МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧОРНОМОРСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ПЕТРА МОГИЛИ

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК

КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

"ЗАТВЕРДЖУЮ"
Перший проректор
Іщенко Н.М.
"__" _____2019 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ЕВОЛЮЦІЙНІ АЛГОРИТМИ»

Спеціальність: 123 Комп'ютерна інженерія

Рівень вищої освіти – третій (освітньо-науковий)

Розробник	Гожий О.П.
Завідувач кафедри розробника	Кондратенко Ю.П.
Завідувач кафедри спеціальності	Дворник О.В.
Гарант освітньої програми	Чуйко Г.П.
Декан факультету КН	Бойко А.П.
Начальник ННІПО	Норд Г.Л.
Начальник НМВ	Калініченко В.І.

Миколаїв – 2019

1. Опис навчальної дисципліни

Найменування показника	Характеристика дисципліни	
Найменування дисципліни	Еволюційні алгоритми	
Галузь знань	12 Інформаційні технології	
Спеціальність	Комп'ютерна інженерія	
Спеціалізація (якщо є)	-	
Освітня програма	Комп'ютерна інженерія	
Рівень вищої освіти	Доктор філософії	
Статус дисципліни	Нормативна	
Курс навчання	2	
Навчальний рік	2019–2020	
Номер(и) семестрів / триместрів:	Денна форма	Заочна форма
	3 семестр	
Загальна кількість кредитів ЄКТС/годин	3 кредитів / 90 годин	
	Денна форма	Заочна форма
Структура курсу: – лекції – групові – годин самостійної роботи студентів	10 20 90	
Відсоток аудиторного навантаження	33 %	
Мова викладання	Українська, англійська	
Форма проміжного контролю	Тестування;	
Форма підсумкового контролю	Іспит	

2. Мета, завдання та результати вивчення дисципліни

Концепція викладання дисципліни:

Дисципліна «Еволюційні алгоритми» присвячена розгляду теоретичних питань побудови комп'ютерних еволюційних алгоритмів та актуальних питань застосування еволюційних алгоритмів для вирішення прикладних задач, а також використання апарату теорії алгоритмів для побудови ефективних алгоритмів.

Еволюційні алгоритми (моделювання загальних закономірностей еволюції). Це системи, які використовують еволюційні принципи розвитку популяції. Вони успішно використовуються для завдань функціональної оптимізації і можуть легко бути описані математичною мовою. Еволюційні моделі. Це системи, які відтворюють біологічні популяції чи системи і поки що не є корисними в прикладному сенсі. Еволюційні моделі більше схожі на

біологічні системи, мають складну поведінку, мало спрямовані на вирішення технічних завдань. До цих систем відносять так зване «штучне життя».

Еволюційні алгоритми - термін, що часто використовується для загального опису алгоритмів пошуку, оптимізації або навчання, що засновані на формалізованих принципах природного еволюційного процесу. Еволюційні методи призначені для пошуку бажаних рішень і засновані на статистичному підході до дослідження ситуацій та ітераційному наближенні системи до шуканого стану. На відміну від точних методів математичного програмування еволюційні методи дозволяють знаходити рішення, близькі до оптимальних, за прийнятний час, а на відміну від відомих евристичних методів оптимізації характеризуються істотно меншою залежністю від особливостей додатку (більш універсальні) і в багатьох випадках забезпечують кращу ступінь наближення до оптимального рішення.

При викладанні курсу «Еволюційні алгоритми» основна увага приділяється аналізу складності і методам побудови ефективних алгоритмів. Розглядаються математичні основи аналізу алгоритмів; міри складності алгоритмів; методи оцінювання алгоритмів. Приділяється увага розгляду алгоритмів класифікації та кластеризації; індуктивним алгоритмам на деревах пошуку; методам та різновидам евристичних алгоритмів; алгоритмам хешування та побудова хеш-функції; алгоритмам основаних на фільтрах Блума, а також різновидам наближених, паралельних та ймовірнісних алгоритмів.

