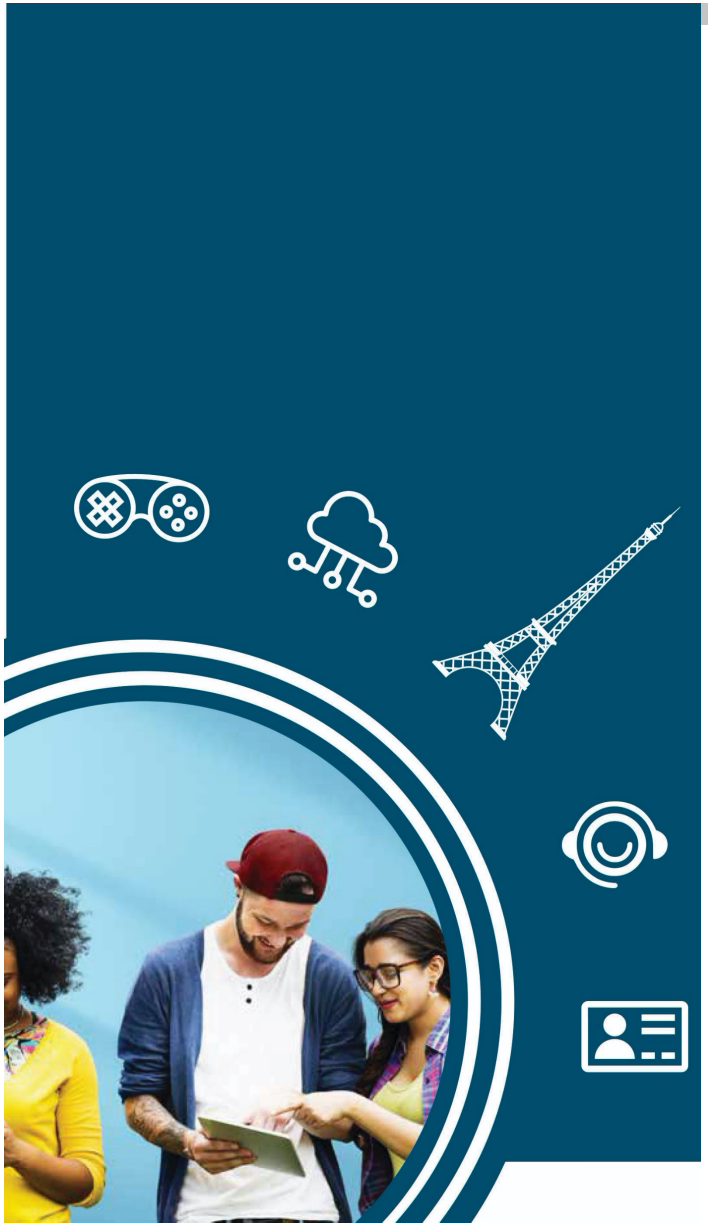


FOUNDATIONS OF STATISTICAL ANALYSIS & MACHINE LEARNING

Amric Trudel
amric.trudel@epita.fr

INTRODUCTION

Amric Trudel

- McGill University graduate (Montréal, Canada)
- École 42, former president of 42AI
- Consultant at OCTO Technology

TODAY'S SCHEDULE

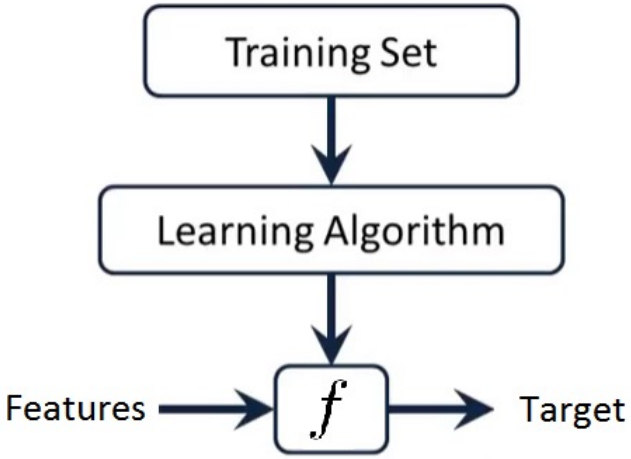
Program

- Introduction to Machine Learning
- Tools & libraries
- Data exploration
  - Overview
  - Implementation
  - Practice

WHAT IS ML?

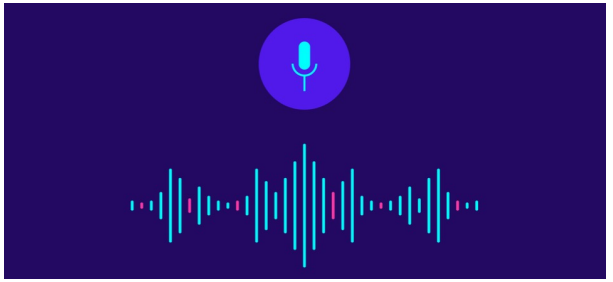
A definition

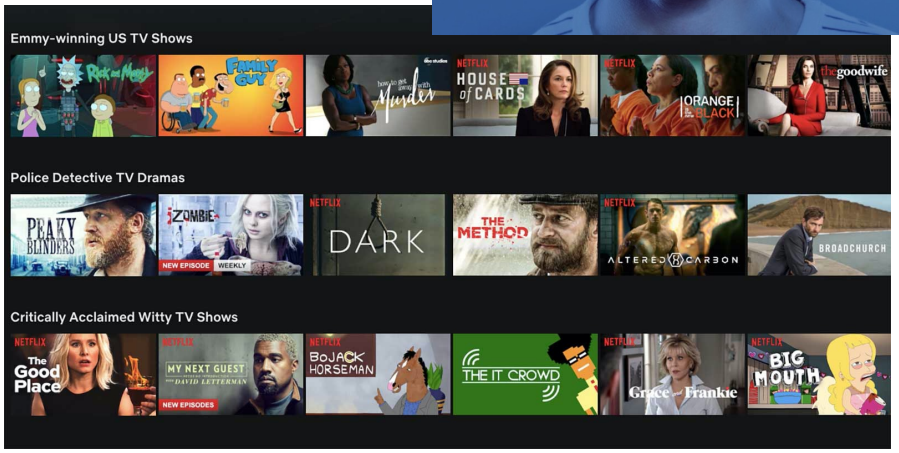
- Machine Learning (ML) is a subset of artificial intelligence that encompasses techniques aiming at making predictions after having built models based on sample data.



WHERE IS ML?

