Applied Data Science

Project

Presented By: Istvan Lengyel

So what is Scikit-learn

Scikit-learn is a Python library for Machine Learning. It provides many useful tools for Supervised Learning and Unsupervised Learning, which was covered in our previous module.

The library is built on Scipy, Numpy, and matplotlib and can easily be installed as follows.

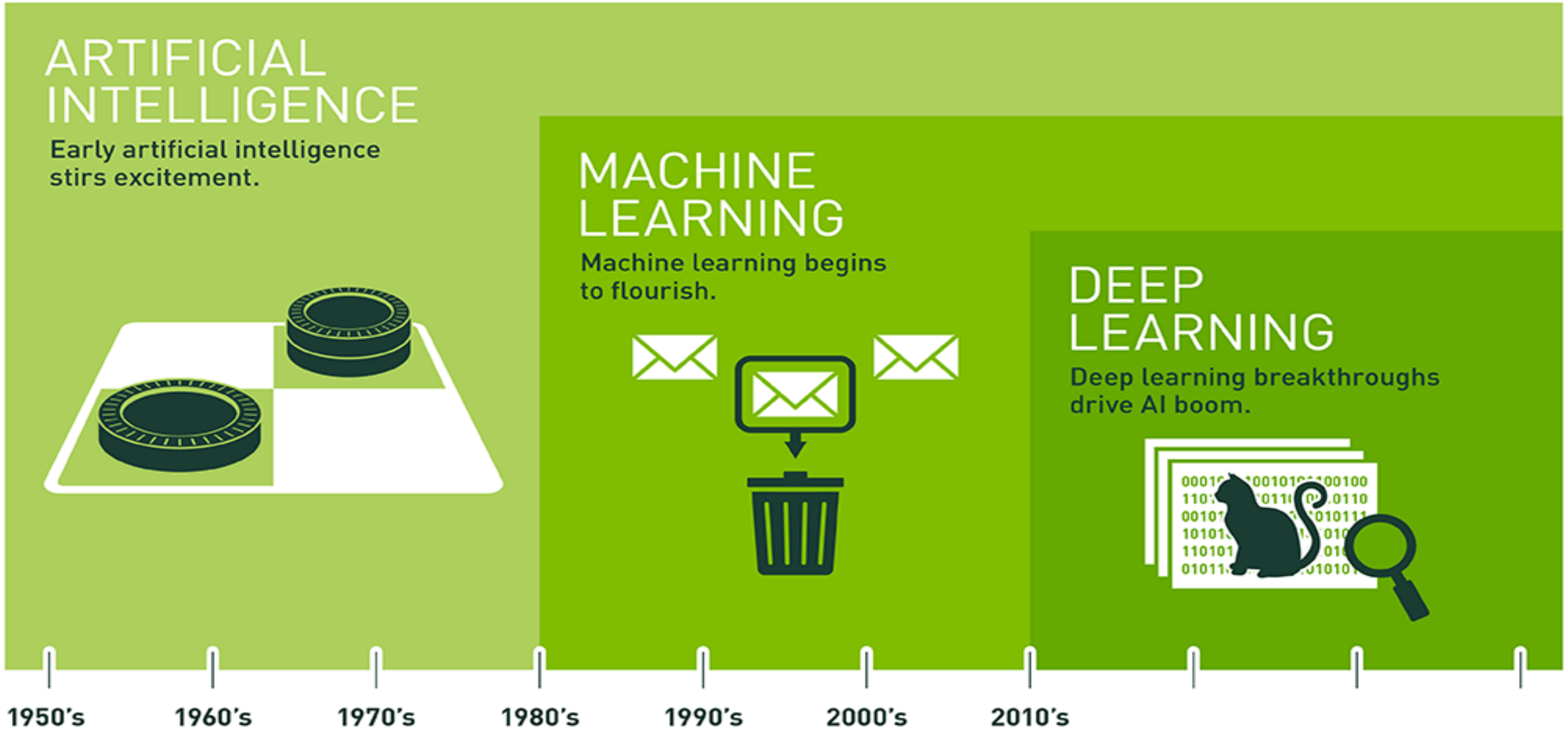
Windows:

```
pip3 install –U scikit–learn
```

Ubunutu:

```
pip3 install –U scikit–learn
```

Reminder Machine Learning is a branch of Artificial Intelligence, as is Deep Learning



ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.

