Supervised ML WS23
Assignment 0

Within this introductory assignment we want to give you the chance to familiarise yourself with Jupyter Notebooks and this course's requirements for your submissions.

Copyrighting and Fare Use

Automatic Testing Guidelines

Your submissions are tested for correctness and plagiarism automatically. Automatic unittesting requires you, as a student, to submit a notebook which contains strictly defined objects. Cell tags, predefined function and variable names,dtypes and more.
Within the notebook we provide detailed instruction which you may want to follow, in order to maximise your final grade.

Name your notebook properly; follow the pattern in the template name:

Assignment_N_NameSurname_matrnumber

1. N - number of assignment
2. NameSurname - your full name where every part of the name starts with a captial letter, no spaces
3. matrnumber - your student number as it appears on your ID card (without K)

You may notice, that most cells are tagged, that is for the unittest routine to recognise them. We highly recommend you to develop your code within provided cells. Adding (and definitley removing) cells might badly influence your sumbission grade. Also changing given function and variable names will cause the unittest to break and therefore lead to point deductions.

Please be careful, and may the force be with you.

LaTeX

During the course you will be often supposed to derive formulas and perform calculations. The perfect tool to use in that case is LaTeX.
By developers definition: "*LaTeX is a high-quality typesetting system which includes features designed for the production of technical and scientific documentation.*"

Tips how to use LaTeX:

- Enter LaTeX mode (inline) by typing iside of **two dollar signs.**
  Your first LaTeX statement $y = x^2$
- For centered, larger, stand-alone formulas use **double dollar signs** on each side:

$$\sqrt{x^2} = y$$

- Use **curly brackets** to define arguments:
  No argument definition: $x^2 \cdot x^3 \neq x^2 + 3$
  Using brackets for argument definition: $x^2 \cdot x^3 = x^{2+3}$
- Go for fancy tags to make your styling readable.
  Blackboard Bold style can be used for sets of numbers. Use **\mathbb** tag. Examples: $\mathbb{R}, \mathbb{Z}, \mathbb{I}, \mathbb{Q}$
  Fractions can be written with **\frac** tag. Example: $\frac{x^2-1}{x-1} = \frac{(x-1)(x+1)}{x-1} = x + 1$
- Adjust bracked hight with **\left \right** tags.
  Formula without brackets adjustments: $y = (\frac{\mathbb{E}(X)}{\mathbb{E}(Y)})^{x-y}$
  Formula with brackets adjustments: $y = \left(\frac{\mathbb{E}(X)}{\mathbb{E}(Y)}\right)^{x-y}$
- Full latex cheatsheet

Write a LaTeX formula (7 points)

Show that for $ax^2 + bx + c$ roots are:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Your solution goes here ↓↓↓

$$ax^2 + bx + c = 0$$

$$ax^2 + bx = -c$$

$$x^2 + \frac{b}{a}x = -\frac{c}{a}$$

$$x^2 + \frac{b}{a}x + \left(\frac{b}{2a}\right)^2 = -\frac{c}{a} + \left(\frac{b}{2a}\right)^2$$

$$x^2 + \frac{b}{a}x + \left(\frac{b}{2a}\right)^2 = -\frac{c}{a} + \frac{b^2}{4a^2}$$

$$\left(x + \frac{b}{2a}\right)^2 = \frac{b^2 - 4ac}{4a^2}$$

$$x + \frac{b}{2a} = \pm\sqrt{\frac{b^2 - 4ac}{4a^2}}$$

$$x = -\frac{b}{2a} \pm \sqrt{\frac{b^2 - 4ac}{4a^2}}$$

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Coding

In order to pass unittests please follow the instructions carefully.

Imports

Install and import packages you need.

In [19]:
```
import numpy as np
from matplotlib import pyplot as plt
```

Write a function (5 points)

Write a function **bivar_normal()** which returns a numpy array of shape (100,2) filled with normally distributed random datapoints with mean=0, std=1.

In [20]:
```
def bivar_normal():#do not change function name
    """Function bivar_normal takes no argument and returns a numpy array

    Returns
    -------
    np.ndarray
        numpy array of shape (100,2) of np.float32 dtype with no rounding of values
    """
    #your code goes here ↓↓↓
    return np.random.normal(0, 1, size=(100, 2)).astype('f')
```

Plot (5 points)

Define a variable **data** through calling a **bivar_normal()** function developed above.
Plot your **data** in a scatterplot by using matplotlib library. Set axis labels and plot title properly.

