Data Science and Visualization (DSV, F23)

1. Data Science and Data

Hua Lu
https://luhua.ruc.dk; luhua@ruc.dk

PLIS, IMT, RUC

Agenda

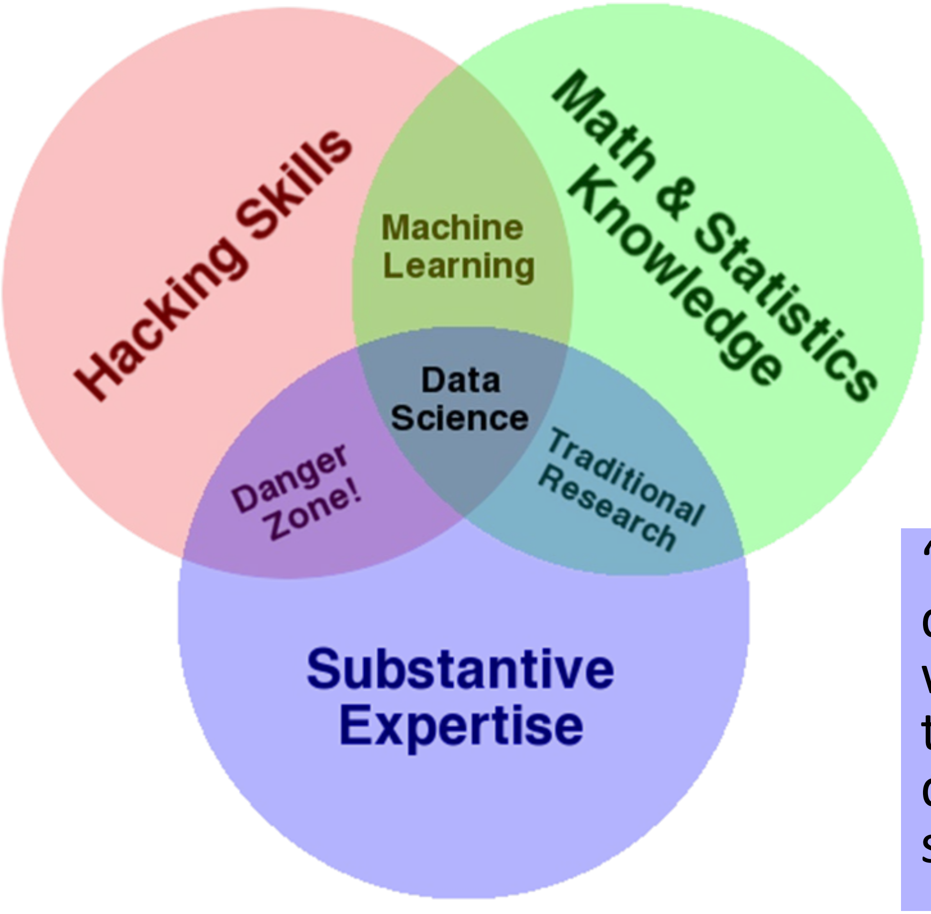
- **What is data science**
- Process of data science
- Data types
- Jupyter Notebook

What is Data Science?

- “Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining, machine learning and big data. “ ---Wikipedia
  - Subject nature
  - Technical means
  - Purpose
  - Related subjects
  - data

Drew Conway’s Venn Diagram

“…to manipulate text files at the command-line, thinking algorithmically, and be interested in learning new tools.”



“(To extract insight from data,) you need to apply appropriate math and statistics methods, which requires at least a baseline familiarity with these tools.”

“…some motivating questions about the world and hypotheses that can be brought to data and tested with statistical methods.”

https://s3.amazonaws.com/aws.drewconway.com/viz/venn_diagram/data_science.html

Fundamental Elements of Data Science

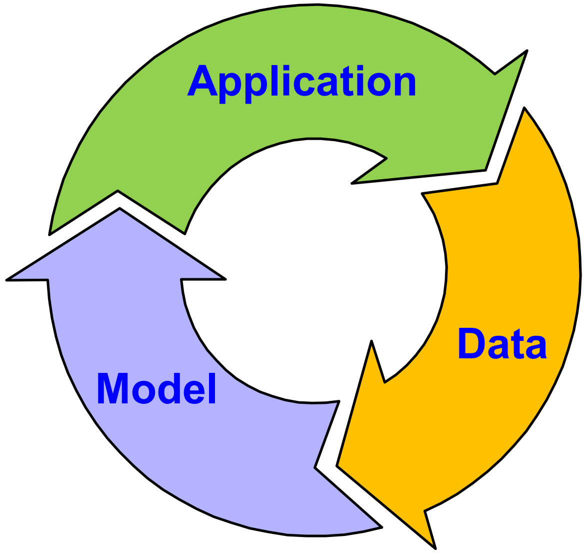
- Domain knowledge
- Computer Science skills
  - Machine learning
  - Data mining
  - Data visualization
  - Database
  - Data type dependent skills, e.g., text, image or multimedia data
- Math and statistics
  - Linear algebra, optimization (very often you just need to find the right tools)
  - Not a must, depending your domain and questions
  - Neither is it the emphasis of this course ☺

Two major concerns of Data Science:

- **Efficiency**: automation, do it *quick*.
- **Effectiveness**: validity, get *valid* insight.