MACHINE LEARNING

Machine learning begins to flourish.

DEEP LEARNING

Deep learning breakthroughs drive AI boom.

Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

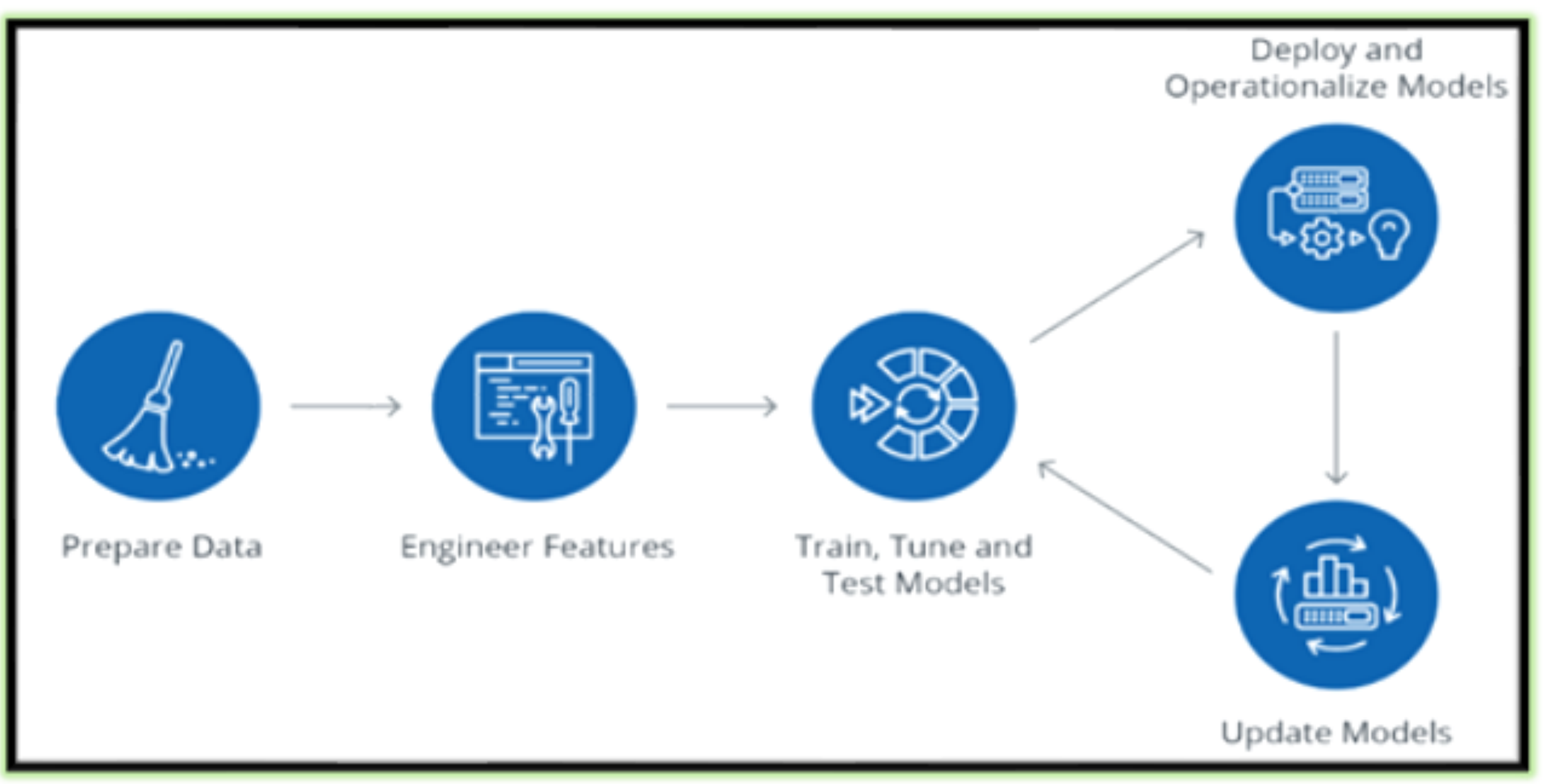
Image Sourced: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep

Definition

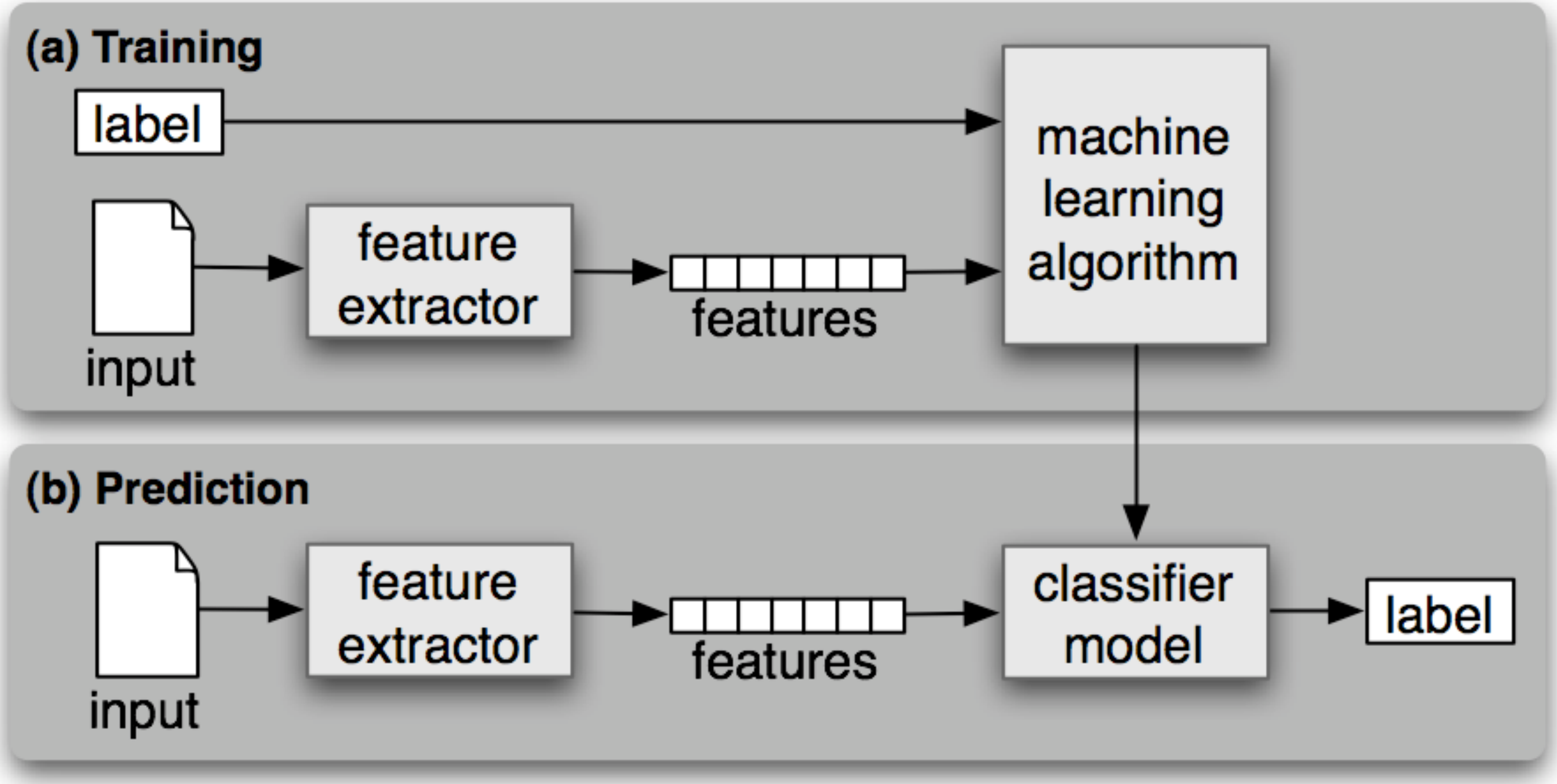
- Model
  - The collection of **parameters** that you are trying to **fit.**
- Data
  - What you are using to **fit** the **Model.**
- Target
  - The value you are trying to **predict** with your **Model.**
- Features
  - Attributes of your **Data** that will be used in **prediction**
- Methods
  - Algorithms that will use your **Data** to **fit** a **Model**

The Process at a high level

- Prepare our data
- Engineer features
- Train, Tune and Test our Model
- Deploy our Model
- Update our Model

