INFO526 Final Project
Meng Wu

Tools: Python
Codes: https://github.com/wum5/INFO526_codes/tree/master/homework

Question: Out of the seven algorithms applied on the Adult data, which one is the best in terms of prediction accuracy and time efficiency?

Methods:

I implemented all the seven algorithms (i.e. Naïve Bayes, Logistic Regression, Decision Tree, K-Nearest Neighbors, Support Vector Machine, Random Forest, and Adaboost) on the adult data. In comparison with the number of examples in the entire set, the number of examples with missing data is relatively small. Thus, I decided to remove all the entries with missing values. In addition, I performed some pre-processing steps before performing particular algorithm. For example, I scaled down all the continuous features between 0 and 1 to prepare the input data for Nearest Neighbor, because some insignificant variables with larger range would be dominating the objective function in this classifier. I also converted all categorical variables to numerical variables (e.g. Male=1; Female=0), because these categorical features had string values cannot be fed into Logistic Regression.

For each algorithm, I used the 10-fold cross-validation on the training examples (split into training set and tuning set) to adjust the hyper-parameters, thus I could better improve and evaluate the model's performance. For example, I combined the cross-validation with recursive feature elimination in logistic regression to find optimal number of features used for subsequent learning and prediction. This step might be important because removing irrelevant features could enhance generalization by reducing overfitting. With the cross-validation, I could also search for the optimal depth of the Decision Tree and the value of *k* in Nearest Neighbors. This step was important because 1) larger *k* produces bias while smaller *k* produce variance; 2) higher depth produces variance while lower depth produce bias. For Support Vector Machine (SVM; the Radial Basis Kernel), I searched for the optimal parameter values of *C* and *gamma* that both have effect on the simplicity of the decision surface. Since large *C* and *gamma* values would make the SVM very computationally expensive, I constrained both of the two values to be in the range of 10^{-3} to 10^{3}. Besides, I randomly sampled ten subsets of training datasets and testing datasets (each contains ~1500 entries) for cross-validation in SVM in to reduce the time during the training step. For Random Forest, I tried to find the optimal number of sampled features and generated trees to improve the predictive accuracy and control over-fitting. For Adaboost, the optimal values of learning rate and the number of trees were searched to control the contributions of weak learners and delivers improved prediction accuracy.

Table 1. Summary of prediction accuracy and run time using seven different algorithms

Algorithm	Naïve Bayes	Logistic Regression	K-nearest Neighbor	Support Vector M	Decision Tree	Random Forest	AdaBoost
Accuracy	78.9%	82.1%	80.6%	74.7%	84.9%	85.0%	85.9%
Run Time (s)	0.13	25.44	101.11	777.03	16.55	59.32	34.08

I used both accuracy and ROC curve to evaluate the performance of predictions. In this case, ROC was more appropriate and informative due to the skewed data structure.

