DASE 4435: Data Mining **(Due: April 1, 2020)**

Project #1

Instructor: Oluwatosin Oluwadare *Name:* Alex Allenspach

Problem 5: Report

(a) Report the number of instances in the created Training and Test datasets for both CI and CA Dataset

Table 1: Number of Instances in CI and CA Datasets.

Dataset	Training	Test	Total
CI	26048	6513	32561
CA	533	133	666

The Credit Approval Dataset originally has 690 instances but 24 instances have a missing value in a continuous attribute. In order for the algorithms to work properly in R, the missing values need to be removed for the continuous attributes.

(b) Report the **accuracy**, **precision**, and **recall** for both datasets for the three Algorithms.

Table 2: CI Performance

Model	Accuracy	Precision	Recall
Decision Tree	84.31%	85.88%	94.94%
Random Forest	85.80%	88.95%	92.82%
Naive Bayes	82.43%	85.19%	93.04%

Table 3: CA Performance

Model	Accuracy	Precision	Recall
Decision Tree	87.97%	90.14%	87.67%
Random Forest	90.23%	92.86%	89.04%
Naive Bayes	82.71%	77.78%	95.89%

(c) Report the **running times** of the three algorithms.

Table 4: Run Time for CI.

Model	Time (s)
Decision Tree	1.685
Random Forest	41.112
Naive Bayes	3.591

Table 5: Run Time for CA.

Model	Time (s)
Decision Tree	0.077
Random Forest	0.962
Naive Bayes	0.129

(d) Comment on the performance and the run-time of the three algorithms on both datasets.

Performance: Random Forest is the winner when it comes down to the accuracy of all three algorithms for both datasets. Naive Bayes did the worst accuracy wise for both datasets. Random Forest has the highest precision percentile for both datasets and has the smallest recall percentile for the CI Dataset, while Decision Tree has the highest recall percentile for CI but lowest recall percentile for CA. The highest recall percentile for CA goes to the Naive Bayes Algorithm.

Run-time: Random Forest was the winner for accuracy; however, for the Census Income dataset it takes about 24 times longer to run than the decision tree model and about 11 times longer than the naive bayes model and has the longest run-time for the Credit Approval Dataset as well. The fastest run-time for both datasets is the Decision Tree Model. My RAM can play a big factor for these run-times and the Census Income dataset is very big as well which is why the random forest algorithm takes a while because it is consisting of many decision trees.

Problem 6: Handling Missing Data

(a) State the Technique you used to handle the Missing data.

The technique I used was the third method from the PowerPoint slides - The attribute mean.

(b) Compare your result with the result reported for Credit approval in Task 5. *[compare in a tabular format for ease of reading].*

Table 6: Number of Instances in Task 5 CA and Task 6 CA.

Dataset	Training	Test	Total
Task 5 CA	533	133	666
Task 6 CA	552	138	690

Table 7: Task 5 CA Performance

Model	Accuracy	Precision	Recall
Decision Tree	87.97%	90.14%	87.67%
Random Forest	90.23%	92.86%	89.04%
Naive Bayes	82.71%	77.78%	95.89%

Table 8: Task 6 CA Performance

Model	Accuracy	Precision	Recall
Decision Tree	86.96%	89.33%	87.01%
Random Forest	87.68%	92.86%	84.42%
Naive Bayes	74.64%	72.83%	87.01%

Table 9: Task 5 CA Run Time

Model	Time (s)
Decision Tree	0.077
Random Forest	0.962
Naive Bayes	0.129

Table 10: Task 6 CA Run Time

Model	Time (s)
Decision Tree	0.128
Random Forest	0.966
Naive Bayes	0.106

(c) Comment on the similarities or differences in the result in (b) above.

NOTE: Every model is using a method on how to handle missing values from the specific library package being used in R. Decision Trees and Naive Bayes can handle missing values, but Random Forest cannot.

Similarities: Random Forest has the best accuracy and highest precision result for both tasks. The random forest precision result is actually the same for both tasks. Naive Bayes has the lowest accuracy and precision percentages in both tasks. Naive Bayes and Decision Tree have the same recall percentage for Task 6 which is also higher than Random Forest. The run-time is basically the exact same for both tasks.

Differences: The overall performance of the three algorithms in Task 6 is less efficient then Task 5. This is most likely due to the fact that I chose the attribute mean instead of the attribute mean for all samples belonging to the same class. There can be cases where a filled in missing value makes no sense for that attribute. Random Forest has a smaller recall percentile for Task 6.