In [21]:
```
#please use predefined names of varibales
data = bivar_normal(); #define data variable
```

```
assert data.dtype == np.float32
assert data.shape == (100, 2)
```

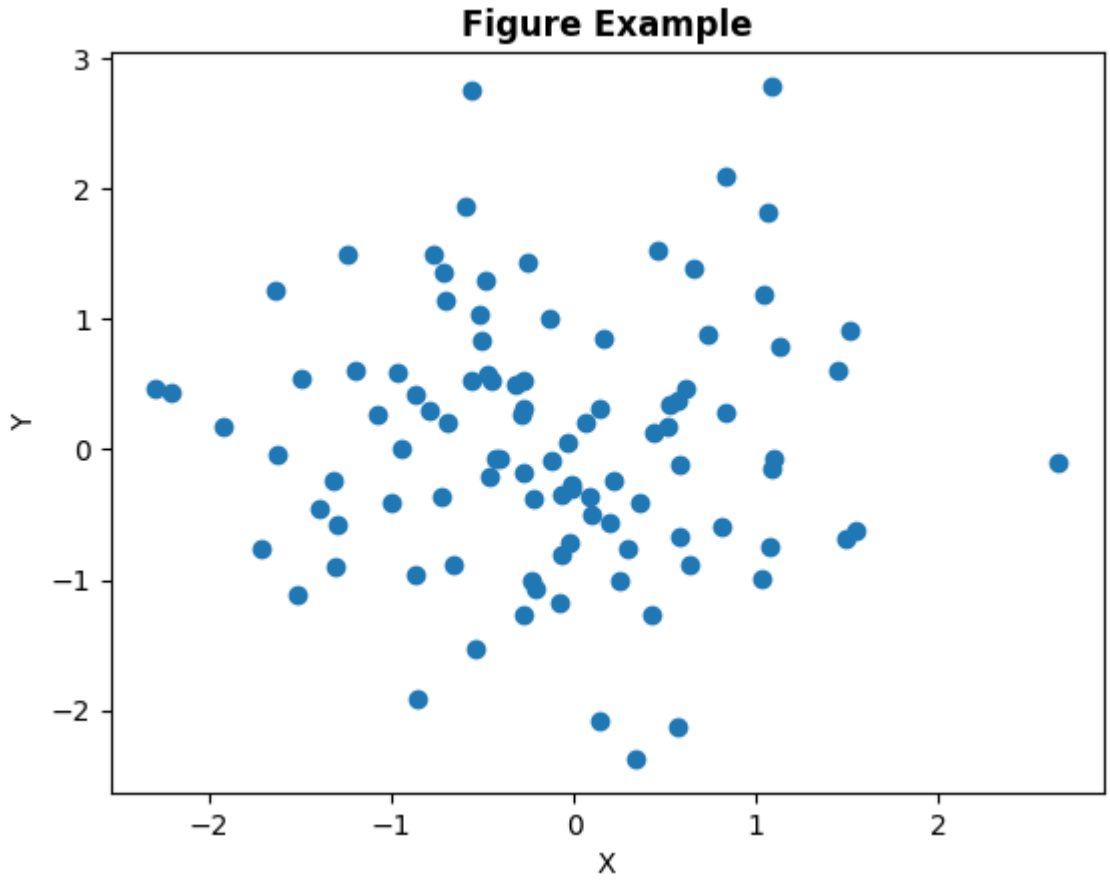
In [22]:
```
def gen_plot(data):#do not change formula name
    """write a function that plots the data

    Returns
    -------
    matplotlib figure
        generated figure object
    """
    fig = plt.figure()
    #your gode goes here ↓↓↓
    plt.scatter(data[:, 0], data[:, 1])
    plt.title('Figure Example', fontweight ='bold')
    plt.xlabel("X")
    plt.ylabel("Y")
    return fig
```

In [23]:
```
# in here the plot is shown
fig = gen_plot(data)
fig.show()

assert isinstance(fig, type(plt.figure()))
```

```
C:\Users\haslh\AppData\Local\Temp\ipykernel_22660\962695308.py:3: UserWarning: Mat
plotlib is currently using module://matplotlib_inline.backend_inline, which is a n
on-GUI backend, so cannot show the figure.
  fig.show()
```




```
<Figure size 640x480 with 0 Axes>
```

Test question (2 points)

What is true? Multiple answers might be correct.

Pegasus is:
a_) half horse, half human
b_) a horse with wings
c_) an elephant flying with its huge ears
d_) a mythical creature

To answer the question assign to variables in the next cell **True** or **False** boolean values.
To earn points **assign values to all variables**

In [24]:
```
#examples for you
example_of_true_variable = True
example_of_false_variable = False

#your answers go here ↓↓↓,
a_ = False
b_ = True
c_ = False
d_ = False
```

Make sure your submission has no errors (1 point)

Remove all errors in code and run all cells to make sure your code is executable.

In [ ]:

Technical cells

The cells below are needed for efficient unittesting. Do not delete or change them in order to receive proper evaluation.
The executability check might help you.

In [25]:
```
def bivar_normal_tl():
    try:
        return bivar_normal().tolist()
    except:
        raise ValueError("Check if your bivar_normal() functions returns numpy nd a
```

In [26]:
```
#executability check
bivar_normal_tl()
print("Executable")
```

```
Executable
```

In [ ]: