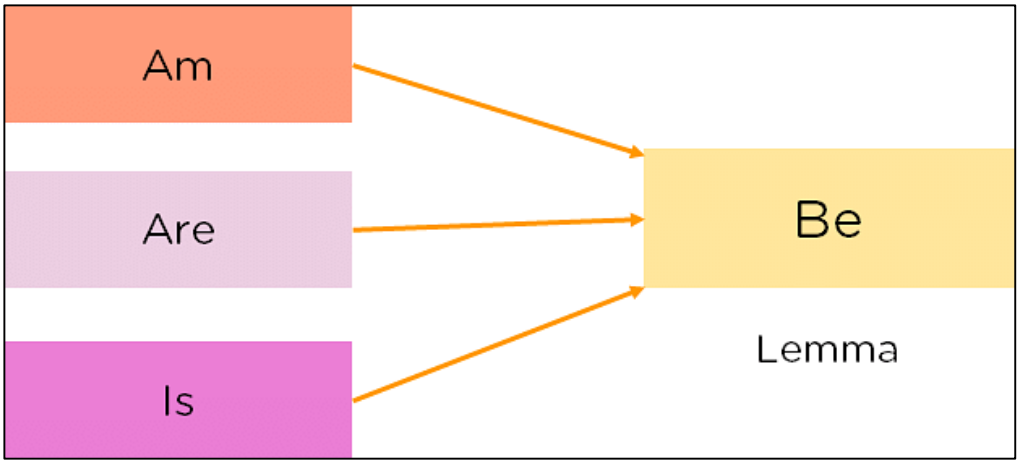


Рисунок 1.11 – Приклад лематизації

Позначення частини мови. Тепер потрібно пояснити машині концепцію іменників, дієслів, артиклів та інших частин мови, додавши ці теги до слів. Це називається «частиною» (рис. 1.12).

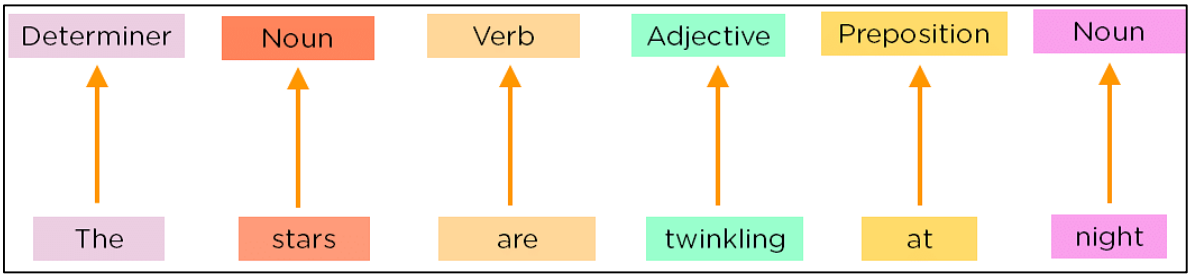


Рисунок 1.12 – Позначення частини мови

Позначення іменованих сутностей. Далі потрібно «познайомити» свою машину зі згадками про поп-культуру та повсякденними іменами, позначивши назви фільмів, важливих особистостей чи місць тощо, які можуть зустрічатися в документі. Це робиться, класифікуючи слова за підкатегоріями. Це допоможе знайти будь-які ключові слова в реченні. Підкатегорії: особа, місцезнаходження, грошова вартість, кількість, організація, фільм [4].

Однією із найвідоміших моделей машинного навчання для обробки природньої мови є BERT. У 2018 році Google розробив цей алгоритм, щоб покращити контекстне розуміння тексту без міток у широкому діапазоні завдань, навчившись передбачати текст, який може бути перед і після (двонаправленого) іншого тексту.

BERT використовується для багатьох мовних завдань:

- розпізнавання того позитивні чи негативні дані про воду;
- допомога ботам у відповідях на запитання;
- автодоповнення під час написання листа;
- розпізнавання слова, які мають багато значень, залежно від навколишнього тексту.

BERT перетворює слова в числа. Цей процес важливий, оскільки моделі машинного навчання використовують числа, а не слова як вхідні дані. Це дозволяє тренувати моделі машинного навчання на ваших текстових даних. Тобто моделі BERT використовуються для перетворення ваших текстових даних, щоб потім використовувати їх з іншими типами даних для прогнозування в моделі ML [5].

Альтернативою чи наступником BERT є RoBERTa. В першу чергу вона покращує BERT шляхом ретельної та розумної оптимізації гіперпараметрів навчання для BERT. Кілька простих і зрозумілих змін разом підвищують продуктивність Roberta та перевершують BERT у майже всіх завданнях, для вирішення яких BERT був розроблений [6].

RoBERTa буквально використовує ту ж архітектуру, що й BERT. Проте, на відміну від BERT, під час попереднього навчання він проходить попереднє навчання лише за допомогою моделювання замаскованої мови (BERT також пройшов попереднє навчання за допомогою передбачення наступного речення). Нижче наведено деякі зміни гіперпараметрів, які RoBERTa використовувала для досягнення кращої продуктивності [6]:

- триваліший час навчання та більше навчальних даних (у 10 разів від 16 ГБ до 160 ГБ);

- більший розмір пакету від 256 до 8000 і більший розмір словника від 30 до 50 Кб ;
- довші послідовності використовуються як вхідні дані, але RoBERTa все ще має максимальне обмеження маркерів у 512 маркерів, як і BERT;
- динамічне маскування дозволяє шаблону маскування бути іншим щоразу, коли послідовність подається в модель, на відміну від BERT, який використовував той самий шаблон маскування.

Ще одним із послідовників BERT є DistilBERT. DistilBERT розроблений для того щоб оптимізувати навчання шляхом зменшення розміру BERT. Distilbert на 40% менший за оригінальну базову модель BERT, на 60% швидший за неї та зберігає 97% своєї функціональності. Він використовує приблизно таку саму загальну архітектуру, як і BERT, але лише з 6 блоками кодувальника (BERT має 12). Ці блоки кодера також ініціалізуються шляхом простого взяття 1 з кожних 2 попередньо підготовлених блоків кодера BERT. Крім того, з Distilbert також видалено вбудовані функції маркерів і об'єднання BERT. На відміну від BERT, Distilbert проходить попереднє навчання лише за допомогою моделювання замаскованої мови (нагадаємо, що BERT навчався за допомогою MLM і прогнозування наступного речення).

Проаналізувавши ці моделі, їх переваги і недоліки, зроблено висновок, що найоптимальнішою моделлю в нашому випадку буде Distilbert, адже він швидший та продуктивніший, а також потребує менше ресурсів для своєї роботи [6].

1.3 Аналіз моделей машинного навчання для класифікації ембеддингів

Класифікація тексту — це процес розуміння значення неструктурованого тексту та впорядкування його в попередньо визначені

категорії (теги). Одним із найпопулярніших завдань класифікації тексту є аналіз настрою, мета якого класифікувати неструктуровані дані за настроєм.

Далі розглянемо деякі найпоширеніші моделі машинного навчання для класифікації, з якими працюватимемо в наступних розділах. Ці моделі входять в одну із найвідоміших бібліотек для роботи із машинним навчанням – «scikit-learn».

Linear Regression. Лінійна регресія – це контрольований алгоритм , який навчається моделювати залежну змінну, p, як функція деяких незалежних змінних (вона ж «особливості»), x_i, шляхом знаходження лінії (або поверхні), яка найкраще «відповідає» даним. Загалом, припускаємо p бути деяким числом і кожен x_i може бути в основному що завгодно. Наприклад: прогнозування ціни будинку за кількістю кімнат у цьому будинку (p: ціна, x_i: кількість кімнат) або прогнозування ваги на основі зростання та віку (p: вага, x_i: висота, x_2: вік).

Загалом, рівняння для лінійної регресії таке [7] :

$$p = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots + \beta_{\text{стор}} x_{\text{стор}} + \epsilon,$$

де p – залежна змінна; те, що намагаємося передбачити;

x_i – незалежні змінні: функції, які використовує наша модель для моделювання у;

β_i – коефіцієнти (так звані «ваги») нашої моделі регресії. Це основи нашої моделі. Саме їх наша модель «навчається» під час оптимізації.

ϵ – незнижувана помилка в нашій моделі. Термін, який об’єднує всі немодельовані частини наших даних [7].

Random Forests. У випадкових лісах кожне дерево в ансамблі будується із зразка, взятого із заміною (тобто початкового зразка) із навчального набору.

Крім того, під час розбиття кожного вузла під час побудови дерева найкраще розбиття визначається або з усіх вхідних функцій, або з випадкової підмножини розміром max_features [8].

Метою цих двох джерел випадковості є зменшення дисперсії оцінки лісу. Дійсно, окремі дерева рішень зазвичай виявляють високу дисперсію та

мають тенденцію до перепідбору. Впроваджена випадковість у лісах дає дерева рішень із дещо роз’єднаними помилками передбачення. Взявши середнє значення цих прогнозів, деякі помилки можна нівелювати. Випадкові ліси досягають зменшеної дисперсії шляхом комбінування різноманітних дерев, іноді ціною незначного збільшення зміщення. На практиці зменшення дисперсії часто є значним, отже, дає загальну кращу модель [8].

Logistic regression. Логістична регресія, незважаючи на свою назву, є лінійною моделлю для класифікації, а не регресії. Логістична регресія також відома в літературі як логіт-регресія, класифікація максимальної ентропії (MaxEnt) або логарифмічний лінійний класифікатор. У цій моделі ймовірності, що описують можливі результати одного випробування, моделюються за допомогою логістичної функції.

Логістична регресія реалізована в LogisticRegression. Ця реалізація може відповідати двійковій регресії, регресії один проти залишку або мультиноміальній логістичній регресії з додатковим $\ell 1, \ell$ 2або регулярізація Elastic-Net [9].

Support Vector Machines**.** Машина опорних векторів або SVM є одним із найпопулярніших алгоритмів керованого навчання, який використовується для задач класифікації та регресії. Однак, в основному, він використовується для проблем класифікації в машинному навчанні.

Мета алгоритму SVM — створити найкращу лінію або межу рішення, яка може розділити n-вимірний простір на класи, щоб можна було легко помістити нову точку даних у правильну категорію в майбутньому. Ця границя найкращого рішення називається гіперплощиною [10].

SVM вибирає крайні точки/вектори, які допомагають створити гіперплощину. Ці екстремальні випадки називають опорними векторами, а отже, алгоритм називають машиною опорних векторів. Розглянемо рисунок 1.13, на якому є дві різні категорії, які класифікуються за допомогою границі рішення або гіперплощини [10].

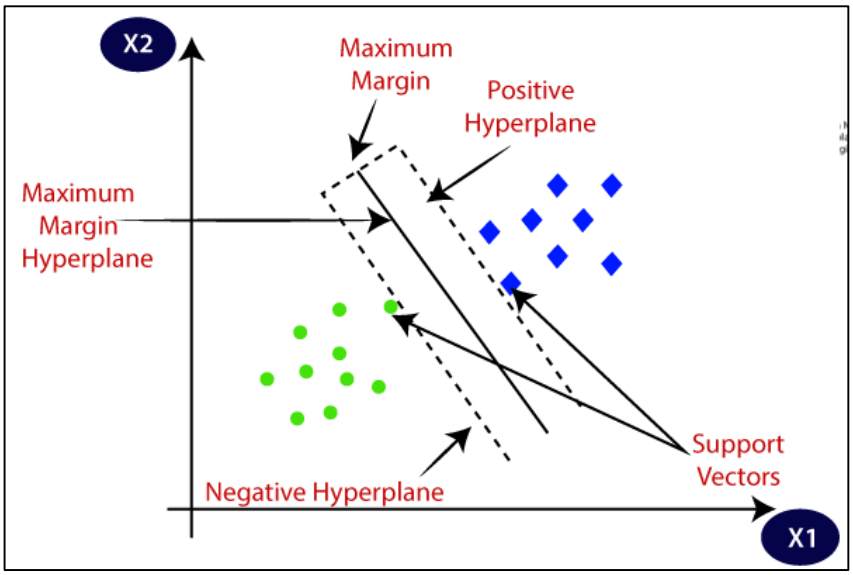


Рисунок 1.13 – Графік Support Vector Machines

LinearSVC має можливість виконувати двійкову та багатокласову класифікацію набору даних (рис. 1.14) [11].

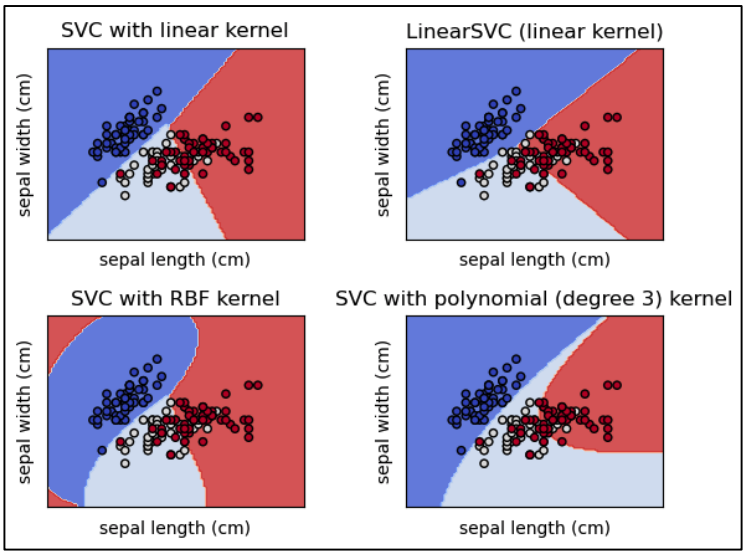


Рисунок 1.14 – Класифікація набору даних LinearSVC

LinearSVC – це ще одна (швидша) реалізація Класифікації опорних векторів для випадку лінійного ядра. Зверніть увагу, що LinearSV Спараметр не приймається kernel, оскільки він вважається лінійним. У ньому також відсутні деякі атрибути SVCand NuSVC, наприклад support_. Як і інші класифікатори, SVCi NuSV Сприймають LinearSVC як вхідні дані два масиви:

масив X форми, що містить навчальні зразки, і масив міток класів (рядків або цілих чисел) форми: (n_samples, n_features) у (n_samples) [11].

