Threshold Selector - Weka

Patricia Figueira Goldberg

September 13, 2013

1 Motivation and Background

1.1 Weka

The Waikato Environment for Knowledge Analysis (Weka) is a workbench that unifies machine learnings techniques with easy access [4]. It was created to unify the various learning algorithms which were spread out on a variety of platforms and data formats. Now Weka uses a unique data format which is called Attribute-Relation File Format (ARFF). The ARFF is an ASCII text file which contains the list of attributes, its types (such as integer or double) and the set of instances. [7] This set of instances is what defines the data set. Exploring Weka more deeply, useful tools such as class distribution can be found. Not only Weka provides a variety of learning algorithms, but also is an Open-Source Software which offers researchers to implement new machine learning schemes. The following work (Threshold Selector - Weka) was founded on Weka's idea of implementing new learning algorithms using the tools that WEKA offers. The specific tool used was the metaclassifier called "Threshold Selector".

1.2 ROC analysis

Receiver Operating Characteristic (ROC) analysis investigates the relationship between the true positive rate (tpr) and the false positive rate (fpr) of a binary classification. [2] The figure 1 shows a confusion matrix, which is a machine learning tool that allows researchers to visualize algorithm's performance. Analysing this matrix, the relation can be seen between P (Actual) and P' (Predicted) is called True Positive (TP), in other words this is the quantity of instances which was correctly classified as positives. At the same time, the relation between N (Actual) and P' (Predicted) is called False Positive (FP), which is the quantity of instances which was incorrectly classified as positives. On ROC analysis we use these both numbers (TP and FP) but as a rate, so the ROC curve is more uniformly distributed, being the rate between 0 and 1. Mathematically speaking, the true positive rate can be acquired by making: tpr = $\frac{TP}{numberofinstances}$. Continuing the same line of thinking: fpr = $\frac{FP}{numberofinstances}$. Still on figure 1, it can be noted that the False Negative (FN), which is the number of instances incorrectly classified as negative, and the True Negative (TN), which is the number of instances correctly classified as negative. Another method to measure the performance that Weka offers is the Mean Absolute Error (MAE). The Mean Absolute Error is used to measure how close the predictions made by the classifier are to

the eventual outcomes. From the confusion matrix, a variety of measures can be calculated, such as Precision, Recall and Fmeasure. Precision, being one of them, can be defined as $\frac{TP}{TP+FP}$. Another useful measure is Recall, which can be defined as $\frac{TP}{TP+FN}$. Fmeasure can be defined as the harmonic mean of precision and recall, mathematically speaking: $Fmeasure = 2 * \frac{Precision*Recall}{Precision+Recall}$. [8]

	P' (Predicted)	N' (Predicted)
P (Actual)	True Positive	False Negative
N (Actual)	False Positive	True Negative

Figure 1: Confusion Matrix [1]

1.3 ROC curve

The ROC curve (figure 2) is the most powerful tool on ROC analysis. It plots, conventionally, the true positive rate against the false positive rate. [2] Each operating point along the curve has its particular score. The list of scores is given by the classifier after the classification of the instances. Each score displays a different performance of the algorithm, meaning that each score has its own confusion matrix. As the main goal of the ROC curve is to maximize the performance of the algorithm, by convenience, the score can be used as a threshold. Different thresholds are displayed along the ROC curve, starting with the highest threshold and finishing with the lowest. The lowest threshold gives the highest proportion of predicted positives on the classification. At the same time, the highest threshold gives the lowest proportion of predicted positives.

1.4 Threshold Selector

The Threshold Selector is a metaclassifier, offered by Weka, which uses the ROC curve to select the best threshold based on the performance. [3] The threshold is selected accordingly to the measure chosen by the user or the default measure (which is the Fmeasure). Therefore, the threshold chosen