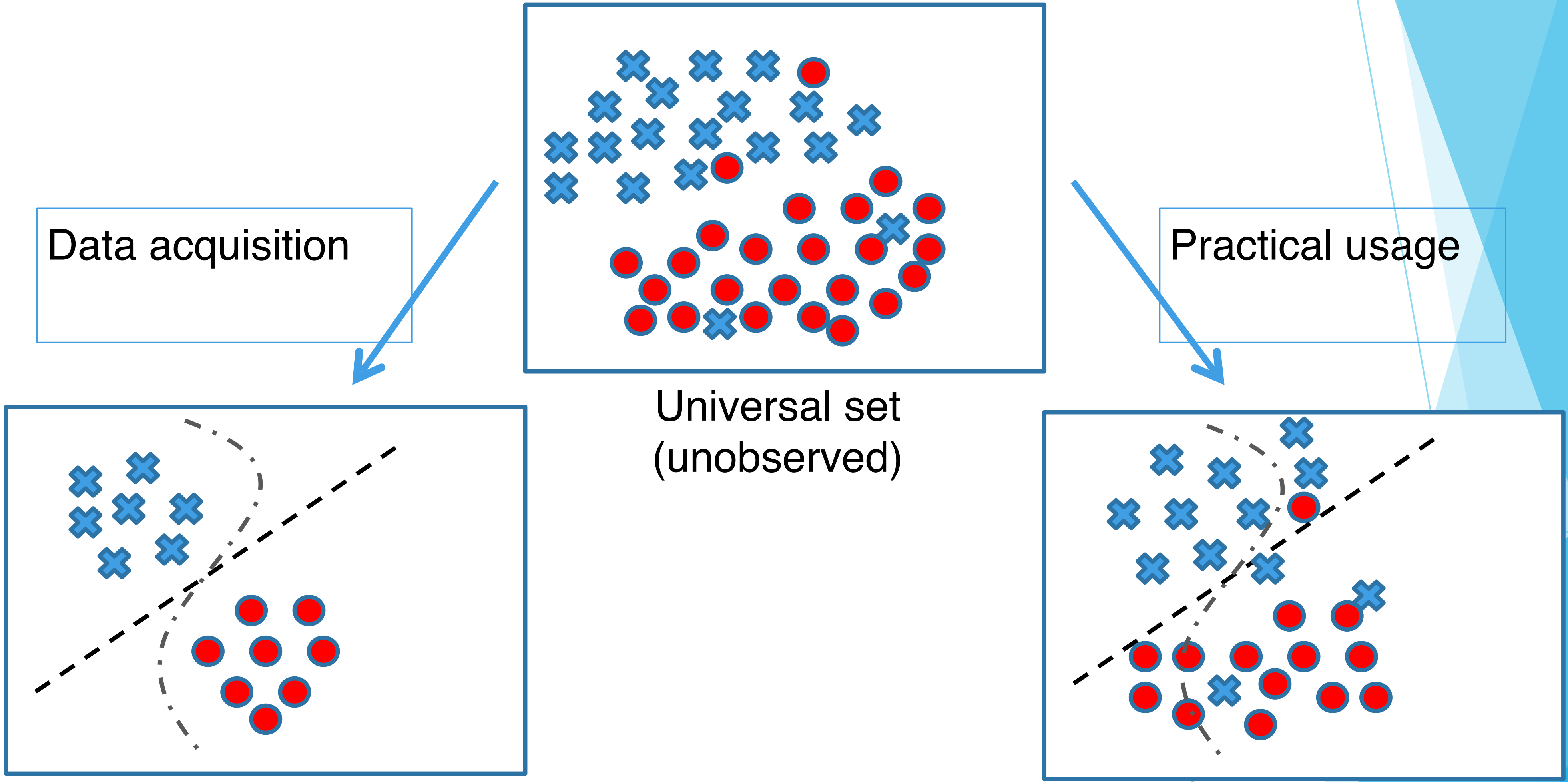


Method



Source: Google.com

Training vs. Testing



Data acquisition

Practical usage

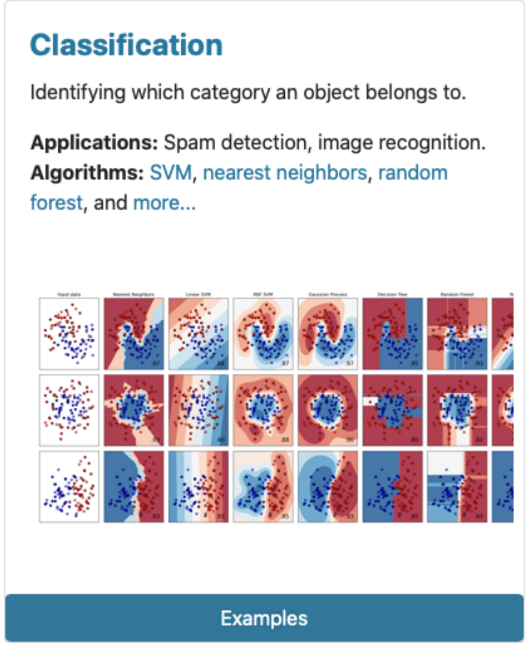
Universal set
(unobserved)

Training set
(observed)

Testing set
(unobserved)

Scikit-learn Algorithms

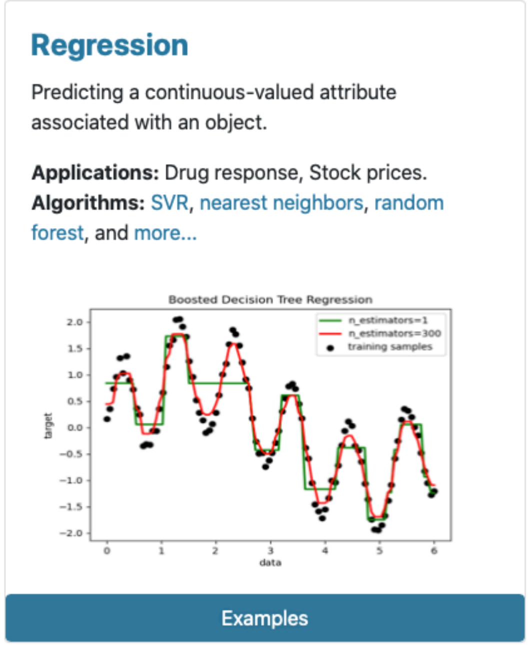
- We discussed a few algorithms last week for Supervised Learning, Unsupervised Learning and Reinforcement Learning. To get an extensive list of what Scikit-learn can offer, visit the following link:
  - https://scikit-learn.org/stable/user_guide.html
  - https://scikit-learn.org/stable/



Classification

Identifying which category an object belongs to.

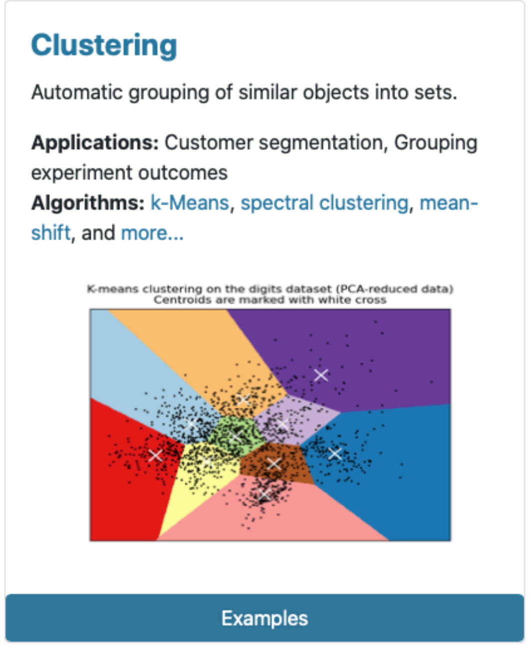
Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...



Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...



Clustering

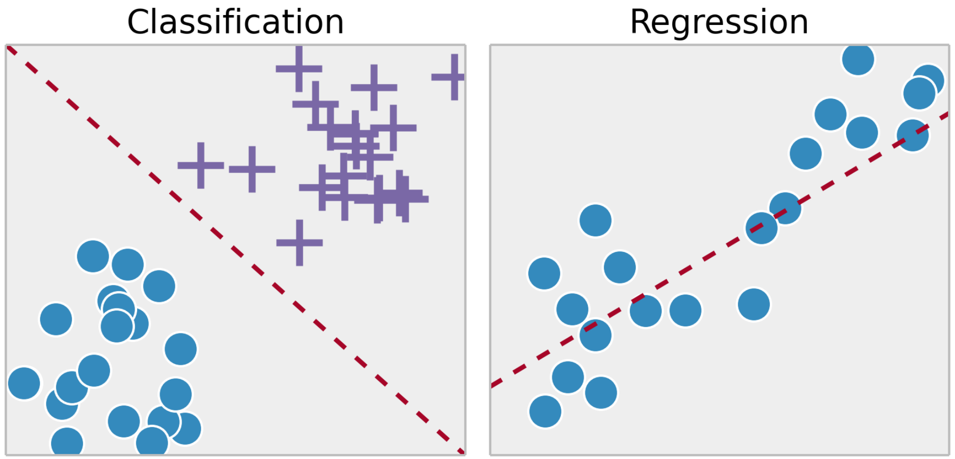
Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