Мета курсу: сформувати у студентів алгоритмічне мислення на побудову комп'ютерних еволюційних алгоритмів і навчити їх методам і засобам аналізу алгоритмів, та методам побудови алгоритмів для вирішення прикладних задач.

«Еволюційні алгоритми» є *нормативною* дисципліною при підготовці докторів філософії з галузі знань *«Інформаційні технології»*. Для здобувачів спеціальності *123 «Комп'ютерна інженерія»* викладається у 3-му семестрі в обсязі 3 кредитів, в тому числі 30 годин аудиторних занять; з них 10 годин лекцій, 20 годин практичних занять; 60 годин самостійної роботи.

Після вивчення дисципліни студент повинен мати наступні ***знання*:**

- Головні властивості комп'ютерних алгоритмів;
- Основні алгоритмічні моделі;
- Методи аналізу складності алгоритмів;
- Етапи побудови алгоритмів для вирішення складних задач;
- Методи дослідження складності окремих етапів в еволюційних алгоритмах;
- Основи побудови еволюційних алгоритмів класифікації та кластеризації;
- Основи побудови складних гібридних алгоритмів та методи аналізу їх складності.

Студент повинен ***вміти****:*

- Досліджувати структуру складних еволюційних алгоритмів;
- Здійснювати інформаційний і морфологічний опис алгоритму на різних кроках;
- Будувати алгоритми обробки великих даних, класифікації та кластеризації;
- Застосовувати ймовірнісні та наближені методи при побудові еволюційних алгоритмів різного типу;
- Вміти використовувати методи аналізу складності еволюційних алгоритмів при дослідженні складних алгоритмів і процесів.

Загальні компетентності

ЗК01. Здатність до абстрактного мислення, аналізу і синтезу.

ЗК04. Здатність дотримуватися етики досліджень, а також правил академічної доброчесності в наукових дослідженнях та науково-педагогічній діяльності.

ЗК07. Здатність оцінювати та забезпечувати якість виконуваних робіт.

ЗК09. Здатність творчо і креативно мислити.

Спеціальні (фахові, предметні) компетентності спеціальності

СК01. Здатність виконувати оригінальні дослідження, досягати наукових результатів, які створюють нові знання у комп'ютерній інженерії та дотичних до неї міждисциплінарних напрямах і можуть бути опубліковані у провідних наукових виданнях з комп'ютерної інженерії та суміжних галузей.

СК04. Здатність ефективно застосовувати методи аналізу, математичне моделювання, виконувати натурні та математичні експерименти при проведенні наукових досліджень.

СК06. Здатність аргументувати вибір методу розв'язання наукової задачі, критично оцінювати отримані результати та захищати прийняті рішення.

Результати навчання

N1. Мати передові концептуальні та методологічні знання об'єктів професійної діяльності комп'ютерної інженерії і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з комп'ютерної інженерії, IT-інфраструктур та інформаційних технологій, отримання нових знань та/або здійснення інновацій.

N2. Знати сучасні методи проведення досліджень в галузі комп'ютерної інженерії та інформаційних технологій, а саме: способи подання, отримання, зберігання, передавання, опрацювання та захисту інформації, математичні моделі обчислювальних процесів, технології виконання обчислень (високопродуктивних, паралельних, розподілених, мобільних, веб-базованих та хмарних, зелених або енергоефективних, безпечних, автономних, адаптивних, інтелектуальних), а також квантових, біомолекулярних, оптичних та оброблення великих даних тощо, а також технології людино-машинної взаємодії та кооперації, доданої та віртуальної реальності.

N4. Знати і розуміти наукові і математичні положення, що лежать в основі функціонування програмних, програмовних і програмно-технічних комп'ютерних засобів, систем та мереж, Інтернету речей, систем для оброблення великих даних.

