Practical Machine Learning: Pipeline Optimization

Milana M. Wolff

May 08, 2024

1 Introduction

In this assignment, we perform pipeline optimization using the wine quality dataset and the primary tumor dataset.
The wine quality dataset is a widely used dataset contains a variety of physicochemical input features, such as wine density and acidity, along with expert ratings for red Vinho Verde wines [1] [2].
The primary tumor dataset is another widely used dataset containing categorical patient data and a binary target feature [3].
We approach the pipeline optimization problem by selecting a scaler (from four methods or passthrough) for the wine quality dataset or an encoder for the categorical primary tumor dataset (from two possible encoding schemes), whether or not to use a feature selector, and by selecting an estimator from a set of classifiers, each with a number of possible different hyperparameter configurations evaluated via Bayesian optimization. We perform nested resampling as well.

2 Dataset Description

The wine quality dataset used for this assignment contains physicochemical quantitative input features and sensory quantitative output features (i.e., an expert wine score) for the red variant of the Portuguese "Vinho Verde" wine. The dataset includes 1599 observations and eleven input features, including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol content. According to the UC Irvine Machine Learning Repository website, "the classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones)", with a total of 1319 observations rated as 5 or 6 and a mere 28 observations rated with the highest and lowest scores (3 and 8). This robust dataset includes no missing values to be imputed. We use the eleven listed input features to predict the wine quality measurement as the target [1] [2].
The primary tumor dataset contains 17 categorical features including the age

Encoder	**Scaler**	**Selector**	**Estimator**
OneHotEncoder(), OrdinalEncoder()	StandardScaler(), MinMaxScaler(), Normalizer(), MaxAbsScaler(), passthrough	VarianceThreshold(), passthrough	SVCClassifier(), RidgeClassifier(), KNeighborsClassifier(), DecisionTreeClassifier(), BaggingClassifier(), RandomForestClassifier()

and sex of the patient, histologic type of tumor, degree of difference, and location of tumor (bone, lung, liver, brain, etc.) as the input features, and binaryClass as the target feature, obtained by converting the multi-class target feature in the original dataset to a two-class nominal target feature by re-labeling the majority class as positive and all others as negative. Some values in the dataset are listed as '?', or unknown. There is 1 missing attribute for the `sex` feature, 67 missing attributes for `histologic-type`, and 155 for `degree-of-difference`. There are 339 instances in this dataset [3].

3 Experimental Setup

We use Python version 3.10.9 running on the `teton-gpu` partition of the Beartooth cluster. (Using Python 3.10.12 (GCC 11.4.0) in a Jupyter/interactive Python notebook on Google Colaboratory repeatedly timed out). We use classifiers, preprocessors, and other pipeline elements from the `scikit-learn` package. We use `GridSearchCV` with 3 outer folds and 5 inner folds for nested resampling over the entire pipeline parameter space, with Bayesian optimiation (implemented via `BayesSearchCV`) applied to search for optimal hyperparameter settings of individual classifiers in the last stage of the pipeline with 3 folds. For the Bayesian search for each estimator, we use 3 iterations, with 3 points sampled in parallel and 3 jobs due to computational constraints.
The pipeline includes a scaler, a selector, and a final estimator component in the case of the wine quality dataset, which uses numeric input features, and an encoder, a selector, and a final estimator component for the primary tumor dataset, which uses categorical input features. As the features in the wine quality dataset are real-numbered values, we do not use an encoder on this dataset. The possible components are tabulated below.

The hyperparameter search spaces for each estimator are tabulated below.

The hyperparameter search spaces for the selector (both datasets) and the scalers with sufficient hyperparameter options to optimize (wine dataset) are in-

Classifier	Support Vector Classifier	K Neighbors Classifier	Ridge Classifier	Decision Tree Classifier	Bagging Classifier	Random Forest Classifier
Hyperparameter Search Space	'C' : (1e-6, 1e+6, prior='log-uniform'), 'gamma' : (1e-6, 1e+1, prior='log-uniform'), 'degree': (1, 8), 'kernel' : 'linear', 'poly', 'rbf'	'n_neighbors' : Integer(1, 100, prior = 'log-uniform'), 'algorithm' : 'ball_tree', 'kd_tree', 'brute', 'leaf_size' : (1, 50)	'tol' : (0.01, 0.1, prior = 'log-uniform'), 'solver' : 'svd', 'cholesky', 'sparse_cg', 'saga', 'lsqr', 'alpha' : (0.1, 1.0, prior = 'log-uniform')	'max_depth' : (1, 10, prior = 'log-uniform'), 'max_features' : None, 'auto', 'sqrt', 'log2', 'min_samples_split' : (0.1, 1.0, prior = 'log-uniform')	'n_estimators' : (50, 500, prior = 'log-uniform'), 'max_features' : (0.1, 5, prior = 'log-uniform')	'n_estimators' : (100, 100000, prior = 'log-uniform'), 'criterion' : 'gini', 'entropy', 'log_loss', 'max_depth' : (1, 10, prior = 'log-uniform')

cluded below as well. We use `BayesSearchCV()` with 3 iterations (due to time and memory limits) for each of these. Other scalers used either do not have sufficient hyperparameters to optimize (for instance, some only offer whether or not to do inplace scaling, which is irrelevant to hyperparameter optimization results), and both encoders used in the pipeline optimization for the primary tumor dataset do not have scoring functions, are not estimators, and therefore cannot easily be optimized using a Bayesian search.

We evaluate the results using balanced accuracy, selected primarily because the wine quality dataset is highly imbalanced.

Standard Scaler (scaler)	Normalizer (scaler)	Variance Threshold (selector)
”with_std” : True, False	”norm” : ’l1’, ’l2’, ’max’	”threshold” : Real(0.0, 10)

4 Results

The best pipeline for the wine quality dataset used a MinMaxScaler(), no selector, and a random forest classifier. This pipeline, the best-performing of those used in the outer loop of nested resampling, achieved a balanced accuracy score of 0.2985. The generalization score over this outer fold was 0.262 ± 0.028. The relatively low performance may be attributed to the limited number of iterations of the Bayesian search. Using a pipeline with a OneHot encoding scheme, a VarianceThreshold selector, and a Ridge Classifier as the estimator (optimized via Bayesian search), a generalization score of 0.767 ± 0.067 was achieved on the primary tumor dataset. Additional results are pending. Results are pending for updated code on the wine quality dataset; this report will be updated when the job running on Beartooth completes. [Update: Still hasn’t completed.]

5 Code

`https://github.com/COSC5557/pipeline-optimization-mwolff2021-1`

6 References

References

[1] Paulo Cortez, A. Cerdeira, F. Almeida, et al. “Wine Quality”. In: (2009). DOI: https://doi.org/10.24432/C56S3T.

[2] In: (). URL: http://www.vinhoverde.pt/en/.

[3] URL: https://www.openml.org/search?type=data&sort=version&status=any&order=asc&exact_name=primary-tumor&id=1003.