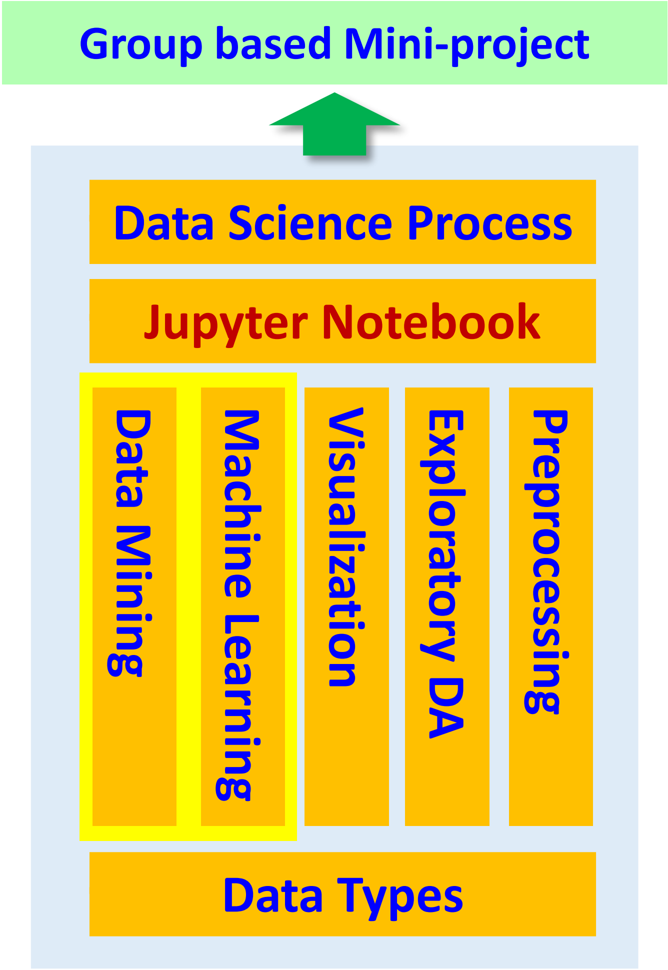
Why is Data Science popular *recently*?

- Data acquisition and collection
    - IoT, personal devices (smartphones), continuous digitialization
- Computing capacity
    - Increased performance of computer processors, e.g., CPU and GPU
    - Increased storage capacity for huge amounts of data
- New (or enhanced) applications
    - Fraud detection and risk control
        - E.g., online commerce, fake accounts in social networks
    - Bioinformatics and new drug discovery
    - 5G, Virtual Reality (VR) and Augmented Reality (AR)



Lecture Topics (*tentative*)

1. Data and data science process
2. Data preprocessing and exploratory data analysis
3. Machine learning essentials and classification I
4. Classification II
5. Classification III
6. Regression
7. Clustering I
8. Clustering II
9. Association rules
10. More visualization, storytelling with data (miniproject status)



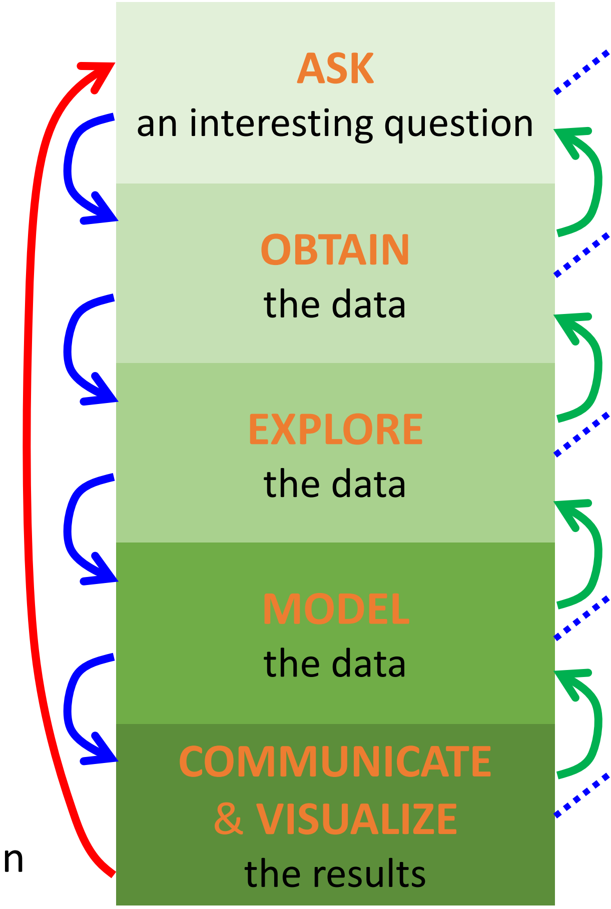
Agenda

- What is data science
- **Process of data science**
- Data types
- Jupyter Notebook

Steps of Data Science

1. Ask an interesting question
   - Skills: science, domain expertise, curiosity
   - Tools: your brain, taling to experts, experience
2. ★ Obtain the data
   - Skills: Web scraping, data cleansing, database queryingg, other CS skills
   - Tools: Python, pandas
3. ★ Explore the data
   - Skills: Get to know data, form hypotheses, patterns or outliers?
   - Tools: numpy, pandas
4. ★ Model the data
   - Skills: regressoin, ML, validation
   - Tools: scikits learn, pandas
5. ★ Communicate & visualize the results
   - Skills: presentation, speaking, writing, visualization
   - Tools: matplotlib, Excel

These are non-exhaustive lists!



- What is the scientific **goal**?
- What would you do if you had all the **data**?
- What do you want to **predict** or **estimate**?

- How were the data **sampled**?
- Which data are **relevant**?
- Are there **ethical** issues?

- **Plot** the data.
- Are there **patterns**?
- Are there **outliers**?

- **Build** the model.
- **Train** the model.
- **Validate** the model.

- What did we **learn**?
- Does the results make **sense**?
- Can we tell a **story**?

