

Міністерство освіти і науки України
Департамент науки і освіти Харківської обласної державної адміністрації
Комунальний заклад «Харківська обласна Мала академія наук
Харківської обласної ради»

Відділення інформаційних технологій
Секція: навчальні, ігрові програми та віртуальна реальність

РОЗРОБКА ЗАДАЧ ДО ДИСЦИПЛІНИ «АНАЛІЗ ДАНИХ»

Роботу виконала:

Ласкавцева Софія Дмитрівна,
учениця 11 класу Комунального
закладу «Харківський академічний
ліцей № 45 Харківської міської ради»

Наукові керівники:

Давидов Вячеслав Вадимович,
завідувач кафедри інформаційних
технологій Приватної установи
«Університет науки, підприємництва
та технологій», доктор технічних наук

Арзубов Микола Олексійович, вчитель
інформатики Комунального закладу
«Харківський академічний ліцей № 45
Харківської міської ради»

РОЗРОБКА НАВЧАЛЬНИХ МАТЕРІАЛІВ ДО ДИСЦИПЛІНИ «АНАЛІЗ ДАНИХ»

Ласкавцева Софія Дмитрівна, Харківське територіальне відділення МАН України, Комунальний заклад «Харківська обласна мала академія наук Харківської обласної ради», Комунальний заклад «Харківський академічний ліцей № 45 Харківської міської ради », 11 клас, м. Харків

Давидов Вячеслав Вадимович, завідувач кафедри інформаційних технологій Приватної установи «Університет науки, підприємництва та технологій», доктор технічних наук

Арзубов Микола Олексійович, вчитель інформатики Комунального закладу «Харківський академічний ліцей № 45 Харківської міської ради»

Робота присвячена методології створення навчальних задач для дисциплін, пов'язаних із аналізом даних, штучним інтелектом, комп'ютерним зором. У роботі доведено можливість адаптації специфічних задач для автоматизованого тестування в системі DOTS, розроблено алгоритм та шаблони для створення аналогічних задач.

У роботі представлено контент, що допоможе учням та студентам засвоювати ключові концепції та навички обробки даних: виявлення пропущених значень та аномалій, класифікація та кластеризація, тощо.

Робота висвітлює сучасні методики та підходи до підготовки навчальних матеріалів, важливість адаптації матеріалів до потреб ринку праці та технологічних та аналітичних трендів, що швидко змінюються.

Робота може бути корисною для викладачів ІТ-дисциплін, які бажають покращити та автоматизувати навчальний процес. Крім того, вона може зацікавити фахівців у сфері інформаційних технологій та аналізу даних, які шукають інноваційні методики викладання та навчання.

Ключові слова: навчальні матеріали, аналіз даних, автоматизація тестування, Pandas, NumPy, OpenCV

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. МЕТОДОЛОГІЯ РОЗРОБКИ ЗАДАЧ ДЛЯ СИСТЕМИ DOTS .	6
1.1. Принцип роботи системи тестування DOTS	6
1.2. Розробка та оформлення умови задачі	7
1.3. Розробка тестів для задач в системі DOTS	9
1.4. Розробка чекерів та валідаторів	10
РОЗДІЛ 2. БЛОК ЗАДАЧ «PANDAS»	11
2.1. Бібліотека Python Pandas	11
2.2. Задачі на використання бібліотеки Pandas.	11
2.2.1. Задача А. «Pandas: sum, max, min».	11
2.2.2. Задача В. «Average and Geometric Mean».	12
2.2.3. Задача С. «Timestamp».	12
2.2.4. Задача D. «Statistics».	13
2.2.5. Задача Е. «Certification».	14
РОЗДІЛ 3. БЛОК ЗАДАЧ «NUMPY»	15
3.1. Бібліотека Python NumPy.	15
3.2. Задачі на використання бібліотеки NumPy	15
3.2.1. Задача А. «NumPy: ones, zeroes, full»	15
3.2.2. Задача В. «NumPy: sum, max, min»..	16
3.2.3. Задача С. «NumPy: where».	16
3.2.4. Задача D. «Numpy: poly»	17
3.2.5. Задача Е. «Numpy: linalg»	18
РОЗДІЛ 4. ЗАДАЧА «COUNTING OBJECTS»	19
4.1. Бібліотека Python OpenCV.	19
4.2. Постановка задачі «Counting Objects»	19
4.3. Генерація тестів.	19
ВИСНОВКИ.	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23
ДОДАТКИ	24

ВСТУП

У сучасному інформаційному суспільстві обробка та аналіз даних стають невід'ємною частиною різних галузей діяльності, тому розвиток ефективних методів навчання та створення матеріалів у цій області виявляються надзвичайно важливими.