N7. Вміти розв'язувати задачі синтезу та аналізу об'єктів дослідження комп'ютерної інженерії та їх окремих складових серед яких: аналогові та цифрові комп'ютери (електронні, квантові, біомолекулярні, оптичні тощо) та комп'ютерні системи універсального або спеціального призначення (стаціонарні, мобільні, вбудовані, розподілені тощо); локальні, глобальні комп'ютерні мережі; кіберфізичні системи, Інтернет речей, системи для оброблення великих даних та штучного інтелекту, IT-інфраструктури; їх програмно- технічні засоби (апаратні, програмні, програмовні, реконфігуровні, системне та прикладне програмне забезпечення), інтерфейси та протоколи взаємодії їх компонентів.

N8. Вміти розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі, інформаційні процеси, технології, методи, способи, інструментальні засоби та системи для автоматизованого та автоматичного проектування; налагодження, виробництва й експлуатації комп'ютерів та комп'ютерних систем і мереж, кіберфізичних систем, Інтернету речей та IT-інфраструктур, розроблення, верифікації та розгортання програмного забезпечення та систем у хмарних та інших середовищах, забезпечення якості, надійності та безпеки а також

ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів комп'ютерній інженерії та дотичних міждисциплінарних напрямах. **N9.** Вміти застосовувати системний підхід, інтегруючи знання з різних дисциплін та враховуючи нетехнічні аспекти, під час розв'язання теоретичних та прикладних задач в предметній області наукових досліджень.

N12. Вміти ефективно поєднувати теорію і практику, задля вирішення науково-прикладних завдань в галузі комп'ютерної інженерії та інформаційних технологій з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

N13. Вміти самостійно проводити експериментальні дослідження в предметній області згідно обраної наукової тематики.

N14. Вміти обґрунтовувати вибір методів розв'язання науково-прикладних задач та критично оцінювати отримані результати, аргументовано захищаючи прийняті рішення.

N17. Здатність адаптуватися до нових умов, самостійно приймати рішення та ініціювати оригінальні дослідницько-інноваційні проекти.

N18. Здатність усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

Знання, які студенти набудуть при вивченні дисципліни «Еволюційні алгоритми», будуть необхідними при подальшому навчанні, а також у науковій діяльності з фахової спеціальності.

3. Програма навчальної дисципліни

№ з/п	Теми	Лекції	Практичні (групові)	Самостійна робота
1	Тема 1. Алгоритми та методи обчислень. Предмет дисципліни.	1	2	6
2	Тема 2. Математичні основи аналізу алгоритмів.	1	2	6
3	Тема 3. Методи розробки алгоритмів.	1	2	6
4	Тема 4. Алгоритми класифікації та кластеризації.	1	2	6
5	Тема 5. Алгоритми на деревах пошуку.	1	2	6
6	Тема 6. Побудова і аналіз евристичних алгоритмів.	1	2	6
7	Тема 7. Алгоритми хешування.	1	2	6
8	Тема 8. Алгоритми основані на фільтрах Блума.	1	2	6
9	Тема 9. Наближені алгоритми. Ймовірнісні алгоритми	1	2	6
10	Тема 10. Паралельні алгоритми.	1	2	6
РАЗОМ		**10**	**20**	**60**

4. Зміст навчальної дисципліни

4.1. План лекцій

№ з/п	Тема заняття / план
1	**Тема 1: Алгоритми та методи обчислень. Предмет дисципліни.** 1) Аналіз якості алгоритмів і розробка методів оцінки якості ефективних алгоритмів. 2) Класифікація алгоритмів.
2	**Тема 2: Математичні основи аналізу алгоритмів.** 1) Заходи складності алгоритмів. 2) Часова та ємкісна складність. 3) Статичні і динамічні міри складності.
3-4	**Тема 3:** Методи розробки алгоритмів. 1) Алгоритми типу «розділяй і володарюй». 2) Динамічне програмування. 3) Жадібні алгоритми. 4) Переборні алгоритми. 5) Пошук з поверненням. 6) Алгоритми локального пошуку.
5-6*	**Тема 4: Алгоритми класифікації та кластеризації.** 1) Алгоритм *K-means*. 2) Алгоритми методу опорних векторів. 3) Алгоритми нормалізації даних. 4) Методи зменшення розмірності. 5) Алгоритм CART. 6) Індуктивні алгоритми.
7*	**Тема 5: Алгоритми на деревах пошуку.** 1) Просторові дерева. 2) Дерево квадрантів. 3) *R*-дерева.
8	**Тема 6: Побудова і аналіз евристичних алгоритмів.** 1) Алгоритм A*. 2) Особливості реалізації. 3) Різновиди алгоритмів евристичного пошуку.
9-10*	**Тема 7: Алгоритми хешування.** 1) Призначення та особливість алгоритмів хешування. 2) Побудова хеш-функції. 3) Стратегії вирішення колізій.
11	**Тема 8: Алгоритми основані на фільтрах Блума.** 1) Призначення та особливість алгоритмів основаних на фільтрах Блума. 2) Головні операції. 3) Застосування та реалізація.

