Assignment 6

Applied Machine Learning

In this assignment we will generate an ensemble of primitive classifiers and compare their performances to the regular classifiers.

Classification of mushrooms, edible or poisonous. Download the `mushroom_dataset.csv` dataset file from the module content. Load the dataset in your model development framework, examine the features to see they are all nominal features. The first column is the class which represents the mushroom is poisonous or not. Apply necessary pre-processing such as nominal to numerical conversions (e.g. `pd.get_dummies`). Make sure sanity check the pipeline and perhaps run your favorite baseline classifier first.

1. [10 pts] Report 10-fold CV performances of `GaussianNB`, linear SVC (use `SVC(kernel='linear', probability=True)`), `MLPClassifier`, and `DecisionTreeClassifier` with default parameters. Now report the `RandomForestClassifier` performance too.

2. [10 pts] Generate an ensemble of 100 classifiers for each of the four classifiers in Q1. stored as a list. In order to create weak ensemble members, set the neural network hidden sizes to (3, 3), max iterations to 30, and tolerance to 1e-1. Set the decision tree parameters to max depth of 5 and max features of 5. We will evaluate these four ensemble classifiers.

   For each of the ensemble, report the first classifier performance in the ensemble.

3. [20 pts] Write a function `ensemble_fit()` to receive the ensemble (i.e. one of the lists in Q2.) and train on one of the subsets of the training data (e.g. `random.sample` can generate a subset). So each classifier will see only a different subset of the training dataset, also called as subsampling the input data for training. (Use all features in the subsample)

4. [20 pts] Write a function `ensemble_predict()` to receive the trained ensemble (i.e. one of the lists in Q3.) and test on the input. Use a voting scheme such as a histogram on the returned predictions by `c.predict()` by each of the weak classifier. The final prediction should be the `np.argmax()` of those counts. (Note that `c.predict_proba()` should have better results.)

5. [20 pts] Report 10-fold CV performances of the ensembles with a subsample ratio of 0.1. Compare to a regular decision tree (same subsample ratio). Now repeat these for subsample of 0.001.

6. [20 pts] Report and plot 10-fold CV performances of the ensembles for the training subsample ratios of (0.0005, 0.001, 0.005, 0.01, 0.03, 0.05, 0.1) on the same graph.

   Add the regular classifiers to the plot with same subsample ratios. (Hint: pass the regular classifier to the same ensemble CV in a list of one element. Same script can be used for this entire step)

   Report your detailed observations.