Adapted from Joe Blitzstein

The First Two Steps

- Ask a Question
  - Brainstorming: scientific goal, data, task
  - People working in academia and industry may have different perspectives.
- Obtain the data
  - From the operation if you’re in business
  - Your own research
  - Generate by your own (smartphone data)
  - Ethical issues: GDPR, privacy, consent from generator and/or concerned parties
  - Open source

Open Source Data in DK

- Data from Danmarks Statistik
    - https://www.dst.dk/da/TilSalg/Forskningsservice/Data
- Open Data Denmark (danske kommuner og regioner)
    - https://www.opendata.dk/
- Danish Agency for Digitalisation
    - https://en.digst.dk/
- Climate and weather data
    - https://www.dmi.dk/

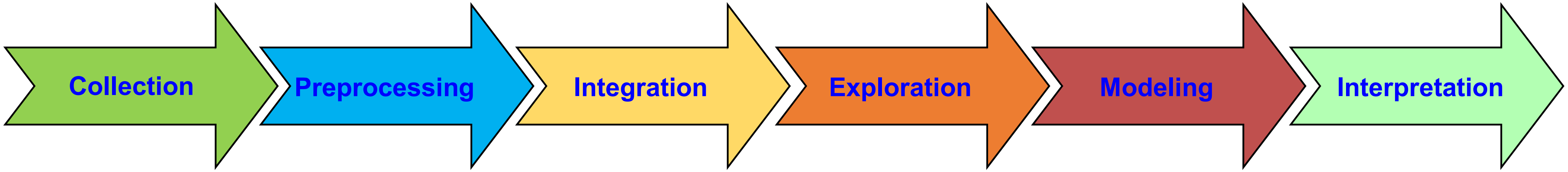
Other Online Data Sources

- Kaggle: Machine Learning and Data Science Community
  - https://www.kaggle.com/
- UCI Machine Learning Repository
  - https://archive.ics.uci.edu/ml/index.php
- AWS Public Data sets
  - https://registry.opendata.aws/
- NASA data
  - Earth data: https://earthdata.nasa.gov/
  - Space data: https://pds.nasa.gov/datasearch/data-search/
- Google
  - Public Data sets: https://cloud.google.com/public-datasets
  - Data Search: https://datasetsearch.research.google.com/
  - Google Research Data: https://research.google/tools/datasets/
- Wikipedia
  - https://en.wikipedia.org/wiki/Wikipedia:Database_download
- More at https://www.dataquest.io/blog/free-datasets-for-projects/

Data Cleansing

- Before/While/After exploring the data, you may find the data quality is not as good as expected.
  - Dirty data (noises)
  - Missing data (holes, NULL values)
  - Redundant data (duplicates)
- In such a case, preprocessing is needed to improve the data quality before it is used in subsequent data modeling.
  - Such preprocessing is called **data cleansing**.
  - Preprocessing is more than cleansing, e.g., data transformation from one format to another is needed sometimes in preprocessing.
- Techniques for data cleansing
  - Depending on how data is generated: Logical, statistical, learning
  - It is a broad research topic itself

Another Data Science Process



- Extraction
- Cleansing
- Transformation
- Annotation

Needed if data comes from multiple sources

- Data mining
- Statistics
- Machine learning

- Visualization
- Communication
- Reporting

Remarks

- Some component may be omitted in a specific case.
  - E.g., integration is only needed when data comes from multiple sources
- Visualization may also be needed in exploration.
- Again, backward loops may be needed.
  - **Iterative**
  - **Interactive**

Agenda

- What is data science
- Process of data science
- **Data types**
- Jupyter Notebook

Data Organization and Storage

Data organization	Structured data	Semi-structured data	Unstructured data
Meaning	Data in well-defined tables	In-between, partly structured	Data without clear structure
Storage and/or data examples	• Relational databases • Data warehouse	• XML • JSON (JavaScript Object Notation) files • BibTex files • CSV (comma separated value) files	• Audio • Image • Video • Text • Natural language

All data objects are homogeneous

Data objects are heterogeneous

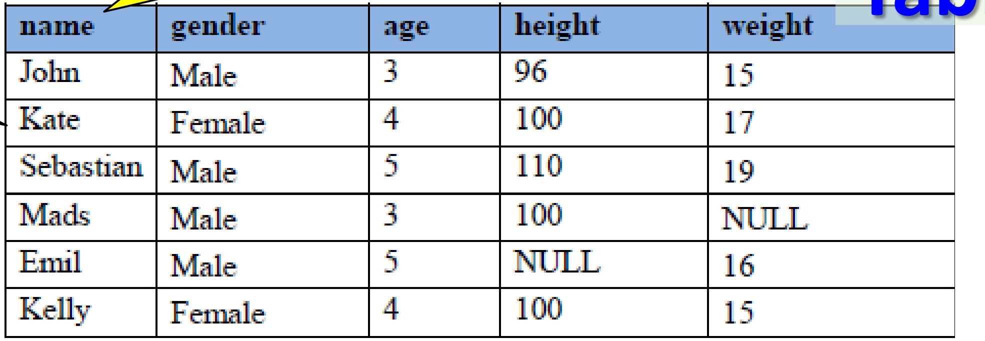
Types of Data Sets (1)

- **Record**
  - Relational records (table)
  - Data matrix (cube)
    - E.g., numerical matrix, crosstabs
  - Document data
    - E.g., text documents: term-frequency vector
  - Transaction data

