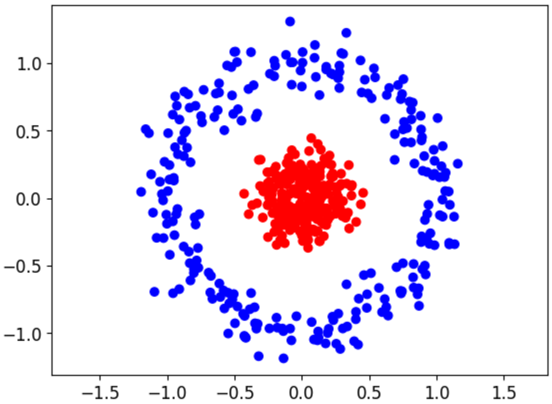
DA5401 Assignment 8

Let's consider an easy example of a non-linear dataset, the circles dataset. The dataset can be generated using the following code-snippet.

```
from sklearn.datasets import make_moons, make_circles
import matplotlib.colors as colors
import matplotlib.pyplot as plt

X, y = make_circles(n_samples=500, noise=0.1, random_state=42, factor=0.2)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
plt.scatter(X[:,0], X[:,1], c=y, cmap=colors.ListedColormap(["blue", "red"]))
plt.axis('equal')
plt.show()
```

The visualization of the dataset shall look like below:



We are going to learn a classifier in ensembling style using only weak classifiers to solve the above non-linear classification problem.

Task 1 [30 points]

Implement the Adaboost algorithm yourself from scratch.

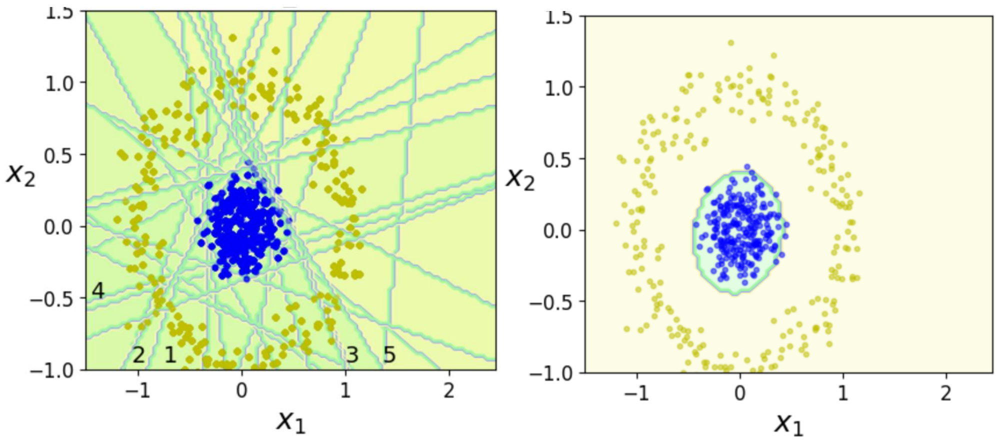
With:

- Samples $x_1 \dots x_n$
- Desired outputs $y_1 \dots y_n, y \in \{-1, 1\}$
- Initial weights $w_{1,1} \dots w_{n,1}$ set to $\frac{1}{n}$
- Error function $E(f(x), y_i) = e^{-y_i f(x_i)}$
- Weak learners $h \colon x \to \{-1, 1\}$

For t in $1 \dots T$:

- Choose $h_t(x)$:
  - Find weak learner $h_t(x)$ that minimizes ϵ_t, the weighted sum error for misclassified points $\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^{n} w_{i,t}$
  - Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- Add to ensemble:
  - $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$
- Update weights:
  - $w_{i,t+1} = w_{i,t} e^{-y_i \alpha_t h_t(x_i)}$ for i in $1 \dots n$
  - Renormalize $w_{i,t+1}$ such that $\sum_i w_{i,t+1} = 1$
  - (Note: It can be shown that $\frac{\sum_{h_t(x_i)=y_i} w_{i,t+1}}{\sum_{h_t(x_i)\neq y_i} w_{i,t+1}} = \frac{\sum_{h_t(x_i)=y_i} w_{i,t}}{\sum_{h_t(x_i)\neq y_i} w_{i,t}}$ at every step, which can simplify the calculation of the new weights.)

See above for the snapshot from Wikipedia. You may follow any ISLP/ESL/Bishop or any other standard books for the Adaboost algorithm for binary classification. In your implementation, \eta should be introduced as a hyper-parameter whose default value shall be 0.5. Your implementation should also generate the following visualizations. The left one is the classifier fit at every iteration. And the right one is the final ensembled classifier decision boundary.



Task 2 [20 points]

Run your Adaboost implementation with several weak classifiers such as LogReg, DecisionStump, DecisionTree(depth=3), Linear SVM, and LDA. Tune the method's hyperparameters (both Adaboost and the underlying weak classifier) for maximizing the classification performance. Based on the data visualization, you can achieve >98% performance fairly easily. Generate the decision boundary visualizations as the above figure pair for each model class.