HW3 – Supervised Machine Learning

This homework will use the California housing data available on sklearn. The dataset does not require any cleaning. We have reserved 20% of the data for testing and 80% of the data for training. Please do not change the first code block. For all questions:

- Use "r2" as the assessment metric and use (`y_train`, `y_test`) as target for all regression task
- Use F1 score as the assessment metric and use (`y_bool_train`, `y_bool_test`) as target for all classification tasks.
- All cross validation should be 5-fold cross validation.

Some of the task require a long running time, please start working on this assignment early.

Q1. Basic Regression (30pt)

Perform meta-parameter search on Gradient Boosting Regressor to find a good combination of meta-parameters. Try to explore learning rate from 0.01 to 0.05 (step size: 0.01) and the maximum depth of the tree from 1 to 11 (step size: 2).

Q1a. (10pt) What's the best model parameter combination? What is its performance on the testing data?

Q1b. (10pt) Make line plot to show the r2 scores of all model parameters in a way that you can understand how r2 is related to the two meta parameters above. What trend do you see?

Q1c. (10pt) If we want to keep improving the performance of the model, what would you do?

Q2. Basic Classification (30pt)

Q2a. (10pt) As the first step, use cross validation to benchmark the performance on some basic model on the training set, including: logistic regression (set solver to liblinear), decision tree and KNN with default parameters. What's their CV performance on the training data?

Q2b. (20pt) Tune parameters of decision tree so that the F1 score after cross validation is at least 0.86 on the training set. You can tune any parameter but keep `max_feature` as default value. For the model with the best meta parameter, what's the F1 score on the testing dataset? Please show complete code on how you narrow down to the final parameters. Only providing the final model will lose points.

Hint: you cannot change the model, but you can try more meta parameters; try larger ranges; use smaller dataset first. Read the documentation to understand what are the parameters that you can tune; also read the slides to understand what they mean.

Grading will follow: 20pt if F1>= 0.86, 15pt if F1>=0.859, 10pt if F1>=0.855, 5pt if F1>=0.85,

Q3. Ensemble (40pt)

We will make some ensemble models and see how they improve the results. We will use some new models not covered in the lecture. Please read and study the documentation at https://scikit-learn.org/stable/modules/ensemble.html. Their usage is similar to what we have covered.

Q3a. (10pt) Construct a voting classifier containing 11 base classifiers, where all base classifiers are the decision tree model with parameters you got from Q2b. Please add `max_features='sqrt'` to your base classifiers. What's the CV performance of this ensemble classifier on the training data compared to the one you get from Q2b?

Q3b. (10pt) Add 11 KNN classifiers (with default parameters) to the above voting classifier (so totally 22 base classifiers now). What's the resulting performance? Explain the result.

Q3c. (10pt) Build a bagging classifier where the base model is the one you get from Q2b. Use grid search to find the best value of `max_features` from 0.1 to 0.9 (0.1 step). What's the best value for this meta parameter? With this parameter, what's the performance on the testing data? Please use the model at

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html

Hint: You can specify `max_features` either from the bagging classifier or from the decision tree. Specifiying from the bagging classifier is the easiest when doing GridSearch.

Q3d. (10pt) There are some packaged ensemble models that we can use directly without the need to specify tree models. Some of them were mentioned in the lecture. Read relevant documentation, select 5 of them and test their performance using CV on the training set.

Submission Instructions

Put your code in the template notebook as provided here: HW3.ipynb. Please submit two files:

1) The ipynb file. Make sure every question is executed with the printed results. Make sure all explanations are printed.

2) Export your code and results in a separate html document. After you are done with the ipynb file, go to File --> Download/Export As --> HTML in Jupyter notebook.

Submit **two documents** using the following naming convention. Please submit them directly **without zipping them.**

CS 396 Introduction to the Data Science Pipeline

```
HW3.ipynb
HW3.html
```