Each **row** is a tuple/record

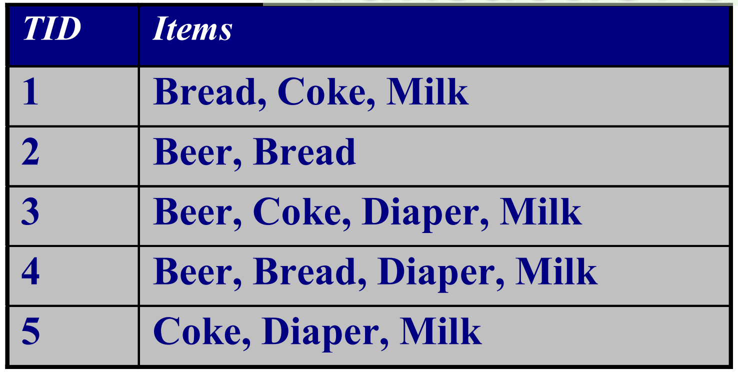
5 **attributes/columns**

Table



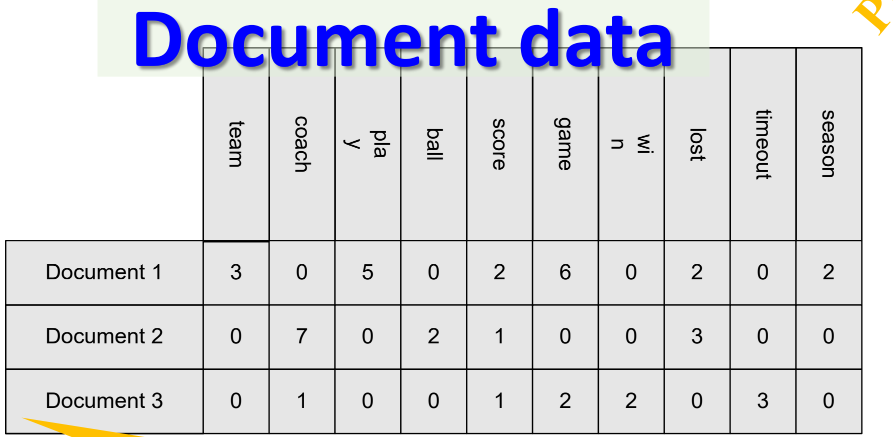
name	gender	age	height	weight
John	Male	3	96	15
Kate	Female	4	100	17
Sebastian	Male	5	110	19
Mads	Male	3	100	NULL
Emil	Male	5	NULL	16
Kelly	Female	4	100	15

Transactions



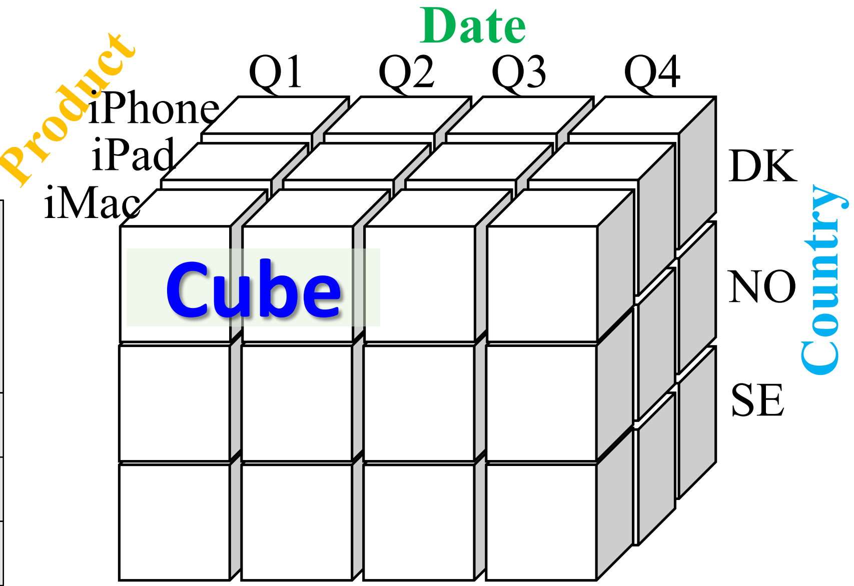
TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Document data



	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Not the *original* documents!

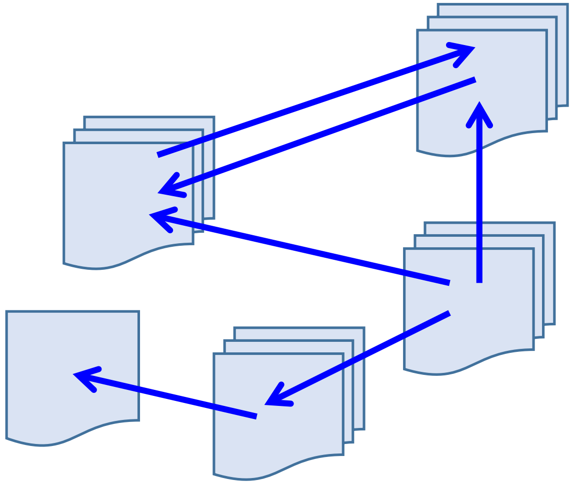


Cube

Types of Data Sets (2)

- **Graph and network**
  - World Wide Web
  - Social or information networks
  - Molecular structures

- Node/vertex for an entity
- Edge/link for the relation between two entities
- An entity can have/produce information; so does an edge
  - Different types of information in different types of graphs