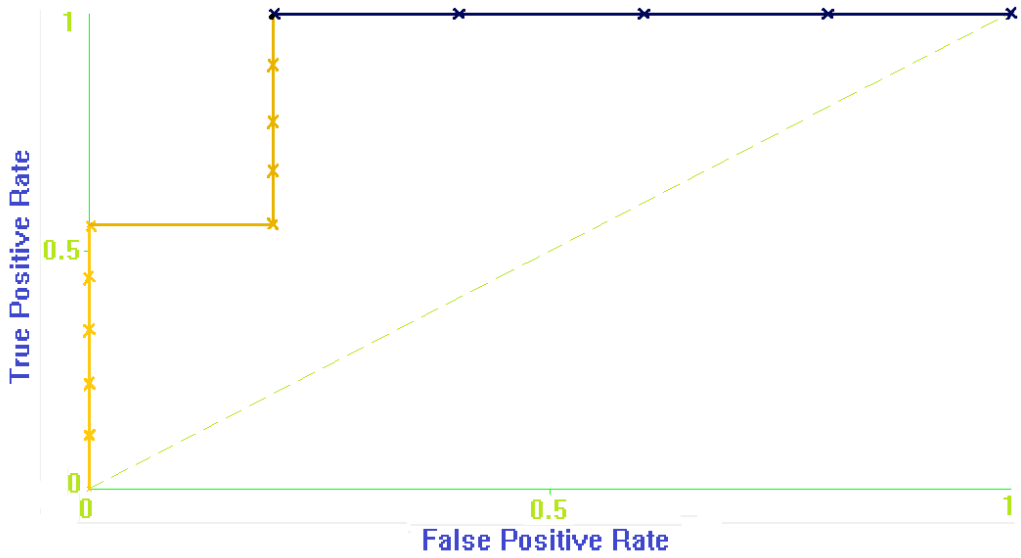

Figure 2: Example of a ROC curve generated by Weka

by the Threshold Selector will be the one which shows the best performance on the desired measure. On this work, this metaclassifier was upgraded to have more four measures: Score driven, Rate driven, Score uniform and Rate uniform. [5]All of these four new measures were based on the score represented on the ROC curve.

2 Solutions

Knowing that the Scores given by the classifier are between 0 and 1 and that the Rate is the uniformly distributed Scores, we have that π is the probability of the instance being positive. The probability π is calculated by following formula: $\frac{\alpha^+}{\alpha^+ + \alpha^-}$, being α^+ the total number of positive instances and α^- the total number of negative instances.

2.1 Score Driven

Score Driven uses the π as a measure to set the Threshold. It assumes that the model is calibrated and that the π is the proportion of positives instances. Knowing that the relation between the threshold and the proportion of predicted positives is inversely proportional then we can deduct that the Threshold (T) is $T = 1 - \pi$, as we want a lower threshold for a higher π.

2.2 Rate Driven

Rate Driven first transforms the list of the score to a list of the score uniformly distributed, which is called rate. As the list of the scores is taken from the ROC curve, the list is on descending order (from one to zero). When making it uniformly distributed, the list changes to ascending order (from zero to one). This was mentioned because we now change the Threshold selection from $T = 1 - \pi$ (as in Score Driven) to $T = \pi$. This threshold is still not the final one. We still have to find the threshold which this rate refers to in the score list (as we were looking before only to the rate list). To find the score which T refers to, we simply do this:
Assuming:
Score (S): $0 \leq S_1 \leq S_2 \leq ... \leq S_n \leq 1$
Rate (R): $0 = R_1 \leq R_2 \leq ... \leq R_n = 1$ - uniformly distribuited

Then:
if exists i such that R[i]=T then
Threshold = S[i]
else if exists i such that ((R[i] < T) & (R[i+1] > T)) then
Threshold = ((S[i] + S[i+1])/2);

2.3 Score Uniform

Score Uniform uses the score of each instance as the probability of that instance being classified as positive or negative. To classify it, the Score

Uniform generates a random number and compares it with this probability. In other words, the instance is classified as positive with the probability p, given by the score and it is classified as negative with probability $1 - p$.

2.4 Rate Uniform

Rate Uniform uses the rate of each instance as the probability p of that instance being classified as positive or negative. By generating a random number, this measure is able to compare this random number with the rate on that instance and classify the instance as positive with probability p and classify it as negative with probability $1 - p$.

3 Histogram

For further analysis of this work, Score Uniform measure was taken for testing purposes (as it is a good example of testing using random numbers). It is good to mention that the Mean Absolute Error (MAE) is unique for each classifier on the dataset. Using the number given by the Mean Absolute Error we can calculate the Expected Incorrectly Classified (EIC) on each data set by making: $EIC = MAE * number of instances$. So, a good way to test the Score Uniform is to run it a thousand times using different datasets and different classifiers. With this result, we can generate a histogram of Incorrectly Classified (IE) instances, defined by: $IE = FP + FN$. To analyse the histogram, we should see if the Expected Incorrectly Classified is located on the area with the most concentration of quantity of Incorrectly Classified on the histogram (figure 3).