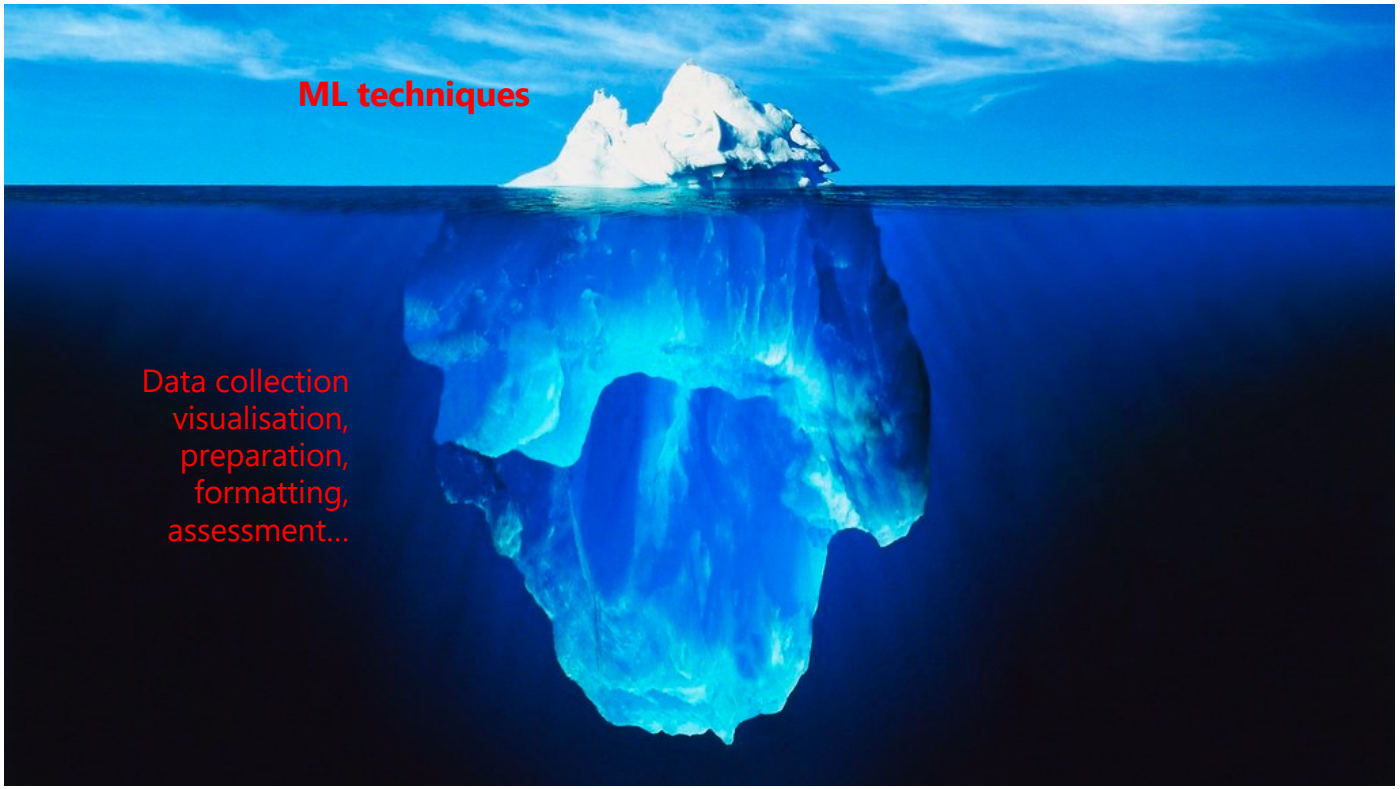
Use cases





HOW TO LEARN ML?

The underestimated part of ML



ML WORKFLOW

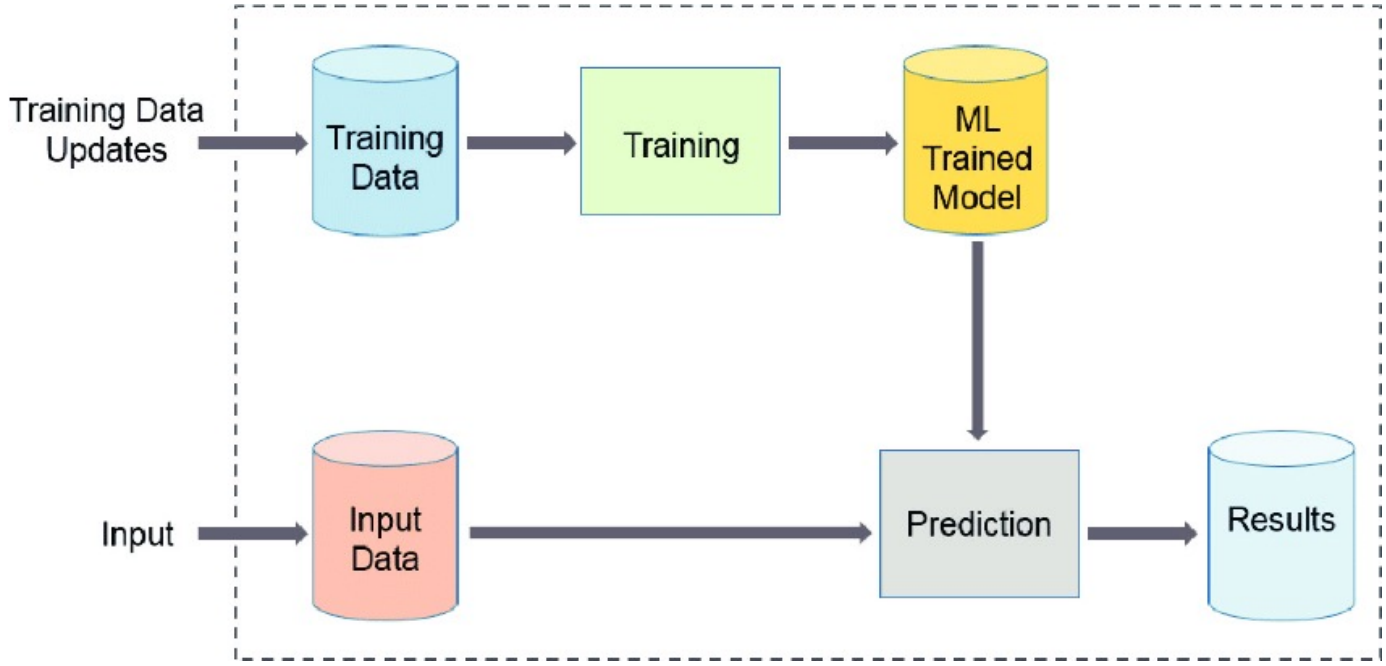
Usual process

- Collect data
- Clean and prepare the data
- Pick a model
- Train the model on the data
- Test and evaluate the model
- Improve the model

