INFO526 Programming Assignment 2
Meng Wu

Tools: Python
Codes: https://github.com/wum5/INFO526_codes/tree/master/hw1

Methods:
I implemented all the three algorithms (i.e. Support Vector Machine, Random Forest, and Ababoost) 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. For each algorithm, I used the 10-fold cross-validation on the training examples (split into training set and tuning set) to better evaluate the model's performance. With cross-validation, I tried to search for the optimal hyper-parameter in each of selected algorithm, such as the value of *C* and kernel type in Support Vector Machine, the number of sampled features and generated trees in Random Forest, and learning rate and the number of trees in Ababoost. Cross-validation step is important because it can efficiently reduce the bias and (or) variance (control the bias-variance tradeoff) in our tests. 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.

Table 1. Summary of prediction accuracy using four different algorithms

Algorithms	Naïve Bayes	Nearest Neighbor	Logistic Regression	Decision Tree	Support Vector M	Random Forest	Aba-Boost
Accuracy	78.90%	80.60%	82.00%	84.90%	75.43%	84.92%	85.94%

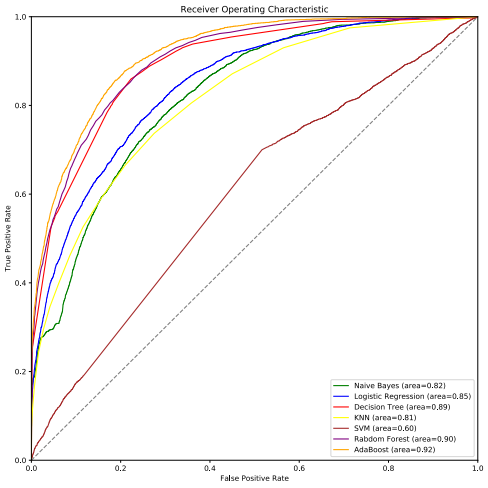

Figure 1. ROC Curve of seven different algorithms

Results and Discussion:
For each algorithm, I just showed the outputs coming from the model with the best performance after tuning the hyper-parameters. From both the accuracy score and ROC curve (Table 1 and Figure 1), 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. However, Support Vector Machine produced a bad prediction result. I think the reason is due to that an inappropriate 'kernel=linear' was used. While I wanted to test other 'kernel=poly/rbf', the program will takes lots of time because the computational complexity of this algorithm is large.