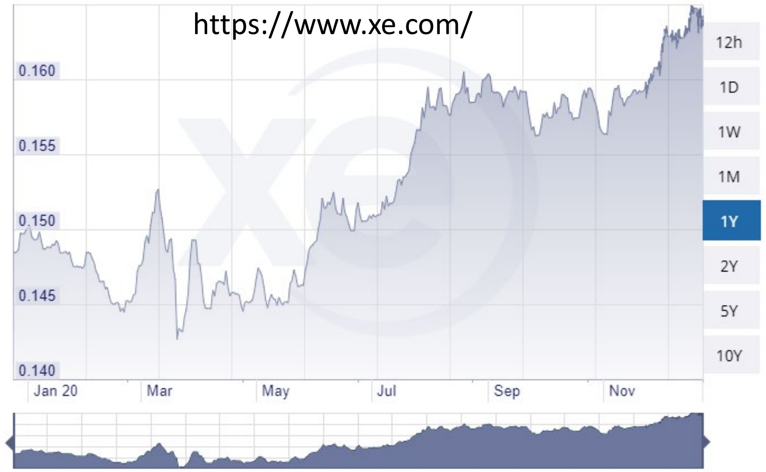


http://clipart-library.com/

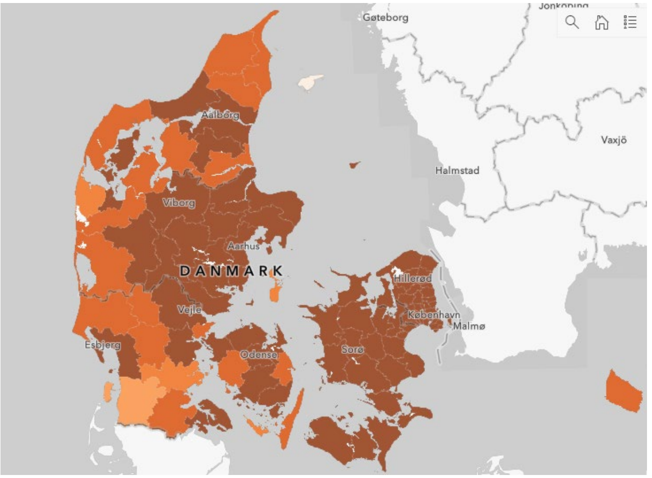
http://clipart-library.com/

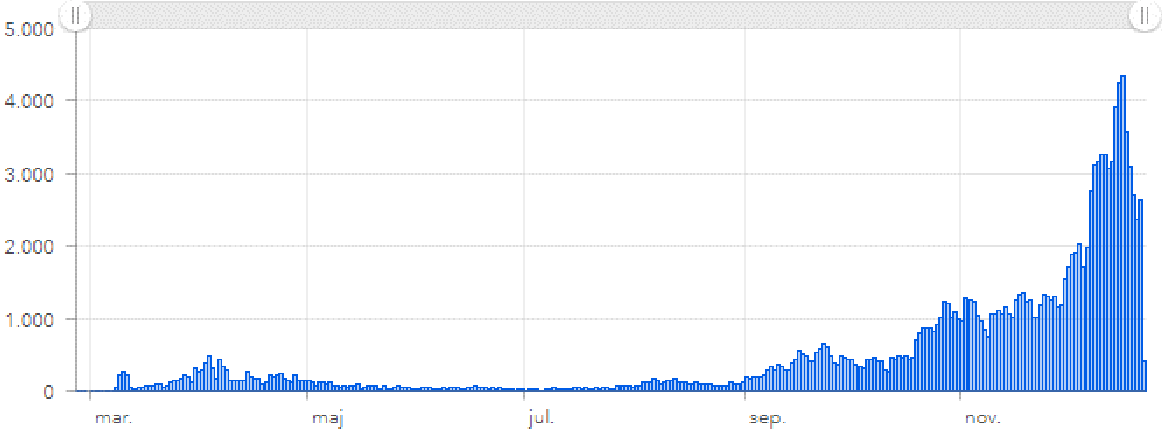
Types of Data Sets (3)

- **Ordered**
  - Video data: sequence of images
  - Temporal data: time-series
  - Sequential data: transaction sequences
  - Genetic sequence data
- **Spatial, image and multimedia**
  - Spatial data: maps
  - Image data
  - Video data



https://www.xe.com/

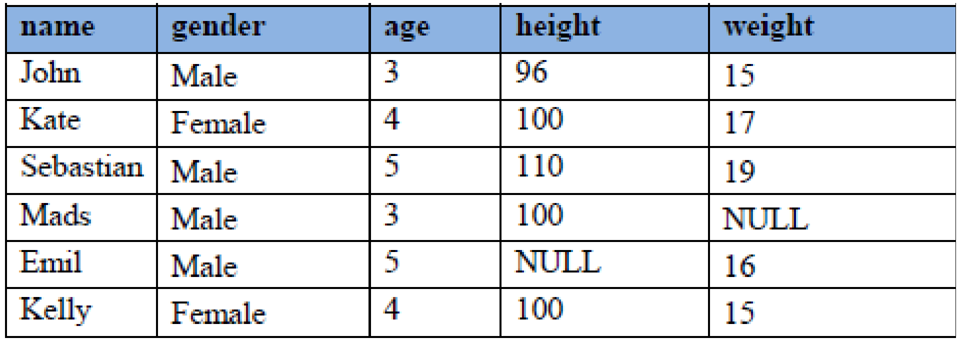




https://covid19.ssi.dk/overvagningsdata

Important Characteristics of *Structured* Data

- **Dimensionality**
  - Number of dimensions/attributes
  - Curse of dimensionality
- **Sparsity**
  - How many cells in a matrix have values?
  - Only presence counts
- **Resolution**
  - Data scale
  - Patterns depend on the scale
- **Distribution**
  - Centrality and dispersion



name	gender	age	height	weight
John	Male	3	96	15
Kate	Female	4	100	17
Sebastian	Male	5	110	19
Mads	Male	3	100	NULL
Emil	Male	5	NULL	16
Kelly	Female	4	100	15

Attributes

- **Attribute** (or dimensions): a data field, representing a characteristic or feature of a data entity.
  - E.g., customer _ID, name, address of a customer.
- Attribute data/value types:
  - Qualitative
    - Nominal
    - Binary
    - Ordinal
  - Quantitative (numeric)
    - Interval-scaled
    - Ratio-scaled

Qualitative Data/Value Types

- **Nominal**: categories, states, or “names of things”
  - *Hair_color* = {*auburn, black, blond, brown, grey, red, white*}
  - marital status, occupation, ID numbers, zip codes
- **Binary**
  - Nominal attribute with only 2 states (0 and 1)
  - Symmetric binary: both outcomes equally important
    - e.g., gender
  - Asymmetric binary: outcomes not equally important.
    - e.g., medical test (positive vs. negative)
    - Convention: assign 1 to most important outcome (e.g., HIV positive)
- **Ordinal**
  - Values have a meaningful order (ranking) but magnitude between successive values is not known.
  - *Size* = {*small, medium, large*}, grades, army rankings

Odering, comparison

Numeric Data/Value Types

- Quantity (integer or real-valued)
- **Interval**
  - Aka interger, measured on a scale of equal-sized units
  - Values have order
    - E.g., temperature in C˚or F˚, calendar dates
  - No ‘absolute’ zero-point
- **Ratio**
  - Inherent, ‘absolute’ zero-point
  - We can speak of values as being an order of magnitude larger than the unit of measurement (10 K˚ is twice as high as 5 K˚).
    - e.g., temperature in Kelvin, length, counts, monetary quantities

Numeric: Discrete vs. Continuous

- Discrete values
    - Has only a finite or countably infinite set of values
    - Represented as integer variables
        - E.g., 5 students in a group, 10 groups in a semester
- Continuous values
    - Has real numbers as attribute values
        - E.g., temperature, height, or weight
    - Practically, real values can only be measured and represented using a finite number of digits in computers
    - Continuous attributes are typically represented as floating-point variables

First Look at Your Data

- Structured, semi-structured, or non-structured data?
- Attributes and data values
- Basic statistical description of your data
  - Mean, media, mode
  - Distribution
  - Plot, e.g., histogram
- Data visualization
  - More than just plotting--better understand the data through visualization.
- Data similarity and dissimilarity
  - Different measures for different data (attribute) types

Agenda

- What is data science
- Process of data science
- Data types
- **Jupter Notebook**
  - Introduction
  - A walk-through example

Jupyter Notebook

- Important application requirements
  - **Iterative**: To repeat some procedures conveniently,
    e.g., machine learning: training model -> validating model -> testing model
  - **Interactive**: E.g., to see the analytics results immediately/visually and then revise code according to the results
- A Web-based powerful tool for *interactively* developing and presenting data science projects.
- Within Jupyter Notebook, one can use Python (or other programming languages) to write and run code *iterative*ly.
- We also use ‘notebook’ to refer to such a document that combines code, text, output and many others.

What is Python?

- An interpreted, high-level and general-purpose programming language.
  - Simple and easy to learn
  - Free and open source
  - Platform-independent
  - Object-oriented: *Everything* is an object.
  - Embeddable
  - Extensive libraries
  - The *lingua franca* for data science
- Interpreted vs. Compiled language
  - *Bytecodes* for virtual machines vs. *Binary codes* for physical machines
  - Analogy: Simultaneous interpretation vs. Book translation

Survey result

- Out of the 47 students who responded to the survey, 37 are not familiar with Python.

Why Python

- General-purpose programming language with many powerful third-party libraries
  - Data loading, visualization, statistics, NLP, image processing and so on
  - Interactive and GUI
  - Integrating with other languages, e.g., C, MATLAB and R
- **Jupyter Notebook**: interactive code running in a browser
- **scikit-learn**: most prominent open source Python library for machine learning
- **Numpy**: multidimensional arrays (matrix), linear algebra, Fourier transform, pseudorandom number generation
- **SciPy**: advanced linear algebra, mathematical function optimization, signal processing, statistical distributions
- **matplotlib**: scientific plotting library (visualization)
- **pandas**: data wrangling and analysis, Excel and SQL-like data, CSV, queries/joins of tables
  - Built on Numpy with convenient data structures, e.g., **Series**, **DataFrame**

Example in Jupyter Notebook

- First Jupyter Notebook
  - Markdown
  - Code
  - Cell
  - Execution
  - …
- First dataset in Jupyter Notebook
  - Fortune 500 data
  - Loading, displaying, exploring
  - Basic visualization
  - …

Concepts of Objects and Classes in Python

- A **class** is an abstraction of all objects of the same type.
  - A class can have variables and methods that together characterize the class.
  - A method is a function defined for a class
  - E.g., Person can be a class with a variable name and a method die.
- An **object** is an concrete instance of a class.
  - E.g., Jack can be an object of class Person. Every person dies, sooner or later.
- One class may have many objects.
  - Think how many persons there are on this planet.
- One object *usually* corresponds to only one class.
- Everything in Python is an object.
- We will learn more later in this course.

PYTHON FOR DATA SCIENCE CHEAT SHEET

Python Scikit-Learn

Introduction

Scikit-learn: "sklearn" is a machine learning library for the Python programming language. Simple and efficient tool for data mining, Data analysis and Machine Learning.

