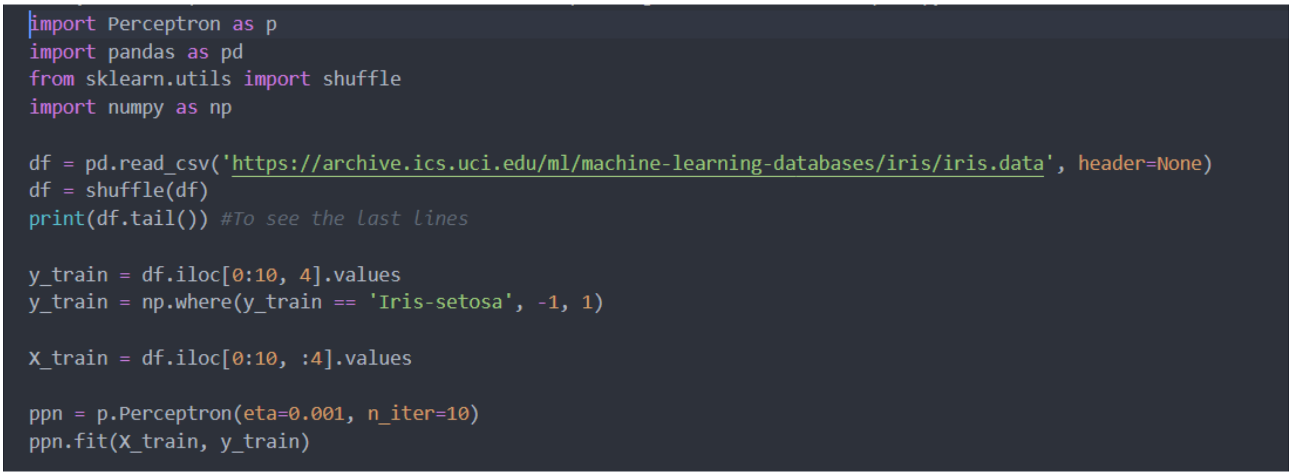
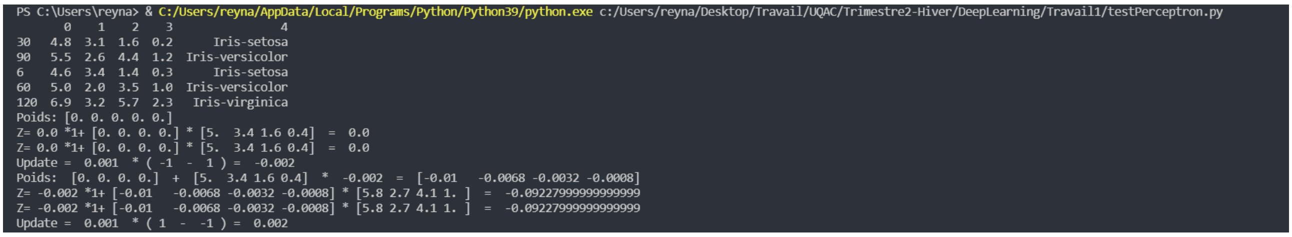
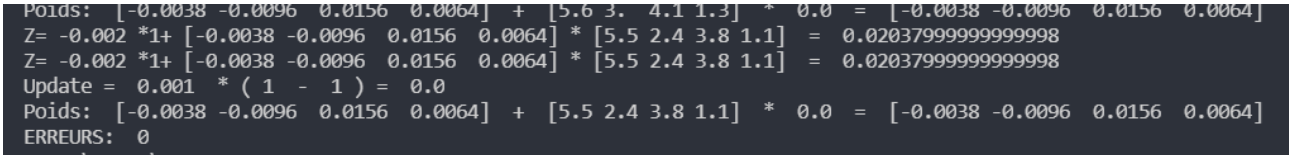
Perceptron donné avec le set iris



```python
import Perceptron as p
import pandas as pd
from sklearn.utils import shuffle
import numpy as np

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data', header=None)
df = shuffle(df)
print(df.tail()) #To see the last lines

y_train = df.iloc[0:10, 4].values
y_train = np.where(y_train == 'Iris-setosa', -1, 1)

X_train = df.iloc[0:10, :4].values

ppn = p.Perceptron(eta=0.001, n_iter=10)
ppn.fit(X_train, y_train)
```

