**Міністерство освіти і науки України**

**Національний університет харчових технологій**

Кафедра інформаційних технологій, штучного

інтелекту та кібербезпеки

**РЕФЕРАТ**

на тему

“JupiterLab та Google Colab спільні та відмінні риси“

Виконав:

Студент групи КН-1-3М

Кучерявий М. В.

**Київ — 2024**

### Вступ

1. Короткий опис JupyterLab і його відмінності від класичного Jupyter Notebook.

### Основні компоненти інтерфейсу JupyterLab

1. Навігаційна панель — доступ до файлів та папок, організація проєктів.
2. Робоча область — основна зона для запуску та редагування комірок коду, блокнотів, текстових файлів.
3. Термінал — вбудована можливість запуску командного рядка для керування процесами та пакетами.

### Google Colab: Особливості та можливості

1. Опис Google Colab — хмарний сервіс для запуску Python-коду
2. Переваги та недоліки Google Colab порівняно з JupyterLab

### Висновок

**Вступ**

JupyterLab — це інтерактивне середовище для роботи з даними, яке є розширеною версією класичного Jupyter Notebook. Відмінністю JupyterLab є його гнучкий, модульний інтерфейс, який дозволяє працювати не лише з блокнотами (ноутбуками), а й з терміналом, текстовими файлами, консолями та інтерактивною візуалізацією в одному робочому просторі.

JupyterLab підтримує багатовіконний режим, де користувачі можуть відкривати кілька вкладок, панелей та вікон, що дозволяє працювати над різними завданнями паралельно та організовувати робочий простір відповідно до своїх потреб. Це значно розширює можливості аналізу даних та полегшує створення складних проєктів. Класичний Jupyter Notebook, натомість, пропонує лише роботу з одним блокнотом на вкладку, з обмеженим інтерфейсом та без можливості одночасно відкривати кілька типів файлів або термінали в одному вікні.

Завдяки своїй функціональності та розширюваності, JupyterLab стає потужним інструментом для аналізу даних, програмування та візуалізації, підходящим для великих проєктів і складних робочих процесів.

### Основні компоненти інтерфейсу JupyterLab

### Навігаційна панель JupyterLab: Доступ до файлів та папок, організація проєктів

Навігаційна панель JupyterLab є основним інструментом для управління файлами та проєктами в цьому середовищі. Вона розташована зліва в інтерфейсі та дозволяє користувачам швидко отримати доступ до файлів, тек та проєктів, створювати нові документи та організовувати робочі простори.

#### Основні можливості навігаційної панелі:

1. **Огляд файлової системи**: Навігаційна панель надає доступ до локальної файлової системи користувача або до кореневої папки JupyterLab на сервері. Користувачі можуть переглядати всі файли й папки, що є доступними, у тому числі блокноти, текстові файли, зображення, а також інші типи файлів, підтримувані JupyterLab. За допомогою простої структури директорій можна швидко переходити між різними частинами проєкту.
2. **Створення та управління файлами і папками**: Навігаційна панель дозволяє створювати нові папки, блокноти, скрипти та інші типи файлів. Це спрощує організацію проєктів, дозволяючи зберігати файли різних типів в окремих папках і систематизувати роботу над проєктом.
3. **Перетягування файлів**: Завдяки функції «drag-and-drop» (перетягування), навігаційна панель дозволяє користувачам переміщувати файли та папки, організовуючи робочий простір відповідно до власних потреб. Це робить структуру проєкту гнучкою та інтуїтивно зрозумілою.
4. **Пошук файлів та швидкий доступ**: Панель підтримує пошук за назвами файлів, що особливо корисно для великих проєктів з численними документами. Це дозволяє швидко знайти потрібний файл або папку.
5. **Зручний доступ до інструментів і розширень**: Навігаційна панель інтегрується з різноманітними розширеннями JupyterLab. Користувачі можуть швидко запускати термінали, консолі або інші інструменти, не залишаючи основний робочий простір, що підвищує ефективність роботи.
6. **Віддалена організація проєктів**: Якщо JupyterLab запущений на віддаленому сервері, навігаційна панель все одно дозволяє доступ до серверної файлової системи. Це важливо для великих проєктів, які зберігаються на сервері, а не локально, та дозволяє зручно працювати над проєктами з будь-якого місця.

### Робоча область: Основна зона для запуску та редагування комірок коду, блокнотів, текстових файлів

Робоча область JupyterLab є центральною частиною інтерфейсу, в якій відбувається більшість дій користувача: запуск і редагування коду, створення блокнотів, редагування текстових файлів, виконання скриптів і аналіз даних. Вона спроєктована так, щоб забезпечити гнучкість і зручність роботи з різними типами файлів, підтримуючи широкий спектр завдань в одному інтегрованому середовищі.

Основні можливості робочої області: запуск та редагування комірок коду. Робоча область дозволяє легко створювати та виконувати комірки коду, що є ключовим елементом в JupyterLab. Користувачі можуть швидко писати, запускати і перевіряти код, а також виконувати аналітичні операції та отримувати миттєвий результат прямо в робочій області. Це робить процес написання коду інтерактивним і зручним.