1.4 Висновки

В даному розділі розглянуто проблему автоматичної класифікації україномовної екологічної інформації. В цілому завдання магістерської роботи є актуальним для будь-яких водойм, а не лише для річок басейну Південного Бугу. Обрані сучасні методи пошуку та аналізу текстової інформації з різних джерел українською мовою, а саме – технологію обробки природної мови.

Після ознайомлення із сучасними методами обробки природної мови: BERT, RoBERTa та DistilBERT, вирішено використовувати DistilBERT, адже він швидший, більш продуктивний через зменшену вагу моделі, що задовільняє наші умови.

Проведено аналіз відомих моделей машинного навчання і вибрано 5 серед них, які найбільш придатні для класифікації ембедингів, які формує мовна NLP-модель. Всі вони мають свої переваги та недоліки, але яка із них оптимальна для вирішення поставленого завдання буде зрозуміло після перегляду та порівняння результатів їх ідентифікації.

2 ВИБІР ОПТИМАЛЬНИХ НАЛАШТУВАНЬ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Огляд датасету для тренування моделей

Для початку роботи з моделями потрібно сформувати датасет й вручну класифікувати його, щоб на його основі тренувати їх й обрати найоптимальнішу.

Для роботи використовується датасет NLP : Reports & News Classification (рис. 2.1).

	text	env_problems	pollution	treatment	climate	biomonitoring
2	У поверхневі води басейну Південного Бугу скиди стічних вод надходять із 341 точковогс	1	1			
3	Згідно з державною статистичною звітністю (форма 2-ТП-Водгосп) у 2011 р. загальний об'	1	1			
4	Показники об'ємів стічних вод серед різних галузей економіки майже однакові але домі	1	1			
5	Частка легкокиснюваних органічних речовин оцінених за БПК20 становить 84% фосфатни	1	1			
6	Функціонування промисловості призводить до скидів забруднювальних речовин у стічні	1	1			
7	Із 92 міських поселень лише у 51 населеному пункті (55%) проводиться централізований	1				
8	Населені пункти які не мають каналізаційної мережі як правило не мають і централізова	1				
9	Виробничі управління житлово-комунального господарства більшість з яких є комуналы	1		1		
10	Обладнання очисних споруд вкрай зношене кількість стічних вод часто перевищує прое	1		1		
11	Нинішній стан каналізаційних мереж також викликає велике занепокоєння оскільки чер	1		1		
12	Внаслідок цього в Україні за рік у середньому трапляється дві аварії на 1 км мережі що іс	1	1			
13	Навантаження по забрудненню внаслідок аварій на цей час оцінити неможливо.	0				
14	Ступінь підключення населення до КОС варіює у дуже широких межах — від 11 до 95%. І	1		1		
15	У великих містах з ЕН близьким або більшим 100 000 ступінь підключення до каналізацій	1		1		
16	На підставі експертного судження прийнято що середньозважений ступінь підключення	1	1	1		
17	Отримані результати наочно показують що основна частка (77%) сполук неорганічного а	1	1			
18	Переважна більшість КОС у басейні Південного Бугу обладнана спорудами для біологічн	1		1		
19	Завдяки цьому методу видаляється лише частина сполук азоту і фосфору.	1	1	1		
20	Технології для поглибленого вилучення вказаних сполук в Україні не застосовуються чер	1		1		

Рисунок 2.1 – Вигляд датасету NLP : Reports & News Classification

Джерелом інформації послугували відкриті дані щодо екологічного стану річок басейну Південного Бугу.

Речення скопійовані з відкритих текстових даних україномовних джерел у форматі PDF разом із різними спецсимволами (номерами сторінок тощо), щоб був певний «шум».

Деякі спеціальні символи були вибірково видалені, включаючи деякі коми та крапки.

Методом для створення самого датасету послугував парсер, створений при виконані бакалаврської дипломної роботи, шляхом парсингу даних з docx

файлу, автоматичної очистки від спец-символів, які заважають коректній роботі моделі машинного навчання, й на фініші виведення готових відформатованих речень до csv файлу.

2.2 Розвідувальний аналіз

Вихідними даними для роботи моделі штучного інтелекту синтезу тексту є датасет із базовими ознаками (рисунок 2.2):

– «text» – одне або кілька речень із звітів чи новин;
– «env_problems» – чи йдеться у тексті про екологічну проблему?
– «pollution» – чи текст про забруднення навколишнього середовища?
– «treatment» – текст про очисні споруди чи екологічні технології?
– «climate» – чи текст про кліматичні показники?
– «biomonitoring» – текст про біологічний, біотичний моніторинг у воді чи в басейні річки?

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У поверхневі води басейну Південного Бугу скид...	1	1	0	0	0
1	Згідно з державною статистичною звітністю (фор...	1	1	0	0	0
2	Показники об'ємів стічних вод серед різних гал...	1	1	0	0	0
3	Частка легкокиснюваних органічних речовин оцін...	1	1	0	0	0
4	Функціонування промисловості призводить до ски...	1	1	0	0	0
...	...	...	...	...	...	...
95	Ділянку в районі Мигійських порогів можна вваж...	0	0	0	0	0
96	Хімічний статус водних тіл повинен відповідати...	0	0	0	0	0
97	Використання даних величин запропоноване групо...	0	0	0	0	0
98	Оцінка хімічного статусу виділених водних тіл ...	0	0	0	0	0
99	Для визначення рівнів надійності оцінки за заг...	0	0	0	0	0

Рисунок 2.2 – Ознаки датасету «NLP : Reports & News Classification»

Проведемо розвідувальний аналіз даних по критерію забруднення (environmental problems).

Переглянемо середню довжину слова в тексті (рис. 2.3)

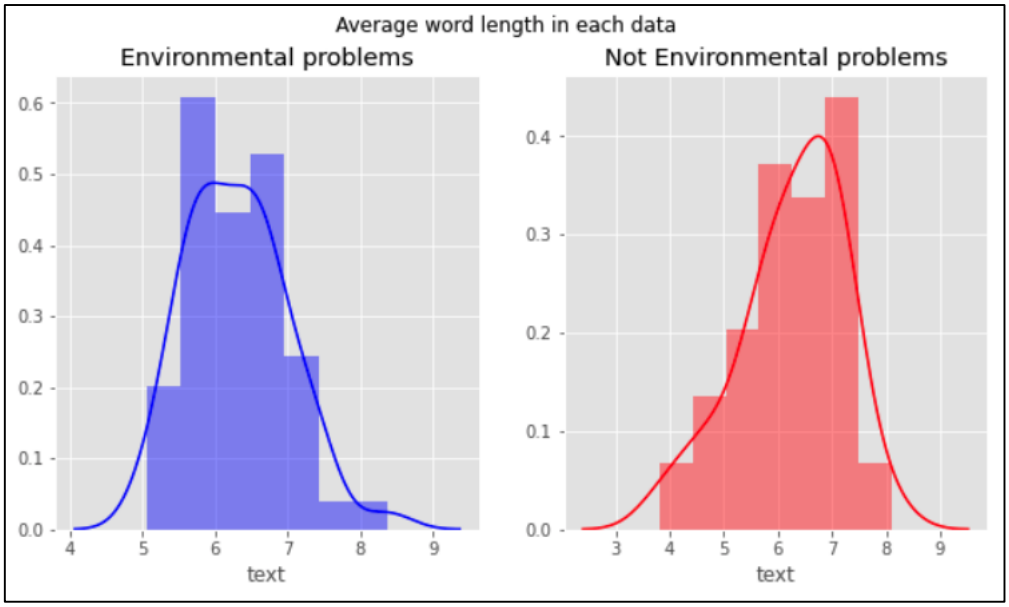


Рисунок 2.3 – Середня довжина слів

Найбільша кількість слів з 6 та 7 символами.

Далі потрібно переглянути стоп-слова в тексті та їх кількість в «Environmental problems» та «Not environmental problems» даних (рис 2.4, 2.5) [12].

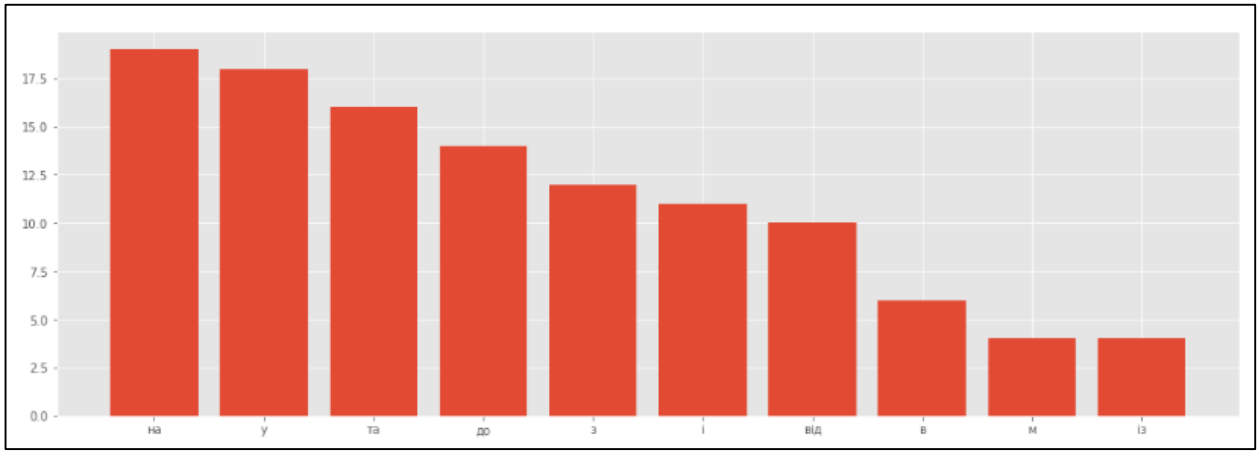


Рисунок 2.4 – Кількість стоп-слів в «Environmental problems» даних

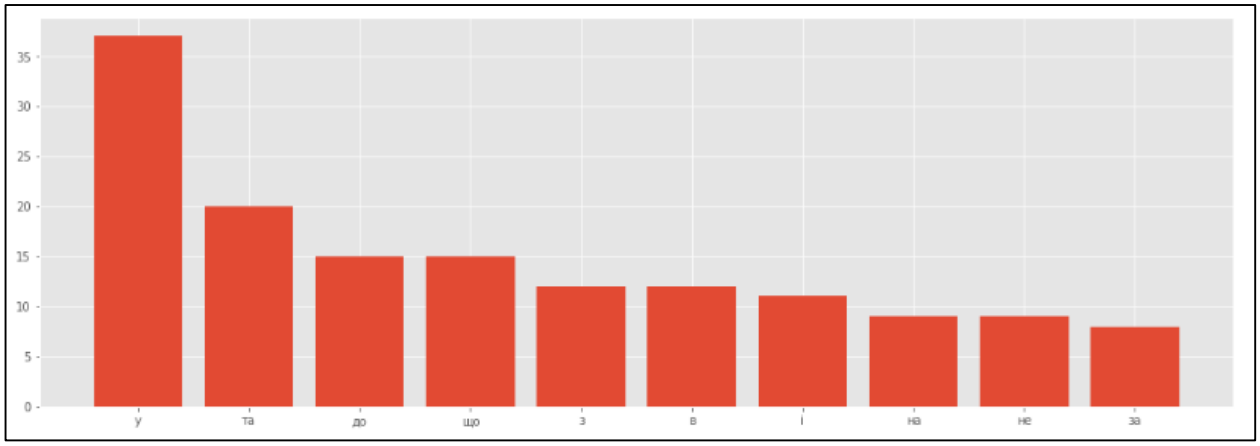


Рисунок 2.5 – Кількість стоп-слів в «Not environmental problems» даних

Серед стоп-слів найпоширенішими є «на» та «у».

Також доволі цікавим буде перегляд загальновживаних слів (рис. 2.6) [12].

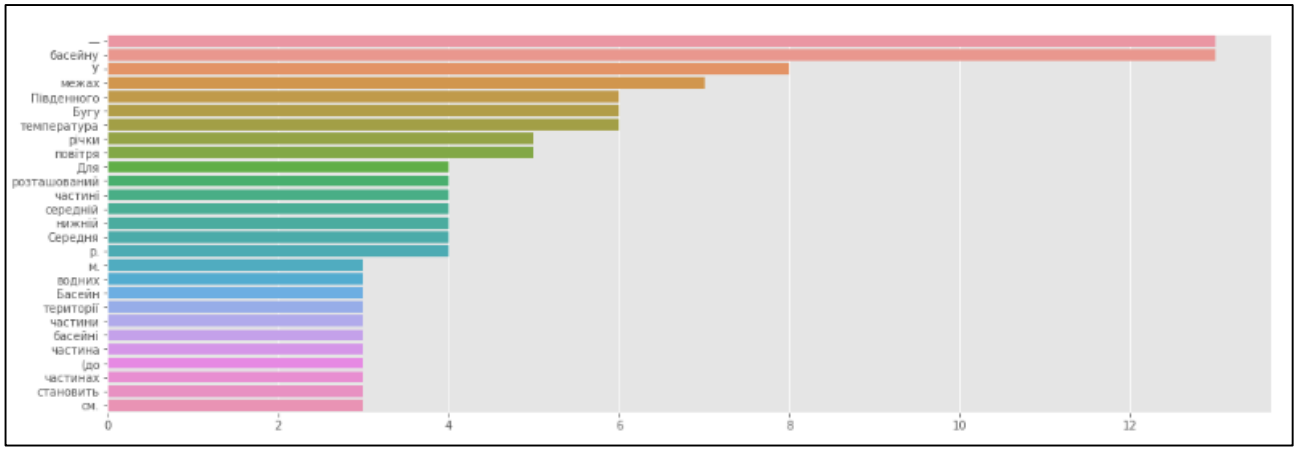


Рисунок 2.6 – Загальновживані слова

Найбільш вживане слово це «басейну», що є закономірним, адже текст про річку «Південний Буг». Проте, через те, що одним із найпоширеніших слів є слово «у» цей текст потребує очищення.

Наступним кроком є створення хмари слів із значеннями під кадром даних категорії, загальновживані в обох типах («Environmental problems» та «Not environmental problems») (рис. 2.7, 2.8) [12].

