

Python for Data Mining

<https://github.com/axtolm/pyDataMining>

Александр Владимирович Толмачев

axtolm@gmail.com

1. Основы языка Python

1.1. Вводная информация. Python и Data Mining.

В этой лекции:

- Для кого и о чем этот курс? Какова цель?
- Как работать с материалами курса?
- Что такое Data Mining и какие задачи можно решать с его помощью?
- Зачем специалистам по Data Mining язык Python?
- Чем хорош пакет Anaconda и как его установить на свой компьютер?
- Какие приложения есть в пакете Anaconda и как их запускать?



Для кого и о чем этот курс? Какова его цель?

Прежде всего – **курс для аналитиков**, а не для программистов.

Его основная цель – показать, как можно быстро начать использовать язык Python для решения прикладных задач в области анализа данных.

Курс может быть полезен как тем, кто уже знаком с языком Python, так и тем, кто сталкивается с ним впервые. Для новичков предусмотрены «Основы языка Python». Специалисты смогут почерпнуть для себя что-то новое из разбираемых прикладных задач, характерных при анализе данных:

- Сбор и извлечение данных из разных источников.
- Предобработка данных.
- Собственно анализ данных с помощью статистических методов и нейронных сетей.
- Визуализация результатов анализа данных.

Как работать с материалами курса?

При освоении курса важна самостоятельная работа.

Оптимальный путь:

- Смотреть видео.
- Скачивать с GitHub код из видео и проходить его самостоятельно у себя на компьютере:
 - Запускать код, который не был детально разобран на видео.
 - Менять стартовые условия и запускать код с ними.
 - Анализировать результат работы кода и оценивать себя.

Что делать, если остались вопросы и непонимание?

- Пересмотреть видео.
- Поработать с документацией (ссылки на нее даются в видео и есть в материалах).
- Найти ответ в интернете, задав конкретный вопрос в поиске.
Есть много ресурсов типа stackoverflow, где такой вопрос уже мог быть задан и на него есть ответ.
В противном случае можно задать его.

Определения. Интеллектуальный анализ данных (Data Mining)

1) Data Mining — это процесс обнаружения в «сырых» данных знаний, необходимых для принятия решений в различных сферах человеческой деятельности (Григорий Пятецкий-Шапиро, 1992 г.).

При этом знания должны быть ранее неизвестными, нетривиальными, практически полезными и доступными интерпретации.

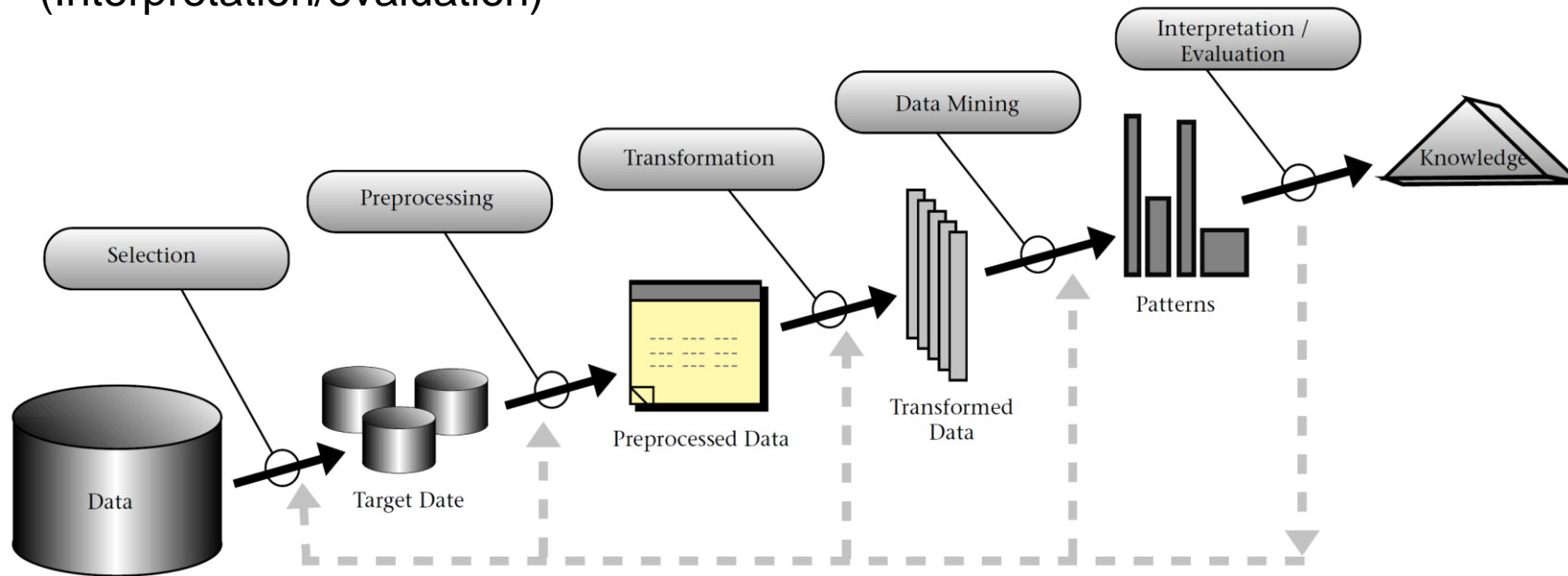
2) Data Mining — это современная концепция анализа данных, предполагающая, что:

- данные могут быть неточными, неполными, противоречивыми, разнородными, и при этом иметь гигантские объёмы;
- алгоритмы анализа данных могут обладать «элементами интеллекта», в частности, способностью обучаться по прецедентам, а их разработка также требует значительных интеллектуальных усилий;
- процессы переработки сырых данных в информацию, а информации в знания не могут быть выполнены вручную и требуют нетривиальной автоматизации.

Data Mining - часть более общего процесса **извлечения знаний из баз данных** («Knowledge Discovery in Databases" или KDD)¹⁾.

Этапы KDD:

1. Отбор данных (Selection)
2. Предварительная обработка данных (Pre-processing)
3. Преобразование данных (Transformation)
4. Интеллектуальный анализ данных (Data Mining)
5. Интерпретация и оценка результатов (Interpretation/evaluation)



Python – полноценный инструмент KDD, используемый на всех этапах.

Почему Python?

- прост в освоении,
- широко распространен,
- много библиотек для решения прикладных задач,
- open source проект,
- кроссплатформенный, работает с CPU и GPU,
- активно развивается,
- есть своя философия.

¹⁾ Fayyad U., Piatetsky-Shapiro G., & Smyth P. From data mining to knowledge discovery in databases. Ai Magazine, vol. 17, no. 3, pp. 37-54, 1996

Подробнее о Data Mining

Шесть классов задач, которые решают с помощью Data Mining:

1. Классификация (Classification)
2. Регрессия (Regression)
3. Кластеризация (Clustering)
4. Обобщение (Summarization)
5. Моделирование зависимостей (Dependency modelling)
6. Обнаружение аномалий (Change and deviation detection)

**Вручную их не выполнить, а для автоматизации нужны инструменты.
Мы будем использовать Python.**

Для Data Mining язык Python хорош как Low Code инструмент

(идеи Low Code/No Code сейчас популярны)

Примеры. Как создать нейронную сеть на Python для задачи классификации объектов

1. Многослойный персептрон

```
from sklearn.neural_network import MLPClassifier    # импорт класса
mlp = MLPClassifier(hidden_layer_sizes=(50,50), max_iter=10000, random_state = 11)    # создаем объект многослойный персептрон
mlp.fit(X_train, Y_train.values.ravel())    # обучаем его
Y_pred_test = mlp.predict(X_test)    # делаем прогноз с помощью обученного персептрона
```

2. Самоорганизующиеся карты Кохонена

```
from sklearn_som.som import SOM    # импорт class SOM
iris_som = SOM(m = 3, n = 1, dim = 2, max_iter = 10000, random_state = 11)    # создаем объект на базе класса SOM
iris_som.fit(train_data)    # обучаем модель
test_predictions = iris_som.predict(test_data)    # делаем прогноз с помощью обученной сети
```

При сборе и предобработке данных такой лаконичности получить не удастся.

