Final Lab Report

CSCI 4030u

Andrew Murdoch 100707816
Andre Forbes 100669470

Task One:

C4.5 Classification model

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

lym_nodes_dimin <= 1
|   changes_in_node = no
|   |   defect_in_node = no: normal (3.0/1.0)
|   |   defect_in_node = lacunar: malign_lymph (2.0)
|   |   defect_in_node = lac_margin: normal (0.0)
|   |   defect_in_node = lac_central: normal (0.0)
|   changes_in_node = lacunar
|   |   exclusion_of_no = no: metastases (10.0/1.0)
|   |   exclusion_of_no = yes
|   |   |   special_forms = no: metastases (3.0/1.0)
|   |   |   special_forms = chalices
|   |   |   |   lym_nodes_enlar <= 2: malign_lymph (3.0)
|   |   |   |   lym_nodes_enlar > 2: metastases (2.0)
|   |   |   special_forms = vesicles: malign_lymph (19.0/1.0)
|   changes_in_node = lac_margin
|   |   block_of_affere = no
|   |   |   extravasates = no
|   |   |   |   lymphatics = normal: metastases (0.0)
|   |   |   |   lymphatics = arched
|   |   |   |   |   early_uptake_in = no: metastases (5.0/1.0)
|   |   |   |   |   early_uptake_in = yes: malign_lymph (4.0/1.0)
|   |   |   |   lymphatics = deformed: metastases (5.0)
|   |   |   |   lymphatics = displaced: malign_lymph (1.0)
|   |   |   extravasates = yes: malign_lymph (4.0)
|   |   block_of_affere = yes: metastases (56.0/3.0)
|   changes_in_node = lac_central
|   |   no_of_nodes_in <= 1
|   |   |   block_of_affere = no: malign_lymph (3.0)
|   |   |   block_of_affere = yes: metastases (2.0)
|   |   no_of_nodes_in > 1: malign_lymph (20.0)
lym_nodes_dimin > 1
|   by_pass = no: metastases (2.0/1.0)
|   by_pass = yes: fibrosis (4.0)
```

Number of Leaves :   21

Size of the tree :   34

Time taken to build model: 0 seconds

Ripper

=== Classifier model (full training set) ===

JRIP rules:
===========

```
(lymphatics = normal) => class=normal (2.0/0.0)
(lym_nodes_dimin >= 2) and (by_pass = yes) => class=fibrosis (4.0/0.0)
(no_of_nodes_in >= 3) and (special_forms = vesicles) => class=malign_lymph (41.0/5.0)
(block_of_affere = no) and (extravasates = yes) => class=malign_lymph (8.0/0.0)
(changes_in_node = lac_central) => class=malign_lymph (8.0/2.0)
 => class=metastases (85.0/11.0)
```

Number of Rules : 6

Time taken to build model: 0.02 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

ID3

=== Classifier model (full training set) ===

Id3

```
changes_in_node = no
|  lymphatics = normal: normal
|  lymphatics = arched
|  |  early_uptake_in = no: metastases
|  |  early_uptake_in = yes: malign_lymph
|  lymphatics = deformed: fibrosis
|  lymphatics = displaced: malign_lymph
```

