CS-EJ3211 Machine Learning with Python

Exercise Session 1

Shamsi Abdurakhmanova

Aalto University
FITech

31.05.23

Table of Contents

1. Course Logistics

2. Machine Learning: Definitions

3. Machine Learning: Components

4. Machine Learning: Implementation

Table of Contents

Teaching staff

- Shamsi Abdurakhmanova shamsiiat.abdurakhmanova@aalto.fi
- Alexander Pavlyuk alexander.pavlyuk@aalto.fi
- Alex Jung alex.jung@aalto.fi

Course info & materials

- Course home page at MyCourses https://mycourses.aalto.fi/course/view.php?id=38429
- Course email cs-ej3211@aalto.fi
- Assignments and timetables see on MyCourses/Notebooks and MyCourses/ML Project
- Main material is jupyter notebooks (text + code)
- Course book by Alex Jung "Machine Learning: The Basics" http://mlbook.cs.aalto.fi/

Course structure

- No lectures
- Jupyter notebook each week on Mon (DL next Mon), 6 notebooks altogether
- Exercise sessions on Wed (14-15.00)
- ML Project peer-graded assignments (starts from 5.6)
- communication via email cs-ej3211@aalto.fi and Slack

Grading

- Notebooks: 6 notebooks x 10 points max = 60 points max (30p min to pass)
- ML project: 3 stages, 40 points max (20p min pass). Project includes submission and peer-grading part
- Grade bounds: 1: 50p, 2: 60p, 3: 70p, 4: 80p, 5: 90p
- Post re-grading requests at Slack #grading or email cs-ej3211@aalto.fi
- No exam
- Bonus tasks: TBA

Course implementation

- Notebooks: on jupyter hub https://jupyter.cs.aalto.fi/
  - notebooks are autograded after DL
  - feedback released 1-2 day after DL on jupyter hub (see MyCourses/FAQ and How to use Jupyter Hub pdf at MyCourses/Notebooks)
  - you can ask solutions review at Slack #grading or email cs-ej3211@aalto.fi
- ML project: at MyCourses/ML Project
- Exercise sessions: on Zoom

Course feedback

- General feedback - at the end of the course (mandatory).
- Specific feedback for each notebook at MyCourses/Notebooks.
- Feedbacks are anonymous.

Ground rules

- No plagiarism
- No bullying, harassment
- See Aalto Code of conduct https://www.aalto.fi/en/aalto-university/code-of-conduct-values-into-practice

Course logistics - Questions?

Table of Contents

1 Course Logistics

2 Machine Learning: Definitions

3 Machine Learning: Components

4 Machine Learning: Implementation

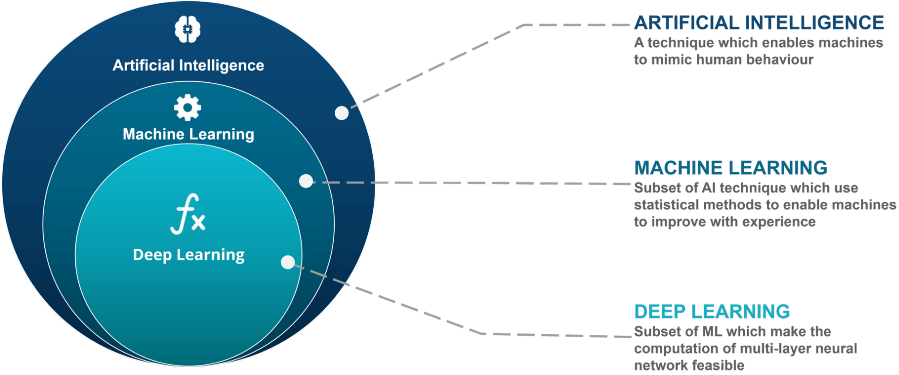
Machine Learning - Definition

> “Machine learning is the study of computer algorithms that improve automatically through experience.”
>
> — Wiki

> “Machine learning is a field of computer science that aims to teach computers how to learn and act without being explicitly programmed.”
>
> — deepai.org