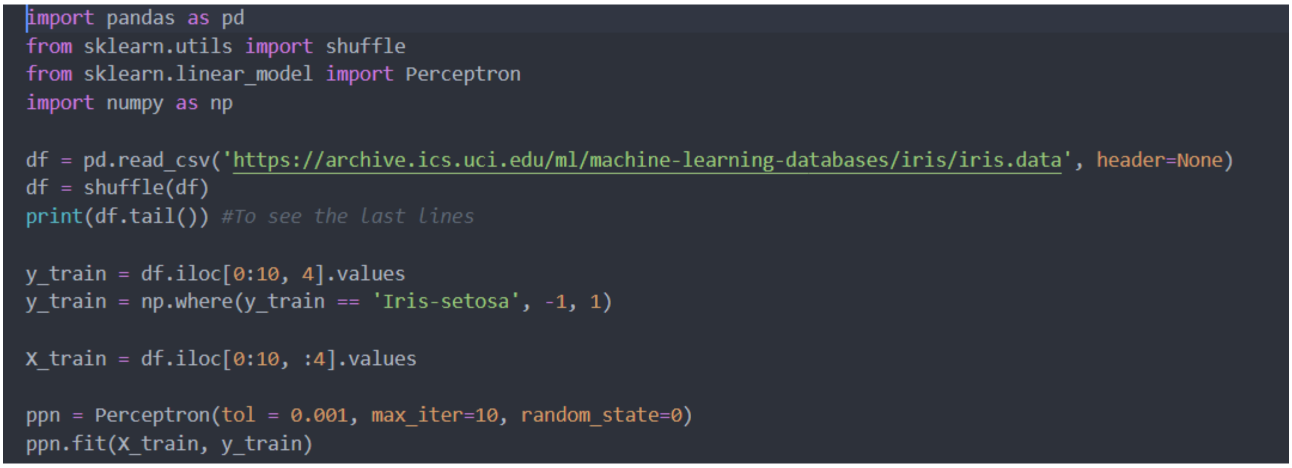


```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail1/testPerceptron.py
       0    1    2    3                4
30   4.8  3.1  1.6  0.2      Iris-setosa
90   5.5  2.6  4.4  1.2  Iris-versicolor
6    4.6  3.4  1.4  0.3      Iris-setosa
60   5.0  2.0  3.5  1.0  Iris-versicolor
120  6.9  3.2  5.7  2.3   Iris-virginica
Poids: [0. 0. 0. 0. 0.]
Z= 0.0 *1+ [0. 0. 0. 0.] * [5.  3.4 1.6 0.4]  =  0.0
Z= 0.0 *1+ [0. 0. 0. 0.] * [5.  3.4 1.6 0.4]  =  0.0
Update =  0.001  * ( -1  -  1 ) =  -0.002
Poids:  [0. 0. 0. 0.]  +  [5.  3.4 1.6 0.4]  *  -0.002  =  [-0.01   -0.0068 -0.0032 -0.0008]
Z= -0.002 *1+ [-0.01   -0.0068 -0.0032 -0.0008] * [5.8 2.7 4.1 1. ]  =  -0.09227999999999999
Z= -0.002 *1+ [-0.01   -0.0068 -0.0032 -0.0008] * [5.8 2.7 4.1 1. ]  =  -0.09227999999999999
Update =  0.001  * ( 1  -  -1 ) =  0.002
```

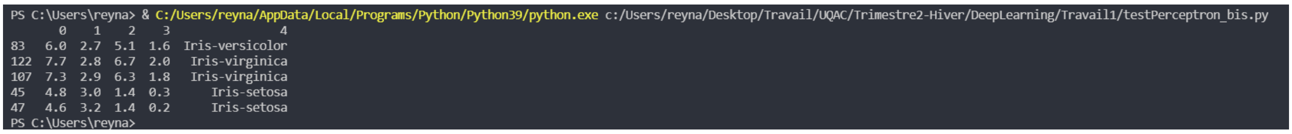


```
Poids:  [-0.0038 -0.0096  0.0156  0.0064]  +  [5.6 3.  4.1 1.3]  *  0.0  =  [-0.0038 -0.0096  0.0156  0.0064]
Z= -0.002 *1+ [-0.0038 -0.0096  0.0156  0.0064] * [5.5 2.4 3.8 1.1]  =  0.020379999999999998
Z= -0.002 *1+ [-0.0038 -0.0096  0.0156  0.0064] * [5.5 2.4 3.8 1.1]  =  0.020379999999999998
Update =  0.001  * ( 1  -  1 ) =  0.0
Poids:  [-0.0038 -0.0096  0.0156  0.0064]  +  [5.5 2.4 3.8 1.1]  *  0.0  =  [-0.0038 -0.0096  0.0156  0.0064]
ERREURS:  0
```

Perceptron de scikit learn avec le set iris

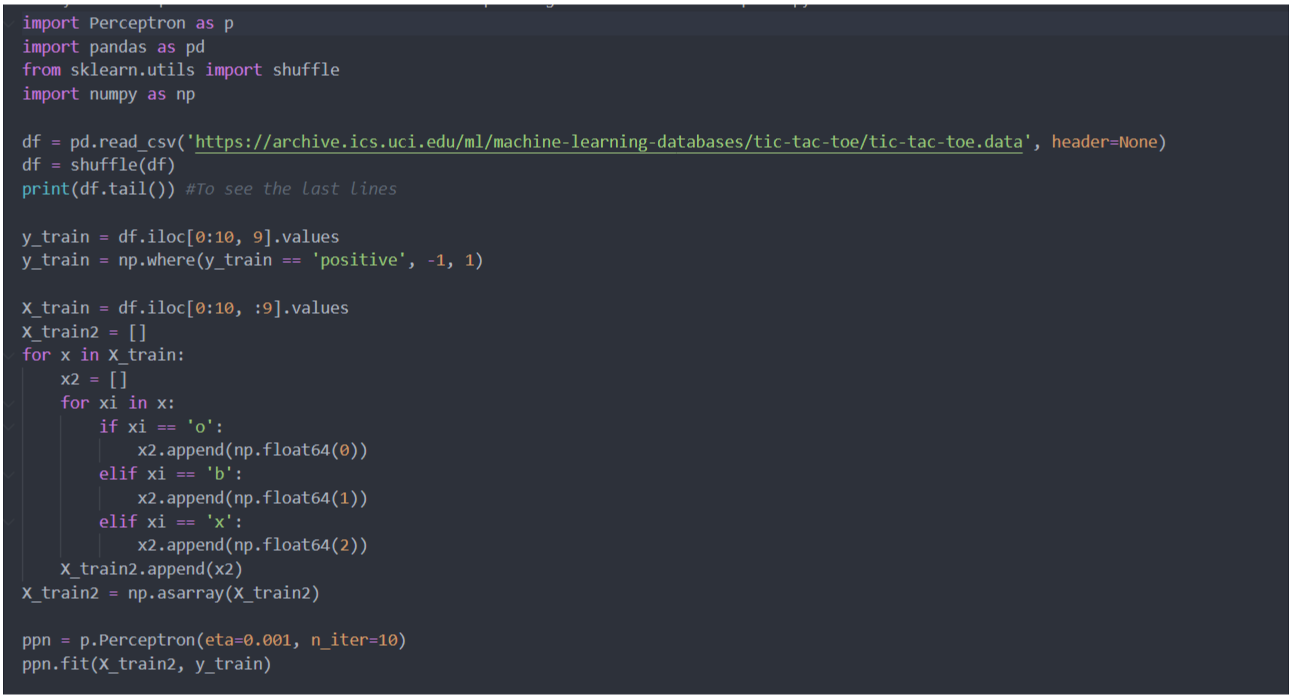


```python
import pandas as pd
from sklearn.utils import shuffle
from sklearn.linear_model import Perceptron
import numpy as np

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data', header=None)
df = shuffle(df)
print(df.tail()) #To see the last lines

y_train = df.iloc[0:10, 4].values
y_train = np.where(y_train == 'Iris-setosa', -1, 1)

X_train = df.iloc[0:10, :4].values

ppn = Perceptron(tol = 0.001, max_iter=10, random_state=0)
ppn.fit(X_train, y_train)
```