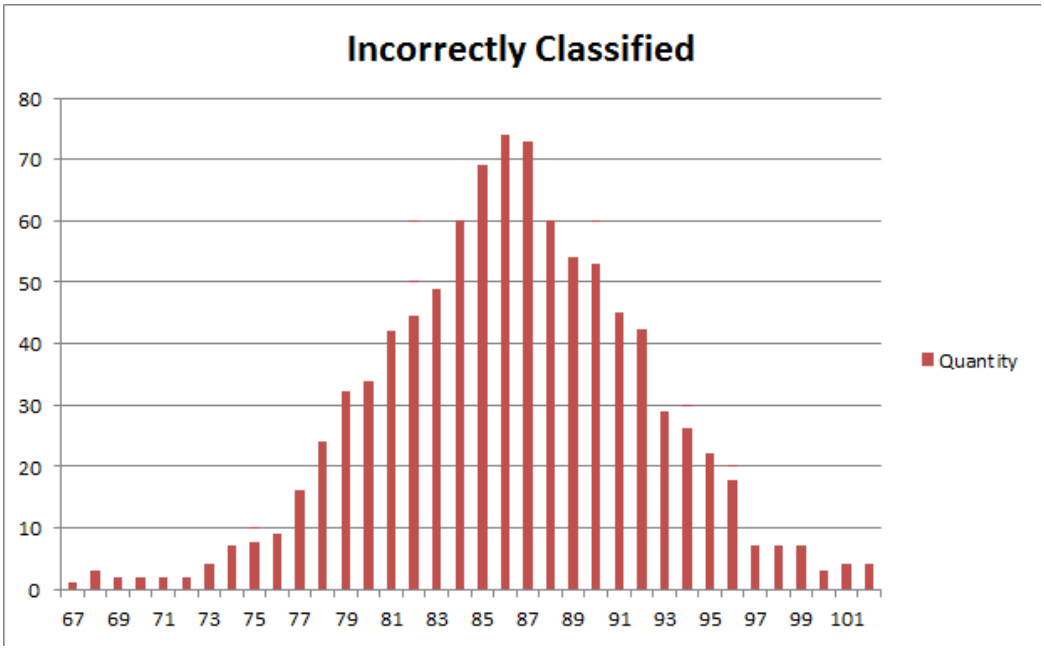

Figure 3: Ideal Histogram which the center (the top of the curve) is the Expected Incorrectly Classified

This part is now divided in 3 different subsection, one for each classifier using the same dataset.

3.1 Naive Bayes

The Naive Bayes classifier is a probabilist classifier which assumes that the predictive attributes are conditionally independent given the class [6]. Knowing that the Mean Absolute Error of Naive Bayes on this data set is 0.2810963547, we could calculate the Expected Incorrectly Classified as: 215.8820004. Analysing the Histogram (figure 4) we that area with a higher concentration of Incorrectly Classified instances is between 211 and 216, interval which includes this Expected Incorrectly Classified.

