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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія класифікації типу вин за їх хімічним складом»

Виконав: студент 2 курсу, групи 1ІСТ-23м
спеціальності 126 – «Інформаційні системи та технології»
_____ Дмитро СОКУР

Керівник: д.т.н., проф. каф. САІТ
_____ Віталій МОКІН
«05» 12 2024 р.

Опонент: проф. каф. КН, д.т.н.,
_____ Ярослав ІВАНЧУК
«08» 12 2024 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«06» 12 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – другий (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. Віталій МОКІН
«17» 09 2024 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Сокуру Дмитру Сергійовичу

1. Тема роботи: «Інформаційна технологія класифікації типу вин за їх хімічним складом»,
керівник роботи: Віталій МОКІН, д.т.н., проф. каф. САІТ
затверджені наказом ВНТУ від «17» 09 2024 року №310
2. Строк подання роботи студентом: до «01» 12 2024 року.
3. Вихідні дані до роботи:
   - датасет Kaggle «Wine Quality Data Set (Red & White Wine)»
4. Зміст текстової частини:
   - загальна характеристика поставленої задачі;
   - вибір оптимальних налаштувань алгоритму для розв'язання поставленої задачі;
   - розроблення інформаційної технології для класифікації типу вина;
   - економічний розділ.
5. Перелік ілюстративного матеріалу:
   - діаграма класів;
   - діаграма діяльності;
   - діаграма активності;
   - результати прогнозування;
   - heatmap ознак використаного датасету;
   - найважливіші ознаки які мають найбільший вплив.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Олександр ЛЕСЬКО, завідувач кафедри ЕПВМ, професор, к.е.н.	10.11.24	30.11.24

7. Дата видачі завдання «17» 09 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Загальна характеристика поставленої задачі	30.09	15.10	вик.
2	Вибір оптимальних налаштувань алгоритму	16.10	31.10	вик.
3	Розробка інформаційної технології	01.11	15.11	вик
4	Підготовка економічної частини	16.11	25.11	вик
5	Оформлення пояснювальної записки, графічного матеріалу і презентації	25.11	30.11	вик

Студент Дмитро СОКУР

Керівник роботи Віталій МОКІН

АНОТАЦІЯ

УДК 004.08

Сокур Д. С. Інформаційна технологія класифікації типів вина за їх хімічним складом. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2024. 101 с.

На укр. мові. Бібліогр.: 29 назв; рис.: 35; табл.: 9.

У даній магістерській роботі розглянуто процес класифікації типів вина на основі даних про його хімічний склад. Використано стандартизований набір даних, який включає такі показники, як кислотність, вміст алкоголю, рівень цукру та інші хімічні характеристики. Виконано розвідувальний аналіз даних, обрано оптимальні методи машинного навчання, а також побудовано кілька моделей класифікації. Найкращою моделлю за результатами тестування виявився алгоритм XGBoost, який забезпечив найвищу точність прогнозування. Проведено оцінку ефективності роботи обраної моделі та визначено її можливе практичне застосування у виноробній галузі.

Ілюстративна частина складається з 6 плакатів із результатами моделювання.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження інформаційної технології класифікації типу вин за їх хімічним складом.

Ключові слова: класифікація типів вина, хімічний склад, машинне навчання, XGBoost.

ABSTRACT

Sokur D. S. Information technology for wine type classification based on its chemical composition. Master's qualification thesis in the specialty 126 – Information Systems and Technologies, educational-professional program – Information Technology for Data and Image Analysis. Vinnytsia: VNTU, 2024. 101 pages.

In Ukrainian. Bibliography: 29 titles; illustrations: 35; tables: 9.

This master's thesis examines the process of classifying wine types based on data about its chemical composition. A standardized dataset was used, including indicators such as acidity, alcohol content, sugar levels, and other chemical characteristics. Exploratory data analysis was conducted, optimal machine learning methods were selected, and several classification models were developed. The XGBoost algorithm was identified as the best-performing model based on testing results, providing the highest prediction accuracy. The effectiveness of the selected model was assessed, and its potential practical application in the winemaking industry was determined.

The illustrative section consists of 6 posters with the results of modeling.

The economic section addresses the feasibility of developing and implementing an information technology for classifying wine types based on their chemical composition.

Keywords**:** wine type classification, chemical composition, machine learning, XGBoost.

ЗМІСТ

ВСТУП

Актуальність теми. Швидкий розвиток інформаційних технологій створює нові можливості для аналізу та прогнозування складних процесів у різних галузях. У виноробній промисловості, де хімічний склад вина визначає його якість і тип, використання сучасних технологій для аналізу даних стає надзвичайно актуальним. Різноманітність компонентів і складність технологічних процесів створюють виклики, які можна вирішити шляхом впровадження методів прогнозування на основі машинного навчання.

Магістерська кваліфікаційна робота присвячена дослідженню методів аналізу та прогнозування типів вина за їх хімічним складом. Актуальність роботи зумовлена потребою в оптимізації технологічних процесів у виноробстві, підвищенні якості продукції та ефективному використанні ресурсів. Використання інформаційних технологій дозволяє не лише підвищити точність класифікації типів вина, а й сприяти стабільності ринкової ситуації та розвитку виноробної галузі загалом.

Мета і завдання роботи. Метою роботи є підвищення точності класифікації типів вина на основі хімічного складу шляхом розробки інформаційної технології.

Розробка інформаційної технології передбачає виконання наступних задач:

- надати загальну характеристику поставленої задачі;
- обрати оптимальні налаштування алгоритму;
- розробити інформаційну технологію для класифікації типу вина;
- зробити розрахунок економічної частини.

Об’єктом дослідження є процес класифікації типів вина на основі даних хімічного складу.

Предметом дослідження є методи та засоби інформаційних технологій і статистичних моделей, які використовуються для аналізу та прогнозування типів вина.

Методи дослідження. У роботі застосовано сучасні підходи аналізу даних, методи машинного навчання та моделі прогнозування.

Новизна одержаних результатів. Дістала подальший розвиток інформаційна технологія класифікації типу вин за їх хімічним складом, за рахунок використання інноваційних інтелектуальних методів аналізу даних, що дозволяє підвищити точність цієї класифікації

Практична цінність. Розроблено алгоритми та програмні засоби на Python для реалізації удосконаленої інформаційної технології, що дозволило отримати результати, які сприятимуть вдосконаленню виробничих процесів, підвищенню якості виноробної продукції та ефективному управлінню ресурсами у виноробній промисловості.

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи подані на Міжнародну науково-практичну інтернет-конференцію «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.).

Публікації результатів магістерської кваліфікаційної роботи. Опубліковано тези на Міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.) [1].

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1 Актуальність класифікації типу вин за їх хімічним складом

Як відомо, вино — це продукт, який має різні варіанти реалізації, різноманітні категорії. Є поширеним в усіх країнах світу. Воно використовується не лише як напій, а й у кулінарії, медицині та як частина різноманітних церемоній та традицій. Смакові та ароматичні властивості визначаються хімічним складом вина. Це впливає і на його стабільність і тривалість зберігання. Знання точного хімічного складу вина дозволяє виробляти продукт з необхідними характеристиками для конкретних ринків та споживачів [2].

Класифікація типів вина за даними хімічного складу є надзвичайно важливою для забезпечення стабільної якості продукції та оптимізації виробничих процесів. Це дозволяє зменшити витрати на виробництво, підвищити ефективність використання сировини, а також знизити кількість відходів та дефектної продукції. В умовах сучасної економіки, де конкуренція та вимоги до якості постійно зростають, здатність точно прогнозувати типи вина стає конкурентною перевагою для виноробів.

Використання методів машинного навчання та аналізу даних для класифікації типів вина є перспективним напрямком досліджень. Це дозволяє враховувати велику кількість факторів та їх взаємодій, що є складним завданням для традиційних методів аналізу. Застосування сучасних технологій може значно покращити точність прогнозів та забезпечити більш надійні результати.

На рисунку 1.1 зображено статистику виробництва вина в 2014 році [3]. З неї помітно, що виробничі обсяги у Європі є найбільшими.

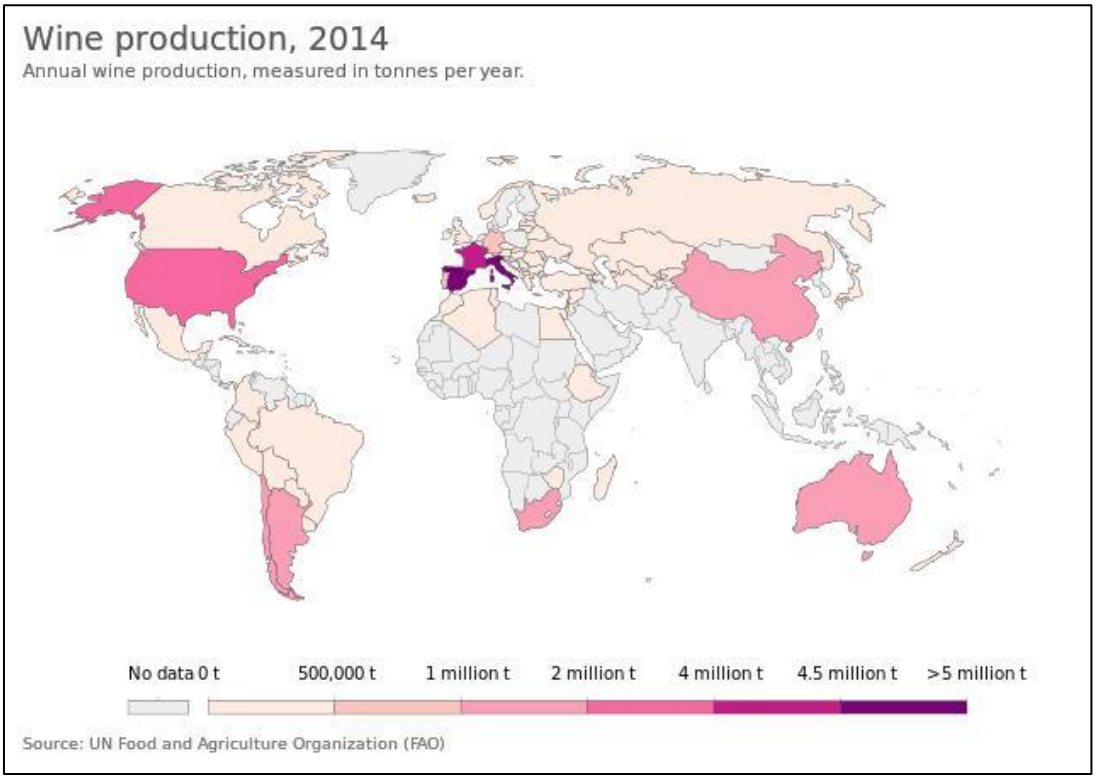


Рисунок 1.1 – Статистика виробництва вина

На рисунку 1.2 можна побачити споживання вина в різних країнах Європи [4]. Можна помітити, що об’єми досить великі.

Рисунок 1.2 – Споживання вина у Європі 2015 рік

При виробництві вина на його тип можуть впливати багато різних чинників. Найважливіші з них:

1. Alcohol (Вміст алкоголю): вміст алкоголю є одним з основних параметрів вина, який визначає його міцність та впливає на смакові властивості. Високий вміст алкоголю додає винам структуру та тіло, тоді як нижчий робить їх легшими та більш освіжаючими.

2. Sulphates (Сульфати): сульфати використовуються як консерванти, що допомагають запобігти окисленню вина та зберегти його свіжість. Вони також мають антибактеріальні властивості, що запобігають розвитку небажаних мікроорганізмів.

3. Citric acid (Лимонна кислота): лимонна кислота додає кислотності винам, що підкреслює їх смакові нотки та освіжає смак. Вона також впливає на стабільність вина під час зберігання.

3. pH (Водневий показник): водневий показник визначає загальний рівень кислотності вина. Вина з низьким pH мають високий рівень кислотності, що робить їх більш освіжаючими, тоді як вина з високим pH є менш кислими і мають більш м'який смак.

4. Density (Щільність): щільність вина впливає на його текстуру та органолептичні властивості. Вища щільність може вказувати на високий вміст цукру або алкоголю, тоді як нижча щільність робить вино легшим.

5. Residual sugar (Залишковий цукор): залишковий цукор впливає на солодкість вина. Вина з високим вмістом залишкового цукру є солодкими, тоді як вина з низьким вмістом цукру є сухими.

6. Chlorides (Хлориди): хлориди впливають на мінеральний склад вина та можуть додавати солонуваті нотки до його смаку. Вони також можуть впливати на текстуру вина.

7. Free sulfur dioxide (Вільний діоксид сірки): вільний діоксид сірки використовується для консервації вина та запобігання окисленню. Його вміст впливає на стабільність вина та його здатність до тривалого зберігання.

8. Total sulfur dioxide (Загальний діоксид сірки): загальний діоксид сірки включає як вільний, так і зв'язаний діоксид сірки. Високий вміст загального діоксиду сірки може впливати на аромат та смак вина, роблячи його більш стабільним.

9. Volatile acidity (Летюча кислотність): летюча кислотність впливає на ароматичні властивості вина, зокрема на наявність оцтових та інших кислотних запахів. Висока летюча кислотність може вказувати на недоліки у вині.

10. Fixed acidity (Фіксована кислотність): фіксована кислотність визначає загальний рівень кислотності вина, що впливає на його смак та стабільність. Висока фіксована кислотність робить вино більш яскравим та освіжаючим.

Крім основних параметрів, існують також додаткові важливі чинники, які впливають на якість та тип вина:

1. Чистота сировини: використання високоякісної сировини з мінімальним вмістом домішок гарантує досягнення високої якості та стабільності вина. Якість винограду та інших інгредієнтів безпосередньо впливає на кінцевий продукт, тому важливо використовувати найкращу сировину для виробництва вина.

2. Температурний режим: виробництво вина потребує точного контролю температури на різних етапах, включаючи ферментацію, зберігання та дозрівання. Температурні коливання можуть суттєво вплинути на смакові властивості, аромат та стабільність вина. Тому важливо підтримувати оптимальні температурні умови протягом всього процесу виробництва.

3. Технологія виробництва: вибір оптимальної технології виробництва, включаючи контроль температури та часу ферментації, впливає на якість та властивості кінцевого продукту. Використання сучасних технологій та обладнання дозволяє досягти стабільності та високої якості вина. Такі технології можуть включати автоматизовані системи контролю, використання спеціальних ємностей для ферментації та зберігання, а також інноваційні методи обробки винограду.

На рисунку 1.3 зображено процес виробництва вина [5].

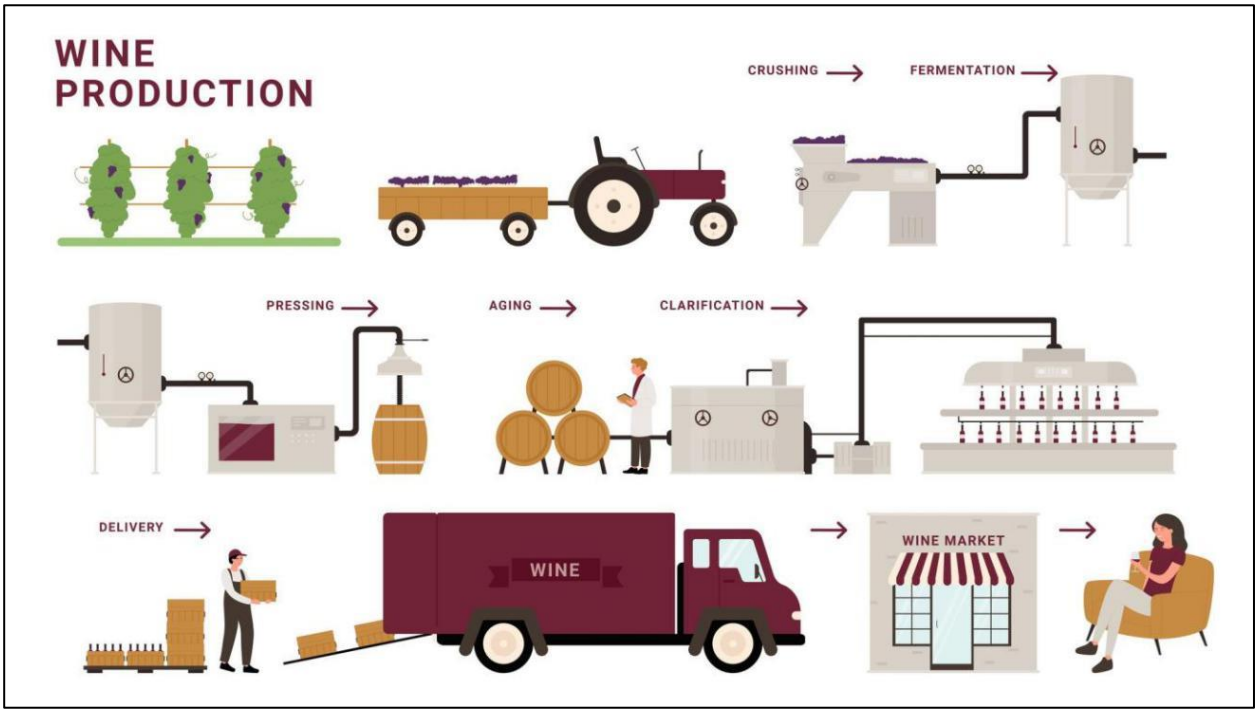


Рисунок 1.3 – Процес виробництва вина

З поданої вище інформації можна зробити висновок, що вино є дуже важливим продуктом у різних сферах, а його виробництвом займається багато країн. Цей продукт має великий вплив у світі, тому актуальність Класифікація його типів за хімічним складом є високою.

Отже, Класифікація типів вина за даними хімічного складу є доволі складною задачею, яка потребує інноваційних підходів і технологій, таких як штучний інтелект з аналітичними алгоритмами великих об’ємів даних та матеріально-технічна база у вигляді сенсорів, датчиків, IoT пристроїв тощо. Чим більше даних можна зібрати, тим кращою буде точність Класифікація.

1.2 Опис датасету

Датасет, використаний для аналізу, містить інформацію про біле та червоне вино. Кожен запис представляє один зразок вина, який описується хімічними характеристиками та його типом (біле або червоне) [6, 7]. Усього датасет складається з 6497 записів і 13 змінних, серед яких:

- 1 категоріальна змінна (“type” — тип вина, біле або червоне);
- 12 числових змінних (хімічні характеристики та оцінка якості).

Основні статистичні характеристики:

- fixed acidity (фіксована кислотність): варіюється від 3.8 до 15.9 із середнім значенням 7.22;
- “volatile acidity” (летюча кислотність): мінімум 0.08, максимум 1.58, середнє 0.34;
- “residual sugar” (залишковий цукор): від 0.6 до 65.8 із середнім значенням 5.44;
- “pH”: значення коливаються від 2.72 до 4.01 із середнім 3.22;
- “alcohol” (вміст алкоголю): від 8% до 14.9%, середнє 10.49%;
- “quality” (якість): оцінки від 3 до 9, із середнім 5.82.

На графіку 1.4 зображено розподіл оцінок якості. Більшість зразків отримали оцінки 5, 6 або 7. Це свідчить про переважну середню якість досліджуваних зразків.

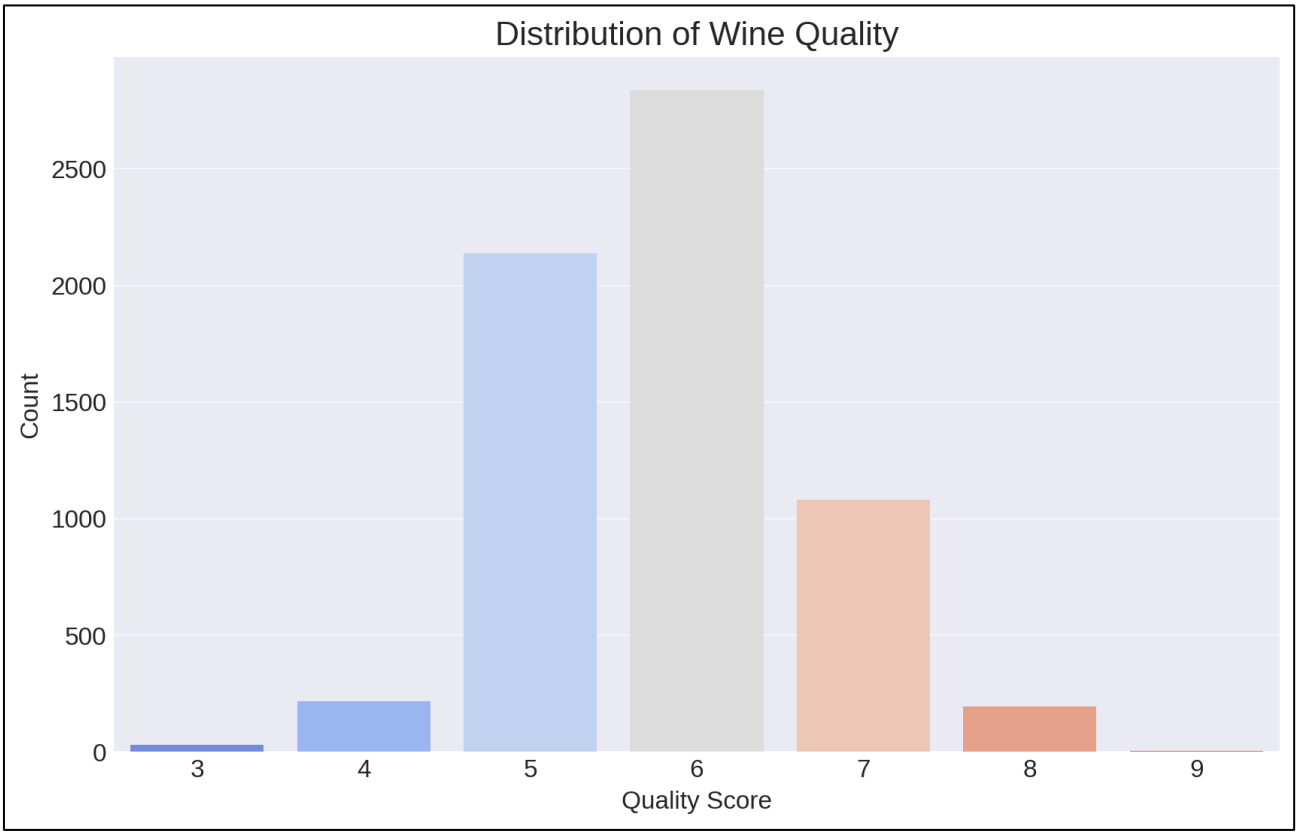


Рисунок 1.4 – Розподіл оцінок якості

Гістограма 1.5 демонструє, що більшість зразків містять від 9% до 12% алкоголю. Високий вміст алкоголю зустрічається рідше.