Рисунок 2.7 – Хмара слів у кластері «Environmental problems»

Рисунок 2.8 – Хмара слів у кластері «Not environmental problems»

Проведемо візуалізацію вкладень (рис 2.9) [12].

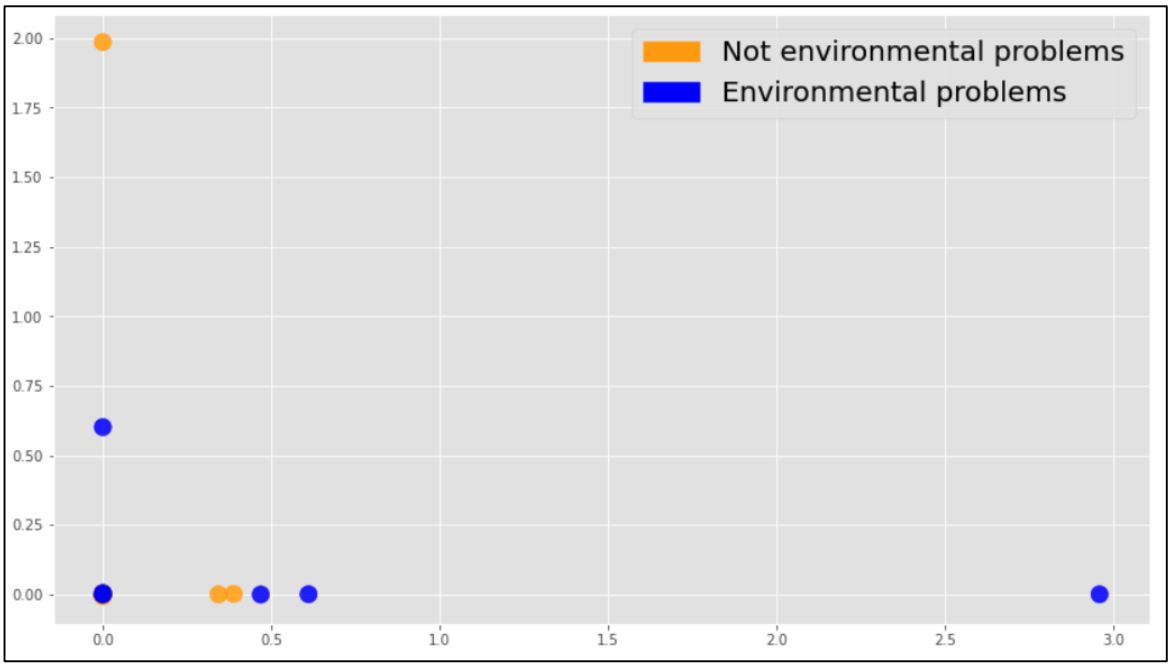


Рисунок 2.9 – Візуалізація вкладень

Ці вкладення виглядають не дуже чітко розділеними. Далі подивимося, чи зможемо ще встановити на них корисну модель (рис. 2.10) [12].

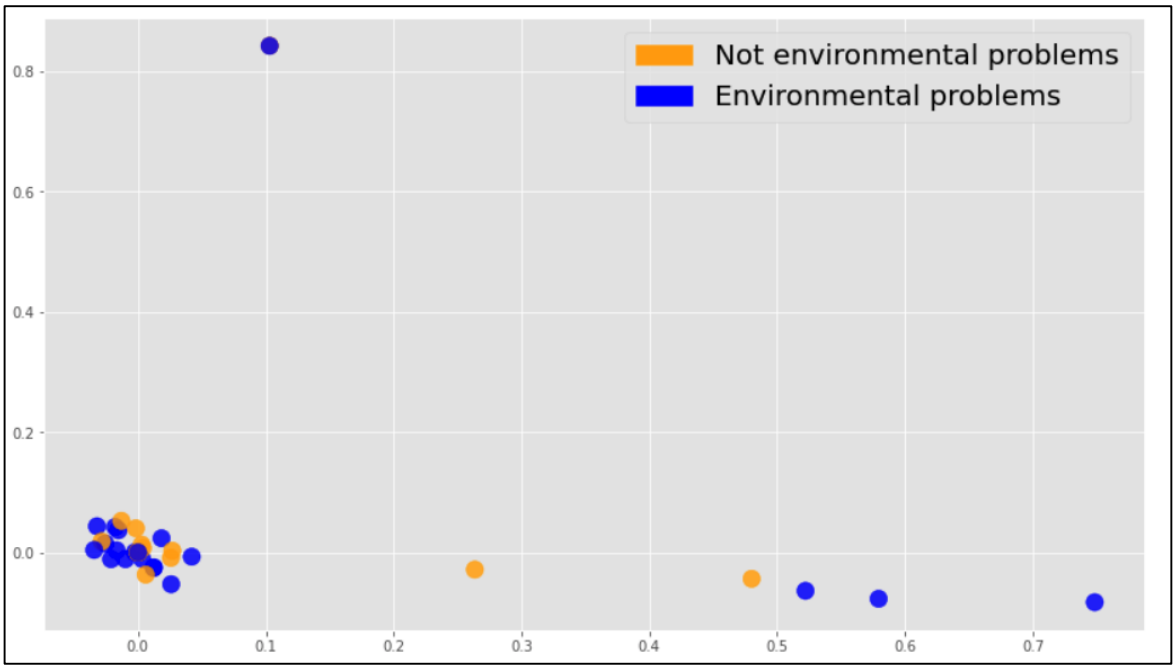


Рисунок 2.10 – Встановлення корисної моделі

2.3 Створення передтренованих моделей для класифікації даних

Для використання інформаційної технології, яка розробляється, потрібні передтреновані моделі-класифікатори. Їх слід робити за алгоритмом, наведеним на рисунку 2.11.

Рисунок 2.11 – Алгоритм розробки моделі класифікаторів

Відповідно алгоритму, першим етапом є створення та розмітка датасету, цей процес був описаний в підрозділі 2.1, саме цей датасет буде використовуватись. Наступним етапом є формування ембедингів, за допомогою методу обробки природньої мови DistilBERT, оскільки саме його обрано при аналізі огляду технологій для вирішення проблеми в підрозділі 1.2. І фінальним кроком на цьому етапі буде тренування моделей, які були

відібрані в підрозділі 1.3 та збирання їх результатів, на основі яких в кінці оберемо найкращі для подальшого використання на довільних датасетах.

Після усіх етапів описаних вище, приступаємо до тренування кожної моделі й їх подальшого порівняння. Для цього спочатку імпортуємо бібліотеки (рис. 2.12) [13].

```
import numpy as np
import pandas as pd

import matplotlib
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score

# models
from sklearn.linear_model import LinearRegression, Perceptron, RidgeClassifier, SGDClassifier, LassoCV
from sklearn.svm import SVC, LinearSVC, SVR
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, ExtraTreesClassifier

from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.gaussian_process import GaussianProcessClassifier

from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn import metrics
import xgboost as xgb
import lightgbm as lgb

# NN models
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras import optimizers
from keras.wrappers.scikit_learn import KerasClassifier

import torch
import transformers as ppb

import warnings
warnings.filterwarnings('ignore')
```

Рисунок 2.12 – Імпортуємо необхідні бібліотеки для роботи

Завантажуємо підготовлений датасет в ноутбук (рис. 2.13).

```
df = pd.read_csv('../input/nlp-reports-news-classification/water_problem_nlp_ua_for_Kaggle_100.csv', delimiter=';',
                 header=0, encoding='cp1251')
df = df.fillna(0)

convert_dict = {'text': str,
                'env_problems': int,
                'pollution': int,
                'treatment': int,
                'climate': int,
                'biomonitoring': int}

df = df.astype(convert_dict)
df
```

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У поверхневі води басейну Південного Бугу скид...	1	1	0	0	0
1	Згідно з державною статистичною звітністю (фор...	1	1	0	0	0
2	Показники об'ємів стічних вод серед різних гал...	1	1	0	0	0
3	Частка легкокиснюваних органічних речовин оцін...	1	1	0	0	0
4	Функціонування промисловості призводить до ски...	1	1	0	0	0
...	...	...	...	...	...	...
95	Ділянку в районі Мигійських порогів можна вваж...	0	0	0	0	0
96	Хімічний статус водних тіл повинен відповідати...	0	0	0	0	0
97	Використання даних величин запропоноване групо...	0	0	0	0	0
98	Оцінка хімічного статусу виділених водних тіл ...	0	0	0	0	0
99	Для визначення рівнів надійності оцінки за заг...	0	0	0	0	0

100 rows × 6 columns

Рисунок 2.13 – Завантаження датасету

Переглядаємо інформацію про завантажений датасет (рис. 2.14).

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   text            100 non-null    object
 1   env_problems    100 non-null    int64 
 2   pollution       100 non-null    int64 
 3   treatment       100 non-null    int64 
 4   climate         100 non-null    int64 
 5   biomonitoring   100 non-null    int64 
dtypes: int64(5), object(1)
memory usage: 4.8+ KB
```

Рисунок 2.14 – Інформація про завантажений датасет

Далі завантажимо попередньо треновану модель DistilBERT для подальшої роботи з нею (рис. 2.15) [13].

```
# For pre-trained DistilBERT:
model_class, tokenizer_class, pretrained_weights = (ppb.DistilBertModel, ppb.DistilBertTokenizer, 'distilbert-base-multilingual-cased')

# Other models: https://huggingface.co/transformers/pretrained_models.html

# Load pretrained model/tokenizer
tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)
```

Рисунок 2.15 – Завантаження DistilBERT

Наступним кроком буде токенізація речень, тобто зведемо дані до необхідного для BERT вигляду (рис. 2.16) [13].

```
# Tokenization the sentences - break them up into word and subwords in the format BERT is comfortable with
tokenized = df['text'].apply((lambda x: tokenizer.encode(x, add_special_tokens=True)))

max_len = 0
for i in tokenized.values:
    if len(i) > max_len:
        max_len = len(i)

padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])
np.array(padded).shape
```

```
(100, 142)
```

+ Code + Markdown

```
# Creation variable to ignore (mask) the data padding
attention_mask = np.where(padded != 0, 1, 0)
print(attention_mask.shape)
attention_mask
```

```
(100, 142)
array([[1, 1, 1, ..., 0, 0, 0],
       [1, 1, 1, ..., 1, 1, 1],
       [1, 1, 1, ..., 0, 0, 0],
       ...,
       [1, 1, 1, ..., 0, 0, 0],
       [1, 1, 1, ..., 0, 0, 0],
       [1, 1, 1, ..., 0, 0, 0]])
```

Рисунок 2.16 – Токенізація речень

Далі на основі токенізованих даних і попередньо тренованої моделі DistilBERT створюємо модель, необхідну для вирішення нашого завдання (рис. 2.17) [13].

```
# Modeling
input_ids = torch.tensor(padded).to(torch.int64)
attention_mask = torch.tensor(attention_mask).to(torch.int64)

with torch.no_grad():
    last_hidden_states = model(input_ids, attention_mask=attention_mask)
```

```
# Last hidden states
features = last_hidden_states[0][:,0,:].numpy()
```

Рисунок 2.17 – Створення моделі DistilBERT

Задаємо параметри для налаштування роботи обраних моделей й зберігаємо середнє й максимальне значення результатів роботи. (рис. 2.18, 2.19) [13].

```
# Set parameters of models
models = pd.DataFrame(columns = ['name', 'model', 'param_grid'])

# Linear Regression
n = 0
models.loc[n, 'name'] = 'Linear Regression'
models.at[n, 'model'] = LinearRegression()
models.at[n, 'param_grid'] = {}

# Random Forest Classifier
n = 1
models.loc[n, 'name'] = 'Random Forest Classifier'
models.at[n, 'model'] = RandomForestClassifier()
models.at[n, 'param_grid'] = {'n_estimators': [40, 50, 60, 100, 500],
                              'min_samples_split': [30, 40, 50, 100, 200],
                              'min_samples_leaf': [10, 12, 15, 20, 50],
                              'max_features': ['auto'],
                              'max_depth': [3, 4, 5, 6],
                              'criterion': ['gini'],
                              'bootstrap': [False]
                             }

# Logistic Regression
n = 2
models.loc[n, 'name'] = 'Logistic Regression'
models.at[n, 'model'] = LogisticRegression()
models.at[n, 'param_grid'] = {'C': np.linspace(0.0001, 100, 20)
                             }

# Support Vector Machines
n = 3
models.loc[n, 'name'] = 'Support Vector Machines'
models.at[n, 'model'] = SVC()
models.at[n, 'param_grid'] = {'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
                              'tol': [1e-3]
                             }
```

Рисунок 2.18 – Параметри моделей

```
# Linear SVC
n = 4
models.loc[n, 'name'] = 'Linear SVC'
models.at[n, 'model'] = LinearSVC()
models.at[n, 'param_grid'] = {'dual':[False],
                              'C': np.linspace(1, 15, 15)
                             }

# Avg values
models.loc[5, 'name'] = 'Mean values'

# Max values
models.loc[6, 'name'] = 'Max values'
```

Рисунок 2.19 – Параметри моделей (Продовження)

Після того, як необхідні параметри задано, маємо результат, поданий на (рис. 2.20) [13].

	name	model	param_grid
0	Linear Regression	LinearRegression()	{}
1	Random Forest Classifier	RandomForestClassifier()	{'n_estimators': [40, 50, 60, 100, 500], 'min_...
2	Logistic Regression	LogisticRegression()	{'C': [0.0001, 5.263252631578947, 10.526405263...
3	Support Vector Machines	SVC()	{'kernel': ['linear', 'poly', 'rbf', 'sigmoid'...
4	Linear SVC	LinearSVC()	{'dual': [False], 'C': [1.0, 2.0, 3.0, 4.0, 5....
5	Mean values	NaN	NaN
6	Max values	NaN	NaN

Рисунок 2.20 – Параметри моделей

Далі створимо функцію для тренування обраних моделей та обчислення отриманих результатів на підготовлених даних і розділимо датасет для тюнінгу, тобто для тренування моделей (рис. 2.21-2.23) [13].

```python
def model_prediction(models, features, labels, test_size, verbose):
    # Models training and data prediction for all models from DataFrame models

    # Splitting train data for model tuning with cross-validation
    #cv_train = ShuffleSplit(n_splits=cv_n_split, test_size=0.2, random_state=random_state)
    train_features, test_features, train_labels, test_labels = train_test_split(features,
                                                                                labels,
                                                                                test_size=test_size,
                                                                                random_state=random_state)
    # Total list of prediction by all models
    total_train_pred = []
    total_test_pred = []

    # Results
    results = models[['name']].copy()
    results['acc_train'] = results['acc_test'] = 0

    for i in range(len(models)-2):
        # Training
        model = GridSearchCV(models.at[i, 'model'], models.at[i, 'param_grid'])
        model.fit(train_features, train_labels)

        # Prediction
        train_pred = model.predict(train_features).round(0).astype('int')
        total_train_pred.append(train_pred)
        test_pred = model.predict(test_features).round(0).astype('int')
        total_test_pred.append(test_pred)

        # Scoring
        acc_train = accuracy_score(train_labels, train_pred)
        acc_test = accuracy_score(test_labels, test_pred)
```

Рисунок 2.21 – Функція для тренування моделей – тюнінг

```python
        if verbose:
            print(f"Model - {models.loc[i, 'name']}")
            print("Best parameters:", model.best_params_)
            print("Accuracy for training data: %0.3f" % acc_train)
            print("Accuracy for test data: %0.3f" % acc_test)
            print('\n')

        # Save results
        results.loc[i,'acc_train'] = acc_train
        results.loc[i,'acc_test'] = acc_test

    # MEAN values
    # Calc prediction
    mean_train_pred = (np.mean(np.array(total_train_pred), axis=0)).astype(int)
    mean_test_pred = (np.mean(np.array(total_test_pred), axis=0)).astype(int)

    # Scoring
    acc_train = accuracy_score(train_labels, mean_train_pred)
    acc_test = accuracy_score(test_labels, mean_test_pred)
    if verbose:
        print(f"Mean prediction values")
        print("Accuracy for training data: %0.3f" % acc_train)
        print("Accuracy for test data: %0.3f" % acc_test)

    # Save results
    n = len(results)-2
    results.loc[n,'acc_train'] = acc_train
    results.loc[n,'acc_test'] = acc_test

    # MAX values
    # Calc prediction
    max_train_pred = (np.max(np.array(total_train_pred), axis=0))
    max_test_pred = (np.max(np.array(total_test_pred), axis=0))
```

Рисунок 2.22 – Функція для тренування моделей – середні значення

```
    # MAX values
    # Calc prediction
    max_train_pred = (np.max(np.array(total_train_pred), axis=0))
    max_test_pred = (np.max(np.array(total_test_pred), axis=0))

    # Scoring
    acc_train = accuracy_score(train_labels, max_train_pred)
    acc_test = accuracy_score(test_labels, max_test_pred)
    if verbose:
        print(f"Maximum prediction values")
        print("Accuracy for training data: %0.3f" % acc_train)
        print("Accuracy for test data: %0.3f" % acc_test)

    # Save results
    n = len(results)-1
    results.loc[n,'acc_train'] = acc_train
    results.loc[n,'acc_test'] = acc_test

    return results
```

Рисунок 2.23 – Функція для тренування моделей – максимальні значення

Далі потрібно написати функцію для передбачення класів даних по заданих таргетах з подальшим збереженням та відображенням результатів статистики (рис 2.24) [13].

```
def target_prediction(models, df, features, target, test_size=0.2, verbose=True):
    # Text classification model and prediction for given feature "target" (with labels) in df

    # Target
    labels = df[target]

    # Extracting the number of examples of each class
    Relevant_len = df[df[target] == 1].shape[0]
    Not_len = df[df[target] == 0].shape[0]

    # Draw bar plot
    plt.rcParams['figure.figsize'] = (7, 5)
    plt.bar(10, Relevant_len, 3, label="Relevant", color='green')
    plt.bar(15, Not_len, 3, label="Not", color='red')
    plt.legend(loc='upper center')
    plt.ylabel('Number of examples')
    plt.title('Propertion of examples for ' + target)
    plt.show()

    # Models training, prediction and save results
    results = model_prediction(models, features, labels, test_size, verbose=verbose)
    results = results.sort_values(by=['acc_test', 'acc_train'], ascending=False)
    results.to_csv(f'{target}-models-scoring.csv', index=False)
    display(results)
```

Рисунок 2.24 – Функція для передбачення по таргетах

Потім викликаємо створену функцію, передаємо необхідні параметри, підготовлені на попередніх етапах. Результатом виклику функції, будуть статистичні дані роботи моделі по окремих таргетах (рис 2.25) [13].

```
%%time
# Solving NLP Classification tasks
print('Solving NLP Classification tasks')
for col in cols:
    target_prediction(models, df, features, col, test_size=0.2, verbose=True)
```

```
Solving NLP Classification tasks
```

Рисунок 2.25 – Виклик функції тренування моделі

Результат роботи моделей по різних таргетах (рис. 2.26-2.30).

	name	acc_train	acc_test
0	Linear Regression	1.000	0.85
2	Logistic Regression	1.000	0.85
3	Support Vector Machines	1.000	0.85
4	Linear SVC	1.000	0.85
6	Max values	0.925	0.85
5	Mean values	1.000	0.80
1	Random Forest Classifier	0.925	0.80

Рисунок 2.26 – Результат роботи моделі по таргету «env_problems»

	name	acc_train	acc_test
5	Mean values	0.9875	0.80
1	Random Forest Classifier	0.9625	0.80
0	Linear Regression	1.0000	0.75
2	Logistic Regression	1.0000	0.70
3	Support Vector Machines	1.0000	0.70
4	Linear SVC	1.0000	0.70
6	Max values	0.9750	0.70

Рисунок 2.27 – Результат роботи моделі по таргету «pollution»

	name	acc_train	acc_test
0	Linear Regression	1.00	0.80
1	Random Forest Classifier	0.85	0.75
2	Logistic Regression	0.85	0.75
3	Support Vector Machines	0.85	0.75
5	Mean values	0.85	0.75
4	Linear SVC	1.00	0.65
6	Max values	1.00	0.65

Рисунок 2.28 – Результат роботи моделі по таргету «treatment»

	name	acc_train	acc_test
0	Linear Regression	1.0000	1.00
2	Logistic Regression	1.0000	1.00
3	Support Vector Machines	1.0000	1.00
4	Linear SVC	1.0000	1.00
6	Max values	1.0000	1.00
1	Random Forest Classifier	0.9625	0.95
5	Mean values	0.9625	0.95

Рисунок 2.29 – Результат роботи моделі по таргету «climate»

	name	acc_train	acc_test
0	Linear Regression	1.0000	0.95
4	Linear SVC	1.0000	0.95
6	Max values	1.0000	0.95
1	Random Forest Classifier	0.9125	0.95
2	Logistic Regression	0.9125	0.95
3	Support Vector Machines	0.9125	0.95
5	Mean values	0.9125	0.95

Рисунок 2.30 – Результат роботи моделі по таргету «biomonitoring»

Після, того, як моделі відпрацювали і результати їх роботи додано до таблиці, можемо зробити порівняльний аналіз, для того, щоб обрати найоптимальнішу.

Як видно на рисунках 2.26-2.30, оптимальною є лінійна регресія, що означає, що дані сильно зашумлені і це заважає працювати більш потужним моделям. Як правило, модель лінійної регресії є кращою, коли вона оверфітиться під дані. Отже, пропонується як оптимальні використовувати дві:

- Лінійна регресія;
- Випадкові ліси (Random Forest).

2.4 Висновки

В даному розділі розглянуто побудову датасету, його розвідувальний аналіз даних та застосування мовних NLP-моделей і технологій для вирішення поставленого завдання.

Проведено підготовку тренувального датасету, який був класифікований вручну, для того, щоб він був максимально результативний і моделі були гарно навченими. Здійснено розвідувальний аналіз для того, щоб можна було робити висновки про те, на скільки створений датасет відповідає заданим вимогам. Потім, описано алгоритм розробки моделі класифікаторів. Сформовано ембеддинги за допомогою методу обробки природньої мови DistilBERT, оскільки саме його обрано при аналізі огляду технологій для вирішення проблеми, після чого відбулось тренування моделей, які були відібрані в підрозділі 1.3, та проведено аналіз результатів їх роботи.

Результатом цього розділу є статистика по роботі кожної з 5 моделей машинного навчання, які розглянуто в першому розділі. Провівши аналіз результатів, встановлено, що моделі лінійної регресії та випадкових лісів (Random Forest) показали найкращий результат, а саме – точність 0.80 та 0.95

відповідно, тому й їх вибрано як найкращі для класифікації в наступному розділі.

3 СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ КЛАСИФІКАЦІЇ УКРАЇНОМОВНОЇ ІНФОРМАЦІЇ ПРО МАСИВИ ВОД БАСЕЙНУ РІЧКИ ПІВДЕННИЙ БУГ

3.1 Алгоритм програмного забезпечення інформаційної технології

Розв’язання поставленої задачі пропонується здійснювати у 2 етапи:

Етап 1. Налаштування інформаційної технології та синтез оптимальних передтренованих моделей-класифікаторів (див. розд. 2.11).

Етап 2. Застосування оптимальних передтренованих моделей-класифікаторів на заданих даних.

Загальний алгоритм розробленої інформаційної технології на етапі 2 (рис. 3.1).

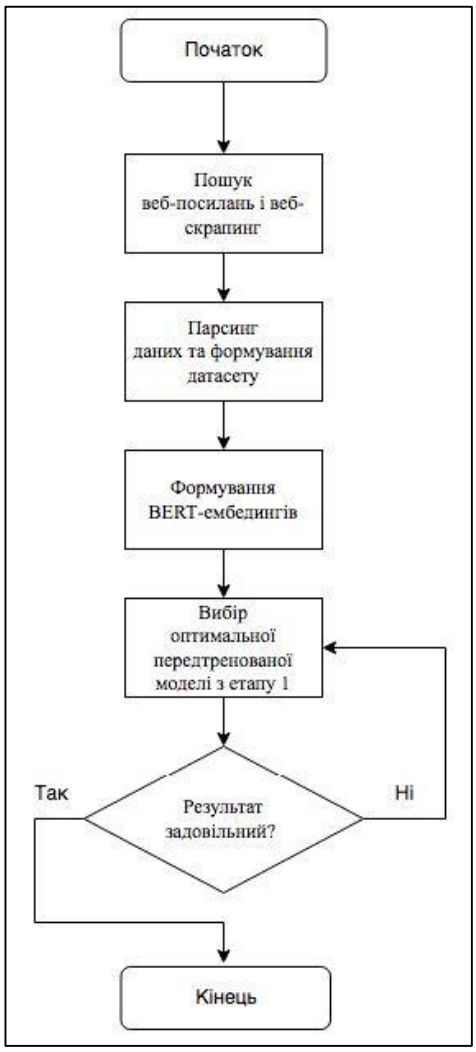


Рисунок 3.1 – Алгоритм інформаційної технології

Алгоритм схожий до першого етапу своїми першими пунктами, адже тут також готуємо датасет для роботи з ним, але в цьому випадку не продовжимо ручної класифікації та інших операцій, а лише збираємо його з джерел та пропускаємо через парсер, щоб отримати csv файл для роботи з моделями-класифікаторами. Після підготовки датасету, розпочинаємо тестування відібраних моделей, які вже треновані на попередньому датасеті з підрозділу 2.1, і якщо результат задовільний, то закінчуємо програму, якщо ж ні – повертаємось до вибору моделі й беремо іншу.

3.2 Формування датасету

Для того, щоб сформувати датасет потрібно знайти веб-посилання ресурси (рис. 3.2) [14], в яких є інформація про екологічні проблеми річок і водойм басейну р. Південний Буг з геоприв'язкою (забруднення води, ґрунту, загибель риби, засуха тощо) або шляхи їх вирішення (прибирання прибережної території, видалення зайвого мулу з дна річки, зариблення, встановлення очисних споруд тощо) і дістати необхідні дані. Для цього можна вручну заповнити форму (рис. 3.3).

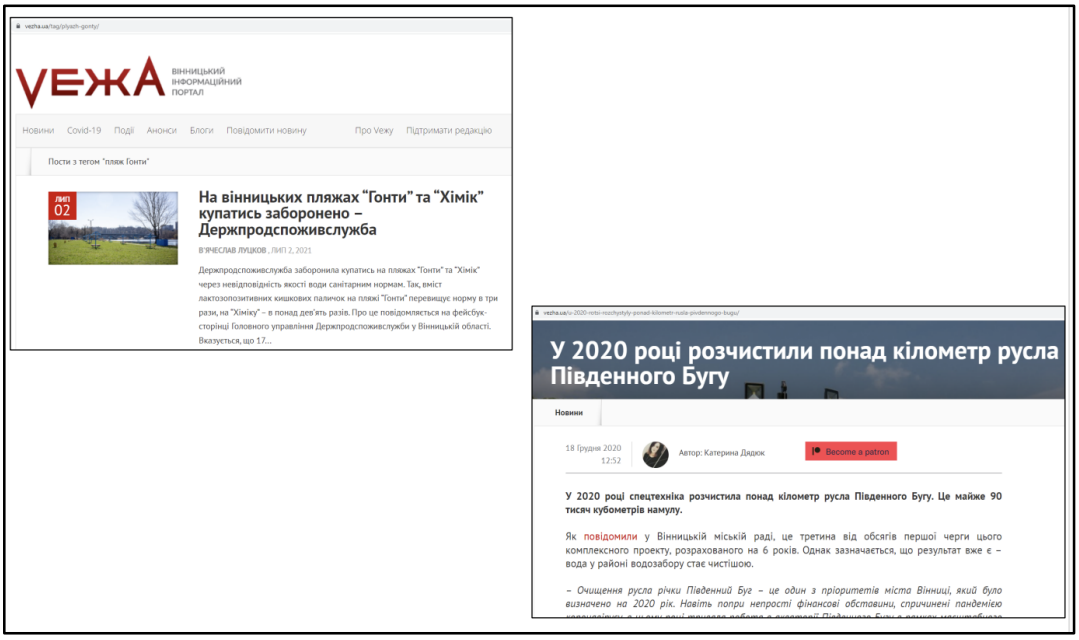


Рисунок 3.2 – Приклад веб-ресурсів, з яких можна витягнути дані

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У басейні Південного Бугу налічується 6582 мал...	0	0	0	0	0
1	Південний Буг бере початок на Волино-Подільськ...	0	0	0	0	0
2	У Кіровоградській, Одеській та Миколаївській о...	0	0	0	0	0
3	Дніпро-Бузький лиман є мілководною затокою Чор...	0	0	0	0	0
4	За своїми розмірами Дніпро-Бузький лиман є най...	0	0	0	0	0
5	Дніпро-Бузький лиман інколи розглядають як дві...	0	0	0	0	0
6	Характерною особливістю басейну Південного Буг...	0	0	0	0	0
7	Ставки. В басейні розташовано 9877 ставків, за...	0	0	0	0	0
8	Водосховища. В басейні розташовано 187 водосхо...	0	0	0	0	0
9	Для водопостачання обласного центру побудовани...	0	0	0	0	0
10	Водогоном довжиною 130 км вода подається з р. ...	0	0	0	0	0

Рисунок 3.3 – Сформований датасет про екологічні проблеми річок і водойм басейну р. Південний Буг

Іншим же варіантом видобування даних із веб-ресурсів є автоматичний веб-скрапинг (видобування матеріалів з веб-ресурсів) і парсинг (пошук конкретно тексту і його розбирання на окремі текстові речення). Для цього можемо використати, наприклад Python-ноутбук «Web scraping from HTML - BUWR SB site-parser» [15] (рис. 3.4).

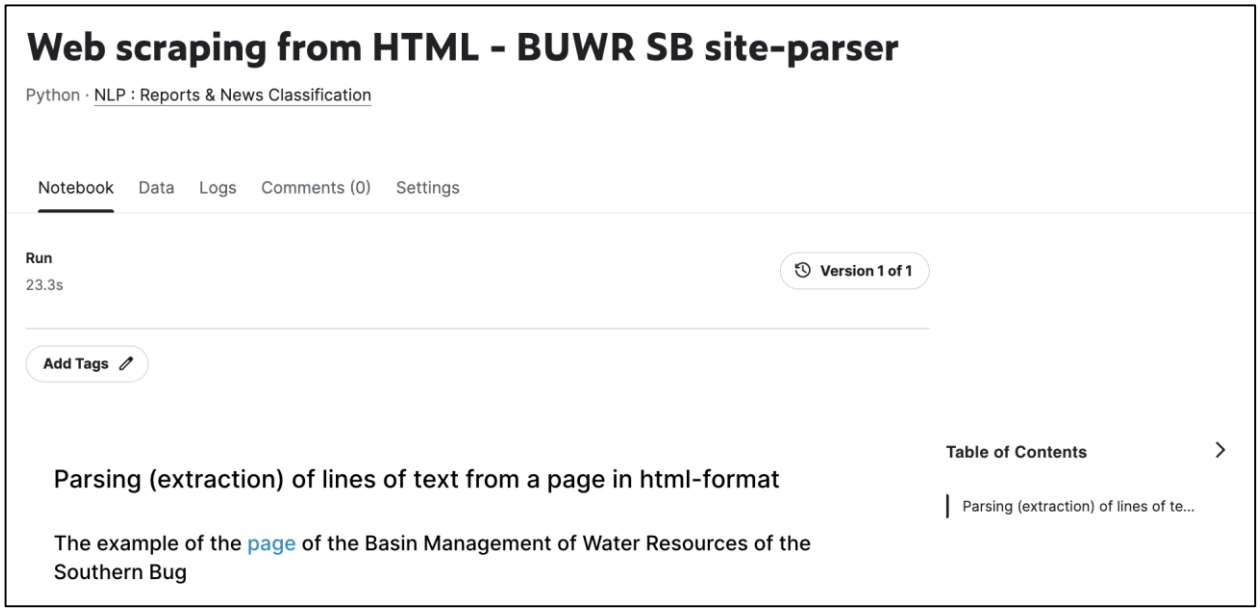


Рисунок 3.4 – Ноутбук по веб-скрапінгу

Для роботи ноутбуку потрібно імпортувати бібліотеки (рис. 3.5) [15].