- Deploy the model to production

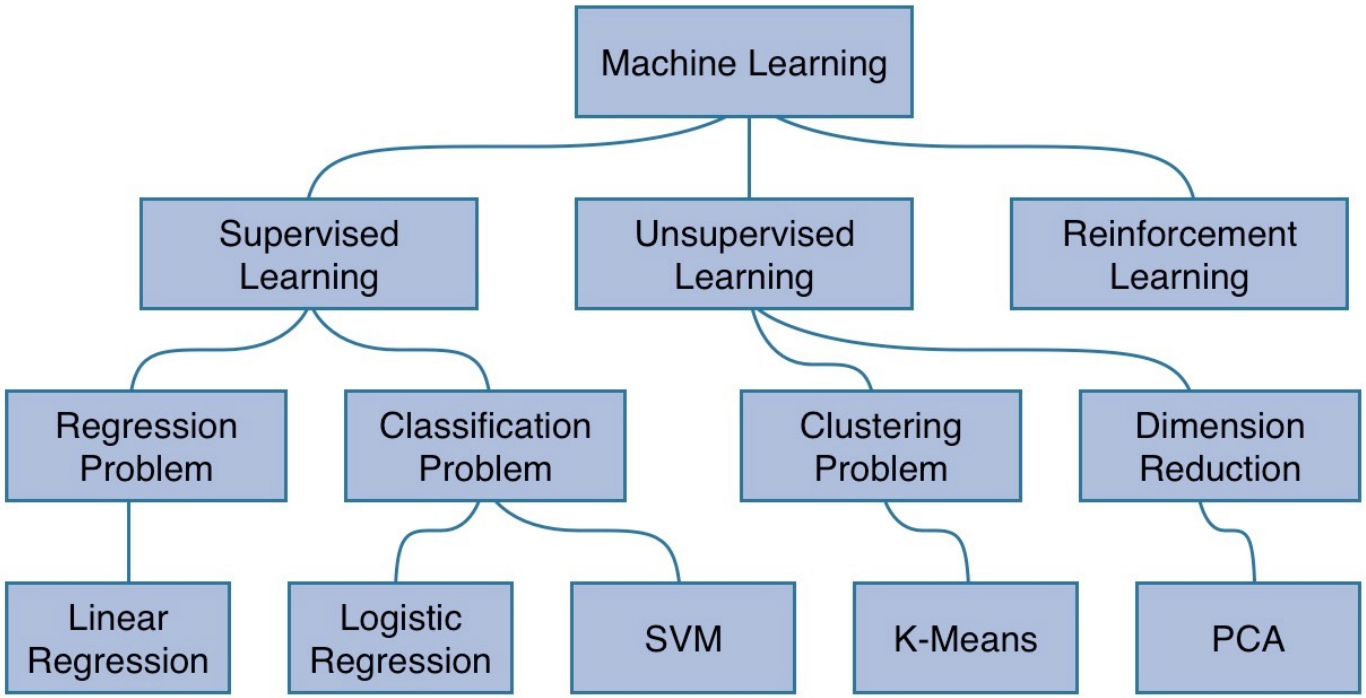
ML WORKFLOW

Usual process



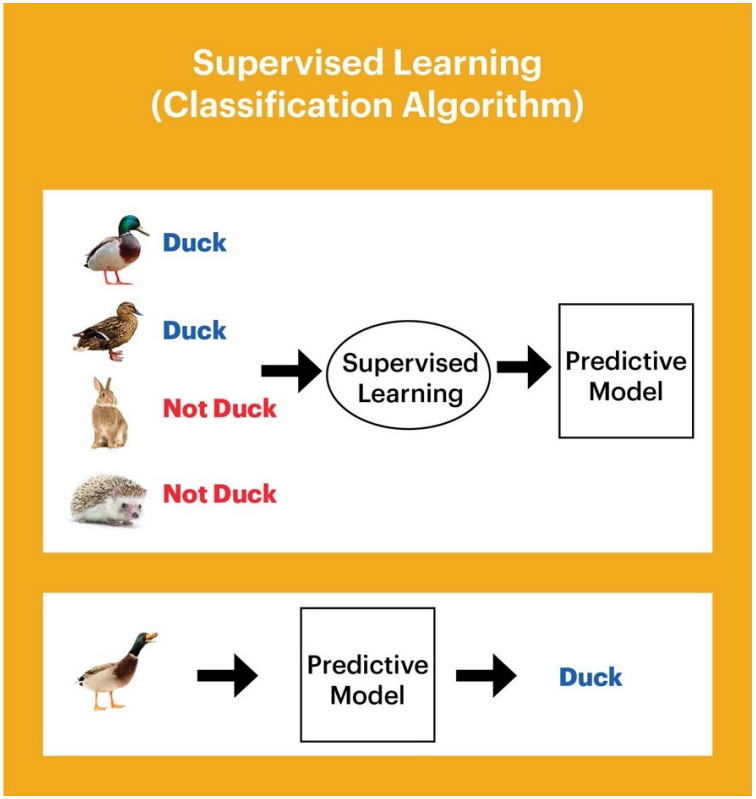
MACHINE LEARNING

And its multiple branches



SUPERVISED VS UNSUPERVISED LEARNING

Labeled vs. unlabeled

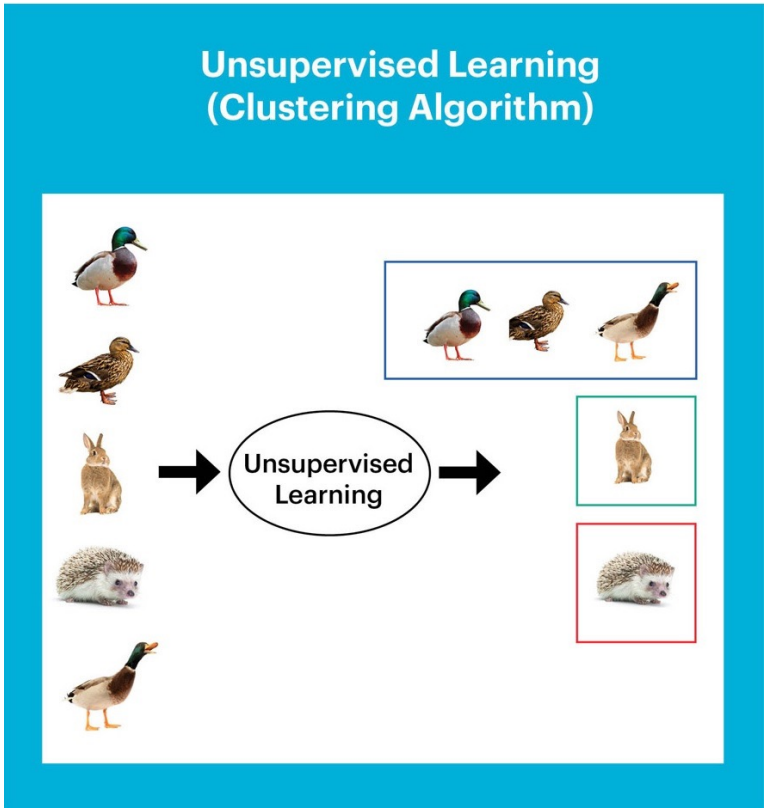


Supervised Learning

- Data is **labeled**
- We want the algorithm to learn the **relationship** between an object's features and its label

