

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КУРСОВА РОБОТА

з дисципліни «Спеціальні розділи програмування»

На тему: «Знайомство з можливостями Anaconda/IPython Notebook.

Аналіз та візуалізація даних датасету»

Виконав:
студент групи ФІ-12
Завалій Олександр

Contents

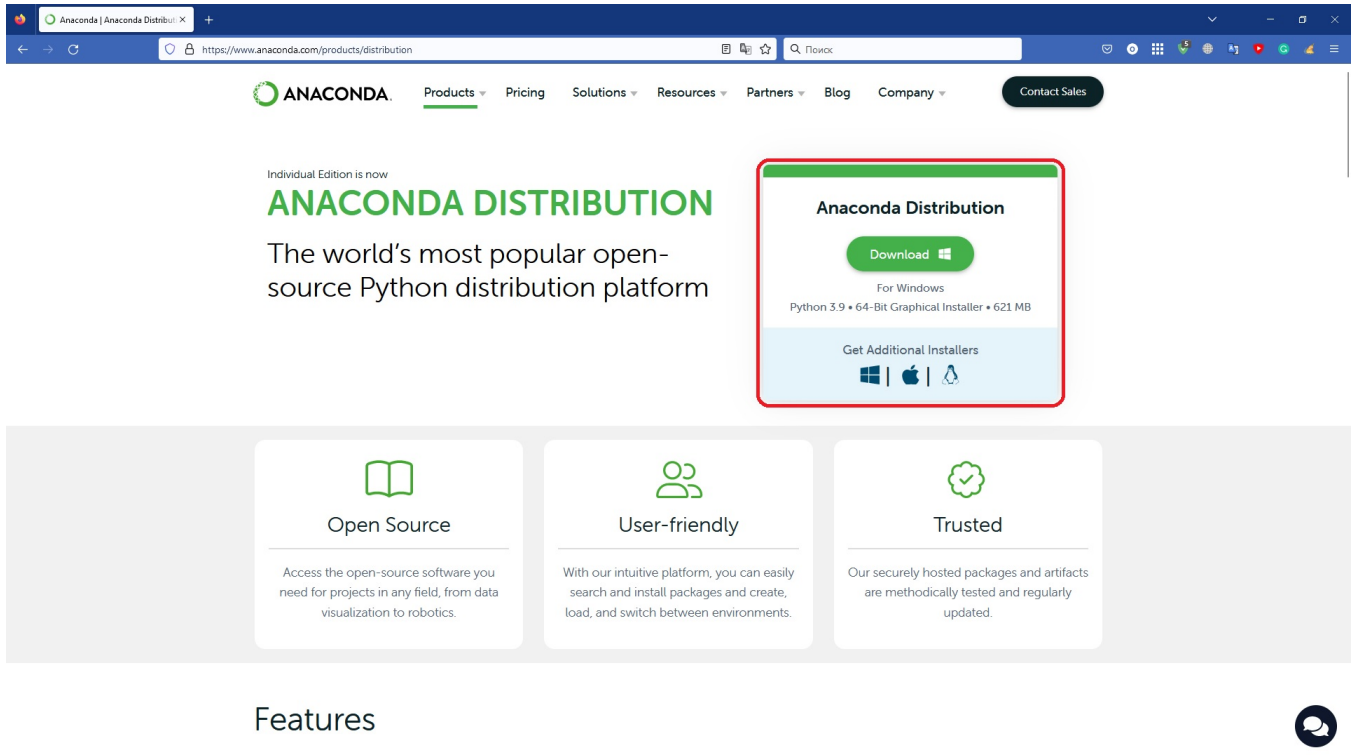
1	Частина	3
1.1	Знайомство з можливостями Anaconda/IPython Notebook	3
1.2	IPython Notebook та Visual Studio Code.	5
1.3	Почнемо знайомство з Jupyter Notebook у Visual Studio Code.	6
1.3.1	Довідка по плагіну "Python".	6
1.3.2	Налаштування середовища Python	6
1.4	Початок роботи.	9
1.5	Source control(Git)	10
2	Частина	12
2.1	Знайомство з датасетом.	12
2.2	Аналіз та візуалізація даних у середовищі IPython Notebook на модельних наборах даних.	13