Работать с Python будем с помощью пакета Anaconda Individual Edition для Windows

- Популярность у 25M+ пользователей в мире.
- Установка на Linux, Windows, Mac OS.
- Основные библиотеки для работы с данными (250+) идут в комплекте поставки.
- 7.5K+ библиотек доступны в облаке.
- Open-source для Data Mining и Machine Learning.



Установка Anaconda Individual Edition:

Скачать установщик <https://www.anaconda.com/products/individual> и запустить его.

Подробности процесса установки: <https://docs.anaconda.com/anaconda/install/>

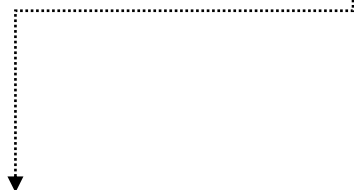
Anaconda Individual Edition - когда и что из компонентов будем использовать в работе



IDE Spyder

Интегрированная среда разработки

- код, требующий отладки
- относительно объемный код
- код с длительным временем исполнения



Для отладки есть все необходимое:

- ✓ менеджер переменных
- ✓ точки останова и пошаговое выполнение
- ✓ профайлер



Jupyter Notebook

Web приложение для работы с документами (текст + код на Python + результаты его выполнения)

- алгоритмы с пошаговым выводом результатов
- код и результаты, которыми нужно поделиться
- отчеты для программистов и непрограммистов



Conda

Менеджер для управления библиотеками и окружением

- установка библиотек
- создание виртуальных окружений



IDE Spyder – интегрированная среда разработки в составе Anaconda Distribution

<https://www.spyder-ide.org/>

кнопки запуска
и отладки кода

точка останова
для отладки

редактор кода

менеджер
переменных

КОНСОЛЬ

The screenshot displays the Spyder IDE interface with the following components:

- Code Editor:** Contains a Python script for data analysis using pandas. A red dot on line 17 indicates a breakpoint.
- Variable Explorer:** A table showing the state of variables in the current environment.
- Console:** Displays the output of the code execution, including the result of a query and the dimensions of a DataFrame.

Name	Type	Size	Value
DATA_PATH	str	62	D:/YandexDisk/5_python/_python_learning/_py_bachelors_09_2021/
ddf	DataFrame	(8, 3)	Column names: ID_student, AVG, RAVG
del_list	list	5	[1, 2, 4, 5, 8]
FILE_EXT	str	4	.csv
FILE_LA_F2017	str	41	lesson5_data_ENGM_F2017_train_atolm092021
LA_F2017	DataFrame	(831, 4)	Column names: ID_student, AVG, RAVG, gender
LA_F2017_Copy1	DataFrame	(826, 4)	Column names: ID_student, AVG, RAVG, gender
LA_F2017_Copy2	DataFrame	(831, 4)	Column names: ID_student, AVG, RAVG, gender
na_df	DataFrame	(3, 4)	Column names: ID_student, AVG, RAVG, gender
na_list	list	3	[1, 4, 5]
outlier_df	DataFrame	(3, 4)	Column names: ID_student, AVG, RAVG, gender
outlier_list	list	3	[1, 2, 8]

```
...: outlier_list = LA_F2017.query('AVG>1 or RAVG>1').index.tolist()
...:
...: LA_F2017_Copy1 = LA_F2017.copy() # Сделаем копию, чтобы удалить в ней
...: # получим в виде списка индексы строк, которые надо удалить - ПРОПУСКИ
...: del_list = list(set(na_list+outlier_list)) # уберем подтопы
...:
...: LA_F2017_Copy1 = LA_F2017_Copy1.drop(del_list) # удаляем строки с индексами из списка
...: ddf = LA_F2017_Copy1.describe()
...:
...: LA_F2017_Copy2 = LA_F2017.copy()
...: LA_F2017_Copy2 == LA_F2017
```