SUPERVISED VS UNSUPERVISED LEARNING

Labeled vs. unlabeled

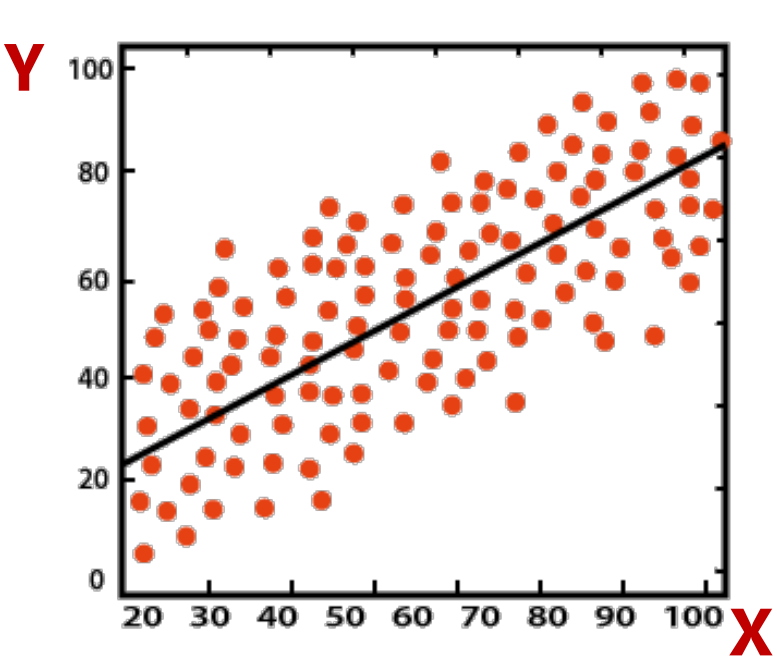


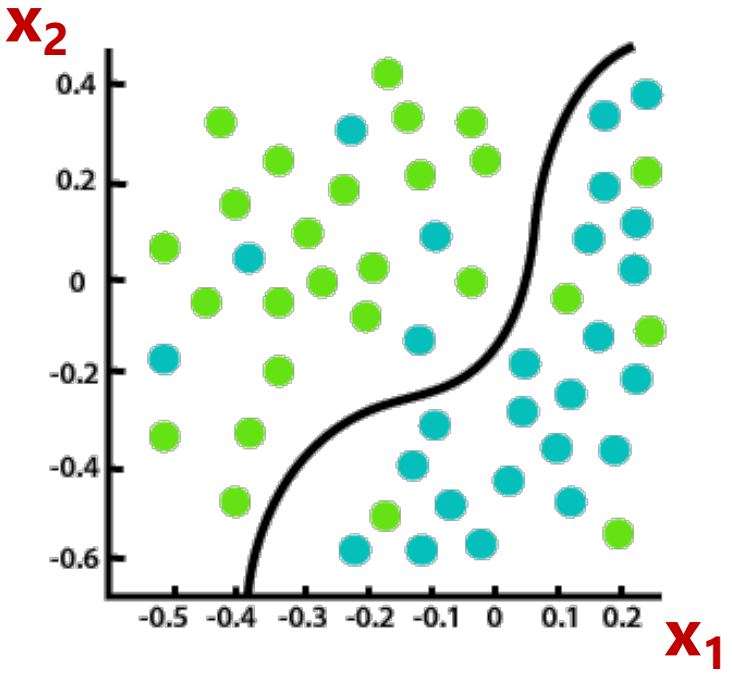
Unsupervised Learning

- Data is **unlabeled**
- We don't know exactly what we are looking for
- We want the algorithm to **find patterns and** structure in the data

SUPERVISED LEARNING

Continuous vs. discrete output



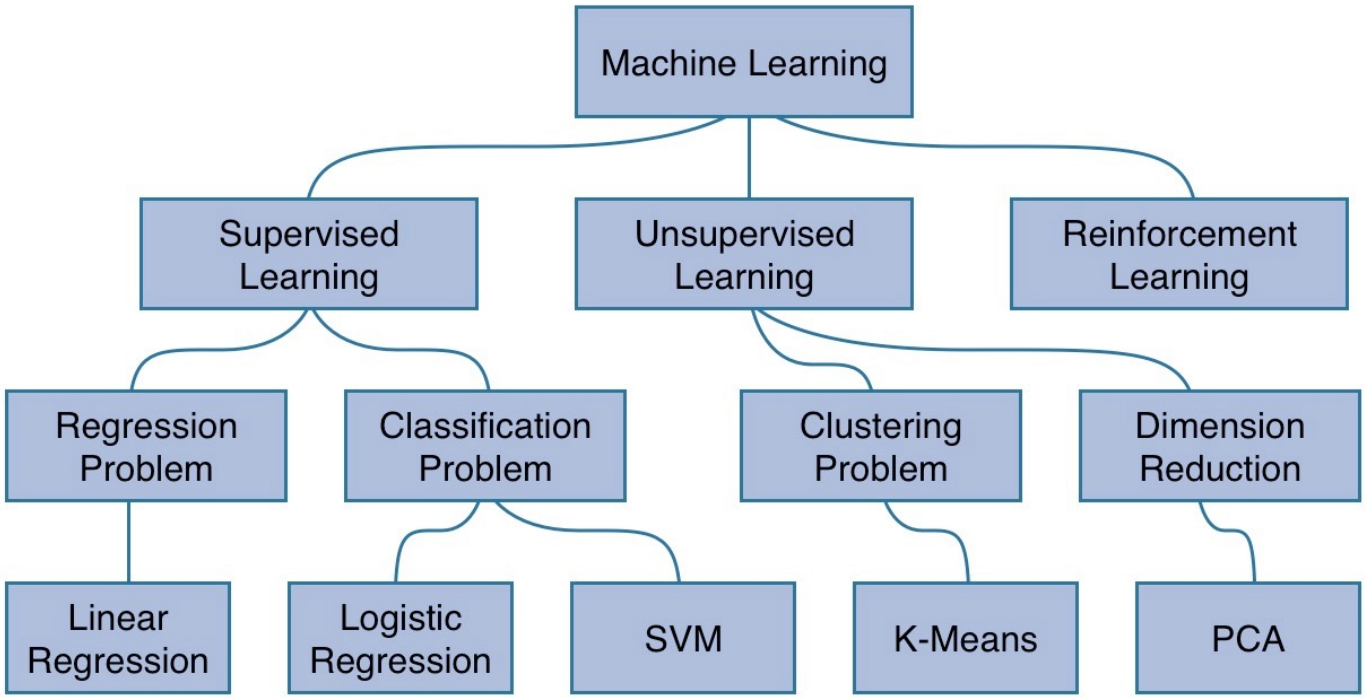


Regression 60.03

Classification "Not SPAM"