1 Частина

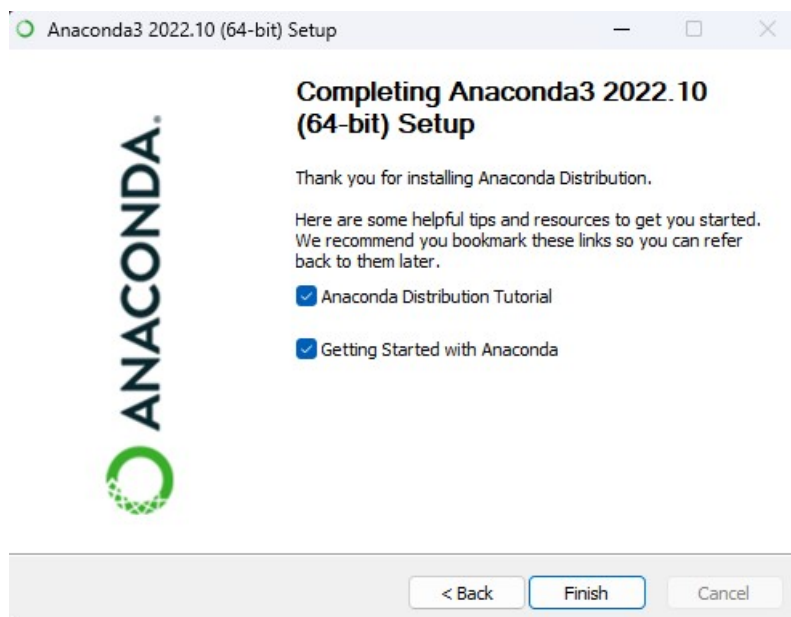
1.1 Знайомство з можливостями Anaconda/IPython Notebook

Для того, щоб почати працювати з Anaconda потрібно її встановити:

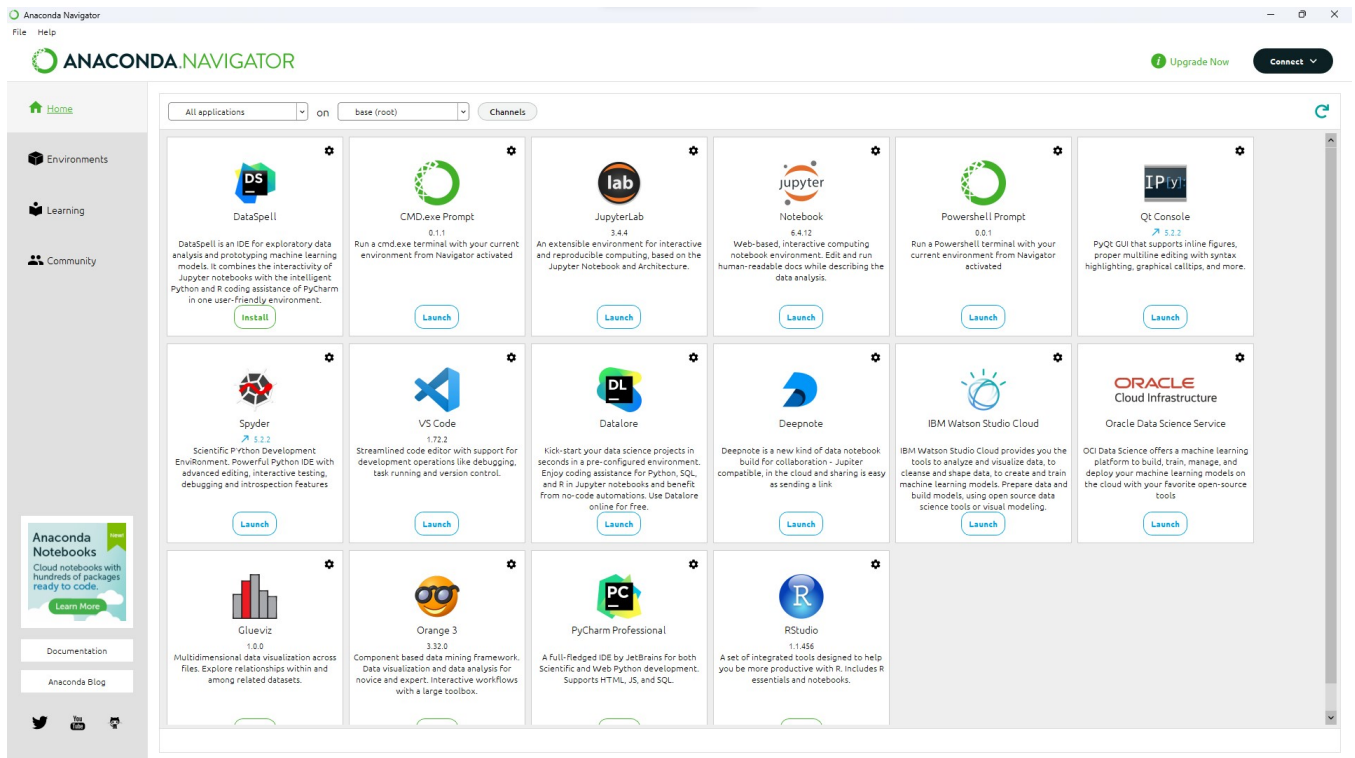
1. Перейти на офіційний сайт [Anaconda Distribution](https://www.anaconda.com/products/distribution) та натиснути кнопку "Download". Також можна безкоштовно спробувати Anaconda в вашому браузері після реєстрації.