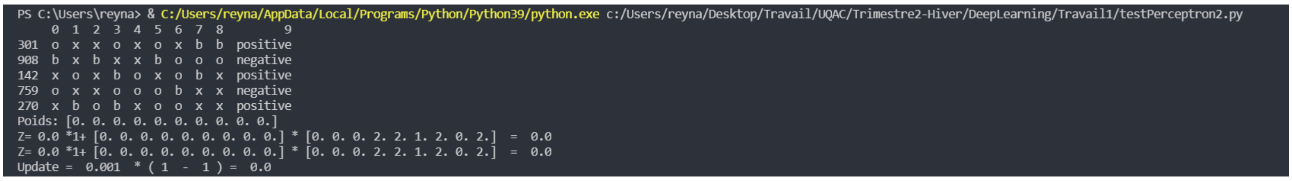


```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail1/testPerceptron_bis.py
       0    1    2    3                4
83   6.0  2.7  5.1  1.6  Iris-versicolor
122  7.7  2.8  6.7  2.0   Iris-virginica
107  7.3  2.9  6.3  1.8   Iris-virginica
45   4.8  3.0  1.4  0.3      Iris-setosa
47   4.6  3.2  1.4  0.2      Iris-setosa
PS C:\Users\reyna>
```

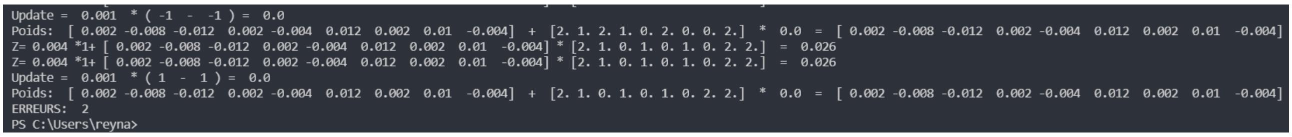
Perceptron donné avec le set tic tac toe



```
import Perceptron as p
import pandas as pd
from sklearn.utils import shuffle
import numpy as np

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/tic-tac-toe/tic-tac-toe.data', header=None)
df = shuffle(df)
print(df.tail()) #To see the last lines

y_train = df.iloc[0:10, 9].values
y_train = np.where(y_train == 'positive', -1, 1)

X_train = df.iloc[0:10, :9].values
X_train2 = []
for x in X_train:
    x2 = []
    for xi in x:
        if xi == 'o':
            x2.append(np.float64(0))
        elif xi == 'b':
            x2.append(np.float64(1))
        elif xi == 'x':
            x2.append(np.float64(2))
    X_train2.append(x2)
X_train2 = np.asarray(X_train2)

ppn = p.Perceptron(eta=0.001, n_iter=10)
ppn.fit(X_train2, y_train)
```

