***Student: Victor Pugliese***

***R#: 11492336***

***Course: Independent Study on Python Machine Learning for Petroleum Engineering Application (PETR 5000)***

***Self-Homework #7***

1. ***If you have trained five different models on the exact same training data, and they all achieve 95% precision, is there any chance that you can combine these models to get better results? If so, how? If not, why?***

We could use a voting ensemble to increase the accuracy of the predictions.

1. ***What is the difference between hard and soft voting classifiers?***

Hard voting classifiers use the predictions of the individuals base classifiers. On the other hand, soft voting classifiers use the probabilities estimation for each class made by the base classifiers.

1. ***Is it possible to speed up training of a bagging ensemble by distributing it across multiple servers? What about pasting ensembles, boosting ensembles, random forests, or stacking ensembles?***

It is possible to speed up training of a bagging, pasting ensembles and random forests by distributing it across multiple servers because each predictor is independent of the others. Booting ensemble use a sequential algorithm, so there is no increase of the speed if we use several servers. Stacking could use parallel computing only for the same level.

1. ***What is the benefit of out-of-bag evaluation?***

The benefit is that we can use those instance, that have not been using for training, for testing the models.

1. ***What makes Extra-Trees more random than regular Random Forests? How can this extra randomness help? Are Extra-Trees slower or faster than regular Random Forests?***

Extra-Trees use a random subset of features for each tree and use a random threshold for each feature. This randomness help because the variance of the prediction is reduced, however the bias is increased. They are faster to train than regular Random Forest.

1. ***If your AdaBoost ensemble underfits the training data, what hyperparameters should you tweak and how?***

We could increase the number of estimators or reduce the regularization hyperparameters.

1. ***If your Gradient Boosting ensemble overfits the training set, should you increase or decrease the learning rate?***

We should decrease the learning rate.

1. ***Load the MNIST data (introduced in Chapter 3), and split it into a training set, a validation set, and a test set (e.g., use the first 40,000 instances for training, the next 10,000 for validation, and the last 10,000 for testing). Then train various classifiers, such as a Random Forest classifier, an Extra-Trees classifier, and an SVM. 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?***

See file: HML\_Chap07\_Exercise\_08

1. ***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. 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?***