Out[1]:

ID_student	AVG	RAVG	gender
0	True	True	True
1	True	False	True
2	True	True	True
3	True	True	True
4	True	True	False
...
826	True	True	True
827	True	True	True
828	True	True	True
829	True	True	True
830	True	True	True

[831 rows x 4 columns]

In [2]:



<https://jupyter.org/>

The Jupyter Notebook – это open-source **web приложение**, которое позволяет создавать **документы**, содержащие **тексты и рисунки**, **код на Python** и **результаты его выполнения**; конвертировать документы в HTML и другие форматы.

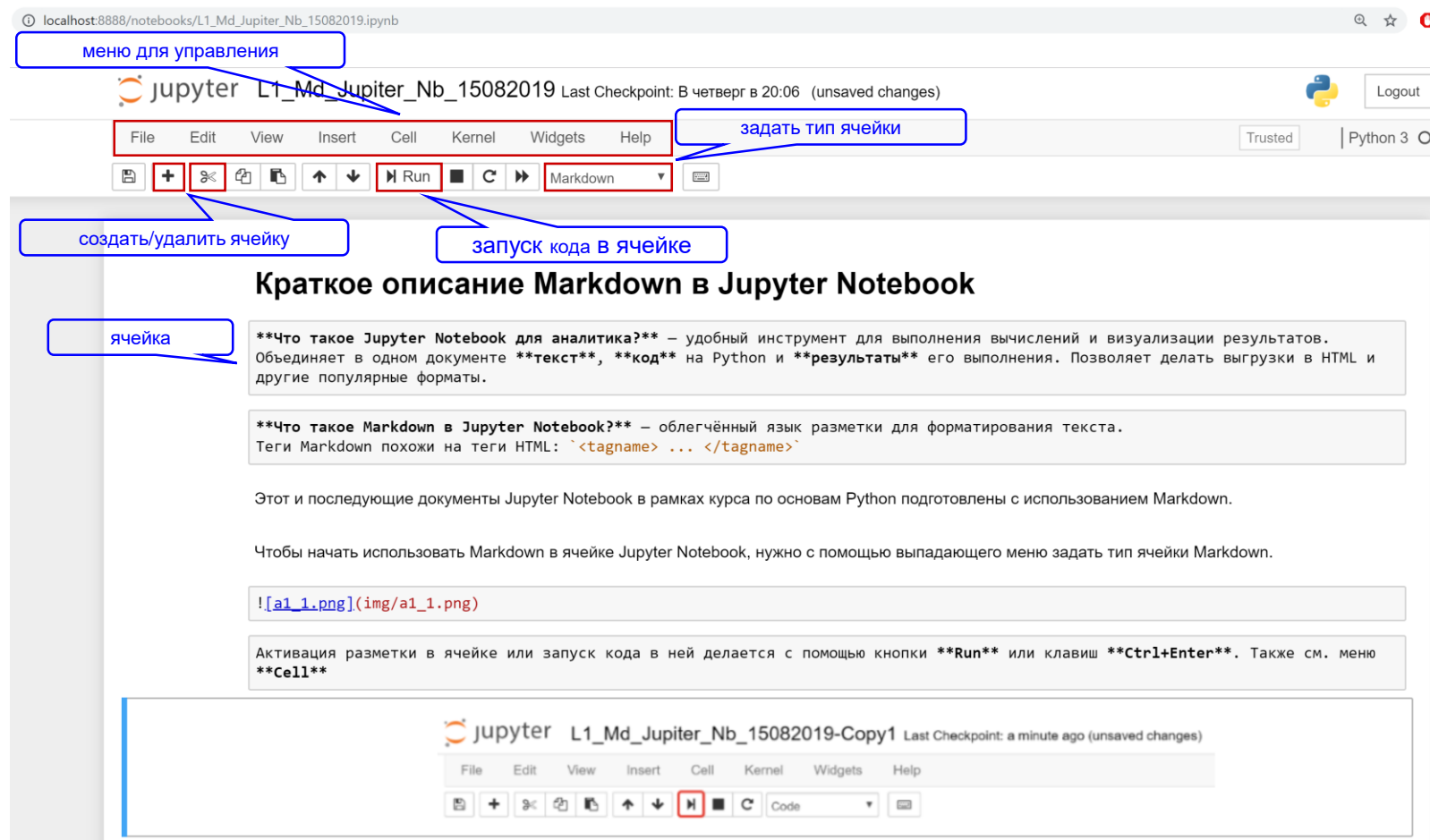
Поставляется в составе
Anaconda Distribution

