From DATA to AI
Machine Learning

IoT Sensor-AI

Petteri Mo 2022

Dataset-Big Data

- Traditional Data-processing application S.w.
- Large or complex
- Big Data analysis-Machine Learning
    A. capturing, storage, analysis
    B. searching, sharing, transferring
    C. visualisation, querying, privacy
    D. Data source
- Machine learning is set of AI

Sensors -IoT-Machine Learning-AI

Sensors

- Perceive internal and external environment
- Obtain feedback about how actions affect the environment
- Input peripherals provide sensory signals

property

- sensitive

A temperature sensor should react to changes in temperature

- insensitive

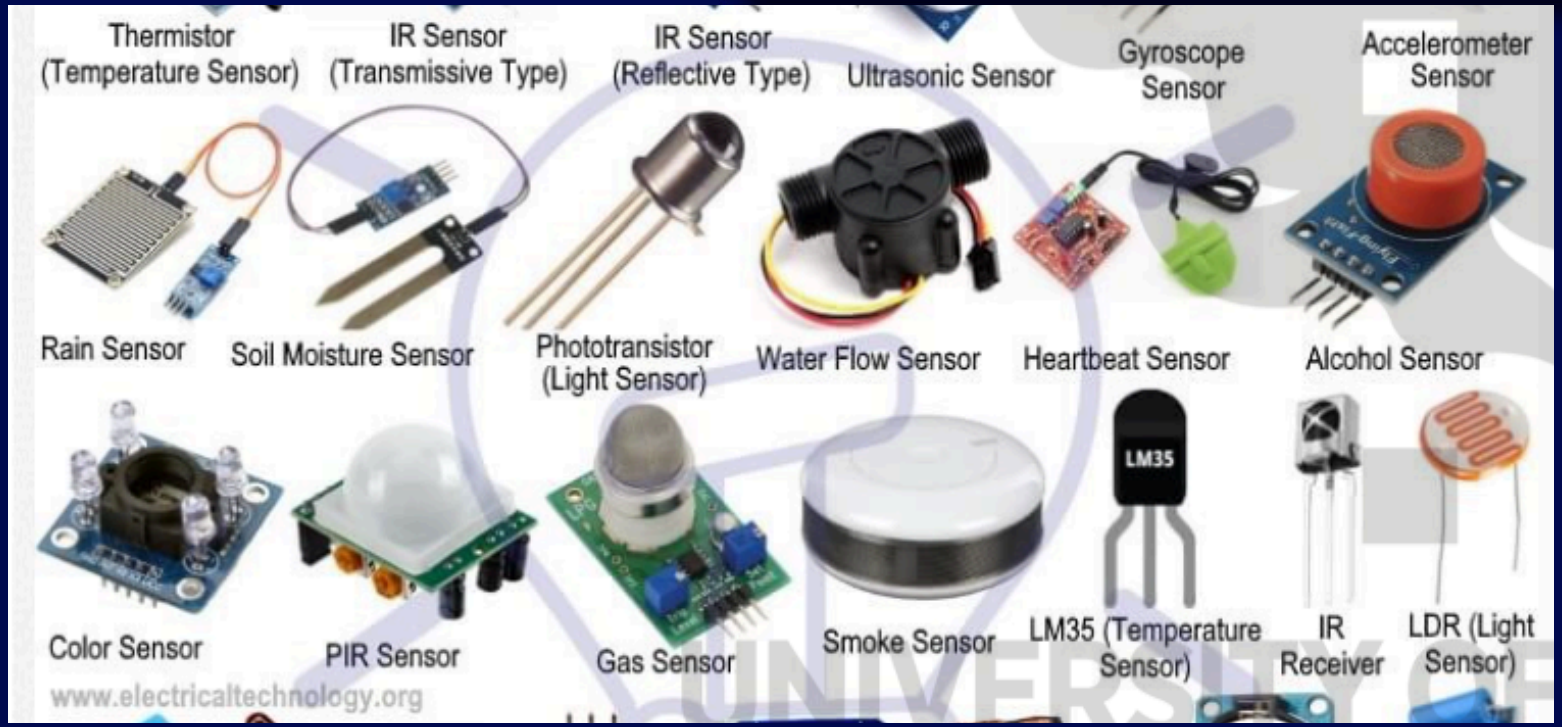
Air pollution sensor should not depend on air flow speed

- no influence on the property measuring

Temperature sensor should not heat up

.

