Introduction to Machine Learning

Assignment 2

Azure note book link –

https://omkarpandit-oap338.notebooks.azure.com/j/notebooks/Omkar_Pandit_Lab_Assignment2.ipynb

Q1. Construct Datasets for Training and Evaluation [5 points]

a. Load a real dataset of your choice that is designed for classi_cation but was not used in class; e.g., from sklearn.datasets, Kaggle, or your own data. (Code)
Dataset used – Wine database

b. Create a 80/20 train/test split of the dataset. (Code).
➔ Number samples in training set - 142
Number samples in testing set – 36

Q2. Optimize Hyperparameter(s) for Each Classification Model [60 points]:

When evaluating each model in this problem, perform stratified 10-fold cross-validation on the training dataset and use the “accuracy" measure to assess performance.

a. Decision tree: find the optimal hyperparameters for the split criterion (i.e., test \gini" and \entropy") and tree depth (i.e., test at least 7 different values) when training a decision tree. Report the optimal hyperparameters found and how many hyperparameter combinations you tested in total. (Code and Write-up)

1. Decision tree with Gini split criterion –
After calculating all cross-validation scores, best score has been recorded for depth – 7.
Best validation Score at depth= 7 → 0.89
Test Score with best set of parameters at depth = 7 → 0.89

2. Decision tree with Entropy split criterion –
After calculating all cross-validation scores, best score has been recorded for depth – 6.
Best validation Score at depth= 6 → 0.93
Test Score with best set of parameters at depth = 6 → 0.94

Conclusion – So, based on 'Test Score' of 'Gini' and 'Entropy', as Test score from Entropy model is greater than Test score for Gini model. Entropy model performed better at depth = 6 than Gini model at depth = 7.

No. of hyperparameter combinations – 7 combinations for deciding maximum depth of a tree for each Gini and Entropy tree. So, in total 14 combinations of maximum depth hyperparameter on 10 folds cross-validation set.

b. K-Nearest Neighbors (K-NN): find the optimal hyperparameters for the distance metric (i.e., test \Euclidean" and \Manhattan") and number of nearest neighbors (i.e., test at least 7 different values) when using k-Nearest Neighbors. Report the optimal hyperparameters found and how many hyperparameter combinations you tested in total. (Code and Write-up)

1. K-nearest Neighbor with Manhattan distance criterion –
   After calculating all validation scores, best score has been recorded for no. of neighbors – 6.
   Best validation Score for no. of neighbors = 6 → 0.76
   Test Score with best set of parameters for no. of neighbors = 6 → 0.94

2. K-nearest Neighbor with Euclidean distance criterion –
   After calculating all validation scores, best score has been recorded for no. of neighbors – 4.
   Best validation Score for no. of neighbors = 4 → 0.76
   Test Score with best set of parameters for no. of neighbors = 4 → 0.92

   So, based on Test scores of KNN Manhattan vs KNN Euclidean, KNN Euclidean is better than KNN Manhattan.

   No. of hyperparameter combinations – 7 combinations for deciding optimal number of neighbors for each KNN Manhattan and KNN Euclidean. So, in total 14 combinations of number of neighbors on 10 folds cross-validation set.

c. Support Vector Machine (SVM): find the optimal hyperparameters for the poly-nomial degree, kernel bandwidth (i.e., gamma), and regularization parameter (i.e., C) when training a kernel SVM with a polynomial kernel. You must evaluate all possible combinations of at least 2-degree values (for the polynomial degree), at least 4 gamma values, and at least 4 C values. Report the optimal hyperparameters found and how many hyperparameter combinations you tested in total. (Code and Write-up)

SVM
Selected hyperparameters for SVM – Polynomial Degree(degree), Kernel Bandwidth (curGamma), Regularization parameter (curC)
Selected values for 'degree' – 3,4,5
Selected values for 'curGamma' – 0.001, 1, 10, 100
Selected values for 'curC' – 0.001, 1, 10, 100
Best values for hyperparameters –
Degree = 3; curGamma = 0.001; curC = 1

No. of hyperparameter combinations – 3 combinations for deciding optimal number of degrees, 4 combinations for deciding best value for curGamma, 4 combinations for deciding best value for curC.
So, total no. of combinations = 3*4*4
= 48

Q3. Comparative Analysis of Optimized Classification Models [35 points]

C. Report the predictive performance on the test dataset for each of the four models from parts (a) and (b) with respect to each of the following evaluation metrics: accuracy, precision, and recall.

| Model | Gini Tree | Entropy Tree | KNN Manhattan | KNN Euclidean | SVM | Gaussian NB |
|---|---|---|---|---|---|---|
| Precision | 0.89 | 0.95 | 0.95 | 0.93 | 0.97 | 0.95 |
| Recall | 0.89 | 0.94 | 0.94 | 0.92 | 0.97 | 0.94 |
| Accuracy | 0.89 | 0.94 | 0.94 | 0.92 | 0.97 | 0.94 |
| F-1 | 0.89 | 0.94 | 0.95 | 0.92 | 0.97 | 0.94 |
| | | | | | | |

Write a discussion analyzing and comparing the performance of the four models from parts (a) and (b). For example, which method(s) perform the best and why do you think so? Which method(s) perform the worst and why do you think so? What do the different performance metrics tell you about the results? Your discussion should consist of two to three paragraphs. (Write-up).

Referred the table 3 C. to compare all the models. So, Test score metric can be a good metric to decide best possible model for our dataset. So, initially in order to decide or to find best combination of hyperparameter values for decision tree model, I have selected Entropy tree model over Gini tree model based on its high accuracy score/test score. Similarly, I have eliminated KNN Euclidean model and have selected KNN Manhattan model based on better accuracy score. Now, for SVM, I have tried different combinations of values of multiple hyperparameters like polynomial degree, Kernel Bandwidth, Regularization parameter. So, after comparing 48 (3*4*4) combinations of

hyperparameter values, I have selected the best combination. The model was selected based on its test score.

Now, if we compare precision and recall values of all four models then, we can observer in my case both precision and recall are highest in case of SVM. So, I can conclude that, in my case for wine data, SVM model will give better results for the selected hyperparameter values. It is also proved that SVM works well on small size datasets. In contrast, as precision and recall value for decision tree with Gini criterion is lowest, it is the worst performing model for wine data. So, Gini model is worst performing as it gave lowest accuracy, precision, recall scores.