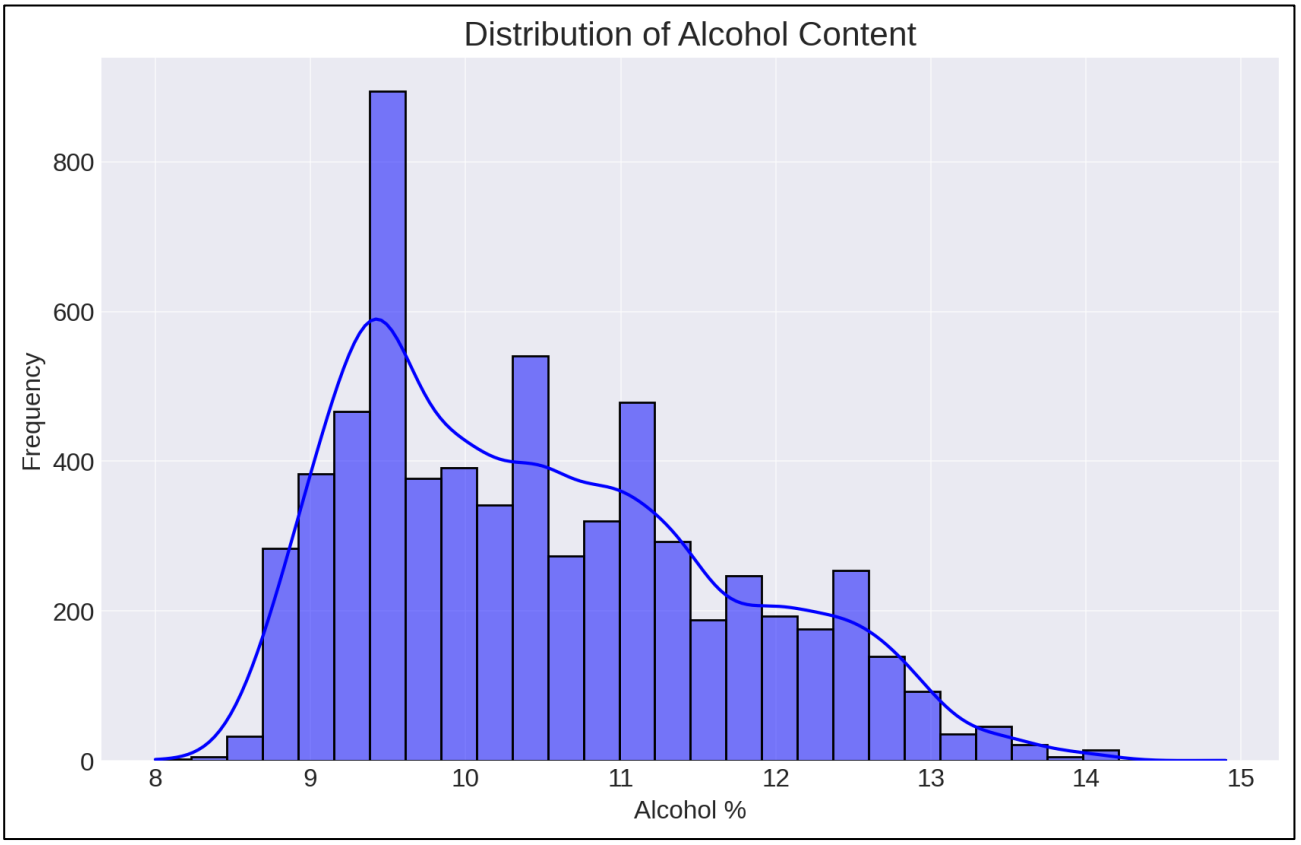


Рисунок 1.5 – Гістограма розподілу алкоголю

На діаграмі 1.6 видно, що біле вино має значно більший розподіл залишкового цукру порівняно з червоним вином.

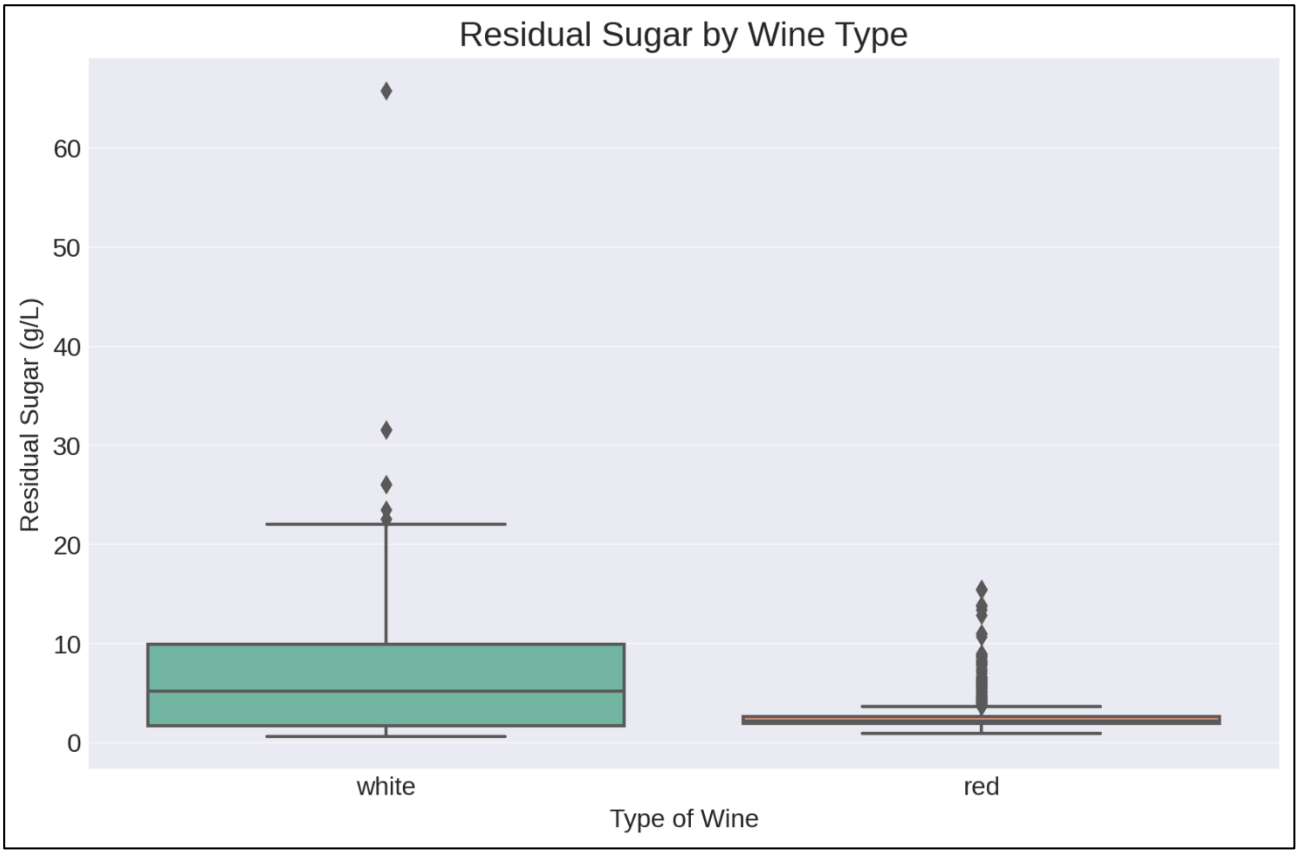


Рисунок 1.6 – Діаграма розподілу залишкового цукру

Датасет містить багату інформацію для аналізу та побудови моделей класифікації типу вина. Хімічні характеристики, такі як залишковий цукор, кислотність та вміст алкоголю, мають суттєвий вплив на тип вина, що дозволяє використовувати ці змінні для передбачення [6].

1.3 Формалізація постановки задачі

Ефективна класифікація типів вина за даними хімічного складу є складною та багатокомпонентною задачею, що потребує ретельного вибору методів обробки даних, моделей машинного навчання, а також метрик для оцінки їхньої продуктивності. Формалізація цієї задачі базується на визначенні ключових етапів: підготовки даних, вибору оптимальної моделі та оцінки її якості.

Для аналізу та класифікації використовуються дані про хімічний склад вина, які включають числові показники, такі як кислотність, залишковий цукор, рівень

хлоридів, густина, pH, концентрація сульфатів, вміст алкоголю тощо. Приклад набору даних, що слугує основою для навчання моделі, наведено на рисунку 1.8:

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
6492	1	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
6493	1	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
6494	1	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
6495	1	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
6496	1	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

Рисунок 1.8 – Набір даних з факторами впливу на тип вина

Одним із найважливіших аспектів є вибір метрик, які дозволяють об’єктивно оцінити точність і продуктивність класифікаційної моделі. У даній роботі обрано метрику accuracy_score як основну для оцінки якості моделі. Метрика accuracy_score широко використовується у задачах класифікації завдяки своїй простоті та універсальності. Вона визначає частку правильно класифікованих зразків від загальної кількості, що дозволяє оцінити загальну точність моделі [8,9].

У контексті задачі класифікації типів вина, де необхідно визначити клас вина (біле чи червоне) на основі його хімічного складу, метрика accuracy_score є логічним вибором. Вона дозволяє врахувати як правильну класифікацію обох типів вина, так і загальну ефективність моделі в умовах, коли розподіл класів є збалансованим.

Однак у задачах, де може спостерігатися дисбаланс між класами (наприклад, якщо більшість зразків належать до одного класу), accuracy_score може бути недостатньо інформативною. У таких випадках можуть використовуватися додаткові метрики, такі як F1-score, яка враховує як точність (precision), так і повноту (recall), або ROC-AUC, яка оцінює якість класифікації незалежно від порога.

Ключовим етапом розв’язання задачі є також попередня обробка даних. Для забезпечення коректної роботи алгоритмів машинного навчання виконується масштабування числових ознак, що дозволяє уникнути домінування ознак із

більшими значеннями. Також здійснюється обробка пропущених даних та виділення ключових ознак шляхом аналізу кореляції між параметрами, щоб зменшити вплив нерелевантних або слабкокорельованих змінних.

Вибір моделі класифікації базується на емпіричних експериментах із використанням алгоритмів, таких як Decision Tree, Random Forest та XGBoost, які показують високу продуктивність у задачах класифікації [6, 8]. Остаточний вибір моделі здійснюється на основі метрики accuracy_score, а також інших показників, якщо це необхідно.

Таким чином, задача класифікації формалізується як розробка інформаційної технології побудови інтелектуальної моделі для класифікації типів вина (біле чи червоне) з оптимізацією за метрикою accuracy_score. Ця модель повинна забезпечити високу точність прогнозування та бути стійкою до змін у вхідних даних.

1.4 Вибір оптимальних інформаційних технологій

Типи вина можна прогнозувати методами машинного навчання, використовуючи різноманітні моделі, такі як дерева рішень, нейронні мережі та інші, які враховують вплив багатьох чинників, таких як вміст алкоголю, сульфати, лимонна кислота, pH, щільність, залишковий цукор, хлориди, діоксид сірки, летюча та фіксована кислотність.

Типові етапи та їх використання в IT для розв'язання задач машинного навчання [6, 10, 11]:

1. Preprocessing (передобробка даних): очищення, нормалізація та кодування даних для подальшого аналізу. Це включає видалення або заповнення пропущених значень, перетворення категоріальних змінних у числові, масштабування та нормалізацію даних.

2. EDA (Exploratory Data Analysis - аналіз даних): виявлення кореляцій, візуалізація даних та пошук патернів для розуміння вхідних даних. Це допомагає ідентифікувати основні характеристики та взаємозв'язки між змінними.

3. FE (Feature Engineering - інженерія ознак): вибір та створення нових ознак для поліпшення роботи моделі машинного навчання. Це може включати створення взаємодій між змінними, розрахунок нових метрик або трансформацію існуючих ознак.

4. Model Tuning (налаштування моделі): вибір та налаштування алгоритму машинного навчання для досягнення кращих результатів. Це включає вибір параметрів моделі (гіперпараметрів), використання методів кросвалідації та оптимізацію моделі для підвищення її продуктивності.

5. Forecasting (прогнозування): застосування моделі для класифікації майбутніх значень або трендів на основі вхідних даних. Це кінцевий етап, на якому модель використовується для передбачення типу вина.

6. Postprocessing (післяобробка): аналіз результатів, оцінка точності та здійснення необхідних корекцій для вдосконалення моделі. Це включає виявлення та корекцію помилок, а також оцінку ефективності моделі за допомогою відповідних метрик.

7. Visualization (візуалізація): представлення даних та результатів моделі у зручній та зрозумілій формі для сприяння прийняттю рішень. Це можуть бути графіки, діаграми, таблиці та інші форми візуалізації, що допомагають краще зрозуміти результати моделювання.

Методи машинного навчання, що застосовуються для Класифікація типів Вина [6, 9]:

1. Регресійні моделі:

Лінійна регресія: використовується для встановлення залежності між вмістом хімічних компонентів та типом вина.

Логістична регресія: застосовується для бінарної класифікації, що дозволяє передбачити, чи є вино червоним або білим.

Регресійні дерева: допомагають виявити нелінійні залежності між змінними та типами вина.

2. Нейронні мережі:

Глибокі нейронні мережі (DNN): здатні виявляти складні неявні залежності в даних про хімічний склад вина.

Згорткові нейронні мережі (CNN): можуть використовуватися для аналізу структурованих даних та виявлення патернів.

Рекурентні нейронні мережі (RNN): ефективні для роботи з послідовними даними, якщо аналізуються часові ряди.

4. Методи кластеризації:

Кластерний аналіз: використовується для групування зразків вина зі схожими характеристиками.

K-середніх (K-means): допомагає виявити групи з подібними хімічними характеристиками.

DBSCAN: ефективний для виявлення кластерів різної форми та щільності.

4. Ансамбль моделей:

Random Forest: використовує ансамбль дерев рішень для покращення точності прогнозів.

Gradient Boosting: підвищує точність за рахунок комбінування кількох слабких моделей у сильну.

Вибір конкретної моделі залежить від характеристик даних, доступності ресурсів та конкретних вимог до класифікація типів вина. Важливо проводити експерименти з різними моделями та методами, щоб знайти найефективніше рішення [12, 13].

1.5 Висновки

У першому розділі дослідження розглянуто проблему класифікація типів вина на основі його хімічного складу. Визначено ключові аспекти та фактори, які можуть впливати на тип вина, такі як вміст алкоголю, сульфати, лимонна кислота, pH, щільність, залишковий цукор, хлориди, діоксид сірки, летюча та фіксована кислотність.

Була сформульована постановка задачі класифікації типів вина, визначені цільові змінні та критерії ефективності моделі.

Також було проведено опис датасету, який містить різноманітні хімічні параметри вина та відомий тип кожного зразка (біле чи червоне). Ці дані будуть використовуватись для навчання та тестування моделей класифікація типів вина на основі його хімічного складу.

Зроблено вибір оптимальних інформаційних технологій, які допоможуть ефективно вирішити задачу класифікації типів вина на основі наданих даних.

2 ВИБІР ОПТИМАЛЬНИХ НАЛАШТУВАНЬ АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Веб платформа Kaggle

Kaggle — це популярна платформа для спеціалістів з аналізу даних та машинного навчання, яка надає доступ до численних інструментів, наборів даних і змагальних середовищ для розв'язання складних задач. Вона є чудовим середовищем для розробки, тестування та оцінки моделей машинного навчання завдяки своїй зручності, масштабованості та активній спільноті.

Основні можливості Kaggle:

1. Платформа надає доступ до великої кількості готових для роботи наборів даних, які охоплюють широкий спектр тем — від прогнозування цін акцій до класифікації зображень. На рисунку 2.1 зображено приклад інтерфейсу наборів даних. Для моєї роботи було використано набір даних Wine Quality Data Set, який включає інформацію про хімічний склад вина.

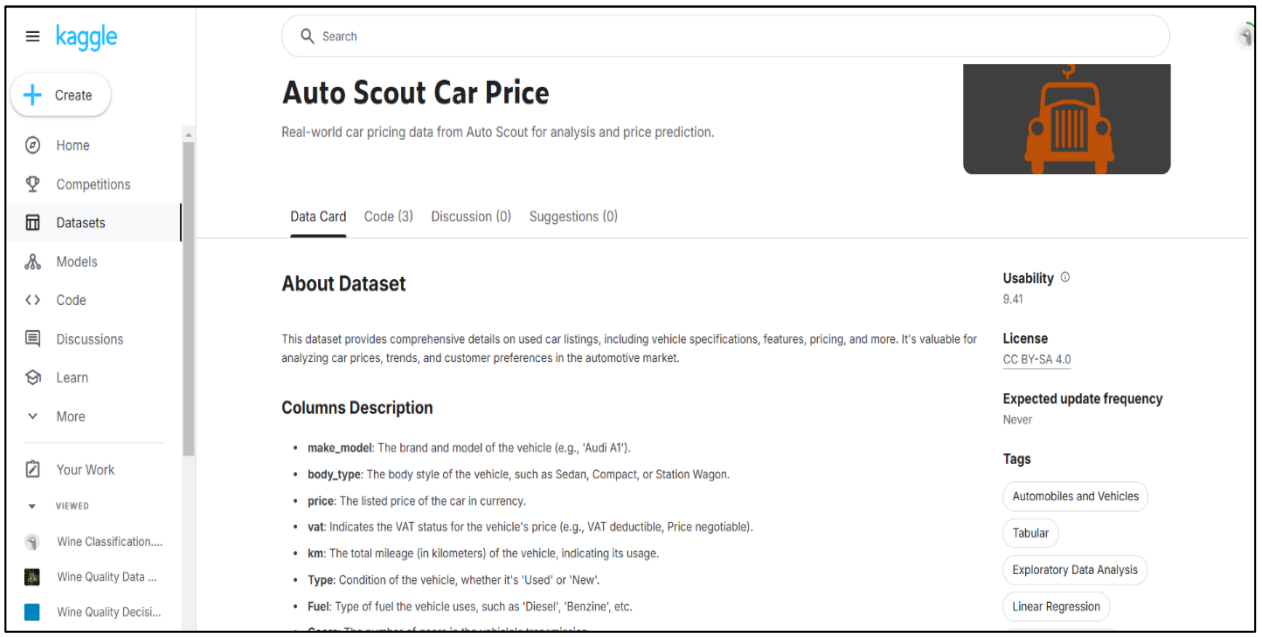


Рисунок 2.1 – Інтерфейс наборів даних на Kaggle [14]

2. Kaggle дозволяє запускати Jupyter-ноутбуки прямо на своїй платформі без необхідності локальної установки програмного забезпечення. На рисунку 2.2

зображено приклад інтерфейсу Jupyter-ноутбуків. Це значно спрощує процес роботи над проектами. У моєму випадку це дозволило швидко тестувати різні моделі машинного навчання та порівнювати їх ефективність.

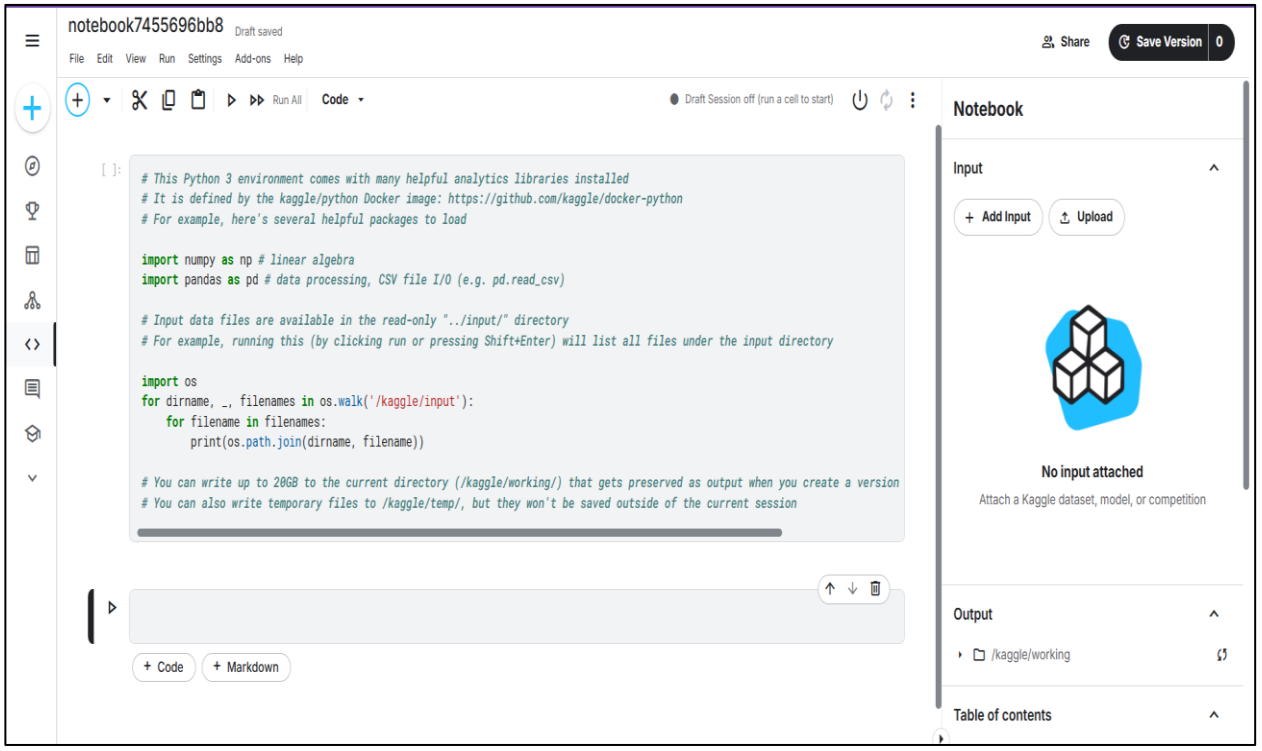


Рисунок 2.2 – Інтерфейс Jupyter-ноутбуків на Kaggle [14]

3. Kaggle відомий своїми змаганнями, в яких учасники змагаються за створення найкращих моделей для вирішення визначених задач. Хоча я не брав участь у змаганнях у межах цієї роботи, інструменти Kaggle Competitions допомагають вдосконалювати свої навички. На рисунку 2.3 зображено приклад інтерфейсу змагань на Kaggle.

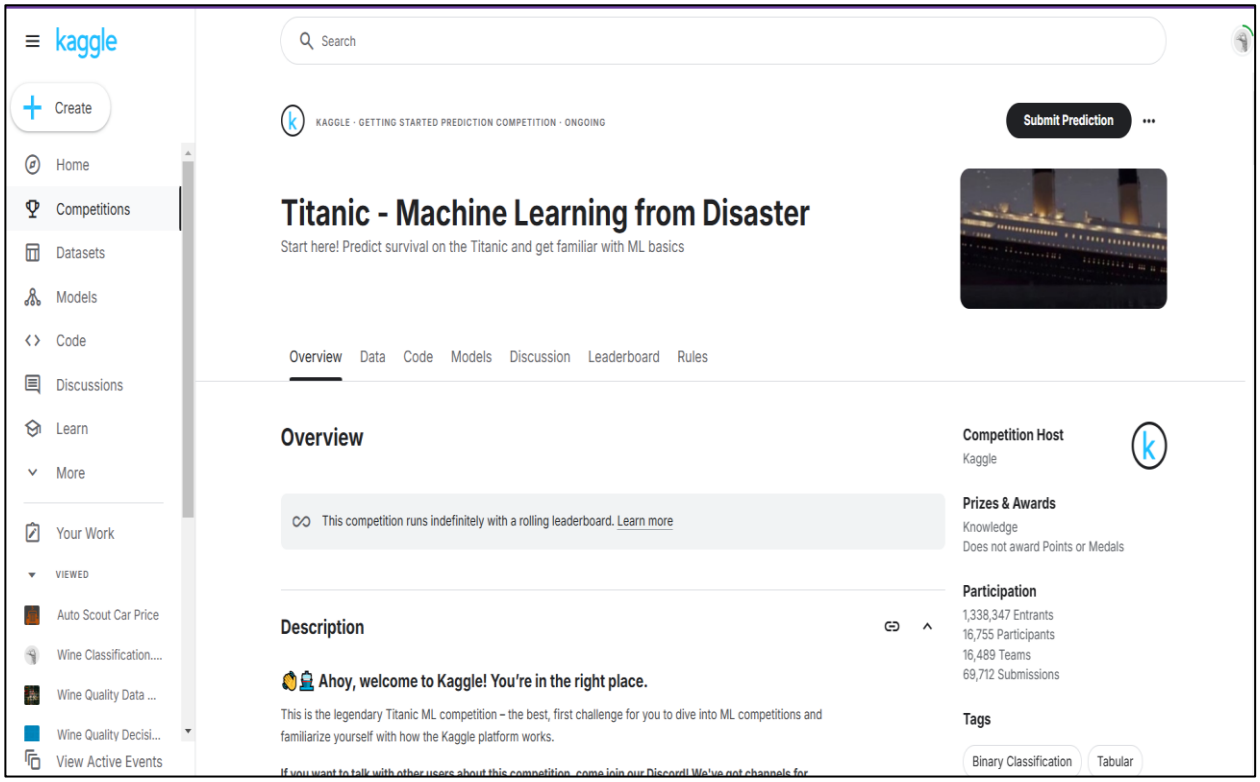


Рисунок 2.3 – Змагання на Kaggle [14]

4. Платформа підтримує основні бібліотеки Python, такі як Pandas, NumPy, Scikit-learn, TensorFlow, що дозволяє ефективно працювати з даними й розробляти моделі машинного навчання. На рисунку 2.4 відображене використання деяких бібліотек.

