МІНІСТЕРСТВО ОСВІТИ І НАУК УКРАЇНИ

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ ІНФОРМАЦІНИХ ТЕХНОЛОГІЙ

Лабораторна робота №1

з дисципліни «Інтелектуальні інформаційні системи»

на тему:

Огляд і порівняння бібліотек для обробки даних

Виконав:

Студент групи ІПЗ 22

Плиска Д.О.

Перевірив:

Сусла М.В.

Тернопіль 2023

Мета: Ознайомитись з основними інструментами для обробки даних.

Хід роботи

Вибір бібліотеки для обробки даних - це важливий етап в розробці проєкту. Він може суттєво вплинути на продуктивність, якість та швидкість роботи вашого програмного забезпечення. Ось критерії, які варто враховувати при виборі бібліотеки для обробки даних:

**Мова програмування**:

Бібліотека повинна підтримувати мову програмування, яку ви використовуєте (наприклад, Python, R, Julia тощо).

Інтеграція бібліотеки з вибраною мовою програмування повинна бути зручною та ефективною.

**Продуктивність**:

Бібліотека повинна бути продуктивною та швидкою у виконанні операцій обробки даних.

Важливо враховувати обсяги даних, з якими ви плануєте працювати, і відповідність бібліотеки цим обсягам.

**Функціональність**:

Бібліотека повинна мати всі необхідні функції і методи для ваших конкретних завдань обробки даних.

Розгляньте наявність підтримки для роботи з різними типами даних (текст, числа, зображення, часові ряди тощо).

**Доступність**:

Переконайтеся, що бібліотеку легко встановлювати і налаштовувати в вашому середовищі розробки.

Вона повинна бути сумісною з вашою операційною системою та версією мови програмування.

**Спільнота користувачів**:

Активна та велика спільнота користувачів може забезпечити підтримку, розв'язання проблем і навіть розширення функціональності.

Перевірте наявність форумів, стековерфлоу, соціальних мереж та інших ресурсів для обговорення бібліотеки.

**Документація**:

Якість документації важлива для швидкого вивчення та використання бібліотеки.

Переконайтеся, що є вичерпна та зрозуміла документація з прикладами використання.

**Ліцензія**:

Перевірте ліцензію бібліотеки, щоб вона була сумісною з вимогами вашого проєкту (наприклад, відкрита ліцензія для проєктів з відкритим кодом).

**Інтеграція**:

Розгляньте можливості інтеграції бібліотеки з іншими інструментами, які ви вже використовуєте (наприклад, бази даних, візуалізаційні інструменти тощо).

**Підтримка і оновлення**:

Перевірте, як часто бібліотека оновлюється і як швидко розробники виправляють помилки.

Важливо мати доступ до нових версій і отримувати підтримку в разі виникнення проблем.

**Безпека**:

Забезпечення безпеки даних - це важливий аспект при обробці даних. Переконайтеся, що бібліотека має заходи для захисту ваших даних від можливих загроз.

Загалом, вибір бібліотеки повинен відповідати конкретним вимогам та потребам вашого проєкту, а також забезпечувати надійну та ефективну обробку даних.

**2**. **Pandas**:

Pandas - це бібліотека для обробки та аналізу даних в мові програмування Python. Основні характеристики бібліотеки Pandas:

1. Структури даних: Pandas надає два основних типи даних - DataFrame і Series - для зручної роботи з табличними даними та одновимірними масивами даних.
2. Зчитування та запис даних: Бібліотека дозволяє легко читати дані з різних джерел, таких як CSV, Excel, SQL-бази даних, та зберігати результати аналізу у різних форматах.
3. Індексація та вибірка: Pandas дозволяє здійснювати швидку та зручну вибірку даних за допомогою індексів, масок та різних умов.
4. Обробка та очищення даних: Бібліотека надає засоби для фільтрації, сортування, групування та агрегації даних, а також обробки пропущених значень.
5. Візуалізація: Pandas має можливості для створення графіків та візуалізації даних за допомогою інтеграції з іншими бібліотеками, такими як Matplotlib та Seaborn.
6. Підтримка для часових рядів: Бібліотека пропонує потужні засоби для роботи з часовими рядами та аналізу часових даних.
7. Pandas є важливою бібліотекою для аналізу даних у Python і дозволяє вченим даних та аналітикам легко виконувати завдання обробки та візуалізації даних.

