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 tasks 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 the 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?

[First implement the function Q1o(), it takes no input, and returns the gridsearchcv object.

Then we can store the object in the variable gbr\_model by using gbr\_model = Q1o().

Then implement the function Q1a(), it takes no input, and returns the tuple of the best model parameter and the scoring on testing data. ]

**Q1b. (10pt)** Make a 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?

[Put your code for plotting in the function Q1b(), calling it will show the plot.

Put your findings in the function Q1b\_finding()]

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

[Put your findings in the function Q1c\_finding().]

**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 mean CV performance on the training data?

[Put your code in the function Q2a(), it takes no input and returns a list of tuples with the model s and CV performance. E.g. [(model'A',0.9)('modelB',0.8),('modelC',0.6)]

]

**Q2b. (20pt)** Tune parameters of decision tree so that the F1 score after cross validation is at least 0.85 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.

[Put your tuning process in the function Q2bo(), it should include at least two rounds of gridsearchcv, and you may print initial gridsearchcv results and include comments on how you refine your range of search, the function will return the final gridsearchcv object.

And we will store it in the variable dt\_model by assigning dt\_model=Q2bo()

For function Q2b(), return the tuple of best model parameters, the mean cv score with that best parameter model, and the scoring of the best model on testing data.

]

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>= 0.85, 15pt if F1>=0.84, 10pt if F1>=0.82, 5pt if F1>=0.80,

**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 (<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html> ) containing a set of 5 random forest classifiers. All of the classifiers should use max\_depth as 5 but the number of estimators should be [5, 10, 15, 20, 25] respectively. What’s the mean CV performance of this ensemble classifier on the training data?

[Put your code in the function Q3a(), it returns the mean CV score of the voting classifier on training data.]

**Q3b. (10pt)** Construct a voting classifier that combines five KNN classifiers. Each KNN classifier uses the default parameters except for the number of neighbors, which takes the values [1, 2, 3, 4, 5]. Evaluate and report the resulting performance (mean CV score) of this ensemble model.

[Put your code in the function Q3b(), it returns the mean CV score of the voting classifier on training data.]

**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. Specifying from the bagging classifier is the easiest when doing GridSearch.

[Put your code in the function Q3c(), it returns a tuple of the best parameter of max\_features and the scoring on testing data.]

**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.

[Put your code in the function Q3d(), it returns a list of tuples with model and the corresponding mean CV score. ]

Submission Instructions

Work on your code in the template notebook as provided on the Canvas assignment page. After you are done, download the file (HW3.ipynb) directly and submit it to Canvas.

Make sure every question is executed with the printed results. Make sure all explanations are printed.

Double-check on if the submission is indeed successful.