```
import numpy as np
import pandas as pd
import requests
from bs4 import BeautifulSoup as bs
```

Рисунок 3.5 – Імпорт бібліотек

3.3 Застосування технології до даних БУВР Південного Бугу щодо гідрографічної мережі

Застосовуємо вище наведені операції до веб-сторінки «Гідрографічна мережа» сайту БУВР Південного Буг [16].

Перед скрапингом текст має наступний вигляд (рис. 3.6).

```
</p>
<p style="text-align: justify;">
 <span style="font-size: 12pt;">
  Дніпро-Бузький лиман інколи розглядають як дві водойми – Дніпро
вський та Бузький, але чіткої межі між ними немає. Якщо межі дніпровсько
ї частини лиману є доволі чіткими, то верхня межа бузької частини є диск
утивною. Найчастіше цю межу проводять по Варварівському мосту, що знаход
иться в м. Миколаїв. Довжина Бузького лиману 47 км, ширина – 11 км, площ
а 162 км
  <sup>
   2
  </sup>
  . Ступінь мінералізації лиману – солонуватий.
 </span>
</p>
<p align="center" style="text-align: center;">
 <span style="font-size: 12pt;">
  <strong>
   Штучні водойми
  </strong>
 </span>
</p>
<p style="text-align: justify;">
 <span style="font-size: 12pt;">
  Характерною особливістю басейну Південного Бугу, що виділяє йог
о з поміж інших великих річок є дуже велика його зарегульованість. В бас
ейні створено майже 10 тисяч штучних водойм, сумарний їх об’єм понад 1,5
```

Рисунок 3.6 – Формат, в якому знаходиться текст у веб-джерелах

Далі за допомогою бібліотеки Beautiful Soup дістанемо дані (рис. 3.7-3.9) [15].

```
# Get the main content with text
ad = soup.find('div', itemprop ="articleBody")
```

Рисунок 3.7 – Знаходження головного тега, в якому розміщена потрібна інформація

```
# Get text
text_all = ad.getText()
text_all
```

Рисунок 3.8 – Використавши результати попереднього коду дістаємо текст

```
['Гідрографічна мережа басейну річки Південний Буг ',
 'Річки',
 'У басейні Південного Бугу налічується 6582 малих річок, загальною до
вжиною близько 20 тис. км, 11 середніх річок загальною довжиною понад
1,6 тис. км та 1 велика річка Південний Буг.',
 'Південний Буг бере початок на Волино-Подільській височині поблизу с.
Холодець Хмельницької області і впадає в Дніпро-Бузький лиман Чорного
моря. Довжина річки становить 806 км, загальне падіння 321 м, середній
схил  водної поверхні 0,4 %.',
 ' Озера',
 'У Кіровоградській, Одеській та Миколаївській областях зустрічаються
невеликі озера. Їх загальна площа складає 190 га. ',
 'Моря',
 'Південний Буг належить до басейну Чорного моря. ',
 'Лимани',
 'Дніпро-Бузький лиман є мілководною затокою Чорного моря, що відокрем
лений від нього Кінбурнською косою. Сполучається лиман з морем Кінбурн
ською протокою, через яку і здійснюється водообмін. На сході лиман меж
```

Рисунок 3.9 – Текст, який дістали шляхом скрапингу

Далі потрібно привести текст до необхідного формату, а саме замінити символи «\xa0» на пробіл, та розділити текст на окремі рядки після символів «\n» (рис. 3.10, 3.11) [15].

```
# Data postprocessing
# - replace '\xa0' to ' '
# - split text to separately lines after symbols '\n'
text_res = text_all[1:].replace('\xa0', ' ').split
('\n')
text_res
```

Рисунок 3.10 – Код для приведення тексту до необхідного формату

```
['Гідрографічна мережа басейну річки Південний Буг ',
 'Річки',
 'У басейні Південного Бугу налічується 6582 малих річок, загальною до
вжиною близько 20 тис. км, 11 середніх річок загальною довжиною понад
1,6 тис. км та 1 велика річка Південний Буг.',
 'Південний Буг бере початок на Волино-Подільській височині поблизу с.
Холодець Хмельницької області і впадає в Дніпро-Бузький лиман Чорного
моря. Довжина річки становить 806 км, загальне падіння 321 м, середній
схил  водної поверхні 0,4 %.',
 ' Озера',
 'У Кіровоградській, Одеській та Миколаївській областях зустрічаються
невеликі озера. Їх загальна площа складає 190 га. ',
 'Моря',
 'Південний Буг належить до басейну Чорного моря. ',
 'Лимани',
 'Дніпро-Бузький лиман є мілководною затокою Чорного моря, що відокрем
лений від нього Кінбурнською косою. Сполучається лиман з морем Кінбурн
ською протокою, через яку і здійснюється водообмін. На сході лиман меж
ує з гирлом Дніпра, на півночі поступово переходить у р. Південний Бу
г.',
 'За своїми розмірами Дніпро-Бузький лиман є найбільшим на Чорному мор
і. Його площа складає 800 км2, об’єм води – 3 км3. Лиман є мілководни
м, його середня глибина становить 3,5-4 м.',
 'Дніпро-Бузький лиман інколи розглядають як дві водойми – Дніпровськи
й та Бузький, але чіткої межі між ними немає. Якщо межі дніпровської ч
```

Рисунок 3.11 – Відформатований текст

Наступним кроком буде видалення речень, довжина яких, менша 50 символів (рис. 3.12, 3.13) [15].

```
# Remove short lines (less 50 symbols)
text_res = [x for x in text_res if len(x) > 50]
text_res
```

Рисунок 3.12 – Код для видалення коротких речень

```
['У басейні Південного Бугу налічується 6582 малих р
ічок, загальною довжиною близько 20 тис. км, 11 сере
дніх річок загальною довжиною понад 1,6 тис. км та 1
велика річка Південний Буг.',
 'Південний Буг бере початок на Волино-Подільській в
исочині поблизу с.Холодець Хмельницької області і вп
адає в Дніпро-Бузький лиман Чорного моря. Довжина рі
чки становить 806 км, загальне падіння 321 м, середн
ій схил  водної поверхні 0,4 %.',
 'У Кіровоградській, Одеській та Миколаївській облас
тях зустрічаються невеликі озера. Їх загальна площа
складає 190 га. ',
 'Дніпро-Бузький лиман є мілководною затокою Чорного
моря, що відокремлений від нього Кінбурнською косою.
Сполучається лиман з морем Кінбурнською протокою, че
рез яку і здійснюється водообмін. На сході лиман меж
ує з гирлом Дніпра, на півночі поступово переходить
у р. Південний Буг.',
```

Рисунок 3.13 – Текст після форматування

Далі текст потрібно записати у форматі DataFrame (це – двовимірна структура даних, як двовимірний масив або таблиця з рядками та стовпцями), вказавши при цьому назви стовбців, та присвоївши їм всім значення 0 (як базове значення, яке в подальшому буде уточнене з використання запропонованої технології), та зберегти це все у форматі .csv (рис. 3.14-3.16) [15].

```
# Save result list of lines as dataset with zero target values
res = pd.DataFrame(columns = ['text', 'env_problems', 'pollution', 'treat
ment', 'climate', 'biomonitoring'])
res['text'] = text_res
res[['env_problems', 'pollution', 'treatment', 'climate', 'biomonitorin
g']] = 0
res
```

Рисунок 3.14 – Перетворення тексту до формату DataFrame

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У басейні Південного Бугу налічується 6582 мал...	0	0	0	0	0
1	Південний Буг бере початок на Волино-Подільськ...	0	0	0	0	0
2	У Кіровоградській, Одеській та Миколаївській о...	0	0	0	0	0
3	Дніпро-Бузький лиман є мілководною затокою Чор...	0	0	0	0	0
4	За своїми розмірами Дніпро-Бузький лиман є най...	0	0	0	0	0
5	Дніпро-Бузький лиман інколи розглядають як дві...	0	0	0	0	0
6	Характерною особливістю басейну Південного Буг...	0	0	0	0	0
7	Ставки. В басейні розташовано 9877 ставків, за...	0	0	0	0	0
8	Водосховища. В басейні розташовано 187 водосхо...	0	0	0	0	0

Рисунок 3.15 – Текст у форматі DataFrame