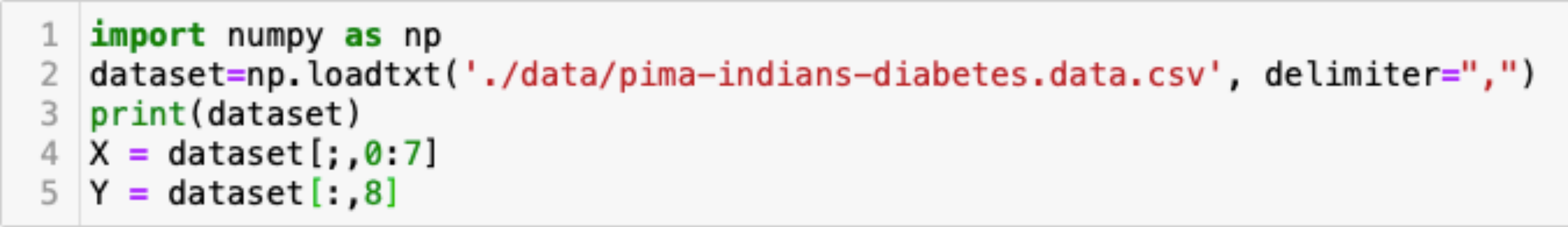
K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross
Examples

Supervised Learning

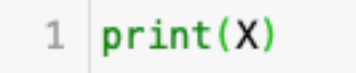
- All supervised estimators in scikit-learn implement a **fit(X, y)** method to fit the model and a **predict(X)** method that, given un-labeled observations **X**, returns the predicted labels of **y**.
- **Classification**: classify the observations in a set of finite labels
- **Regression**: If the goal is to predict a continuous target variable



Loading our data

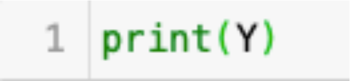


```python
import numpy as np
dataset=np.loadtxt('./data/pima-indians-diabetes.data.csv', delimiter=",")
print(dataset)
X = dataset[;,0:7]
Y = dataset[:,8]
```



```python
print(X)
```

Features



```python
print(Y)
```

Target

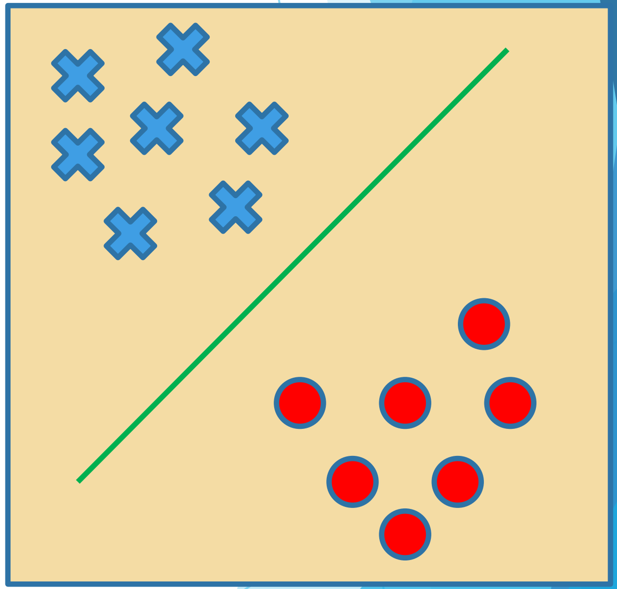
Normalising and standardising our data

- We use the **preproccessing** library from **sklearn** for **normalising** and **standardising** our **data**.
- The majority of models are highly sensitive to data scaling. Prior to running an algorithm, normalisation and standardisation should be performed
- **Normalization** involves replacing nominal features, so that each of them would be in a range from 0 to 1
- **Standardization** involves data **preproccessing**, after which each feature has an average 0 to 1 dispersion(all data does not follow normal distribution)