Обрана тема роботи, «Розробка задач до дисципліни «Аналіз даних», є актуальною через потребу в удосконаленні навчальних програм та методик викладання із застосуванням сучасних підходів та технологій.

Метою цієї роботи є створення ефективних завдань, спрямованих на відпрацювання базових знань та навичок у сфері аналізу даних.

Завдання:

1. Розробити та адаптувати навчальні завдання для системи DOTS, які спрямовані на розвиток навичок в аналізі даних, штучному інтелекті та комп'ютерному зорі;
2. Створити шаблони для автоматизованого тестування аналогічних завдань в системі DOTS;
3. Зробити корисний інструмент для викладачів з метою автоматизації навчального процесу;
4. Покращити навчальний процес і підготовку майбутніх ІТ-спеціалістів, сприяючи вдосконаленню системи DOTS.

Проект розробки задач для дисципліни «Аналіз даних» передбачає підвищення якості підготовки учнів та студентів – майбутніх ІТ-спеціалістів. Предметом роботи є розробка задач із основ датасайнс, придатних для автоматизованого тестування в системі DOTS.

Представлені в роботі набори задач спрямовані на узагальнення ключових концепцій та навичок роботи з бібліотеками мови програмування Python, такими як Pandas, NumPy та OpenCV.

Курс «Introduction to Data Science» викладається в університетах на різних ІТ-спеціальностях. Автоматизація перевірки полегшить та прискорить процес

оцінювання робіт студентів, допоможе впроваджувати більш ефективні методи навчання в освітній процес.

Важливим аспектом роботи є врахування вимог ринку праці та стрімкі зміни технологій і аналітичних трендів. Є сподівання, що робота буде корисною не тільки для викладачів ІТ-дисциплін, які прагнуть вдосконалити та автоматизувати навчальний процес, а також для фахівців у сфері інформаційних технологій та аналізу даних, які шукають інноваційні підходи викладання та навчання.

РОЗДІЛ 1

МЕТОДОЛОГІЯ РОЗРОБКИ ЗАДАЧ ДЛЯ СИСТЕМИ DOTS

DOTS – це тестувальна система для dots.org.ua. Систему тестування DOTS було розроблено в Молодіжному науковому товаристві в 2006-2007 роках для проведення змагань із програмування. Далі ця система трансформувалася в потужну спеціалізовану LMS для навчання програмуванню, алгоритмам та структурам даних. Зараз система DOTS працює на десятках піддоменах загальноосвітніх закладів, вишів, окремих катедр, гуртків програмування, Наприклад, qbit.dots.org.ua, ag45.dots.org.ua, nure.dots.org.ua, hneu.dots.org.ua тощо.

1.1. Принцип роботи системи тестування DOTS

Вебінтерфейс тестувальної системи DOTS надає учням та студентам доступ до наборів алгоритмічних завдань. Такі набори завдань називаються контестами або турнірами, їх створюють користувачі з правами викладача.

Розв'язком задачі є текстовий файл із кодом програми, яку складено однією з дозволених мов програмування. Якщо користувач вважає, що вирішив якусь задачу, він посилає її на перевірку через інтерфейс системи тестування.

Результат перевірки надається через систему тестування протягом декількох хвилин після відправки. Можна надсилати рішення однієї й тієї ж задачі кілька разів або чергувати посилки рішень різних завдань.

Перевірка програми-рішення здійснюється шляхом автоматичного запуску її на кількох секретних наборах вхідних даних (тестів). Конкретний тест зараховується як правильний, якщо програма закінчила свою роботу за вказаний час, не перевищила обмеження за пам'яттю, а формат та вміст вихідних даних є правильними для цього тесту. Рішення задачі вважається правильним тоді і лише тоді, коли воно успішно пройшло всі тести. У більшості типів контестів, що підтримує тестувальна система DOTS, рішення може бути зараховано частково.

На рисунку 1.1. представлена блок-схема алгоритму роботи тестувальної системи DOTS.