12-13	**Тема 9. Наближені алгоритми. Ймовірнісні алгоритми.** 1) Типи задач, які вирішуються алгоритмами. 2) Особливості реалізації.

14-15	**Тема 10: Паралельні алгоритми.** 1) Типи задач, які вирішуються алгоритмами. 2) Особливості реалізації.

4.2 План практичних (групових) занять

№ з/п	Тема заняття / план
1-3	**Тема 4. Алгоритми класифікації та кластеризації.** *Робота 1. Алгоритми класифікації та кластеризації.* **Мета роботи:** Вивчення основних етапів процедур класифікації та кластеризаці. **Завдання:** І) На заданій вибірці даних реалізувати: 1) Наївний Байєсівський класифікатор. 2) Алгоритм найближчих сусідів. 3) Алгоритм KNN. Порівняти результати роботи алгоритмів. ІІ) На заданому наборі даних реалізувати алгоритм SVM. Провести нормалізацію та агрегацію даних.
4-6	**Тема 5. Алгоритми на деревах пошуку.** *Робота 2.* Алгоритми на деревах пошуку. **Мета роботи:** Вивчення основних етапів алгоритму CART. **Завдання:** Реалізувати алгоритм CART за заданими вхідними даними.
7-9	**Тема 6. Побудова і аналіз евристичних алгоритмів.** *Робота 3. Побудова і аналіз евристичних алгоритмів.* * **Мета роботи:** Освоїти алгоритми A та D. **Завдання:** За варіантом завдання реалізувати евристичний пошук за допомогою алгоритмів A та D. Скласти програмний код; Визначити швидкодію алгоритму; Визначити * складність алгоритму; Зробить висновки.
10-12	**Тема 7. Алгоритми хешування.** *Робота 4. Алгоритми хешування.* **Мета роботи:** Вивчення основних етапів та придбання практичних навиків по побудові алгоритмів хешування. **Завдання:** Побудувати алгоритм хешування за заданими вхідними даними. Скласти програмний код. Визначити складність алгоритму. Визначити швидкодію алгоритму. Зробить висновки.
13-15	**Тема 9. Наближені алгоритми. Ймовірнісні алгоритми.** *Робота 5. Наближені алгоритми. Ймовірнісні алгоритми.* **Мета роботи:** Вивчення основних етапів та придбання практичних навиків по побудові наближених та ймовірнісних алгоритмів. **Завдання:** Побудувати наближені та ймовірнісні алгоритми за заданими вхідними даними. Скласти програмний код. Визначити складність алгоритму. Визначити швидкодію алгоритму. Зробить висновки.

4.3. Завдання для самостійної роботи

4.3.1. Загальні положення

Одним з основних напрямів успішного засвоєння матеріалів навчальної дисципліни є самостійна робота студентів над основною й додатковою літературою з вивчення й використання сучасних комп'ютерних технологій при рішенні вимірювальних задач.

Основними видами самостійної роботи є:

1. Вивчення лекційного матеріалу.
2. Вивчення рекомендованої літератури.
3. Вивчення термінів і основних понять з тем навчальної дисципліни.
4. Підготовка до лабораторних занять і розробка ескізів документів з кожної лабораторної роботи.
5. Підготовка до тестового контролю з навчальної дисципліни.
6. Підготовка до виконання контрольних робіт з навчальної дисципліни.
7. Робота з опрацювання та вивчення рекомендованої літератури.
8. Систематизація вивченого матеріалу перед заліком.