```
# Save rezult as csv-file
res.to_csv('BUWR-SB-basin-water-resources.csv', index=False)
```

Рисунок 3.16 – Збереження тексту до файлу формату .csv

Після створення датасету застосовуємо BERT та передтреновані моделі, вибрані у підрозділі 2.3: лінійну регресію та випадкові ліси (рис. 3.17, 3.18) [15].

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У басейні Південного Бугу налічується 6582 малих річок, загальною довжиною близько 20 тис. км, 11 середніх річок загальною довжиною понад 1,6 тис. км та 1 велика річка Південний Буг.	1	1	1	1	0
1	Південний Буг бере початок на Волино-Подільській височині поблизу с.Холодець Хмельницької області і впадає в Дніпро-Бузький лиман Чорного моря. Довжина річки становить 806 км, загальне падіння 321 м, середній схил водної поверхні 0,4 ‰.	1	1	0	1	0
2	У Кіровоградській, Одеській та Миколаївській областях зустрічаються невеликі озера. Їх загальна площа складає 190 га.	0	0	1	0	1
3	Дніпро-Бузький лиман є мілководною затокою Чорного моря, що відокремлений від нього Кінбурнською косою. Сполучається лиман з морем Кінбурнською протокою, через яку і здійснюється водообмін. На сході лиман межує з гирлом Дніпра, на півночі поступово переходить у р. Південний Буг.	1	1	0	1	0
4	За своїми розмірами Дніпро-Бузький лиман є найбільшим на Чорному морі. Його площа складає 800 км2, об'єм води – 3 км3. Лиман є мілководним, його середня глибина становить 3,5-4 м.	0	0	1	1	0
5	Дніпро-Бузький лиман інколи розглядають як дві водойми – Дніпровський та Бузький, але чіткої межі між ними немає. Якщо межі дніпровської частини лиману є доволі чіткими, то верхня межа бузької частини є дискутивною. Найчастіше цю межу проводять по Варварівському мосту, що знаходиться в м. Миколаїв. Довжина Бузького лиману 47 км, ширина – 11 км, площа 162 км2. Ступінь мінералізації лиману – солонуватий.	1	1	1	1	1
6	Характерною особливістю басейну Південного Бугу, що виділяє його з поміж інших великих річок є дуже велика його зарегульованість. В басейні створено майже 10 тисяч штучних водойм, сумарний їх об'єм понад 1,5 км3, що практично дорівнює стоку в маловодний рік 95% забезпеченості.	1	1	1	0	0
7	Ставки. В басейні розташовано 9877 ставків, загальною площею понад 56,4 тис. га та сумарним об'ємом 644 млн.м3. Внаслідок проведеної у 2013 році в областях басейну інвентаризації водних об'єктів, загальна кількість ставків порівняно із 2012 роком збільшилася на 122 шт. Найбільша кількість ставків побудована у Вінницькій (3443), Черкаській (2206) та Кіровоградській (2188) областях.	1	1	1	0	0
8	Водосховища. В басейні розташовано 187 водосховищ місткістю 894 млн.м3, загальною площею водного дзеркала майже 30 тис. га. На самому руслі р.Південний Буг побудовано 16 водосховищ місткістю 316 млн. м3, які використовуються для потреб гідроенергетики, водопостачання та рекреації. Найбільша кількість водосховищ побудована в Кіровоградській (64) і Вінницькій (42) областях.	1	1	1	0	0
9	Для водопостачання обласного центру побудований водовід із басейну Дніпра с.Чернилівка – м. Хмельницький довжиною 34 км пропускною здатністю 0,8 м3/с. За 2013 рік подано 20 млн.м3.	1	1	1	1	1
10	Водогоном довжиною 130 км вода подається з р. Рось (басейн Дніпра) до м.Умань Черкаської області, пропускною здатністю 0,14 м3/с. За 2013 рік подано 3,9 млн.м3.	1	1	1	1	1

Рисунок 3.17 – Класифікація за допомогою моделі лінійної регресії

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У басейні Південного Бугу налічується 6582 малих річок, загальною довжиною близько 20 тис. км, 11 середніх річок загальною довжиною понад 1,6 тис. км та 1 велика річка Південний Буг.	0	0	0	0	0
1	Південний Буг бере початок на Волино-Подільській височині поблизу с.Холодець Хмельницької області і впадає в Дніпро-Бузький лиман Чорного моря. Довжина річки становить 806 км, загальне падіння 321 м, середній схил водної поверхні 0,4 ‰.	0	0	0	0	0
2	У Кіровоградській, Одеській та Миколаївській областях зустрічаються невеликі озера. Їх загальна площа складає 190 га.	0	0	0	0	0
3	Дніпро-Бузький лиман є мілководною затокою Чорного моря, що відокремлений від нього Кінбурнською косою. Сполучається лиман з морем Кінбурнською протокою, через яку і здійснюється водообмін. На сході лиман межує з гирлом Дніпра, на півночі поступово переходить у р. Південний Буг.	0	0	0	0	0
4	За своїми розмірами Дніпро-Бузький лиман є найбільшим на Чорному морі. Його площа складає 800 км2, об'єм води – 3 км3. Лиман є мілководним, його середня глибина становить 3,5-4 м.	0	0	0	0	0
5	Дніпро-Бузький лиман інколи розглядають як дві водойми – Дніпровський та Бузький, але чіткої межі між ними немає. Якщо межі дніпровської частини лиману є доволі чіткими, то верхня межа бузької частини є дискутивною. Найчастіше цю межу проводять по Варварівському мосту, що знаходиться в м. Миколаїв. Довжина Бузького лиману 47 км, ширина – 11 км, площа 162 км2. Ступінь мінералізації лиману – солонуватий.	1	0	0	0	0

Рисунок 3.18 – Класифікація за допомогою моделі випадкові ліси (Random Forest)

Із рисунків наведених вище, можемо зробити висновок, що модель випадкові ліси (Random Forest) дає більш адекватний результат.

3.4 Застосування технології до аналізу звіту БУВР Південного Бугу

На сайті БУВР Південного Бугу доступний річний звіт за 2019 р.

Завантажуємо його. Застосовуємо усі вище наведені прийоми і технології. Формуємо датасет. Далі потрібно імпортувати необхідні для роботи бібліотеки (рис. 3.19).

```
import numpy as np
import pandas as pd
import pickle

import matplotlib
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# models
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics

# NLP
import torch
import transformers as ppb

import warnings
warnings.filterwarnings('ignore')
```

Рисунок 3.19 – Імпортовані бібліотеки

Після цього потрібно задати модель, з якою будемо працювати (рис. 3.20).

```
# Model choice
model_name = "random-forest"
# model_name = "linear-regression"
```

Рисунок 3.20 – Вибір моделі

Далі необхідно, в залежності від назви моделі, обрати модель з ноутбука (рис. 3.21).

```
# Get notebook_models name
if model_name=="random-forest":
    notebook_models = "nlp-for-ua-bert-cls-with-rf"
elif model_name=="linear-regression":
    notebook_models = "nlp-for-ua-bert-cls-with-linreg"
```

Рисунок 3.21 – Вибір моделі з ноутбуку

Наступним кроком буде встановлення параметрів для імпорту датасету (рис. 3.22).

```
# Set parameters
random_state = 0
notebook_data = "buwrsbbasinwaterdataset"
data_name = "BUWR-SB-basin-water-resources-st.csv"
```

Рисунок 3.22 – Встановлення параметрів

Завантажимо датасет по заданих параметрах (рис. 3.23, 3.24).

```
df = pd.read_csv(f'/kaggle/input/{notebook_data}/{data_name}', encoding='cp1251')
df = df.fillna(0)

convert_dict = {'text': str,
                'env_problems': int,
                'pollution': int,
                'treatment': int,
                'climate': int,
                'biomonitoring': int}

df = df.astype(convert_dict)
df
```

Рисунок 3.23 – Завантаження датасету

	text	env_problems	pollution	treatment	climate	biomonitoring
0	І. Загальна характеристика району річкового ба...	0	0	0	0	0
1	Її річковий басейн площею 63700 км2 межує з рі...	0	0	0	0	0
2	Витік річки знаходиться на Подільській височин...	0	0	0	0	0
3	Річковий басейн Південного Бугу розміщений на ...	0	0	0	0	0
4	Невеликі частини річкового басейну розташовані...	0	0	0	0	0
...	...	...	...	...	...	...
755	Для забезпечення населення питною водою яка ві...	0	0	0	0	0
756	Основними проблемами скидання неочищених зворо...	0	0	0	0	0
757	В Кіровоградській області наявна сировинна баз...	0	0	0	0	0
758	Відвали шахтних порід скиди шахтних вод викиди...	0	0	0	0	0
759	Серед підприємств що відносяться до еколого не...	0	0	0	0	0

760 rows × 6 columns

Рисунок 3.24 – Загальний вигляд датасету

Для того, щоб отримати детальну інформацію про датасет, використаємо наступний код (рис. 3.25).

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 760 entries, 0 to 759
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   text           760 non-null    object
 1   env_problems   760 non-null    int64 
 2   pollution      760 non-null    int64 
 3   treatment      760 non-null    int64 
 4   climate        760 non-null    int64 
 5   biomonitoring  760 non-null    int64 
dtypes: int64(5), object(1)
memory usage: 35.8+ KB
```

Рисунок 3.25 – Код для отримання інформації про датасет

Далі завантажимо попередньо треновану модель DistilBERT для подальшої роботи з нею, наступним кроком буде токенізація речень, тобто зведемо дані до необхідного для BERT вигляду, далі на основі токенізованих даних і попередньо тренованої моделі DistilBERT створюємо модель необхідної для вирішення нашого завдання (рис. 3.26).

```
def BERT_modeling(df, text='text'):
    # Using DistilBERT for the dataframe df[text]

    model_class, tokenizer_class, pretrained_weights = (ppb.DistilBertModel, ppb.DistilBertTokenizer, 'distilbert-base-multilingual-cased')
    # Other models: https://huggingface.co/transformers/pretrained_models.html

    # Load pretrained model/tokenizer
    tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
    model = model_class.from_pretrained(pretrained_weights)

    # Tokenization the sentences - break them up into word and subwords in the format BERT is comfortable with
    tokenized = df[text].apply((lambda x: tokenizer.encode(x, add_special_tokens=True)))

    max_len = 0
    for i in tokenized.values:
        if len(i) > max_len:
            max_len = len(i)

    padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])

    # Creation variable to ignore (mask) the data padding
    attention_mask = np.where(padded != 0, 1, 0)

    # Modeling
    input_ids = torch.tensor(padded).to(torch.int64)
    attention_mask = torch.tensor(attention_mask).to(torch.int64)

    with torch.no_grad():
        last_hidden_states = model(input_ids, attention_mask=attention_mask)

    # Last hidden states
    features = last_hidden_states[0][:,0,:].numpy()

    return features

features = BERT_modeling(df, 'text')
```

Рисунок 3.26 – Робота із BERT

Далі викликаємо функцію для класифікації даних за заданими таргетами (рис. 3.27).

```
def model_prediction(df, test_features, target):
    # Models training and data prediction for all models from DataFrame models

    # Target
    labels = df[target]

    # Model loading
    with open(f'/kaggle/input/{notebook_models}/model-{model_name}-{target}.pkl','rb') as f:
        model = pickle.load(f)

    # Prediction
    test_pred = model.predict(test_features)
    test_pred = [int(x>0) for x in test_pred]

    return test_pred
```

Рисунок 3.27 – Класифікація даних по таргетах

Задаємо список таргетів для роботи (рис. 3.28).

```
# List of the target features in df
cols = df.columns.tolist()[1:]
print('Target columns:', cols)
```

```
Target columns: ['env_problems', 'pollution', 'treatment', 'climate', 'biomonitoring']
```

Рисунок 3.28 – Таргети для роботи

Запуск функції для класифікації речень із датасету (3.29).

```
# Solving NLP Classification tasks
print('Solving NLP Classification tasks')
for col in cols:
    df[col] = model_prediction(df, features, col)
```

Рисунок 3.29 – Класифікація речень

Після класифікації отримуємо опрацьований датасет (3.30, 3.31).

```
# Results display
pd.set_option('max_colwidth',1000)
df.to_csv(f'{notebook_data}-{notebook_models}-classified.csv', index=False)
print(f"The result of classification using the model {model_name}")
display(df)
```

```
The result of classification using the model random-forest
```

Рисунок 3.30 – Код для візуалізації результату класифікації

	text	env_problems	pollution	treatment	climate	biomonitoring
0	І. Загальна характеристика району річкового басейну Південного Бугу Опис річкового басейну Південного Бугу Річка Південний Буг належить до числа великих річок басейну Чорного моря і є найбільшою річковий басейн якої повністю розташований у межах України.	1	0	0	0	0
1	Її річковий басейн площею 63700 км2 межує з річковими басейнами Дністра (на заході) та Дніпра (на півночі та сході).	0	0	0	0	0
2	Витік річки знаходиться на Подільській височині поблизу с. Холодець Хмельницької області.	0	0	0	0	0
3	Річковий басейн Південного Бугу розміщений на території семи областей України найбільші частини площі якого припадають на Вінницьку (257%) Кіровоградську (242%) Миколаївську (232%) і Черкаську (132%).	0	0	0	0	0
4	Невеликі частини річкового басейну розташовані у межах Одеської Хмільницької та Київської областей (карта 1).	0	0	0	0	0
...	...	...	...	...	...	...
755	Для забезпечення населення питною водою яка відповідає стандартам необхідно передбачити державну підтримку у фінансуванні робіт по заміні (реконструкції) діючих систем водопостачання впровадження в сільській місцевості високоефективних низько затратних модульних універсальних очисних систем для природних стічних і зворотних вод.	1	1	0	0	0
756	Основними проблемами скидання неочищених зворотних вод у природні водні об'єкти є неефективність роботи існуючих очисних споруд їх зношеність значна енергоємність при експлуатації моральна застарілість відсутність коштів (в основному комунальні підприємства) на будівництво нових очисних споруд.	1	1	0	0	0
757	В Кіровоградській області наявна сировинна база атомної енергетики та уранодобувної промисловості обумовлює радіаційну ситуацію.	1	0	0	0	0
758	Відвали шахтних порід скиди шахтних вод викиди в атмосферне повітря вміщують природні радіонукліди уранового ряду з чим пов'язаний радіаційний вплив на довкілля водні ресурси та населення.	1	1	0	0	0
759	Серед підприємств що відносяться до еколого небезпечних об'єктів є ДП «Східний гірничо-збагачувальний комбінат» Інгульська Новокостянтинівська та Смолінська шахти) ТОВ «Побужський феронікелевий комбінат» та Обласне комунальне виробниче підприємство «Дніпро Кіровоград».	1	1	0	0	0

760 rows × 6 columns

Рисунок 3.31 – Класифікований датасет з використанням моделі випадкових лісів

Як видно на рис. 3.31, результат є задовільним, отже створена інформаційна технологія довела свою ефективність.

3.5 Висновки

В першому підрозділі представлено архітектуру інформаційної технології та описано алгоритм її роботи.

У другому підрозділі описано процес формування датасету із веб-ресурсів, в яких є інформація про екологічні проблеми річок і водойм басейну р. Південний Буг з геоприв'язкою шляхом веб-скрапингу.

У третьому підрозділі реалізовано класифікацію тестового датасету за допомогою моделей лінійної регресії та випадкових лісів. Як результат, маємо,

що модель випадкових лісів (Random Forest) краще впоралась із поставленим завданням.

Після класифікації на тестовому датасеті, реалізовано класифікацію даних із доступного річного звіту за 2019 р з сайту БУВР Південного Бугу. Класифікація відбувалась за допомогою передтренованої моделі машинного навчання, яка показала кращий результат в підрозділі 3.3, тобто - за допомогою моделі випадкових лісів (Random Forest). Ця модель показала гарні результати, як на навчальному датасеті, так і на тестовому, що означає, що інформаційна технологія виконує своє завдання успішно

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку – інформаційної NLP-технології класифікації україномовної інформації про стан водних ресурсів. Особливістю програми є підвищення точності класифікації текстової україномовної інформації про масиви вод басейну річки Південний Буг. Подібні технології вже існують, але – для англомовних текстів та інших задач. Зокрема, немає адаптацій до задачі класифікації інформації україномовних джерел про масиви вод басейну річки Південний Буг. Розроблювана технологія матиме високий показник точності автоматичної класифікації.