Рис. 1.1. Блок-схема алгоритму роботи тестувальної системи

1.2. Розробка та оформлення умови задачі

Умови задач, що використовуються в системі DOTS, розробляються в LaTeX [1]. У лістингу 1.1 представлено шаблон умови задачі в системі DOTS.

```

\begin{problem}{Task Name}{input}{output}{Time Limit}{Memory Limit}
% Тут умова задачі
% Далі треба чітко описати формат вхідних даних
\InputFile
% Далі треба чітко описати формат вихідних даних
\OutputFile
% Далі треба вказати обмеження на вхідні дані
\Constraints
% Приклади
\Examples
\begin{example}
\exmpfile{01.in}{01.out}%
\exmpfile{02.in}{02.out}%
\end{example}
% Пояснення до прикладів (за потреби)
\Explanations
\end{problem}
  
```

Лістинг 1.1. Шаблон задачі (LaTeX)

Після підготовки умови задачі, LaTeX файл конвертується в зображення за допомогою спеціального bash-скрипта, який наведено в лістингу 1.2.

```
#!/bin/bash
cd statements
cat begin.document_ua problem_ua.tex end.document > prob_ua.tex texi2pdf prob_ua.tex
pdftoppm -tiff prob_ua.pdf > prob_ua.tif
convert -transparent white -trim prob_ua.tif problem_ua.png
rm -f *.tif *.pdf *.log *.dvi *.aux prob_*
cd ..
```

Лістинг 1.2. Bash-скрипт makepng.sh

Цей bash-скрипт є інструментом для автоматизації конвертації файлів LaTeX у зображення. Після переходу в потрібний каталог, скрипт об'єднує тексти з файлів «begin.document_ua», «problem_ua.tex», і «end.document» у новий файл «prob_ua.tex». Далі цей файл компілюється в PDF за допомогою команди texi2pdf. Отриманий PDF перетворюється в TIFF формат, використовуючи pdftoppm.

Після конвертації в TIFF, скрипт застосовує програму «convert» для перетворення зображення у формат PNG. Цей процес також включає зміну білого кольору на прозорий та обрізку зображення за його межами.

Скрипт видаляє тимчасові файли, щоб позбутися зайвого сміття. Після завершення процесу конвертації, він повертається до попереднього робочого каталогу.

На рис. 1.2 представлено зображення, яке можна отримати з пустого шаблону за допомогою скрипта makepng.sh.

Ліміт часу:	Time Limit
Ліміт використання пам'яті:	Memory Limit
Формат вхідних даних	
Формат вихідних даних	
Обмеження	
Приклади	
тест	відповідь
Пояснення до прикладів	

Рис. 1.2. Приклад шаблону задачі

1.3. Розробка тестів для задач в системі DOTS

Розробка тестів для задач в системі DOTS є важливою складовою процесу тестування програмного забезпечення. Вхідні та вихідні дані для таких тестів мають вирішальне значення, оскільки вони визначають коректність роботи алгоритмів та правильність їхньої реалізації.

Вхідні дані для задач – це набір параметрів, які передаються алгоритму для обробки. Вони містять у собі різноманітні значення та варіанти вхідних сценаріїв, що дозволяє оцінити поведінку алгоритму в різних умовах. Важливо, щоб вхідні дані були ретельно обрані, адже вони визначають можливі варіації в роботі алгоритму і відображають його адаптивність до різних умов введення. Ось декілька важливих аспектів, які слід враховувати при розробці тестів для:

- покриття коду. Тестові вхідні дані повинні покривати всі можливі гілки коду алгоритму, включаючи усі варіанти умов та шляхи виконання;
- екстремальні та граничні значення. Варто включити в тести екстремальні та граничні значення для оцінки роботи алгоритму в критичних умовах;
- реалістичність даних. Вхідні дані повинні відповідати реальним умовам використання алгоритму;
- порівняння з очікуваними результатами. Важливо перевіряти фактичні вихідні дані алгоритму з очікуваними результатами для виявлення неузгоджень.

Розробка тестів – це ітеративний процес, у якому важливо тримати під контролем якість та покриття усіх важливих аспектів задачі. При розробці тестів до задач, що представлені в цій роботі, використовувалися такі підходи:

- ручна розробка. Зазвичай використовується для створення невеликих тестових даних, які будуть представлені в умові задачі, а також іноді для створення тестів з екстремальними та граничними значеннями;
- автоматизована розробка. Потребує написання власних спеціальних програм-генераторів, які будуть покривати певний аспект або діапазон значень;

- реальні дані. Для задач з основ аналізу даних дуже гарно підходять дані, які можна отримати з відкритих джерел. Це можуть бути, наприклад, датасети з kaggle або дампи таблиць баз даних.

1.4. Розробка чекерів та валідаторів

Розробка чекерів та валідаторів – це важливий крок у забезпеченні коректності та надійності роботи тестувальної системи DOTS.

