

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

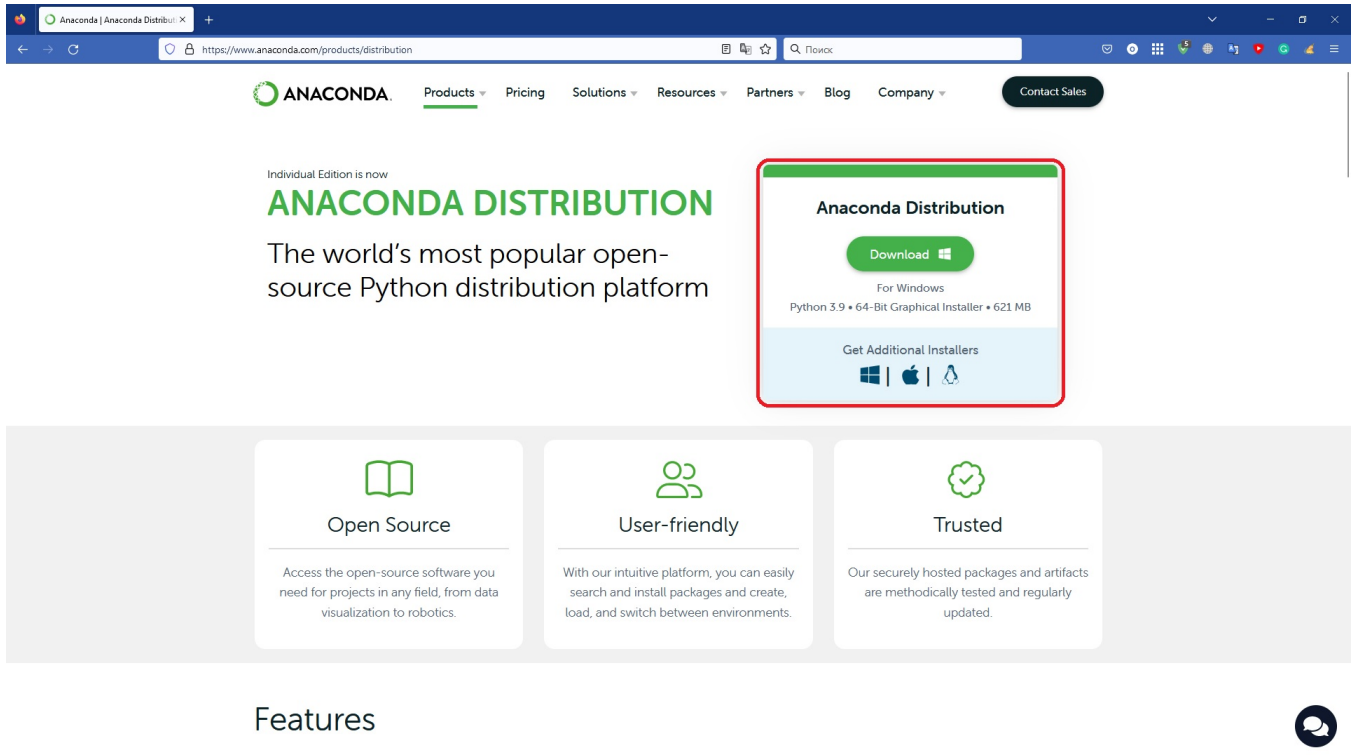
1	Частина	3
1.1	Знайомство з можливостями Anaconda/IPython Notebook	3
1.2	IPython Notebook та Visual Studio Code.	5
1.3	Почнемо знайомство з Jupyter Notebook у Visual Studio Code.	6
1.3.1	Довідка по плагіну "Python".	6
1.3.2	Налаштування середовища Python	6
1.4	Початок роботи.	9
1.5	Source control(Git)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