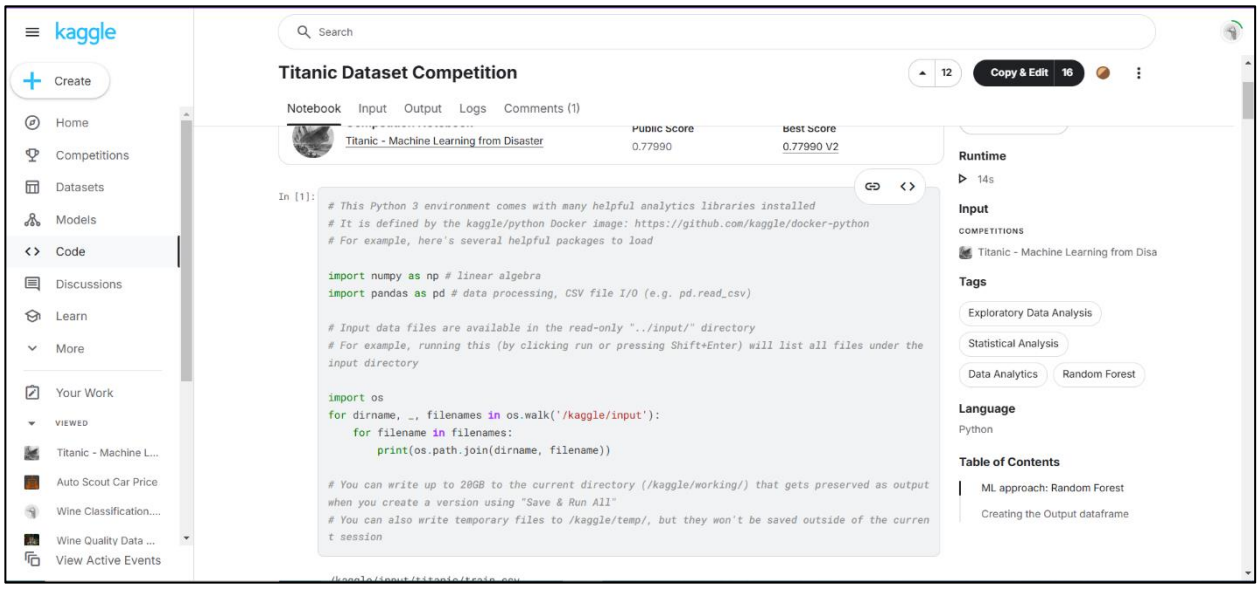


Рисунок 2.4 – Використання популярних бібліотек у ноутбуках Kaggle [14]

5. Kaggle — це також місце, де користувачі можуть взаємодіяти, обмінюватися досвідом, задавати питання і ділитися кодом. Завдяки цьому я зміг знайти кілька корисних ідей для оптимізації своїх моделей. На рисунку 2.6 приклад інтерфейсу форуму спільноти Kaggle.

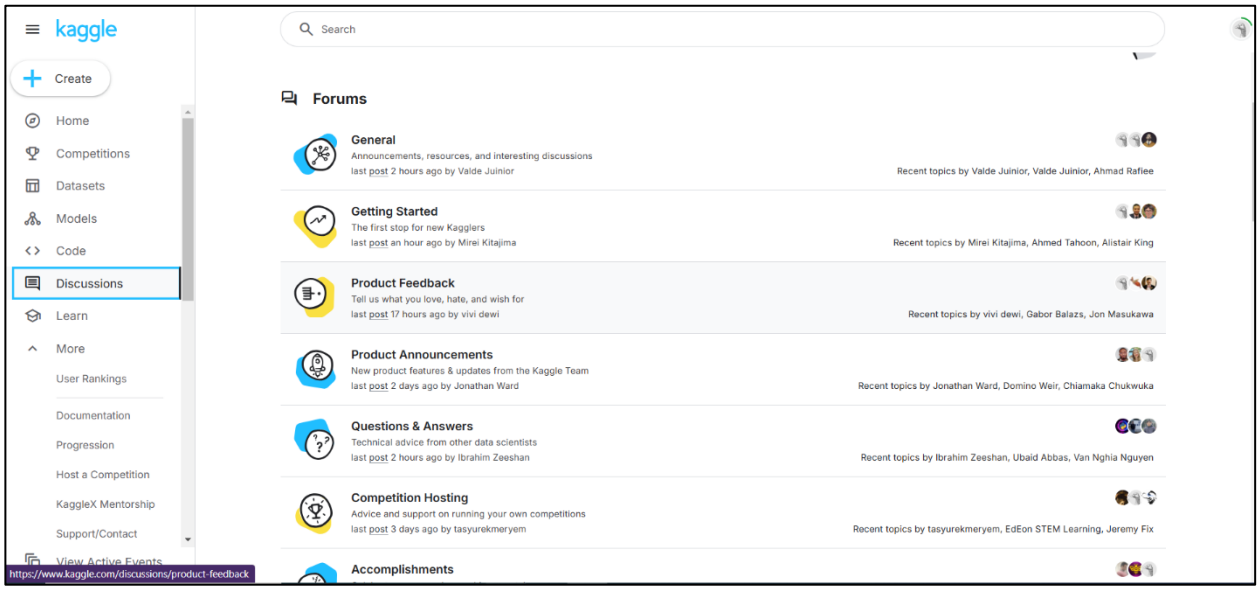


Рисунок 2.5 – Форум спільноти Kaggle [14]

У ході роботи платформа Kaggle використовувалася для таких завдань:

- Пошук і завантаження датасету Wine Quality Data Set.
- Попередня обробка даних та їх аналіз у Jupyter-ноутбуках.
- Тестування моделей XGBoost, Random Forest і Decision Tree.
- Оцінка точності моделей на основі крос-валідації.
- Платформа забезпечила необхідні інструменти для повного циклу розробки моделі, від підготовки даних до отримання фінальних результатів.

2.2 Мова програмування Python

Python — це універсальна та потужна мова програмування, яка набула широкого поширення в галузі аналізу даних, машинного навчання та штучного інтелекту. Її ключова перевага полягає у простоті використання та доступності

багатого екосистемного середовища, що дозволяє вирішувати складні завдання швидко та ефективно. Python не лише легка у вивченні, але й забезпечує високий рівень гнучкості для розробників, дозволяючи створювати як прості скрипти, так і складні системи [15].

Окрім того, Python підтримується на багатьох платформах, включаючи Windows, macOS та Linux, що робить її кросплатформною мовою програмування. Завдяки цьому вона ідеально підходить для створення додатків у різних середовищах без значних змін у коді. Для розробників у сфері аналізу даних це означає можливість легко адаптувати свої проекти до різних умов, не витрачаючи часу на перевірку сумісності.

Завдяки величезній кількості доступних бібліотек Python став стандартом у багатьох галузях. Наприклад, у фінансовій аналітиці Python використовується для створення моделей прогнозування, в медицині — для аналізу медичних зображень, а в промисловості — для оптимізації процесів виробництва. Його здатність обробляти великі обсяги даних у поєднанні зі швидкою розробкою алгоритмів робить Python лідером серед інструментів для роботи з Big Data.

Крім того, Python має зручний інструментарій для візуалізації даних, що є критично важливим для аналітиків. Такі бібліотеки, як Matplotlib і Seaborn, дозволяють створювати наочні графіки та діаграми, які полегшують розуміння складних наборів даних. Це робить Python незамінним у будь-якій роботі, де необхідно доносити результати аналізу до кінцевого користувача.

Ще однією важливою перевагою Python є його інтеграція з іншими мовами програмування, такими як C, C++, R та Java. Це дозволяє розробникам використовувати Python як інструмент для швидкої розробки та тестування, а інші мови — для реалізації високопродуктивних компонентів. У результаті Python стає універсальним рішенням для роботи над проектами будь-якого масштабу.

Переваги Python для аналізу даних та машинного навчання:

1. Python має зрозумілий і лаконічний синтаксис, що дозволяє швидко створювати прототипи моделей. Це особливо важливо для студентів і спеціалістів-

початківців, адже написання коду на Python займає менше часу порівняно з іншими мовами, такими як Java чи C++.

2. Python надає доступ до великої кількості бібліотек для обробки даних, створення моделей машинного навчання та їхньої візуалізації [15]:

- Pandas — дозволяє працювати з таблицями даних, фільтрувати, сортувати, групувати і виконувати інші маніпуляції.
- NumPy — надає ефективні інструменти для роботи з числовими масивами та математичними функціями.
- Matplotlib і Seaborn — бібліотеки для створення графіків і візуалізації даних.
- Scikit-learn — універсальна бібліотека для побудови моделей машинного навчання, таких як класифікація, регресія та кластеризація.
- XGBoost — високопродуктивна бібліотека для реалізації градієнтного бустингу.

3. Python має одну з найбільших і найактивніших спільнот розробників, що дозволяє легко знаходити рішення до проблем, вивчати нові інструменти та обмінюватися досвідом.

4. Python підтримує інтеграцію з іншими мовами програмування та технологіями, що робить його гнучким інструментом для великих проектів.

Параметр	Python	R	Java	MATLAB
Простота синтаксису	Дуже висока	Середня	Низька	Висока
Наявність бібліотек	Велика кількість	Дуже велика кількість	Обмежена	Середня
Популярність	Найвища	Висока	Середня	Середня
Вартість	Безкоштовно	Безкоштовно	Безкоштовно	Платно
Швидкість виконання	Середня	Середня	Висока	Висока
Гнучкість	Дуже висока	Низька	Висока	Низька

Рисунок 2.6 – Порівняння Python з іншими мовами програмування для аналізу даних

Python використовувався для виконання всіх основних етапів класифікації типів вина. Завдяки його можливостям, вдалося вирішити низку завдань, таких як обробка та візуалізація даних, побудова моделей і їхня оцінка.

Для роботи з даними використовувалася бібліотека Pandas. Вона дозволила ефективно фільтрувати дані, видаляти пропуски, нормалізувати та стандартизувати їх. На рисунку 2.7 зразок вхідних даних для класифікації.

Fixed Acidity	Volatile Acidity	Citric Acid	Residual Sugar	Alcohol	Quality
7.4	0.7	0.0	1.9	9.4	5
7.8	0.88	0.0	2.6	9.8	5

у

Рисунок 2.7 – Зразок вхідних даних для класифікації

Для аналізу кореляцій між характеристиками вина було використано Seaborn, що дозволило виявити залежності між кислотністю, рівнем алкоголю та якістю вина [15].

Python продемонстрував себе як ефективний інструмент для реалізації всіх етапів роботи — від обробки даних до класифікації типів вина. Завдяки своїм бібліотекам, він забезпечив точність, швидкість і зручність у роботі.

2.3 Розвідувальний аналіз даних

Проведемо розвідувальний аналіз даних хімічного складу вина. Спершу перевіримо зв’язки між ознаками. На рисунку 2.8 наведено heatmap, на якій видно, що деякі параметри мають кореляцію між собою.

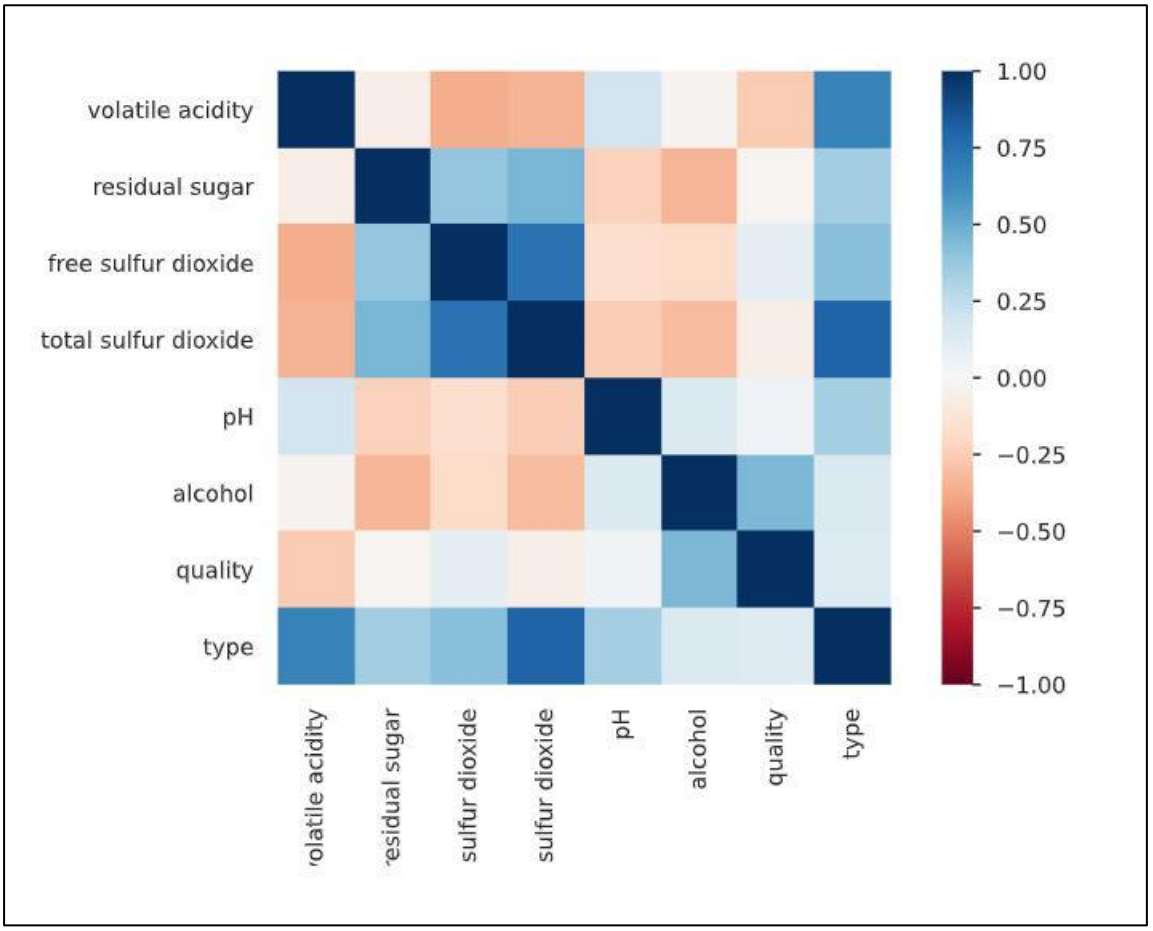


Рисунок 2.8 – Heatmap ознак використаного датасету

З аналізу heatmap можна зробити кілька спостережень:

Вміст алкоголю має помітну кореляцію з якістю вина, що вказує на те, що більш міцні вина зазвичай мають вищу оцінку якості.

Загальний діоксид сірки має негативну кореляцію з pH, що свідчить про зниження рівня кислотності при збільшенні вмісту діоксиду сірки.

Летюча кислотність має слабку негативну кореляцію з залишковим цукром, вказуючи на те, що вина з високим вмістом летючої кислотності зазвичай мають менше залишкового цукру.

Вміст алкоголю має помітну негативну кореляцію з параметром pH, що вказує на те, що вина з вищим вмістом алкоголю мають нижчий pH [16].

Тип вина (біле або червоне) має значну кореляцію з кількома параметрами, такими як вміст алкоголю, залишковий цукор та діоксид сірки, що дозволяє використовувати ці параметри для класифікації типу вина. На рисунку 2.9 міститься опис датасету.

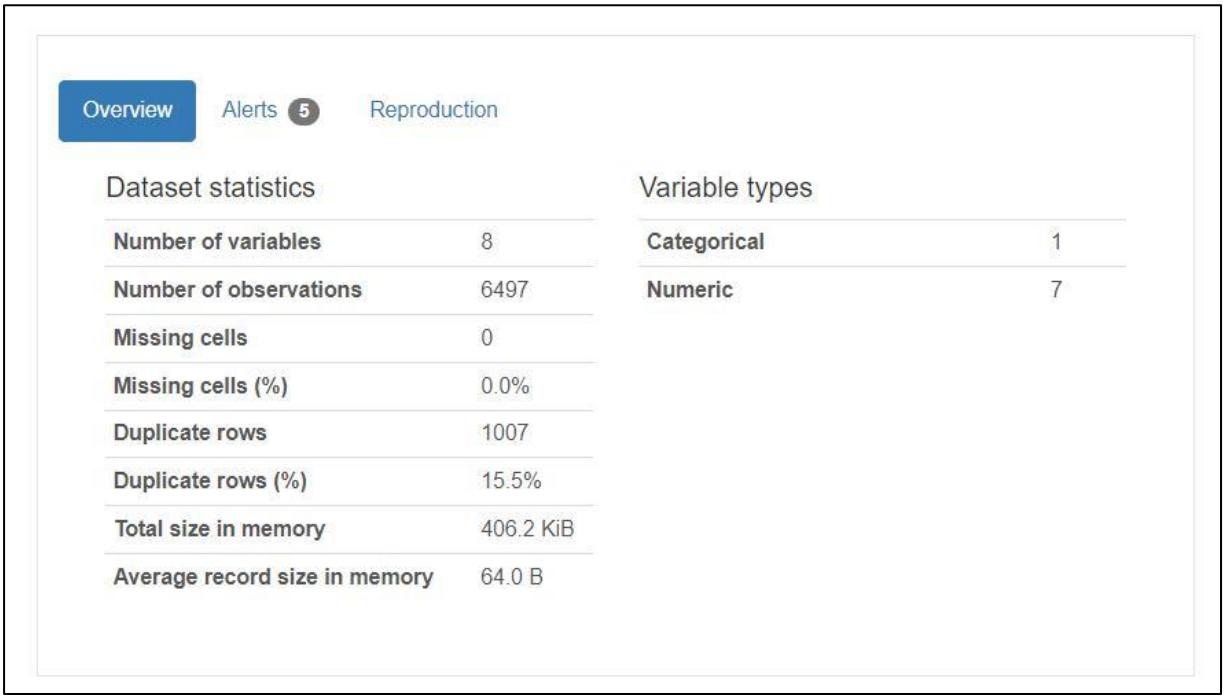


Рисунок 2.9 – Опис датасету

Статистика датасету:

1. Кількість змінних: 8;
2. Кількість спостережень: 6497;
3. Пропущені значення: 0 (0.0%);
4. Дублікати рядків: 1007 (15.5%);
5. Загальний розмір у пам'яті: 406.2 KiB;
6. Середній розмір запису у пам'яті: 64.0 B.

Типи змінних:

1. Категоріальні: 1;
2. Числові: 7.

Ця інформація демонструє, що дані добре структуровані, без пропущених значень, але містять значну кількість дубльованих рядків. На рисунку 2.10 можна побачити взаємозв'язок між "volatile acidity" та "quality" вина.

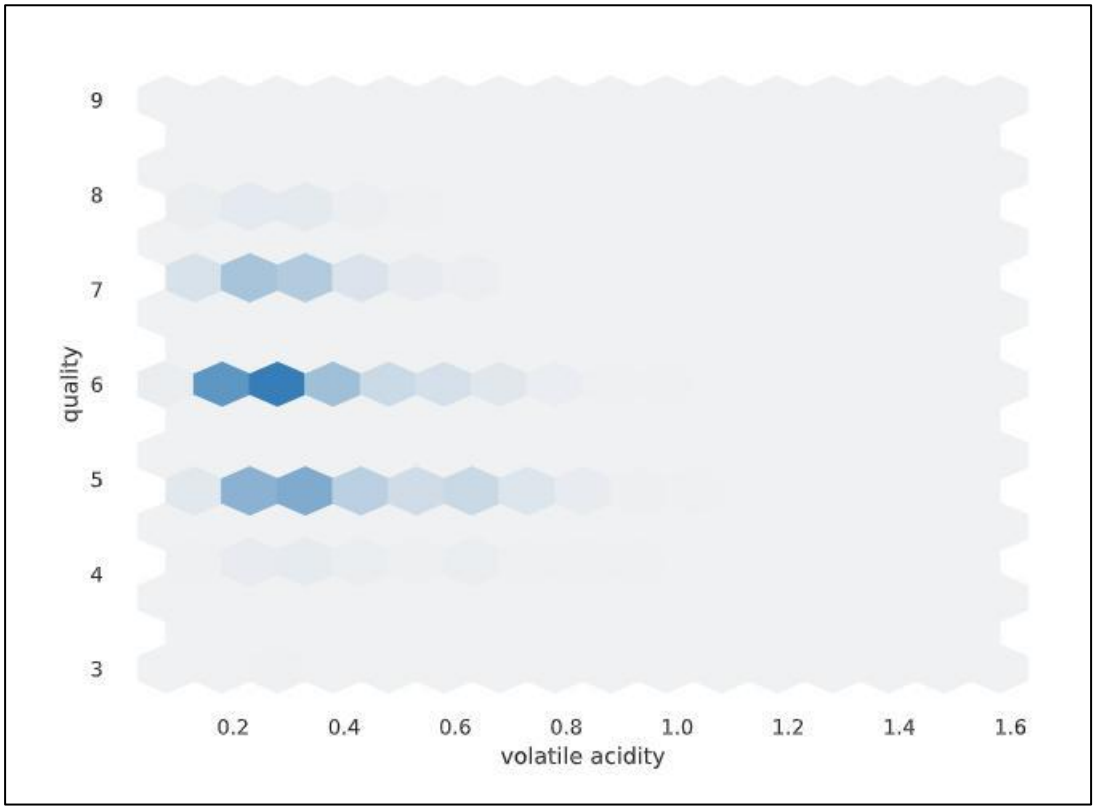


Рисунок 2.10 – Взаємозв'язок між "volatile acidity" та "quality" вина

Якість 6: найбільша кількість зразків вина з якістю 6 має летючу кислотність приблизно від 0.2 до 0.6.

Якість 5: для вина з якістю 5 летюча кислотність також коливається від 0.2 до 0.6.

Якість 7: невелика кількість зразків з летючою кислотністю в діапазоні 0.3 до 0.5.

Це свідчить, що вища летюча кислотність може негативно впливати на якість вина, оскільки зразки з найвищою якістю концентруються при нижчих значеннях кислотності.

На рисунку 2.11 можна побачити взаємозв'язок між "volatile acidity" та "residual sugar" у вині.