Ячейка – ключевой элемент

В ячейке может быть:

- **текст**, который можно форматировать с помощью Markdown
- **код на Python** и результаты его выполнения
- **рисунок**

Есть еще **JupyterLab** – Web IDE для Jupyter notebooks





Conda – это open source менеджер для управления библиотеками и окружением, который работает на Windows, macOS, Linux.

Поставляется в составе
Anaconda Distribution

Используем для установки
библиотек с командной
строки в терминале:

conda install PKGNAME

PKGNAME – имя
устанавливаемого пакета

```
C:\Windows\system32\cmd.exe

(base) C:\Users\atolm>conda info

      active environment : base
      active env location : C:\Anaconda3
            shell level : 1
      user config file : C:\Users\atolm\.condarc
 populated config files : C:\Users\atolm\.condarc
         conda version : 4.7.10
    conda-build version : 3.18.8
         python version : 3.7.3.final.0
    virtual packages : __cuda=10.0
      base environment : C:\Anaconda3 (writable)
        channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                      https://repo.anaconda.com/pkgs/main/noarch
                      https://repo.anaconda.com/pkgs/r/win-64
                      https://repo.anaconda.com/pkgs/r/noarch
                      https://repo.anaconda.com/pkgs/msys2/win-64
                      https://repo.anaconda.com/pkgs/msys2/noarch
         package cache : C:\Anaconda3\pkgs
                        C:\Users\atolm\.conda\pkgs
                        C:\Users\atolm\AppData\Local\conda\conda\pkgs
      envs directories : C:\Anaconda3\envs
                        C:\Users\atolm\.conda\envs
                        C:\Users\atolm\AppData\Local\conda\conda\envs
         platform : win-64
        user-agent : conda/4.7.10 requests/2.22.0 CPython/3.7.3 Windows/10 Windows/10.0.18362
       administrator : False
           netrc file : None
       offline mode : False
```

Запуск приложений Spyder, Jupyter Notebook, Conda из пакета Anaconda

Запуск Spyder

Windows menu Start

- Anaconda3 (64-bit)
- **Spyder** (anaconda3)

Запускается как самостоятельное приложение в своем окне

Запуск Conda

Windows menu Start

- Anaconda3 (64-bit)
- **Anaconda Powershell Prompt (anaconda3)**

Запускается терминал с командной строкой

Запуск Jupyter Notebook

Windows menu Start

- Anaconda3 (64-bit)
- **Jupyter Notebook** (anaconda3)

Запускается в окне браузера

1. Основы языка Python

1.1. Вводная информация. Python и Data Mining.

Подведем итоги. На этой лекции:

- Вы услышали вводную информацию по курсу.
- Узнали, что такое Data Mining и какие задачи можно решать с его помощью.
- Выяснили, какое место занимает Data Mining в более общем процессе извлечения знаний.
- Обсудили, чем при анализе данных может быть полезен язык Python.
- Установили пакет Anaconda на свой компьютер.
- Научились запускать приложения Spyder, Jupyter Notebook, Conda