Datahantering & Numpy

RESTORY...

Agenda

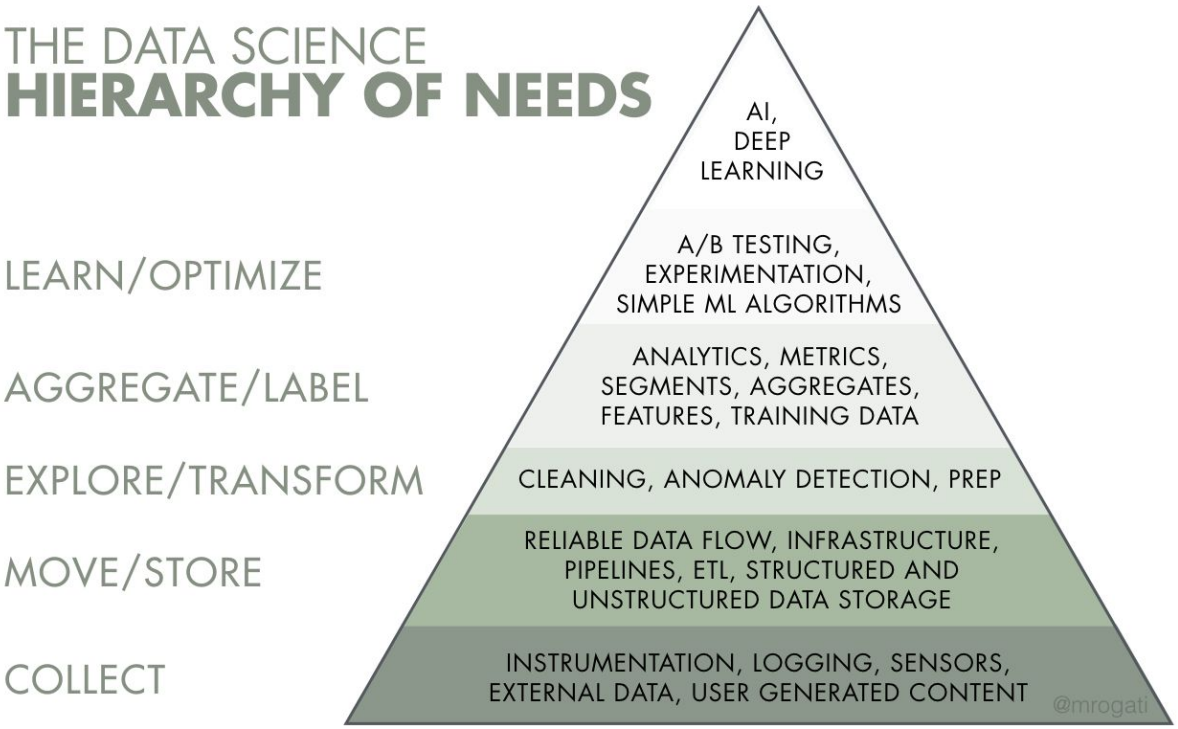
- Data
  - Hur ser data ut?
- hantering (processing)
  - (cleaning)
  - normalisering
  - datatypes
  - transforming
  - selecting
- numpy
- regex
- Miljöer

Tips! I en Jupyter cell skriv `%quickref` *för att få en lista med magiska funktioner.*

Data Science

Data scientist - samla in, processa, analysera, presentera resultat

Data analyst- ingen sk. *predictive modelling* utan tittar på det historiska datat i ett affärssammanhang. Hur kan de gå vidare?



Data Processing / Cleaning / Wrangling

Data är sällan perfekt och har ofta värden som saknas, fel format, tomma kolumner eller rader. Då resultaten är väldigt beroende av vad vi “stoppar in”, är det väldigt viktigt att ta bort dessa fel.

Data wrangling - rengöring och manipulering (= transform, merge, group-by, reshape)

Typer av data:

- **Kvalitativa** - kan observeras subjektivt, inte mätas. Beskrivande
  - **Nominal** - går ej att rangordna
    ex. Man / Kvinna / Annat
  - **Ordinal** - kan rangordnas
    ex. bra, medel, dåligt
- **Kvantitativa** - fakta, kan mätas
  - **Heltal** -Discrete/Categorical:
    ex. 0,1,2 eller # av studenter
  - **Reela tal** - Continuous, skala
    ex. 0,0.001, 0.02.. eller vikt