Example of Sensors



Reference: Helsinki Yliopisto Petteri Nurmi

Sensors -IoT-Machine Learning-AI

Sensors by Industrial fields

- Medical
- Sciences
- Business

property

- Medical

pressure, force, airflow, oxygen, pulse oximetry, temperature,

- Sciences

Solar cells and photographic film detect light.

Microphones detect sound.

Thermometers sense the change in heat or temperature.

Pressure gauges detect touch.

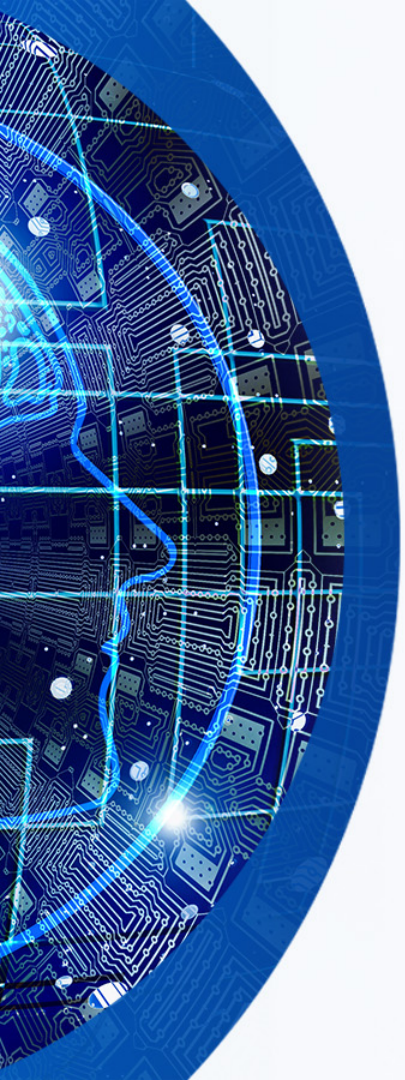
Scales detect and measure the effect of gravity.

There are many devices that detect various chemicals and even odour

A magnetometer detects magnetic fields.

An electric meter detects electricity.

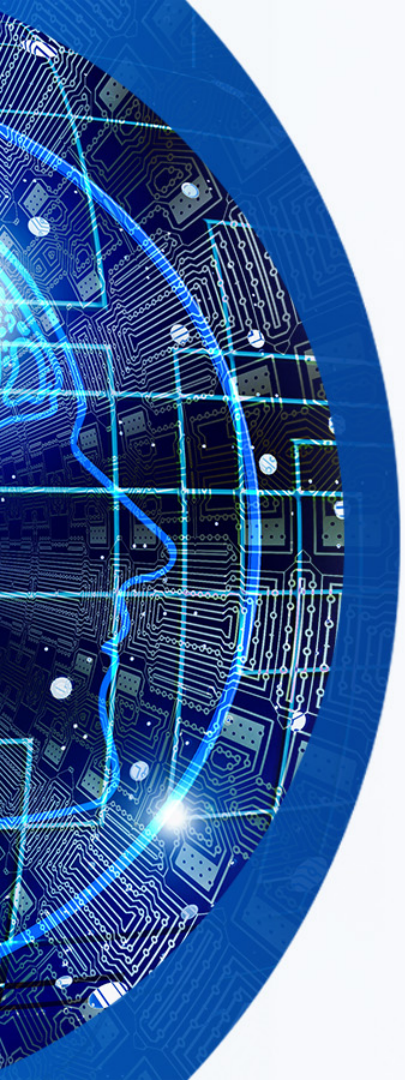
A Geiger counter detects atomic radiation