> “Machine learning fits fancy mathematical models to data.”
>
> — A.Jung

Machine Learning - Relation to other fields



ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

image source

Machine Learning - Applications

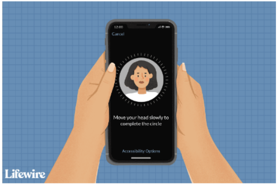
grammarly

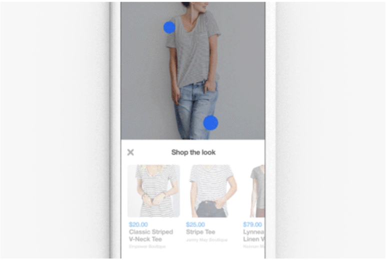
amazon

Google

Spotify

NETFLIX





Machine Learning - Applications

- Natural Language Processing
- Insurance Claim Analysis
- Bioinformatics and Medical Diagnosis
- Image Processing and Pattern Recognition
- Search Engines
- Financial Market Analysis

from deepai.org

Machine Learning - Why and When?

Why:

- Automation (make process faster and cheaper)
- Accuracy, reliability, consistency, unbias (do the task better than humans)
- Discovery (explore the data to get new insights or find new relations)

When:

- There is a complex pattern to learn
- There is a (lots of) data exist

Table of Contents

1 Course Logistics

2 Machine Learning: Definitions

3 Machine Learning: Components

4 Machine Learning: Implementation

Components of ML

Goal: Formulate a real-life problem as a Machine Learning problem.

Components of ML

Goal: Formulate a real-life problem as a Machine Learning problem.

Solution: Decompose a problem into 3 components

- **Data**
- **Model (Hypothesis space)**
- **Loss**

Data

Data is a collection of data points.

A data point is any object that conveys information. Data points might be students, radio signals, trees, forests, images, RVs, real numbers or proteins. We characterize data points using two types of properties: features and labels.

Features are properties of a data point that can be measured or computed in an automated fashion.

The label of a data point represents a higher-level facts or quantities of interest. In contrast to features, determining the label of a data point typically requires human experts (domain experts).

ML aims at predicting the label of a data point based solely on its features.

Data - Examples

- **Data points**: house, text, genome sequence, image, speech
- **Features**: number of rooms, word “cat” count in the text, frequency of AATCAGTT motif, pixel values, spectrum of the signal
- **Labels**: price, article topic, cell type, identify objects in the image, emotional state

Cat dataset - Features or Labels?

- 612x612 RGB pixel values

- Cat
- No cat

Linnerud dataset - Features or Labels?

- Chin-ups
- Sit-ups
- Jumps

- Weight
- Waist
- Pulse

The hypothesis space

The hypothesis space of a ML method is a subset of all possible maps from the feature space to label space. The design choice of the hypothesis space should take into account available computational resources and statistical aspects.

$$h : \mathcal{X} \rightarrow \mathcal{Y}$$

Loss

Loss function $\mathcal{L}(y, \hat{y})$ measures the quality of a hypothesis map.

Examples:

- MSE $\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$
- MAE $\frac{1}{m}\sum_{i=1}^{m}|y^{(i)} - \hat{y}^{(i)}|$
- Logistic Loss, Cross Entropy
- Hinge Loss
- 0/1 Loss

Problem formulation in ML

Machine Learning: "Given an image of a histological analyses of a tissue, how likely that this sample is malignant tumor?"

Problem formulation:

- **Data**:
  - Data point - image of a tissue sample
  - Features - pixels
  - Labels - $\{$cancer, no cancer$\}$
- **Model**: Logistic regression
- **Loss**: Logistic loss

Machine Learning: Components - Questions?

Table of Contents

1 Course Logistics

2 Machine Learning: Definitions

3 Machine Learning: Components

4 Machine Learning: Implementation

Python libraries

Python classes

- **Objects** are Python's abstraction for data. All data in a Python program is represented by objects or by relations between objects.

- Creating a new class creates a new type of object

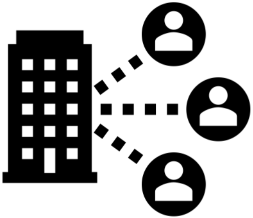
- Class allows to create new **instances** of that type

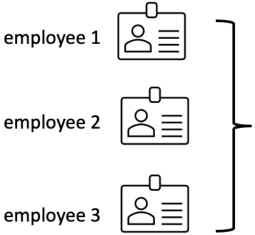
- Each class instance can have **attributes** attached to it for maintaining its state.

