Classification of Income: A Machine Learning Study Comparing Parametric and Nonparametric Models

Bohan Song, Dongyang Wang, Wenjin Zhang

Introduction

This project aims to apply our knowledge about parametric and non-parametric machine learning methods to a classification problem. We want to study the annual salaries of adults differ, in terms of whether they have reached 50K per year. The parametric methods include Logistic Regression and Naive Bayes classifier; and the non-parametric methods are Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbor (KNN). We will evaluate these methods based on four metrics, accuracy, specificity, sensitivity, and Area Under Curve (AUC) for the ROC curve.

Data Processing

We obtained our data from Kaggle.[1] During data processing, we removed missing values. First, we have transformed all the column names into a suitable format for R by replacing dots with underlines. We realize that the column education and education number are 1-1 correspondence indicating one's education background.[2] Therefore, we decided to keep the education column while removing the education number column. Furthermore, the fact that the feature "country" is really sparse might cause convergence issues in models. We categorize all the countries into continents, Asia, Europe, North America, South America, and others. In total, we have split our data into a training set and a test set in 4:1 ratio. If a model needs to determine hyperparameters, we further hold out 20% in the training set for validation usage, which comes down to a 64% training set, a 16% validation set, and a 20% test set.

Modeling

The modeling of the data involves some accommodation for different models. For example, the KNN required changing the categorical variables into dummy variables. RF has normalized two heavy-tailed features, capital gain, and capital loss where most values are 0 and non-zero values are in the thousands. Also, the SVM has normalized the numeric data, so as to not to overly emphasize the variables that have large magnitudes.[3]

Furthermore, cross validation or the train-validation-test scheme have been used to determine the hyperparameter, such as the number of neighbors in KNN and the cost in SVM. After determining the best hyperparameter to use, we further modeled with the test set to evaluate our models.

---

[1] "Adult income dataset." *Kaggle*. See https://www.kaggle.com/datasets/wenruliu/adult-income-dataset
[2] "The Adult dataset." *University of Toronto*. http://www.cs.toronto.edu/~delve/data/adult/adultDetail.html
[3] "Is it essential to do normalization for SVM and Random Forest?" *Stack Exchange*. https://stats.stackexchange.com/questions/57010/is-it-essential-to-do-normalization-for-svm-and-random-forest

Results

Among other metrics, we have determined to choose accuracy, specificity, sensitivity, and the AUC. Admittedly, there are a lot of other metrics to choose from. But since we don't have a particular research question, we do not have a single metric that can be the most important. Therefore, we have included the four that are commonly used in machine learning.

	Accuracy	Sensitivity	Specificity	AUC
Logistic	0.8474295	0.6843709	0.9029342	0.9090615
Naive Bayes	0.8213378	0.8193501	0.8382353	0.7682861
Decision Tree	0.8615810	0.7572559	0.8892308	0.8913589
KNN	0.7980	0.8210	0.7970	0.5252
Random Forest	0.8625608	0.7479920	0.8949234	0.9160436
SVM	0.8533997	0.7762109	0.8703804	0.8991008