```
changes_in_node = lacunar
|  special_forms = no
|  |  bl_of_lymph_c = no
|  |  |  changes_in_stru = no: null
|  |  |  changes_in_stru = grainy: metastases
|  |  |  changes_in_stru = drop_like
|  |  |  |  extravasates = no: metastases
|  |  |  |  extravasates = yes: malign_lymph
|  |  |  changes_in_stru = coarse: metastases
|  |  |  changes_in_stru = diluted
|  |  |  |  block_of_affere = no: malign_lymph
|  |  |  |  block_of_affere = yes: metastases
|  |  |  changes_in_stru = reticular: null
|  |  |  changes_in_stru = stripped: null
|  |  |  changes_in_stru = faint: metastases
|  |  bl_of_lymph_c = yes: fibrosis
|  special_forms = chalices
|  |  block_of_affere = no
|  |  |  lymphatics = normal: null
|  |  |  lymphatics = arched: malign_lymph
|  |  |  lymphatics = deformed: malign_lymph
|  |  |  lymphatics = displaced: metastases
|  |  block_of_affere = yes: metastases
|  special_forms = vesicles
|  |  early_uptake_in = no
|  |  |  lymphatics = normal: null
|  |  |  lymphatics = arched: metastases
|  |  |  lymphatics = deformed: fibrosis
|  |  |  lymphatics = displaced: null
|  |  early_uptake_in = yes
|  |  |  changes_in_stru = no: metastases
|  |  |  changes_in_stru = grainy: null
|  |  |  changes_in_stru = drop_like: null
|  |  |  changes_in_stru = coarse: malign_lymph
|  |  |  changes_in_stru = diluted: malign_lymph
|  |  |  changes_in_stru = reticular: malign_lymph
|  |  |  changes_in_stru = stripped: malign_lymph
|  |  |  changes_in_stru = faint: malign_lymph
changes_in_node = lac_margin
|  block_of_affere = no
|  |  extravasates = no
|  |  |  lymphatics = normal: null
|  |  |  lymphatics = arched
|  |  |  |  changes_in_stru = no: null
```

```
| | | |  changes_in_stru = grainy: null
| | | |  changes_in_stru = drop_like
| | | | |  changes_in_lym = bean: null
| | | | |  changes_in_lym = oval: metastases
| | | | |  changes_in_lym = round: malign_lymph
| | | |  changes_in_stru = coarse: malign_lymph
| | | |  changes_in_stru = diluted: malign_lymph
| | | |  changes_in_stru = reticular: metastases
| | | |  changes_in_stru = stripped: null
| | | |  changes_in_stru = faint
| | | | |  early_uptake_in = no: metastases
| | | | |  early_uptake_in = yes: malign_lymph
| | |  lymphatics = deformed: metastases
| | |  lymphatics = displaced: malign_lymph
| |  extravasates = yes: malign_lymph
|  block_of_affere = yes
| |  changes_in_stru = no: null
| |  changes_in_stru = grainy: metastases
| |  changes_in_stru = drop_like: metastases
| |  changes_in_stru = coarse: metastases
| |  changes_in_stru = diluted
| | |  no_of_nodes_in = '(-inf-3.5]': metastases
| | |  no_of_nodes_in = '(3.5-inf)': malign_lymph
| |  changes_in_stru = reticular: null
| |  changes_in_stru = stripped: malign_lymph
| |  changes_in_stru = faint
| | |  no_of_nodes_in = '(-inf-3.5]': metastases
| | |  no_of_nodes_in = '(3.5-inf)': malign_lymph
changes_in_node = lac_central
|  lym_nodes_enlar = '(-inf-2.5]'
| |  changes_in_stru = no: null
| |  changes_in_stru = grainy
| | |  block_of_affere = no: malign_lymph
| | |  block_of_affere = yes: metastases
| |  changes_in_stru = drop_like: null
| |  changes_in_stru = coarse: null
| |  changes_in_stru = diluted: metastases
| |  changes_in_stru = reticular: null
| |  changes_in_stru = stripped: null
| |  changes_in_stru = faint: malign_lymph
|  lym_nodes_enlar = '(2.5-inf)': malign_lymph