1 Частина

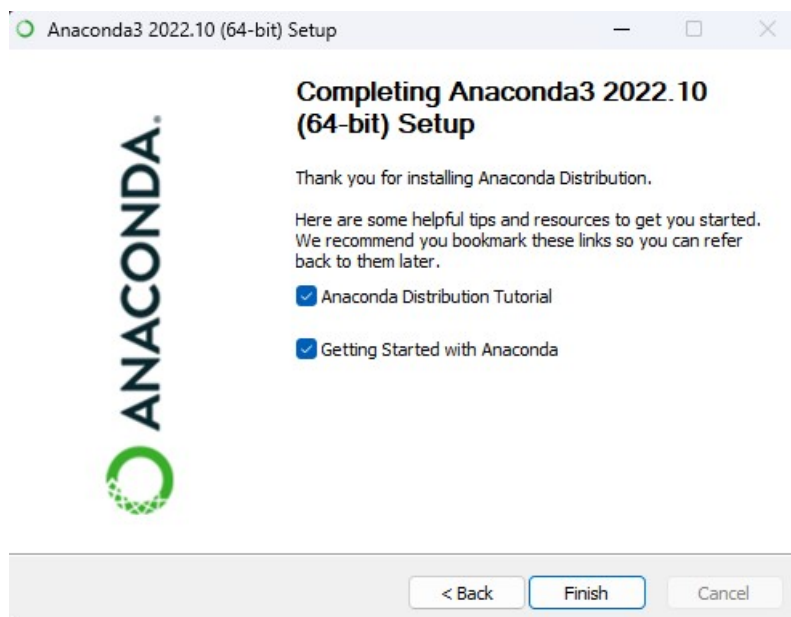
1.1 Знайомство з можливостями Anaconda/IPython Notebook

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

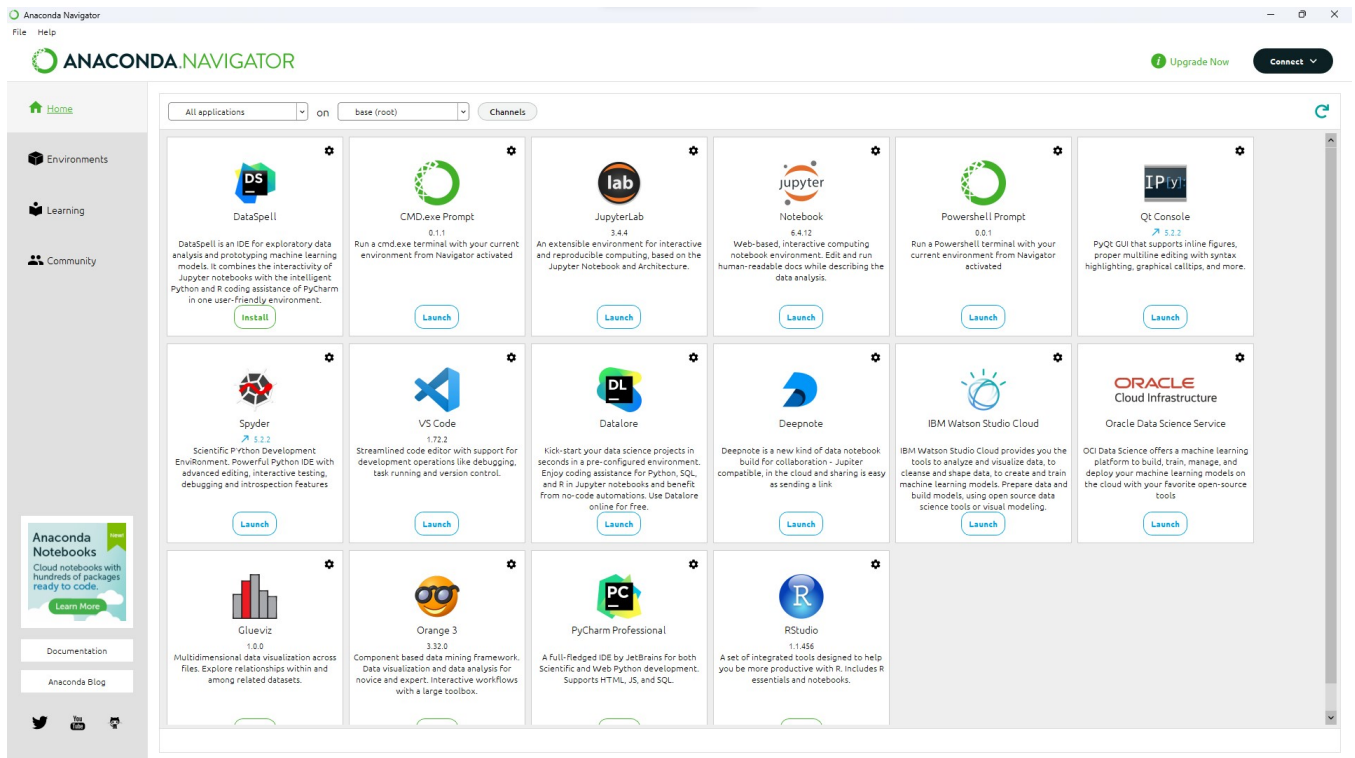
1. Перейти на офіційний сайт [Anaconda Distribution](https://www.anaconda.com/products/distribution) та натиснути кнопку "Download". Також можна безкоштовно спробувати Anaconda в вашому браузері після реєстрації.