Чекер – це програма, що забезпечує можливість автоматично порівнювати фактичний вихід алгоритму з очікуваними результатами. Чекер виявляє відмінності або невідповідності в роботі програми, що дозволяє оперативно визначити правильність реалізації.

Валідатор – це програма, що перевіряє тести на відповідність форматам вхідних та вихідних даних, а також контролює дотримання обмежень на вхідні дані, що прописано в умові задачі.

Чекери та валідатори для системи DOTS розробляються за допомогою вільної бібліотеки testlib [2].

Детальніше ознайомитись з практичним застосуванням результатів дослідження можна у довідці про впровадження (Додаток А).

Інформація щодо даних тестового доступу в системі DOTS для ознайомлення з задачами надана у Додатку Б.

РОЗДІЛ 2

БЛОК ЗАДАЧ «PANDAS»

2.1. Бібліотека Python Pandas

Pandas [3] – це потужна бібліотека мови програмування Python [4] для обробки та аналізу даних. Вона надає прості інструменти для роботи з табличними даними. Pandas дозволяє ефективно завантажувати, зберігати, обробляти та аналізувати табличні дані, використовуючи різноманітні функції для фільтрації, групування, агрегації та візуалізації. Бібліотека Pandas може бути використана для побудови моделей машинного навчання або обробки даних перед їх подальшим аналізом.

2.2. Задачі на використання бібліотеки Pandas

У цьому розділі представлено п'ять задач, розв'язування яких передбачає активне використання бібліотеки Python Pandas. До всіх задач розроблено умову, тести, перевіряючі програми, авторські рішення.

2.2.1. Задача А. «Pandas: sum, max, min»

Функція `pd.read_csv()` [8, с. 11] в бібліотеці Pandas використовується для читання даних з файлів у форматі CSV і завантаження їх у `DataFrame` – основну структуру даних у Pandas. Вона дозволяє імпортувати дані з файлів CSV у Python для подальшої обробки та аналізу.

Основні параметри `pd.read_csv()`, що наведені в лістингу 2.1, включають шлях до файлу, що містить дані, а також параметри для налаштування процесу читання, такі як розділювач стовпців, кодування, індексація тощо.

```
import pandas as pd
df = pd.read_csv('input.csv', sep = ';')
```

Лістинг 2.1. Зчитування даних з файлу SCV та збереження їх в `DataFrame`

Коротка умова задачі: «Є прямокутна таблиця з цілих чисел, збережена у файл CSV. Поля таблиці розділяються символом «;», перший рядок таблиці містить заголовки стовпців. Треба обчислити суму всіх чисел в таблиці, а також знайти максимальний та мінімальний елементи в таблиці».

Розробку тестів до цієї задачі можна легко автоматизувати будь-якою мовою програмування. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки `testlib`.

Детальна інформація щодо цієї та інших задач розташована у репозиторії [7] на GitHub.

2.2.2. Задача В. «Average and Geometric Mean»

За допомогою бібліотеки `Pandas` можна легко та швидко рахувати кількість унікальних елементів в таблиці, видаляти стовпці та рядки, створювати нові рядки та стовпці в таблицях.

Коротка умова задачі: «Є таблиця даних. Таблиця зберігається у форматі CSV (Character-Separated Values), поля відокремлюються символом «;». Кожна комірка цієї таблиці може бути: цілим числом, дробовим числом, пустою, заповнена пробільними символами (пробілами або табуляціями) або символьним рядком.

Потрібно спочатку видалити всі рядки, в яких не всі значення є цілими числами. А потім видалити всі стовпчики, в яких середнє арифметичне всіх значень дорівнює середньому геометричному».

Розробку тестів до цієї задачі можна легко автоматизувати будь-якою мовою програмування. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки `testlib`.

2.2.3. Задача С. «Timestamp»

Бібліотека `Pandas` надає широкі можливості для роботи з датами та часом у наборі даних. Ось деякі з основних можливостей для обробки дат та часу в `Pandas`:

- читання та перетворення дати та часу (`pd.to_datetime`);

- отримання компонентів дати та часу;
- сортування за датою (`df.sort_values`);
- групування та агрегування за періодами (`df.groupby`) [8, с. 13];
- заповнення пропущених дат у діапазоні.

Коротка умова задачі: «Дамп однієї з таблиць бази даних зберегли у форматі CSV. Цей файл містить такі дані: мітку час у форматі timestamp (POSIX-час, тобто кількість секунд що минула з 00:00:00 UTC 1 січня 1970 року); id користувача; id задачі; бал за вирішення задачі у відсотках. Треба знайти кількість правильних рішень протягом певного періоду часу».