add code

Linear Classifier

- We can use the following:
  - Logistic regression
  - SVM



Logistic Regression

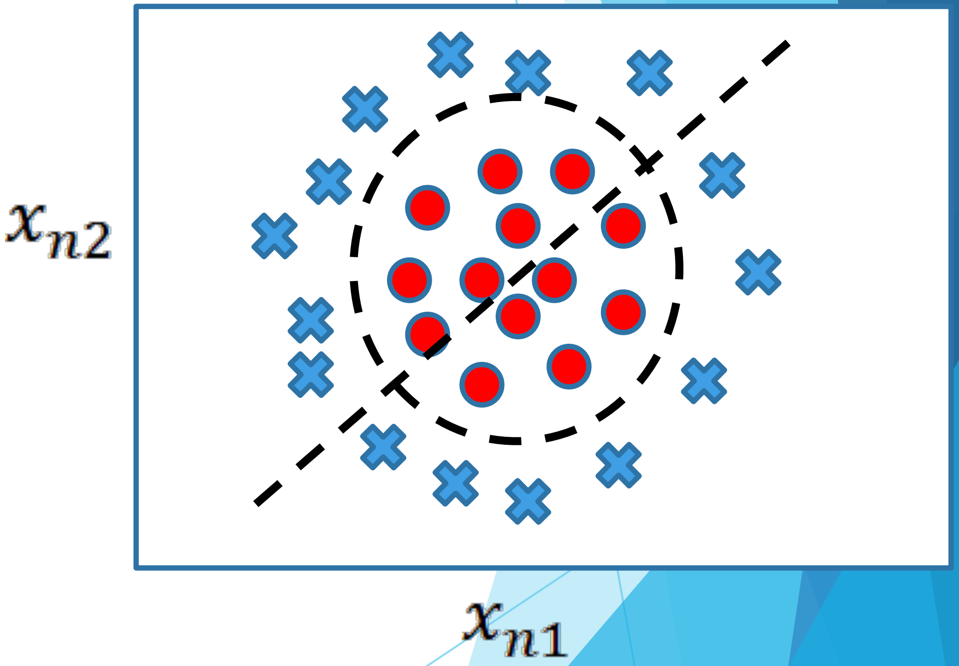
- **Logistic Regression:** Most often used for solving tasks of classification (binary), but multi-sclass classification (the so-called one-vs-arest method) is also allowed.

Support Vector Machine

- **Support Vector Machines:** one of the most popular machine learning algorithms used mainly for the classification problem.

- SVM allows multi-class classification with the help of the one-vs-all method.