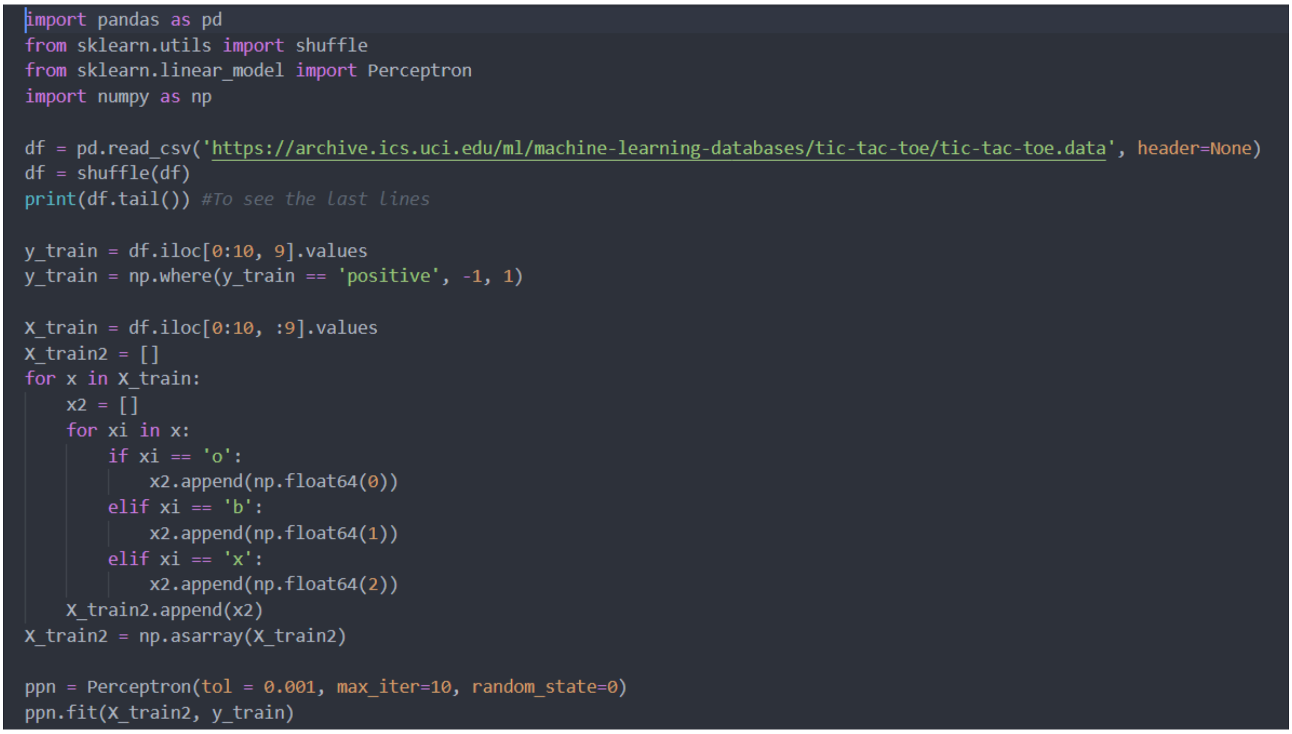


```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail1/testPerceptron2.py
     0  1  2  3  4  5  6  7  8         9
301  o  x  x  o  x  o  x  b  b  positive
908  b  x  b  x  x  b  o  o  o  negative
142  x  o  x  b  o  x  o  b  x  positive
759  o  x  x  o  o  o  b  x  x  negative
270  x  b  o  b  x  o  o  x  x  positive
Poids: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Z= 0.0 *1+ [0. 0. 0. 0. 0. 0. 0. 0. 0.] * [0. 0. 0. 2. 2. 1. 2. 0. 2.]  =  0.0
Z= 0.0 *1+ [0. 0. 0. 0. 0. 0. 0. 0. 0.] * [0. 0. 0. 2. 2. 1. 2. 0. 2.]  =  0.0
Update =  0.001  * ( 1  -  1 ) =  0.0
```

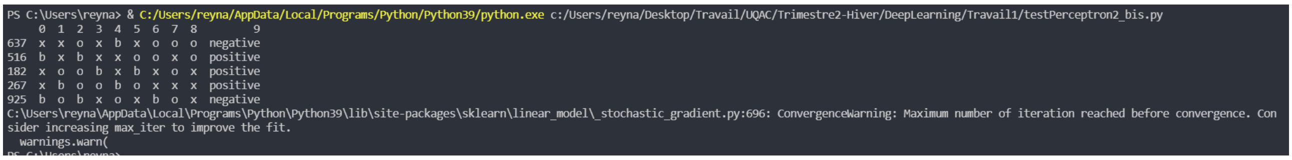


```
Update =  0.001  * ( -1  -  -1 ) =  0.0
Poids:  [ 0.002 -0.008 -0.012  0.002 -0.004  0.012  0.002  0.01  -0.004]  +  [2. 1. 2. 1. 0. 2. 0. 0. 2.]  *  0.0  =  [ 0.002 -0.008 -0.012  0.002 -0.004  0.012  0.002  0.01  -0.004]
Z= 0.004 *1+ [ 0.002 -0.008 -0.012  0.002 -0.004  0.012  0.002  0.01  -0.004] * [2. 1. 0. 1. 0. 1. 0. 2. 2.]  =  0.026
Z= 0.004 *1+ [ 0.002 -0.008 -0.012  0.002 -0.004  0.012  0.002  0.01  -0.004] * [2. 1. 0. 1. 0. 1. 0. 2. 2.]  =  0.026
Update =  0.001  * ( 1  -  1 ) =  0.0
Poids:  [ 0.002 -0.008 -0.012  0.002 -0.004  0.012  0.002  0.01  -0.004]  +  [2. 1. 0. 1. 0. 1. 0. 2. 2.]  *  0.0  =  [ 0.002 -0.008 -0.012  0.002 -0.004  0.012  0.002  0.01  -0.004]
ERREURS:  2
PS C:\Users\reyna>
```

