Ensemble Exercises - Part 2

Summary

The first exercises are conceptual, while the last ones are more applied. The last two exercises are taken from *Hands-on Machine Learning ...* by A. Geron, chapter 7, which means you can compare your solutions to the ones provided with the book. However you are strongly encouraged to try and solve it in your own way first.

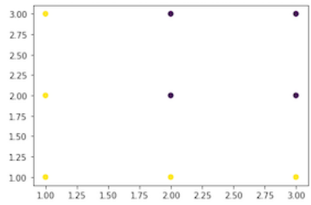
Exercise 1

In your own words,

- Explain the difference between Bagging, Random Forests, and Extra trees.
- Explain the difference between Bagging (and its associated methods) and Boosting methods.
- Explain how the out-of-bag evaluation can be used with bagging.

Exercise 2

Here we want to use AdaBoost and build a small ensemble of only three decision tree stumps. For the following training data build your ensemble classifier by hand (you may automate the computations if you like).



1. assign weights to all the data points
2. build the first stump by finding the best split of the data set
3. compute what is the amount of say (weight) of the stump

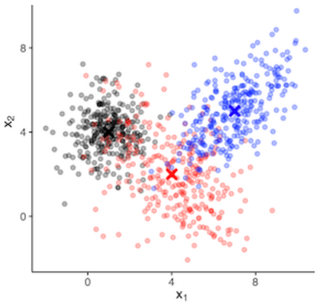
- Repeat the steps above to build two more stumps
- Compute the training error of your final ensemble classifier
- Make a sketch of the decision boundary

Exercise 3

In your own words, explain the differece between AdaBoost and Gradient Boosting.

Exercise 4

Now use the datasets Ex1-training.csv and Ex1-test.csv



- Use GradientBoostingClassifier from sklearn to build an ensemble of trees with max_dept=3.
- Visualize the decision boundaries (you may reuse/expand some code from Exercise_DT_setup in exercise 12).
- try different values for n_estimators and learning_rate.
- Compare the decision boundaries with what you have obtained from classification methods in previous weeks, e.g, Decision Trees, knn, LR, LDA and QDA.

Exercise 5

- Load the MNIST data and split it into a training set, a validation set.
- Then train various classifiers, such as a Random Forest classifier, an Extra-Trees classifier, and some other classifier of your choice.
- Next, try to combine them into an ensemble that outperforms them all on the validation set, using a soft or hard voting classifier.
- Once you have found one, try it on the test set. How much better does it perform compared to the individual classifiers?

In [1]:
```
# Setting up
from sklearn.model_selection import train_test_split
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Get Data
from sklearn.datasets import load_digits

X, y = load_digits(return_X_y=True)

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33
, random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.33,
random_state=42)
```

In [2]:
```
# more code here
```

Exercise 6

- Run the individual classifiers from the previous exercise to make predictions on the validation set, and create a new training set with the resulting predictions: each training instance is a vector containing the set of predictions from all your classifiers for an image, and the target is the image's class. Train a classifier on this new training set.
- Congratulations, you have just trained a blender (a technique derived from stacking), and together with the classifiers they form a stacking ensemble! Now let's evaluate the ensemble on the test set. For each image in the test set, make predictions with all your classifiers, then feed the predictions to the blender to get the ensemble's predictions. How does it compare to the voting classifier you trained earlier?