Machine Learning

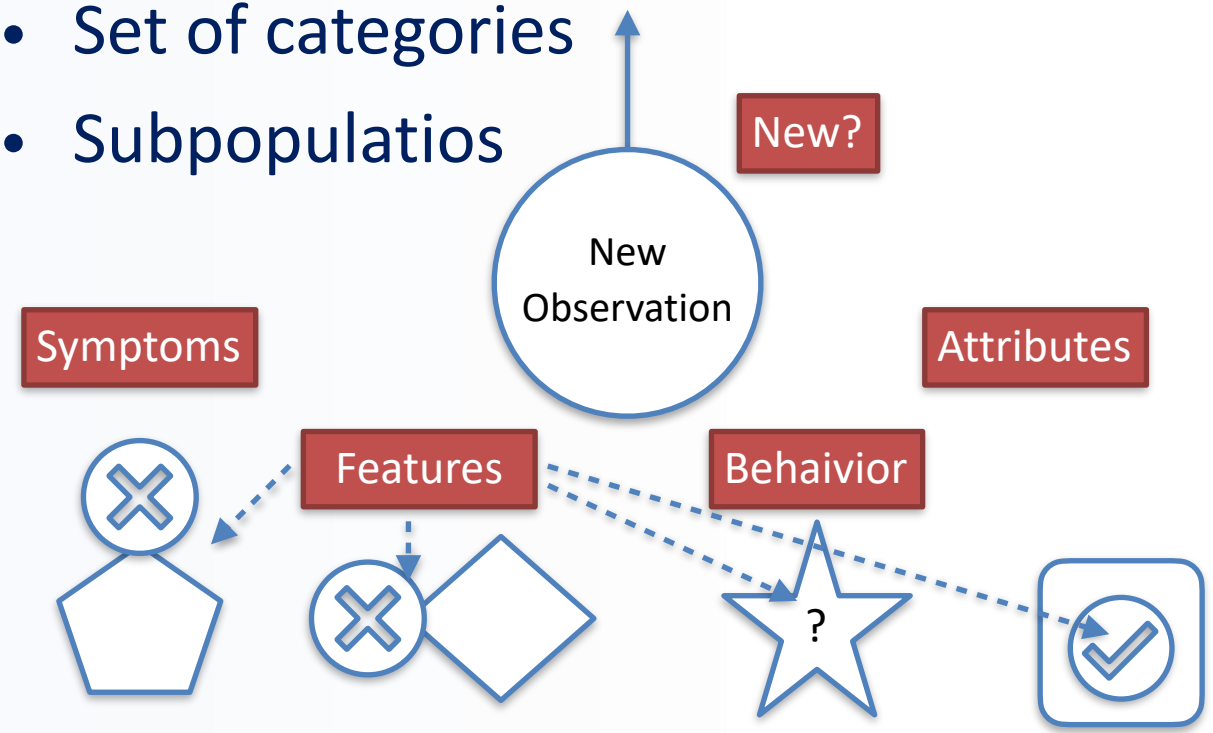
- Supervised learning
- Unsupervised learning
- Reinforcement learning

