Course Project 1 – Data Mining

David Vadnais

# Task 5

## Report the number of instances in the created Training and Test datasets for both CI and CA Dataset.

|  |  |  |
| --- | --- | --- |
| Data Set | Training Instances | Test Instances |
| Census Income (CI) | 26,049 | 6,513 |
| Credit Approval (CA) | 552 | 138 |

## Report the accuracy, precision, and recall for both datasets for the three Algorithms.

Census Income (CI)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree | Random Forest | Naïve Bayes |
| Accuracy | 0.793489943190542 | 0.8472286196837095 | 0.7861200675571933 |
| Precision | 0.8533440773569702 | 0.9262691377921032 | 0.8529411764705882 |
| Recall | 0.8533440773569702 | 0.9262691377921032 | 0.8533440773569702 |

Credit Approval (CA)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree | Random Forest | Naïve Bayes |
| Accuracy | 0.7608695652173914 | 0.855072463768116 | 0.8260869565217391 |
| Precision | 0.7142857142857143 | 0.8428571428571429 | 0.8 |
| Recall | 0.7142857142857143 | 0.8428571428571429 | 0.7142857142857143 |

## Report the running times the three algorithms

Census Income (CI)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree | Random Forest | Naïve Bayes |
| Run Time (s) | 0.16969679999965592 | 2.149424799998087 | 0.07788979999895673 |

Credit Approval (CA)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree | Random Forest | Naïve Bayes |
| Run Time (s) | 0.012620500001503387 | 0.11774680000235094 | 0.00979130000268924 |

## Comment on the performance and the run-time of the three algorithms on both datasets.

Random Forest performs the best however, it is the slowest. This is likely due to me using 100 estimators (partial dts). With fewer estimators this would likely have run slower and had worse performance. Given more time we would like to run the random forest at many different estimator numbers to see if there is a sweet spot between run time and performance. Naïve Bayes performed quickly and well. It seems to scale better then decision trees for prediction but perhaps not as well when it comes to run speed. In conclusion given infinite time we would choose random forest and given a time crunch we would choose naïve bayes. Since random forest is an ensemble algorithm it could be interesting to weight the forest and the naïve bayes classifier with some scaling factor so that it is relevant and see if it gets better results in both time and accuracy then the current algorithms on their own.

# Task 6

## State the Technique you used to handle the Missing data.

I am going to handle the missing data by filling it in automatically. I convert all my input features to unique hashes based on the string and I convert all my outputs to a dictionary of unique strings. This allows me to automatically create a unique string id for each empty data point on the input and I manually convert output empty strings to a null string. This will create computation overhead and increase error by introducing false features.

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

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree | Random Forest | Naïve Bayes |
| Accuracy – Normal | 0.7608695652173914 | 0.855072463768116 | 0.8260869565217391 |
| Accuracy – Filling Data | 0.7681159420289855 | 0.8333333333333334 | 0.855072463768116 |
| Precision – Normal | 0.7142857142857143 | 0.8428571428571429 | 0.8 |
| Precision -- Filling Data | 0.7391304347826086 | 0.7857142857142857 | 0.8285714285714286 |
| Recall – Normal | 0.7142857142857143 | 0.8428571428571429 | 0.7142857142857143 |
| Recall -- Filling Data | 0.7391304347826086 | 0.7857142857142857 | 0.7391304347826086 |
| Run Time (s) – Normal | 0.012620500001503387 | 0.11774680000235094 | 0.00979130000268924 |
| Run Time (s) -- Filling Data | 0.015384700000140583, | 0.12162629999875207 | 0.01296600000205217 |

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

As we predicted filling in the missing data with null increases the time complexity of the algorithm. Surprisingly, the accuracy, recall, and precision are improved in Naïve Bayes and the decision tree. This could be caused by how the data was randomly split. Maybe this test train set produced better results. I would hypothesis that this would not occur if we augmented our random subspace ensemble with bagging. Random forest held to our predicted pattern similar to bagging above we think the ensemble nature overtook a chance improvement.