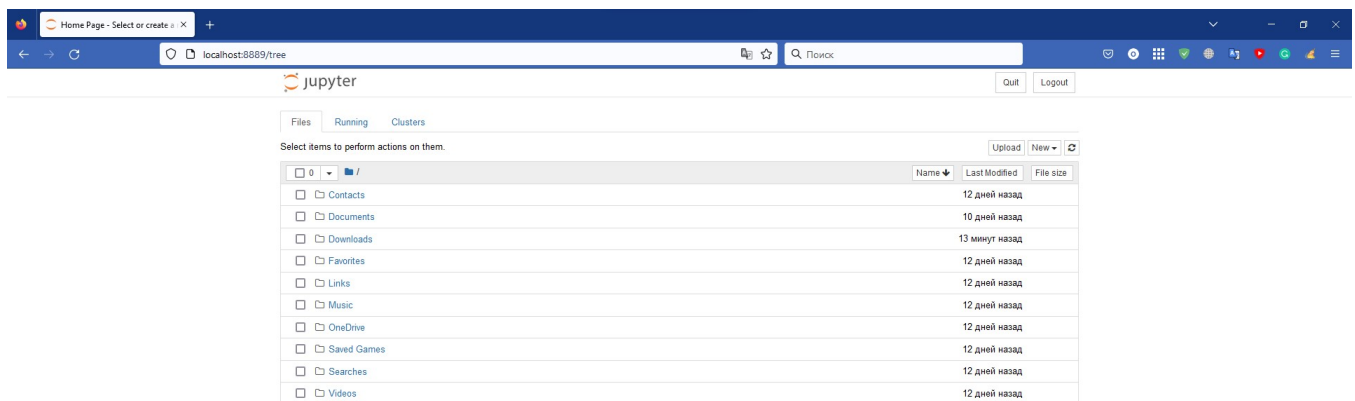
2. Після завантаження встановлюємо застосунок. Отримаємо таке повідомлення.



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.

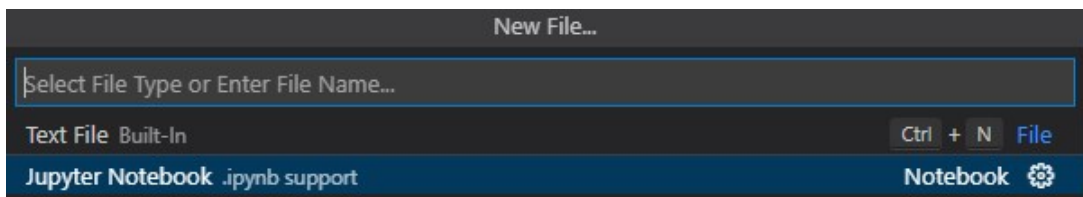
Надалі я буду працювати з IPython Notebook у редакторі вихідного коду Visual Studio Code. Нижче наведені переваги використання VC Code.