- Deep learning
- Dataset
- Models &. Algorithm

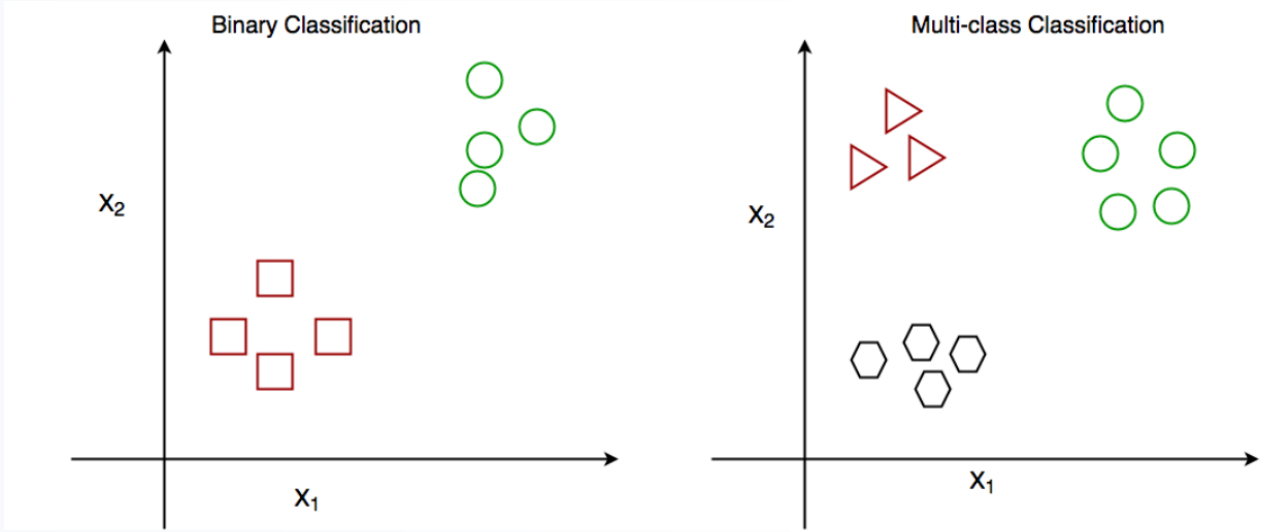


Classification-ML/Statistics

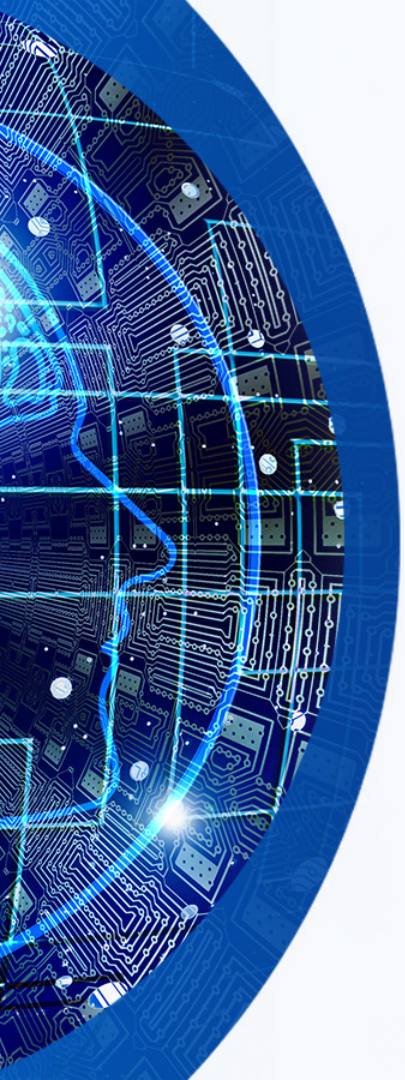
- Set of categories
- Subpopulatios



Types of classification



x1 and x2 are the variables upon which class is predicted



Types of classification

- Binary Classification
  - Given data into 2 distinct classes
  - Health conditions: determine if the person has a certain disease 1/not 0
- Multiclass Classification, classes more than 2

Sensors -IoT-Machine Learning-AI

ML models

- Train the model: classifier on this dataset

Types of Classifiers(Algorithms)

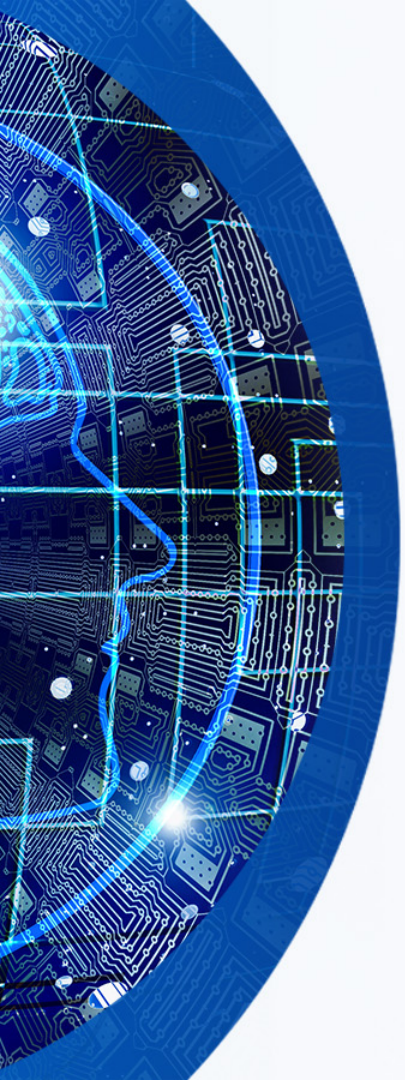
- Linear Classifiers: Logistic Regression
- Tree-Based Classifiers: Decision Tree Classifier
- Support Vector Machines
- Artificial Neural Networks
- Bayesian Regression
- Gaussian Naive Bayes Classifiers
- Stochastic Gradient Descent (SGD) Classifier
- Ensemble Methods: Random Forests, AdaBoost, Bagging Classifier, Voting Classifier, ExtraTrees Classifier

Sensors -IoT-Machine Learning-AI

Where we are

Practical Applications of classification

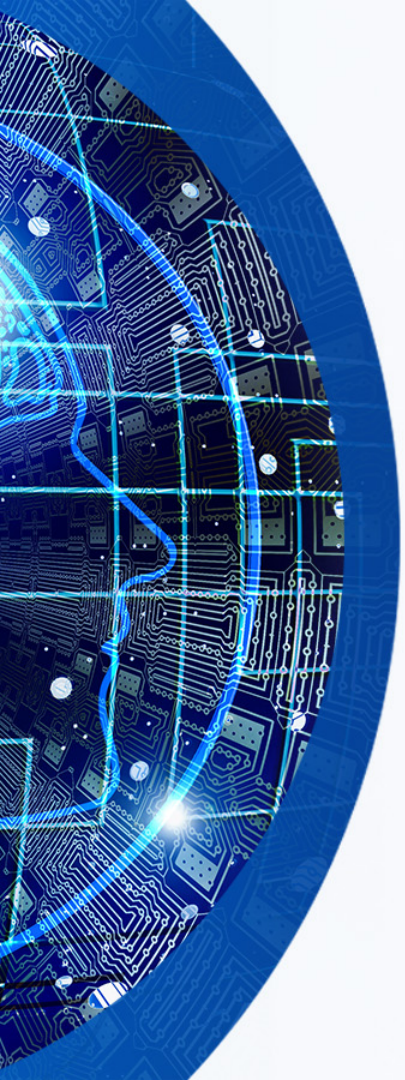
1. Google's self-driving car uses deep learning-enabled classification techniques which enables it to detect and classify obstacles.
2. Spam E-mail filtering is one of the most widespread and well-recognized uses of Classification techniques.
3. Detecting Health Problems, Facial Recognition, Speech Recognition, Object Detection, and Sentiment Analysis all use Classification at their core.



Implementation &. Example

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import datasets
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB

iris = datasets.load_iris()
X = iris.data
y = iris.target
# Splitting X and y into training and testing datasets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state= 1)
#Gaussian Naive Bayes
gnb = GaussianNB()
#Train the model
gnb.fit(X_train, y_train)
```



Implementation &. Example

```
#Make predictions
gnb_pred = gnb.predict(X_test)
#Print the accuracy
print("Accuracy of Gaussian Naive Bayes:", accuracy_score(y_test, gnb_pred)*100)
```

```
# decision tree classifier
dt = DecisionTreeClassifier(random_state=0)
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)
print("Accuracy of Decision Tree Classifier:", accuracy_score(y_test,
dt_pred)*100)
```

```
#SVM classifier
svm_clf = svm.SVC(kernel='linear') # linear kernel
svm_clf.fit(X_train, y_train)
#make prediction
svm_clf_pred = svm_clf.predict(X_test)
#print the accuracy of SVM classifier
print("Accuracy of SVM Classifier:", accuracy_score(y_test, svm_clf_pred)*100)
```

Hands on experience

How classification works

- Python
- Scikit-learn provides all the classifiers
- Pandas for data i/o
- Standard dataset(Iris)

Practical Applications of classification

1. python3.9 /Users/petterimo/Data/irisClassification.py
2. Accuracy of Gaussian Naive Bayes: 96.66666666666667
3. Accuracy of Decision Tree Classifier: 96.66666666666667
4. Accuracy of SVM Classifier: 100.0

6. Process finished with exit code 0