Data från webben - format

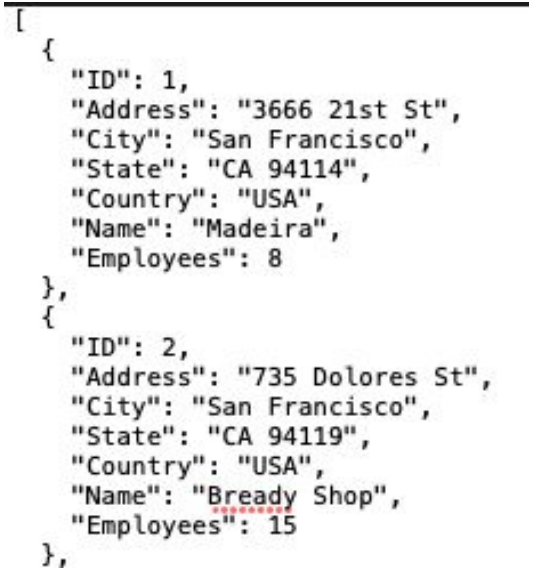
csv i excel och txt

ID	Address	City	State	Country	Name	Employees
1	3666 21st St	San Francisco	CA 94114	USA	Madeira	8
2	735 Dolores St	San Francisco	CA 94119	USA	Bready Shop	15
3	332 Hill St	San Francisco	California 94114	USA	Super River	25
4	3995 23rd St	San Francisco	CA 94114	USA	Ben's Shop	10
5	1056 Sanchez St	San Francisco	California	USA	Sanchez	12
6	551 Alvarado St	San Francisco	CA 94114	USA	Richvalley	20

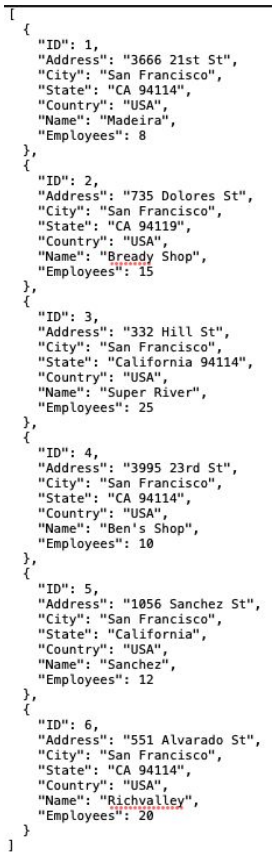
```
ID,Address,City,State,Country,Name,Employees
1,3666 21st St,San Francisco,CA 94114,USA, Madeira,8
2,735 Dolores St,San Francisco,CA 94119,USA,Bready Shop,15
3,332 Hill St,San Francisco,California 94114,USA,Super River,25
4,3995 23rd St,San Francisco,CA 94114,USA,Ben's Shop,10
5,1056 Sanchez St,San Francisco,California,USA,Sanchez,12
6,551 Alvarado St,San Francisco,CA 94114,USA,Richvalley,20
```

Data från webben- format

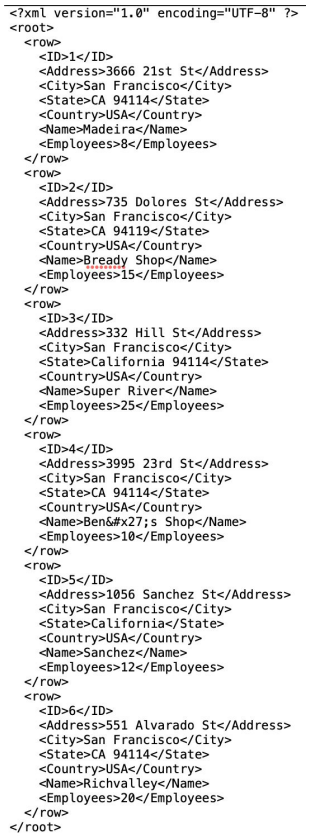
json



```
[
  {
    "ID": 1,
    "Address": "3666 21st St",
    "City": "San Francisco",
    "State": "CA 94114",
    "Country": "USA",
    "Name": "Madeira",
    "Employees": 8
  },
  {
    "ID": 2,
    "Address": "735 Dolores St",
    "City": "San Francisco",
    "State": "CA 94119",
    "Country": "USA",
    "Name": "Bready Shop",
    "Employees": 15
  },
```