Тести до цієї задачі були отримані з реальних даних, а саме з дамтів бази даних тестувальної системи DOTS. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки `testlib`.

2.2.4. Задача D. «Statistics»

Бібліотека `Pandas` пропонує різноманітні методи для обчислення статистичних показників. Ці методи допомагають отримати широкий спектр статистичної інформації для числових даних у `Pandas DataFrame`, дозволяючи проводити базовий аналіз та розуміння характеристик даних.

Коротка умова задачі: «Дамп однієї з таблиць бази даних зберегли у форматі CSV. Цей файл містить такі дані: мітку час у форматі timestamp; id користувача; id задачі; id мови програмування, бал за вирішення задачі у відсотках. Треба з цих даних отримати статистику щодо використання мов програмування».

У табл. 2.1 представлено статистику, щодо використання мов програмування на міському турнірі юних інформатиків.

Таблиця 2.1

Статистика щодо використання мов програмування

Мова програмування	Кількість авторів	Кількість рішень	Кількість «ОК» рішень	Кількість «СЕ» рішень
C++	13	462	67	15
Python	12	144	15	11
C#	4	20	0	3
Pascal	3	9	4	3
JavaScript	3	45	2	2
Java	1	2	0	2
	14	696	88	39

Тести до цієї задачі були отримані з реальних даних, а саме з дамтів бази даних тестувальної системи DOTS. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки `testlib`.

2.2.5. Задача E. «Certification»

Бібліотека `Pandas` має велику кількість функцій для обробки і роботи з даними. Найважливіші з них це групування або злиття даних, а також можливість застосувати певну функцію до кожного елемента таблиці.

Коротка умова задачі: «Сертифікація IT SKILLS STANDARD визначає рівень алгоритмічної підготовки учня або студента та складається з трьох алгоритмічних контестів, кожен контест містить три задачі. Кожна задача сертифікації оцінюється максимум у 100 балів, але система перевірки передбачає оцінювання неповних рішень (відсоток від максимального балу).

Нехай учасник сертифікації набрав $p\%$ балів, тоді він може отримати сертифікат: «Very Good» (якщо $60 \leq p < 74$); «Excellent» (якщо $74 \leq p < 90$); «Outstanding» (якщо $90 \leq p \leq 100$).

Дамп однієї з таблиць бази даних сертифікації зберегли у форматі CSV. Цей файл містить такі дані: id користувача; id задачі; id контесту, бал за вирішення задачі у відсотках. Треба за цими даними підбити підсумки сертифікації».

Тести до цієї задачі були отримані з реальних даних, а саме з дамтів бази даних тестувальної системи DOTS. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки `testlib`.

РОЗДІЛ 3

БЛОК ЗАДАЧ «NUMPY»

3.1. Бібліотека Python NumPy

NumPy [5] (Numerical Python) – це потужна бібліотека для мови програмування Python, розроблена для ефективної обробки і аналізу чисельних даних. NumPy забезпечує високопродуктивні структури даних, такі як масиви та матриці, та багатий набір математичних функцій.

3.2. Задачі на використання бібліотеки NumPy

У цьому розділі представлено п'ять задач, розв'язування яких передбачає активне використання бібліотеки Python NumPy. До всіх задач розроблено умову, тести, перевіряючі програми, авторські рішення.

Задачі «В» та «С» мають специфічний формат вхідних даних – NPZ. Це внутрішній формат бібліотеки NumPy для зберігання даних у вигляді масивів з використанням стиснення gzip. Файли NPZ можуть включати один або декілька масивів. Цей формат ефективний для зберігання та обміну великими обсягами даних. Для зчитування файлів типу NPZ використовується функція `np.load()`.

3.2.1. Задача А. «NumPy: ones, zeroes, full»

У бібліотеці NumPy є декілька простих інструментів для заповнення та модифікації матриць і масивів [8, с. 6]:

- `zeroes()` та `ones()`. Заповнення масивів нулями або одиницями відповідно;
- `full()`. Заповнення масивів певним значенням;
- `eye()`. Створення одиничної матриці.

Ці методи NumPy дозволяють створювати, заповнювати та генерувати матриці та масиви з різними значеннями та структурами, що корисно для подальших обчислень та аналізу даних.

Коротка умова задачі: «Нехай N – розмірність квадратної матриці. Напишіть програму з використанням бібліотеки NumPy, яка згенерує матрицю з N рядків і N стовпців. Заповніть цю матрицю нулями. На головній і побічній діагоналях, а також по периметру замініть нулі на одиниці».