Perceptron de scikit learn avec le set tic tac toe

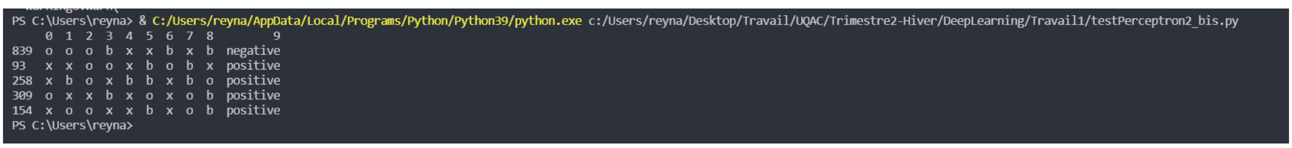


```python
import pandas as pd
from sklearn.utils import shuffle
from sklearn.linear_model import Perceptron
import numpy as np

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/tic-tac-toe/tic-tac-toe.data', header=None)
df = shuffle(df)
print(df.tail()) #To see the last lines

y_train = df.iloc[0:10, 9].values
y_train = np.where(y_train == 'positive', -1, 1)

X_train = df.iloc[0:10, :9].values
X_train2 = []
for x in X_train:
    x2 = []
    for xi in x:
        if xi == 'o':
            x2.append(np.float64(0))
        elif xi == 'b':
            x2.append(np.float64(1))
        elif xi == 'x':
            x2.append(np.float64(2))
    X_train2.append(x2)
X_train2 = np.asarray(X_train2)

ppn = Perceptron(tol = 0.001, max_iter=10, random_state=0)
ppn.fit(X_train2, y_train)
```