Table 1. Metrics of parametric (green) and non-parametric (blue) methods

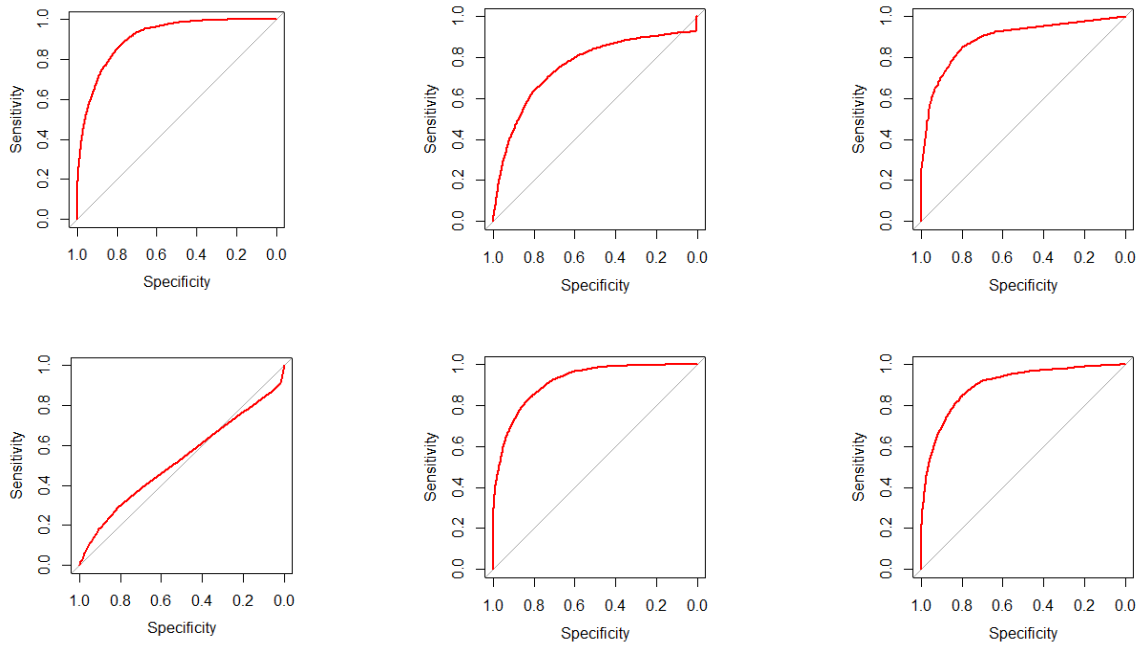

Figure 1. ROC curve plot, first row Logistic, NB, DT (left to right), second row KNN, RF, SVM

Based on the above results, the KNN model barely has any predictive power, since about 75% of the people in the test set have income less than 50K. Random Forest has the highest AUC (91.6%) while KNN has the highest sensitivity(82.1%). Notice that runtime for KNN is ten times more than other methods such as Naive Bayes and yet their difference in sensitivity is negligible. If prediction on whether an adult earns an annual salary of 50K is prioritized in research, we will recommend using Naive Bayes rather than KNN because KNN has the least performance in the other three metrics. If the overall prediction is more essential, then we will recommend Logistic Regression and Random Forest. Ranger package and glm package in R are really good tools to reduce runtime. As a result, Random Forest and Logistic Regression are the fastest among all the methods.

Note that Neural Network has been attempted, but discarded because the running time is too much, taking approximately 4-5 hours. But its performance is notable, close to that of logistic regression.

As a result, the table of metrics can be used to determine the best model that we may use when making future predictions.

References

"Adult income dataset." *Kaggle*. See https://www.kaggle.com/datasets/wenruliu/adult-income-dataset

"Is it essential to do normalization for SVM and Random Forest?" *Stack Exchange*. https://stats.stackexchange.com/questions/57010/is-it-essential-to-do-normalization-for-svm-and-random-forest

"The Adult dataset." *University of Toronto*. http://www.cs.toronto.edu/~delve/data/adult/adultDetail.html

Team Contribution

Bohan has contributed to the Naive Bayes classifier and Random Forest. He writes his own implementation of Naïve Bayes with the classic assumption that all the features are independent. The distribution for every categorical variable is constructed based on multinomial distribution, whereas the distribution for every numerical variable is constructed based on kernel density estimation. Moreover, he builds a random forest using the ranger package which tunes the parameter, depth of the tree and mtry, on a held-out validation set.

Wenjin has contributed to the Logistic Regression and Decision Tree method. He selects best parameters based on forward and backward algorithms and uses validation set to figure out the best threshold for logistic regression. For the decision tree, not only does he build the model with optimum tree depth, but also use the package rpart.plot to draw the tree.

Dongyang has contributed to KNN, SVM, and Neural Network. In the models, he normalized data, implemented cross validation for parameter tuning, and evaluated the models. He was also responsible for drafting the R Code and standardizing the format.

For the code, consult the following GitHub link
https://github.com/dongyangwang30/Classification-of-Income-Parametric-and-Nonparametric-Models