Розробку тестів до цієї задачі можна легко автоматизувати будь-якою мовою програмування. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки testlib.

3.2.2. Задача В. «NumPy: sum, max, min»

Бібліотека NumPy має широкий набір функцій для роботи з масивами та матрицями, що дозволяє виконувати різні операції, від математичних обчислень до маніпулювання даними:

- математичні операції;
- операції порівняння та логічні операції;
- статистичні функції;
- індексування та зрізи;
- зміна форми масивів.

Коротка умова задачі: «У файлі NPZ зберігається кілька одновимірних числових NumPy масивів. У кожному масиві треба знайти: суму всіх елементів; мінімальний елемент; максимальний елемент; індекси всіх мінімальних елементів; індекси всіх максимальних елементів».

Розробку тестів до цієї задачі та запис вхідних даних у форматі NPZ було автоматизовано за допомогою бібліотеки NumPy. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки testlib.

3.2.3. Задача С. «NumPy: where»

У NumPy можна вибирати елементи, які задовольняють певним умовам, за допомогою булевих індексів. Ось декілька прикладів:

- застосування умови безпосередньо до масиву:
 - `arr = np.array([1, 2, 3, 4, 5])`
 - `# Вибір елементів, більших за 3`

- `result = arr[arr > 3]`
- комбінування умов:
 - `# Вибір елементів, які більші за 2 та менші або рівні 5`
 - `result = arr[(arr > 2) & (arr <= 5)]`
- використання умов із двома масивами:
 - `arr2 = np.array([10, 20, 30, 40, 50])`
 - `# Вибір елементів з arr2, для яких відповідні елементи arr > 3`
 - `result = arr2[arr > 3]`
- використання функцій:
 - `# Вибір елементів, які задовольняють певну умову наприклад, np.where`
 - `result = np.where(arr > 3, arr, 0)`
 - `# Вибирає елементи > 3, інші замінює на 0`

Коротка умова задачі: «У файлі NPZ зберігається кілька одновимірних числових NumPy масивів. У кожному масиві треба знайти елементи, значення яких знаходиться в діапазоні від 1 до 9 включно».

Розробку тестів до цієї задачі та запис вхідних даних у форматі NPZ було автоматизовано за допомогою бібліотеки NumPy. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки testlib.

3.2.4. Задача D. «NumPy: poly»

Бібліотека NumPy також надає зручні засоби для роботи з поліномами. У цій бібліотеці було розроблено велику кількість різноманітних функцій. Серед них є такі як: операції додавання, віднімання та множення поліномів, а також знаходження значення полінома, його коренів, похідної та інші.

Коротка умова задачі: «Нехай P – поліном n -го ступеня. Вам відомо, що у цього полінома рівно n дійсних коренів. Напишіть програму з використанням бібліотеки NumPy, яка знайде коефіцієнти полінома P ; обчислить значення полінома в заданих точка; обчислить значення похідної від заданого полінома в заданих точках».

Розробку тестів до цієї задачі можна легко автоматизувати будь-якою мовою програмування. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки `testlib`.

3.2.5. Задача E. «Numpy: linalg»

NumPy обладнана великим набором функцій для виконання операцій лінійної алгебри. Прикладами можуть бути функції для розв'язання системи лінійних рівнянь, обчислення детермінанта матриці, транспонування матриці та інші.

Коротка умова задачі: «Нехай A – прямокутна матриця з N рядків та M стовпців. Відомо що $|M - N| = 2$. Треба написати програму, яка видаляє з матриці A два рядки або два стовпці так, щоб вийшла квадратна матриця B з максимальним детермінантом».

Розробку тестів до цієї задачі можна легко автоматизувати будь-якою мовою програмування. Перевіряючі програми реалізовано мовою програмування C++, із використанням вільної бібліотеки `testlib`.

РОЗДІЛ 4

ЗАДАЧА «COUNTING OBJECTS»

4.1. Бібліотека Python OpenCV

OpenCV [6] (Open Source Computer Vision Library) – це бібліотека комп'ютерного зору та обробки зображень, що надає різноманітні інструменти для виконання операцій над зображеннями та відео. Вона дозволяє виявляти об'єкти, обличчя, відстежувати рух, застосовувати фільтри, виконувати кольорову обробку, а також розв'язувати задачі машинного навчання.