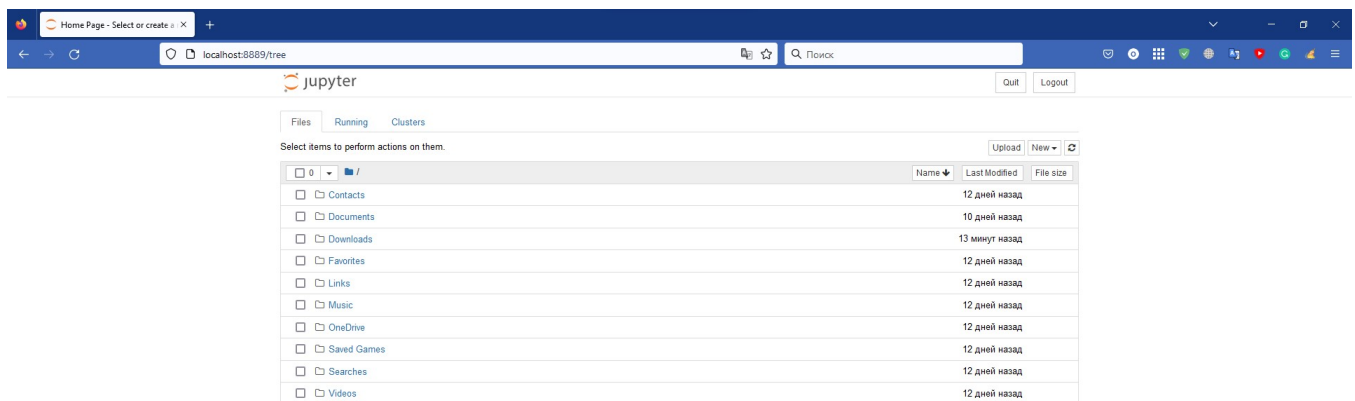
2. Після завантаження встановлюємо застосунок. Отримаємо таке повідомлення.



3. Далі відкриваємо застосунок Anaconda Navigator і обираємо Jupyter Notebook. Або одразу відкрити Jupyter Notebook. Він знаходиться в папці, куди була встановлена Anaconda.



4. Ось так виглядає Jupyter Notebook.



1.2 IPython Notebook та Visual Studio Code.

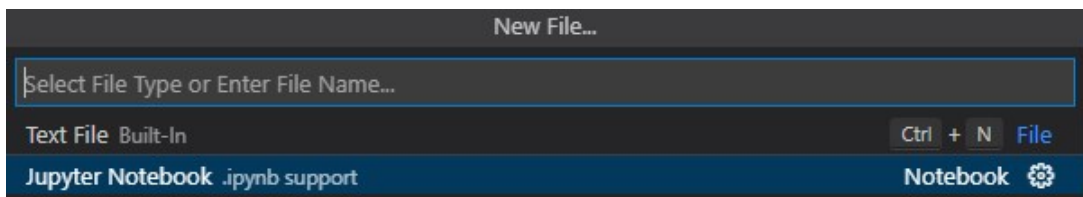
Надалі я буду працювати з IPython Notebook у редакторі вихідного коду Visual Studio Code. Нижче наведені переваги використання VC Code.