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"

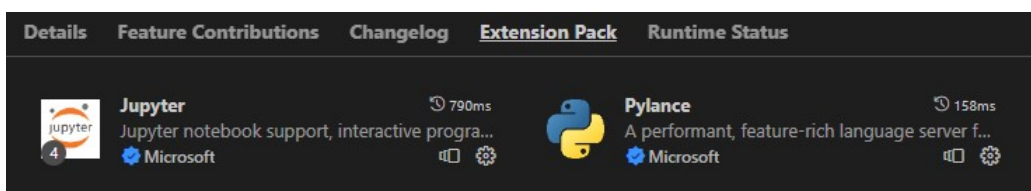


5. Зберігаємо файл у папці.

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.



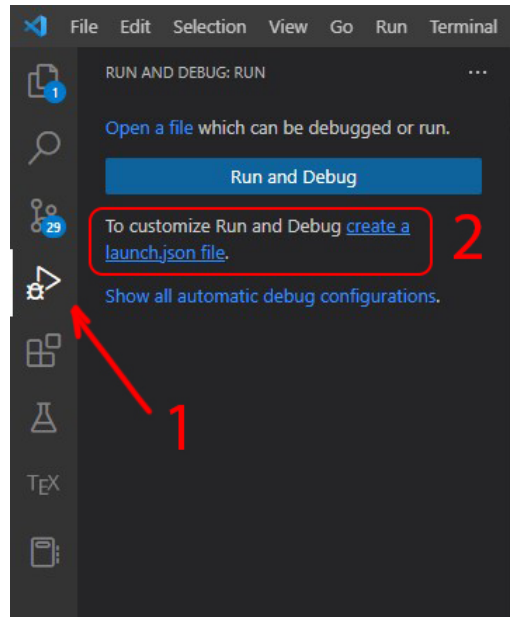
Однак [Pylance](#) є необов'язковий, тобто розширення Python залишатиметься повністю функціональним без нього. Також є можливість його видалити без втрати функціоналу.

1.3.2 Налаштування середовища Python

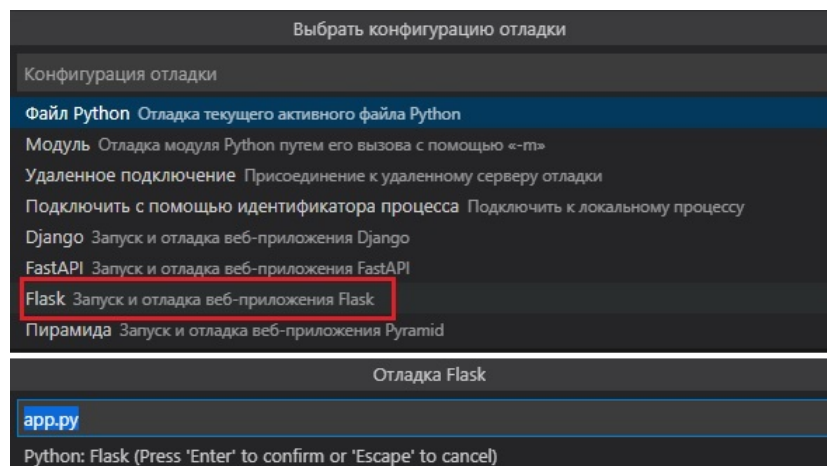
1. Виберіть інтерпретатор Python.