Time taken to build model: 0.01 seconds
```

Explanations:

The C4.5 algorithm is used as a decision tree classifier that can generate an output based on an inputted sample of data. For each node of the tree, this algorithm will create branches that will split the data into subsets of the data from the patterns in the dataset. This method uses entropy to generate the probability of each event happening. After creating the tree, pruning is used with the C4.5 algorithm to remove any redundant branches that do not help to generate the decisions. This is used to ensure that errors are not included when creating the tree.

The Ripper algorithm uses rule learning to generate the decisions. This is done in a three step process. The first part of the process creates conditions for each rule to properly classify data into subsets. The second stage is the pruning process. This occurs when the entropy for a given rule does not decrease as the rule becomes more specific. After the initial run through, these first two steps are repeated further until all the rules have been optimized which is the final process.

ID3 is another decision tree algorithm used to predict possible outcomes. This algorithm is the precursor to the C4.5 algorithm. This starts off by calculating the Information Gain (Entropy) and splits the dataset into certain subsets. For each subset, it will break off into certain results based upon the information in each. For example, it could produce a “yes” or “no” statement as well as break off into another subset and repeat the same process again. This whole process is repeated until all possible results can be generated.

Task Two:

C4.5

=== Summary ===

Correctly Classified Instances	420	97.2222 %
Incorrectly Classified Instances	12	2.7778 %
Kappa statistic	0.9444	
Mean absolute error	0.0892	
Root mean squared error	0.1831	
Relative absolute error	17.8311 %	
Root relative squared error	36.5759 %	
Total Number of Instances	432	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.053	0.944	1.000	0.971	0.946	0.983	0.964	0
	0.947	0.000	1.000	0.947	0.973	0.946	0.983	0.981	1
Average	0.972	0.025	0.974	0.972	0.972	0.946	0.983	0.973	

a	b	← Classified as
204	0	A = 0
12	216	B = 1

ID3

=== Summary ===

Correctly Classified Instances 408 94.4444 %
Incorrectly Classified Instances 16 3.7037 %
Kappa statistic 0.9245
Mean absolute error 0.0377
Root mean squared error 0.1943
Relative absolute error 7.6849 %
Root relative squared error 39.1873 %
UnClassified Instances 8 1.8519 %
Total Number of Instances 432

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.980	0.055	0.943	0.980	0.962	0.925	0.964	0.934	0
	0.945	0.020	0.981	0.945	0.963	0.925	0.946	0.941	1
Average	0.962	0.036	0.963	0.962	0.962	0.925	0.955	0.938	

a	b	← Classified as

200	4	A = 0
12	208	B = 1

Ripper

=== Summary ===

Correctly Classified Instances 390 90.2778 %
Incorrectly Classified Instances 42 9.7222 %
Kappa statistic 0.8053
Mean absolute error 0.1314
Root mean squared error 0.277
Relative absolute error 26.2643 %
Root relative squared error 55.3461 %
Total Number of Instances 432

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.912	0.105	0.866	0.912	0.899	0.806	0.938	0.879	0
	0.895	0.088	0.919	0.985	0.907	0.806	0.938	0.942	1
Average	0.903	0.096	0.903	0.903	0.903	0.806	0.938	0.912	

a	b	← Classified as
186	18	A = 0
24	204	B = 1

K-Nearest Neighbor

=== Summary ===

Correctly Classified Instances 378 87.5 %
Incorrectly Classified Instances 54 12.5 %
Kappa statistic 0.7512
Mean absolute error 0.191
Root mean squared error 0.3228
Relative absolute error 38.1693 %
Root relative squared error 64.5029 %
Total Number of Instances 432

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.941	0.184	0.821	0.941	0.877	0.758	0.933	0.902	0
	0.816	0.059	0.939	0.816	0.873	0.758	0.933	0.935	1

Average	0.875	0.118	0.883	0.875	0.875	0.758	0.933	0.919	

a	b	← Classified as
192	12	A = 0
42	186	B = 1

Naive Bayesian Classification

=== Summary ===

Correctly Classified Instances 420 97.2222 %
Incorrectly Classified Instances 12 2.7778 %
Kappa statistic 0.9444
Mean absolute error 0.1863
Root mean squared error 0.2323
Relative absolute error 37.2363 %
Root relative squared error 46.4131 %
Total Number of Instances 432

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.053	0.944	1.000	0.971	0.946	0.975	0.961	0
	0.947	0.000	1.000	0.947	0.973	0.946	0.975	0.985	1
Average	0.972	0.025	0.974	0.972	0.972	0.946	0.975	0.973	

a	b	← Classified as
204	0	A = 0
12	216	B = 1

Neural Networks

=== Summary ===

Correctly Classified Instances 404 93.5185 %
Incorrectly Classified Instances 28 6.4815 %

Kappa statistic 0.8709
Mean absolute error 0.068
Root mean squared error 0.2322
Relative absolute error 13.5875 %
Root relative squared error 46.3993 %
Total Number of Instances 432

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.123	0.879	1.000	0.936	0.878	0.967	0.941	0
	0.877	0.000	1.000	0.877	0.935	0.878	0.967	0.981	1
Average	0.935	0.058	0.943	0.935	0.935	0.878	0.967	0.962	

a	b	← Classified as
204	0	A = 0
28	200	B = 1

Explanations:

From these six different testing algorithms, it is evident that the C4.5 algorithm has the highest averages across all the table values and the K-Nearest Neighbor algorithm has the lowest. This is due to the fact that the C4.5 algorithm has a more efficient algorithm for smaller datasets and produces a more thorough result by pruning where K-Nearest Neighbor does not.

The K-Nearest Neighbor is an algorithm that uses previously inputted data to produce an output for data that is unlabeled. To aid in the process of generating outputs, this algorithm assumes that similar results are close in proximity. This algorithm starts out with choosing a K value that has the lowest number of errors. To find the right K value, multiple runs of the program will determine the right number. Once a K value is chosen, the distance between both the test data set and the training data set is calculated. This is calculated by finding the euclidean distance between the two. From there the distance is added to a collection that holds the distance and the index of the data. Once all the data has been tested and the euclidean distances have been calculated, this information is sorted in order of distances in ascending order. From the collection, select the first K number of entries and the labels for them. Depending on what type of data set this algorithm is trying to solve, will determine what value is returned. If the data set is a regression problem that has to have a decimal point value, the mean is returned. If not then it is considered classified and will return the mode.

Naive Bayesian Classification uses probability to produce an outcome. Given a data set, this classification can be used with Bayes' theorem, $P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$. In this equation, A is the information that is needed to be produced and B is the set of features that would affect the outcome. This will produce the probability for the expected output. There are three types of this classifier: multinomial, bernoulli and gaussian. Multinomial is used generally for classifying documents and uses frequency of words in the documents to classify them. For Bernoulli, the use is similar but uses a true or false base system to classify. Finally, the Gaussian is used to take a value that is assumed to be sampled from a gaussian distribution.

A neural network is a type of algorithm that works similar to that of the human brain. This algorithm will take in inputs, and then stores the information into nodes. These nodes are then weighed by the amount of important information they store compared to the other nodes, these are known as the hidden layer. Once all the nodes have been weighed, output nodes are then generated. The hidden layer and output is then recalibrated based based on the errors found in the outputs, and this will repeat until the proper conditions are met with the algorithm. After this, the final output is based upon the sum of all the hidden layers.

Task Three:

C4.5

=== Summary ===

Correctly Classified Instances 178 85.9903 %
Incorrectly Classified Instances 29 14.0097 %
Kappa statistic 0.7168
Mean absolute error 0.1958
Root mean squared error 0.3288
Relative absolute error 39.4502 %
Root relative squared error 65.6306 %
Total Number of Instances 207

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.766	0.064	0.916	0.776	0.840	0.725	0.901	0.857	+
	0.936	0.224	0.823	0.936	0.876	0.725	0.901	0.872	-
Average	0.860	0.149	0.867	0.860	0.859	0.725	0.901	0.865	

a	b	← Classified as
76	22	A = +

7	102	B = -

Naive Bayes Classifier

=== Summary ===

Correctly Classified Instances 156 75.3623 %
Incorrectly Classified Instances 51 24.6377 %
Kappa statistic 0.4968
Mean absolute error 0.2468
Root mean squared error 0.4633
Relative absolute error 49.7186 %
Root relative squared error 92.494 %
Total Number of Instances 207

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.561	0.073	0.873	0.561	0.683	0.529	0.880	0.869	+
	0.927	0.439	0.701	0.927	0.798	0.529	0.880	0.887	-
Average	0.754	0.266	0.783	0.754	0.744	0.529	0.880	0.879	

a	b	← Classified as
55	43	A = +
8	101	B = -

Neural Networks

=== Summary ===

Correctly Classified Instances 160 77.2947 %
Incorrectly Classified Instances 47 22.7053 %
Kappa statistic 0.5401
Mean absolute error 0.2173
Root mean squared error 0.4352
Relative absolute error 43.7768 %
Root relative squared error 86.8833 %
Total Number of Instances 207

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class

	0.663	0.128	0.823	0.663	0.734	0.550	0.869	0.864	+
	0.872	0.337	0.742	0.872	0.802	0.550	0.869	0.840	-
Average	0.773	0.238	0.780	0.773	0.770	0.550	0.869	0.851	

a	b	← Classified as
65	33	A = +
14	95	B = -

Task Four:

C4.5

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances 283 84.2262 %
Incorrectly Classified Instances 53 15.7738 %
Kappa statistic 0.7824
Mean absolute error 0.0486
Root mean squared error 0.1851
Relative absolute error 26.5877 %
Root relative squared error 61.3413 %
Total Number of Instances 336

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.951	0.036	0.951	0.951	0.951	0.915	0.962	0.915	cp
	0.844	0.066	0.793	0.844	0.818	0.762	0.907	0.784	im
	0.865	0.032	0.833	0.865	0.849	0.821	0.904	0.669	pp
	0.571	0.030	0.690	0.571	0.625	0.589	0.855	0.635	imU
	0.700	0.028	0.609	0.700	0.651	0.629	0.890	0.655	om
	0.600	0.006	0.600	0.600	0.600	0.594	0.993	0.604	omL
	0.000	0.000	?	0.000	?	?	0.490	0.006	imL
	0.000	0.000	?	0.000	?	?	0.479	0.006	imS
Avg.	0.842	0.040	?	0.842	?	?	0.920	0.787	

Ripper
Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	271	80.6548 %
Incorrectly Classified Instances	65	19.3452 %
Kappa statistic	0.7311	
Mean absolute error	0.0608	
Root mean squared error	0.2013	
Relative absolute error	33.2586 %	
Root relative squared error	66.7354 %	
Total Number of Instances	336	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.951	0.088	0.889	0.951	0.919	0.857	0.943	0.882	cp
	0.766	0.054	0.808	0.766	0.787	0.726	0.928	0.821	im
	0.788	0.025	0.854	0.788	0.820	0.789	0.924	0.751	pp
	0.514	0.060	0.500	0.514	0.507	0.449	0.852	0.435	imU
	0.750	0.013	0.789	0.750	0.769	0.755	0.874	0.602	om
	0.400	0.015	0.286	0.400	0.333	0.326	0.767	0.165	omL
	0.000	0.000	?	0.000	?	?	0.708	0.086	imL
	0.000	0.000	?	0.000	?	?	0.380	0.006	imS
Avg.	0.807	0.061	?	0.807	?	?	0.916	0.764	

Naive Bayesian Classification
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	287	85.4167 %
Incorrectly Classified Instances	49	14.5833 %
Kappa statistic	0.8002	
Mean absolute error	0.0429	
Root mean squared error	0.1639	
Relative absolute error	23.461 %	
Root relative squared error	54.3314 %	
Total Number of Instances	336	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.958	0.041	0.945	0.958	0.951	0.915	0.986	0.973	cp
0.727	0.031	0.875	0.727	0.794	0.745	0.966	0.904	im
0.846	0.032	0.830	0.846	0.838	0.808	0.945	0.901	pp
0.829	0.060	0.617	0.829	0.707	0.677	0.937	0.630	imU
0.900	0.009	0.857	0.900	0.878	0.870	0.996	0.964	om
0.600	0.000	1.000	0.600	0.750	0.772	0.996	0.883	omL
0.000	0.006	0.000	0.000	0.000	-0.006	0.058	0.006	imL
0.000	0.003	0.000	0.000	0.000	-0.004	0.148	0.005	imS
Avg. 0.854	0.036	0.861	0.854	0.854	0.819	0.960	0.897	

K-Nearest Neighbor

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	270	80.3571 %
Incorrectly Classified Instances	66	19.6429 %
Kappa statistic	0.7295	
Mean absolute error	0.0535	
Root mean squared error	0.2189	
Relative absolute error	29.238 %	
Root relative squared error	72.5574 %	
Total Number of Instances	336	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.930	0.052	0.930	0.930	0.930	0.878	0.942	0.900	cp
0.727	0.081	0.727	0.727	0.727	0.646	0.814	0.609	im
0.846	0.046	0.772	0.846	0.807	0.771	0.903	0.695	pp
0.486	0.056	0.500	0.486	0.493	0.435	0.713	0.304	imU
0.750	0.006	0.882	0.750	0.811	0.803	0.896	0.680	om
1.000	0.003	0.833	1.000	0.909	0.911	0.999	0.867	omL
0.000	0.006	0.000	0.000	0.000	-0.006	0.695	0.010	imL
0.000	0.000	?	0.000	?	?	0.698	0.010	imS
Avg. 0.804	0.054	?	0.804	?	?	0.878	0.715	

Neural Networks

Time taken to build model: 0.32 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	289	86.0119 %
Incorrectly Classified Instances	47	13.9881 %
Kappa statistic	0.8066	
Mean absolute error	0.0484	
Root mean squared error	0.1704	
Relative absolute error	26.479 %	
Root relative squared error	56.4913 %	
Total Number of Instances	336	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.965	0.036	0.952	0.965	0.958	0.927	0.980	0.962	cp
	0.831	0.062	0.800	0.831	0.815	0.759	0.951	0.870	im
	0.846	0.032	0.830	0.846	0.838	0.808	0.952	0.806	pp
	0.629	0.037	0.667	0.629	0.647	0.608	0.935	0.580	imU
	0.850	0.009	0.850	0.850	0.850	0.841	0.977	0.887	om
	0.800	0.003	0.800	0.800	0.800	0.797	0.997	0.786	omL
	0.000	0.000	?	0.000	?	?	0.187	0.005	imL
	0.000	0.000	?	0.000	?	?	0.340	0.007	imS
Avg.	0.860	0.039	?	0.860	?	?	0.956	0.859	

It appears that neural networks or multilayer perceptrons have the highest percentage of accuracy in the cross validation tests above (10-fold) with 86% of instances correctly classified. Naive Bayesian and C4.5 closely follow with 85.4% and 84.2% respectively. K-Nearest Neighbor performed and RIPPER performed approximately five percent worse than the others.

KNN depends greatly on distances between points. As you increase the number of dimensions, your distances are going to be less representative, this is called the curse of dimensionality. Taking this into account, KNN might perform slightly better if the number of features were to be reduced. RIPPER works well on datasets with imbalanced data meaning most of the data belongs to a single class (default class). Knowing this we could say based on the test results that the ecoli dataset we are working with is probably more balanced than imbalanced.

The difference between misclassification rates is very small for multilayer perceptrons, Naive Bayesian and C4.5. So I wouldn't say there is a clear winner in these tests. KNN and RIPPER performed measurably worse and could be labeled losers in this case.

Task Five:

C4.5

Correctly Classified Instances 8480 94.2222 %
Incorrectly Classified Instances 520 5.7778 %
Kappa statistic 0
Total Cost 2600
Average Cost 0.2889
Mean absolute error 0.1094
Root mean squared error 0.2333
Relative absolute error 99.966 %
Root relative squared error 100 %
Total Number of Instances 9000

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	1.000	0.942	1.000	0.970	?	0.500	0.942	No
	0.000	0.000	?	0.000	?	?	0.500	0.058	Yes
Average	0.942	0.942	?	0.942	?	?	0.500	0.891	

a	b	← Classified as
8480	0	A = no
520	0	B = yes

Cost = 8480*(0) + 0*(50) + 520*(5) + 0*(0) = 2600

Naive Bayesian Classification

Correctly Classified Instances 8431 93.6778 %
Incorrectly Classified Instances 569 6.3222 %
Kappa statistic 0.0237
Total Cost 5545
Average Cost 0.6161
Mean absolute error 0.111
Root mean squared error 0.2418
Relative absolute error 101.442 %
Root relative squared error 103.652 %
Total Number of Instances

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.933	0.979	0.943	0.993	0.967	0.037	0.647	0.965	No
	0.021	0.007	0.155	0.021	0.037	0.037	0.647	0.098	Yes
Average	0.937	0.923	0.897	0.937	0.914	0.037	0.647	0.915	

a	b	← Classified as
8420	60	A = no
509	11	B = yes

Cost = 8420*(0) + 60*(50) + 509*(5) + 11*(0) = 5545

For pre-processing, the ‘ANUMMER_10’ and ‘MAHN_HOECHST’ attributes were removed, due to either having a full column of unknowns or because the column has the same values in another column. In WEKA, The filters of ‘ReplaceMissingValues’ and ‘Discretize’ were applied to the dataset to generate information.