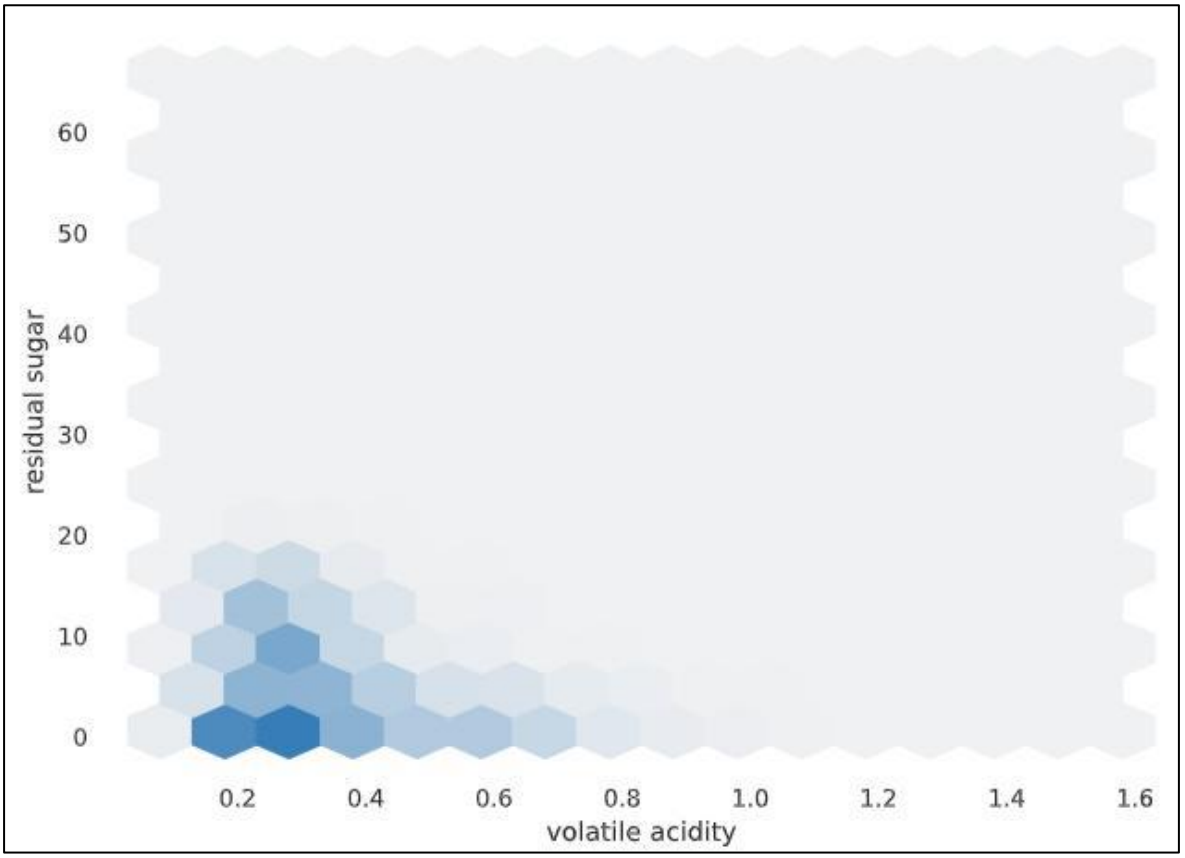


Рисунок 2.11 – Взаємозв'язок між "volatile acidity" та "residual sugar" у вині

Найбільша кількість зразків: зразки з низькою летючою кислотністю (близько 0.2-0.4) мають різний рівень залишкового цукру, з більшістю зразків, що мають низький рівень цукру (менше 10).

Розподіл цукру: зразки з високим рівнем залишкового цукру (більше 20) зустрічаються рідше і зазвичай мають низьку летючу кислотність.

Це свідчить, що більшість вин з низькою летючою кислотністю мають широкий діапазон рівнів залишкового цукру, в той час як вина з високим рівнем цукру зазвичай мають низьку летючу кислотність.

На рисунку 2.12 можна побачити взаємозв'язок між "volatile acidity" та "free sulfur dioxide" у вині.

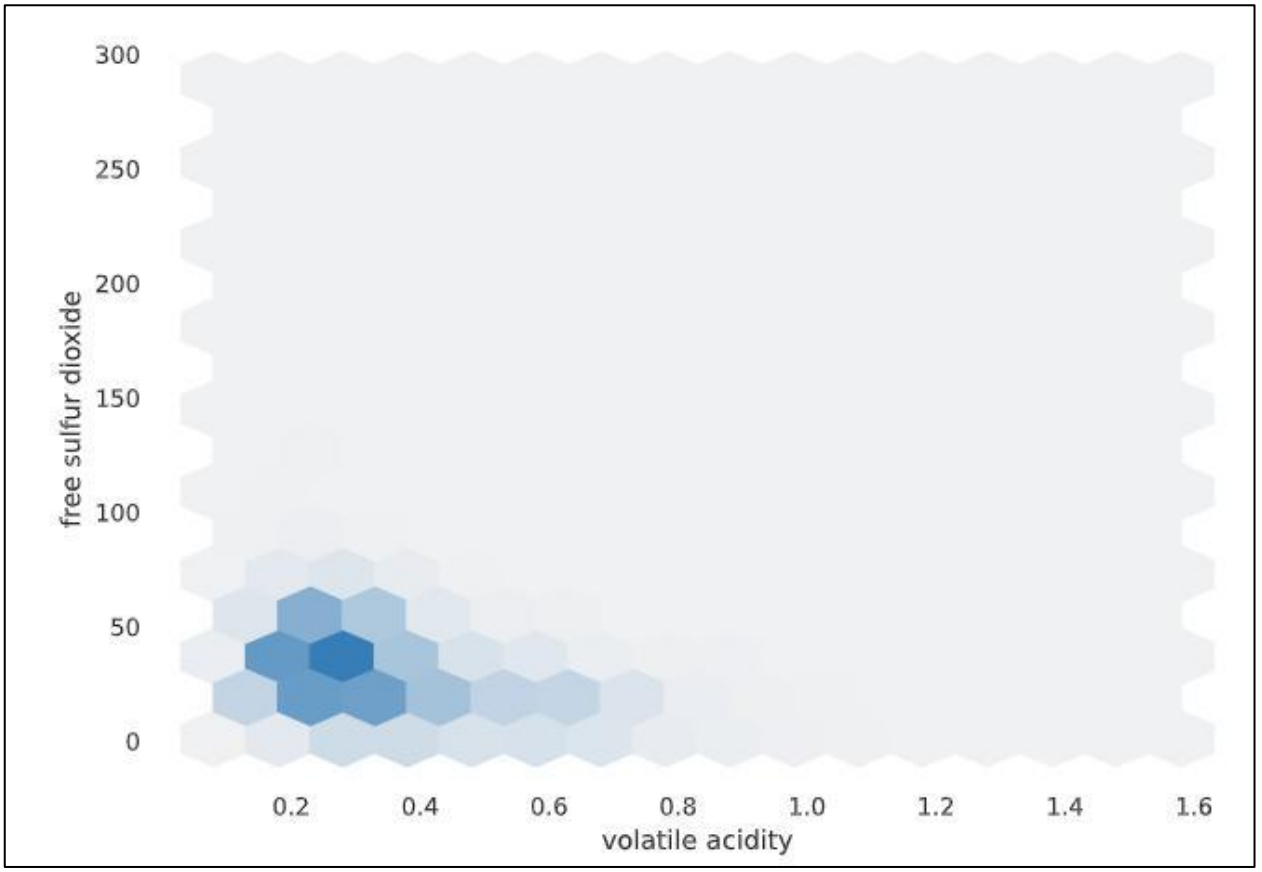


Рисунок 2.12 – Взаємозв'язок між "volatile acidity" та "free sulfur dioxide" у вині

Найбільша кількість зразків: зразки з низькою летючою кислотністю (0.2-0.4) мають широкий діапазон вільного діоксиду сірки (0-50).

Розподіл діоксиду сірки: високий рівень вільного діоксиду сірки (понад 100) рідко зустрічається при високій летючій кислотності.

Це свідчить, що більшість зразків вина з низькою летючою кислотністю можуть мати різні рівні вільного діоксиду сірки, тоді як зразки з високою кислотністю зазвичай мають нижчі рівні вільного діоксиду сірки.

На рисунку 2.13 можна побачити взаємозв'язок між "free sulfur dioxide" та "total sulfur dioxide" у вині.

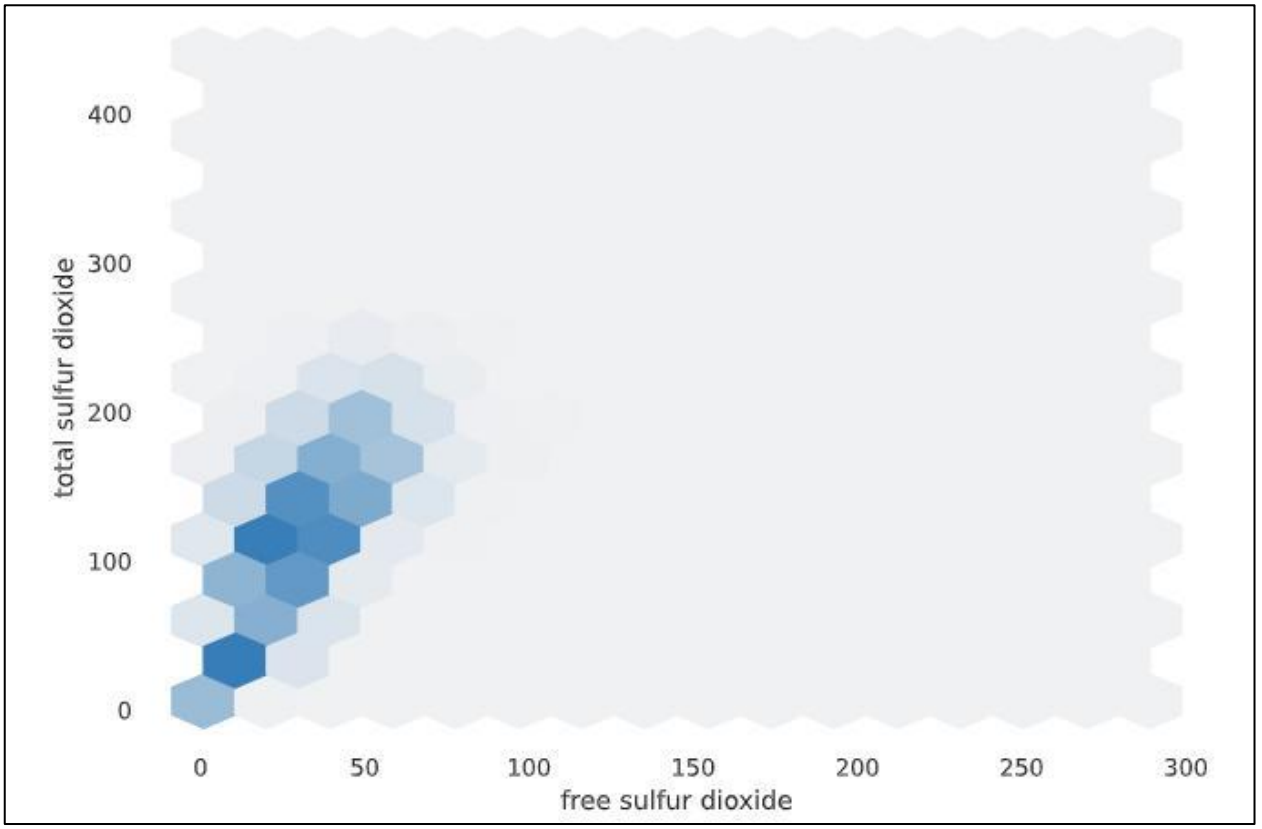


Рисунок 2.13 – Взаємозв'язок між "free sulfur dioxide" та "total sulfur dioxide" у вині

Пряма залежність: існує пряма залежність між рівнями вільного та загального діоксиду сірки. Більшість зразків мають обидва показники в діапазоні до 100.

Концентрація: більшість зразків мають рівні вільного діоксиду сірки до 50 та загального діоксиду сірки до 200.

Це свідчить про те, що збільшення вільного діоксиду сірки супроводжується збільшенням загального діоксиду сірки, що є очікуваним для хімічного складу вина.

На рисунку 2.14 можна побачити взаємозв'язок між "pH" та "alcohol" у вині.

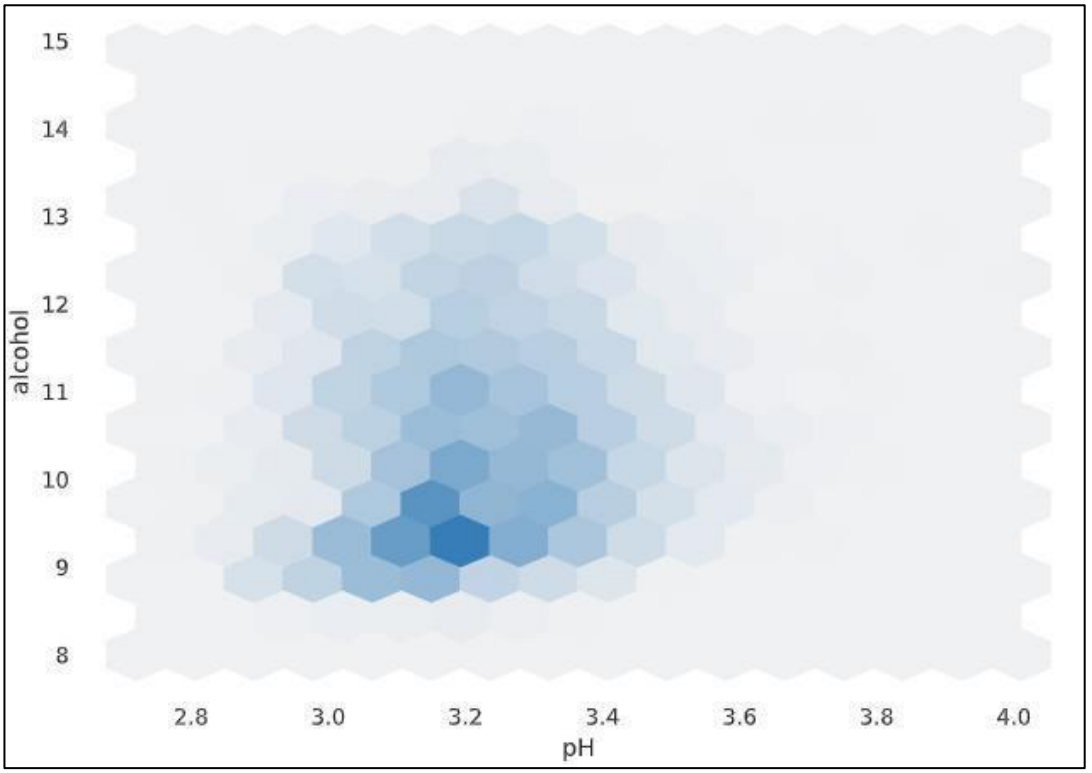


Рисунок 2.14 – Взаємозв'язок між "pH" та "alcohol" у вині

Концентрація зразків: більшість зразків мають значення pH від 3.0 до 3.4 та вміст алкоголю від 9 до 11%.

Розподіл значень: зразки з високим вмістом алкоголю (понад 13%) та значенням pH понад 3.6 є рідкісними.

Це свідчить про те, що більшість вин мають середні значення pH та алкоголю, що відповідає стандартам якості вина.

На рисунку 2.15 можна побачити взаємозв'язок між "residual sugar" та "free sulfur dioxide" у вині.

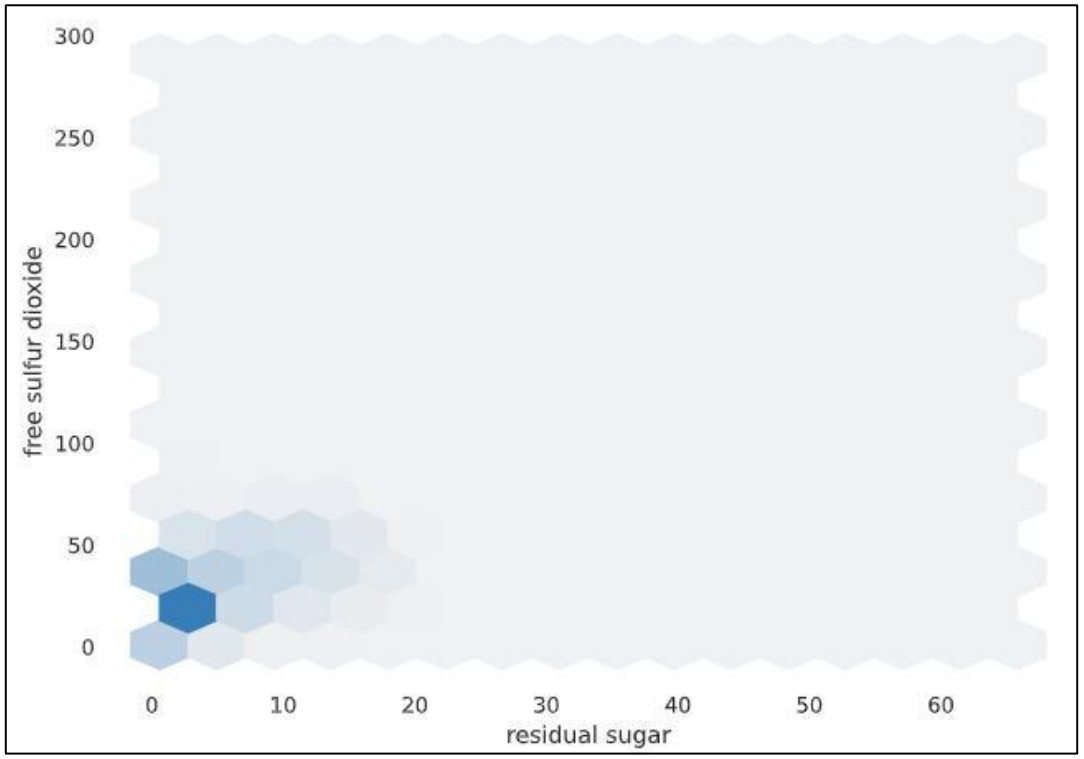


Рисунок 2.15 – Взаємозв'язок між "residual sugar" та "free sulfur dioxide" у вині

Концентрація зразків: більшість зразків мають залишковий цукор (residual sugar) у межах 0–10 г/л і вміст вільного діоксиду сірки (free sulfur dioxide) до 50 мг/л.

Розподіл значень: зразки з вищим вмістом залишкового цукру (понад 30 г/л) та діоксиду сірки (понад 100 мг/л) є рідкісними.

Це свідчить про те, що основна частина вин має низький вміст залишкового цукру та помірний рівень діоксиду сірки, що відповідає типовим характеристикам столових вин.

На рисунку 2.16 можна побачити взаємозв'язок між "quality" та "alcohol" у вині.

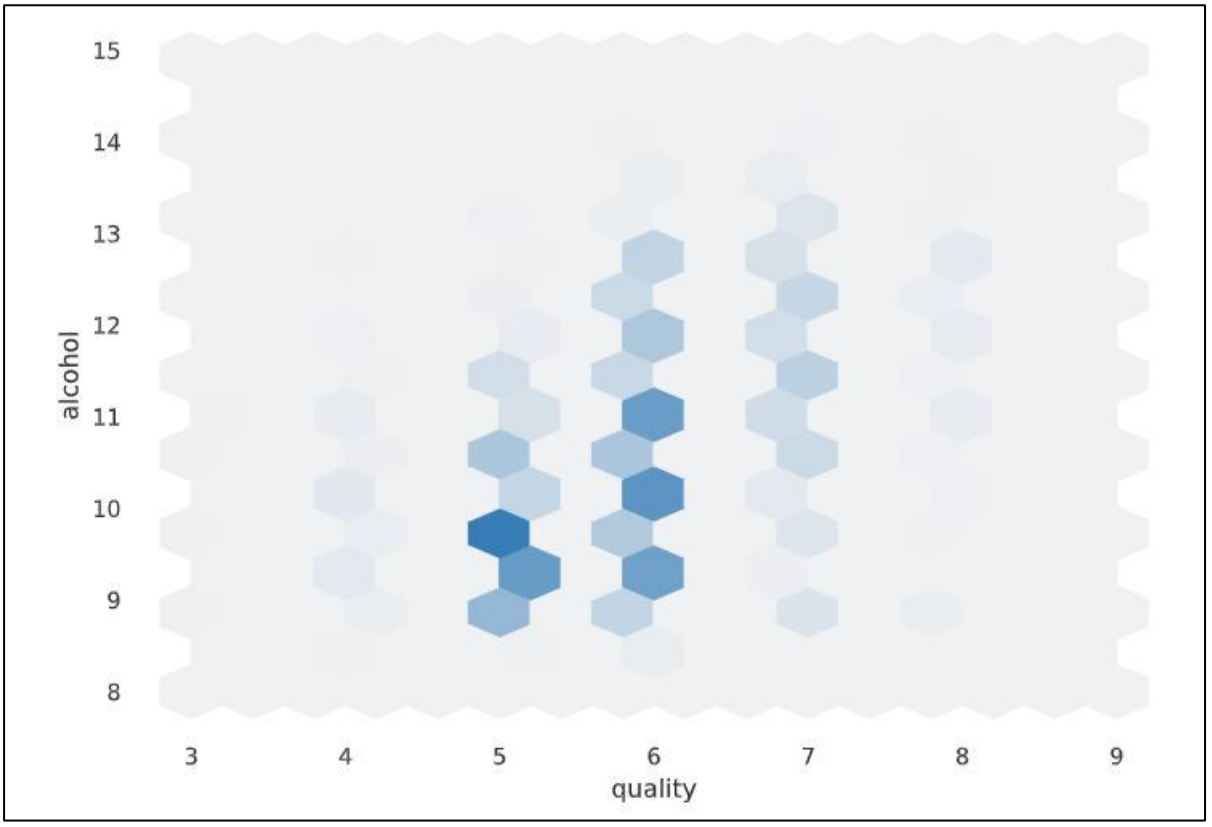


Рисунок 2.16 – Взаємозв'язок між "quality" та "alcohol" у вині

Концентрація зразків: вина з якістю 5 та 6 мають найбільше зразків з вмістом алкоголю від 9 до 11%.

Розподіл якості: зразки з вищим вмістом алкоголю (понад 12%) зазвичай мають якість 6 і вище, а зразки з низьким вмістом алкоголю (менше 9%) рідко мають високу якість.

Це свідчить про те, що вміст алкоголю може бути показником якості вина, з помірними рівнями алкоголю, що частіше зустрічаються в винах середньої якості.

На рисунку 2.17 візуально відображена кількість даних в датасеті.

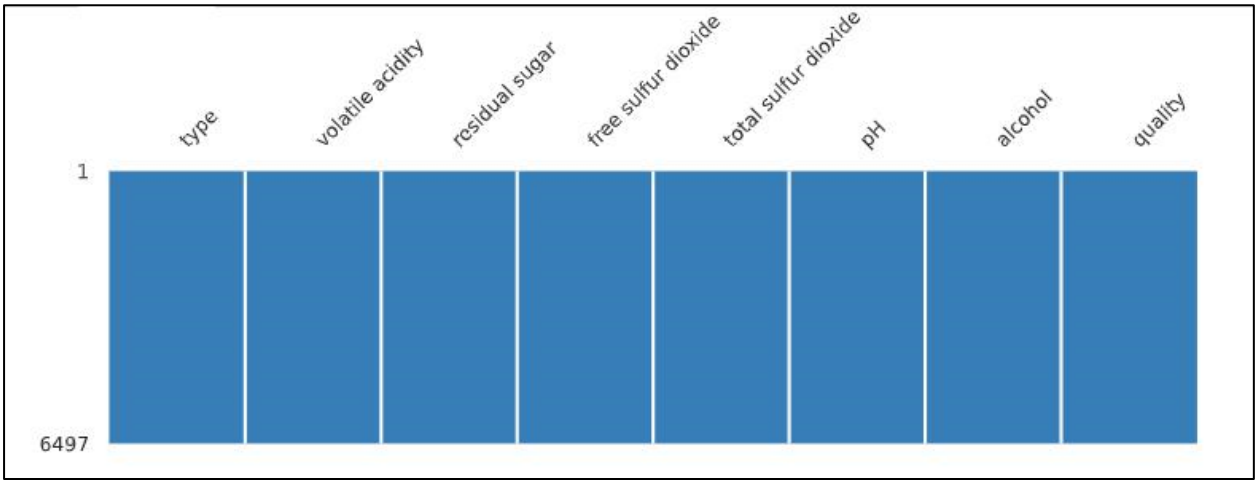


Рисунок 2.17 – Наявність даних у датасеті

Повнота даних: усі стовпці (тип, летюча кислотність, залишковий цукор, вільний діоксид сірки, загальний діоксид сірки, pH, вміст алкоголю та якість) заповнені для всіх 6497 зразків.

