Лабораторна робота №2
Наука про дані: підготовчий етап

Мета роботи: ознайомитися з основними кроками по роботі з даними – workflow від постановки задачі до написання пояснювальної записки, зрозуміти постановку задачі та природу даних, над якими виконується аналітичні операції
Основні поняття: сирі дані (raw data), підготовка даних (data preparation)

Наука про дані. Вступ

З часом обсяг даних стає все більшим (Big Data), а тому все відчутнішою стає потреба в спеціалістам в області роботи з даними (Data Science Specialization). В межах робіт в цьому напрямку фахівцями виконуються роботи з пошуку даних, їх попередньої підготовки, фільтрування, початкового аналізу, виявлення закономірностей та пояснення отриманих результатів.
Для успішного виконання робіт в цій області потрібно не тільки володіти відповідним математичним апаратом, розуміти семантику області, в якій виконується дослідження, але й володіти ефективними засобами для автоматизації процесу аналізу даних. Таким засобом автоматизації може виступати мова програмування, яка підтримує роботу з численними типами даних, відзначається широким спектром бібліотек для розширення функціоналу та відповідає сучасним вимогам до мов програмування.
Чому ж для вирішення цього кола задач доволі часто використовується мова Python? Вона є досить простою та зрозумілою, не залежить від платформи (існують інтерпретатори під більшість сучасним платформ), є мовою загального призначення (на відміну від Matlab чи R). До того ж ця мова має потужне товариство користувачів та все зростаючу популярність.

Підготовка до роботи з даними
Налаштування робочого середовища

Щодо складу та структури середовища Python досить взяти до уваги наступне:

- Вбудовані класи та функції (`__builtins__`) не потребують підключення відповідних модулів (`import`);
- Вбудовані модулі (близько 40) – додаткові класи та функції (не доступні без `import`);
- Python Standard Library, PSL (понад 200 модулів) – при інсталяції Python за замовчуванням додається набір модулів із класами та функціями (не потребують встановлення), які можна імпортувати в робоче середовище – може сягати;
- Python Package Index, PyPI (понад 42000 станом на 2014) — популярні пакети обробки даних, framework (django) – потребують встановлення.
-

Порада: для виконання лабораторних робіт Ви можете використовувати будь-яке середовище для роботи на Python, однак досить корисним було б ознайомитись з IPython notebook , який встановлюється разом із IDE Spyder в дистрибутиві Anaconda.
Чимало популярних пакетів (`numpy`, `scipy`, `pandas`, `matplotlib`), які часто використовуються для роботи з даними, належать саме до останньої групи (Python Package Index), а тому буде корисним (особливо для тих, хто не зміг присвятити достатньо часу курсовій роботі в минулому семестрі ☺) розглянути процедуру встановлення таких додаткових пакетів, а також самого інтерпретатору.

Для початку розглянемо налаштування робочого середовища на прикладі ОС Windows 7 x64. Потрібно встановити коректну версію інтерпретатора для своєї системи (ті, хто віддає перевагу Linux-системам, можуть отримати консультацію щодо встановлення у викладача, який веде практичні заняття, та у одногрупників ☺).
На даний момент має місце перехідний період від Python 2 до Python 3 (триватиме до 2020 р.), а тому будемо розглядати роботу з Python 3, який вже має достатню кількість розробників. На момент написання методичних вказівок актуальною була версія 3.7.6, MSI-пакет https://www.python.org/downloads/ , а також для новачків рекомендовано використовувати дистриб'ютор анаконда, який можна встановити з Python актуальної версії за адресою https://www.anaconda.com/distribution/
Завантажуємо Windows x86-64 MSI installer та встановлюємо (за замовчуванням - C:\Python37). Після завершення процедури перевіримо коректність встановлення та версію – запустіть Python GUI IDLE (даний shell встановлюється автоматично разом з інтерпретатором та базовими бібліотеками):

```
>>> import sys
>>> print(sys.version)
```

