COMP5511-Assignment2

AI for Healthcare-Part 1

Wong Chi Fung (Student Id: 24012947g)

Description

In this assignment, Light Gradient-Boosting Machine (LightGBM) and traditional machine learning models have been trained to detect a specific health condition by analyzing medical data with validation accuracy. LightGBM is characterized as a gradient boosting framework that uses tree-based learning algorithms with high accuracy, low memory usage and short training time (1). Feature engineering and hyperparameter optimization were performed to enhance algorithm's predicting power.

Feature Engineering

Features important for classification were identified by two distinct models: F-test and mutual information (MI). F-test can identify significant linear relationships between variables and targets, whereas MI can capture all kinds of relationships (2). It was anticipated that the features chosen by either of the two distinct feature importance methods would capture the key information that help predicting health condition.

Furthermore, the selected features were checked for collinearity. Collinearity between variables is usually undesirable as it possibly increases variance of the estimate and leads to over-fitting problem. If there is a strong collinearity between them, indicator with the larger index would be dropped.

To deal with missing value in the data, each sample's missing values were imputed using the mean value from 3 nearest neighbors found in the training set. Their features should be similar if samples are close to each other. Then, the data was standardized to share a mean of 0 and a standard deviation of 1, which can help model converging efficiently.

Hyperparameter Optimization

An automated hyperparameter tuning library called FLAML has been used to adjust the hyperparameter of LightGBM to optimize its accuracy performance on validation data set (3). It uses a strategy called Frugal Optimization for Cost-related Hyperparameter (CFO), which begins searching from low-cost hyperparameter set and gradually move to high-cost region (4). It helps models converge into a local optimum with good accuracy rate using an economical configuration.

Algorithms Implementation

Baseline Model

Several traditional machine learning models derived from Scikit-Learn library served as the baseline in this task. Their corresponding code implementation is below:

Traditional Machine Learning Model	**Model in Scikit-Learn**
Support Vector Machine (SVM)	sklearn.svm.SVC
Logistic Regression (LR)	sklearn.linear_model.LogisticRegression
Decision Tree	sklearn.tree.DecisionTreeClassifier
Random Forest	sklearn.ensemble.RandomForestClassifier
Adaptive Boosting Classifier (AdaBoostClassifer)	sklearn.ensemble.AdaBoostClassifier
K-Nearest Neighbors (KNN)	sklearn.neighbors.KNeighborsClassifier
Naïve Bayes (NB)	sklearn.naive_bayes.GaussianNB
Multi-Layer Perceptron (MLP)	sklearn.neural_network.MLPClassifier

LightGBM and FLAML

It is simple to implement LightGBM using FLAML code, because it has been configured into class AutoML in FLAML library (5). Without calling the original library of LightGBM, LightGBM with default setting can be called using 'flaml.default.LGBMClassifier'. The required code to train and automatically optimize LightGBM using strategy CFO is only two lines as demonstrated below:

```
automl = AutoML(hpo_method='cfo')
automl.fit(train_X, train_y, X_val=valid_X,y_val=valid_y,task="classification",
max_iter=5000,train_time_limit=10,estimator_list=["lgbm"] ,metric="accuracy",seed=seed,verbose=1)
```

The best hyperparameter configuration found within 5000 iterations is below:

```
{'n_estimators': 14, 'num_leaves': 4, 'min_child_samples': 22, 'learning_rate': 0.517586978696495,
'log_max_bin': 7, 'colsample_bytree': 0.8197537630180586, 'reg_alpha': 0.0016373615477813118, 'reg_lambda':
1.529900789026623}
```

Data

The medical dataset comprised 500 patient data, including 70 medical indicators and a binary target variable. The dataset was divided into 3 subsets, including training set (300 samples), validation set (100 samples), and testing set (100 samples). All features were numeric data. In the following, all data analysis were performed in the training set to select and transform features for training model.

Function ‘f_classif’ and ‘mutual_info_classif’ from Scikit-Learn were used to compute F-test and mutual information (MI) for samples without any missing values, respectively (6). Features were selected if they had a p-value from the F-test below the predefined threshold of 0.05 (-log10(p-value) larger than 1.30) or if their MI score exceeded 0.03 (Figure 1). Out of 70 indicators, 14 were identified to be important and selected for subsequent processing.

In the subsequent correlation analysis, out of the 14 important features, none was highly correlated with each other under absolute correlation threshold of 0.8, suggesting that the features could provide different insights for machine learning model to learn (Figure 2). As a result, they were all preserved for machine learning.

To address the missing values in the dataset, each sample's missing values was imputed with the mean of the three nearest neighbors identified in the training set using Scikit-Learn function 'KNNImputer' (6). After imputation, the data was standardized for scaling.

