Hyperparameter Optimization

Introduction:

In this assignment, I am going to optimize several ML models using Bayesian optimization to find the best and optimum model for wine quality dataset. I am going to select several ML algorithms, optimize them for their hyperparameters, using them to predict the results, and comparing the results to find out which one is showing better performance as well as providing insight for optimization process. Here, I used the regression learners including k-Nearest Neighbors (hyperparameters: k and distance), Random Forest (hyperparameters: num.trees, min.node.size, and mtry), Support Vector Machine (hyperparameters: cost, epsilon, and gamma), and Gradient Boosting (hyperparameters: nrounds and max_depth). Root Mean Square Error was used as objective function and calculated for each iteration to find the model with minimum RMSE.

Dataset Description:

I am using the data from white wine which has 12 features and 4898 observations. Features include fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality. I used all of the mentioned attributes to predict wine quality which has a numeric scale between 0 to 10. The dataset has no missing value. A summary of the dataset is presented in table 1.

Table 1. statistical summary of the white wine dataset

Feature	Min	Mean	Max	SD
Fixed acidity	3.80	6.85	14.20	0.84
Volatile acidity	0.08	0.27	1.10	0.10
Citric acid	0.00	0.33	1.66	0.12
Residual sugar	0.60	6.39	65.80	5.07
Chlorides	0.01	0.04	0.34	0.02
Free sulfur dioxide	2.00	35.31	289.0	17.00
Total sulfur dioxide	9.00	138.4	440.0	42.50
Density	0.98	0.99	1.04	0.003
pH	2.72	3.19	3.82	0.15
Sulphates	0.22	0.49	1.08	0.11
Alcohol	8.00	10.51	14.20	1.23
Quality	3	5.87	9	0.88

Experimental Setup:

I used R programming language and the libraries such as mlr3 (to implement ML tasks), mlr3learners (to train the models), mlr3tuning (to tune the hyperparameters), mlr3mbo (to implement Bayesian Optimization), mlr3viz (to visualize the results), readr (to read the data), caret (to split the data), and dplyr (to manage the data). I decided to fit the regression models to predict the wine quality, therefore, regression learners including k-Nearest Neighbors ("regr.kknn"), Random Forest ("regr.ranger"), Support Vector Machine ("regr.svm"), and Gradient Boosting ("regr.xgboost") were used in this exercise. The hyperparameters, related value ranges, and optimum value (results after optimization) for each ML model are presented in table 2. The Bayesian optimization approach was used in this practice to minimize the objective function (which is RMSE).

Table 2. Hyperparameters and related value range and best value for ML models

Model	Hyperparameter	Range	Optimum value
k-Nearest Neighbors	k	1-20	11
	distance	1-20	1.03
Random Forest	num.trees	100-1000	667
	min.node.size	1-20	1
	mtry	1-10	3
Support Vector Machine	cost	0.1-10	1.99
	epsilon	0.01-2	0.16
	gamma	0.01-2	0.41
Gradient Boosting	nrounds	10-300	30
	max_depth	1-20	15

To tackle this problem and select the best ML algorithm after optimization process, I divided the data into two sets (training and validation). I applied 5-fold cross validation to training dataset and Bayesian approach to optimize with 100 iterations for each model. At each iteration, the RMSE value was calculated and the hyperparameters with the lower RMSE value were selected. Afterward, I predicted the result for the validation set using optimized hyperparameters and again calculated the RMSE. Thus, we have RMSE for optimized model (training using 5-fold CV) and RMSE for validation to compare the models together. Considering that the validation set prevents bias and overfitting. Calculation of the error just for train and test sets are not enough and can cause bias in our choice.

Results:

The results of the optimization process for training dataset (5-fold CV) were presented in figure 1. This plot illustrated the RMSE for each iteration and model. We can conclude that the Random Forest model is showing better performance and Gradient Boosting is the second best based on this figure.

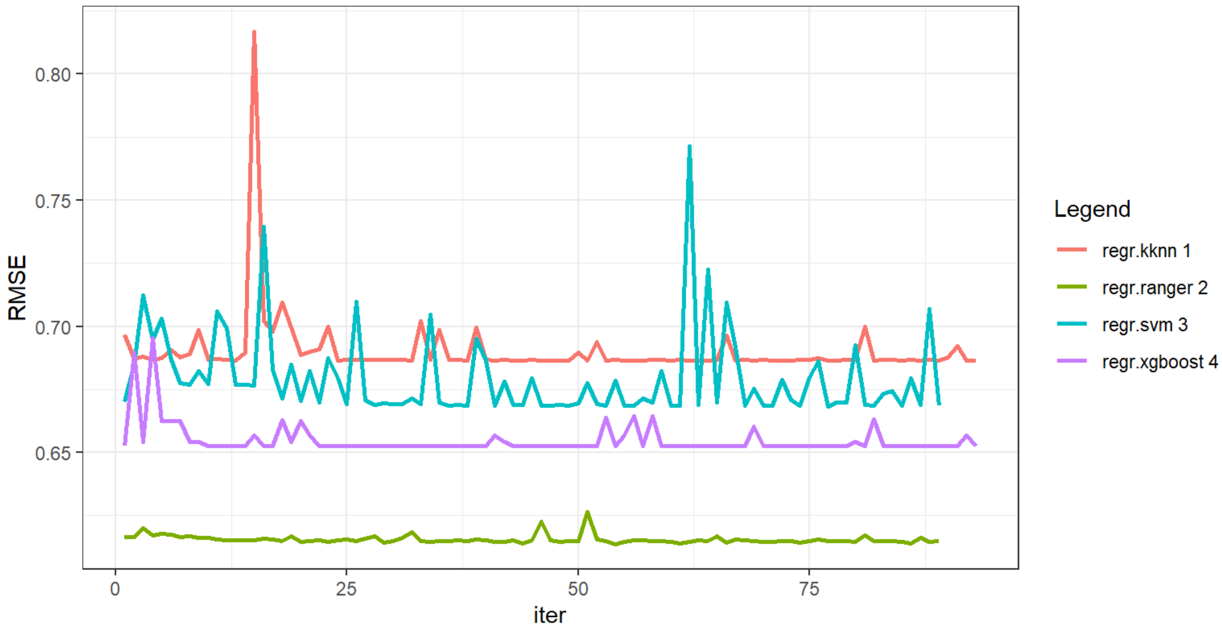

Figure 1. Optimization process for ML models.

The results of the best run after optimization for training (5-fold CV) and validations using optimum hyperparameters were presented in table 3.

Table 3. The RMSE for the training and validation sets using optimum hyperparameters.

Model	RMSE for 5-fold CV	RMSE for validation
k-Nearest Neighbors	0.68	0.67
Random Forest	0.61	0.60
Support Vector Machine	0.67	0.66
Gradient Boosting	0.65	0.64

We know the lower RMSE represents the better fit. Based on table 3, Random Forest shows better performance with RMSE = 0.60 in comparison with the other models. Despite algorithm selection exercise, RMSE of 5-fold CV and validation were approximately equal for all the models. All in all, we can see slight improvement in the results after optimization of the models compared to algorithm selection exercise.