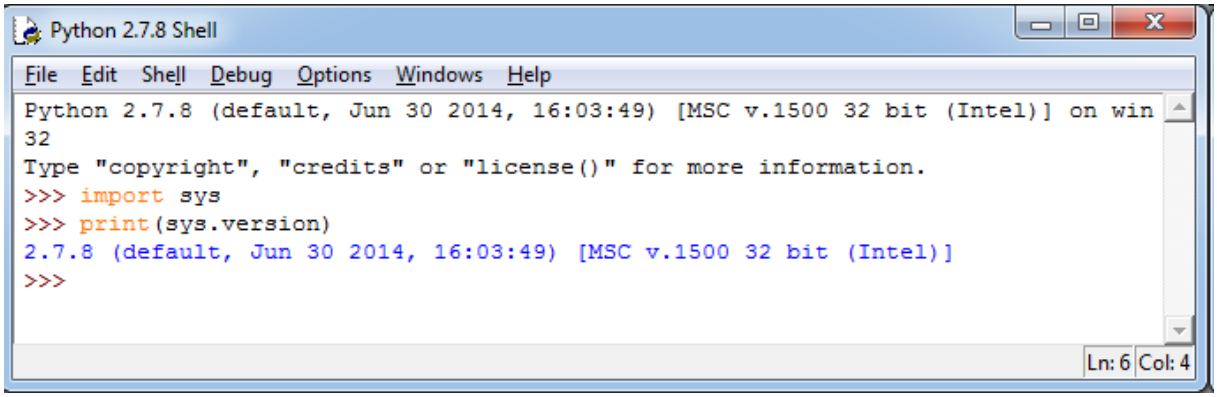

Рис. 1.1 Python GUI IDLE, перевірка версії

Для того, щоб в ОС Windows використовувати скрипти Python, в тому числі для простого встановлення бібліотек з командного рядка, необхідно визначити системні зміні, це можна зробити у наступний спосіб (рис. 1.3):

- Комп'ютер - властивості;
- Додаткові параметри системи;
- В розділі системні змінні знайти змінну PATH та додати до переліку шляхів наступне значення: C:\Python37;C:\Python37\Scripts; (за умови, що встановили Python за замовчуванням);
- Зберігаємо внесені зміни та перезавантажуємо систему.

