CSE4502 Programming Assignment #3

This programming assignment has two parts.

Part 1: Fashion MNIST

Load the fashion MNIST data set, and split it into a training set, a validation set, and a test set (e.g., use 40,000 instances for training, 10,000 for val- idation, and 10,000 for testing). Then train various classifiers, such as a Random Forest classifier, an Extra-Trees classifier, and an MLP classifier (use code mlp_clf = MLPClassifier(random_state=42)).

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?

Run the individual classifiers 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. Congratulations, you have just trained a blender, 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?

Part 2: Letter Recognition

Load the letter-recognition.data.csv file, and do the letter classifications. You are free to choose all the machine learning algorithms we have covered so far. Moreover, apply the ensemble learning covered in Chapter 7 to improve your classification results.

Data Set Information:

The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer

values from 0 through 15. We train on the first 16000 items and then use the resulting model to predict the letter category for the remaining 4000.

Attribute Information:

1. lettr capital letter (26 values from A to Z)
2. x-box horizontal position of box (integer)
3. y-box vertical position of box (integer)
4. width width of box (integer)
5. high height of box (integer)
6. onpix total # on pixels (integer)
7. x-bar mean x of on pixels in box (integer)
8. y-bar mean y of on pixels in box (integer)
9. x2bar mean x variance (integer)
10. y2bar mean y variance (integer)
11. xybar mean x y correlation (integer)
12. x2ybr mean of x * x * y (integer)
13. xy2br mean of x * y * y (integer)
14. x-ege mean edge count left to right (integer)
15. xegvy correlation of x-ege with y (integer)
16. y-ege mean edge count bottom to top (integer)
17. yegvx correlation of y-ege with x (integer)

Write your code using file name ensemble.ipynb, and submit this file to HuskyCT. Note you need to use Markdown to explain your approaches. Include discussions on the importance of the above 17 attributes. Also include charts and graphs whenever applicable. This can help the TA better grade your work.