4.2. Постановка задачі «Counting Objects»

У зображенні формату «PNG» представлена деяка кількість прямокутників. Вони можуть відрізнятися за параметрами, такими як поворот відносно осей координат, колір (кольорові чи чорно-білі границі і заповнення) та формою кутів (закруглені або гострі). Також варто зазначити те, що прямокутники на зображенні не перетинаються та не накладаються один на одного.

Відповідь у цій задачі одне число, це – кількість прямокутників на зображенні.

Для розв'язання цієї задачі використовується бібліотека Python OpenCV, що забезпечує інструменти для обробки зображень та виконання операцій комп'ютерного зору. Очікується, що програма буде здатна точно визначити кількість прямокутників на зображенні, враховуючи їхні різноманітні властивості.

4.3. Генерація тестів

Для генерації тестів була розроблена програма, що написана мовою програмування Pascal із використанням бібліотеки GraphABC. Вона генерує тестові зображення з прямокутниками, кожен із яких має різні характеристики:

- розмір зображення (W, H): ширина та висота зображення задаються параметрами W (ширина) та H (висота) у коді програми. Ширина згенерованих тестів знаходиться в діапазоні від 100 до 800 пікселів. Висота згенерованих тестів знаходиться в діапазоні від 100 до 800 пікселів;
- мінімальний та максимальний радіус описаного біля прямокутника кола (Rmin, Rmax): вони визначаються параметрами Rmin (мінімальний радіус) та Rmax (максимальний радіус) у коді програми. Мінімальний радіус згенерованих тестів знаходиться в діапазоні від 10 до 25 пікселів. Максимальний радіус згенерованих тестів знаходиться в діапазоні від 100 до 300 пікселів;
- максимальна кількість прямокутників (Count): Максимальна кількість прямокутників на зображенні задається параметром Count у коді програми. Кількість прямокутників згенерованих тестів знаходиться в діапазоні від 100 до 2000. Але варто зазначити, що параметр Count не завжди є відповіддю на тест. Відповідь може бути меншою, коли усі прямокутники неможливо розмістити на зображенні;
- тип прямокутника. Прямокутники можуть бути квадратами або прямокутниками;
- кольори. Можливість згенерувати кольорову чи чорно-білу границю і заповнення прямокутника;
- поворот. Прямокутники можуть бути повернуті на випадковий кут або залишатися паралельними до осей координат;
- форма кутів. Прямокутники можуть мати заокруглені або загострені кути;
- випадковість. Є можливість випадкової генерації усіх параметрів для прямокутників.

Представлено чотири приклади зображень, що були згенеровані програмою (Додаток В).

Для перевірки рішень було згенеровано 70 тестів і відповіді до них. Вони складаються з 7 блоків складності:

1. 01-05: Незаповнені квадрати з чорно-білою границею.
2. 06-10: Незаповнені прямокутники з чорно-білою границею.
3. 11-15: Квадрати з кольоровими або чорно-білими границями, із кольоровим або чорно-білим заповненням, із границею іншого або однакового із заповненням кольору.
4. 16-20: Прямокутники з кольоровою або чорно-білою границею, із кольоровим або чорно-білим заповненням, із границею іншого або однакового з заповненням кольору.
5. 21-40: Квадрати та прямокутники з випадковим кутом нахилу.
6. 41-50: Квадрати та прямокутники із заокругленими кутами.
7. 51-70: Тести з випадковими характеристиками.

Такий набір тестів допомагає легко визначити, чи правильно програма визначає кількість прямокутників на зображенні згідно з вищезазначеними умовами.

ВИСНОВКИ

1. У даній роботі було розроблено та адаптовано навчальні задачі для використання в системі DOTS, спрямовані на розвиток навичок у сфері аналізу даних, штучного інтелекту та комп'ютерного зору. Показано можливість адаптації задач дисципліни «Аналіз даних» під формат автоматизованого тестування в системі DOTS.

2. Розроблено шаблони для створення аналогічних задач, а також забезпечення їх автоматизоване тестування в системі DOTS. Робота сприяє покращенню якості підготовки учнів та студентів у роботі з трьома бібліотеками: NumPy, Pandas, OpenCV. Застосування автоматизованих методів тестування робить навчання більш ефективним та швидким, чим підвищує рівень підготовки учнів.

3. Робота може слугувати корисним ресурсом для викладачів ІТ-дисциплін, спрямованих на покращення та автоматизацію навчального процесу. Крім того, вона може привернути увагу фахівців у сфері інформаційних технологій та аналізу даних, які прагнуть застосовувати сучасні методики викладання та навчання.