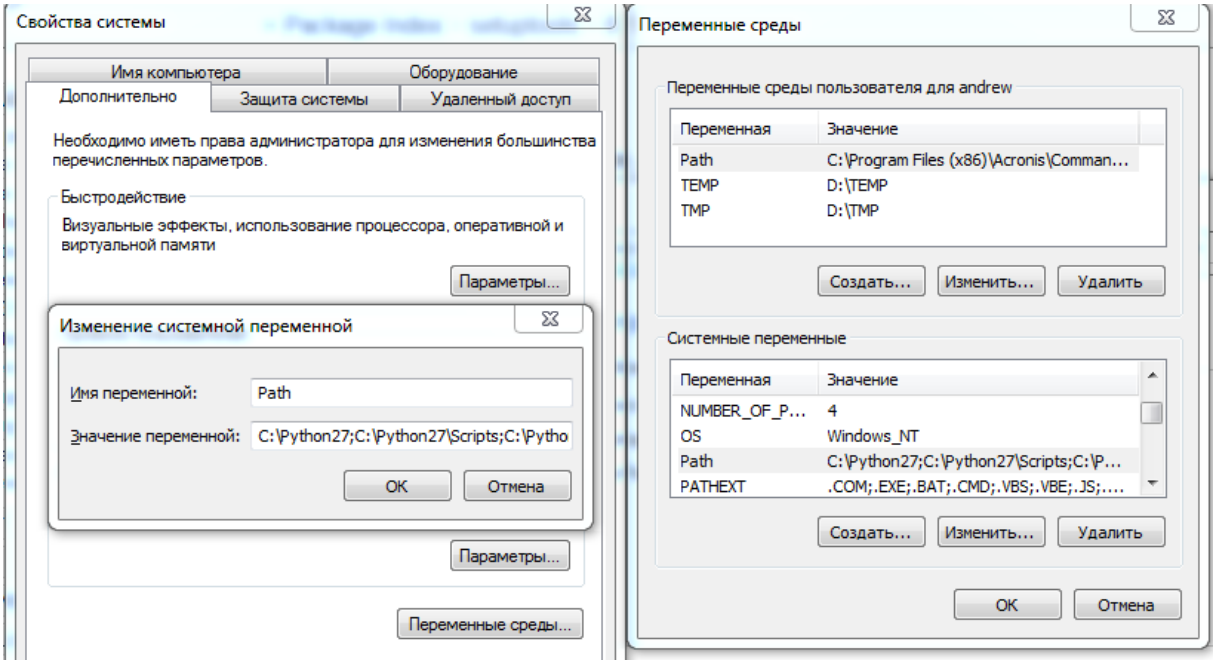

Рис. 1.3 Встановлення значення змінної PATH

Робота з пакетним менеджером PIP (основи)

Для того, щоб отримати довідку по роботі з Python, введіть команду:

```
pip help
```

Для встановлення наявних пакетів:

```
pip install <ім'я_пакету >
```

Для видалення пакету:

```
pip uninstall <ім'я_пакету >
```

Для пошуку пакету:

```
pip search <пошуковий_параметр>
```

Під час встановлення пакетів може знадобитися встановлення додатково програмного забезпечення (компілятор Fortran95, C та C++ тощо) – уважно читайте інструкції та помилки, які видає pip.
Для того, щоб встановити деякі з приведених нижче модулів, потрібно скачати і встановити Microsoft Visual C++ 9.0 :
http://aka.ms/vcpython27

Встановимо модулі matplotlib, numpy, pandas, ipython:

```
pip install numpy pandas matplotlib ipython
```

Також варто звикати до того, що при потребі Ви будете встановлювати додаткові модулі для виконання тих чи інших специфічних задач.

Робота з Anaconda

Із офіційного сайту завантажте файл для встановлення Anaconda з необхідною версією python. При встановлені на Windows в почергових стандартних виборах опцій виберіть встановлення для одного юзера, що є рекомендованим, а також додавання до PATH.
Використовуючи дистриб'ютор Anaconda можливо в командному рядку проводити схожі операції з бібліотеками, як і з PIP, проте вагомою перевагою є можливість створення віртуальних середовищ (virtual environment) який далі будемо називати env. Всередині кожного активного env можливо встановити свій список бібліотек, не переживаючи за функціональність інших env. Таким чином якщо при встановленні бібліотек ви зіпсували зв'язки і нічого більше не працює, можна створити новий env і

почати заново його псувати. В таблиці наведено порівняння основних команд командного рядку/терміналу conda та pip. Для написання програмного коду в jupyter notebook необхідно в командному рядку ввести jupyter notebook і створити ipython notebook в інтерфейсі браузера.

Task	Conda package and environment manager command	Pip package manager command
Install a package	conda install $PACKAGE_NAME	pip install $PACKAGE_NAME
Update a package	conda update --name $ENVIRONMENT_NAME $PACKAGE_NAME	pip install --upgrade $PACKAGE_NAME
Update package manager	conda update conda	Linux/macOS: pip install -U pip Win: python -m pip install -U pip
Uninstall a package	conda remove --name $ENVIRONMENT_NAME $PACKAGE_NAME	pip uninstall $PACKAGE_NAME
Create an environment	conda create --name $ENVIRONMENT_NAME python	X
Activate an environment	conda activate $ENVIRONMENT_NAM	X
Deactivate an environment	conda deactivate	X
Search available packages	conda search $SEARCH_TERM	pip search $SEARCH_TERM
Install package from specific source	conda install --channel $URL $PACKAGE_NAME	pip install --index-url $URL $PACKAGE_NAME

List installed packages	conda list --name $ENVIRONMENT_NAME	pip list
Create requirements file	conda list --export	pip freeze
List all environments	conda info --envs	X
Install other package manager	conda install pip	pip install conda
Install Python	conda install python=x.x	X
Update Python	conda update python	X

Постановка задачі

- Проаналізувати часові ряди глобальних продуктів по оцінці вегетаційного здоров'я VHI (vegetation health index) http://www.star.nesdis.noaa.gov/smcd/emb/vci/VH/index.php, який надається Національною адміністрацією океанів та атмосфери США NOAA (http://www.noaa.gov/);
- Виявити особливості ходу індексу впродовж вегетаційного періоду (вересень року-попередника – липень поточного року) в розрізі областей України;
- Додаткові завдання по фільтрації даних від викладача, який веде практику.