4.3.2 Обов'язкові види самостійної роботи

Фіксований перелік тем для виконання індивідуальних з дисципліни у семестрі студентам не пропонується. Теми обираються студентами самостійно та є засобом поглиблення знань про алгоритми та засоби їх побудови, які розглядаються в межах дисципліни. Крім того, можуть бути розглянутими деякі специфічні використання методів та алгоритмів.

Теми індивідуальних занять узгоджуються з викладачем протягом семестру, до початку залікового тижня.

Теми інформаційних повідомлень співпадають з темами та основними питаннями, які розглядаються на лекціях. В інформаційних повідомленнях також можуть розглядатись новітні засобі та методи в сфері обчислювальних алгоритмів.

4.3.3. Додаткові теми для самостійної роботи

1. Статичні і динамічні міри складності. Тимчасова і місткість складності.
2. Оцінки в гіршому і середньому випадках.
3. Моделі обчислень. Рам і роз-машини.
4. Рівномірний і логарифмічний вагові критерії при оцінці тимчасової і ємнісної складнощів алгоритмів.
5. Нерозгалужені програми, бітові обчислення, дерева рішень.
6. Подання послідовностей, множин, дерев, графів і т. п.
7. Стеки, черги, деки. Способи подання. Операції над ними.
8. Обходи дерев і графів в глибину і ширину.
9. Копіювання дерев. Довжина шляхів.
10. Деревовидні структури для завдання ОБ'ЄДНАТИ - ЗНАЙТИ.
11. Процедури ШУКАТИ та ОБ'ЄДНАТИ і їх модифікації шляхом перебудови даних: стиснення шляху і балансування. Оцінка складності відповідних алгоритмів.
12. Внутрішня сортування (масивів).
13. Нижні оцінки складності алгоритмів сортування, заснованих на порівняннях елементів.
14. Хешування, або метод обчислює адреси. Хеш-функції. Дозвіл колізій.
15. Процедури пошуку, включення і виключення в Хеш-таблицях.

16. Операції над довгими числами.
17. Алгоритми "розділяй і володарюй".
18. Динамічне програмування.
19. "Жадібні" алгоритми.
20. Пошук з поверненням.
21. Алгоритми локального пошуку.
22. Наближені алгоритми.
23. Теоретико-числові алгоритми.
24. Класи P і NP. Поняття NP-повної задачі.
25. Програмування алгоритму ―помнож на три і додай одиницю‖.
26. Програмування алгоритму пошуку виходу з лабіринту.
27. Математична індукція і перевірка правильності алгоритмів.
28. Зв’язані списки, вставлення і видалення елементів зі списку.
29. Робота зі стеком і чергами.
30. Алгоритмічне зображення множин, графів і дерев.
31. Структурне програмування ―з верху до низу‖.
32. Рекурсивні алгоритми, обчислення факторіалів та функції Акермана.
33. Рекурсивний обхід графа у глибину та ширину.
34. Порівняння рекурсії з ітерацією.
35. Алгоритми з відходом назад, тур коня і задача про вісім ферзів.
36. Метод часткових цілій і задача про джип.
37. Метод ―поділяй і пануй‖, надходження мінімумів і максимумів.
38. Динамічне програмування, алгоритм множення матриць.
39. Метод ―гілок і меж‖, задача про комівояжера.
40. Евристичні та наближені алгоритми.
41. Алгоритм пошуку кістякового дерева мінімальної ваги.
42. Булеві програми, комп’ютерна арифметика.
43. Алгоритми на графах; пошук найкоротшого шляху у графі.
44. Аналіз і обчислення арифметичних виразів; скобковий та польський запис.
45. Алгоритми сортування масивів; швидке сортування.
46. Ігрові алгоритми на прикладі ігор ―нім‖ і ―гекс‖.
47. Чисельні алгоритми; зображення дійсних чисел; плаваюча кома.
48. Ймовірнісні алгоритми; генератори випадкових чисел.
49. Скінченні та магазинні автомати; ідентифікація символьних рядків.
50. Машини з довільним доступом до пам’яті.
51. Рекурсивні функції, обчислення рекурсивних функцій.
52. Логічні формули і правила виведення, доведення висловлювань. 53. Предикати і квантори; доведення в теорії предикатів першого порядку.