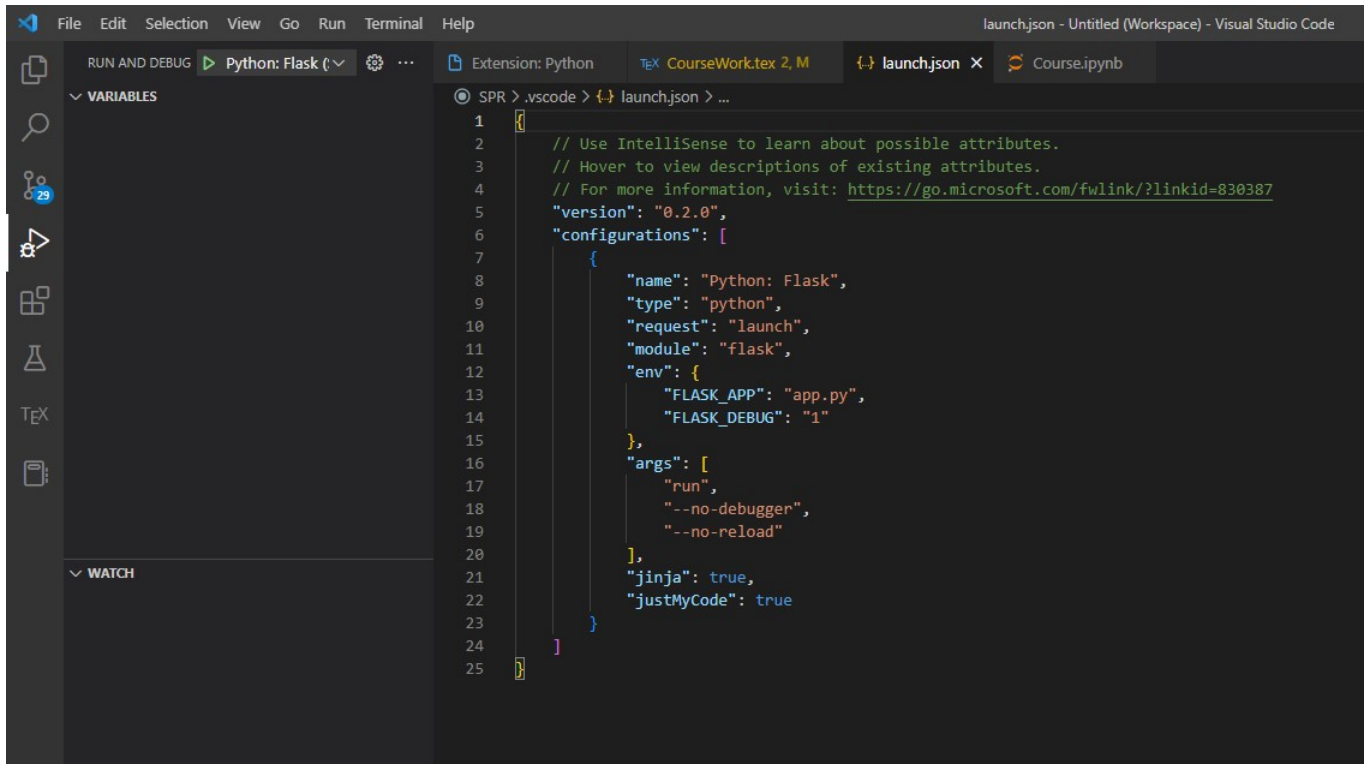
2. Настройте debugger через Debug Activity Bar.