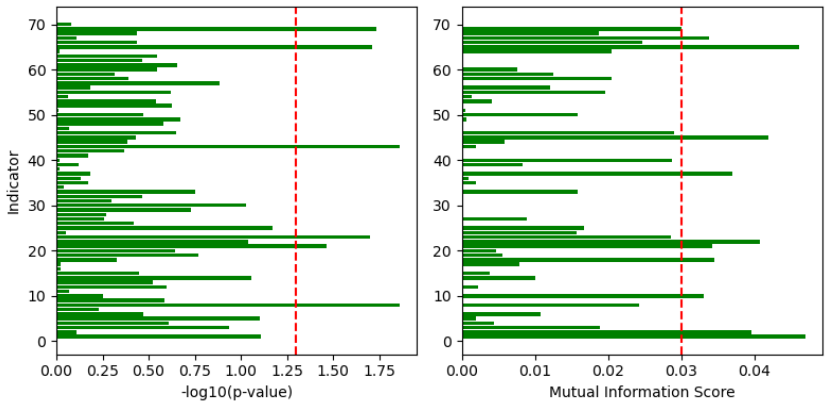

Figure 1. Comparison of -log10(p-value) from F-test (left) and mutual information (right) of all indicators
The corresponding thresholds for selection are depicted as red dash line

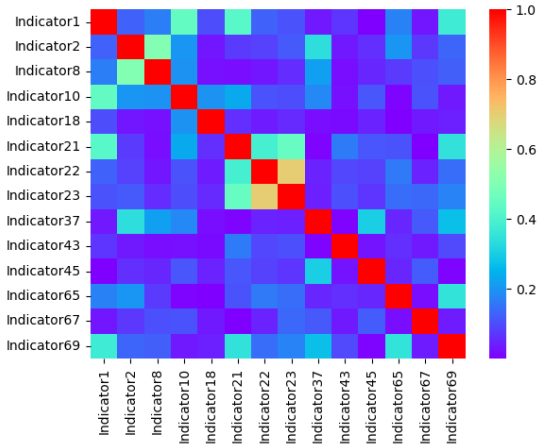

Figure 2. Heatmap of absolute correlation between the important indicators

Experimental Setting and Performance

I compared the performance of my model before and after optimization, as well as against baseline models, using various evaluation metrics, including accuracy, precision, recall, specificity, F1 score, and Area Under the Receiver Operating Characteristic Curve (AUC). The evaluation results indicate that the optimized LightGBM outperformed its default version and other traditional models in most tests, though it scored lower than the SVM and Random Forest in terms of specificity (Table 1).

Model	Accuracy	Precision	Recall	Specificity	F1	AUC
SVM	0.52	0.63	0.19	0.88	0.29	0.53
LR	0.51	0.57	0.25	0.79	0.35	0.52
Decision Tree	0.43	0.44	0.38	0.48	0.41	0.43
Random Forest	0.50	0.56	0.19	0.83	0.29	0.51
AdaBoostClassifer	0.57	0.62	0.44	0.71	0.52	0.58
KNN	0.48	0.50	0.33	0.65	0.40	0.49
NB	0.55	0.61	0.37	0.75	0.46	0.56
MLP	0.47	0.49	0.33	0.63	0.39	0.48
LightGBM	0.50	0.54	0.25	0.77	0.34	0.51
Optimized LightGBM	0.63	0.73	0.46	0.81	0.56	0.64

Table 1. Evaluation results of machine learning models on validation set. The score highlighted in yellow indicates the highest value for each metric.

Findings and Discussions

The Optimized LightGBM model stands out as the best performer, achieving the highest accuracy (0.63), precision (0.73), and F1 score (0.56). This suggests it effectively balances sensitivity and specificity while minimizing false positives. As the default LightGBM did not perform as well as its optimized version, it shows that the potential of LightGBM was effectively harnessed by FLAML and its optimizing algorithm CFO. Despite its success, its recall rate is still relatively low (0.46), even though it ranks among the highest, suggesting further feature engineering is needed to capture information of positive candidates. And SVM (0.88) and Random Forest (0.83) perform better in terms of specificity, suggesting strong capability in classifying negative cases. In the future, we can combine SVM, Random Forest and LightGBM to build a more generalized ensemble model.

References

1. *LightGBM.* [Online] https://lightgbm.readthedocs.io/.
2. Comparison of F-test and mutual information. *Scikit-Learn.* [Online] https://scikit-learn.org/1.5/auto_examples/feature_selection/plot_f_test_vs_mi.html.
3. Zhu, Chi Wang and Qingyun Wu and Markus Weimer and Erkang. *FLAML: A Fast and Lightweight AutoML Library.* s.l. : MLSys, 2021.
4. Huang, Qingyun Wu and Chi Wang and Silu. *Frugal Optimization for Cost-related Hyperparameters.* s.l. : AAAI, 2021.
5. *FLAML.* [Online] https://microsoft.github.io/FLAML/.
6. *Scikit-Learn.* [Online] https://scikit-learn.org/.