Цілісність даних: всі стовпці мають 100% заповненість без пропущених значень.

Це свідчить про відмінну якість та повноту даних, що забезпечує надійність подальшого аналізу та моделювання.

2.4 Масштабування ознак

Для проведення аналізу та побудови моделей булостандартизовано датасет. Це означає, що всі ознаки в наборі даних піддали процесу стандартизації, що дозволило забезпечити однаковий масштаб для всіх змінних [16, 17]. Такий підхід застосовано для досягнення кількох важливих цілей:

1. Однорідність даних: завдяки стандартизації, ознаки не відрізняються значно за величиною та діапазоном. Це робить дані більш порівнюваними між собою і дозволяє уникнути домінування ознак з великими числовими значеннями над іншими під час навчання моделі.

2. Покращення роботи алгоритмів: багато алгоритмів машинного навчання, таких як градієнтний бустинг та випадкові ліси, працюють краще з масштабованими даними. Стандартизація допомагає уникнути проблем із

чисельною стабільністю та прискорює процес оптимізації, що в результаті підвищує продуктивність моделей.

3. Поліпшення збіжності: стандартизація сприяє швидшій і стабільнішій збіжності алгоритмів градієнтного спуску, які використовуються у моделях градієнтного бустингу та нейронних мережах. Це дозволяє зменшити час на навчання моделей та підвищити точність прогнозів.

Процес стандартизації включає перетворення кожної ознаки таким чином, щоб її середнє значення було рівним нулю, а стандартне відхилення одиниці. Це забезпечує, що всі ознаки мають однакову вагу під час навчання моделей.

Таким чином, масштабування ознак стало важливим етапом у підготовці даних для аналізу, що забезпечило більш точні та надійні результати класифікації. Використання стандартизованого датасету дозволило моделям краще навчитися та зменшити ризик помилок, пов'язаних з різною шкалою вимірювань ознак.

2.5 Вибір моделей

Загальні відомості про кожну з моделей що планується використати:

XGBoost Classifier використовує вдосконалений метод бустингу, який підвищує точність за допомогою оптимізації гіперпараметрів та регуляризації. Цей алгоритм застосовується для регресійних задач та оптимізує функцію втрат для покращення точності класифікації [18].

Основні характеристики:

1. Ансамблевий метод: використовує кілька дерев, де кожне наступне дерево намагається виправити помилки попередніх.
2. Градієнтний бустинг: застосовується для регресійних задач, оптимізуючи функцію втрат за допомогою градієнтного спуску.
3. Висока точність: завдяки регуляризації та оптимізації, метод забезпечує високу точність класифікації.

На рисунку 2.18 можна побачити конкретну реалізацію XGBoost Classifier.

In [29]:

```
#XGBoost Regressor
cv_train = KFold(n_splits=3, shuffle=True, random_state=42)
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 4, 5],
    'subsample': [0.8, 0.9],
    'learning_rate': [0.01, 0.05, 0.1],
    'gamma': [0, 1, 5],
    'min_child_weight': [1, 3, 5],
    'colsample_bytree': [0.8, 0.9],
    'alpha': [0, 0.1, 1],
    'lambda': [1, 2]
}

xgb = XGBRegressor(random_state=42)
xgb.set_params(early_stopping_rounds=10)
xgb_CV = RandomizedSearchCV(xgb, param_distributions=param_grid, n_iter=50, cv=cv_train, verbose=F
alse, random_state=42, n_jobs=-1)
xgb_CV.fit(train, target_train, eval_set=[(valid, target_valid)], verbose=False)

print(xgb_CV.best_params_)

y_train_xgb = xgb_CV.predict(train)
r2_score_acc = round(r2_score(target_train, y_train_xgb) * 100, 1)
print(f'Accuracy of XGBoost Regressor model training is {r2_score_acc}')
result.loc[result['model'] == 'XGBoost Regressor', 'train_score'] = r2_score_acc

y_val_xgb = xgb_CV.predict(valid)
r2_score_acc_valid = round(r2_score(target_valid, y_val_xgb) * 100, 1)
result.loc[result['model'] == 'XGBoost Regressor', 'valid_score'] = r2_score_acc_valid
print(f'Accuracy of XGBoost Regressor model prediction for valid dataset is {r2_score_acc_valid}')
```

Рисунок 2.18 – XGBoost Classifier

1. n_estimators: кількість дерев у моделі. Значення (50, 100, 200) дозволяють вибрати оптимальне число дерев для збалансування між точністю та продуктивністю.

2. max_depth: максимальна глибина дерев. Значення (3, 4, 5) допомагають контролювати складність моделі, запобігаючи перенавчанню.

3. subsample: частка вибірки для побудови кожного дерева. Значення (0.8, 0.9) використовуються для запобігання перенавчанню шляхом вибору випадкових підмножин даних.

4. learning_rate: коефіцієнт навчання. Значення (0.01, 0.05, 0.1) допомагають знижувати вагу нових дерев, що робить процес навчання більш стійким.

5. gamma: мінімальне зменшення показника втрат для поділу вузла. Значення (0, 1,5) використовуються для запобігання поділам, які не покращують модель.

6. min_child_weight: мінімальна кількість зразків у листі. Значення (1, 3, 5) допомагають уникати надмірних поділів, які можуть призвести до перенавчання.

7. colsample_bytree: частка ознак, що використовуються для побудови кожного дерева. Значення (0.8, 0.9) використовуються для зменшення кореляції між деревами, покращуючи узагальнюваність моделі.

8. alpha: регуляризаційний параметр L1. Значення (0, 0.1, 1) допомагають зменшити складність моделі.

9. lambda: регуляризаційний параметр L2. Значення (1, 2) допомагають контролювати ваги моделі, запобігаючи перенавчанню.

Ці параметри забезпечують оптимальну конфігурацію моделі, дозволяючи досягти високої точності та стабільності прогнозів.

Random Forest Classifier використовує ансамбль рішень, де кожне дерево прогнозує значення на основі випадкової підмножини ознак і вузлів дерева. Цей метод застосовує безліч дерев для усереднення результатів і отримання остаточного прогнозу, оцінюючи важливість ознак, аналізуючи частоту їх використання для розділення вузлів дерев.

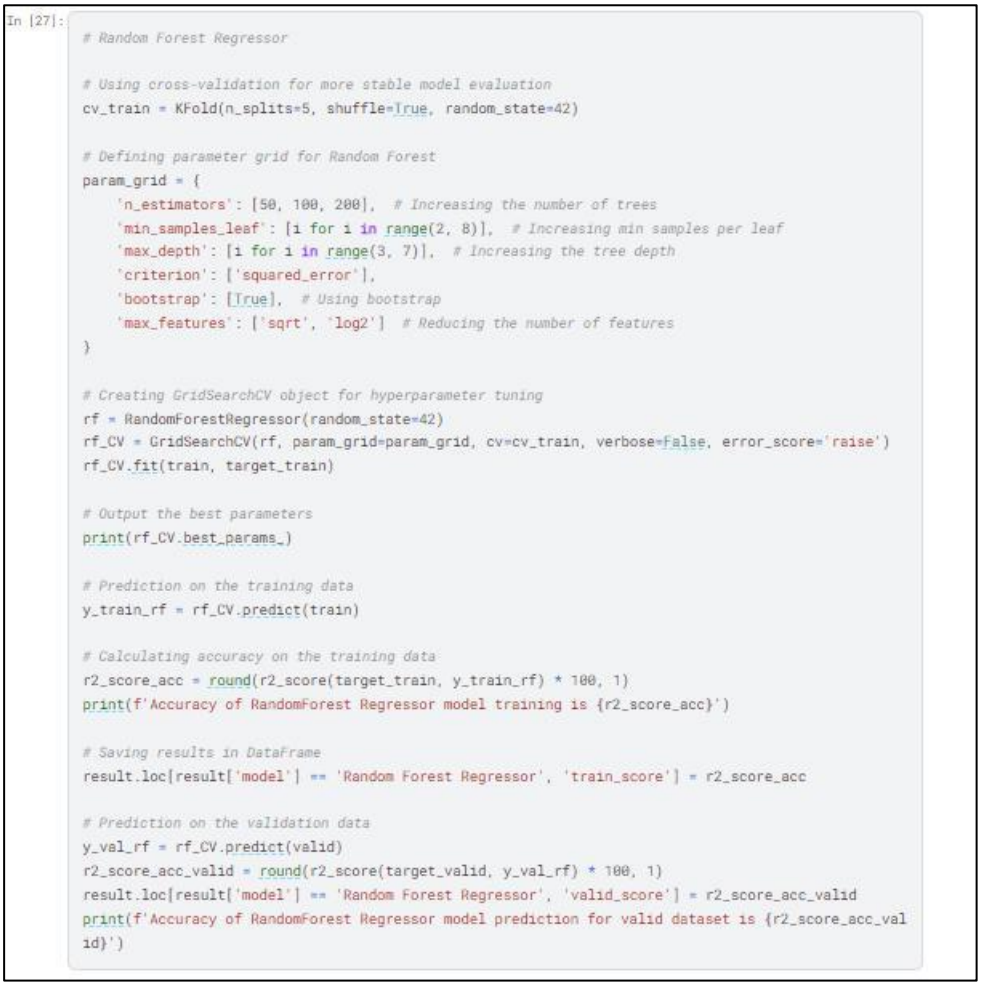
Основні характеристики:

1. Ансамблевий метод: використовує багато дерев рішень, кожне з яких прогнозує значення на основі випадкової підмножини даних.

2. Усереднення результатів: поєднує результати всіх дерев для отримання точного прогнозу.

3. Оцінка важливості ознак: визначає важливість ознак за частотою їх використання для розділення вузлів дерев.

На рисунку 2.19 можна побачити конкретну реалізацію Random Forest Classifier.



```
In [27]:
# Random Forest Regressor

# Using cross-validation for more stable model evaluation
cv_train = KFold(n_splits=5, shuffle=True, random_state=42)

# Defining parameter grid for Random Forest
param_grid = {
    'n_estimators': [50, 100, 200],  # Increasing the number of trees
    'min_samples_leaf': [i for i in range(2, 8)],  # Increasing min samples per leaf
    'max_depth': [i for i in range(3, 7)],  # Increasing the tree depth
    'criterion': ['squared_error'],
    'bootstrap': [True],  # Using bootstrap
    'max_features': ['sqrt', 'log2']  # Reducing the number of features
}

# Creating GridSearchCV object for hyperparameter tuning
rf = RandomForestRegressor(random_state=42)
rf_CV = GridSearchCV(rf, param_grid=param_grid, cv=cv_train, verbose=False, error_score='raise')
rf_CV.fit(train, target_train)

# Output the best parameters
print(rf_CV.best_params_)

# Prediction on the training data
y_train_rf = rf_CV.predict(train)

# Calculating accuracy on the training data
r2_score_acc = round(r2_score(target_train, y_train_rf) * 100, 1)
print(f'Accuracy of RandomForest Regressor model training is {r2_score_acc}')

# Saving results in DataFrame
result.loc[result['model'] == 'Random Forest Regressor', 'train_score'] = r2_score_acc

# Prediction on the validation data
y_val_rf = rf_CV.predict(valid)
r2_score_acc_valid = round(r2_score(target_valid, y_val_rf) * 100, 1)
result.loc[result['model'] == 'Random Forest Regressor', 'valid_score'] = r2_score_acc_valid
print(f'Accuracy of RandomForest Regressor model prediction for valid dataset is {r2_score_acc_val
id}')
```

Рисунок 2.19 – RandomForestClassifier

1. n_estimators: кількість дерев (50, 100, 200). Збільшення кількості дерев може покращити точність, але збільшує час навчання.

2. min_samples_leaf: мінімальна кількість зразків у листі (1, 2, 4, 6, 8). Це допомагає уникнути надмірного поділу.

3. max_depth: максимальна глибина дерев (3, 5, 7). Контролює складність моделі, запобігаючи перенавчанню.

4. criterion: показник якості розділення ('squared_error'). Використовується для мінімізації помилок.

5. bootstrap: використання bootstrap (True). Допомагає зменшити варіативність та підвищити стабільність моделі.

6. max_features: кількість ознак для кожного поділу ('sqrt', 'log2'). Зменшення кількості ознак допомагає уникнути перенавчання.

Decision Tree Classifier побудований на основі дерева рішень, яке розбиває дані на категорії для класифікації числових значень. Виконує розділення на основі порівняння ознак і встановлення критерію розділення, рекурсивно розбиваючи дані на підмножини та прогнозуючи, використовуючи середнє значення цільової змінної в кожному листі дерева [19].

Основні характеристики:

1. Модель дерева рішень: виконує розділення даних на категорії на основі порівняння ознак.

2. Рекурсивне розділення: розбиває дані на підмножини на основі критерію розділення.

3. Простота та інтерпретація: легко інтерпретується і розуміється, але може мати тенденцію до перенавчання.

На рисунку 2.20 можна побачити конкретну реалізацію Decision Tree Classifier.

In [25]:
```
# Decision Tree Regressor
decision_tree = DecisionTreeRegressor()
param_grid = {
    'min_samples_leaf': [1, 2, 4, 6, 10],
    'max_depth': [3, 5, 7, 10, 15],
    'min_samples_split': [2, 5, 10, 15, 20],
    'max_features': ['auto', 'sqrt', 'log2', None]
}


decision_tree_CV = GridSearchCV(decision_tree, param_grid=param_grid, cv=cv_train, verbose=False)
decision_tree_CV.fit(train, target_train)
print("Best parameters for Decision Tree: ", decision_tree_CV.best_params_)


y_train_decision_tree = decision_tree_CV.predict(train)


r2_score_acc = round(r2_score(target_train, y_train_decision_tree)*100, 1)
print(f'Accuracy of Decision Tree model training is: {r2_score_acc}')


result.loc[result['model'] == 'Decision Tree Regressor', 'train_score'] = r2_score_acc


y_val_decision_tree = decision_tree_CV.predict(valid)
r2_score_acc_valid = round(r2_score(target_valid, y_val_decision_tree)*100, 1)
result.loc[result['model'] == 'Decision Tree Regressor', 'valid_score'] = r2_score_acc_valid
print(f'Accuracy of Decision Tree on valid data: {r2_score_acc_valid}')
```

Рисунок 2.20 – DecisionTreeClassifier

1. min_samples_leaf: значення (1, 2, 4, 6, 10) встановлені для уникнення перенавчання, забезпечуючи мінімальну кількість зразків у кожному листі дерева. Це до-помагає зробити дерево менш чутливим до шуму в даних.

2. max_depth: значення (3, 5, 7, 10, 15) обрані для контролю глибини дерева. Обмеження глибини допомагає запобігти надмірному підгонці моделі до тренувальних даних, зберігаючи баланс між точністю та узагальнюваністю.

3. min_samples_split: значення (2, 5, 10, 15, 20) використовуються для визначення мінімальної кількості зразків, необхідних для поділу вузла. Це допомагає за-побігти надмірному поділу вузлів, забезпечуючи стабільність та узагальнюваність моделі.

4. max_features: значення ('auto', 'sqrt', 'log2', None) обрані для визначення кількості ознак, що використовуються для поділу кожного вузла. Це дозволяє знайти оптимальний баланс між продуктивністю та точністю, зменшуючи ризик перенавчання.

Ці моделі будуть використані для класифікації типів вина на основі хімічного складу, забезпечуючи різні підходи до обробки даних та їх аналізу.

2.6 Висновки

У другому розділі дослідження проведено розвідувальний аналіз даних, що дозволило оцінити взаємозв'язки між різними параметрами хімічного складу вина.

За допомогою heatmap було ідентифіковано основні кореляції, які впливають на тип вина, що стало важливим кроком для подальшого моделювання.

Також було обрано та детально розглянуто оптимальні моделі для виконання задачі класифікації типів вина. Розглянуто моделі GradientBoostingClassifier, RandomForestClassifier та DecisionTreeClassifier, кожна з яких має свої переваги і підходи до обробки та аналізу даних.

На основі проведеного аналізу та вибраних моделей буде здійснено подальше класифікації типів вина, що дозволить підвищити точність і надійність результатів, а також оптимізувати виробничі процеси у виноробстві

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ КЛАСИФІКАЦІЇ ТИПУ ВИНА

3.1 Розроблення інформаційної технології класифікації типу вина

Розробка інформаційної технології класифікації типів вина передбачає поетапну реалізацію процесу, що включає аналіз даних, вибір оптимальних моделей, навчання, тестування та оцінку результатів. Задля досягнення мети створюється програмний інструмент, побудований на основі об'єктно-орієнтованої архітектури, яка забезпечує модульність і зручність масштабування.

Процес класифікації починається з виконання розвідувального аналізу даних (EDA). Цей етап включає ідентифікацію ключових характеристик, аналіз кореляцій між ознаками та оцінку якості вихідних даних. Далі дані поділяються на тренувальний, валідаційний і тестовий набори, що дозволяє оцінити продуктивність моделі на різних етапах її оптимізації.

Ключовим етапом є вибір моделей машинного навчання, які можуть включати алгоритми, такі як XGBoost, Random Forest і Decision Tree. Моделі навчаються на тренувальному наборі даних, а їхня ефективність перевіряється на валідаційному наборі. На основі точності класифікації обирається найкраща модель, яка після цього додатково тренується на об'єднаному наборі тренувальних і валідаційних даних. Завершальний етап — це оцінка точності класифікації на тестовому наборі даних, що є остаточним критерієм якості моделі.

Для реалізації інформаційної технології була створена UML-діаграма класів, яка описує основну архітектуру розробленого програмного забезпечення. Головний контролер (MainController) управляє основним робочим процесом через метод run_pipeline(). Основний функціонал забезпечується класом WineClassifier, який відповідає за завантаження даних, навчання моделі та виконання прогнозування. Окремі підсистеми, такі як FeatureEngineer (інженерія ознак), EDA (розвідувальний аналіз даних), TreeModel (побудова моделей) та DataVisualizer (візуалізація), реалізують спеціалізовані задачі.

На наступній UML-діаграмі зображено основні класи та взаємозв'язки між ними.

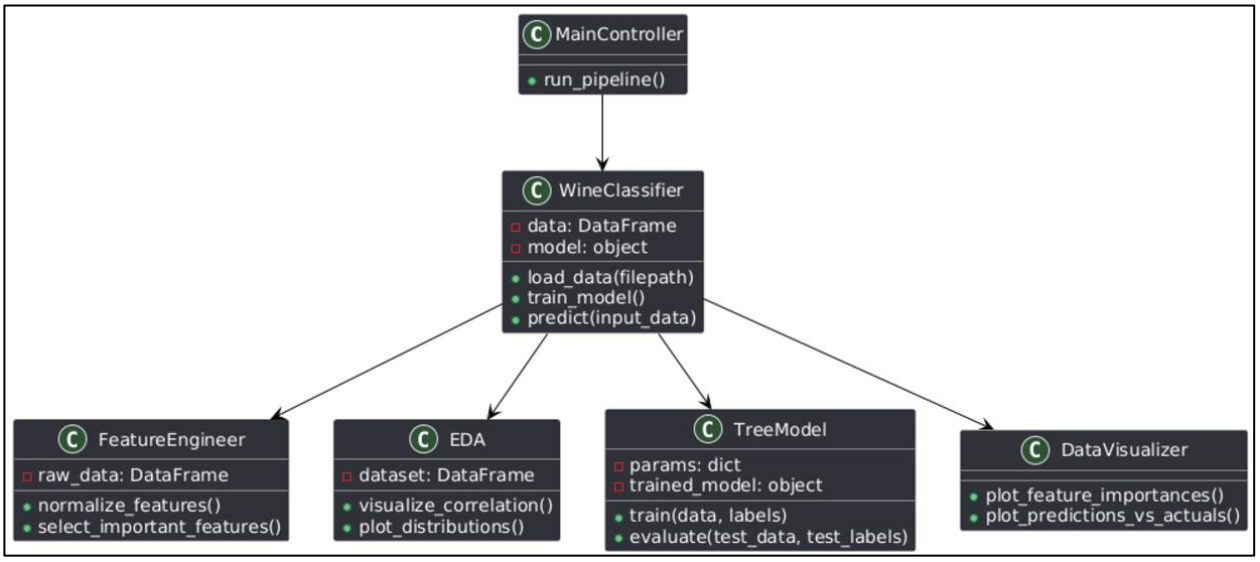


Рисунок 3.1 – UML діаграма класів

Клас FeatureEngineer займається нормалізацією ознак та відбором найбільш інформативних параметрів. EDA виконує аналіз розподілів даних і візуалізацію кореляцій між ознаками. TreeModel відповідає за тренування моделей і їхню оцінку. DataVisualizer дозволяє графічно представляти важливість ознак і порівнювати результати прогнозів із реальними значеннями. Така модульна архітектура дозволяє легко розширювати та модифікувати систему в майбутньому.

Для більшої наочності основні кроки роботи були структуровані у вигляді блок-схеми, наведеної нижче.

Детальний опис етапів блок-схеми:

- Завантаження та попередня обробка даних

На першому етапі дані завантажуються у вигляді таблиць (DataFrame). Попередня обробка включає видалення або заповнення пропущених значень, а також приведення даних до стандартного формату, що є критично важливим для подальшої роботи з алгоритмами машинного навчання.

- Чи дані чисті?

На цьому етапі виконується перевірка даних на наявність пропущених значень, дублікатів і некоректних записів. Якщо дані чисті, вони передаються для виконання розвідувального аналізу (EDA). У випадку наявності помилок виконується їх обробка.

- Розвідувальний аналіз даних (EDA)

Проводиться дослідження структури даних, аналіз розподілів ознак, виявлення кореляцій між параметрами та пошук потенційних аномалій. На цьому етапі також створюються візуалізації, які дозволяють краще зрозуміти залежності між ознаками та їхню значущість.

- Обробка та стандартизація даних

Дані масштабуються до одного діапазону, щоб уникнути диспропорцій у моделюванні через різницю в одиницях вимірювання. Наприклад, хімічні показники, що мають різні масштаби, можуть бути стандартизовані методом Z-оцінок.

- Розділення даних на тренувальний, валідаційний та тестовий набори

Дані діляться на три набори: тренувальний (train), валідаційний (validation) і тестовий (test). Це забезпечує коректність навчання та оцінки моделі на незалежних даних.

- Навчання моделей класифікації

Виконується навчання кількох моделей класифікації, таких як DecisionTree, Random Forest, XGBoost, і т.д. Кожна модель тренується на тренувальному наборі, а її точність перевіряється на валідаційному.

- Перехресна перевірка (cross-validation)

Для запобігання перенавчанню моделі проводиться перехресна перевірка. Цей метод дозволяє оцінити, як модель буде працювати на нових, невідомих даних.

- Оцінка продуктивності моделі

На цьому етапі для кожної моделі обчислюються метрики точності, такі як F1-міра, точність (accuracy), повнота (recall) та інші. Оцінка продуктивності дозволяє порівняти моделі між собою.

- Вибір найкращої моделі та налаштування гіперпараметрів

Найкраща модель обирається на основі результатів оцінки. Для подальшого підвищення її ефективності виконуються експерименти з налаштуванням гіперпараметрів, таких як глибина дерева, кількість дерев у лісі, рівень регуляризації тощо.

- Прогнозування на тестових даних

Після навчання та налаштування найкраща модель використовується для прогнозування типів вина на тестових даних.

- Візуалізація результатів прогнозування

Результати моделі відображаються у вигляді графіків, що показують реальні значення та прогнозовані. Також візуалізуються важливі ознаки, які найбільше вплинули на результат класифікації.