1. Редактор VC Code підтримує декілька мов програмування, у тому числі: JavaScript, C++, PHP, Python, Java, Objective-C, PowerShell, Visual Basic, Markdown, JSON, HTML, CSS... Тому дуже просто писати код в IPython Notebook в одній вкладці, та паралельно редагувати Latex документ наприклад.
2. Data viewer. Провідник змінних Jupyter Variables показує додаткову корисну інформацію про розмір і тип кожної змінної. Також є можливість переглядати DataFrame або Series на окремій вкладці, щоб не завантажувати файли на інші платформи для перегляду.
3. Code formatting. Використовується формат коду, такий як yapf або black, для форматування більш складного коду pandas. VS Code застосує вибраний формататор, щоб очистити ваш вкладений код.
4. Split editors. Дуже просто писати код в IPython Notebook в одній вкладці, та паралельно редагувати Latex документ в іншій вкладці.
5. Git integration. VS Code легко інтегрується з git.
6. Change kernels. Якщо ви використовуєте conda або віртуальне середовище, дуже корисно мати можливість швидко змінити середовище.
7. Supports WSL. VS Code добре інтегрується з WSL, тому ви можете розробляти на Windows або Linux.
8. Plugins.

1.3 Почнемо знайомство з Jupyter Notebook у Visual Studio Code.

Для того, щоб працювати з Jupyter Notebook у Visual Studio Code потрібно:

1. Перейти на офіційний сайт [Visual Studio Code](#) і завантажити редактор коду.
2. Натиснути комбінацію клавіш "Ctrl+Shift+X" відкриється розділ "Extensions" тут потрібно завантажити Python.
3. Створюємо або відкриваємо папку у VS Code explorer.
4. Натискаємо "File-New File-Jupyter Notebook"

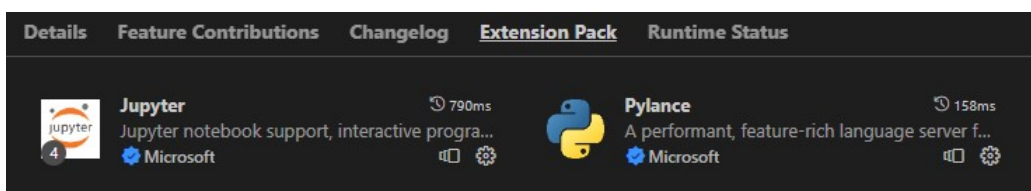


5. Зберігаємо файл у папці.

1.3.1 Довідка по плагіну "Python".

Цей плагін офіційно був створений та підтримується компанією Microsoft. Розширення включає такі функції, як IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, тощо.

Тобто Python автоматично встановить розширення [Pylance](#) та [Jupyter](#), щоб забезпечити вам найкращий досвід роботи з файлами Python і блокнотами Jupyter.



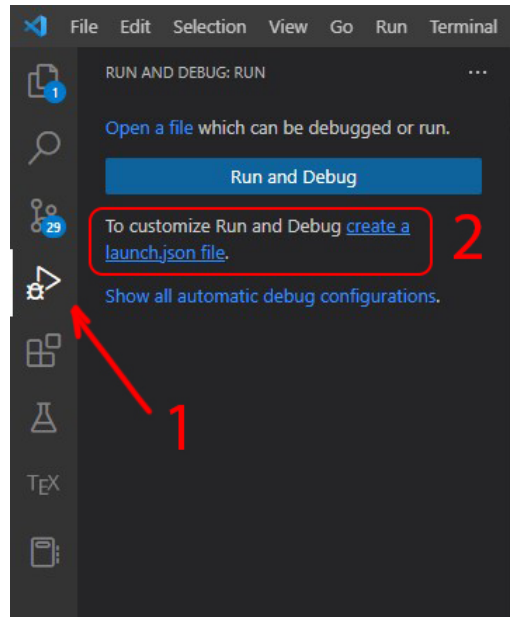
Однак [Pylance](#) є необов'язковий, тобто розширення Python залишатиметься повністю функціональним без нього. Також є можливість його видалити без втрати функціоналу.

1.3.2 Налаштування середовища Python

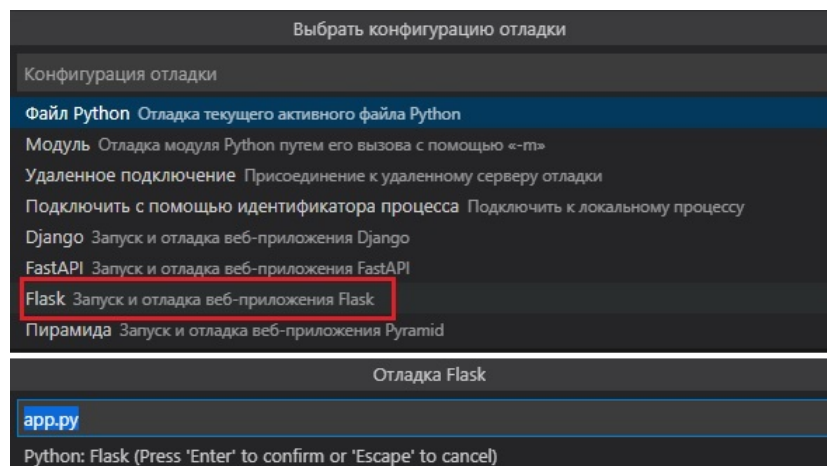
1. Виберіть інтерпретатор Python.