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



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

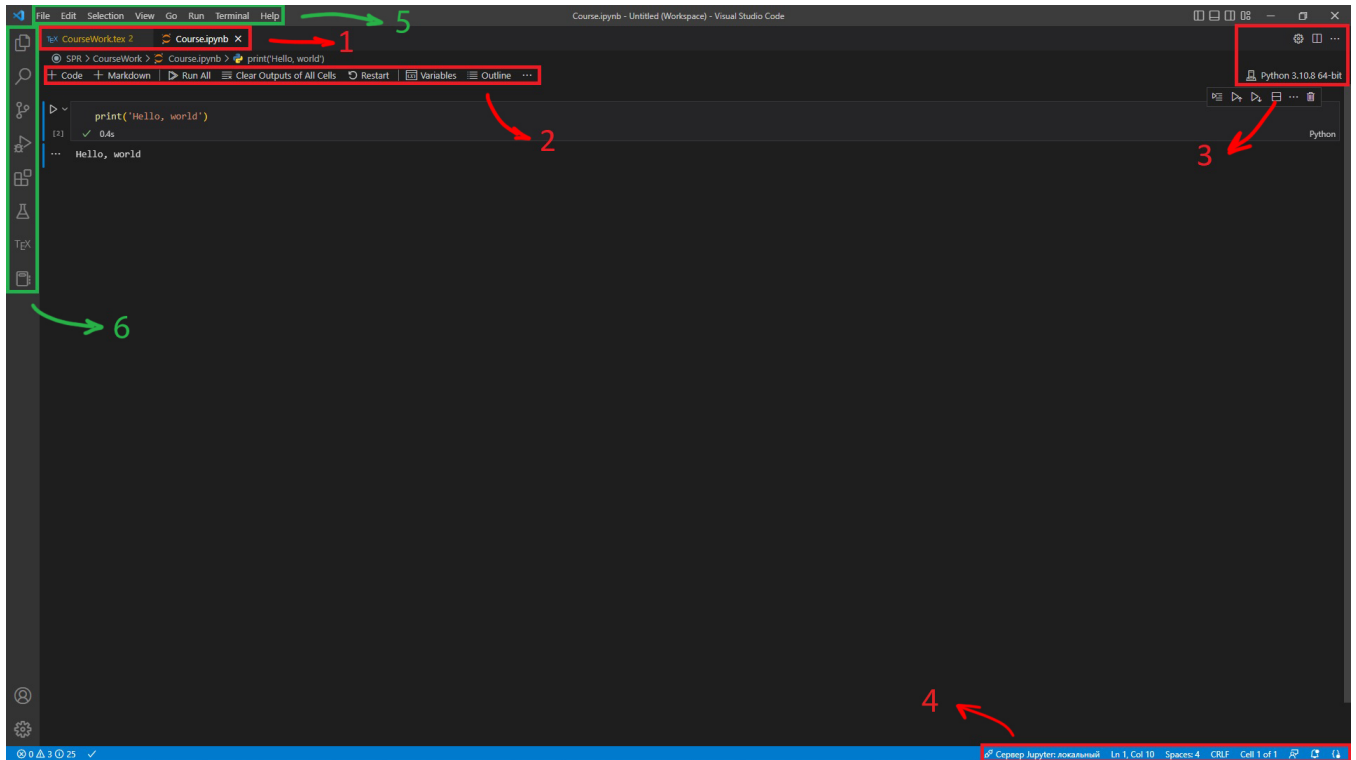


The screenshot shows the Visual Studio Code interface with the `launch.json` file open in the editor. The file is located in the `.vscode` directory. The configuration is for the `Python: Flask` environment. The `launch.json` file contains the following JSON configuration:

```
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 left sidebar shows the `Variables` and `Watch` panels. The top bar indicates the current workspace is `launchjson - Untitled (Workspace) - Visual Studio Code`.

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)

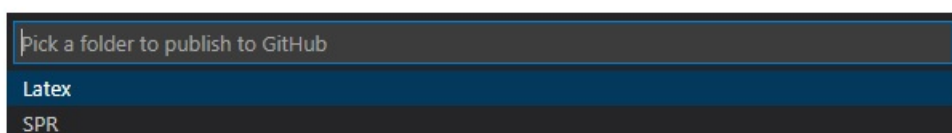
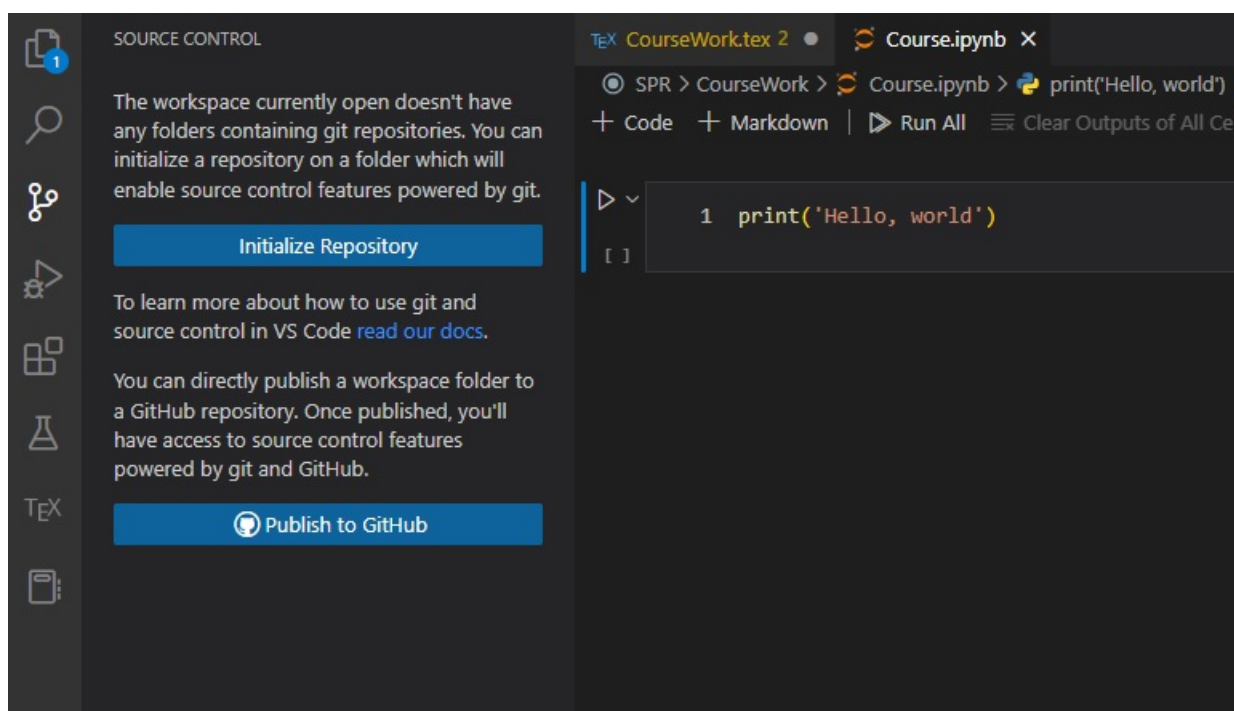
Source control - це клас систем, відповідальних за керування змінами в комп'ютерних програмах, документах, великих веб-сайтах чи інших колекціях інформації.

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

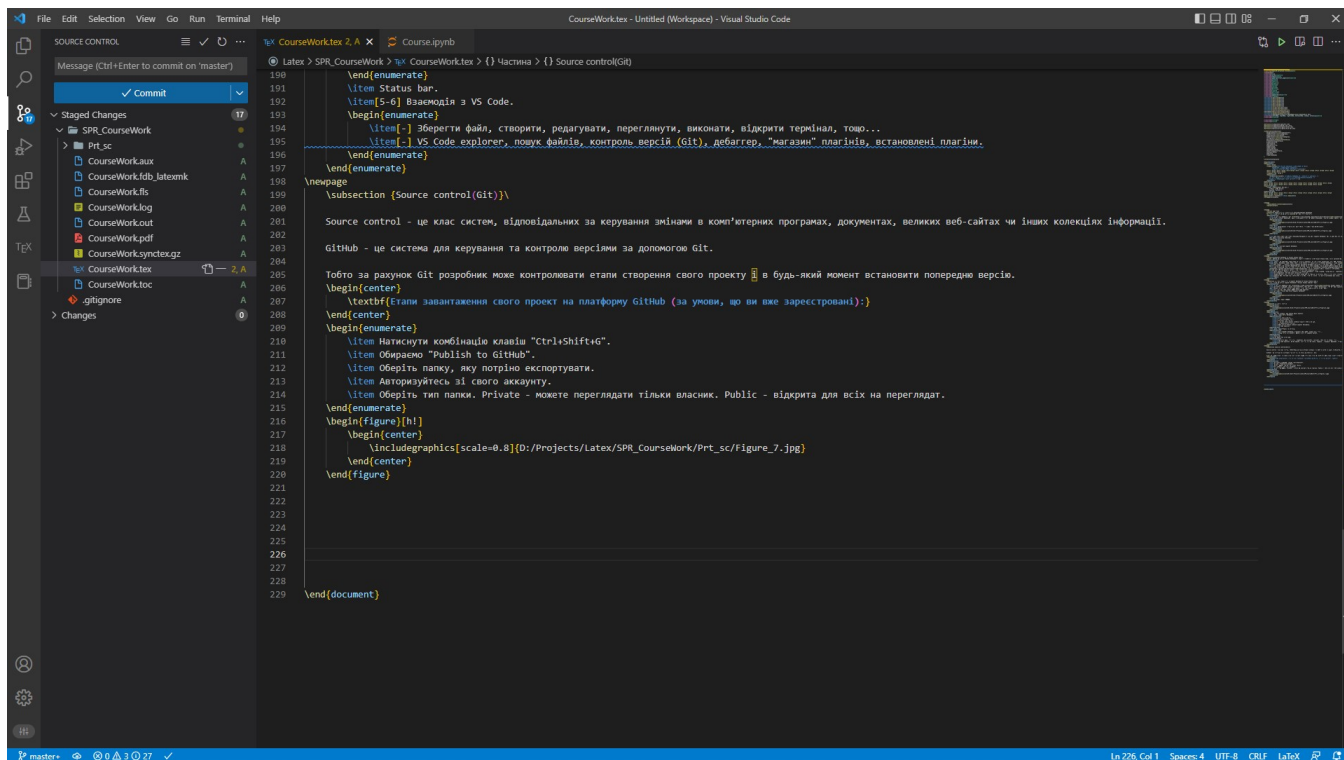
Тобто за рахунок Git розробник може контролювати етапи створення свого проекту і в будь-який момент встановити попередню версію.

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 буду зберігатись на GitHub.



2 Частина



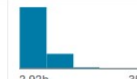