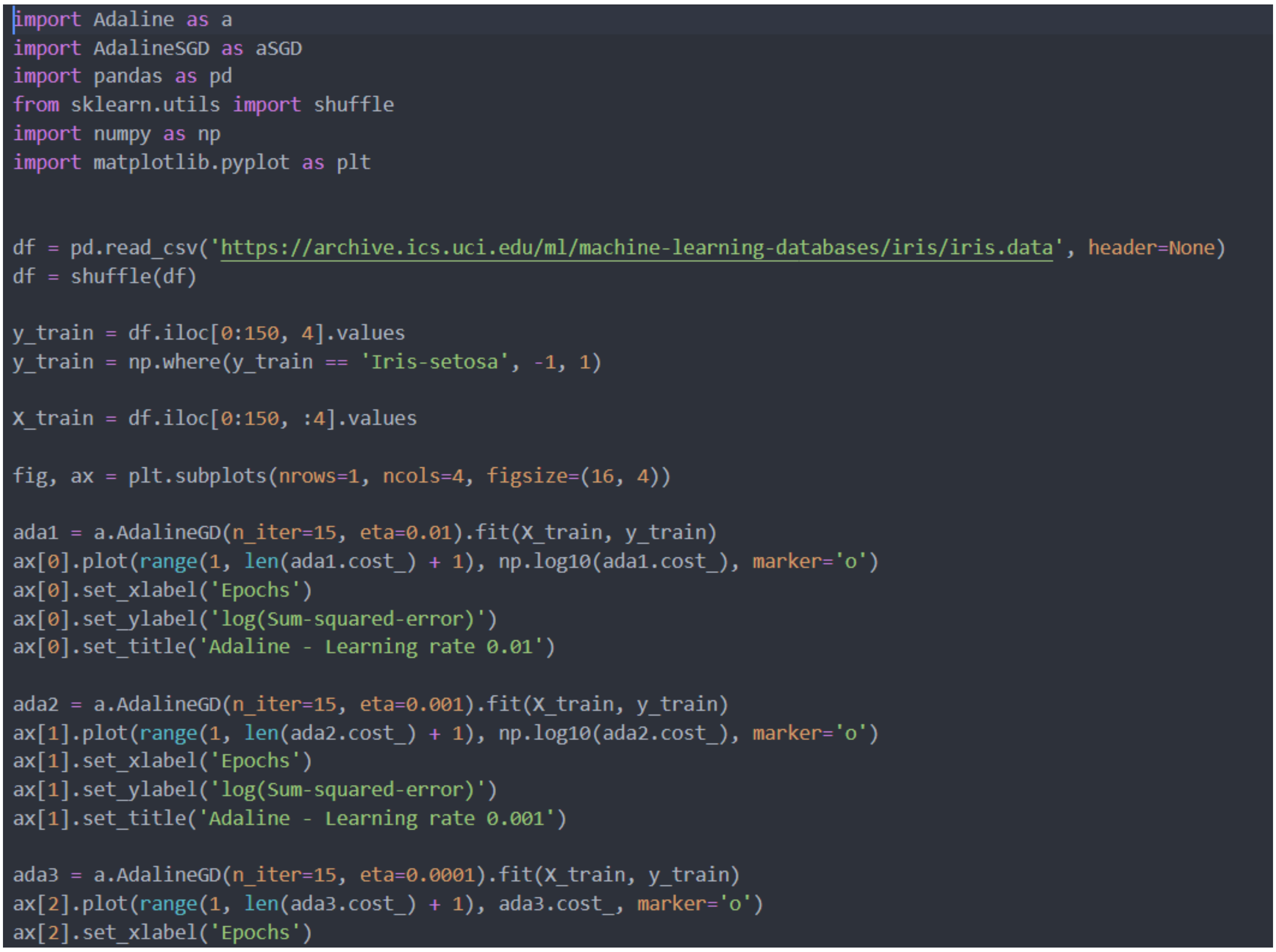
```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail1/testPerceptron2_bis.py
     0  1  2  3  4  5  6  7  8         9
637  x  x  o  x  b  x  o  o  o  negative
516  b  x  b  x  x  o  o  x  o  positive
182  x  o  o  b  x  b  x  o  x  positive
267  x  b  o  o  b  o  x  x  x  positive
925  b  o  b  x  o  x  b  o  x  negative
C:\Users\reyna\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\linear_model\_stochastic_gradient.py:696: ConvergenceWarning: Maximum number of iteration reached before convergence. Con
sider increasing max_iter to improve the fit.
  warnings.warn(
PS C:\Users\reyna>
```

max_iter = 15

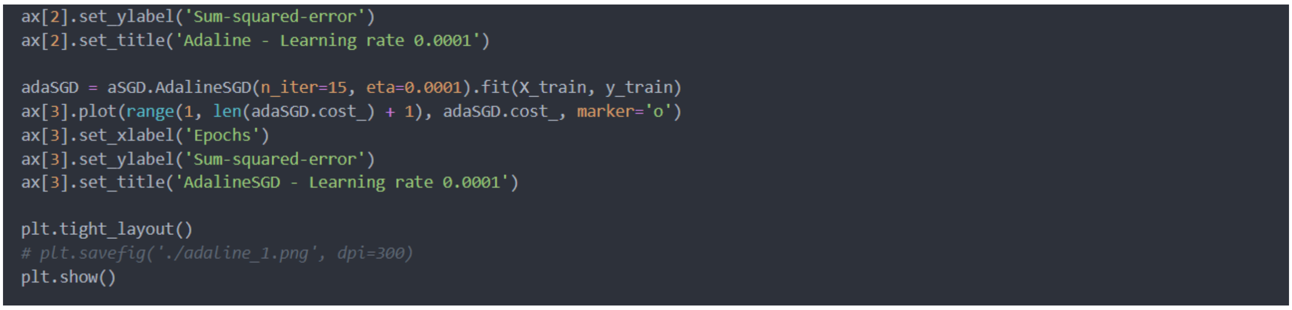


```
  warnings.warn(
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail1/testPerceptron2_bis.py
     0  1  2  3  4  5  6  7  8         9
839  o  o  o  b  x  x  b  x  b  negative
93   x  x  o  o  x  b  o  b  x  positive
258  x  b  o  x  b  b  x  b  o  positive
309  o  x  x  b  x  o  x  o  b  positive
154  x  o  o  x  x  b  x  o  b  positive
PS C:\Users\reyna>
```