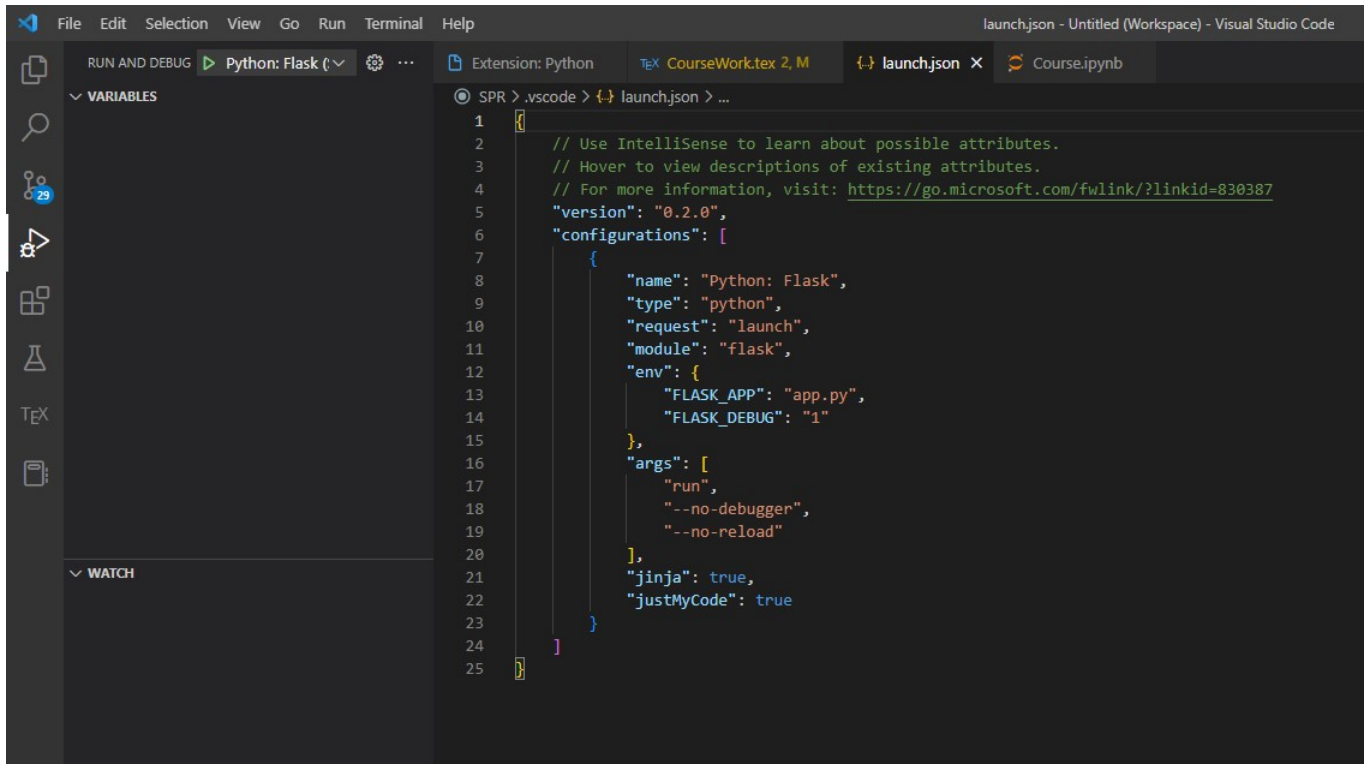
2. Настройте debugger через Debug Activity Bar.



3. Выбираем поле "Flask" и нажимаем "Enter".



4. У цьому файлі можна налаштувати середовище Python та усі встановлені плагіни.