- Побудова графіків важливості ознак

Завдяки візуалізації важливості ознак можна визначити, які параметри хімічного складу вина найбільше впливають на його класифікацію. Це допомагає покращити інтерпретацію моделі та виявити ключові фактори для оптимізації виробництва вина. На рисунку 3.2 зображена UML діаграма діяльності.

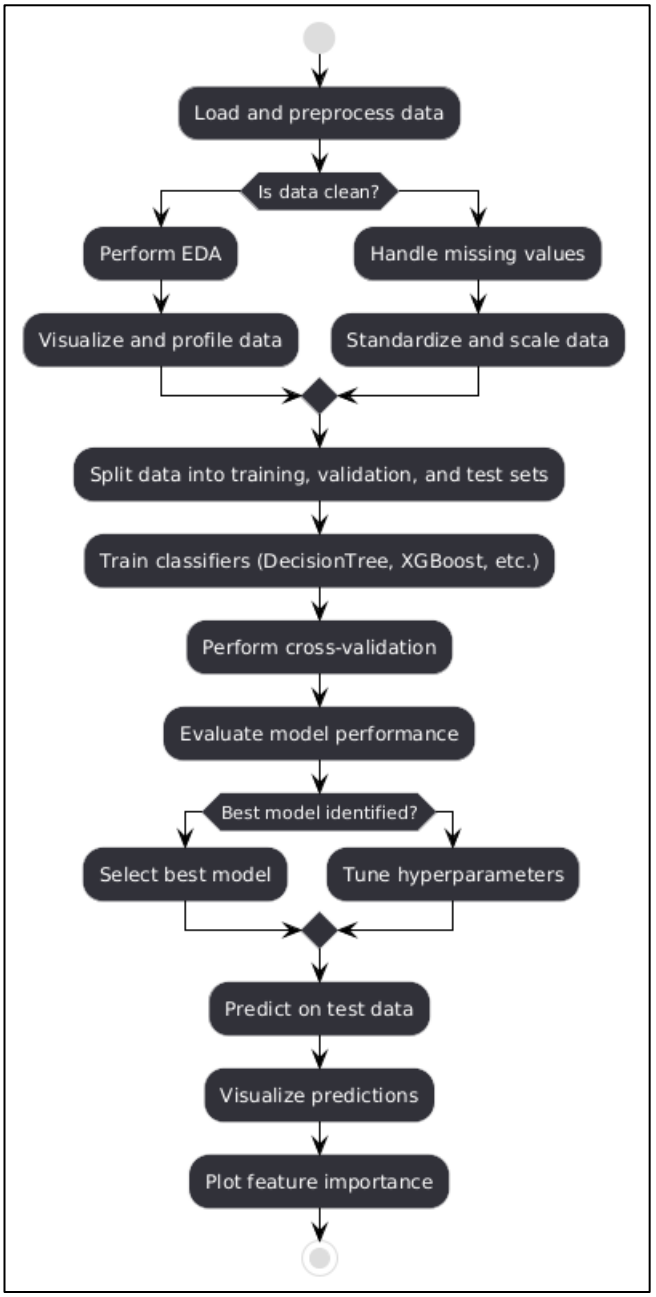
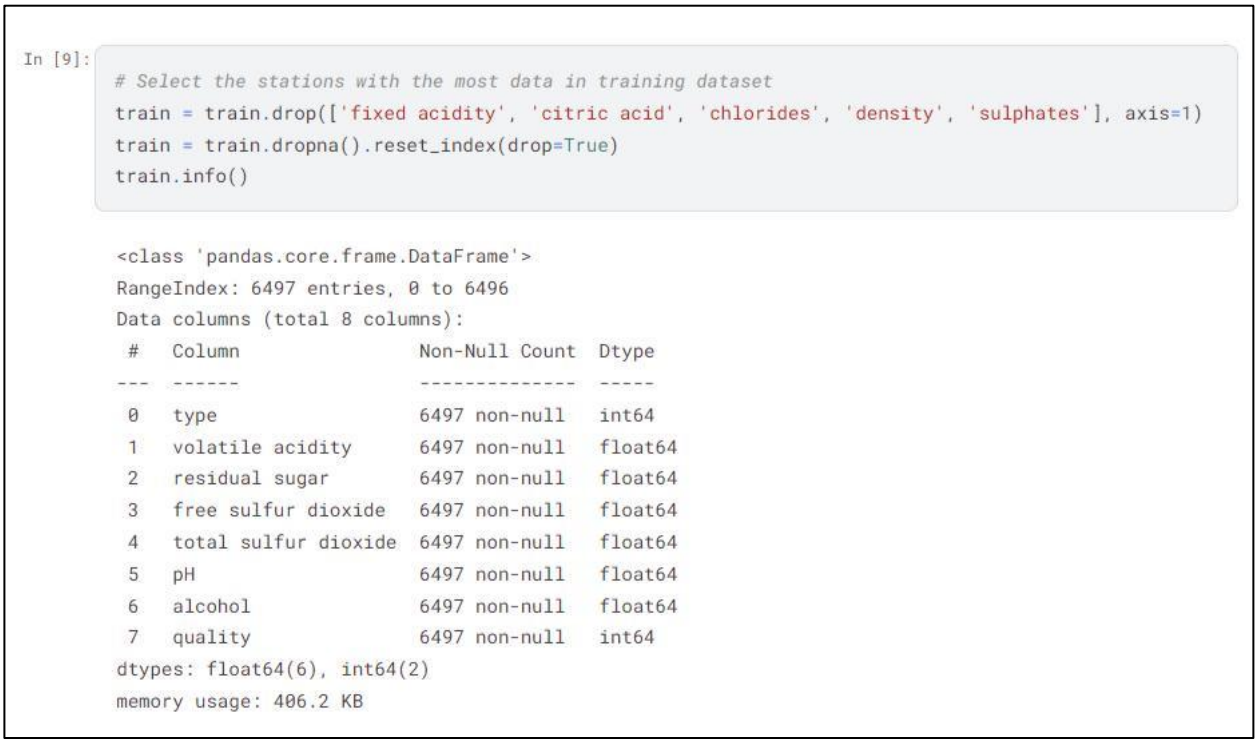


Рисунок 3.2 – UML діаграма діяльності

3.2 Класифікація тестових даних

Спочатку видалили стовпці ('fixed acidity', 'citric acid', 'chlorides', 'density','sulphates'), які не мають значного впливу на класифікацію типів вина, з трену-вального набору даних. Це було зроблено для того, щоб зменшити надмірну інформацію та спростити моделювання. Після цього дані було очищено від пропусків і скинуто індекси для забезпечення правильності подальших обчислень [20,21]. На рисунку 3.3 відображений відбір даних.



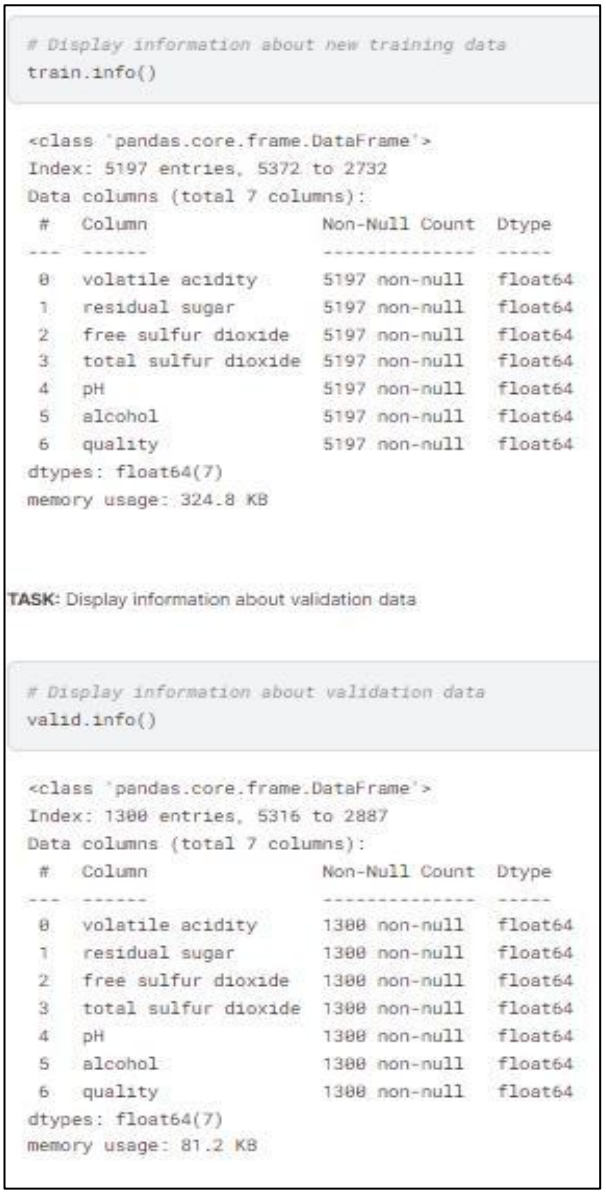
In [9]:

```
# Select the stations with the most data in training dataset
train = train.drop(['fixed acidity', 'citric acid', 'chlorides', 'density', 'sulphates'], axis=1)
train = train.dropna().reset_index(drop=True)
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   type                  6497 non-null   int64
 1   volatile acidity      6497 non-null   float64
 2   residual sugar        6497 non-null   float64
 3   free sulfur dioxide   6497 non-null   float64
 4   total sulfur dioxide  6497 non-null   float64
 5   pH                    6497 non-null   float64
 6   alcohol               6497 non-null   float64
 7   quality               6497 non-null   int64
dtypes: float64(6), int64(2)
memory usage: 406.2 KB
```

Рисунок 3.3 – Відбір даних

Для оцінки продуктивності моделей дані були поділені на тренувальну та валідаційну вибірки за допомогою функції train_test_split. Цей поділ дозволяє навчати модель на одній частині даних і перевіряти її на іншій, що допомагає визначити її здатність узагальнювати знання на нові дані. Використання валідаційної вибірки допомагає уникнути перенавчання, коли модель показує гарні результати на тренувальних даних, але не може адекватно працювати з новими даними [22, 23]. На рисунку 3.4 зображено створення тренувальної та валідаційної вибірки.



```
# Display information about new training data
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5197 entries, 5372 to 2732
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   volatile acidity      5197 non-null   float64
 1   residual sugar        5197 non-null   float64
 2   free sulfur dioxide   5197 non-null   float64
 3   total sulfur dioxide  5197 non-null   float64
 4   pH                    5197 non-null   float64
 5   alcohol               5197 non-null   float64
 6   quality               5197 non-null   float64
dtypes: float64(7)
memory usage: 324.8 KB
```

TASK: Display information about validation data

```
# Display information about validation data
valid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1300 entries, 5316 to 2887
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   volatile acidity      1300 non-null   float64
 1   residual sugar        1300 non-null   float64
 2   free sulfur dioxide   1300 non-null   float64
 3   total sulfur dioxide  1300 non-null   float64
 4   pH                    1300 non-null   float64
 5   alcohol               1300 non-null   float64
 6   quality               1300 non-null   float64
dtypes: float64(7)
memory usage: 81.2 KB
```

Рисунок 3.4 – Створення тренувальної та валідаційної вибірки

Проведено кросвалідацію за допомогою ShuffleSplit, що дозволяє неодноразово ділити дані на тренувальні та тестові підвибірки з подальшим випадковим перемішуванням. Це забезпечує надійні та стабільні оцінки моделей, оскільки перевірки виконуються на різних підмножинах даних. Кросвалідація допомагає мінімізувати вплив випадкових коливань у даних та забезпечити об'єктивну оцінку моделі [24, 25].

Для класифікації були використані три різні моделі: Decision Tree Classifier, Random Forest Classifier та XGBoost Classifier. Для кожної моделі було виконано наступні кроки:

1. Визначення сітки параметрів для пошуку оптимальних гіперпараметрів за допомогою GridSearchCV. Це дозволяє обрати найкращі параметри для моделі для досягнення максимальних результатів.

2. Навчання моделі на тренувальних даних, під час якого модель вивчає взаємозв'язки між ознаками та цільовою змінною.

3. класифікація на тренувальних та валідаційних даних для оцінки того, наскільки добре модель навчилася на тренувальних даних і наскільки ефективно вона узагальнює на нові дані.

4. Оцінка точності моделі за допомогою метрики r2_score та збереження результатів. r2_score показує, наскільки добре модель пояснює варіацію цільової змінної.

3.3 Результати класифікації та візуалізація

Графіки є ефективним інструментом для візуалізації результатів класифікації моделей на тренувальних, валідаційних та тестових даних. Вони демонструють, наскільки точно моделі можуть передбачити значення цільової ознаки ('type') в порівнянні з реальними даними. Візуалізація дозволяє краще зрозуміти ефективність роботи моделей, виявити можливі проблеми та ділянки для покращення, що є критично важливим для подальшої оптимізації моделей.

На рисунку 3.5 зображена діаграма роботи моделей на тренувальних даних, а на рисунку 3.6 зображена діаграма роботи моделей на валідаційних даних.

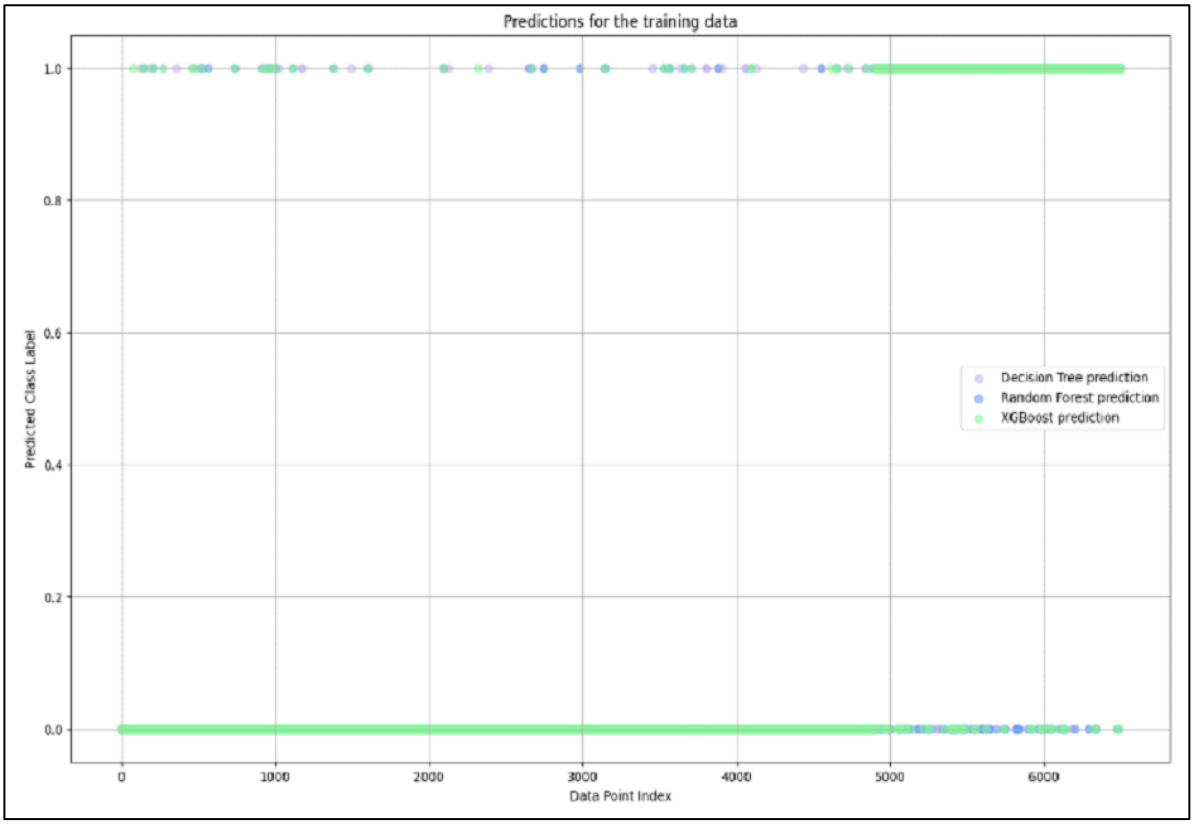


Рисунок 3.5 – Діаграма роботи моделей на тренувальних даних

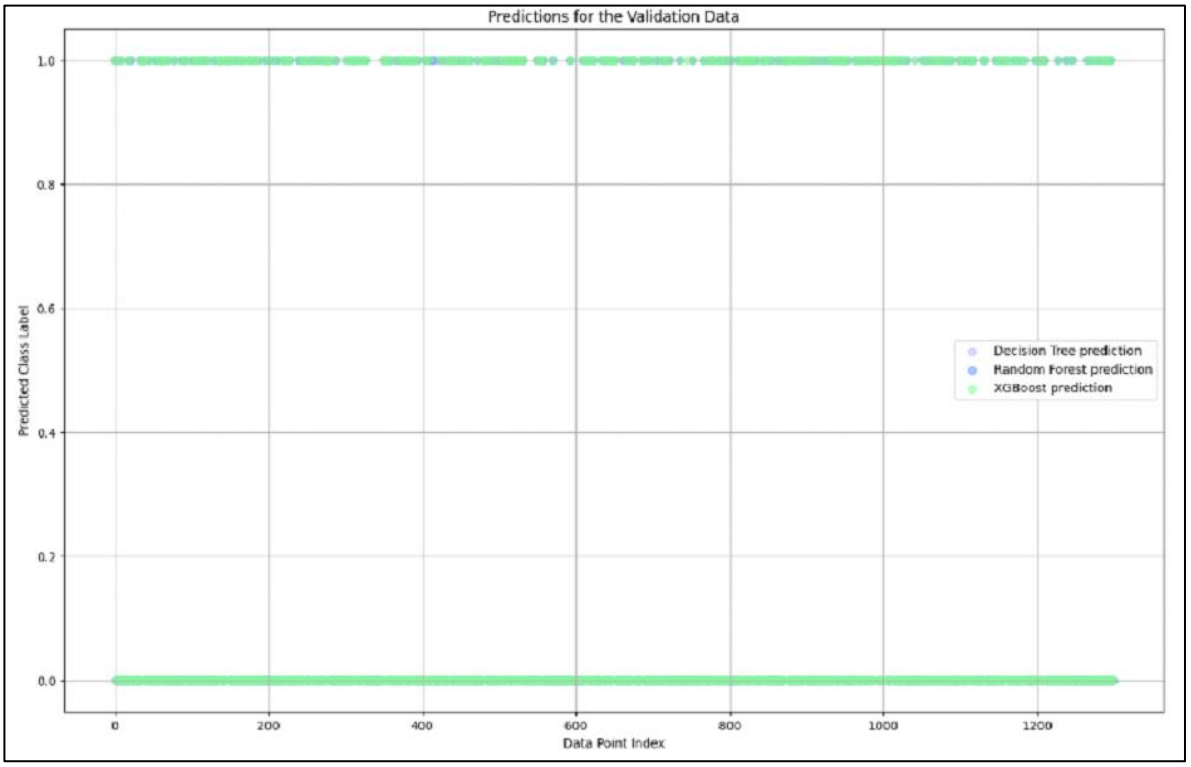


Рисунок 3.6 – Діаграма роботи моделей на валідаційних даних

XGBoost Classifier продемонстрував найвищу точність на тренувальних даних (99.1%) та на валідаційних даних (97.5%). Це свідчить про те, що ця модель найкраще вивчає залежності в даних і добре узагальнює на нових даних, забезпечуючи високий рівень точності. Висока продуктивність XGBoost пояснюється його здатністю ефективно обробляти складні і нелінійні взаємозв'язки між ознаками, що робить його потужним інструментом для задач класифікації.

Random Forest Classifier показав високу точність на тренувальних даних (98.4%) і також добру точність на валідаційних даних (96.8%). Ця модель показала гарні результати, хоча і трохи поступається XGBoost у здатності узагальнювати на нових даних. Random Forest використовує ансамбль дерев рішень, що сприяє зниженню ймовірності перенавчання моделі та покращує її стабільність. Високий результат на валідаційних даних вказує на те, що модель може добре справлятися з новими даними, забезпечуючи точні прогнози.

Decision Tree Classifier продемонстрував найнижчу точність серед трьох моделей на тренувальних даних (98.4%) і на валідаційних даних (96.3%). Хоча ця модель має найнижчі значення результатів, вона все ще є надійною, але трохи менш стабільною порівняно з іншими моделями. Однією з причин цього може бути схильність дерев рішень до перенавчання, особливо коли модель не обмежена у своїй глибині. Проте, навіть з урахуванням цього, Decision Tree Classifier залишається корисним інструментом завдяки своїй простоті і легкості інтерпретації.

На рисунку 3.7 зображені результати класифікації.

	model	train_score	valid_score
2	XGBoost Classifier	99.1	97.5
1	Random Forest Classifier	98.4	96.8
0	Decision Tree Classifier	98.4	96.3

Рисунок 3.7 – Результати класифікації

На рисунку 3.8 наведено найважливіші ознаки, що впливають на класифікацію типів вина. Найбільш важливими ознаками виявилися летюча кислотність, загальний діоксид сірки, залишковий цукор, pH, вільний діоксид сірки, вміст алкоголю та якість вина. Ці ознаки були виділені як найважливіші, оскільки вони мають суттєвий вплив на результати класифікації.

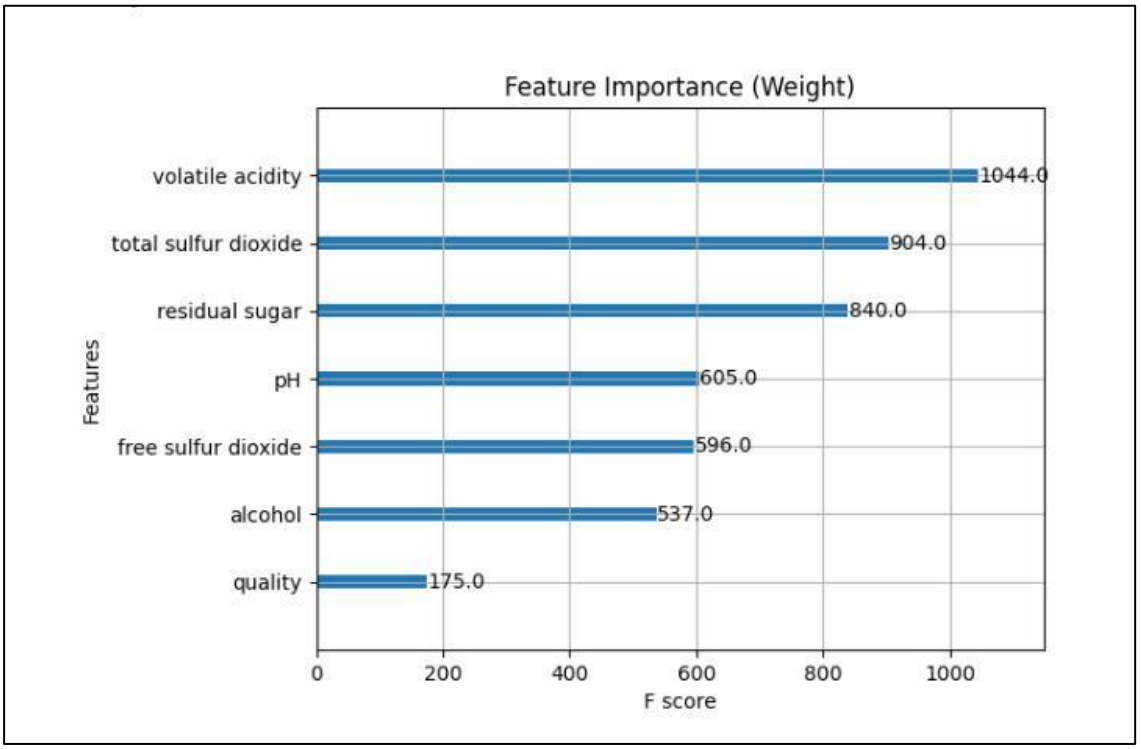


Рисунок 3.8 – Найважливіші ознаки які мають найбільший вплив

3.4 Висновки

У третьому розділі було розроблено інформаційну технологію класифікації типів вина. Спочатку було проведено аналіз даних та вибір найкращих моделей для класифікації, після чого показані та описані результати.