4. Розроблені завдання сприяють підвищенню ефективності навчання та підготовки майбутніх ІТ-спеціалістів. Вони також відкривають можливості вдосконалення системи DOTS для більш широкого впровадження в сучасний освітній процес.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. LaTeX Documentation. The LaTeX Project. Офіційна документація. URL: <https://www.latex-project.org/help/documentation/> (дата звернення: 06.03.2024)
2. Testlib. GitHub репозиторій. URL: <https://github.com/MikeMirzayanov/testlib> (дата звернення: 06.03.2024)
3. Pandas Documentation. Pandas. Офіційна документація. URL: <https://pandas.pydata.org/docs/> (дата звернення: 06.03.2024)
4. Python Documentation. Python Software Foundation. Версія 3.12.2. URL: <https://docs.python.org/3/> (дата звернення: 06.03.2024)
5. NumPy Documentation. NumPy. Офіційна документація. URL: <https://numpy.org/doc/stable/reference/> (дата звернення: 06.03.2024)
6. OpenCV. OpenCV.org. Офіційний веб-сайт. URL: <https://opencv.org/> (дата звернення: 06.03.2024)
7. Репозиторій GitHub. Веб-сайт. URL: <https://github.com/gormathos/man> (дата звернення: 06.03.2024)
8. Іванчук Я.В., Месюра В. І., Яровий А. А., Манжілевський О. Д. Інтелектуальний аналіз даних та машинне навчання. Частина 1. Базові методи та засоби аналізу даних. Навчальний посібник. Вінниця: ВНТУ, 2021. 70 с. (дата звернення: 06.03.2024)

ДОДАТОК А

Довідка про впровадження результатів науково-дослідної роботи

УКРАЇНА
 міністерство освіти і науки
 Харківський національний педагогічний університет імені Г.С.Сковороди
 ідентифікаційний код 02125583
 61002, м.Харків, вул.Алчевських, 29
 тел. 700-69-09

ДОВІДКА
 про результати науково-дослідної роботи
 на № _____ Ласкавцевої Софії Дмитрівни, учениці 11 класу КЗ
 «Харківський академічний ліцей №45» Харківської міської ради на тему «Розробка задач до дисципліни «Аналіз даних»»

Розвиток інформаційно-комунікаційних технологій, технологій штучного інтелекту та засобів аналізу даних впливають на всі галузі сучасної економіки, в тому числі й на систему освіти. Потребує змін і зміст підготовки майбутнього вчителя, оскільки саме йому прийдеться готувати покоління учнів до високотехнологічного майбутнього. В ракурсі розв'язання цих проблем знаходиться дослідження Ласкавцевої Софії, яке присвячене розробці набору навчальних задач, пов'язаних із аналізом даних, штучним інтелектом, і комп'ютерним зором.

Розроблені авторські задачі інтегровані у систему автоматизованого тестування в системі DOTS, яка з 2011 року використовується у Харківському національному педагогічному університеті імені Г.С.Сковороди в процесі навчання дисциплін «Основи програмування», «Теоретичні основи інформатики», «Практикум з розв'язування задач з програмування». Для навчального процесу є зручним, що кожна задача, запропонована автором дослідження, супроводжується набором тестів, які уможливають автоматизовану перевірку правильності й коректності отриманих розв'язків. Використання матеріалів наукової роботи Ласкавцевої Софії у навчальному процесі університету сприяло розширенню джерельної бази, методичних підходів до викладання дисципліни «Практикум з розв'язування задач з програмування» та «Основи штучного інтелекту» для здобувачів спеціальності 014.Середня освіта (інформатика).

Результати наукової роботи Ласкавцевої Софії Дмитрівни доповідались на засіданні кафедри інформатики (протокол № 12 від 30 січня 2024 р.) й оцінені як такі, що заслуговують на впровадження в практику підготовки майбутнього вчителя інформатики.

Зав. кафедри інформатики
 Харківського національного педагогічного університету імені Г.С.Сковороди,
 доктор педагогічних наук, професор

Олефіренко Н.В.

Підпис: _____
 засвідчується згідно з актом

Облікові записи в системі DOTS

Для ознайомлення з задачами та тестування розв'язків цих задач відкрито тестовий доступ до відповідних практикумів із програмування в системі DOTS - <https://qbit.dots.org.ua>.

- Обліковий запис для практикуму з Numpy
логін: test_numpy
пароль: numpy
- Обліковий запис для практикуму з Pandas
логін: test_pandas
пароль: pandas
- Обліковий запис для практикуму з OpenCV
логін: test_opencv
пароль: test_opencv

ДОДАТОК В

Приклади згенерованих тестів до задачі «Counting Objects»