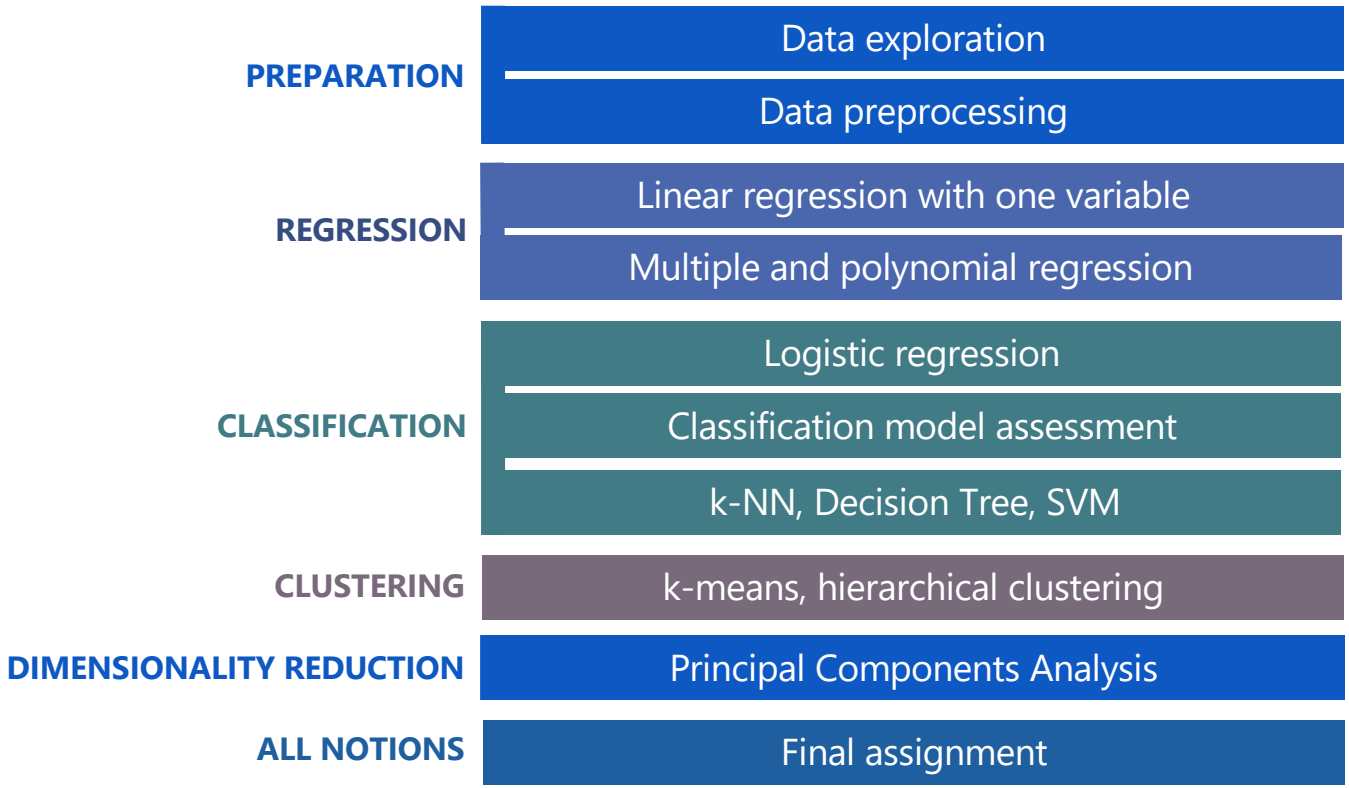
MACHINE LEARNING

And its multiple branches



COURSE PROGRAM

Structure



COURSE ORGANIZATION

- Class Schedule:
  - 20 min: homework correction
  - 50 min: theoretical background
  - 20 min: Python demo (by the teacher)
  - 60 min: practical exercise (by the students, to be finished at home)

- Material:
  - Slides and demo uploaded on Teams before each session
  - Practical exercise available on Teams as an assignment (with deadline)
  - Practical exercise correction uploaded on Teams after the deadline

- Grades:
  - Practical exercises: 40 %
  - Final exam: 60 %

METHODS

Interactions

- Don’t hesitate to **ask questions** during the teacher presentation
- When asked a question by the teacher during class, **don’t worry about providing a correct answer**. Just answer according to your current knowledge and intuition.
- You should take advantage as much as possible of the time dedicated to the **practical exercise** to ask specific questions to the teacher and make sure everything is sufficiently clear so that you can finish the assignment at home.

TOOLS & LIBRARIES

Installation

- Python 3.8 (or above). Make sure it includes pip (needed to install libraries).
- A development IDE: e.g. Visual Studio Code, Spyder.
- Jupyter notebook

```
pip install jupyterlab
```

- Libraries: scikit-learn, numpy, matplotlib, pandas

```
python -m pip install pandas
python -m pip install numpy
python -m pip install -U scikit-learn
python -m pip install -U matplotlib
```

TOOLS & LIBRARIES

Import

- Import the libraries you need before to use it in your code.

```
import numpy as np                    ← operations on array
import pandas as pd                   ← data manipulation
import sklearn                        ← ML algorithms
import matplotlib.pyplot as plt       ← plotting
import …
```

COURSE PROGRAM

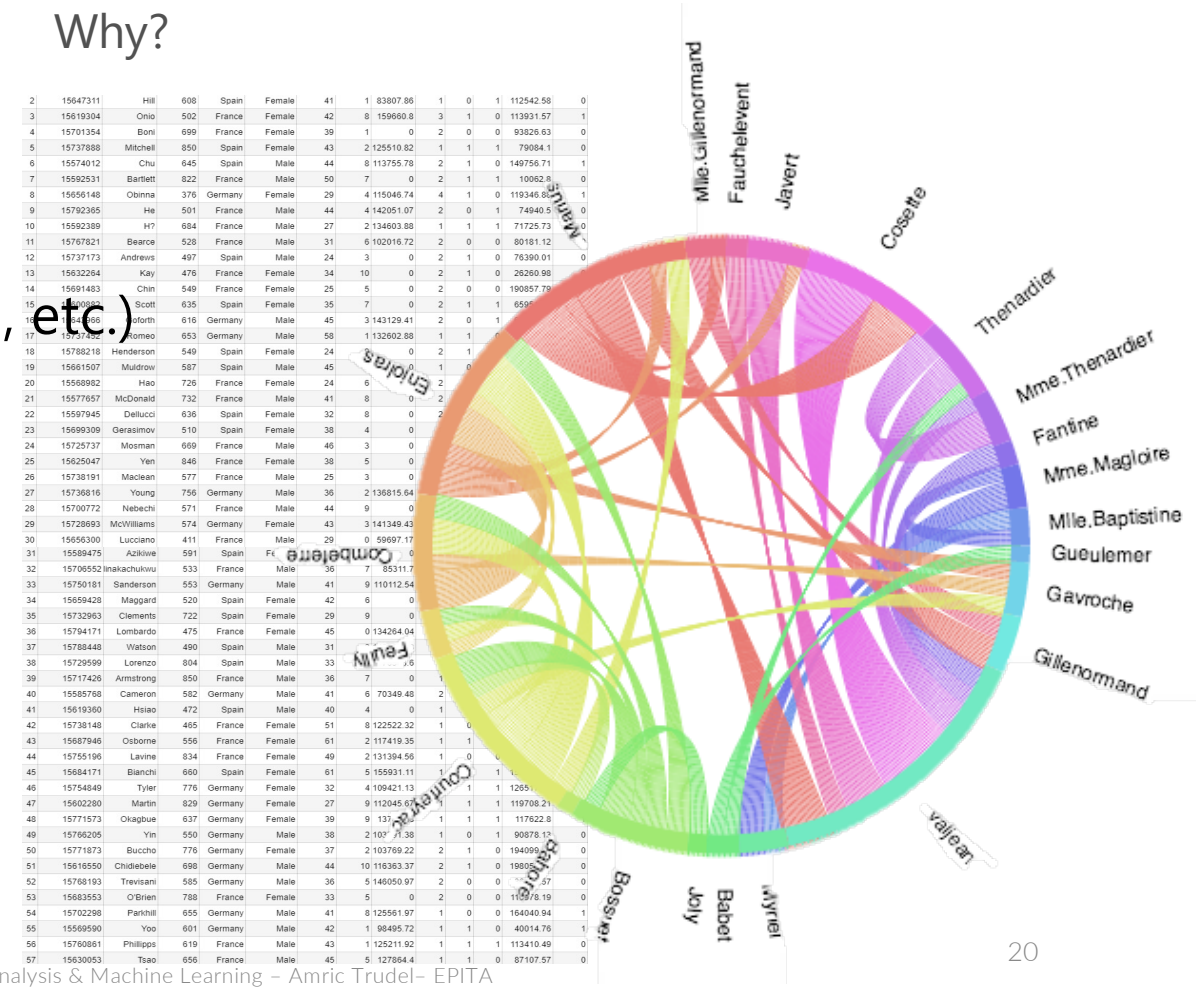
Structure

PREPARATION	Data exploration
	Data preprocessing
REGRESSION	Linear regression with one variable
	Multiple and polynomial regression
CLASSIFICATION	Logistic regression
	Classification model assessment
	k-NN, Decision Tree, SVM
CLUSTERING	k-means, hierarchical clustering
DIMENSIONALITY REDUCTION	Principal Components Analysis
ALL NOTIONS	Final assignment

EXPLORE THE DATA SET

- Understand your data
- Identify the variables
- Get some intuition (correlation, etc.)
- Identify problems in the data

Why?