- Non-linear case



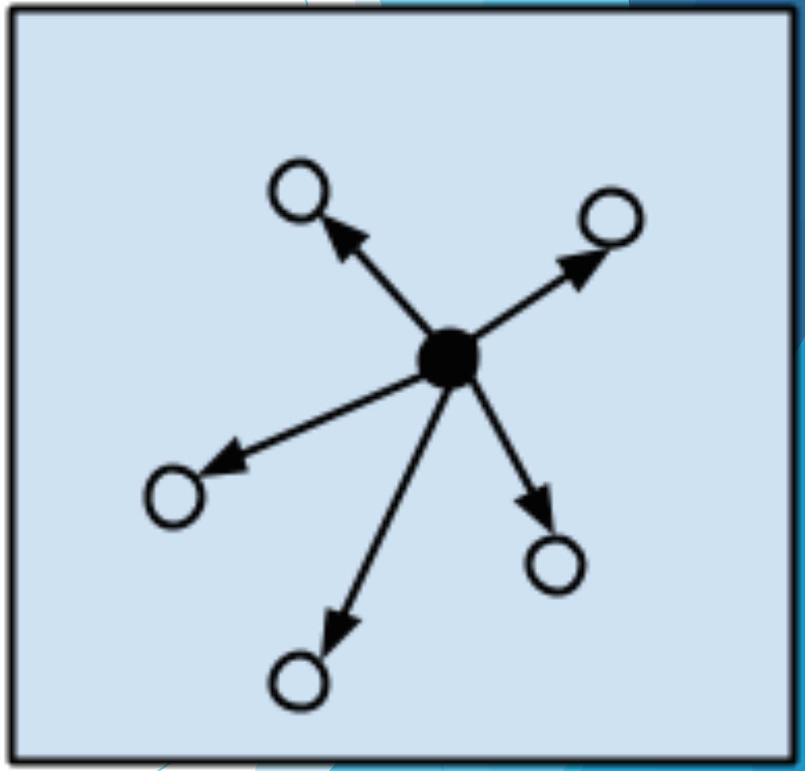
Naive Bayes

- Naive Bayes: One of the most well-known machine learning algorithms.
  - Graphical algorithm that encodes the joint probability distribution of a data set using Bays theorem.
  - Captures probabilistic relationships between variables
  - Based on probability that instances (data) belong in each category
- This method often provides good quality in multi-class classification problems.

K-Nearest Neighbours

- k-Nearest Neighbours: often used as part of a more complex classification algorithm.

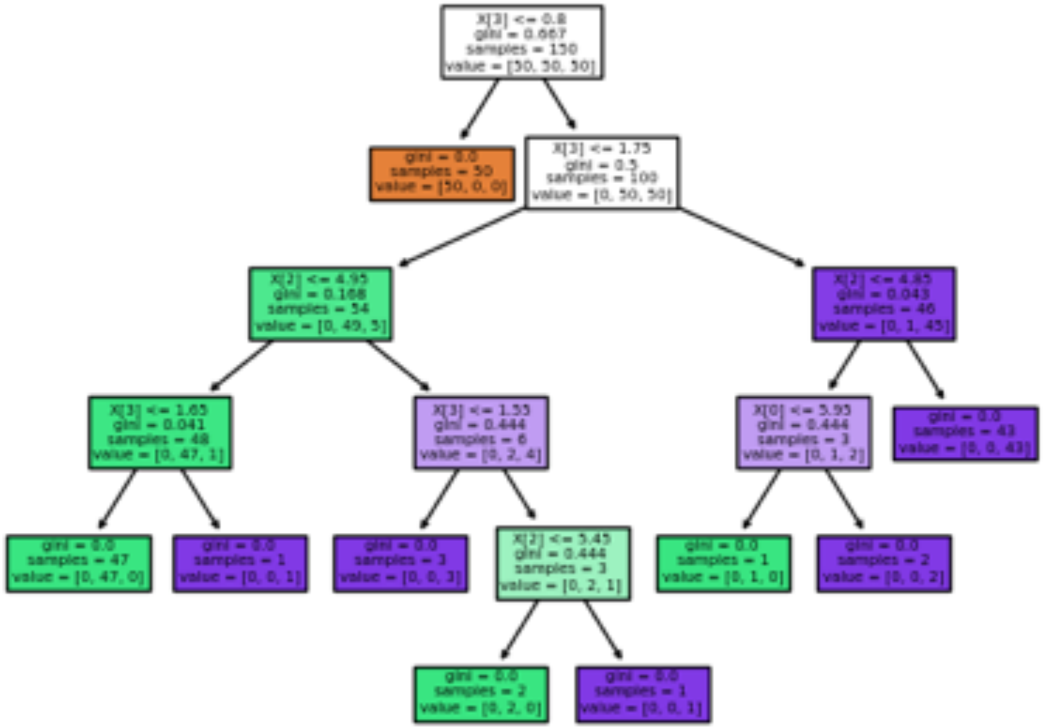
- When parameters (metrics mostly) are set well, the algorithm often gives good quality also in regression problems.



Decision Trees

- Decision Trees: often used in problems having category features

- Used for regression and classification problems. The trees are very well suited for multi-class classification.


X[3] <= 0.8
gini = 0.667
samples = 150
value = [50, 50, 50]
gini = 0.0
samples = 50
value = [50, 0, 0]
X[3] <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
X[2] <= 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]
X[2] <= 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]
X[3] <= 1.65
gini = 0.041
samples = 48
value = [0, 47, 1]
X[3] <= 1.55
gini = 0.444
samples = 6
value = [0, 2, 4]
X[0] <= 5.95
gini = 0.444
samples = 3
value = [0, 1, 2]
gini = 0.0
samples = 43
value = [0, 0, 43]
gini = 0.0
samples = 47
value = [0, 47, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 3
value = [0, 0, 3]
X[2] <= 5.45
gini = 0.444
samples = 3
value = [0, 2, 1]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 2
value = [0, 0, 2]
gini = 0.0
samples = 2
value = [0, 2, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]