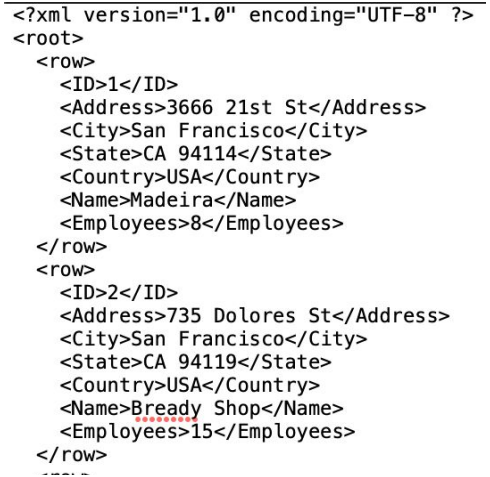


```
[
  {
    "ID": 1,
    "Address": "3666 21st St",
    "City": "San Francisco",
    "State": "CA 94114",
    "Country": "USA",
    "Name": "Madeira",
    "Employees": 8
  },
  {
    "ID": 2,
    "Address": "735 Dolores St",
    "City": "San Francisco",
    "State": "CA 94119",
    "Country": "USA",
    "Name": "Bready Shop",
    "Employees": 15
  },
  {
    "ID": 3,
    "Address": "332 Hill St",
    "City": "San Francisco",
    "State": "California 94114",
    "Country": "USA",
    "Name": "Super River",
    "Employees": 25
  },
  {
    "ID": 4,
    "Address": "3995 23rd St",
    "City": "San Francisco",
    "State": "CA 94114",
    "Country": "USA",
    "Name": "Ben's Shop",
    "Employees": 10
  },
  {
    "ID": 5,
    "Address": "1056 Sanchez St",
    "City": "San Francisco",
    "State": "California",
    "Country": "USA",
    "Name": "Sanchez",
    "Employees": 12
  },
  {
    "ID": 6,
    "Address": "551 Alvarado St",
    "City": "San Francisco",
    "State": "CA 94114",
    "Country": "USA",
    "Name": "Richvalley",
    "Employees": 20
  }
]
```



```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <row>
    <ID>1</ID>
    <Address>3666 21st St</Address>
    <City>San Francisco</City>
    <State>CA 94114</State>
    <Country>USA</Country>
    <Name>Madeira</Name>
    <Employees>8</Employees>
  </row>
  <row>
    <ID>2</ID>
    <Address>735 Dolores St</Address>
    <City>San Francisco</City>
    <State>CA 94119</State>
    <Country>USA</Country>
    <Name>Bready Shop</Name>
    <Employees>15</Employees>
  </row>
  <row>
    <ID>3</ID>
    <Address>332 Hill St</Address>
    <City>San Francisco</City>
    <State>California 94114</State>
    <Country>USA</Country>
    <Name>Super River</Name>
    <Employees>25</Employees>
  </row>
  <row>
    <ID>4</ID>
    <Address>3995 23rd St</Address>
    <City>San Francisco</City>
    <State>CA 94114</State>
    <Country>USA</Country>
    <Name>Ben&#x27;s Shop</Name>
    <Employees>10</Employees>
  </row>
  <row>
    <ID>5</ID>
    <Address>1056 Sanchez St</Address>
    <City>San Francisco</City>
    <State>California</State>
    <Country>USA</Country>
    <Name>Sanchez</Name>
    <Employees>12</Employees>
  </row>
  <row>
    <ID>6</ID>
    <Address>551 Alvarado St</Address>
    <City>San Francisco</City>
    <State>CA 94114</State>
    <Country>USA</Country>
    <Name>Richvalley</Name>
    <Employees>20</Employees>
  </row>
</root>
```

XML



```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <row>
    <ID>1</ID>
    <Address>3666 21st St</Address>
    <City>San Francisco</City>
    <State>CA 94114</State>
    <Country>USA</Country>
    <Name>Madeira</Name>
    <Employees>8</Employees>
  </row>
  <row>
    <ID>2</ID>
    <Address>735 Dolores St</Address>
    <City>San Francisco</City>
    <State>CA 94119</State>
    <Country>USA</Country>
    <Name>Bready Shop</Name>
    <Employees>15</Employees>
  </row>
```

RESTORY...