4.3.4. Вибіркові види самостійної роботи

Студентам пропонується виконання творчих завдань для самостійного опрацювання (одне – за вибором студента). Виконання творчих завдань не є обов’язковим, але може бути зараховане як залікове завдання, що надає шанс набрати бажану кількість балів до сесії.

Кожне творче завдання оцінюється в 10 балів та являє собою практичне завдання з виконання певних підалгоритмів або окремих програмних модулів для моделювання алгоритм. За результатами виконання творчого завдання студент повинен оформити звіт, в якому будуть

задокументовані на скріншотах послідовно всі дії з виконання завдання, до отримання кінцевого очікуємого результату.

4.4. Забезпечення освітнього процесу

- Ноутбук, проектор, екран.
- Комплект слайд-презентацій по курсу.
- Програмне забезпечення для демонстрацій слайд-презентацій.

5. Підсумковий контроль

Кожне іспитове завдання складається з теоретичної та практичної частини. Перелік теоретичних питань наведений нижче:

1. Що називається алгоритмом? Основні властивості алгоритмів.
2. Основні конструкції алгоритмічних мов високого рівня.
3. Що означає перевірка правильності алгоритму?
4. Що означає доведення правильності алгоритму?
5. Що означає тестування алгоритму?
6. Що передбачає аналіз алгоритму на складність?
7. Що таке список і чим він відрізняється від масиву?
8. Що таке стек і які операції можна з ним виконувати?
9. Які структури даних використовуються для зображення графів?
10. Що таке Класифікація?
11. Що таке Кластеризація?
12. Як обходити граф у глибину і ширину?
13. Що таке рекурсивний алгоритм?
14. Порівняйте рекурсивні та ітераційні алгоритми?
15. Що означає принцип ―поділяй і пануй‖ при побудові алгоритмів?
16. Що означає принцип балансування в теорії алгоритмів?
17. Алгоритм динамічного програмування.
18. Що означає алгоритм спрямованого перебору?
19. Що таке метод гілок і границь?
20. Що таке евристичний алгоритм?
21. Основні алгоритми на графах.
22. Як визначити зв’язність графа?
23. Як розв’язати задачу пошуку найкоротших маршрутів у графі?
24. Ітераційний алгоритм пошуку кореня рівняння.
25. Що називається ймовірнісним алгоритмом? Наведіть приклади.
26. Генератори випадкових чисел. Наведіть приклади.
27. Що називається машиною з довільним доступом до пам’яті?
28. Що називається рекурсивною функцією?
29. Що називається алгоритмічною проблемою, яку не можна розв’язати.
30. Що називається рекурсивною і рекурсивно переліченою множиною чисел?
31. Наведіть приклади рекурсивно перелічених, але не рекурсивних множин.
32. Генетичні алгоритми.
33. Мурашині алгоритми.

34. Еволюційні стратегії.
35. Еволюційне програмування.
36. Генетичне програмування.

Повна відповідь по кожному з двох питань (теоретичному та практичному) оцінюється 20 балами.

Сума балів підсумовується з балами, набраними протягом семестру, і округляється з точністю до цілих одиниць. В результаті студенту в індивідуальний план та відомість виставляється підсумок балів та результат за шкалою оцінювання: національною та ЄКТС.