Importing Convention - import sklearn

Preprocessing

Data Loading

- Using NumPy:

```
>>>import numpy as np
>>>a=np.array([(1,2,3,4),(7,8,9,10)],dtype=int)
>>>data = np.loadtxt('file_name.csv',
    delimiter=',')
```

- Using Pandas:

```
>>>import pandas as pd
>>>df=pd.read_csv('file_name.csv',header=0)
```

Train-Test Data

```
>>>from sklearn.model_selection
import train_test_split

>>> X_train, X_test, y_train, y_test =
train_test_split(X,y,random_state=0)
```

Data Preparation

- Standardization

```
>>>from sklearn.preprocessing import
StandardScaler
>>>get_names = df.columns
>>>scaler =
preprocessing.StandardScaler()
>>>scaled_df = scaler.fit_transform(df)
>>>scaled_df =
pd.DataFrame(scaled_df,
columns=get_names)m
```

- Normalization

```
>>>from sklearn.preprocessing import
   Normalizer
>>>pd.read_csv("File_name.csv")
 >>>x_array = np.array(df['Column1'])
#Normalize  Column1
 >>>normalized_X =
preprocessing.normalize([x_array])
```

Working On Model

Model Choosing

Supervised Learning Estimator:

- Linear Regression:

```
>>> from sklearn.linear_model import
LinearRegression
>>> new_lr =
LinearRegression(normalize=True)
```

- Support Vector Machine:

```
>>> from sklearn.svm import SVC
>>> new_svc = SVC(kernel='linear')
```

- Naive Bayes:

```
>>> from sklearn.naive_bayes import
GaussianNB
>>> new_gnb = GaussianNB()
```

- KNN:

```
>>> from sklearn import neighbors
>>>
knn=neighbors.KNeighborsClassifier(n_ne
ighbors=1)
```

Unsupervised Learning Estimator:

- Principal Component Analysis (PCA):

```
>>> from sklearn.decomposition import
PCA
>>> new_pca= PCA(n_components=0.95)
```

- K Means:

```
>>> from sklearn.cluster import KMeans
>>> k_means = KMeans(n_clusters=5,
random_state=0)
```

Train-Test Data

Supervised:

```
>>>new_ lr.fit(X, y)
>>> knn.fit(X_train, y_train)
>>>new_svc.fit(X_train, y_train)
```

Unsupervised :

```
>>> k_means.fit(X_train)
>>> pca_model_fit =
new_pca.fit_transform(X_train)
```

Post-Processing

Prediction

Supervised:

```
>>> y_predict =
new_svc.predict(np.random.random((3,5)))
>>> y_predict = new_lr.predict(X_test)
>>> y_predict = knn.predict_proba(X_test)
```

Unsupervised:

```
>>> y_pred = k_means.predict(X_test)
```

Model Tuning

Grid Search:

```
>>> from sklearn.grid_search import GridSearchCV
>>> params = {"n_neighbors": np.arange(1,3), "metric":
         ["euclidean", "cityblock"]}
>>> grid = GridSearchCV(estimator=knn,
param_grid=params)
>>> grid.fit(X_train, y_train)
>>> print(grid.best_score_)
>>> print(grid.best_estimator_.n_neighbors)
```

Randomized Parameter Optimization:

```
>>> from sklearn.grid_search import RandomizedSearchCV
>>> params = {"n_neighbors": range(1,5), "weights":
["uniform", "distance"]}
>>> rsearch = RandomizedSearchCV(estimator=knn,
param_distributions=params, cv=4, n_iter=8, random_state=5)
>>> rsearch.fit(X_train, y_train)
>>> print(rsearch.best_score_)
```

Evaluate Performance

Classification:

1. Confusion Matrix:

```
>>> from sklearn.metrics import
      confusion_matrix
>>> print(confusion_matrix(y_test,
      y_pred))
```

2. Accuracy Score:

```
>>> knn.score(X_test, y_test)
>>> from sklearn.metrics import
      accuracy_score
>>> accuracy_score(y_test, y_pred)
```

Regression:

1. Mean Absolute Error:

```
>>> from sklearn.metrics import mean_absolute_error

>>> y_true = [3, -0.5, 2]
>>> mean_absolute_error(y_true, y_predict)
```

2. Mean Squared Error:

```
>>> from sklearn.metrics import mean_squared_error
>>> mean_squared_error(y_test, y_predict)
```

3. R² Score :

```
>>> from sklearn.metrics import r2_score
>>> r2_score(y_true, y_predict)
```

Clustering:

1. Homogeneity:

```
>>> from sklearn.metrics import
homogeneity_score
>>> homogeneity_score(y_true,
y_predict)
```

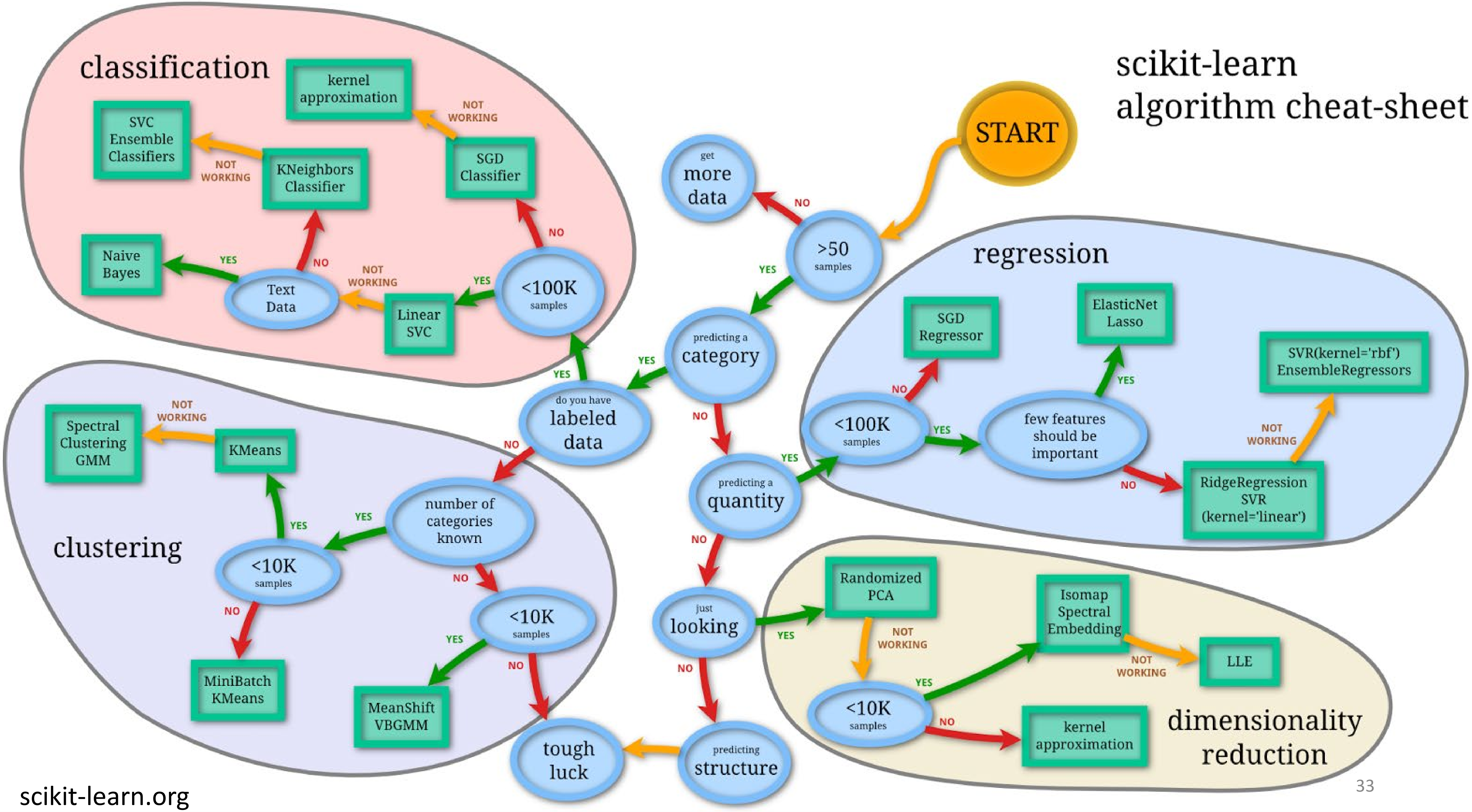
2. V-measure:

```
>>> from sklearn.metrics import
v_measure_score
>>> metrics.v_measure_score(y_true,
y_predict)
```

Cross-validation:

```
>>> from
sklearn.cross_validation
import cross_val_score
>>>
print(cross_val_score(knn,
X_train, y_train, cv=4))
>>>
print(cross_val_score(new_
lr, X, y, cv=2))
```

FURTHERMORE:
Python for Data Science Certification Training Course



scikit-learn algorithm cheat-sheet

scikit-learn.org

Mini-Projects and Exam

- Group based
  - 1 to 5 students per group
- Data
  - You can choose whatever dataset you want to work on, but not those (to be) used in the examples or exercises in the course.
- Deliverable
  - Jupyter Notebook script with project description, code, comments and URL of data
- Mini-project submission deadline to Digital Exam
  - Each group uploads only one submission to Digital Exam
  - **23:59 May 02, 2022**
- Exam
  - Oral, 20 minutes in total. (Internal censor: TA Jialiang Li and Masoumeh Vahedi)
  - It starts with a short presentation of the mini-projects.
  - We will refer to your mini-project report in the exam.

Tips for this course

- Spend sufficient time and get your hands dirty in coding!
- Don’t be afraid of programming in Jupyter Notebook
  - The best way to learn programming is programing ☺
  - Dare to try, dare to fail.
- Find datasets and play with them in Jupyter Notebook
- Read sample codes and learn by following examples.
- Refer to (online) documentations frequently.
- When you cannot resolve your problems, ask your fellow students, the teacher and TAs.

Summary

- Process of data science
  - Main steps
  - Iterative and interactive
- Data types
  - Data organization: Structured, semi-structured, unstructured
  - Data storage: CSV, Excel, database…
- Jupyter Notebook
  - Basic functionalities
  - Loading and displaying data
  - Data statistics

References

- Jupyter Notebook
  - How to use Jupyter Notebook in 2020: A Beginner's Tutorial: https://www.dataquest.io/blog/jupyter-notebook-tutorial/
- Data Science
  - Sinan Ozdemir: *Principle of Data Science*, Packt, 2016
    - Chapters 1-3
- Python
  - Swaroop C. H.: A Byte of Python. 2008.
  - Python Tutorial: https://www.tutorialspoint.com/python/index.htm

Exercises for today

1. Install Anaconda *Individual Edition* on your computer
   - https://www.anaconda.com/products/individual
2. Download the Titanic dataset (available in Moodle) to your own computer, and do the following in Jupyter Notebook
   1. Load the data
   2. Show the data (first 5 rows, last 5 rows and all rows)
   3. Check the data types of all dimensions
   4. Show the statistics of all dimensions
   5. Describe what you do in your notebook using Markdown