Аналогом може бути MailChannels, ціна 708 $/рік підписка (28320 грн/рік) грн, ціна 1720 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, у відповідності із таблиці 4.1 [17].

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах

Продовження табл. 4.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	3	3	4
Сума	41	40	42
Середньоарифметична сума балів	(41+40+42) / 3 = 41		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3 [17].

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що програмний продукт відрізняється від існуючих тим, що дана технологія дозволяє підвищити точність класифікації текстової україномовної інформації про масиви вод басейну річки Південний Буг і має адаптацію до задачі класифікації інформації україномовних джерел про масиви вод басейну річки Південний Буг. Розроблювана технологія матиме високий показник точності автоматичної класифікації [17].

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$З_о = \frac{M}{T_р} \cdot t, \qquad (4.1)$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн;

$T_р$ – число робочих днів за місяць, 22 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	27000	1227,27	35	42954,545
Програміст	22000	1000,00	35	35000,000
Всього				77954,55

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата розробників, які брали участь в розробці обладнання [17].

Додаткову заробітну плату прийнято розраховувати як 13 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 13\ \% / 100\ \%. \quad (4.2)$$

$З_д = (77954,55 \cdot 13\ \% / 100\ \%) = 10134,09$ (грн).

Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22\ \% / 100\%. \quad (4.3)$$

$Н_з = (77954,55 + 10134,09) \cdot 22\ \% / 100\ \% = 19379,50$ (грн).

Витрати на матеріали на комплектуючі.

Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючи, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{Тв} \cdot \frac{t_{вик}}{12} \text{ [грн]}, \quad (4.4)$$

де Ц – балансова вартість обладнання, [грн];

Т – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{вик}$ – термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 20000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,59 міс.

$$A_{обл} = \frac{2000}{2} * \frac{1{,}59}{12} = 1325{,}76 \text{ (грн)}. \quad (4.5)$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5. Вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів є безкоштовною [17].

Таблиця 4.5 – Амортизаційні відрахування на матеріальні і нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	20000	2	1,59	1325,758
Офісне обладнання (меблі)	20000	4	1,59	662,879
Приміщення	750000	20	1,59	4971,591
Всього				6960,23

Тарифи на електроенергію. Для непобутових споживачів тарифи на електроенергію (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \qquad (4.6)$$

де В – вартість 1 кВт-години електроенергії для 1 класу підприємства, В = 6,2 грн./кВт;

П – встановлена потужність обладнання, кВт. П = 0,4 кВт;

Ф – фактична кількість годин роботи обладнання, годин;

$К_п$ – коефіцієнт використання потужності, $К_п$ = 0,9;

$В_е = 0,9 \cdot 0,4 \cdot 8 \cdot 35 \cdot 6,2 = 624,96$ (грн).

Інші витрати та загальновиробничі витрати [17]. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.7)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$I_в$ = 77954,55 * 90% / 100% = 70159,09 (грн).

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників:

$$Н_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.8)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$Н_{нзв}$ = 77954,55 * 105 % / 100 % = 81852 (грн).

Витрати на проведення науково-дослідної роботи [17]. Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$В_{заг}$ = 77954,55+10134,09+19379,50+6960,23+624,96+70159,09+81852 =
= 267064,69 (грн).

Розрахунок загальних витрат. Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів

визначається за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} \text{ [грн]}, \quad (4.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η=0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9. Оберемо η = 0,5, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$ЗВ$ = 267064,69 / 0,5 = 534129 (грн).

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки [17].

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

– абсолютного економічного ефекту (чистого дисконтованого доходу);

– внутрішньої економічної дохідності (внутрішньої норми дохідності);

– терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами [17].

Вдосконалення програмного засобу для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \qquad (4.10)$$

де $\pm\Delta Ц_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів, які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_o$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки,

$Ц_o = Ц_б \pm \Delta Ц_o$;

$Цб$ – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0{,}8333$;

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році ϑ =18%.

Припустимо, що при прогнозованій ціні 9500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на *500* грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 500 шт., протягом другого року – на 400 шт., протягом третього року на 300 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$\Delta П_1$ = (0*500 + (9500 + 500)*500)* 0,8333* 0,4) * (1 - 0,18) = 1298333,281 (грн).

$\Delta П_2$ = (0*500 + (9500 + 500)*(500+400)* 0,8333* 0,4) * (1 - 0,18) = 2459999,902 (грн).

$\Delta П_3$ = (0*500 + (9500 + 500)*(500+400+300)* 0,8333* 0,4) * (1 - 0,18) = 3279999,869 (грн).

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 7038333,05 грн.

Розрахунок ефективності вкладених інвестицій та періоду їх окупності [17].

Розраховуємо приведену вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (4.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05...0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

ПП=(1298333,281/(1+0,1)^1)+(2459999,902/(1+0,1)^2)+(3279999,869/(1+0,1)^3) = 1180302,98 + 2033057,77 + 2464312,448 = 5677673,201 (грн).

Далі розраховують величину початкових інвестицій *PV*, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} * ЗВ, \quad (4.12)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}=2...5$, але може бути і більшим;
ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

PV = 2 * 534129 = 1068258,75 (грн).

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід *(NPV, Net Present Value)* для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП – PV, \quad (4.13)$$

$E_{абс}$ = 5677673,201 – 1068258,75 = 4609414,45 (грн).

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності *(IRR, Internal Rate of Return)* вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно [17].

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.14)$$

$T_{ж}$ – життєвий цикл наукової розробки, роки.

$E_в$ = 3 $(1\sqrt{+}\sqrt{4609414,45/1068258,75} - 1 = 0,745$.

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d = (0,09…0,14);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{\min} = 0,14 + 0,05 = 0,19. \quad (4.16)$$

Так як $E_в > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в}, \quad (4.17)$$

$T_{ок}$ = 1 / 0,745 = 1,34 р.

Оскільки $T_{ок}$ < 3-х років, а саме термін окупності рівний 1,34 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновки

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 534129 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є високо конкурентоспроможним. Період окупності складе близько 1,34 роки.

ВИСНОВКИ

Дана магістерська робота присвячена розробці інтелектуальної інформаційної технології класифікації україномовної інформації про масиви вод басейну річки Південний Буг. В ній описано загальну характеристику об’єкту досліджень. Здійснено постановку задачі та обрано оптимальні інформаційні технології для її вирішення.

Розглянуто важливість створення інтелектуальної інформаційної системи, яка має пришвидшити класифікацію інформації. Також, розглянуто методи вирішення подібних задач, зокрема – методи і технології оброблення природномовних даних (NLP) для перетворення тексту на ембеддинги, а також – моделі машинного навчання, які використовуються для класифікації тексту за цими ембеддингами.

Розглянуто етапи створення та аналізу датесету і застосування до нього NLP-моделей та моделей машинного навчання. Для роботи з цими технологіями необхідним є створення датасету із відповідним набором даних. Для полегшення цього завдання розроблено автоматичний парсер для парсингу даних із різних інформаційних джерел різного формату.

Здійснено розроблення інтелектуальної інформаційної технології автоматичної класифікації україномовної екологічної інформації. Також обрано дві можливі оптимальні моделі. Їх перевірка на ряді прикладів довела, що дійсно оптимальною моделлю є модель випадкових лісів (Random Forest), яка має точність класифікації 0.95 і дає більш адекватний результат.

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 534129 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок про те, що

розроблений програмний продукт за ціною дешевший за аналог і є високо конкурентоспроможним. Період окупності складе близько 1,34 роки.

За результатами даної роботи у ряді прикладі вдалось підвищити точність класифікації текстових україномовних даних про водні ресурси, які синтезуються автоматично з використанням методів штучного інтелекту. Отже, поставлене у магістерській кваліфікаційній робот завдання виконано в повному обсязі.

За результатами даної роботи написані тези доповіді, які були апробовані на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2021-2022 рр.) та опубліковані у матеріалах цієї конференції.

Дана магістерська кваліфікаційна робота виконана на замовлення басейнової ради Південного Бугу, куди й передано її результати. Про них вже йшлось в обговоренні до п.4 порядку денного засідання ради (див. протокол № 12 від 07.12.2022 р. на сайті БУВР Південного Бугу).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В. Б. Мокін, Є. М. Крижановський, М. А. Гораш, Д. В. Пасічнюк, О. В. Радецький, С. М. Хмарук. Інформаційні технології збирання та систематизування даних про стан та проблеми масивів вод на прикладі басейну річки Південний Буг. Молодь в науці: дослідження, проблеми, перспективи (МН-2022). Вінниця, 2022. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/viewFile/16236/13677

2. Special Characters and Symbols in Word. URL: https://www.lifewire.com/use-special-characters-symbols-in-word-3540286

3. Нібулон. URL: https://www.nibulon.com/news/novini-kompanii/kompaniya-nibulon-rozpochinae-drugii-etap-dnopogliblennya-pivdennogo-bugu.html

4. What is NLP: An Introductory Tutorial to Natural Language Processing. URL: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/what-is-natural-language-processing-nlp

5. BERT. URL: https://h2o.ai/wiki/bert/

6. Everything you need to know about ALBERT, RoBERTa, and DistilBERT. URL: https://towardsdatascience.com/everything-you-need-to-know-about-albert-roberta-and-distilbert-11a74334b2da

7. Linear Regression. URL: https://mlu-explain.github.io/linear-regression/

8. Random Forests. URL: https://scikit-learn.org/stable/modules/ensemble.html - random-forests

9. Logistic Regression. URL: https://scikit-learn.org/stable/modules/linear_model.html - logistic-regression

10. Support Vector Machine Algorithm. URL: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

11. LinearSVC. URL: https://scikit-learn.org/stable/modules/svm.html

12. ER-EnvProbl – NLP: Bag of Words, TF IDF, GloVe. URL: https://www.kaggle.com/code/kenywhite/er-envprobl-nlp-bag-of-words-tf-idf-glove

13. NLP for UA: BERT CLS & 5 Classifiers. URL: https://www.kaggle.com/code/kenywhite/nlp-for-ua-bert-cls-5-classifiers?scriptVersionId=113854168

14. Вежа вінницький інформаційний канал. URL: https://vezha.ua/na-vinnytskyh-plyazhah-gonty-ta-himik-kupatys-zaboroneno-derzhprodspozhyvsluzhba/

15. Web scraping from HTML - BUWR SB site-parser. URL: https://www.kaggle.com/code/vbmokin/web-scraping-from-html-buwr-sb-site-parser

16. Гідрографічна мережа. URL: https://buvrpb.davr.gov.ua/vodni-resursy/hidrohrafichna-merezha

17. Методичні вказівки до написання магістерської кваліфікаційної роботи для студентів. URL: http://pdf.lib.vntu.edu.ua/books/2022/Yaremchuk_2018_43.pdf

18. Савченко А.С., Синельніков О.О. Методи та системи штучного інтелекту: навч. посіб. Київ, 2021, 9 с.

19. Грицай Д.І. Розумні роботи. Щомісячний правовий часопис. 2020 № 1/2 С. 104

20. В. Б. Мокін, Є. М. Крижановський. Геоінформаційні системи в екології: навч. посіб / за ред. Є. М. Крижановського. Вінниця : ВНТУ, 2014. 61 с.

21. Мокін В. Б., Ю. Собко, Є. М. Крижановський, М.В. Дратований, Г. В. Горячев. Створення інформаційної системи моніторингу забруднення атмосферного повітря міста на основі технології «Інтернет речей», Вісник ВПІ. 2017. № 3. С. 49-58.

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«_19_» _____09_____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«ІНФОРМАЦІЙНА NLP-ТЕХНОЛОГІЯ КЛАСИФІКАЦІЇ УКРАЇНОМОВНОЇ ІНФОРМАЦІЇ ПРО СТАН ВОДНИХ РЕСУРСІВ»
08-53.МКР.002.02.000.ТЗ

Керівник: д.т.н., проф. каф. САІТ
_____ Мокін В. Б.
«_19_» _____09_____ 2022 р.

Розробив: студент гр. 2ІСТ-21м
_____ Пасічнюк Д.В.
«_19_» _____09_____ 2022 р.

Вінниця 2022

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № 203 по ВНТУ від «14» _09_ 2022 р., та індивідуальне завдання на МКР, затверджене протоколом № 3 засідання кафедри САІТ від «14» __09__ 2022 р.

2. Джерела розробки:

- План управління річковим басейном Південного Бугу: аналіз і заходи річкового басейну (англ.) / Мокін В. та ін. ; за ред. С. Афанасьєв, А. Петерс, В. Сташук, О. Ярошевич., Київ, 2014. 188 с.
- NLP for UA : BERT CLS & 10 Classifiers. URL: https://www.kaggle.com/code/vbmokin/nlp-for-ua-bert-cls-10-classifiers
- NLP for UA : BERT CLS for the notebook result. URL: https://www.kaggle.com/code/vbmokin/nlp-for-ua-bert-cls-for-the-notebook-result
- NLP : Reports & News Classification. URL: https://www.kaggle.com/datasets/vbmokin/nlp-reports-news-classification

3. Мета і призначення роботи:

Підвищення точності класифікації текстової україномовної інформації про масиви вод басейну річки Південний Буг.

4.Вихідні дані для проведення робіт:

Датасет Kaggle «NLP : Reports & News Classification» з даними про водні ресурси.

5.Методи дослідження:

Методи машинного навчання, регресійні моделі та моделі, які побудовані на основі дерев рішень.

6. Етапи роботи і терміни їх виконання:

1. Характеристика проблеми класифікації україномовної інформації про стан водних ресурсів ..... 19.09 – 30.09
2. Вибір оптимальних налаштувань інформаційної технології для розв’язання поставленої задачі ..... 01.10 – 16.10
3. Створення інтелектуальної інформаційна технології класифікації україномовної інформації про масиви вод басейну річки Південний Буг ..... 17.10 – 13.11
4. Економічна частина ..... 14.11 – 23.11
5. Оформлення пояснювальної записки. ..... 24.11 – 29.11

7. Очікувані результати та порядок реалізації:

Адекватна класифікація текстових україномовних даних про водні ресурси, які синтезуються автоматично з використанням методів штучного інтелекту.

8. Вимоги до розробленої документації.

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання».