**Ray**:

Ray - це бібліотека для програмування на мові Python, яка призначена для розподіленого обчислення та паралельної обробки даних. Вона надає можливість створювати ефективні та масштабовані додатки, які використовують розподілений ресурсний доступ і паралельні задачі. Основні функціональність Ray включає в себе:

1. Розподілене обчислення: Ray надає зручний спосіб створювати обчислювальні задачі та розподіляти їх на різних вузлах кластера для покращення продуктивності.
2. Паралельність: Бібліотека спрощує створення паралельних функцій та обробку даних за допомогою конструкцій, таких як Parallel, Map і інші.
3. Масштабованість: Ray підтримує динамічне масштабування, що дозволяє додавати або знімати ресурси в залежності від потреб додатка.
4. Розподіл стану: За допомогою Ray можна розподіляти стан додатка між різними процесами або вузлами, що дозволяє зберігати та обмінюватися даними між ними.
5. Підтримка задач машинного навчання: Ray включає в себе інструменти для розподіленого навчання моделей машинного навчання та інші завдання, пов'язані з обробкою даних.

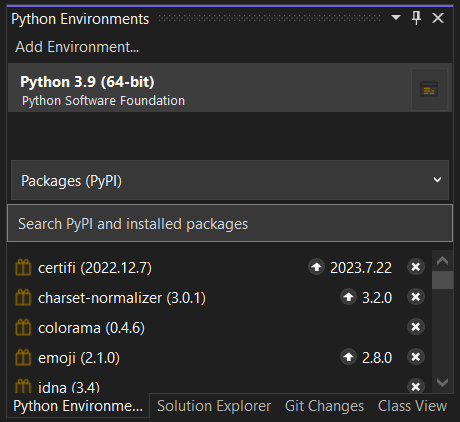
**3.** Підготовка до роботи з бібліотеками:

Першим кроком буде встановлення середовища програмування «Visual Studio» та додатково встановлюємо Python, для роботи в даному середовищі.

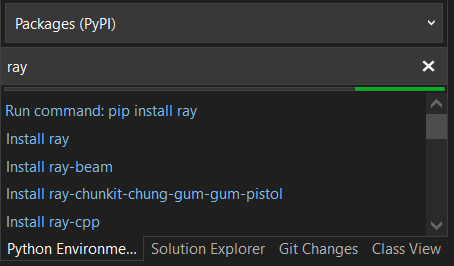
Наступний крок в середовищі програми в пошуковій стрічці вводимо Manage Python Packages, та натискаємо на першу кнопку:



Далі нам стає доступне нове меню, де ми можемо знайти і завантажити наші бібліотеки:



Для завантаження у полі для пошуку вводимо «ray», «numpy» та «pandas» і по черзі натискаємо на кнопку Install “назва бібліотеки”:



Розглянемо основні функціональності бібліотек Ray і Pandas для обробки даних:

**Ray:**

**Розподілені обчислення:** Ray надає можливість створювати розподілені застосунки, що дозволяє використовувати багатоядерні та кластерні ресурси для обчислень. Це корисно для обробки великих обсягів даних та складних обчислень.

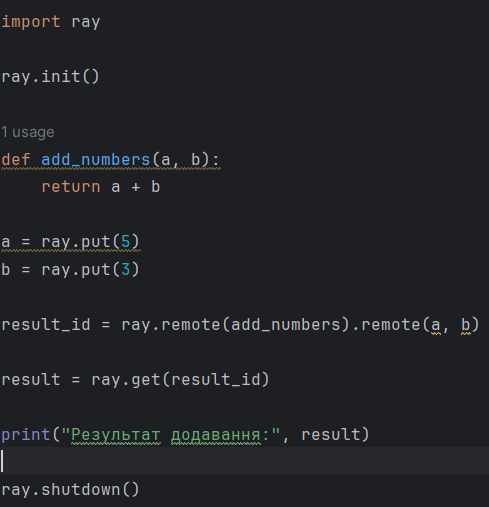
**Task Scheduling:** Ray надає простий спосіб визначення та планування завдань (tasks) для виконання. Ви можете використовувати ray.remote для визначення функцій, які можна виконувати паралельно.

**Розподілені об'єкти:** Ray дозволяє створювати та розподіляти об'єкти між різними процесами та вузлами кластера, що полегшує спільний доступ до даних.

**Розподілені DataFrame (Ray Dataframe):** Ray пропонує розподілену версію DataFrame, яка дозволяє працювати з великими наборами даних та виконувати SQL-подібні запити до них.

**Розподілені засоби для машинного навчання (Ray RLlib):** Розподілена бібліотека для підтримки машинного навчання на базі Ray. Вона включає в себе алгоритми підсиленого навчання і допомагає в навчанні агентів у великих середовищах.

**Приклад використання бібліотеки ray:**

****

У цьому прикладі ми використовуємо Ray для створення віддалених об'єктів a і b, а потім викликаємо функцію add\_numbers паралельно за допомогою ray.remote. Результат обчислення отримується за допомогою ray.get.

**Pandas:**

**DataFrame:** Pandas відомий своєю потужною структурою даних - DataFrame. Вона дозволяє зручно зберігати та обробляти табличні дані. DataFrame подібний до SQL-таблиці або Excel-аркуша, і в ньому можна виконувати операції додавання, вилучення, фільтрації та групування даних.