Типові задачі для розв'язання

Практична робота № 0: Алгоритми класифікації. Алгоритм створення байесівського класифікатора

Вступ. Наївний байесовский: класифікатор (*Naive Bayes classifier*) - це простий класифікатор, заснований на використанні теореми Байеса, яка описує ймовірність події з урахуванням пов'язаних з ним умов. Такий класифікатор створюється за допомогою присвоювання міток класів екземплярів задачі. Останні подаються у вигляді векторів значень ознак. При цьому вважається, що значення будь-якого заданої ознаки не залежить від значень інших ознак. Його припущення про незалежність розглянутих ознак і становить наївну частину байєсівського класифікатора.

Послідовність виконання практичної роботи.

Для створенням наївного байєсівського класифікатора необхідно:

1 .Створити новий файл *Python* і імпортувати наступні пакети.

```
import numpy as np import matplotlib.pyplot as
plt from sklearn.naive_bayes import GaussianNB
from sklearn import cross_validation


      from utilities import visualize_classifier
```

2. Завантажити вхідні дані з файлу `data_multivar_nb(N).txt` За відповідним варіантом (Nномер варіанта за журналом).
3. Створити байесівський класифікатор. В даному випадку використовується гауссівський наївний байесовский класифікатор, в якому передбачається, що значення, асоційовані з кожним класом, діють за законом розподілу Гаусса.

```
# Создание наивного байесовского классификатора
 classifier = GaussianNB()
```

4. Провести навчання класифікатора, використовуючи тренувальні дані.

Запустимо класифікатор на тренувальних даних і спрогнозуємо результати.

```
# Прогнозирование значений для тренировочных данных
y_pred = classifier.predict(X)
```

5. Обчислити якість (accuracy) 1 класифікатора, порівнявши передбачені значення з істинними мітками, а потім візуалізувати результат.

```
    # Вычисление качества классификатора accuracy = 100.0 * (y== y_pred)
.sum() / X.shape[0] print("Accuracy of Naive Bayes classifier =",
round(accuracy, 2), "%")
     #Визуализация результатов работы классификатора
visualize_classifier(classifier,  X, y)
```

6. Підготувати звіт.

6. Критерії оцінювання та засоби діагностики результатів навчання *а)*

для денної форми навчання:

№ з/п	Вид діяльності (завдання)	Максимальна кількість балів
1	Практична робота №1	7
2	Практична робота №2	7

10

3	Практична робота №3	7
4	Практична робота №4	7
5	Практична робота №5	7
9	Виконання контрольних тестових завдань (1, 2)	10
10	Самостійна робота студента	15
11	Разом за семестр	**60**
12	**Іспит**	**40**
13	**Всього**	**100**

Критерії оцінювання завдань для досягнення максимальної кількості балів

Максимальна кількість балів (7-6) – студент з високою якістю самостійно виконав весь обсяг робіт, відповідає на всі питання, пов'язані з виконаними роботами, та робить додаткові розрахунки, які йому пропонує викладач. У викладача немає претензій щодо реалізації та вимог до виконання роботи.

5-4 бали – студент з достатньою якістю виконав всі завдання, але в процесі роботи він робив деякі помилки, які, після вказування на них викладачем, самостійно виправляв. На деякі питання він відповідає з похибкою. Запропоновані викладачем додаткові розрахунки робить з деякою потугою. Не всі вимоги до виконання роботи дотримані.

3-2 бали – студент самостійно виконав всі роботи, але якість реалізації недостатня (помилки при розрахунках, не всі вимоги до роботи дотримані). На питання щодо виконання робіт відповідає не зовсім чітко. Є помилки при відповідях.

1 бали – студент самостійно виконав не всі роботи, при цьому якість реалізації недостатня (помилки при розрахунках, не дотримується вимог до оформлення роботи). На питання щодо виконання робіт відповідає не чітко. Є грубі помилки при відповідях.

0 балів – студент не виконав роботу.

При отриманні незадовільної оцінки студент має право виправити всі помилки або виконати нові варіанти завдань, якщо викладач невпевнений, що студент виконав їх самостійно. Такий варіант пропонується, коли студент має багато пропусків занять.