Семантика задачі

VHI – вегетаційний індекс, який базується на відбитті видимого світла рослинним покривом, яке характеризує ступінь здоров'я рослинності. Цей індекс базується на поєднанні індексу VCI (Vegetation Condition Index), який описує ступінь пригніченості рослинного покриву та індексі

температурного режиму TCI (Temperature Condition Index), які були запропоновані в 1995 році Ф. Коганом:

$$VHI = 0.5 * VCI + 0.5 * TCI, \quad (1.1)$$

$$VCI = 100 * \frac{(NDVI - NDVI_{\min})}{(NDVI_{\max} - NDVI_{\min})}, \quad (1.2)$$

$$TCI = 100 * (BT_{\max} - BT) / (BT_{\max} - BT_{\min}), \quad (1.3)$$

де BT, $BT_{\max}$, $BT_{\min}$ - усереднені сезонні значення яскравосної температури, її абсолютний максимум та мінімум відповідно, $NDVI$, $NDVI_{\max}$, $NDVI_{\min}$ — усереднені значення нормалізованого різницевого вегетаційного індексу NDVI.

Зміст VHI-індексу

- VHI <40 – стресові умови;
- VHI > 60 – сприятливі умови;
- VHI < 15 – посуха, інтенсивність якої від середньої до надзвичайної;
- VHI < 35 – посуха, інтенсивність якої від помірної до надзвичайної.

Для розуміння фізичної природи даних, з якими Ви будете працювати, варто ознайомитися з роботами.
Для того, щоб перейти безпосередньо до роботи з даними на сторінці http://www.star.nesdis.noaa.gov/smcd/emb/vci/VH/index.php перейдіть за посиланням VH Info By Province, на якій зі списку країн оберіть Україну. В результаті виконання цих дій відобразиться інтерактивна карта України (рис. 1.4):

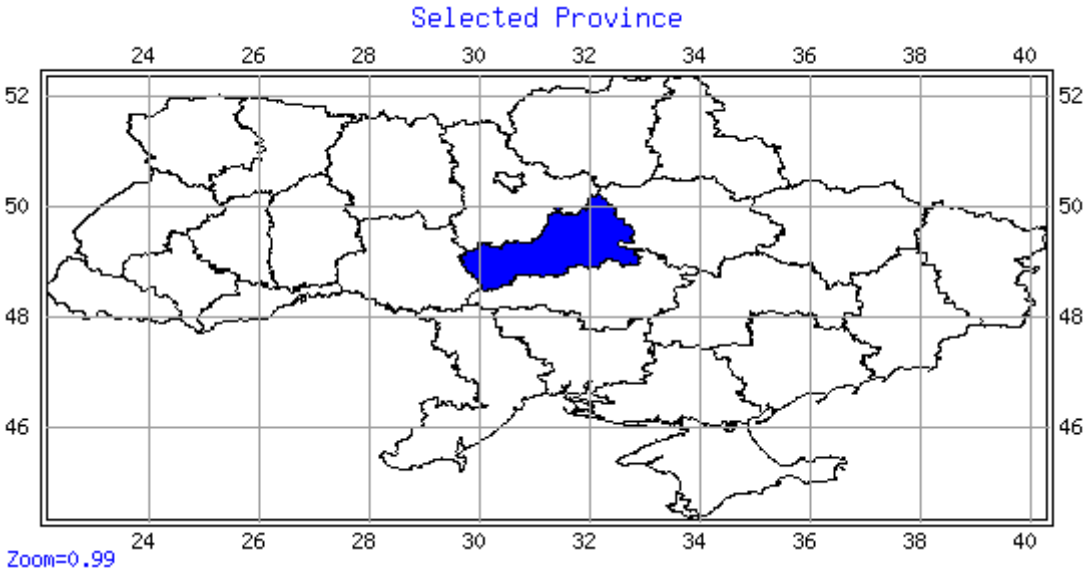

Рис. 1.4 Інтерактивна карта України на порталі NOAA

Нижче інтерактивної карти є посилання time series data, за яким можна завантажити дані з ходом вегетаційного індексу для виділеної області.