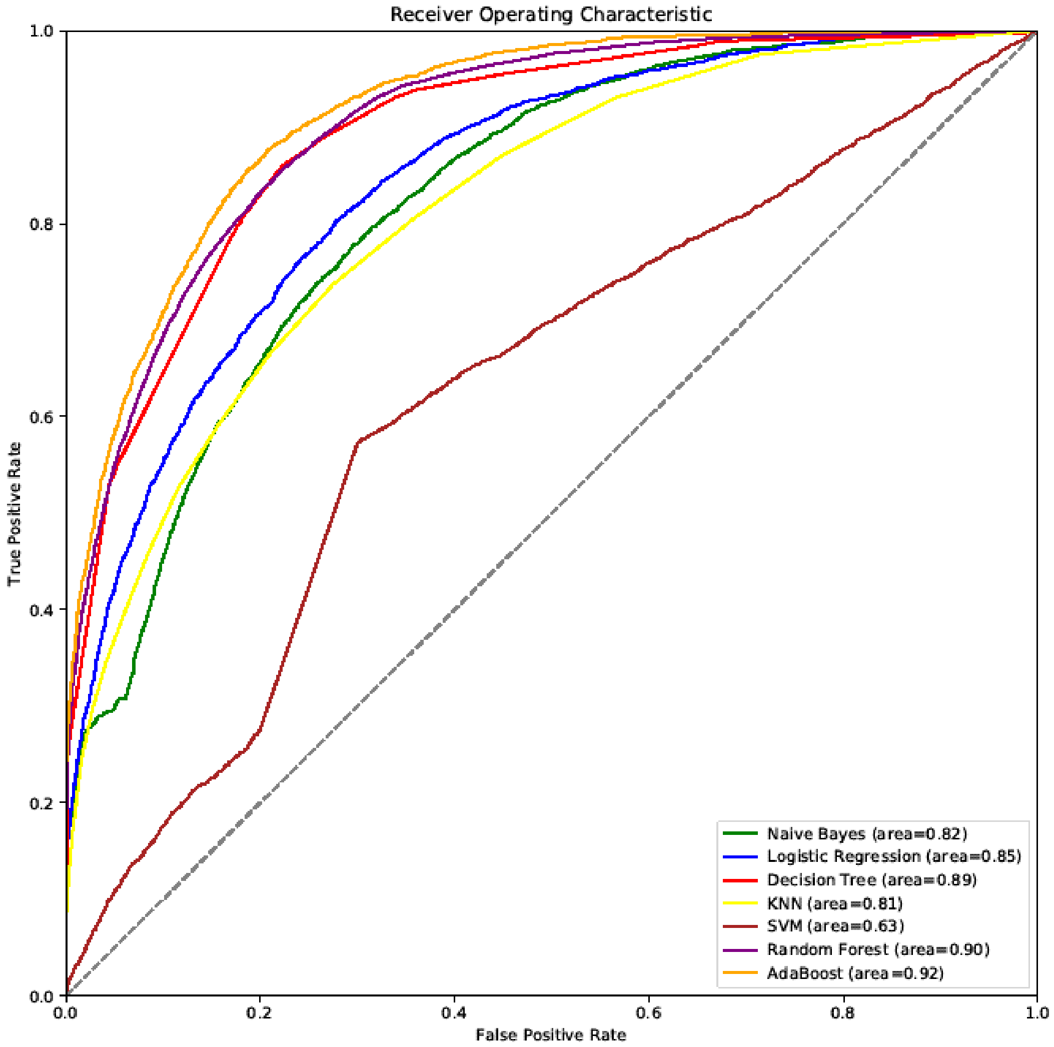

Figure 1. ROC Curve of seven different algorithms

Results and Discussion

For each algorithm, I just showed the outputs coming from the model with the best performance after tuning the hyper-parameters. In terms of time efficiency, Naïve Bayes outperformed other algorithms in this classification problem. That is probably due to the reason that Naïve Bayes assumes conditional independence between features given class labels and thus greatly reduces the number of parameters to learn. The rest of applied algorithms also finished in an acceptable time, except for the Support Vector Machine (which was only applied to the part of original training dataset).

From both the accuracy score and ROC curve (Table 1 and Figure 1), six algorithms showed descent prediction performance (accuracy >78% and ROC area >0.8). Decision Tree, Random Forest, and AdaBoost outperformed the other four algorithms. I hypothesized that the

Decision Tree gave a better prediction here was due to the reason that it is more flexible, robust to outliers and noisy data, insensitive to Monotone and irrelevant inputs. Random Forest and Ababoost showed the best results and both improve a little bit relative to Decision Tree by reducing bias and (or) variance in original decision tree. In contrast, Support Vector Machine produced a bad prediction result. I think the putative reason was that I failed to find an appropriate values of *gamma* and *C* in my constrained searching ranges. However, performing Support Vector Machine on this large dataset is computationally expensive, especially when *C* and *gamma* are large, which makes the optimal search hard to finish in an acceptable time.

In conclusion, considering both prediction accuracy and time efficiency, I conclude that Decision Tree, Random Forest, and AdaBoost are the best algorithms in dealing with the classification problem on the Adult data.