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту	для іспиту
90-100	A	відмінно	зараховано
82-89	B	добре	
75-81	C		
67-74	D	задовільно	
60-66	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з можливістю повторного складання

Студент, який за обидва контрольні заходи набрав менше, ніж 60 бал, здає підсумковий семестровий залік по всьому матеріалу курсу, що вивчається в період екзаменаційної сесії. До заліку студент допускається, якщо отримав протягом семестру не менше 30 балу.

7. Рекомендовані джерела інформації

7.1. Основні:

1. Petrowski A., Ben-Hamida S. Evolutionary Algorithms. Wiley, 2017. 256 p.
2. Fogel D.B., Baeck T., Michalewicz Z. Evolutionary Computation 1: Basic Algorithms and Operators. United States : CRC Press, 2018. 378 p.
3. Mirjalili S.· Evolutionary Algorithms and Neural Networks. Theory and Applications. Springer International Publ., 2018. 156 p.
4. Chen G., Zelinka I. Evolutionary Algorithms, Swarm Dynamics and Complex Networks Methodology, Perspectives and Implementation. Springer Berlin Heidelberg, 2017. 312 p.

7.2 Додаткові:

5. Глибовець М. М., Петльована М. В., Кирієнко О. В. Застосування еволюційних алгоритмів для розв'язання задачі апроксимації зображень многокутниками. НАУКОВІ ЗАПИСКИ НаУКМА. 2017. Том 198. Комп'ютерні науки. С. 21–26.
6. Іщук Я. Ю. Про деякі аспекти навчання еволюційних алгоритмів. С. 246–251. URL: https://sj.npu.edu.ua/index.php/kosn/article/download/44/45
7. Skakalina E.V. Evolutionary algorithms in solving logistical problems. Modelare matematica, optimizare si tehnologii informationale : Materiale Conferintei Internationale, 19–24 martie 2018. Chisinau : Evrica, 2018. Vol. 1. P. 310–315. URL: http://reposit.pntu.edu.ua/handle/PoltNTU/6526.
8. Козін І. В., Селютін Є. К. Особливості пошуку оптимальних класифікацій: еволюційні алгоритми. Вісник Запорізького національного університету. Фізико-математичні науки. 2019. № 2. С. 62–68.

9. Ткачук В. М., Козленко М. І. Класичні генетичні оператори в реалізації квантового генетичного алгоритму. Proceedings of the 2019 Scientific Seminar on Innovative Solutions in Software Engineering, м. Івано-Франківськ, 10 грудня 2019 р. Івано-Франківськ, 2019. С 14–15, doi: https://doi.org/10.5281/zenodo.4082131

7.3. Internet-ресурси:

17. Еволюційні алгоритми: що це таке і для чого вони потрібні. URL: https://irinin.com/navchannia/evolyutsijni-algoritmi-shcho-tse-take-i-dlya-chogo-voni-potribni.html

18. Frąckiewicz M. Використання ChatGPT еволюційних алгоритмів для досягнення можливостей AGI. URL: https://ts2.space/uk/використання-chatgpt-еволюційних-алгоритм/

19. Мартинова О. В., Степанова К. В. Генетичний алгоритм для розв’язання оптимізаційних задач. Сучасні проблеми управління підприємствами: теорія та практика. 18 – 19 березня 2019 року. С. 1–3. URL: http://www.repository.hneu.edu.ua/jspui/bitstream/123456789/21379/1/тези_Мартинова%2C%20Cтєпанова.pdf

20. Генетичні алгоритми. Сучасні методи комп'ютерного моделювання. Мурашиний алгоритм. URL: https://csc-knu.github.io/gen/ant/

Для глибокого оволодіння матеріалом дисципліни необхідно перш за все досконало вивчити перший розділ, який несе в собі основне теоретичне навантаження і методологію вивчення послідуючих розділів. Практичні завдання поєднують матеріал попередніх та наступних розділів, тому їх глибоке засвоєння дає можливості значно легше оволодівати кожним з нових розділів дисципліни.

Студентам для вивчення навчального матеріалу надається конспект лекцій з надлишком навчального матеріалу для самостійного опрацювання, а також перелік літератури для засвоєння теоретичного матеріалу.