EXPLORE THE DATA SET

Load

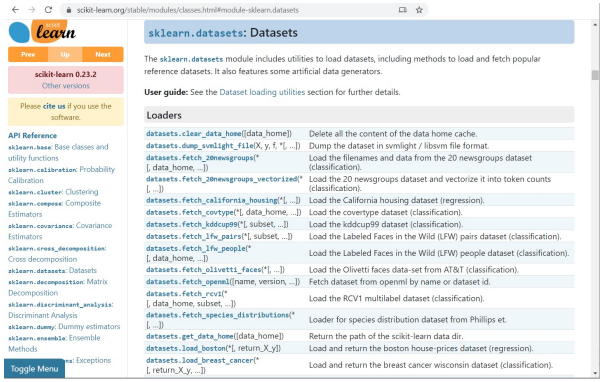
- Load a data set from a local file:

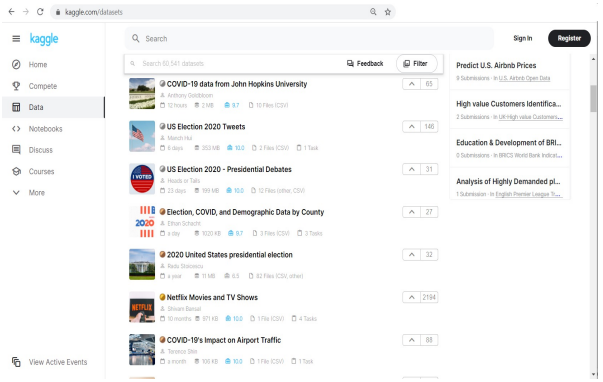
```
dataset = pd.read_csv('iris.csv')
```

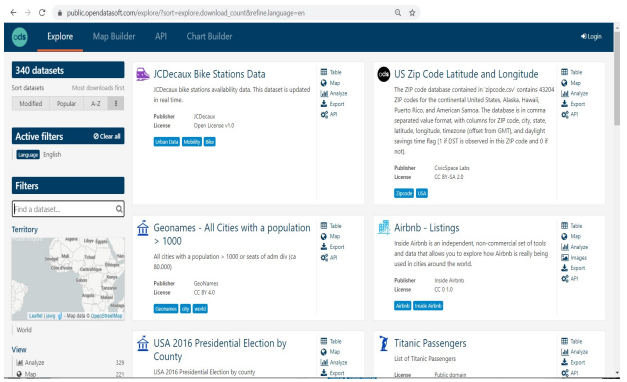
- Load a "reference" data set from a library:

```
dataset = sklearn.datasets.load_iris()
```

- Several bases of data sets:



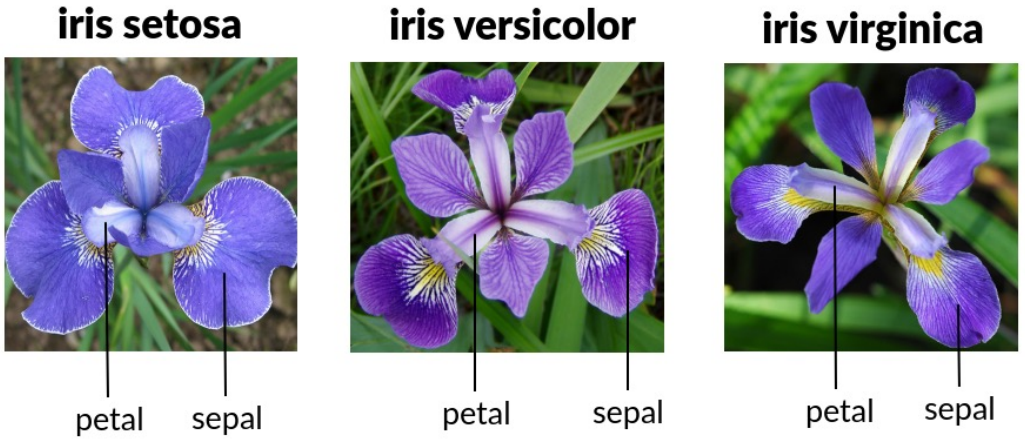




EXPLORE THE DATA SET

Example dataset

- Iris data set: characteristics of three species of iris.

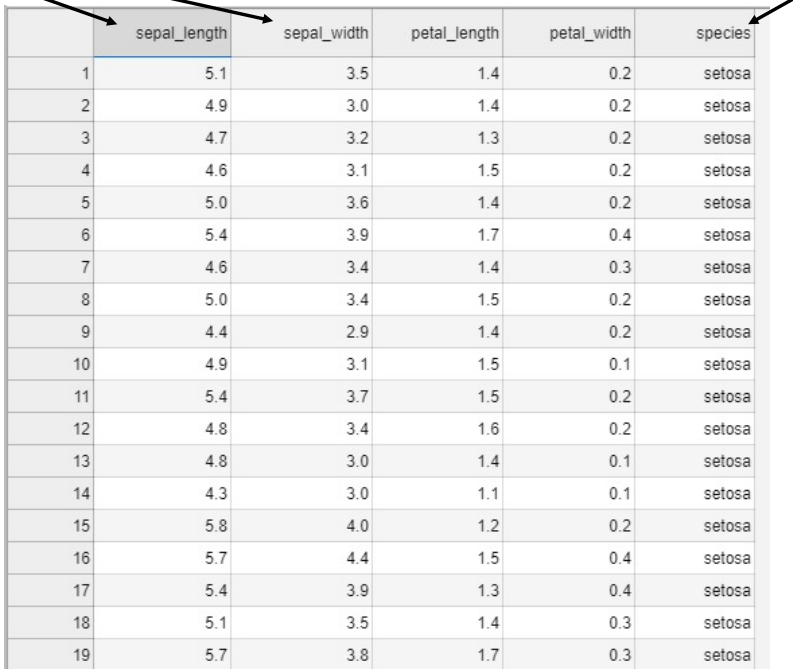


EXPLORE THE DATA SET

Load

Features/
independent variables

Label/ target value /
dependent variable



	sepal_length	sepal_width	petal_length	petal_width	species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa

EXPLORE THE DATA SET

Get general information

- Information on format:

```
dataset.shape
dataset.columns.values
dataset.info()
dataset.dtypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

- Information on content:

```
dataset.head()
dataset.tail()
dataset.sample(n=7)
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

EXPLORE THE DATA SET

Basic operations

- Conversion:

```
pd.DataFrame(data=dataset, index=…, columns=…) # array to DataFrame
df.to_numpy()    # DataFrame to array
```

- Selecting:

```
df['feature']  # column
df[0:3]        # rows
df[df['feature']>0]  # rows that satisfy a condition
```

- Manipulation:

```
df_copy = df.copy()
df['new_feature'] = ['high' if x>10 else 'low' for x in df['feature']]
df['feature'].values.reshape(-1,1)
```