2.1 Знайомство з датасетом.

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

Date	# Close (BTC)	# Volume (BTC)	# Close (ETH)	# Volume (ETH)	# Close (USDT)	# Volume (USDT)	# Close (BNB)	# Volume (BNB)
Date	adjacent close price of Bitcoin Coin on that particular date (in USD)	Volume of BTC on that particular date	adjacent close price of Ethereum Coin on that particular date (in USD)	Volume of ETH on that particular date	adjacent close price of Tether Coin on that particular date (in USD)	Volume of USDT on that particular date	adjacent close price of Binance Coin on that particular date (in USD)	Volume of BNB on that particular date
								
13Nov17	3.24k	2.92b	84.3	622m	0.97	430m	1.51	9284
13Nov22	67.6k	351b	4.81k	84.5b	1.08	279b	676	18.0b
2017-11-13 00:00:00+00:00	6559.490234375	6263249920	316.71608341796875	1041889984	1.0093499422073364	767884032	1.6866199970245361	12238800
2017-11-14 00:00:00+00:00	6635.75	3197110016	337.6310119628906	1069680000	1.0068299770355225	429857984	1.592579960823059	7829600
2017-11-15 00:00:00+00:00	7315.5400390625	4200800128	333.35699462890625	722665984	1.0031800270080566	449671008	1.5304499864578247	7615500
2017-11-16 00:00:00+00:00	7871.60994140625	5123809792	330.92401123046875	797254016	1.002120018005371	650278976	1.5779199600219727	8928640