Для класифікації було використано три різні моделі: XGBoost Classifier, Decision Tree Classifier та Random Forest Classifier. Результати моделей показали наступні значення метрики точності:

XGBoost Classifier:

- Точність на тренувальних даних: 99.1%
- Точність на валідаційних даних: 97.5%

Decision Tree Classifier

- Точність на тренувальних даних: 98.4%
- Точність на валідаційних даних: 96.3%

Random Forest Classifier:

- Точність на тренувальних даних: 98.4%
- Точність на валідаційних даних: 96.8%

Для підвищення точності класифікації типів вина можна виконати наступні дії:

1. Збір додаткових характеристик вина: включення детальнішої інформації про хімічний склад вина, наприклад, рівень різних кислот, концентрацію мінералів, та інші властивості, може допомогти уточнити модель і зробити класифікації більш точною. Чим більше вхідних даних, тим краще модель може врахувати різноманітні фактори, що впливають на тип вина.

2. Умови виготовлення вина: важливо враховувати умови виготовлення, такі як температура ферментації, вологість, тип дріжджів і час витримки. Ці фактори можуть мати значний вплив на тип вина. Використання додаткових даних про виробничі процеси може покращити класифікацію.

3. Регулярна перевірка та оцінка моделі: важливо регулярно перевіряти та оцінювати точність моделі за допомогою валідації на незалежних даних. Якщо модель не дає достатньо точних прогнозів, можливо, потрібно налаштувати параметри моделі, використовувати інші методи аналізу або переглянути вхідні дані.

4. Використання інтернету речей (IoT): застосування IoT та сенсорів у виноробстві може допомогти збирати реальні дані про умови виготовлення. Це може включати вимірювання температури, вологості, тиску та інших факторів. Ці дані можуть бути використані для покращення точності моделі класифікації.

Виконання цих рекомендацій може допомогти підвищити точність класифікації типів вина та дозволить більш ефективно контролювати процеси виготовлення.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія класифікації типу вин за їх хімічним складом» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями [27]. Реалізація оцінювання здійснена в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	3	2	3
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	4	3	4
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	3	4	4

Продовження таблиці 4.1

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	38	39	40
Середньоарифметична сума балів $СБ_c$	39,0		

За результатами розрахунків, наведених в таблиці 4.1, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в [27].

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія класифікації типу вин за їх хімічним складом» становить 39 балів, що, відповідно до [27], свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [28]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i \,, \tag{4.1}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *і-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$;

β_i – відносне значення *і-го* технічного показника якості нової розробки.

Результати порівняння зведемо до таблиці 4.2.

Таблиця 4.2 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований продукт	Відношення параметрів нової розробки до аналога	Питома вага показника
Точність класифікації	%	85	97	1,14	0,35
Швидкість обробки одного зразка	с	1,5	1	1,5	0,15
Рівень автоматизації обробки нових даних	%	60	85	1,42	0,15
Точність інтерпретації результатів	%	50	95	1,9	0,25
Рівень масштабованості великих баз даних	бал	7	9,2	1,31	0,1

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1{,}14 \cdot 0{,}35 + 1{,}5 \cdot 0{,}15 + 1{,}42 \cdot 0{,}15 + 1{,}9 \cdot 0{,}25 + 1{,}31 \cdot 0{,}1 = 1{,}44.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,44 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія класифікації типу вин за їх хімічним складом», під час

планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [27]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.2)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 18300{,}00 \cdot 15 / 21 = 13071{,}43 \ грн.$$

Проведені розрахунки зведемо до таблиці 4.3.

Таблиця 4.3 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник дослідження	18300,00	871,43	15	13071,43
Інженер-розробник програмного забезпечення	17500,00	833,33	21	17500,00
Технік	8200,00	390,48	10	3904,76
Всього				34476,19

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія класифікації типу вин за їх хімічним складом» розраховуємо за формулою:

$$З_p = \sum_{i=1}^{n} C_i \cdot t_i \,, \tag{4.3}$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} \,, \tag{4.4}$$

де M_M – розмір мінімальної місячної заробітної плати, приймемо *Мм*=8000,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення [27];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок;

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 8000{,}00 \cdot 1{,}10 \cdot 1{,}15 \,/\, (21 \cdot 8) = 60{,}24 \; грн.$$
$$З_{p1} = 60{,}24 \cdot 6{,}20 = 373{,}48 \; грн.$$

Проведені розрахунки зведемо до таблиці 4.4.

Таблиця 4.4 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка автоматизованого робочого місця розробника програмного забезпечення	6,20	2	1,10	60,24	373,48
Підготовка комп'ютерного обладнання	8,00	4	1,50	82,14	657,14

Продовження таблиці 4.4

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Інсталяція програмного забезпечення розробки технології класифікації вин	5,60	5	1,70	93,10	521,33
Компіляція програмних блоків класифікації	4,30	5	1,70	93,10	400,31
Формування бази даних дослідження	20,00	3	1,35	73,93	1478,57
Тестування ПЗ	6,00	2	1,10	60,24	361,43
Всього					3792,26

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.5)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 11%.

$З_{дод}$ = (34476,19 + 3792,26) · 11 / 100% = 4209,53 грн.

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.6)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (34476,19 + 3792,26 + 4209,53) · 22 / 100% = 9345,16 грн.

4.3.3 Сировина та матеріали

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^{n} H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^{n} B_j \cdot Ц_{вj}\ , \qquad (4.7)$$

де H_j – норма витрат матеріалу *j*-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу *j*-го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, (K_j = *1,1 ... 1,15*);

B_j – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

$M_1 = 4,0 \cdot 215,00 \cdot 1,1 - 0 \cdot 0 = 946,00$ *грн.*

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір	215,00	4,0	0	0	946,00
Папір для записів	135,00	3,0	0	0	445,50
Органайзер офісний	195,00	4,0	0	0	858,00
Канцелярське приладдя (набір офісного працівника)	210,00	4,0	0	0	924,00
Картридж для принтера	985,00	1,0	0	0	1083,50
Диск оптичний	22,50	3,0	0	0	74,25
Flesh-пам'ять 64 GB	340,00	1,0	0	0	374,00
Тека для паперів	120,00	3,0	0	0	396,00
Інші матеріали	210,00	1,0	0	0	231,00
Всього					5332,25

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна технологія класифікації типу вин за їх хімічним складом» відсутні.

4.3.5 Спецустаткування для наукових (експериментальних) робіт

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (4.8)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань устаткування.

$$B_{спец} = 10245,00 \cdot 1 \cdot 1,1 = 11269,50 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.6:

Таблиця 4.6 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Обладнання для збору даних про фізико-хімічні властивості вин (оптичні сенсори, датчики густини, датчики кислотності)	1	10245,00	11269,50
Всього			11269,50

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.9)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.і}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_і$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_і = 1,10…1,12$);

k – кількість найменувань програмних засобів.

$$B_{прг} = 8560,00 \cdot 1 \cdot 1,1 = 9416,00 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.7:

Таблиця 4.7 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows	1	8560,00	9416,00
Прикладний пакет Microsoft Office	1	7730,00	8503,00
Забезпечення бази даних Rabbit MQ	1	11460,00	12606,00
Прикладний пакет мови програмування C#	1	8650,00	9515,00
.NET Framework4.8	1	5280,00	5808,00
Абонентна плата доступу	1	400,00	440,00
Jupyter Notebook	1	4300,00	4730,00
Доступ до платформи Kaggle	1	230,00	253,00
Всього			51271,00

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.10)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (31250,00 \cdot 2) / (3 \cdot 12) = 1736,11 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника програмного забезпечення	31250,00	3	2	1736,11
Комплекс пристроїв передачі даних мережі Internet	8460,00	3	2	470,00
Робоче місце інженера-програміста	10200,00	5	2	340,00
Пристрій виводу інформації	6890,00	5	2	229,67
Оргтехніка	10670,00	5	2	355,67
Лабораторія	345600,00	30	2	1920,00
Всього				5051,44

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ($B_е$) розраховуємо за формулою:

$$B_е = \sum_{i=1}^{n} \frac{W_{уі} \cdot t_i \cdot Ц_е \cdot K_{вні}}{\eta_i}, \quad (4.11)$$

де $W_{уі}$ – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн; приймемо $Ц_е = 10,98$ грн;

$К_{вп i}$ – коефіцієнт, що враховує використання потужності, $К_{вп i} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$В_е = 0,50 \cdot 160,0 \cdot 10,98 \cdot 0,95 / 0,97 = 878,40 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.9.

Таблиця 4.9 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника програмного забезпечення	0,50	160,0	878,40
Комплекс пристроїв передачі даних мережі Internet	0,04	160,0	70,27
Робоче місце інженера-програміста	0,12	160,0	210,82
Пристрій виводу інформації	0,50	50,0	274,50
Оргтехніка	0,50	3,0	16,47
Всього			1450,46

4.3.9 Службові відрядження

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \qquad (4.12)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св} = 20\%$.

$$В_{св} = (34476,19 + 3792,26) \cdot 20 / 100\% = 7653,69 \text{ грн.}$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \quad (4.13)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 30%.

$$B_{сп} = (34476,19 + 3792,26) \cdot 30 / 100\% = 11480,54 \text{ грн.}$$

4.3.11 Інші витрати

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.14)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$$I_в = (34476,19 + 3792,26) \cdot 50 / 100\% = 19134,23 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.15)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$$B_{нзв} = (34476,19 + 3792,26) \cdot 100 / 100\% = 38268,45 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія класифікації типу вин за їх хімічним складом» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}. \quad (4.16)$$

$B_{заг}$= 34476,19 +3792,26 +4209,53 +9345,16 +5332,25 +0,00 +11269,50 + 51271,00 + 5051,44 +1450,46 +7653,69 +11480,54 +19134,23 +38268,45 = 202734,70 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.17)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

$$ЗВ = 202734,70 / 0,9 = 225260,77 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

Результати дослідження проведені за темою «Інформаційна технологія класифікації типу вин за їх хімічним складом» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

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

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 15500 осіб;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 4200,00 грн;

$\pm\Delta Ц_о$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 360,60 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [27]:

$$\Delta П_i = (\pm\Delta Ц_о \cdot N + Ц_о \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.18)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =40%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1$ =(360,60·15500,00+4560,60·500)·0,83·0,4·(1-0,18/100%)=2142419,90 грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2$ =(360,60·15500,00+4560,60·1250)·0,83·0,4·(1-0,18/100%)=3073603,21 грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3$ =(360,60·15500,00+4560,60·2250)·0,83·0,4·(1-0,18/100%)=4315180,96 грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4$ =(360,60·15500,00+4560,60·3150)·0,83·0,4·(1-0,18/100%)=5432600,93 грн.

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

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.19)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,12;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$ПП$ =2142419,90/(1+0,12)[1]+3073603,21/(1+0,12)[2]+4315180,96/(1+0,12)[3]+
+5432600,93/(1+0,12)[4]=1912874,91+2450257,66+3071460,57+3452516,10=10887109
,25 грн.

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \quad (4.20)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 225260,77 грн.

$$PV = k_{інв} \cdot ЗВ = 2 \cdot 225260{,}77 = 450521{,}55 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{4.21}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 10887109,25 грн;

PV – теперішня вартість початкових інвестицій, 450521,55 грн.

$E_{абс} = ПП - PV$ = 10887109,25 - 450521,55 = 10436587,70 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{4.22}$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 10436587,70 грн;

PV – теперішня вартість початкових інвестицій, 450521,55 грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+10436587,70/450521,55)^{1/4} = 1,22.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \tag{4.23}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні d =0,12;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,2.

$\tau_{мін}$ = 0,12+0,2 = 0,32 < 1,22 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія класифікації типу вин за їх хімічним складом» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \tag{4.24}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 1,22 = 0,82 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія класифікації типу вин за їх хімічним складом» становить 39,0 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,44 рази.

Також термін окупності становить 0,82 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія класифікації типу вин за їх хімічним складом».

ВИСНОВКИ

Робота присвячена розробці інформаційної технології класифікації типу вин з використанням сучасних технологій аналізу даних

У першому розділі зроблено загальну характеристику поставленої задачі, та було визначено, що класифікація типів вина є важливим завданням. Було розглянуто проблеми, пов'язані з класифікацією його характеристик, формалізовано постановку задачі та обрано інформаційні технології для її вирішення.

У другому розділі обрано оптимальні налаштування алгоритму, виконано розвідувальний аналіз даних, а також стандартезовано датасет. Також були відібрані оптимальні моделі для класифікації типів вина.

У третьому розділі було розроблено інформаційну технологію класифікації типів вина, результати якої оцінювалися за метрикою accuracy score. Також було зроблено діаграму класів та діаграму діяльності. Оптимальною моделлю стала XGBoost Classifier, яка показала найвищу точність: 99.1% на тренувальних і 97.5% на валідаційних даних. Такий результат свідчить про те, що модель ефективно вивчає залежності в даних і добре узагальнює їх на нових наборах даних, при цьому різниця між тренувальною та валідаційною точністю менша за 10%, що є показником відсутності суттєвого перенавчання. Decision Tree Classifier також продемонстрував гідні результати з точністю 98.4% на тренувальних і 96.3% на валідаційних даних. Однак ця модель трохи поступається XGBoost у здатності узагальнювати дані, що, ймовірно, пов’язано з меншою складністю алгоритму та меншою гнучкістю. Random Forest Classifier показав точність 98.4% на тренувальних і 96.8% на валідаційних даних, забезпечуючи стабільні результати, проте її ефективність трохи нижча, ніж у XGBoost, що може бути зумовлено недостатньо ретельною оптимізацією гіперпараметрів. Усі моделі продемонстрували високий рівень продуктивності, але XGBoost Classifier найбільш повно відповідає вимогам завдання. З огляду на

отриманий результат можна сказати, що підвищення точності класифікації досягнуто.

У четвертому розділі було проведено економічне дослідження, результати якого підтверджують високу комерційну привабливість розробки. Рівень комерційного потенціалу становить 39,0 бала, що свідчить про його значимість і перевищення середнього рівня. Технічні параметри розробки демонструють перевагу над існуючими аналогами в 1,44 рази за узагальненим коефіцієнтом якості, що свідчить про конкурентоспроможність продукту. Крім того, термін окупності розробки становить лише 0,82 року, що є значно нижчим від типового трирічного показника, підтверджуючи її економічну ефективність. Такі результати можуть зацікавити потенційних інвесторів у фінансуванні впровадження розробки та її виходу на ринок.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія класифікації типу вин за їх хімічним складом», оскільки розробка є перспективною з точки зору як технічної ефективності, так і комерційної вигоди.

Тези отриманих результатів подані на Міжнародну науково-практичну інтернет-конференцію «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.) з публікацією.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сокур Д. С., Мокін В. Б., Розроблення інформаційної технології класифікації типу вин за їх хімічним складом. Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.). URL: https://conferences.vntu.edu.ua/index.php/. (Дата звернення: 10.11.2024).

2. Wine - Worldwide. 2024. URL: https://www.statista.com/outlook/cmo/alcoholic-drinks/wine/worldwide (дата звернення: 06.10.2024).

3. Wine production, 2021 URL: https://our-worldindata.org/grapher/wine-production (дата звернення: 06.10.2024).

4. Niall McCarthy. Europe's Biggest Wine Drinkers. 2016 URL: https://www.statista.com/chart/4837/europes-biggest-wine-drinkers/ (дата звернення: 11.10.2024).

5. Микола Ладуба. Путівник з вина: історія, технологія виробництва, класифікація та види, а також рекомендації щодо дегустації. 2023 URL: https://itc.ua/ua/articles/putivnyk-z-vyna-istoriya-tehnologiya-vyrobnytstva-klasyfikatsiya-ta-vydy-a-takozh-rekomendatsiyi-shhodo-degustatsiyi/ (дата звернення: 11.10.2024).

6. В. Б. Мокін, М. В. Дратований. Наука про дані: машинне навчання та інтелектуальний аналіз даних: електронний навчальний посібник комбінованого (локального та мережевого) використання Вінниця: ВНТУ, 2024. URL: https://docs.vntu.edu.ua/card.php?id=8163 (дата звернення: 11.10.2024).

7. Iannone R., Miranda S., Riemma S., De Marco I. Improving environmental performances in wine production by a life cycle assessment analysis. 2016. URL: https://www.sciencedirect.com/science/arti-cle/abs/pii/S0959652615003492 (дата звернення: 11.10.2024).

8. Goldammer T. Wine Production. 2022. URL: https://www.wine-production.com/ (дата звернення: 11.10.2024).

9. Aeberhard S. UC Irvine "Wine". 1991. URL: https://archive.ics.uci.edu/dataset/109/wine (дата звернення: 12.10.2024).

10. Vennerød C.B., Kjærran A., Bugge E.S. Long Short-term Memory RNN. *Arxiv*, 2022. URL: https://arxiv.org/abs/2105.06756 (дата звернення: 12.10.2024).

11. O'Shea K., Nash R. An Introduction to Convolutional Neural Networks. *Arxiv*, 2015. URL: https://arxiv.org/abs/1511.08458 (дата звернення: 12.10.2024).

12. DeepGrid. Backpropagation in Convolutional Neural Networks. 2016. URL: https://jefkine.com/backpropagation-in-convolutional-neural-net-works (дата звернення: 12.10.2024).

13. Almeida F., Xexéo G. Word Embeddings Survey. *Arxiv*, 2019. URL: https://arxiv.org/abs/1901.09069 (дата звернення: 12.10.2024).

14. Kaggle. URL: https://www.kaggle.com – (дата звернення: 14.10.2024).

15. Andreas C. Müller; Sarah Guido. Introduction to machine learning with python. *O'Reilly Media, Inc*., 2016. 1288 с. URL: https://www.nrigroupindia.com/e-book/Introduction%20to%20Machine%20Learning%20with%20Python%20(%20PDFDrive.com%20)-min.pdf (дата звернення: 22.10.2024).

16. Riturajsaha. Understanding TF-IDF (Term Frequency-Inverse Document Frequency). *GeeksForGeeks*, 2023. URL: https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency (дата звернення: 22.10.2024).

17. Ethayarajh K. Word Embedding Analogies: Understanding King - Man + Woman Queen. GitHub, 2020. URL: https://kawinethaya-rajh.github.io/embedding-analogies (дата звернення: 22.10.2024).

18. Peer D. Впровадження алгоритмів машинного навчання для класифікації завдань. URL: https://peerdh.com/uk/blogs/programming-insights/implementing-machine-learning-algorithms-for-task-classification (дата звернення: 22.10.2024).

19. Кравченко С.М., Гришкун Є.О., Власенко О.В. Методи класифікації машинного навчання. URL: https://tech.vernadskyjournals.in.ua/journals/2020/3_2020/part_1/21.pdf (дата звернення: 22.10.2024).

20. Бойко Н. І. Дослідження алгоритмів машинного навчання для побудови математичних моделей. URL: https://science.lpnu.ua/uk/jcpee/vsi-vypusky/vypusk-11-nomer-2-2021/doslidzhennya-algorytmiv-mashynnogo-navchannya-dlya (дата звернення: 22.10.2024).

21. Шустер О. Г. Типи задач у машинному навчанні: від класифікації до кластеризації. URL: https://freshtech.global/ua/blog/types-of-tasks-in-machine-learning-from-classification-to-clustering (дата звернення: 04.11.2024).

22. Mindful Machine Learning / S. Sauer та ін. *European Journal of Psychological Assessment*. 2018. Т. 34, № 1. С. 6–13. URL: https://doi.org/10.1027/1015-5759/a000312 (дата звернення: 14.11.2024).

23. Nair D. P. S. Analyzing Titanic Disaster using Machine Learning Algorithms. *International Journal of Trend in Scientific Research and Development*. 2017. Volume-2, Issue-1. С. 410–416. URL: https://doi.org/10.31142/ijtsrd7003 (дата звернення: 04.11.2024).

24. Гавриленко С. Ю. Розробка методу ідентифікації стану комп'ютерних систем на основі класифікаційних алгоритмів. URL: https://repository.kpi.kharkov.ua/items/de76f7d3-bdc2-4a86-aa48-eca3507ff5bc (дата звернення: 14.11.2024).

25. Siphendulwe, Z., Atemkeng, M., Hamlomo, S. Wine feature importance and quality prediction: A comparative study of machine learning algorithms with unbalanced data. URL: https://arxiv.org/abs/2310.01584 (дата звернення: 04.11.2024).

26. Deep J. Programming : 4 books in 1: python programming and crash course, machine learning for beginners, python machine learning. *Independently Published*, 2020. 502 с. URL: https://dokumen.pub/programming-4-books-in-1-python-programming-amp-crash-course-machine-learning-for-beginners-python-machine-learning-y-1794097.html (дата звернення: 04.11.2024).

27. Mennen V. Machine Learning Basics : a Comprehensive Guide about Machine Learning: Machine Learning. Independently Published, 2021. URL:

https://www.researchgate.net/publication/380157485_Machine_Learning_Basics_A_Comprehensive_Guide_A_Review (дата звернення: 15.11.2024).

28. В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця : ВНТУ, 2021. 42 с.

29. В. В. Кавецький, В. О. Козловський, І. В. Причепа. Економічне обґрунтування інноваційних рішень: практикум. Вінниця : ВНТУ, 2016. 113 с.

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. Віталій МОКІН
«___» _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСИФІКАЦІЇ ТИПУ ВИН ЗА ЇХ ХІМІЧНИМ СКЛАДОМ»
08-34.МКР.002.02.000.ТЗ

Керівник: д.т.н., проф. каф. САІТ
_____ Віталій МОКІН
«___» _____ 2024 р.

Розробив: студент гр. 1ІСТ-23м
_____ Дмитро СОКУР
«___» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 2024 р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САІТ від «__» _____ 2024 р.

2. Джерела розробки:

- В. Б. Мокін, М. В. Дратований. Наука про дані: машинне навчання та інтелектуальний аналіз даних: електронний навчальний посібник комбінованого (локального та мережевого) використання Вінниця: ВНТУ, 2024. – 258 с.
- Бойко Н. І. Дослідження алгоритмів машинного навчання для побудови математичних моделей. URL: https://science.lpnu.ua/uk/jcpee/vsi-vypusky/vypusk-11-nomer-2-2021/doslidzhennya-algorytmiv-mashynnogo-navchannya-dlya.

3. Мета та призначення роботи.

Метою дослідження є підвищення точності класифікації типів вина на основі хімічного складу шляхом розробки інформаційної технології. Це забезпечить виробникам вина можливість оптимізації технологічних процесів, покращення якості продукції, а також сприятиме більш ефективному використанню ресурсів у виноробній галузі.

4. Вихідні дані для проведення робіт:

датасет Kaggle «Wine Quality Data Set (Red & White Wine)»

5. Методи дослідження:

У даній роботі використовуються методи аналізу даних, статистичного моделювання, машинного навчання, а також візуалізація результатів для оцінки ефективності класифікації.

6. Етапи роботи і терміни їх виконання:

1. Загальна характеристика поставленої задачі _____– _____
2. Вибір оптимальних налаштувань алгоритму _____– _____
3. Розробка інформаційної технології _____– _____
4. Підготовка економічної частини _____– _____
5. Оформлення пояснювальної записки _____– _____

7. Очікувані результати та порядок реалізації:

Розробка інформаційної технології для класифікації типу вина на основі хімічного складу, визначення оптимальної моделі та її налаштувань для досягнення найвищої точності прогнозування.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання магістерських кваліфікаційних робіт для студентів спеціальності 126 «Інформаційні системи та технології» (освітня програма «Інформаційні технології аналізу даних та зображень»)

9. Порядок контролю та приймання

Публічний захист ..... «___» _____ 2024 р.