EXPLORE THE DATA SET

Compute basic statistics

- Count values:

```
dataset.isna().sum()
dataset.duplicated().sum()
dataset.nunique() #count unique elements
(dataset['species']=='setosa').sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
```

- Mean, min/max, standard deviation, etc.:

```
dataset.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

EXPLORE THE DATA SET

DataFrame vs array

DataFrame (Pandas)

```
dataset
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
type(dataset)
```

```
pandas.core.frame.DataFrame
```

```
dataset['sepal_width'][0]
```

```
3.5
```

```
dataset.iloc[0]
dataset.iloc[0,:]
```

```
sepal_length       5.1
sepal_width        3.5
petal_length       1.4
petal_width        0.2
species         setosa
Name: 0, dtype: object
```

```
dataset.iloc[0][1]
```

```
3.5
```

Array (NumPy)

```
dataset.to_numpy()
dataset.values
```

```
array([[5.1, 3.5, 1.4, 0.2, 'setosa'],
       [4.9, 3.0, 1.4, 0.2, 'setosa'],
       [4.7, 3.2, 1.3, 0.2, 'setosa'],
       [4.6, 3.1, 1.5, 0.2, 'setosa'],
       [5.0, 3.6, 1.4, 0.2, 'setosa'],
       [5.4, 3.9, 1.7, 0.4, 'setosa'],
       [4.6, 3.4, 1.4, 0.3, 'setosa'],
       [5.0, 3.4, 1.5, 0.2, 'setosa'],
       [4.4, 2.9, 1.4, 0.2, 'setosa'],
       [4.9, 3.1, 1.5, 0.1, 'setosa'],
       [5.4, 3.7, 1.5, 0.2, 'setosa'],
       [4.8, 3.4, 1.6, 0.2, 'setosa'],
       [4.8, 3.0, 1.4, 0.1, 'setosa'],
       [4.3, 3.0, 1.1, 0.1, 'setosa'],
       [5.8, 4.0, 1.2, 0.2, 'setosa'],
       [5.7, 4.4, 1.5, 0.4, 'setosa'],
```

```
type(dataset.values)
```

```
numpy.ndarray
```

```
dataset.values[0][1]
dataset.values[0,1]
```

```
3.5
```

```
type(dataset.values[0,1])
```

```
float
```

```
dataset.values[0,4]
```

```
'setosa'
```

```
type(dataset.values[0,4])
```

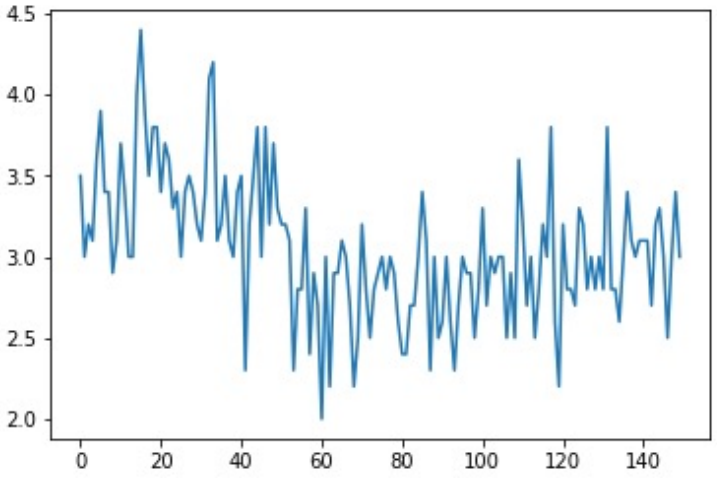
```
str
```

EXPLORE THE DATA SET

Visualize

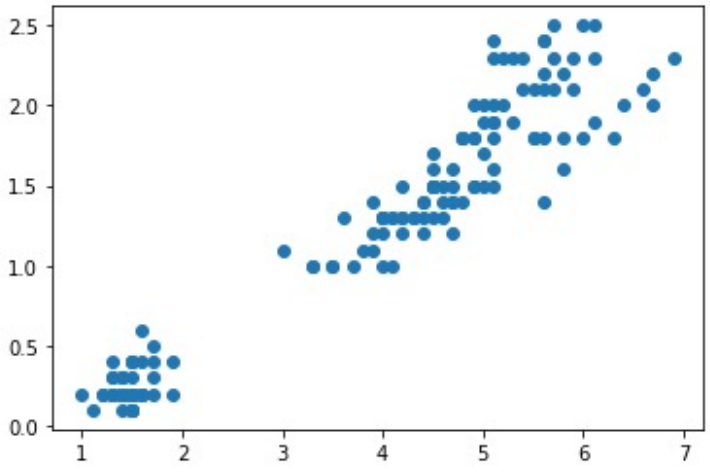
- Curve plot:

```
plt.plot(dataset['sepal_width'])
plt.show()
```



- Scatter plot:

```
plt.scatter(dataset['petal_length'],
   dataset['petal_width'])
```

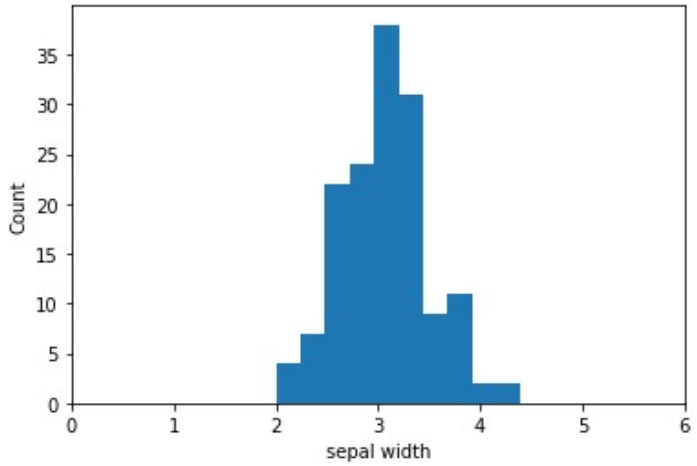


EXPLORE THE DATA SET

Visualize

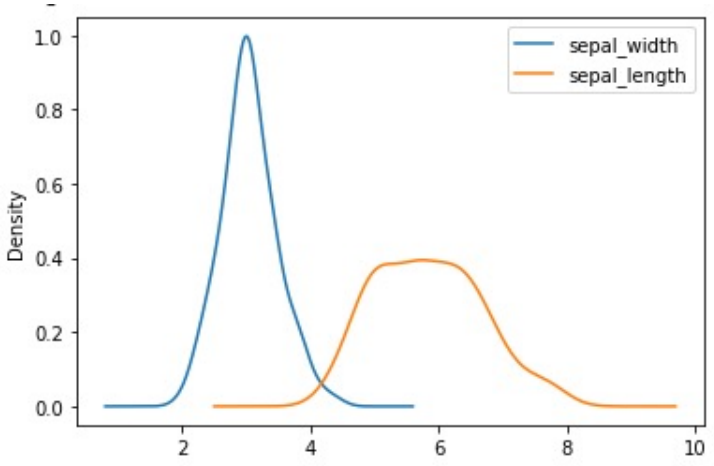
- Histogram:

```
plt.hist(dataset['sepal_width'])
```



- Density plot:

```
dataset[['sepal_width',
   'sepal_length']].plot(kind='density')
```

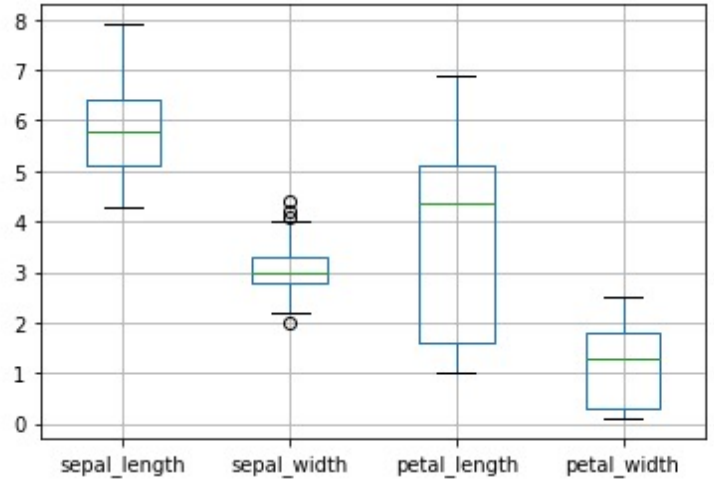


EXPLORE THE DATA SET

Visualize

- Boxplot:

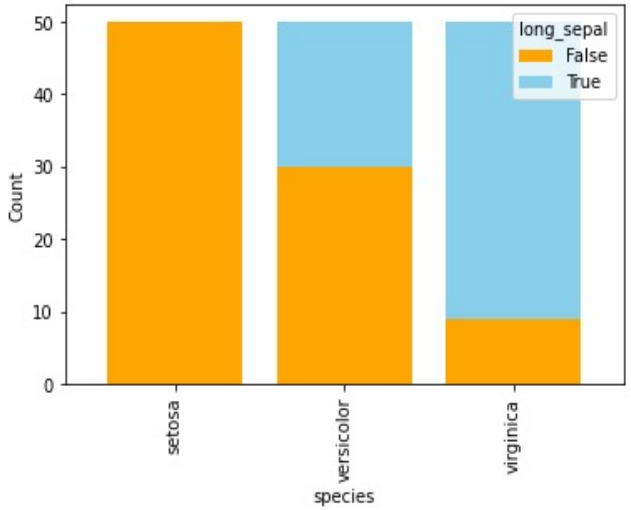
```
dataset[['sepal_length', 'sepal_width',
   'petal_length', 'petal_width']].boxplot()
```



- Stacked bars:

```
df_plot =
   dataset.groupby(['species','long_sepal']).size()\
   .reset_index()\
   .pivot(index='species', columns='long_sepal', values=0)

df_plot.plot(kind='bar', stacked=True,
   color=['orange', 'skyblue'], width=0.8)
```

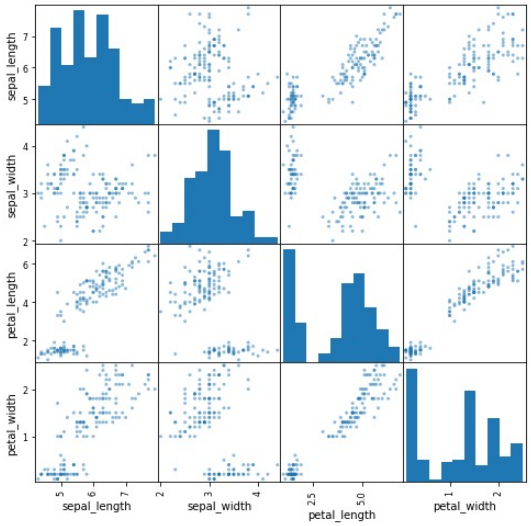


EXPLORE THE DATA SET

Visualize

- Scatter matrix:

```
scatter_matrix(dataset.loc[:,:4])
```

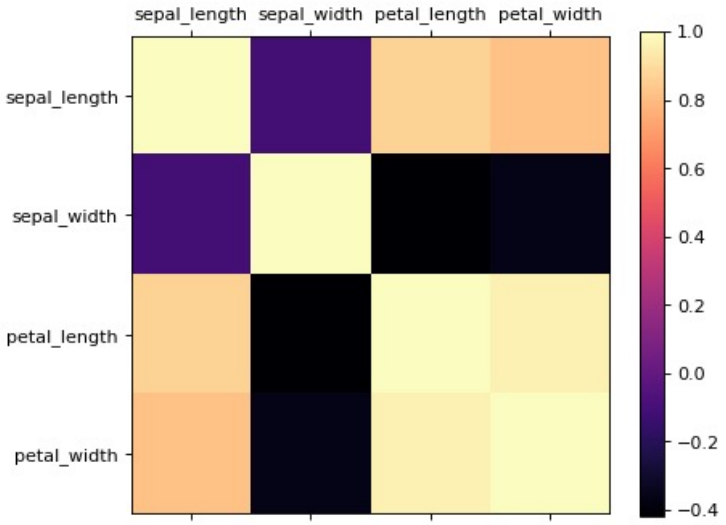


- Correlation matrix:

```
correlation = dataset.loc[:,:4].corr()
matshow(correlation)
```

alternative

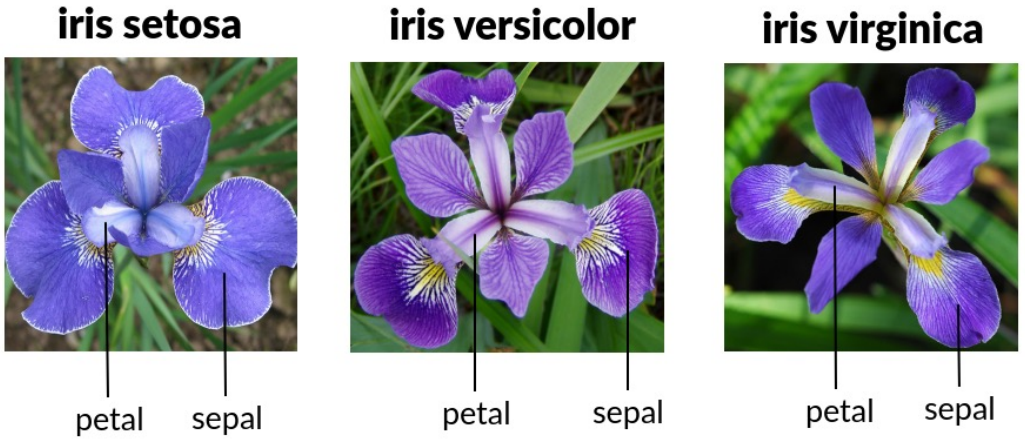
```
import seaborn as sns
sns.heatmap(correlations, annot=True)
```



EXPLORE THE DATA SET

Implementation

- Iris data set: characteristics of three species of iris.
- Objectives:
  - Explore the data
  - Work with Jupyter notebook



EXPLORE THE DATA SET

Practice

- Titanic data set: “what sorts of people were more likely to survive?”
- Guidelines:
  - Data set is provided
  - Work with Jupyter notebook
  - Start during the course, finish at home
  - Report should be submitted before Monday 23:59 PM on Teams

EXPLORE THE DATA SET

Practical exercise

Advice for the assignment:

- Submit an executed Jupyter notebook
- Always check out the documentation of the libraries you use
- Don’t do copy-paste from others’ work (or from anywhere else)
- Always write code that you understand (you may be asked to demonstrate your understanding of the code you submitted)
- Provide written explanations (not only numbers) when an interpretation is needed