2017-11-17 00:00:00+00:00	7708.990234375	4651670016	332.3940124511719	621732992	1.001389980316162	639398016	1.5103600025177002	8508840
2017-11-18 00:00:00+00:00	7790.14990234375	3667190016	347.61199951171875	649638976	1.002310037612915	588136000	1.6743299961090088	32167300
2017-11-19 00:00:00+00:00	8036.490234375	3149319936	354.385986328125	1181529984	1.0024399757385254	485752992	1.6137700008071582	38081700
2017-11-20 00:00:00+00:00	8200.6396484375	3488450048	366.7300109863281	807027008	1.0019199848175049	449595008	1.6244499683380127	12994500
2017-11-21 00:00:00+00:00	8071.259765625	4277609984	360.4010009765625	949912000	0.9968470199584961	622649024	1.5740599632263184	5440790
2017-11-22 00:00:00+00:00	8253.5498046875	3633530112	380.6520000566406	800819008	1.0001499652862549	499812000	1.6117500066757202	7177000
2017-11-23 00:00:00+00:00	8038.77001953125	4225179904	410.1659851074219	1845600000	0.9946309924125671	735883008	1.548699975013733	8392640
2017-11-24 00:00:00+00:00	8253.6904296875	5050610176	474.9110107421875	2292829952	1.00232994556427	840627968	1.717710018157959	12155400
2017-11-25 00:00:00+00:00	8790.919921875	4342060032	466.2760009765625	1422080000	1.0054099559783936	651987968	1.8738700151443481	12839900

2.2 Робота у середовищі IPython Notebook на модельних наборах даних.