Зразки коду

Скачування файлу за посиланням:

```
import  urllib2
url="http://www.star.nesdis.noaa.gov/smcd/emb/vci/gvix/G04/ts_L1/ByProvince/Mean/L1_Mean_UKR.R16.txt"
vhi_url = urllib2.urlopen(url)
out = open('vhi_id_16.csv','wb')
out.write(vhi_url.read())
out.close()
print "VHI is downloaded..."
```

Зчитати csv-файлу у фрейм, вивести імена стовпців та перший рядок:

```
import pandas as pd
df = pd.read_csv('vhi_id_16.csv',index_col=False, header=1)
print list(df.columns.values)
print df[:1]
```

Вибрати із фрейму записи за умовою (запис із таблиці для поточної області, за 2000-й рік та 18-й тиждень):

```
df[(df['year']==2000) & (df['week']==18)]
```

Хід виконання роботи

- Створити env в якому будуть встановлені всі необхідні бібліотеки та налаштування для данної лабораторної роботи
- Для кожної із адміністративних одиниць України завантажити тестові структуровані файли, що містять значення VHI-індексу. Ця процедура має бути автоматизована, параметром процедури має бути індекс (номер) області. При зберіганні файлу до його імені потрібно додати дату та час завантаження;
- Зчитати завантажені текстові файли у фрейм (детальніше про роботу із фреймами буде розказано у подальших лабораторних роботах). Імена стовбців фрейму мають бути змістовними та легкими для сприйняття (не повинно бути спеціалізованих символів, пробілів тощо). Ця задача має бути реалізована у вигляді окремої процедури, яка на вхід приймає шлях до директорії, в якій зберігаються файли;
- Реалізувати процедуру, яка змінить індекси областей, які використані на порталі NOAA на наступні (виключно старі індекси на нові):

№ області	Назва	№ області	Назва
1	Вінницька	13	Миколаївська
2	Волинська	14	Одеська
3	Дніпропетровська	15	Полтавська
4	Донецька	16	Рівенська
5	Житомирська	17	Сумська
6	Закарпатська	18	Тернопільська
7	Запорізька	19	Харківська
8	Івано-Франківська	20	Херсонська
9	Київська	21	Хмельницька
10	Кіровоградська	22	Черкаська
11	Луганська	23	Чернівецька
12	Львівська	24	Чернігівська
		25	Республіка Крим

- Реалізувати процедури для формування вибірок наступного виду (включаючи елементи аналізу):
    - Ряд VHI для області за вказаний рік;
    - Пошук екстремумів (min та max) для вказаних областей та років;
    - Ряд VHI за вказаний діапазон років для вказаних областей;
    - Для всього набору даних виявити роки, протягом яких екстремальні посухи торкнулися більше вказаного відсотка областей по Україні (20% областей - 5 областей з 25). Повернути роки, назви областей з екстремальними посухами та значення VHI;
    - Аналогічно для помірних посух

Примітка: Захист лабораторної роботи буде супроводжуватися додатковими завданнями по формуванням довільних вибірок та виконанням простого аналізу часового ряду (min, max, average, пошук підрахунок кількості спостережень, які відповідають тим чи іншим вимогам тощо). Для успішної здачі лабораторної умови необхідно чітко орієнтуватися в даних, з якими Ви працюєте.

Контрольні запитання

1. Основні кроки по вирішенню задачі аналізу даних
2. В чому переваги Python над іншими мовами програмування у сфері вирішення задач Data Science?
3. Дати визначення поняттю «ідеальний набір даних»
4. Які, на Вашу думку, існують джерела даних?
5. Які дані можна вважати «чистими» (clean data)?

Література

1. http://docs.python.org/tutorial/
2. http://www.greenteapress.com/thinkpython/
3. http://www.diveintopython.net/
4. Kevin Sheppard. Introduction to Python for econometrics, statistics and data analysis. Self-published, University of Oxford, version 2.1 edition, February 2014
5. Finn Arup Nielsen. Data Mining with Python. Draft, December, 2014