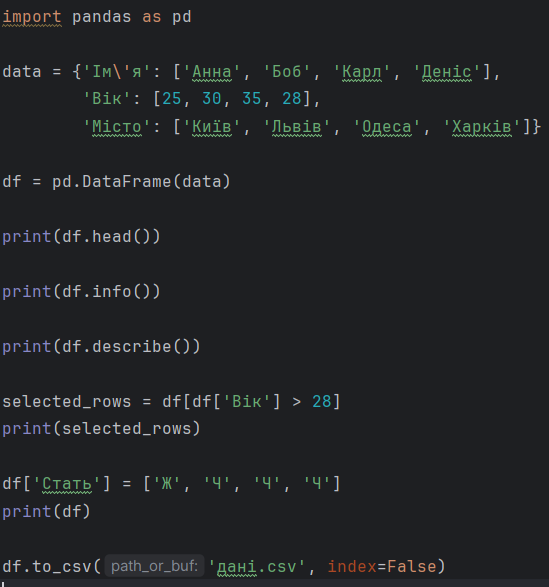
**Зчитування та запис даних:** Pandas підтримує різні формати даних, такі як CSV, Excel, SQL, JSON, і багато інших. Ви можете легко зчитувати дані з цих джерел та зберігати результати обробки.

**Обробка та очистка даних:** Pandas має багато функцій для обробки та очистки даних, таких як вилучення дублікатів, заповнення пропущених значень, перейменування стовпців тощо.

**Агрегація та групування:** Ви можете використовувати Pandas для агрегації та групування даних за допомогою функцій, таких як groupby та agg.

**Візуалізація даних:** Pandas підтримує інтеграцію з бібліотеками для візуалізації даних, такими як Matplotlib і Seaborn, що дозволяє створювати графіки та діаграми для аналізу даних.

**Приклад використання бібліотеки pandas:**



У цьому прикладі ми спершу створюємо DataFrame df зі списком словників, а потім використовуємо різні функції Pandas для роботи з ним, такі як head(), info(), describe(), фільтрація за умовою, додавання нового стовпця та збереження у файл CSV. Pandas надає багато інших функцій для обробки та аналізу даних, і це лише простий приклад його можливостей.

Порівняння бібліотек Ray і Pandas можна провести за кількома ключовими критеріями:

**Розподілені обчислення:**

**Ray:** Ray спеціалізується на розподілених обчисленнях і дозволяє використовувати багатоядерні та кластерні ресурси для паралельної обробки даних. Він призначений для масштабованих обчислень.

**Pandas:** Pandas призначений для роботи з даними на одному комп'ютері і не надає розподіленої обробки.

**Типи даних:**

**Ray:** Ray пропонує розподілені структури даних, такі як розподілені DataFrames і розподілені об'єкти.

**Pandas:** Pandas спеціалізується на табличних даних і надає DataFrame, Series та інші структури для роботи з одним набором даних.

**Обсяг даних:**

**Ray:** Ray призначений для обробки великих обсягів даних, які можуть не поміщатися в пам'ять одного комп'ютера.

**Pandas:** Pandas найбільш зручний для роботи з даними, які поміщаються в пам'ять одного комп'ютера, і не підтримує автоматичну розподілену обробку.

**Швидкодія:**

**Ray:** Ray може бути швидшим для паралельної обробки великих даних за умови правильного використання розподілених обчислень.

**Pandas:** Pandas часто швидший для роботи з невеликими або середніми обсягами даних на одному комп'ютері, оскільки він працює в пам'яті.

**Зручність використання:**

**Ray:** Ray може бути складним для використання через розподілену природу і потребує деякої експертизи в розподілених обчисленнях.

**Pandas:** Pandas є дуже зручним та легким у використанні і є вибором для більшості аналітиків даних та дослідників.

**Сценарії застосування:**

**Ray:** Ray ідеально підходить для завдань, які вимагають великої обчислювальної потужності, таких як обробка великих даних, машинне навчання на кластерах тощо.

**Pandas:** Pandas використовується для аналізу даних, створення звітів, очищення даних та інших завдань, які можна виконати на одному комп'ютері.

**Висновок:** під час виконання лабораторної роботи я вивчив та проаналізував різні аспекти вибору бібліотек для обробки даних, такі як мова програмування, продуктивність, функціональність, доступність та активність спільноти користувачів. Я розглянув різні бібліотеки, провів їх огляд та надав короткий опис кожної. Після цього я встановив ці бібліотеки та детально дослідив їх основні функції та можливості для обробки даних. Нарешті, я провів порівняльний аналіз цих бібліотек з використанням визначених критеріїв.