9. Порядок приймання роботи

1. Публічний захист ..... «_19_» _____12___ 2022 р.
2. Початок розробки ..... «_19_» _____09_____ 2022 р.
3. Граничні терміни виконання МКР..... «_30_» _____11_____ 2022 р.

Розробив студент групи 2ІСТ-21м _____ Пасічнюк Д.В.

Додаток Б

Протокол перевірки кваліфікаційної роботи на наявність

текстових запозичень

Назва роботи: «Інформаційна NLP-технологія класифікації україномовної інформації про стан водних ресурсів»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ
Науковий керівник: Мокін В. Б. д.т.н., проф. каф. САІТ

Показники звіту подібності Unicheck

Оригінальність	99 %
Схожість	1 %

Аналіз звіту подібності (відмітити потрібне):

- ✓ запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату;
- виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри;
- виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Опис прийнятого рішення:

Робота допускається до захисту

Особа, відповідальна за перевірку Жуков С. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи Пасічнюк Д.В.

Керівник роботи Мокін В. Б

Додаток В

Лістинг програми

```
import numpy as np
import pandas as pd
import pickle
import matplotlib
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# models
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
# NLP
import torch
import transformers as ppb
import warnings
warnings.filterwarnings('ignore')

# Model choice
# model_name = "random-forest"
model_name = "linear-regression"
# Get notebook_models name
if model_name=="random-forest":
    notebook_models = "nlp-for-ua-bert-cls-with-rf"
elif model_name=="linear-regression":
    notebook_models = "nlp-for-ua-bert-cls-with-linreg"
```

```
# Set parameters
random_state = 0
notebook_data = "web-scraping-from-html-buwr-sb-site-parser"
data_name = "BUWR-SB-basin-water-resources.csv"
df = pd.read_csv(f'/kaggle/input/{notebook_data}/{data_name}')
df = df.fillna(0)
convert_dict = {'text': str,
                'env_problems': int,
                'pollution': int,
                'treatment': int,
                'climate': int,
                'biomonitoring': int}

df = df.astype(convert_dict)
df
df.info()

def BERT_modeling(df, text='text'):
   # Using DistilBERT for the dataframe df[text]
   model_class, tokenizer_class, pretrained_weights = (ppb.DistilBertModel, ppb.DistilBertTokenizer, 'distilbert-base-multilingual-cased')
   # Other models: https://huggingface.co/transformers/pretrained_models.html
   # Load pretrained model/tokenizer
   tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
   model = model_class.from_pretrained(pretrained_weights)
   # Tokenization the sentences - break them up into word and subwords in the format BERT is comfortable with
   tokenized = df[text].apply((lambda x: tokenizer.encode(x, add_special_tokens=True)))
```

```
    max_len = 0
    for i in tokenized.values:
        if len(i) > max_len:
            max_len = len(i)
    padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])

    # Creation variable to ignore (mask) the data padding
    attention_mask = np.where(padded != 0, 1, 0)
    # Modeling
    input_ids = torch.tensor(padded).to(torch.int64)
    attention_mask = torch.tensor(attention_mask).to(torch.int64)
    with torch.no_grad():
        last_hidden_states = model(input_ids, attention_mask=attention_mask)
    # Last hidden states
    features = last_hidden_states[0][:,0,:].numpy()
    return features
features = BERT_modeling(df, 'text')

def model_prediction(df, test_features, target):
    # Models training and data prediction for all models from DataFrame models
    # Target
    labels = df[target]
    # Model loading
    with open(f'/kaggle/input/{notebook_models}/model-{model_name}-
{target}.pkl','rb') as f:
        model = pickle.load(f)
    # Prediction
    test_pred = model.predict(test_features)
    test_pred = [int(x>0) for x in test_pred]
```

```
    return test_pred
# List of the target features in df
cols = df.columns.tolist()[1:]
print('Target columns:', cols)

# Solving NLP Classification tasks
print('Solving NLP Classification tasks')
for col in cols:
    df[col] = model_prediction(df, features, col)

# Results display
pd.set_option('max_colwidth',1000)
df.to_csv(f'{notebook_data}-{notebook_models}-classified.csv', index=False)
print(f"The result of classification using the model {model_name}")
display(df)
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА NLP-ТЕХНОЛОГІЯ КЛАСИФІКАЦІЇ УКРАЇНОМОВНОЇ ІНФОРМАЦІЇ ПРО СТАН ВОДНИХ РЕСУРСІВ

Виконав: студент гр. 2ІСТ-21м
_____ Пасічнюк Д.В.
«_01_» _____12_____ 2022 р.

Керівник: д.т.н., проф. каф. САІТ
_____ Мокін В. Б.
«_02__» _____12_____ 2022 р.

Нормоконтроль: к.т.н., доцент
_____ Жуков С. О.
«_02_» _____12_____ 2022 р.

Рисунок Г.1 – Алгоритм розробки моделі класифікаторів

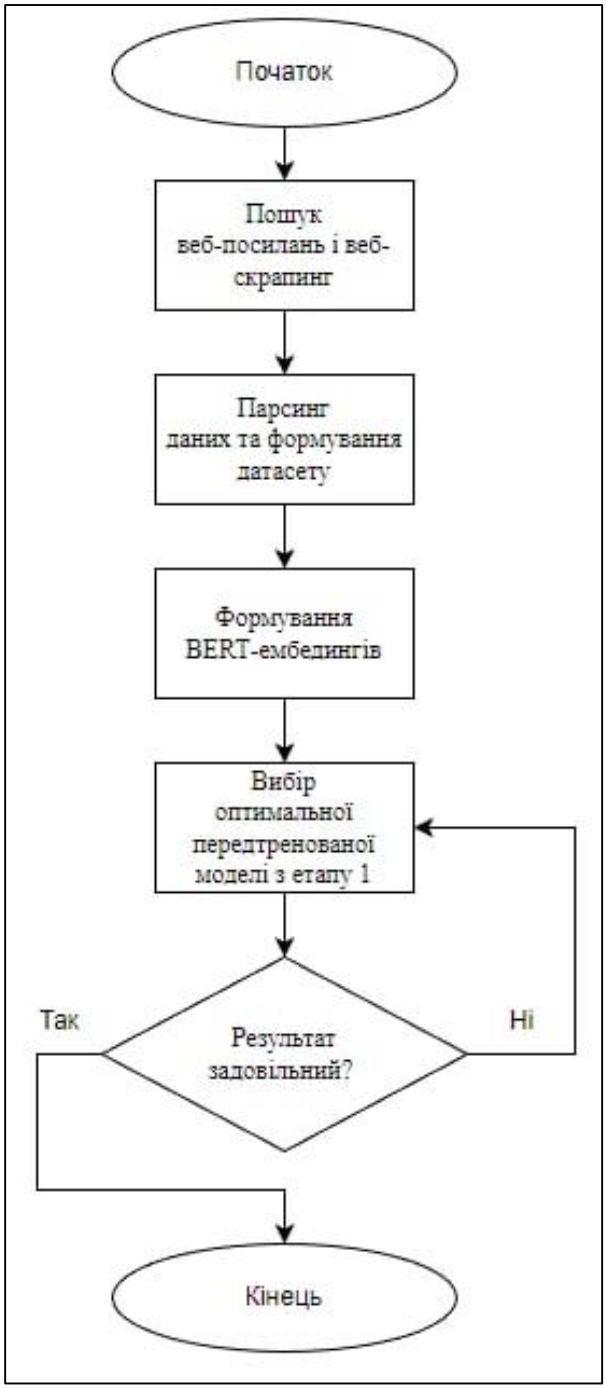


Рисунок Г.2 – Алгоритм інформаційної технології

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У басейні Південного Бугу налічується 6582 мал...	0	0	0	0	0
1	Південний Буг бере початок на Волино-Подільськ...	0	0	0	0	0
2	У Кіровоградській, Одеській та Миколаївській о...	0	0	0	0	0
3	Дніпро-Бузький лиман є мілководною затокою Чор...	0	0	0	0	0
4	За своїми розмірами Дніпро-Бузький лиман є най...	0	0	0	0	0
5	Дніпро-Бузький лиман інколи розглядають як дві...	0	0	0	0	0
6	Характерною особливістю басейну Південного Буг...	0	0	0	0	0
7	Ставки. В басейні розташовано 9877 ставків, за...	0	0	0	0	0
8	Водосховища. В басейні розташовано 187 водосхо...	0	0	0	0	0
9	Для водопостачання обласного центру побудовани...	0	0	0	0	0
10	Водогоном довжиною 130 км вода подається з р. ...	0	0	0	0	0

Рисунок Г.3 – Сформований датасет для тестування моделей

	name	acc_train	acc_test
5	Mean values	0.9875	0.80
1	Random Forest Classifier	0.9625	0.80
0	Linear Regression	1.0000	0.75
2	Logistic Regression	1.0000	0.70
3	Support Vector Machines	1.0000	0.70
4	Linear SVC	1.0000	0.70
6	Max values	0.9750	0.70

Рисунок Г.4 – Результат роботи відібраних моделей

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У басейні Південного Бугу налічується 6582 малих річок, загальною довжиною близько 20 тис. км, 11 середніх річок загальною довжиною понад 1,6 тис. км та 1 велика річка Південний Буг.	1	1	1	1	0
1	Південний Буг бере початок на Волино-Подільській височині поблизу с.Холодець Хмельницької області і впадає в Дніпро-Бузький лиман Чорного моря. Довжина річки становить 806 км, загальне падіння 321 м, середній схил водної поверхні 0,4 ‰.	1	1	0	1	0
2	У Кіровоградській, Одеській та Миколаївській областях зустрічаються невеликі озера. Їх загальна площа складає 190 га.	0	0	1	0	1
3	Дніпро-Бузький лиман є мілководною затокою Чорного моря, що відокремлений від нього Кінбурнською косою. Сполучається лиман з морем Кінбурнською протокою, через яку і здійснюється водообмін. На сході лиман межує з гирлом Дніпра, на півночі поступово переходить у р. Південний Буг.	1	1	0	1	0
4	За своїми розмірами Дніпро-Бузький лиман є найбільшим на Чорному морі. Його площа складає 800 км2, об'єм води – 3 км3. Лиман є мілководним, його середня глибина становить 3,5-4 м.	0	0	1	1	0
5	Дніпро-Бузький лиман інколи розглядають як дві водойми – Дніпровський та Бузький, але чіткої межі між ними немає. Якщо межі дніпровської частини лиману є доволі чіткими, то верхня межа бузької частини є дискутивною. Найчастіше цю межу проводять по Варварівському мосту, що знаходиться в м. Миколаїв. Довжина Бузького лиману 47 км, ширина – 11 км, площа 162 км2. Ступінь мінералізації лиману – солонуватий.	1	1	1	1	1
6	Характерною особливістю басейну Південного Бугу, що виділяє його з поміж інших великих річок є дуже велика його зарегульованість. В басейні створено майже 10 тисяч штучних водойм, сумарний їх об'єм понад 1,5 км3, що практично дорівнює стоку в маловодний рік 95% забезпеченості.	1	1	1	0	0
7	Ставки. В басейні розташовано 9877 ставків, загальною площею понад 56,4 тис. га та сумарним об'ємом 644 млн.м3. Внаслідок проведеної у 2013 році в областях басейну інвентаризації водних об'єктів, загальна кількість ставків порівняно із 2012 роком збільшилася на 122 шт. Найбільша кількість ставків побудована у Вінницькій (3443), Черкаській (2206) та Кіровоградській (2188) областях.	1	1	1	0	0
8	Водосховища. В басейні розташовано 187 водосховищ місткістю 894 млн.м3, загальною площею водного дзеркала майже 30 тис. га. На самому руслі р.Південний Буг побудовано 16 водосховищ місткістю 316 млн. м3, які використовуються для потреб гідроенергетики, водопостачання та рекреації. Найбільша кількість водосховищ побудована в Кіровоградській (64) і Вінницькій (42) областях.	1	1	1	0	0
9	Для водопостачання обласного центру побудований водовід із басейну Дніпра с.Чернилівка – м. Хмельницький довжиною 34 км пропускною здатністю 0,8 м3/с. За 2013 рік подано 20 млн.м3.	1	1	1	1	1
10	Водогоном довжиною 130 км вода подається з р. Рось (басейн Дніпра) до м.Умань Черкаської області, пропускною здатністю 0,14 м3/с. За 2013 рік подано 3,9 млн.м3.	1	1	1	1	1

Рисунок Г.5 – Класифікація за допомогою моделі лінійної регресії

	text	env_problems	pollution	treatment	climate	biomonitoring
0	У басейні Південного Бугу налічується 6582 малих річок, загальною довжиною близько 20 тис. км, 11 середніх річок загальною довжиною понад 1,6 тис. км та 1 велика річка Південний Буг.	0	0	0	0	0
1	Південний Буг бере початок на Волино-Подільській височині поблизу с.Холодець Хмельницької області і впадає в Дніпро-Бузький лиман Чорного моря. Довжина річки становить 806 км, загальне падіння 321 м, середній схил водної поверхні 0,4 ‰.	0	0	0	0	0
2	У Кіровоградській, Одеській та Миколаївській областях зустрічаються невеликі озера. Їх загальна площа складає 190 га.	0	0	0	0	0
3	Дніпро-Бузький лиман є мілководною затокою Чорного моря, що відокремлений від нього Кінбурнською косою. Сполучається лиман з морем Кінбурнською протокою, через яку і здійснюється водообмін. На сході лиман межує з гирлом Дніпра, на півночі поступово переходить у р. Південний Буг.	0	0	0	0	0
4	За своїми розмірами Дніпро-Бузький лиман є найбільшим на Чорному морі. Його площа складає 800 км2, об'єм води – 3 км3. Лиман є мілководним, його середня глибина становить 3,5-4 м.	0	0	0	0	0
5	Дніпро-Бузький лиман інколи розглядають як дві водойми – Дніпровський та Бузький, але чіткої межі між ними немає. Якщо межі дніпровської частини лиману є доволі чіткими, то верхня межа бузької частини є дискутивною. Найчастіше цю межу проводять по Варварівському мосту, що знаходиться в м. Миколаїв. Довжина Бузького лиману 47 км, ширина – 11 км, площа 162 км2. Ступінь мінералізації лиману – солонуватий.	1	0	0	0	0

Рисунок Г.6 – Класифікація за допомогою моделі випадкових лісів (Random Forest)