Lab 1. Introduction

Intro to Machine Learning
Fall 2018, Innopolis University

Lecture recap

- Importance of and reasons for machine learning
- What is learning (a very simple example)
- Different types of learning
- Predictors and response variables
- Regression and classification
- Goals of learning
- Parametric and non-parametric models
- Assessing the quality of learning

Questions about the lecture

Was the material already familiar to you?

What new things have you learned?

What was hard to understand?

Tools we will use, language we will speak

For communication: English / Russian

For programming: Python

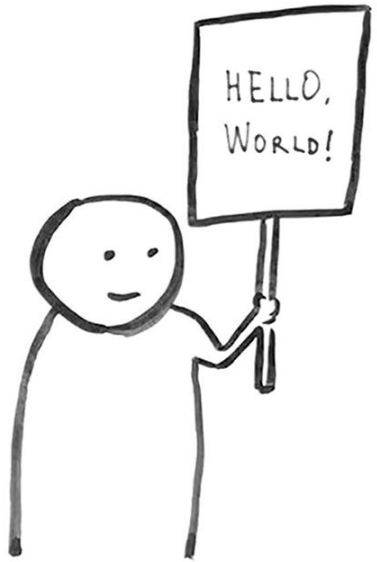
IDE of your choice:

- PyCharm
- Jupyter Notebook
- Spyder

Tools we will use, language we will speak

1) Install Python - go to https://www.python.org/downloads/ download and install the latest version
2) Install PyCharm - follow instructions for your OS on this page - https://www.jetbrains.com/help/pycharm/install-and-set-up-pycharm.html
    - You can apply for educational license here - https://www.jetbrains.com/student/ and use the full version of IDE

“A journey of a thousand miles begins with a single step”



Let’s do it!

```
print("hello, world!")

greetings = "hello, world!"
print(greetings)

x = 21
y = 34
print(x+y)
```

Strings and input

```
str1 = "Hello, "
str2 = 'world'
str1 += str2
print(str1)

name = input()
print("Hello,", name)
```

Integers

```
x = 2018
y = 100
print(x/y) #20.18
print(x//y) #20

x = 2**10
print(x) #1024
x = 2**100
print(x) #1267650600228229401496703205376
```

Types

```
x = "hello"
print(type(x)) #<class 'str'>
x = 42
print(type(x)) #<class 'int'>

n = int(input())
print(2**n)

x = None
```

Import

```
import math
z = math.sqrt(144)
print(z) #12.0

from math import sqrt
z = sqrt(144)
print(z) #12.0
```

Conditions

```
x = True
y = False
if x and y:
  print("Both true")
elif x or y:
  print("One is true")
else:
  print("Both false")
```

Lists

```
lst = [1,1,2,3,5,8,13,21]
print(lst[0]) #1
print(lst[-1]) #21
print(lst[:3]) #[1, 1, 2]
print(lst[3:]) #[3, 5, 8, 13, 21]
print(lst[1:3]) #[1, 2]
print(lst[:]) #[1, 1, 2, 3, 5, 8, 13, 21]
print(lst[2:6:2]) #[2, 5]

r = range(10) #[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

For

```
sum = 0
for i in range(10):
   sum += i
print(sum) #45


for c in "Hello":
   print(c)
```

While

```
sum = 0
i = 9
while i > 0:
   sum += i
   i -= 1
print(sum) #45
```

Continue, break

```
i = 10
sum = 0
while True:
    i -= 1
    if i == 0:
        break
    if i % 2 == 1:
        continue
    sum += i
print(sum) #20
```

Functions

```
from math import sqrt
def hypotenuse(x, y):
    z = x**2 + y**2
    return sqrt(z)

h = hypotenuse(3, 4)
print(h)
```

Dict

```
x = dict()
x["Yandex"] = 51.7
x["Google"] = 43.3
x["Ohters"] = 5.0
print(x["Yandex"]) #51.7
#print(x["Mail.ru"]) #Exception
x = {"Yandex" : 51.7, "Google" : 43.3, "Others" : 5.0}
print(x) #{'Yandex': 51.7, 'Google': 43.3, 'Others': 5.0}
```

Some libraries we will use

- NumPy
- Matplotlib
- Pandas
- Scikit-learn

Some libraries we will use

- NumPy - intended for processing large multidimensional arrays and matrices, and has an extensive collection of high-level mathematical functions and implemented methods
- Matplotlib
- Pandas
- Scikit-learn

Some libraries we will use

- NumPy
- Matplotlib - a low-level library for creating two-dimensional diagrams and graphs. With its help, you can build diverse charts, from histograms etc.
- Pandas
- Scikit-learn

Some libraries we will use

- NumPy
- Matplotlib
- Pandas - library that provides high-level data structures and a vast variety of tools for analysis
- Scikit-learn

Some libraries we will use

- NumPy
- Matplotlib
- Pandas
- Scikit-learn - based on NumPy is one of the best libraries for working with data. It provides algorithms for many standard machine learning tasks such as clustering, regression, classification, dimensionality reduction, and model selection

Install packages

pip install numpy matplotlib

Let’s take a look at some datasets

Download an archive containing several csv files from Moodle

Let’s load them one by one

```
import pandas as pd

df1 = pd.read_csv("PathToTheFile/FileName.csv")

print(df1)
```

Let’s take a look at some datasets

Play with settings to see more:

```
with pd.option_context('display.max_rows', 10,
'display.max_columns', 20, 'display.width', 1000):
    print(df1)
```

Let’s take a look at some datasets

Iris 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
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa

What are dependent (response) and what are independent (predictor) variables?
What are the variable types? quantitative / qualitative
Is it a supervised / unsupervised problem?
What is the task here?
How many classes are there? What are they?

Let’s take a look at some datasets

Property dataset

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view
0	20140915T000000	392000.0	4	3.75	2220	3797	1.5	0	0
1	20140730T000000	300000.0	3	2.25	1960	1585	2.0	0	0
2	20150325T000000	440000.0	2	1.50	1330	15873	1.0	0	0
3	20150219T000000	800500.0	4	2.50	1780	11130	1.0	0	0
4	20140701T000000	485000.0	4	1.75	1430	4096	2.0	0	0
...	...	...	...	...	...	...	...	...	...
17285	20140521T000000	525000.0	3	2.75	2100	10362	2.0	0	0
17286	20141114T000000	630000.0	3	3.25	3800	13995	2.0	0	3

What are dependent (response) and what are independent (predictor) variables?
What are the variable types? quantitative / qualitative
Is it a supervised / unsupervised problem?
What is the task here?

Let’s take a look at some datasets

Mall_customers dataset

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
..	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28

What are dependent (response) and what are independent (predictor) variables?

What are the variable types? quantitative / qualitative

Is it a supervised / unsupervised problem?

What is the task here?

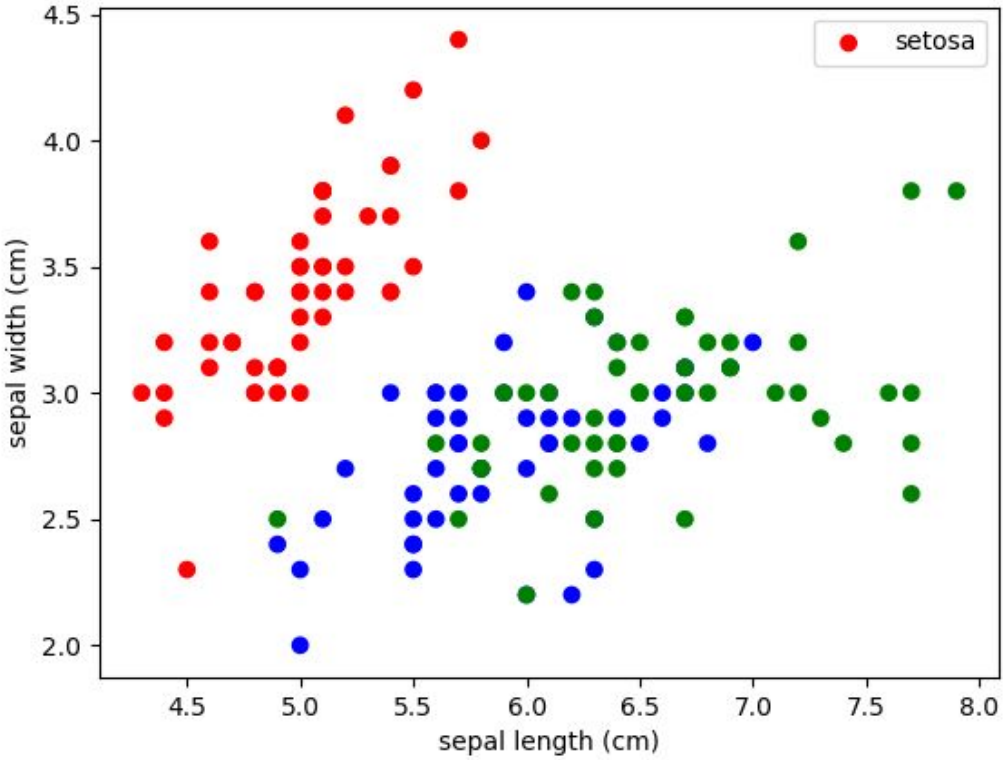
Visualization

```
import pandas as pd
from matplotlib import pyplot as plt

iris = pd.read_csv("datasets/iris.csv")
colors = {'setosa':'red', 'versicolor':'blue', 'virginica':'green'}

plt.scatter(iris.sepal_length, iris.sepal_width,
                                c=iris.species.apply(lambda x: colors[x]))
plt.xlabel("sepal length (cm)")
plt.ylabel("sepal width (cm)")
plt.legend(colors)
plt.show()
```

Visualization



How response and predictor variables are related to each other?

Will linear model be a good fit or not?

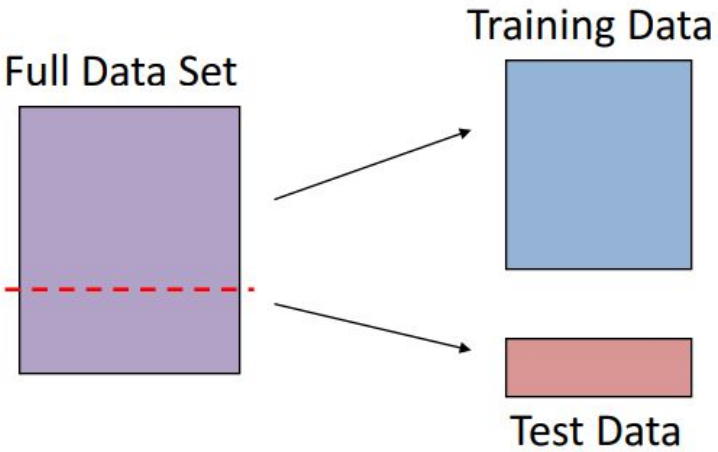
Visualization

- Build the same kind of scatter plot but for petal length/width. Tell what you see

- Build a scatter plot for Mall_customers dataset. (x, y axes - for income and spendings, colors based on gender). Do you see any patterns?

Training data vs Test data

- Training data: data used to build the model
- Testdata: new data, not used in the training process



Why is it important to split?

How to split?

Exercise

Write a python function which splits a dataframe into training and test sets:

- Inputs: X (features), y(target), fraction
- Outputs: X_train, y_train, X_test, y_test
- Data should be shuffled before splitting
- Don’t use ready utilities for train-test split. DIY!

Test it by passing **iris** data set and verifying the fraction

That’s it for today! Questions?