numpy

numpy

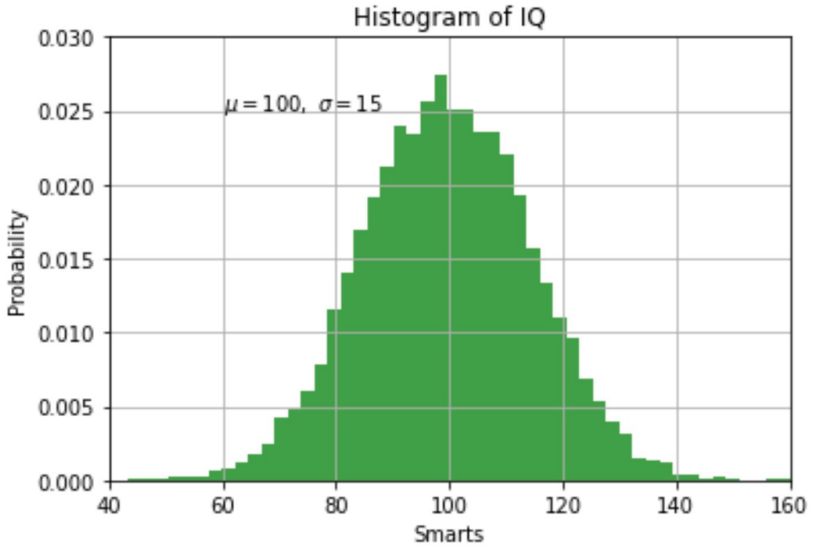
grunden till pandas

matematiska operationer

scientific calculations

matplotlib

ML (tensors i tensorflow)



Övningar: https://numpy.org/numpy-tutorials/content/save-load-arrays.html

numpy vs lista i Python

- Mycket snabbare än listor.
  - Endast 1 datatyp per array.
  - Kontinuerligt minne - samma operation utförs samtidigt på flera element.
  - Använder cache på ett mer effektivt sätt
- Optimerad för operationer med arrayer.

Multiplikation av en python lista resulterade i att den listan duplicerades.

```
a = [1, 2, 3]
a*2
```

```
[1, 2, 3, 1, 2, 3]
```

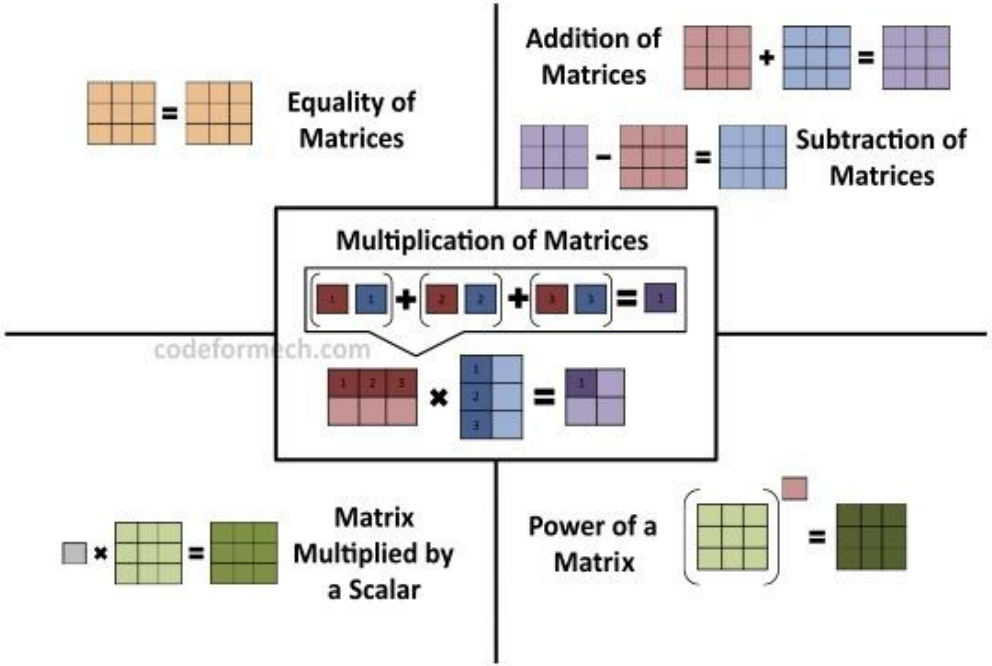
I numpy utförs de matematiska beräkningarna *per element*.

```
import numpy as np
b = np.array([1,2,3])
b * 2
```

```
array([2, 4, 6])
```

Kan även utföra booleanska operationer på en hel array.

matrisoperationer



numpy - funktioner

- **np.array** - skapa array, se upp med dimensioner, för 2D array måste båda vara lika stora etc. (a i ex. nedan)
- **a.ndim** - dimension
- a.**shape** - (rader, kolumner)
- a.**reshape** - ändra form
- a.**dtype** - datatyp, kan specifiera int8, int16, float etc.
- a.**itemsize** - storlek i byte per element
- a.**nbytes** - storlek av hela arrayen
- **[]** med : och - samma sorts indexering som i listor
- **np.linspace**(0, 10, 33) - 33 element jämnt fördelade från 0 till 10.
- **np.random.rand(r, c, d)** - slumpmässiga siffror mellan *0 och 1* formatet r- rader, c- kolumner och d-dimensioner
- **np.random.randint** - matris med slumpmässiga heltal i formatet som anges med size=(3, 3). Kan även återskapa matriser av samma form med random värden, eller ett specifikt nummer.

använd a.**copy()** för att kopiera en array, annars har de samma minnesreferens.

np.**genfromtxt** - generar en array från en txt fil. Kan hantera saknade värden.

numpy.loadtxt - laddar data from olika filtyper i en array.

np.**fromregex** - skapar en array från textfil efter regex manipulering

numpy - XML

Vi kan även fånga upp information från webben och sedan konvertera till olika format.

Ett exempel är att läsa av ett RSS-feed och sedan spara det i XML format.

RSS är vanlig text formaterad som XML.

XML kan sedan konverteras till en dictionary eller numpy.array.

```
#Python code to illustrate parsing of XML files
# importing the required modules
import csv
import requests
import xml.etree.ElementTree as ET

def loadRSS():

    # url of rss feed
    url = 'http://www.hindustantimes.com/rss/topnew

    # creating HTTP response object from given url
    resp = requests.get(url)

    # saving the xml file
    with open('topnewsfeed.xml', 'wb') as f:
        f.write(resp.content)
```

numpy - json

Python har ett **json** bibliotek som kan ladda upp en json fil i t.ex. en dictionary.

Används i samarbete med pythons inbyggda filhantering.

Sedan kan man använda **numpy.array(*dict*)** för att konvertera denna till en numpy array.

```
import json

# Opening JSON file
f = open('data.json')

# returns JSON object as
# a dictionary
data = json.load(f)

# Iterating through the json
# list
for i in data['emp_details']:
    print(i)

# Closing file
f.close()
```

regex

Data Processing - Regular expressions

Letar efter mönster som matchar det vi efterfrågar. Kan vara **specifika** eller **generella**. Resultaten kan sedan exporteras.

- **Literals** - bokstavligen vad vi vill ha. t.ex. bokstaven ‘q’
- **Metacharacters** - “kod” för vad vi vill ha: `^`, `[]`, `.` m.fl.
- **Escape sequence** - sök efter tecken som är vanligtvis speciella, såsom metatecken.

Öva: regex.alf.nu

`A-Z 0-9 - % _ . @`

Ex: Kolla om något har formatet av en e-mail address:

```
^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}$
```

^ - Börjar med (string eller rad)

[] - mönstret vi vill ha (literals, set of characters)

$ - slutar med

+ - en eller fler

. - vilket tecken som helst förutom \n (newline)

{2,} - 2 eller fler

Miljöer

conda environment

Skapa en ny conda “miljö” med följande paket:

- python 3.9.7 (senaste version som fungerar med pandas)
- pandas 1.3.5 (installerar både pandas och numpy)
- jupyter (kan utelämnas om du använder Google Colab)
- jupyterlab (se ovan)
- openpyxl (för att öppna xml)

Ex: To create an environment with a specific version of Python and multiple packages:

```
conda install python=3.9.7 pandas=1.3.5 jupyter jupyterlabs
```

Miljöer - alternativ

Google Colab

- Samma som jupyter (samma filtyp) men resurserna ligger på en server.
- Kan dela med dig genom länk (men även github).
- Industristandard.
- Has advanced collaboration features.

Colab - welcome

Sagemaker

Amazons alternativ.

- Jupyter.
- Kan installera paket.
- CPU - active for 12 hours.
- GPU - maximum 4 hours.
- Runtime will be restarted when the time limit is reached.
- Files - saved to the persistent storage (15GB).

studio-lab-examples (git)

Kaggle

Resurser på Kaggle:

- Data (datasets)
- Kurser
  - Python
  - Pandas
  - Data Science
  - Machine learning...
- Tävlingar
- Notebooks - logga in med Google.
  - Docker miljöer

Länkar

- Conda user guide
- Data Science | Google Cloud
- Regex cheatsheet
- Python regex: HOW TO
- Python regex w3schools
- Om Data science
- Operationer på matriser
- numpy.org
- numpy tutorials (official)
- Sagemaker
- Google vs. Sagemaker

Övningar:

- XML - parsing
- XML - JSON
- JSON - XML
- Saving and sharing numpy arrays
- Hitta ett rss-feed från en sida. Ladda in det i ett XML format och senare i en numpy array.