Блокноти (ноутбуки) Jupyter: Робоча область JupyterLab підтримує блокноти, що дозволяють комбінувати комірки коду з текстовими поясненнями, візуалізаціями та математичними формулами. Блокноти є основним інструментом для документування та аналізу даних, оскільки вони дозволяють структурувати робочий процес і зрозуміло представляти результати.

Текстові файли та редактор коду: Робоча область включає текстовий редактор з підтримкою різних мов програмування. Це дозволяє редагувати файли коду Python, JavaScript, Markdown та інші формати прямо в середовищі JupyterLab. Текстові файли можна використовувати для збереження заміток, коментарів, документації та додаткового коду, який може бути корисним у проєкті.

Візуалізація даних: Робоча область підтримує інтерактивні графіки та діаграми, які можна створювати безпосередньо в блокнотах за допомогою таких бібліотек, як Matplotlib, Seaborn та інші. Це дає можливість візуалізувати великі обсяги даних та миттєво бачити їх зміни залежно від параметрів аналізу.

Мультизадачність та гнучкість: У робочій області можна одночасно відкривати кілька блокнотів, текстових файлів, терміналів і консолей. JupyterLab дозволяє працювати в багатовіконному режимі, переміщати та розділяти робочі вікна відповідно до своїх потреб. Це підходить для складних проєктів, де потрібно одночасно обробляти кілька видів даних або виконувати різні операції.

Підтримка терміналів і консолей: Робоча область підтримує вбудовані термінали, що дозволяють користувачам виконувати командний рядок прямо в інтерфейсі. Консолі з інтегрованою підтримкою Python та інших мов надають ще одну можливість для тестування фрагментів коду без необхідності створення блокнотів.

Розширення функціоналу: Робоча область JupyterLab інтегрується з розширеннями, що дозволяють додавати нові функції, як-от підтримку інших мов програмування, роботу з базами даних, інтеграцію з хмарними сервісами та інші можливості. Це робить JupyterLab дуже гнучким інструментом для вирішення різноманітних завдань.

### Термінал JupyterLab: Вбудована можливість запуску командного рядка для керування процесами та пакетами

Термінал в JupyterLab дозволяє виконувати командний рядок безпосередньо в середовищі JupyterLab, забезпечуючи зручний доступ до операційної системи та зовнішніх утиліт. Це спрощує робочий процес, дозволяючи виконувати операції, такі як керування файлами, запуск фонових процесів, інсталяція бібліотек та робота з віртуальними середовищами, не покидаючи інтерфейсу.

Термінал дозволяє працювати з різними оболонками командного рядка, такими як Bash, Zsh та інші, підтримує інсталяцію та оновлення пакетів через pip або conda, а також взаємодіє з іншими інструментами JupyterLab, такими як блокноти та консолі.

### Google Colab: Особливості та можливості

### Google Colab — хмарний сервіс для запуску Python-коду

Google Colab (або Colaboratory) — це безкоштовний хмарний сервіс, який надає можливість запускати Python-код у інтерактивних блокнотах. Це рішення забезпечує зручний інтерфейс для виконання аналізу даних, машинного навчання, наукових обчислень і тестування моделей без необхідності встановлення програмного забезпечення або наявності потужних локальних обчислювальних ресурсів.

Google Colab підтримує структуру блокнотів, подібно до Jupyter Notebook, де можна комбінувати код, текст, графіки та візуалізації. Це робить роботу з даними зручною та зрозумілою.

Однією з найбільших переваг Google Colab є можливість використання потужних обчислювальних ресурсів безкоштовно, таких як графічні процесори (GPU) та тензорні процесори (TPU). Це дає змогу працювати з великими наборами даних і виконувати складні обчислення, наприклад, тренування нейронних мереж.

Colab повністю працює в хмарі, що означає, що користувачам не потрібно турбуватись про налаштування середовища або обчислювальну потужність. Всі обчислення виконуються на серверах Google, а самі блокноти зберігаються в Google Drive, що забезпечує доступність з будь-якого пристрою.

Блокноти, створені в Colab, автоматично зберігаються в Google Drive, що дозволяє легко ділитися ними, працювати з ними на різних пристроях і забезпечує автоматичне збереження.

Colab дозволяє легко імпортувати популярні бібліотеки для машинного навчання, такі як NumPy, Pandas і багато інших. Це забезпечує гнучкість і простоту в роботі з даними та побудові моделей.

Подібно до інших продуктів Google, Colab дозволяє кільком користувачам одночасно працювати над одним блокнотом, що спрощує командну роботу та колаборацію.

Colab також дозволяє завантажувати файли безпосередньо з локальної машини або з хмарних сховищ, таких як Google Drive, що дає зручний доступ до даних.

**Переваги та недоліки Google Colab порівняно з JupyterLab**

Google Colab має кілька суттєвих переваг порівняно з JupyterLab. Одна з основних — це безкоштовний доступ до потужних обчислювальних ресурсів, таких як графічні процесори (GPU) та тензорні процесори (TPU), що дозволяє значно пришвидшити виконання складних обчислень, зокрема тренування глибоких нейронних мереж. Це робить Colab ідеальним вибором для тих, хто займається машинним навчанням або працює з великими наборами даних, оскільки дозволяє виконувати ресурсоємні завдання без необхідності мати потужне локальне обладнання. В JupyterLab для роботи з такими ресурсами потрібно або мати доступ до потужних серверів, або налаштовувати зовнішні сервіси, що може бути складніше та вимагає більше часу.

Крім того, Colab повністю працює в хмарі, що усуває необхідність налаштовувати сервери або турбуватися про обчислювальні потужності. Все, що потрібно — це веб-браузер та доступ до Інтернету. Це робить платформу доступною для користувачів з будь-яких пристроїв без потреби в спеціальному програмному забезпеченні або додаткових налаштуваннях. У JupyterLab, хоча також можливе хмарне використання, зазвичай потрібно самостійно налаштовувати сервери чи використовувати локальні ресурси, що може бути складніше, особливо для новачків.

Ще однією перевагою Google Colab є інтеграція з Google Drive, що дозволяє автоматично зберігати робочі блокноти та інші файли. Це забезпечує легкий доступ до них з будь-якого пристрою та дозволяє здійснювати спільну роботу, надаючи можливість редагувати документи одночасно кільком користувачам. В JupyterLab файли зберігаються локально, і для доступу з різних пристроїв або для спільної роботи потрібно використовувати сторонні сервіси чи налаштовувати спеціальну синхронізацію.

Colab пропонує простоту використання завдяки інтуїтивно зрозумілому інтерфейсу та відсутності необхідності в додаткових налаштуваннях, що робить платформу привабливою для новачків. Всі основні бібліотеки для машинного навчання, такі як NumPy, Pandas та інші, вже встановлені, що дозволяє швидко приступити до роботи без необхідності їх встановлювати вручну. У JupyterLab також можна працювати з цими бібліотеками, але їх інсталяція та налаштування можуть зайняти більше часу, особливо якщо потрібно працювати з великою кількістю різних інструментів або конфігурацій.

З іншого боку, у Google Colab є ряд обмежень, які можуть бути недоліками для деяких користувачів. Одним з таких обмежень є ліміти на час роботи сеансів та використання GPU/TPU. Наприклад, сеанс може автоматично завершитись через певний період бездіяльності, а доступ до GPU або TPU також має обмеження за часом або обсягом використання, що може бути незручним для великих або тривалих проєктів. У JupyterLab такого роду обмежень немає, оскільки ви самостійно контролюєте ресурси та час роботи.

Іншим обмеженням є залежність від Інтернет-з'єднання для роботи з Colab. Якщо з'єднання з Інтернетом нестабільне або відсутнє, робота з Colab стає неможливою. У JupyterLab ж можна працювати в автономному режимі, що дає можливість виконувати задачі навіть без підключення до мережі, якщо все налаштовано локально.

Ще однією суттєвою відмінністю є гнучкість налаштувань. У Google Colab ви обмежені певними параметрами середовища та не можете здійснювати глибокі налаштування операційної системи або встановлювати деякі сторонні утиліти. В JupyterLab ж є більше можливостей для налаштування, створення віртуальних середовищ та інтеграції з іншими інструментами. Це дозволяє гнучко адаптувати робоче середовище під специфічні вимоги проєкту, що робить JupyterLab кращим вибором для більш складних або кастомізованих робочих процесів.

Щодо мов програмування, Google Colab в основному підтримує Python, хоча є можливість використання й інших мов, але вони не настільки широко підтримуються. JupyterLab, навпаки, має велику кількість доступних ядер для різних мов, таких як Python, R, Julia, Scala та інші, що робить його більш універсальним для різних типів проєктів.

Нарешті, JupyterLab має більше можливостей для інтеграції з іншими інструментами для аналізу даних, таких як IDE, сторонні бібліотеки або платформи для роботи з великими даними. В Google Colab такі можливості більш обмежені, оскільки вся робота відбувається в межах одного середовища, і інтеграція з іншими системами може бути складнішою.

**Висновок**

Google Colab і JupyterLab — це потужні інструменти для аналізу даних та машинного навчання, кожен з яких має свої переваги та обмеження. Colab ідеально підходить для користувачів, які шукають простоту, безкоштовний доступ до GPU/TPU і хмарну інфраструктуру без необхідності налаштовувати середовище. Це робить його зручним для новачків та тих, хто працює з великими обчислювальними задачами. Однак, для користувачів, які потребують більшої гнучкості в налаштуванні середовища або офлайн-роботи, JupyterLab може бути кращим вибором завдяки своїй універсальності та можливості інтеграції з іншими інструментами. Вибір між цими платформами залежить від конкретних потреб користувача та вимог до проєкту.