Початок розробки ..... «___» _____ 2024 р.
Граничні терміни виконання МКР ..... «___» _____ 2024 р.

Розробив студент групи 1ІСТ-23м _____ Дмитро СОКУР

Додаток Б

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія класифікації типу вин за їх хімічним складом»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ
Керівник: Віталій МОКІН, д.т.н., проф. каф. САІТ

Показники звіту подібності Turnitin

Оригінальність 98 % Загальна схожість 2 %

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 відсутності самостійності її автора. Роботу направити на доопрацювання.
○ 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.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.
Автор роботи _____ (підпис) Дмитро СОКУР

Опис прийнятого рішення

Робота допускається до захисту

Особа, відповідальна за перевірку _____ (підпис) Сергій ЖУКОВ

Додаток В

Лістинг програмного забезпечення

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.filterwarnings("ignore")
from IPython.display import HTML
from sklearn.model_selection import RandomizedSearchCV, KFold
import numpy as np
import pandas as pd
import ydata_profiling as pp
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, KFold, ShuffleSplit, GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import xgboost as xgb
from xgboost.sklearn import XGBClassifier
from sklearn.metrics import r2_score, accuracy_score, ConfusionMatrixDisplay
from xgboost import plot_importance
import seaborn as sns
import matplotlib as mpl
train = pd.read_csv('/kaggle/input/wine-quality-data-set-red-white-wine/wine-quality-white-and-red.csv')
train['type'] = train['type'].map({'white': 0, 'red': 1})
train.to_csv('wine-quality-modified.csv', index=False)
train = train.head(6497)
train.info()
test = pd.read_csv('/kaggle/input/wine-quality-data-set-red-white-wine/wine-quality-white-and-red.csv')
test['type'] = test['type'].map({'white': 0, 'red': 1})
test.to_csv('wine-quality-modified.csv', index=False)
test = test.head(6497)
test.tail()
train = train.drop(['fixed acidity', 'citric acid', 'chlorides', 'density', 'sulphates'], axis=1)
```

```
train = train.dropna().reset_index(drop=True)
train.info()
target = train.pop('type')
print("Shape of train data after popping target:", train.shape)
print("Shape of target data:", target.shape)
test = test.drop(['fixed acidity', 'citric acid', 'chlorides', 'density', 'sulphates'], axis=1)
test = test.dropna().reset_index(drop=True)
pp.ProfileReport(test)
scaler = StandardScaler()
train = pd.DataFrame(scaler.fit_transform(train), columns = train.columns)
# Display training data
train
print("Training data after standardization:")
print(train.head())
# Standartization data
scaler = StandardScaler()
target_test = test.pop('type')
test = pd.DataFrame(scaler.fit_transform(test), columns = test.columns)
# Display test
Test
# Initialize the Decision Tree Classifier
decision_tree = DecisionTreeClassifier()

# Define the parameter grid for the classifier
param_grid = {
    'min_samples_leaf': [1, 2, 4, 6, 10],
    'max_depth': [3, 5, 7, 10, 15],
    'min_samples_split': [2, 5, 10, 15, 20],
    'max_features': ['auto', 'sqrt', 'log2', None]
}

# Perform GridSearchCV to find the best hyperparameters
decision_tree_CV = GridSearchCV(decision_tree, param_grid=param_grid, cv=cv_train,
verbose=False)
```

```
decision_tree_CV.fit(train, target_train)

# Print the best hyperparameters
print("Best parameters for Decision Tree: ", decision_tree_CV.best_params_)

# Predict on the training data
y_train_decision_tree = decision_tree_CV.predict(train)

# Calculate the accuracy on the training data
train_accuracy = round(accuracy_score(target_train, y_train_decision_tree) * 100, 1)
print(f'Accuracy of Decision Tree model training is: {train_accuracy}')

# Store the training score in the results DataFrame
result.loc[result['model'] == 'Decision Tree Classifier', 'train_score'] = train_accuracy

# Predict on the validation data
y_val_decision_tree = decision_tree_CV.predict(valid)

# Calculate the accuracy on the validation data
valid_accuracy = round(accuracy_score(target_valid, y_val_decision_tree) * 100, 1)
result.loc[result['model'] == 'Decision Tree Classifier', 'valid_score'] = valid_accuracy
print(f'Accuracy of Decision Tree on valid data: {valid_accuracy}')
# Using cross-validation for more stable model evaluation
cv_train = KFold(n_splits=5, shuffle=True, random_state=42)

# Defining parameter grid for Random Forest Classifier
param_grid = {
    'n_estimators': [50, 100, 200],  # Number of trees
    'min_samples_leaf': [i for i in range(2, 8)],  # Minimum samples per leaf
    'max_depth': [i for i in range(3, 7)],  # Maximum depth of the tree
    'criterion': ['gini', 'entropy'],  # Criteria for splitting
    'bootstrap': [True],  # Using bootstrap sampling
    'max_features': ['sqrt', 'log2']  # Number of features considered for splitting
}
```

```
# Creating GridSearchCV object for hyperparameter tuning
rf = RandomForestClassifier(random_state=42)
rf_CV = GridSearchCV(rf, param_grid=param_grid, cv=cv_train, verbose=False,
error_score='raise')
rf_CV.fit(train, target_train)

# Output the best parameters
print("Best parameters for Random Forest: ", rf_CV.best_params_)

# Prediction on the training data
y_train_rf = rf_CV.predict(train)

# Calculating accuracy on the training data
train_accuracy = round(accuracy_score(target_train, y_train_rf) * 100, 1)
print(f'Accuracy of Random Forest Classifier model training is {train_accuracy}')

# Saving results in DataFrame
result.loc[result['model'] == 'Random Forest Classifier', 'train_score'] = train_accuracy

# Prediction on the validation data
y_val_rf = rf_CV.predict(valid)
valid_accuracy = round(accuracy_score(target_valid, y_val_rf) * 100, 1)
result.loc[result['model'] == 'Random Forest Classifier', 'valid_score'] = valid_accuracy
print(f'Accuracy of Random Forest Classifier on validation data is {valid_accuracy}')
cv_train = KFold(n_splits=3, shuffle=True, random_state=42)

# Parameter grid for XGBoost Classifier
param_grid = {
    'n_estimators': [50, 100, 200],  # Number of boosting rounds
    'max_depth': [3, 4, 5],  # Maximum tree depth for base learners
    'subsample': [0.8, 0.9],  # Subsample ratio of the training data
    'learning_rate': [0.01, 0.05, 0.1],  # Step size shrinkage
    'gamma': [0, 1, 5],  # Minimum loss reduction to make a further partition
```

```
    'min_child_weight': [1, 3, 5],  # Minimum sum of instance weight needed in a child
    'colsample_bytree': [0.8, 0.9],  # Subsample ratio of columns when constructing each tree
    'alpha': [0, 0.1, 1],  # L1 regularization term on weights
    'lambda': [1, 2]  # L2 regularization term on weights
}

# Initialize the XGBoost Classifier
xgb = XGBClassifier(random_state=42)

# Set early stopping rounds
xgb.set_params(early_stopping_rounds=10)

# Create the RandomizedSearchCV object
xgb_CV = RandomizedSearchCV(xgb, param_distributions=param_grid, n_iter=50, cv=cv_train,
verbose=False, random_state=42, n_jobs=-1)

# Fit the model using cross-validation
xgb_CV.fit(train, target_train, eval_set=[(valid, target_valid)], verbose=False)

# Output the best parameters
print("Best parameters for XGBoost: ", xgb_CV.best_params_)

# Predict on the training data
y_train_xgb = xgb_CV.predict(train)

# Calculate accuracy on the training data
train_accuracy = round(accuracy_score(target_train, y_train_xgb) * 100, 1)
print(f'Accuracy of XGBoost Classifier model training is {train_accuracy}')

# Store the training score in the results DataFrame
result.loc[result['model'] == 'XGBoost Classifier', 'train_score'] = train_accuracy

# Predict on the validation data
y_val_xgb = xgb_CV.predict(valid)
```

```
# Calculate accuracy on the validation data
valid_accuracy = round(accuracy_score(target_valid, y_val_xgb) * 100, 1)
result.loc[result['model'] == 'XGBoost Classifier', 'valid_score'] = valid_accuracy
print(f'Accuracy of XGBoost Classifier on validation data is {valid_accuracy}')
# Building plot for predictions on the training data
x = np.arange(len(train))
plt.figure(figsize=(16, 10))

# Plotting the actual target data (converted to scatter for classification)
plt.scatter(x, target_train, label="Target data", color='#f899e7', alpha=0.6)

# Plotting predictions for each classification model
plt.scatter(x, y_train_decision_tree, label="Decision Tree prediction", color='#beb1fa', alpha=0.6)
plt.scatter(x, y_train_rf, label="Random Forest prediction", color='#6699ff', alpha=0.6)
plt.scatter(x, y_train_xgb, label="XGBoost prediction", color='#81f696', alpha=0.6)

# Adding plot title and legend
plt.title('Predictions for the Training Data')
plt.legend(loc='best')
plt.grid(True)
plt.xlabel('Data Point Index')
plt.ylabel('Predicted Class Label')

plt.show()
# Building plot for predictions on the validation data
x = np.arange(len(valid))
plt.figure(figsize=(16, 10))

# Plotting predictions for each classification model
plt.scatter(x, y_val_decision_tree, label="Decision Tree prediction", color='#beb1fa', alpha=0.6)
plt.scatter(x, y_val_rf, label="Random Forest prediction", color='#6699ff', alpha=0.6)
plt.scatter(x, y_val_xgb, label="XGBoost prediction", color='#81f696', alpha=0.6)
```

```
# Adding plot title and legend
plt.title('Predictions for the Validation Data')
plt.legend(loc='best')
plt.grid(True)
plt.xlabel('Data Point Index')
plt.ylabel('Predicted Class Label')

plt.show()
# Building plot for predictions on the test data
x = np.arange(len(test))
plt.figure(figsize=(16, 10))

# Plotting predictions for each classification model
plt.scatter(x, y_test_decision_tree, label="Decision Tree prediction", color='#beb1fa', alpha=0.6)
plt.scatter(x, y_test_rf, label="Random Forest prediction", color='#6699ff', alpha=0.6)
plt.scatter(x, y_test_xgb, label="XGBoost prediction", color='#81f696', alpha=0.6)

# Adding plot title and legend
plt.title('Predictions for the training data')
plt.legend(loc='best')
plt.grid(True)
plt.xlabel('Data Point Index')
plt.ylabel('Predicted Class Label')

plt.show()
# Displaying the sorted results of the classification modeling
sorted_result = result.sort_values(by=['valid_score', 'train_score'], ascending=False)

# Show the sorted DataFrame
sorted_result
# Selecting models with minimal overfitting
result_best = result[(result['train_score'] - result['valid_score']).abs() < 5]

# Sorting the selected models by validation and training scores
```

```
result_best_sorted = result_best.sort_values(by=['valid_score', 'train_score'], ascending=False)

# Display the sorted DataFrame of models with minimal overfitting
result_best_sorted
# Prediction for the training data
y_train_xgb = xgb_CV.predict(train)

# Plotting Feature Importance
plt.figure(figsize=(10, 6))
plot_importance(xgb_CV.best_estimator_, importance_type='weight', title='Feature Importance
(Weight)')
plt.title('Feature Importance (Weight)')
plt.show()
# Training the XGBoost classifier
xgb = XGBClassifier()
xgb.fit(train, target_train)

# Plotting Feature Importance
plt.figure(figsize=(10, 6))
plot_importance(xgb, importance_type='weight')
plt.title('Feature Importance (Weight)')

# Show the plot
plt.show()
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСИФІКАЦІЇ ТИПУ ВИН ЗА ЇХ ХІМІЧНИМ СКЛАДОМ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2024 р.

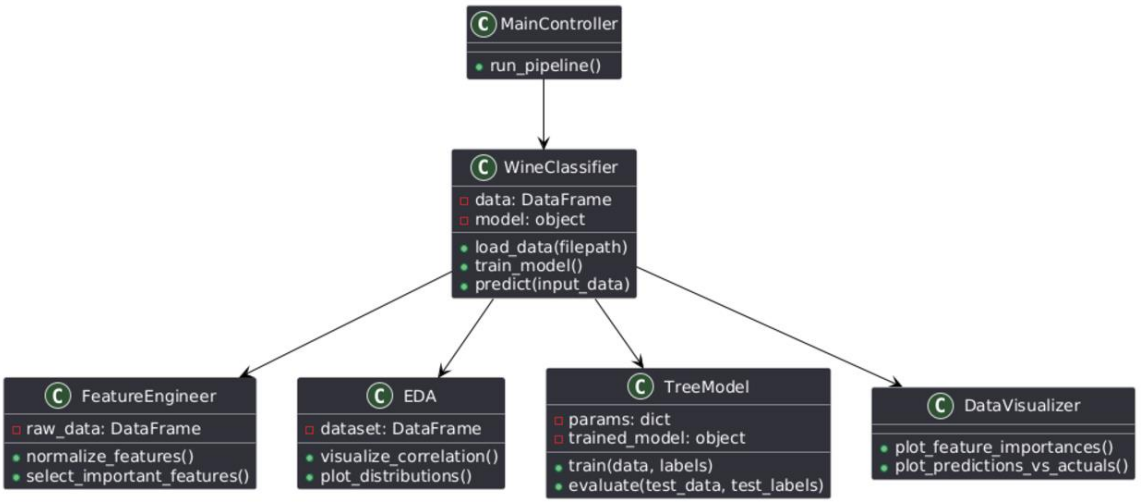


Рисунок Г.1 – UML-діаграма класів

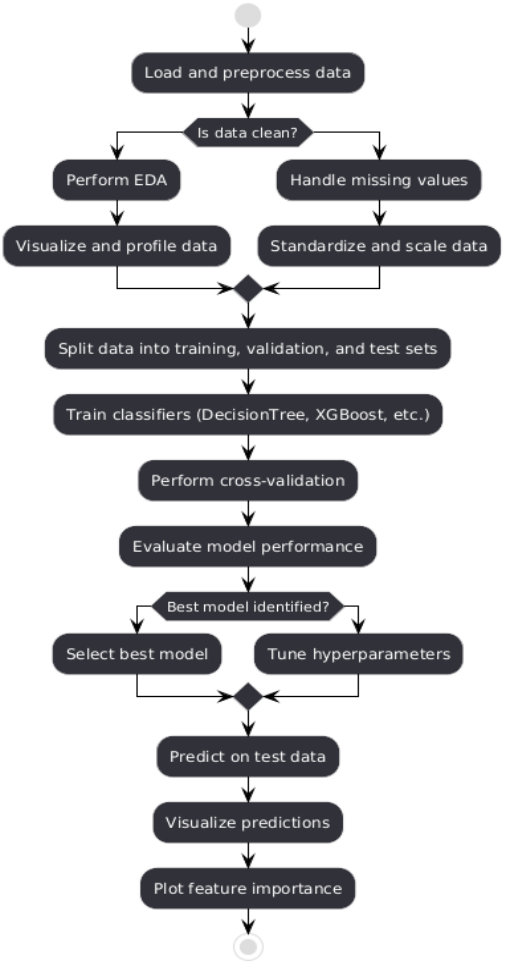


Рисунок Г.2 – UML діаграма діяльності

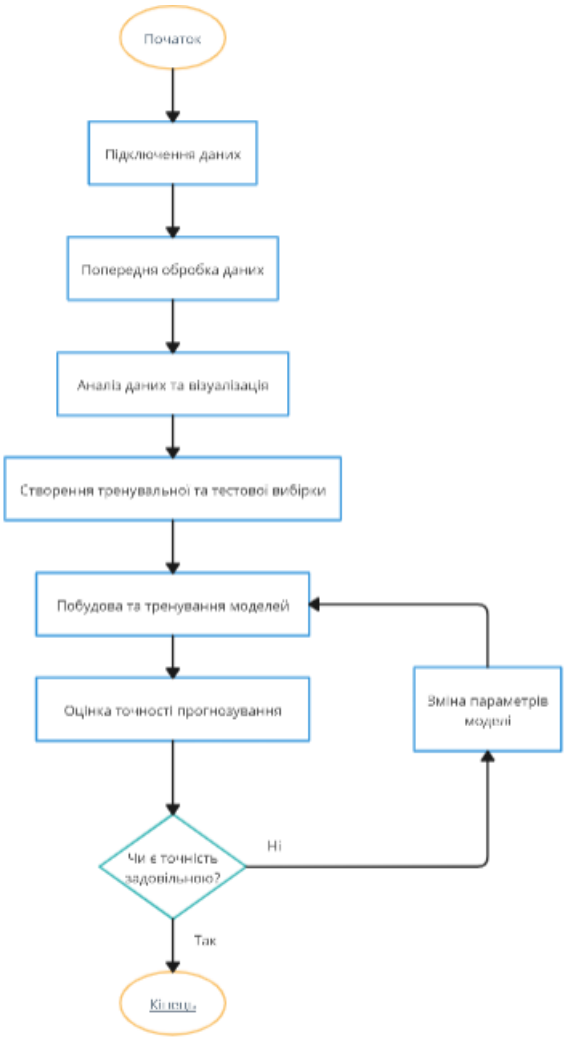


Рисунок Г.3 – UML діаграма активності

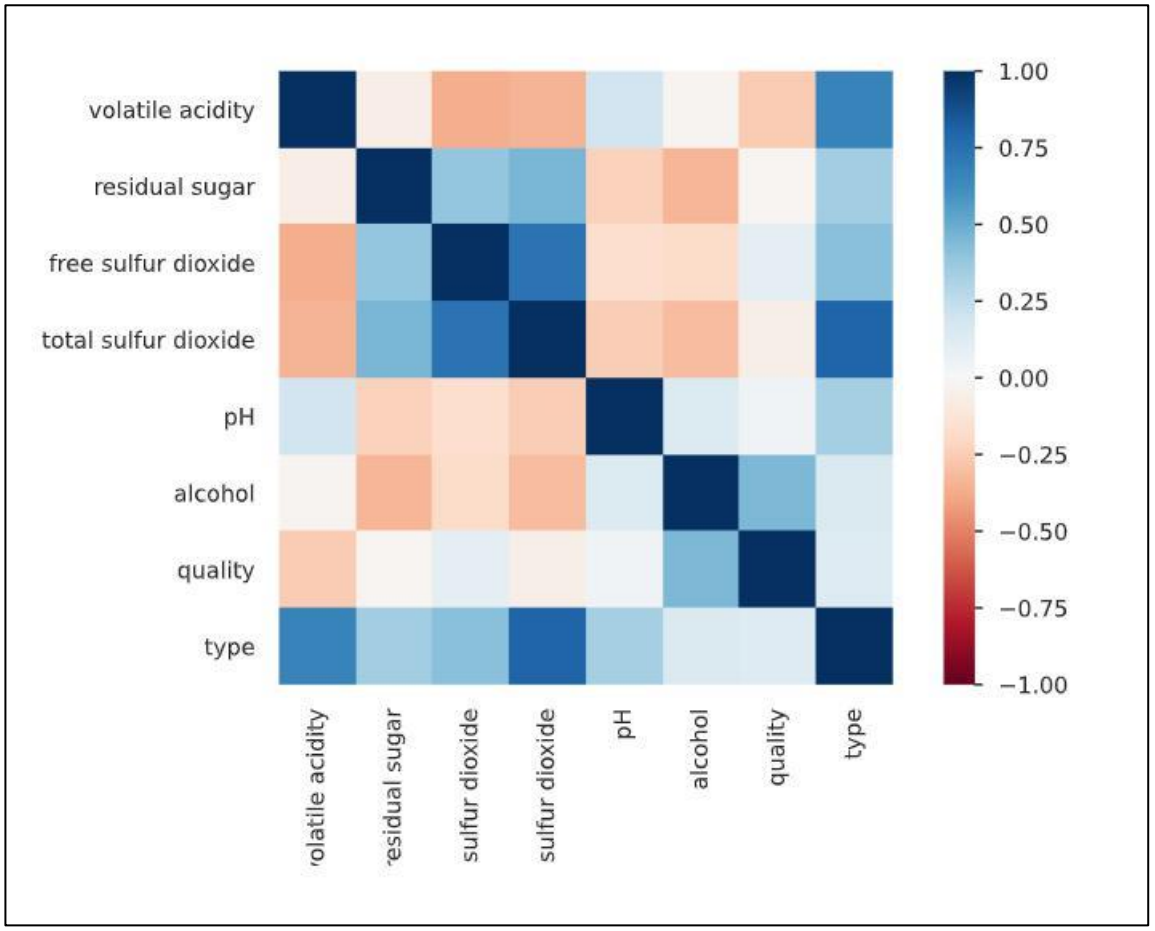


Рисунок Г.4 – Heatmap ознак використаного датасету

	model	train_score	valid_score
2	XGBoost Classifier	99.1	97.5
1	Random Forest Classifier	98.4	96.8
0	Decision Tree Classifier	98.4	96.3

Рисунок Г.5 – Результати класифікації

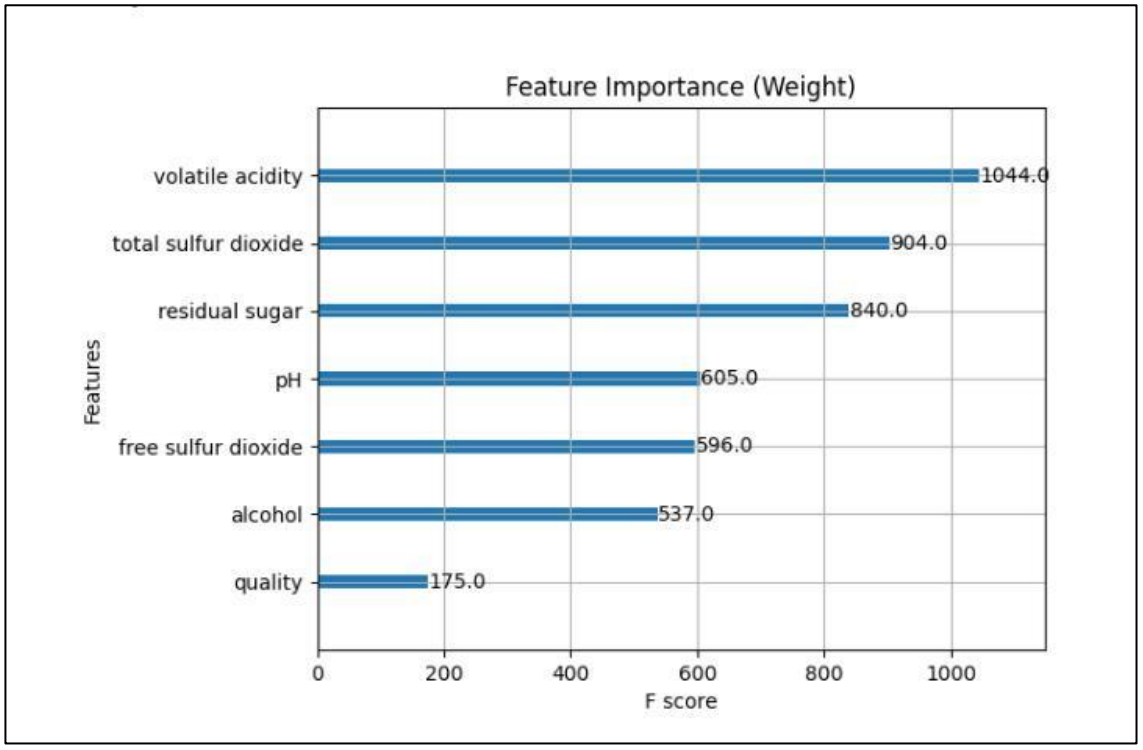


Рисунок Г.6 – Найважливіші ознаки які мають найбільший вплив