Test Adaline



```
import Adaline as a
import AdalineSGD as aSGD
import pandas as pd
from sklearn.utils import shuffle
import numpy as np
import matplotlib.pyplot as plt


df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data', header=None)
df = shuffle(df)

y_train = df.iloc[0:150, 4].values
y_train = np.where(y_train == 'Iris-setosa', -1, 1)

X_train = df.iloc[0:150, :4].values

fig, ax = plt.subplots(nrows=1, ncols=4, figsize=(16, 4))

ada1 = a.AdalineGD(n_iter=15, eta=0.01).fit(X_train, y_train)
ax[0].plot(range(1, len(ada1.cost_) + 1), np.log10(ada1.cost_), marker='o')
ax[0].set_xlabel('Epochs')
ax[0].set_ylabel('log(Sum-squared-error)')
ax[0].set_title('Adaline - Learning rate 0.01')

ada2 = a.AdalineGD(n_iter=15, eta=0.001).fit(X_train, y_train)
ax[1].plot(range(1, len(ada2.cost_) + 1), np.log10(ada2.cost_), marker='o')
ax[1].set_xlabel('Epochs')
ax[1].set_ylabel('log(Sum-squared-error)')
ax[1].set_title('Adaline - Learning rate 0.001')

ada3 = a.AdalineGD(n_iter=15, eta=0.0001).fit(X_train, y_train)
ax[2].plot(range(1, len(ada3.cost_) + 1), ada3.cost_, marker='o')
ax[2].set_xlabel('Epochs')
```



```
ax[2].set_ylabel('Sum-squared-error')
ax[2].set_title('Adaline - Learning rate 0.0001')

adaSGD = aSGD.AdalineSGD(n_iter=15, eta=0.0001).fit(X_train, y_train)
ax[3].plot(range(1, len(adaSGD.cost_) + 1), adaSGD.cost_, marker='o')
ax[3].set_xlabel('Epochs')
ax[3].set_ylabel('Sum-squared-error')
ax[3].set_title('AdalineSGD - Learning rate 0.0001')

plt.tight_layout()
# plt.savefig('./adaline_1.png', dpi=300)
plt.show()
```

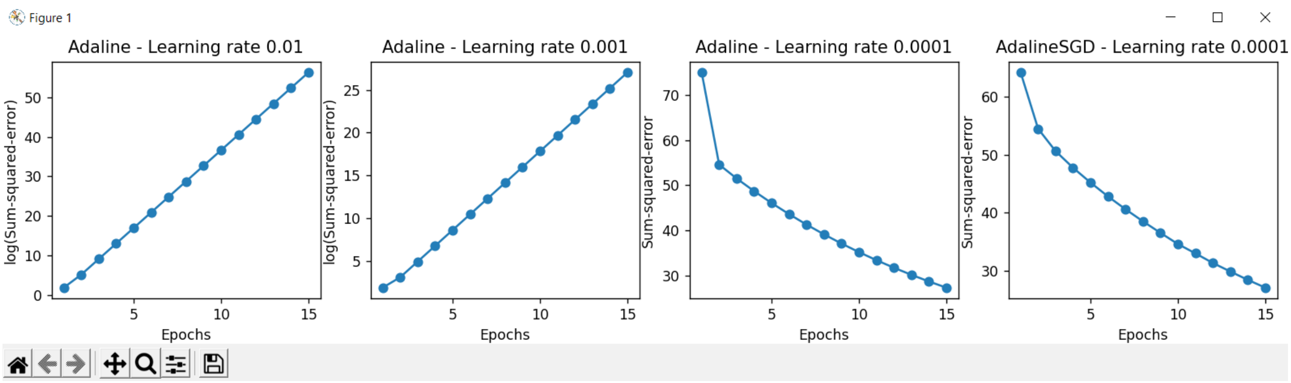


Figure 1
Adaline - Learning rate 0.01
log(Sum-squared-error)
Epochs
Adaline - Learning rate 0.001
log(Sum-squared-error)
Epochs
Adaline - Learning rate 0.0001
Sum-squared-error
Epochs
AdalineSGD - Learning rate 0.0001
Sum-squared-error
Epochs