CMPUT 566

Final Project Report

Yongfan Wang 1730251

Introduction

The “Adult” datasets [1] have been widely discussed over the past few years. If we knew the determinants of income, or what factors can improve income, we can give correct suggestions to the people who want a higher salary. This project is going to find a better way to predict the income (> 50k or <= 50k) of a person with certain condition. In general, it is a two-category classification problem and is a good example for practicing machine learning.

Problem Formulation

Dataset Analysis

The “Adult” dataset, also known as “Census Income” datasets, is extracted from the U.S. Census database. It contains over 40000 instances and has been split into a training dataset with 32561 instances and a test dataset with 16282 instances. In total, there are 76.07% of instances with annual income less than 50k. 14 variables are used to predict the result, including age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country. Six of them are continuous variables, while the other 8 are discrete. There are also some missing values in the dataset.

Among these variables, “fnlwgt” is just the sequence number, “native-country” has little effect on annual income, and “education” is positively correlated with “education-num”, therefore, these variables are removed in the data pre-processing step.

Task Analysis

Both discrete and continuous variables with missing values are in the dataset. After evaluation, this project will use three algorithms to complete the prediction: Logistic Regression, Decision Tree [2] and Naïve Bayes [3]. The advantages of these three algorithms are as follows:

- Logistic Regression: easiest and classic, and quite useful for classification problem.
- Naïve Bayes: has strong support from mathematical theory, not complex and insensitive to missing values.
- Decision Tree: simple and interpretable, and suitable for datasets with missing values.

Meanwhile Linear Regression will also be used and show that it not suitable for classification problems.

Approaches and Baselines

This is a basic two-category classification problem with enough instances to train and test. To imply the training-validation-test infrastructure, 10% instances from “adult.data” are used for validation, and the rest of them are used for training. All instances from “adult.test” are used for test.

For three main models, the results with default settings from “sklearn” are set as baselines. Then GridSearchCV() function is used to imply a systematic parameter tuning process with cross validation to improve its reliability. All algorithms use “sklearn” to imply by few lines of function call.

Linear Regression

Linear Regression is not a good solution to classification problems, so everything is set to default and get the result.

Logistic Regression

Baseline

From sklearn official website, the main parameters of sklearn.linear_model.LogisticRegression() are set as follows:

- *penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*

But the program will get a warning "ConvergenceWarning: lbfgs failed to converge". After increase the *max_iter* parameter from 100 to 10000, this problem is solved. Function score() from sklearn is used to evaluate the accuracy.

Then the baseline of Logistic Regression:

```
Validation accuracy of Logistic Regression =  0.8233523739328051
Test accuracy of Logistic Regression =  0.8233523739328051
```

Tuning

Two main parameters are tuning according to the following grid to tune the model:

- *{'solver': ('lbfgs', 'newton-cg', 'saga'),*

  'penalty': ('l1', 'l2', 'none')}

Naïve Bayes

Baseline

Since Naïve Bayes has no hyperparameters that need tuning, only baseline need to be considered. GaussianNB() from sklearn.naive_bayes has been used for build the model.

In order to get a more average result, cross_val_score() from sklearn.model_selection has been used and set it to 10-fold cross-validation method.

Decision Tree

Baseline

The main parameters of sklearn.tree.DecisionTreeClassifier() are:

- **, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0*

And then the baseline of Decision Tree:

```
Validation accuracy of Decision Tree = 0.8151673319005219
Test accuracy of Decision Tree = 0.8159203980099502
```

Tuning

Four hyperparameters are tuning according to the following grid:

- *{'criterion':('gini', 'entropy'),*

  'max_depth':(None, 10, 15, 20),

  'min_samples_split':(2, 6, 9, 12),

  'min_samples_leaf':(1, 3, 6, 9),}

Evaluation Metric

Since 76.07% instances are people with income <= 50k, if simply predict everyone's income is <= 50k, the accuracy should be 76.07%. Therefore, every model should have at least 76% accuracy to beat the simple prediction, otherwise it is meaningless model. In other words, if the accuracy of the model is higher than 0.7607, it is considered as a successful solution to this classification problem.

Results

Linear Regression

```
Linear Regression accuracy =  0.2520024089647642
```

The result shows that linear regression is a very bad approach for this problem. It only has 25.2% accuracy which is far away from the metric.

Logistic Regression

```
LRs best training accuracy after tuning = 0.8249044618997677
LRs validation accuracy after tuning = 0.8268345102855389
LRs best parameters after tuning: {'penalty': 'l2', 'solver': 'lbfgs'}
LRs final accuracy = 0.8233523739328051
```

The best hyperparameters combination is l2-penlty with "lbfgs" solver, which is equal to the default setting. The best training accuracy after tuning is 0.8249 and final test accuracy after tuning is 0.8234. The accuracies are higher than the metric accuracy 0.7607, therefore logistic regression is a successful solution to this problem.