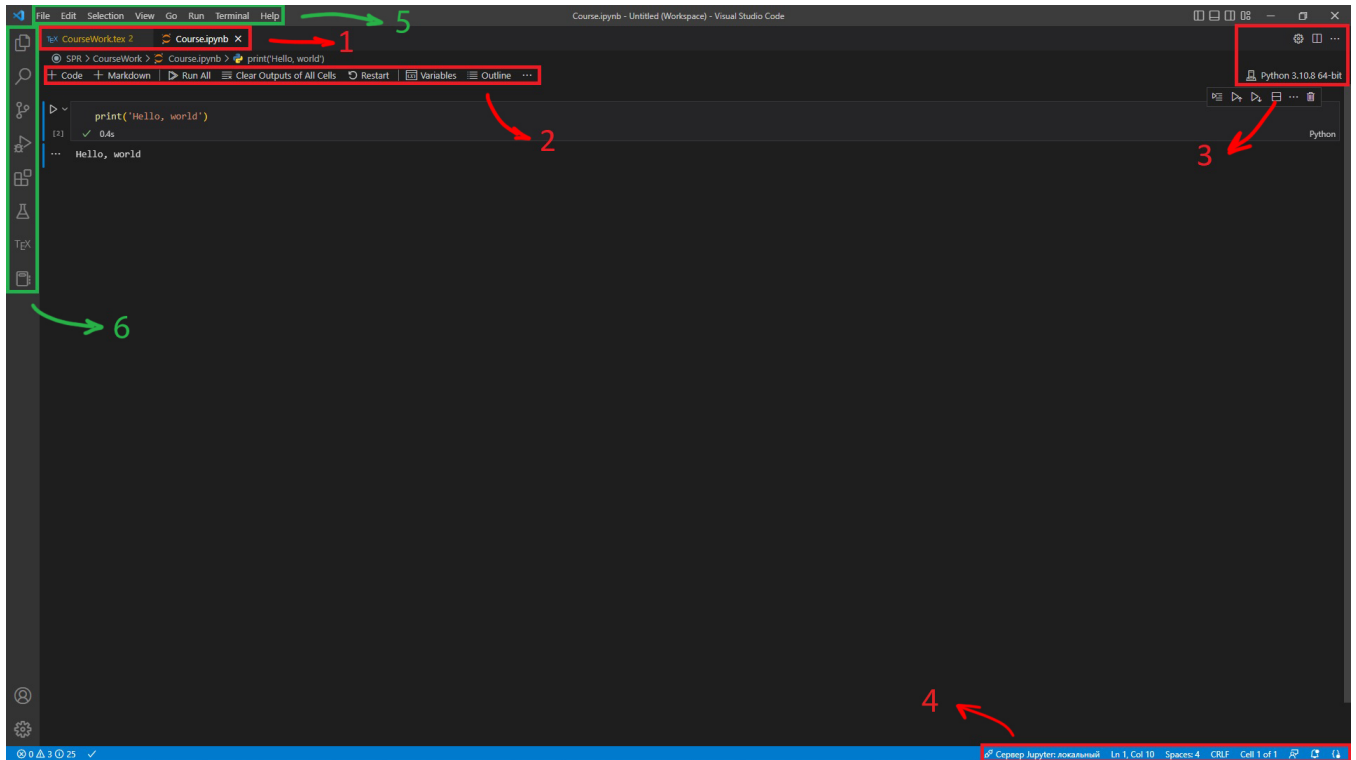


The screenshot shows the Visual Studio Code interface with the `launch.json` file open in the editor. The file is located in the `.vscode` directory. The configuration is for the `Python: Flask` environment. The `launch.json` file contains the following JSON configuration:

```
1 {
2     // Use IntelliSense to learn about possible attributes.
3     // Hover to view descriptions of existing attributes.
4     // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5     "version": "0.2.0",
6     "configurations": [
7         {
8             "name": "Python: Flask",
9             "type": "python",
10            "request": "launch",
11            "module": "flask",
12            "env": {
13                "FLASK_APP": "app.py",
14                "FLASK_DEBUG": "1"
15            },
16            "args": [
17                "run",
18                "--no-debugger",
19                "--no-reload"
20            ],
21            "jinja": true,
22            "justMyCode": true
23        }
24    ]
25 }
```

The interface also shows the `Extension: Python` tab and the `launch.json` file in the Explorer view. The `WATCH` section is visible in the bottom left corner.

1.4 Початок роботи.



1. Відкриті вкладки. Так званий Split editors.

2. Панель взаємодії з Notebook.

- Створення комірки.
- Додати markdown текст.
- Виконати всі комірки.
- Прибрати відображення вихідних даних в усіх комірках.
- Перезавантажити комірки.
- Відкрити провідник змінних Jupyter Variables.
- VS Code explorer.

3. Панель налаштування середовища.

- Кастомізація Notebook, додавання нумерації, зміна теми, тощо...
- Зміна ядра. Є можливість підключитися до Jupyter Server.

4. Status bar.

5-6 Взаємодія з VS Code.

- Зберегти файл, створити, редагувати, переглянути, виконати, відкрити термінал, тощо...
- VS Code explorer, пошук файлів, контроль версій (Git), дебаггер, "магазин" плагінів, встановлені плагіни.

1.5 Source control(Git)

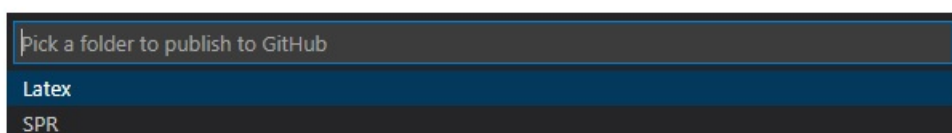
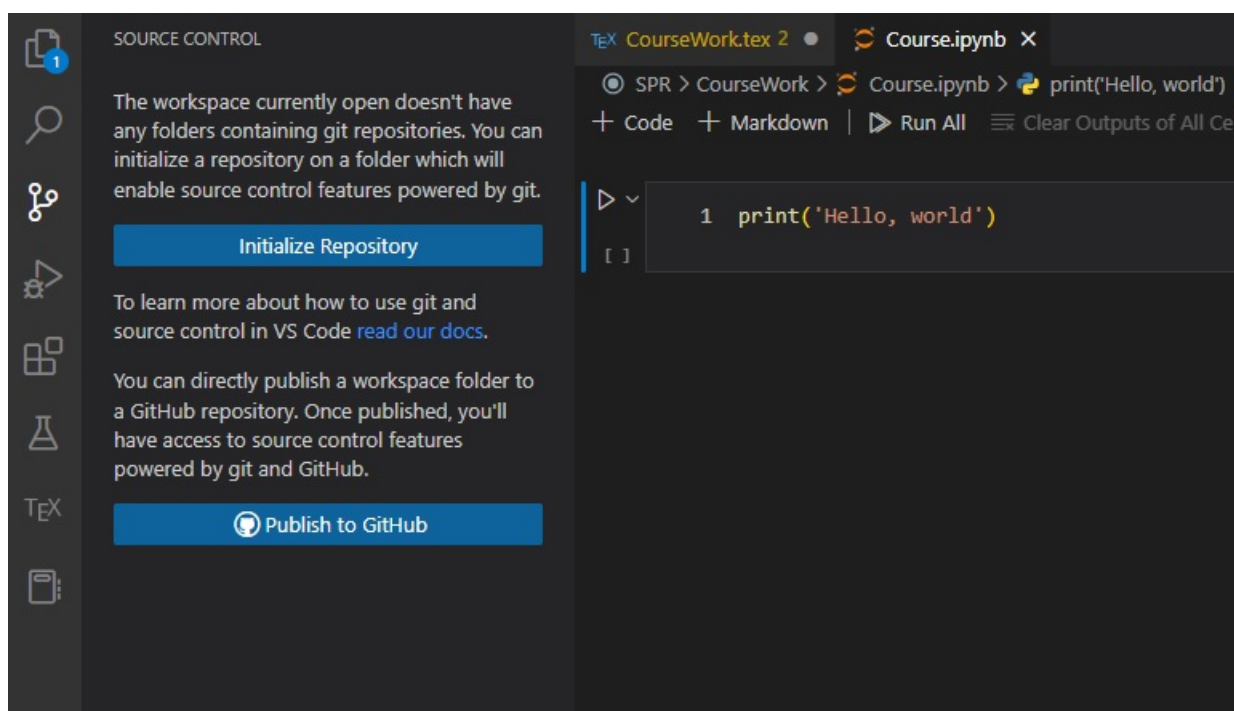
Source control - це клас систем, відповідальних за керування змінами в комп'ютерних програмах, документах, великих веб-сайтах чи інших колекціях інформації.

GitHub - це система для керування та контролю версіями за допомогою Git.

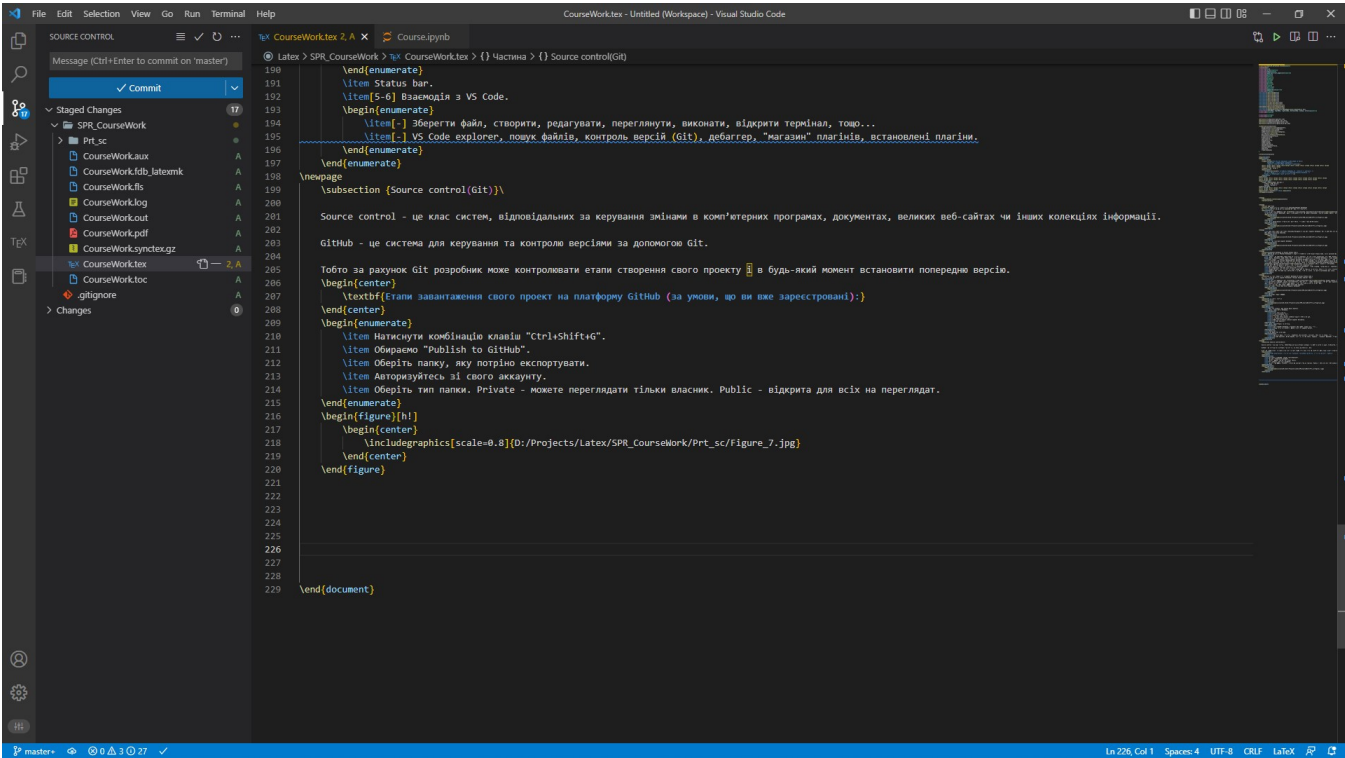
Тобто за рахунок Git розробник може контролювати етапи створення свого проекту і в будь-який момент встановити попередню версію.

Етапи завантаження свого проект на платформу GitHub (за умови, що ви вже зареєстровані):

1. Натиснути комбінацію клавіш "Ctrl+Shift+G".
2. Обираємо "Publish to GitHub".
3. Оберіть папку, яку потрібно експортувати.
4. Авторизуйтеся зі свого аккаунту.
5. Оберіть тип папки. Private - можете переглядати тільки власник. Public - відкрита для всіх на переглядат.



Тепер усі зміни у проєкті будуть зберігатись на GitHub.



2 Частина

2.1 Знайомство з датасетом.

Встановити Cryptocurrency Prices Dataset можна з сайту Kaggle.

About Dataset

Цей набір даних містить історичні ціни та обсяги 4 криптовалют з 9 листопада 2017 року по 27 серпня 2022 року.

- BTC - Bitcoin
- BNB - Binance coin
- ETH - Ethereum
- USDT - Tether

2.2 Аналіз та візуалізація даних у середовищі IPython Notebook на модельних наборах даних.