- Class instances can also have **methods** (defined by its class) for modifying its state.

Python classes - Example

Company employees





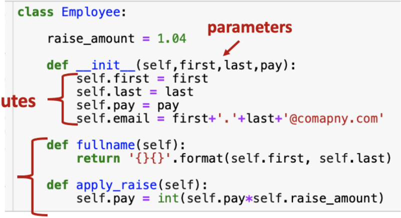
Attributes:

- first name
- last name
- pay
- email

Methods:

- get full name
- pay raise

Python classes - Example



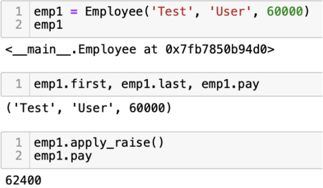
```
class Employee:

    raise_amount = 1.04

    def __init__(self,first,last,pay):
        self.first = first
        self.last = last
        self.pay = pay
        self.email = first+'.'+last+'@comapny.com'

    def fullname(self):
        return '{}{}'.format(self.first, self.last)

    def apply_raise(self):
        self.pay = int(self.pay*self.raise_amount)
```

parameters

attributes

methods

instantiate a class =
create an object of that type



```
emp1 = Employee('Test', 'User', 60000)
emp1
```

```
<__main__.Employee at 0x7fb7850b94d0>
```

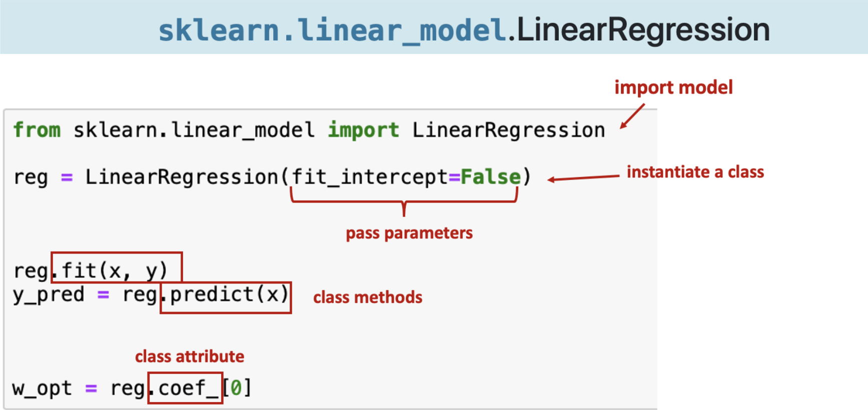
```
emp1.first, emp1.last, emp1.pay
```

```
('Test', 'User', 60000)
```

```
emp1.apply_raise()
emp1.pay
```

```
62400
```

Python classes - sklrean



sklearn.linear_model.LinearRegression

```
from sklearn.linear_model import LinearRegression

reg = LinearRegression(fit_intercept=False)


reg.fit(x, y)
y_pred = reg.predict(x)


w_opt = reg.coef_[0]
```

import model

instantiate a class

pass parameters

class methods

class attribute

Using ML models with sklearn

Import model from sklearn:

- `from sklearn.linear_model import LinearRegression`
- `from sklearn.tree import DecisionTreeClassifier`
- `from sklearn.svm import SVR`

Instantiate a class (create an object of that class):

- `reg_lin = LinearRegression(fit_intercept=False)`
- `clf_dt = DecisionTreeClassifier(random_state=0)`
- `reg_svm = SVR(C=1.0, epsilon=0.2)`

Using ML models with sklearn

Fit a model:

- `reg_lin.fit(X,y)`
- `clf_dt.fit(X,y)`
- `reg_svm.fit(X,y)`

Make predictions:

- `reg_lin.predict(X)`
- `clf_dt.predict(X)`
- `reg_svm.predict(X)`

The hypothesis space

Choosing the right estimator with sklearn.

Machine Learning: Implementation - Questions?