Naïve Bayes

```
Validation accuracy of Naive Bayes = 0.8108688977586737
Test accuracy of Naive Bayes = 0.8003193907008169
Validation accuracy of Naïve Bayes with 10-fold cv = 0.8025785747994337
Test accuracy of Naïve Bayes with 10-fold cv = 0.7989065283264178
```

The average accuracy of Naïve Bayes with 10-fold cross-validation is 0.7989. Although it is higher than the metric accuracy, it is not a good way to solve the problem compared to Logistic Regression. And after cross-validation, the accuracy drops a little. Therefore, Naïve Bayes is consider as a successful solution but not a good one.

Decision Tree

```
DT’s best training accuracy after tuning = 0.8572209602084235
DT’s validation accuracy after tuning = 0.8529321461467608
DT’s best parameters after tuning: {'criterion': 'entropy', 'max_depth': 10,
'min_samples_leaf': 6, 'min_samples_split': 9}
DT’s final accuracy = 0.8542472821079786
```

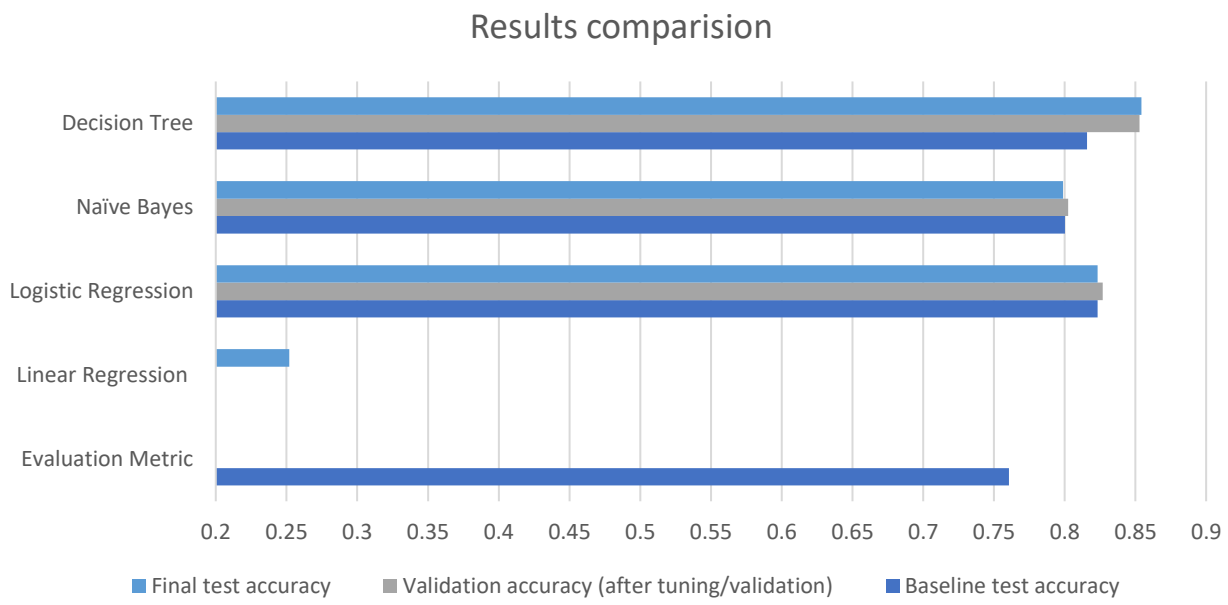
The best hyperparameters combination for decision tree is:

- {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 6, 'min_samples_split': 9}

The best training accuracy after tuning is 0.8572 and the final test accuracy after tuning is 0.8542. The Decision Tress method has the highest accuracy among these models, and it is much faster than Logistic Regression. Therefore, it is the best method among these four methods.

Final Results

The final results are shown in the table. Baseline, validation accuracy after tuning/cross-validation and final test accuracy are chosen to demonstrate the performance.



Obviously, Decision Tree model has improved after parameters tuning, and it becomes the best among all models.

Conclusion

Based on the above experimental results, it can be observed that the accuracy of the Logistic Regression model before parameters tuning is higher than the other models, but Decision Tree becomes the best model after parameter tuning, with 85.72% accuracy. Meanwhile, after cross-validation, the accuracy of Naïve Bayes has dropped a little. But in other classification problems, Naïve Bayes usually performed well. I think the reason why it has a poor performance in this problem is that Naïve Bayes model always

assume properties independent to each other. But in this dataset, this assumption does not hold. For example, marital-status is highly correlated with age.

After checking some related and similar work, I found that although Decision Tree is not complicated, it is quite useful. 85% accuracy is also a good result among other algorithms for this problem. I think it is because Decision Tree is able to deal with both the continuous and discrete variables, and it is capable of producing feasible and effective results on large data sources in a relatively short period of time. However, the shortcomings of Decision Tree are also obvious. It is likely to overfitting, and sometimes ignores the relations between properties. Therefore, although machine learning is a good technology for prediction, finding a suitable model for specific problem is also very important.

References

[1] Blake, C. L., Keogh, E., Merz, C.J.: *UCI Repository of Machine Learning Databases*. Irvine, CA : University of California, Department of Information and Computer Science. [https://archive.ics.uci.edu/ml/datasets/adult]

[2] Pfahringer, B., G. Holmes, and R. Kirkby. "Optimizing the Induction of Alternating Decision Trees." *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)* 2035: 477–87. Accessed April 18, 2022. doi:10.1007/3-540-45357-1_50.

[3] Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996