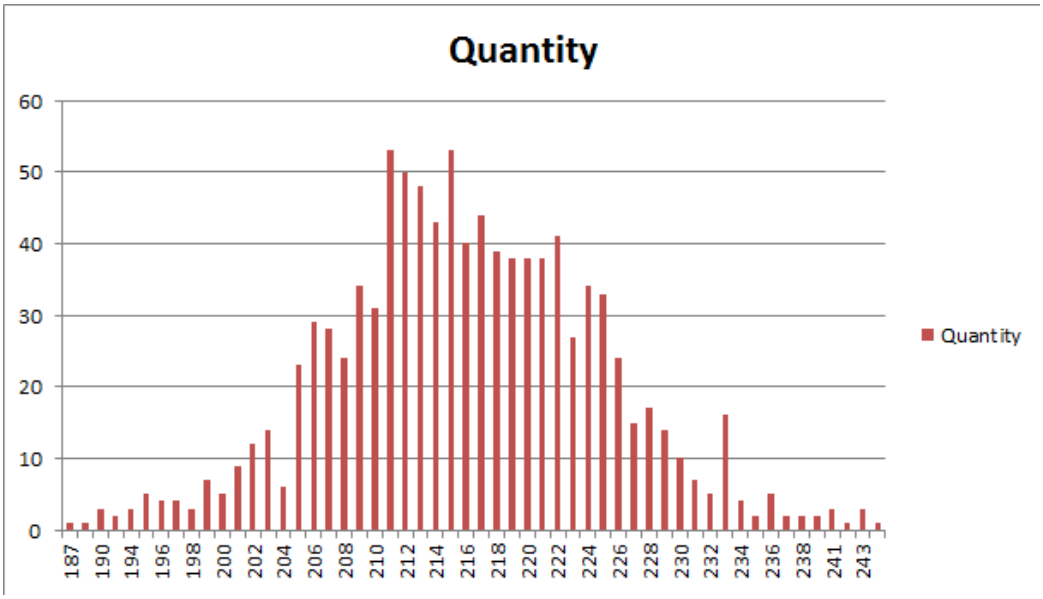

Figure 4: Histogram using Naive Bayes

3.2 J48 (Decision Tree)

Decision tree is a classifier expressed as a recursive partition of the instance space forming a model, which in Machine Learning is called tree. [9] Having calculated the Expected Incorrectly Classified as 183.0201055 the histogram can be analysed (figure 5). Looking at the area with a higher concentration of Incorrectly Classified instances, the interval of the ideal Incorrectly Classified is between 181 and 185.

3.3 ZeroR

ZeroR is the simplest classification method which classifies all the instances as the target class, which can be either positive or negative (remember that,

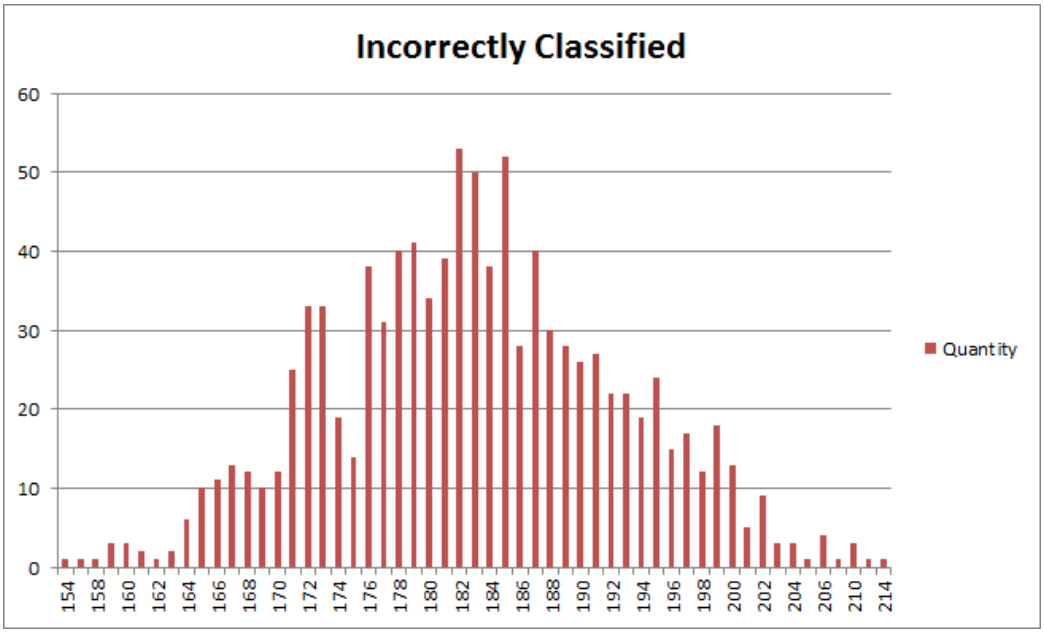

Figure 5: Histogram using Decision Tree

on this case, we are only assuming binary classification). Although there is no predictability power in ZeroR, we will use it for determining the performance of this method as the Histogram should look closer to the ideal one as well. In fact on this case the Expected Incorrectly Classifed (which is 349.0493506) is located on the area with a higher concentration of Incorrectly Classified instances on the histogram (figure 6), which is between 345 and 353.

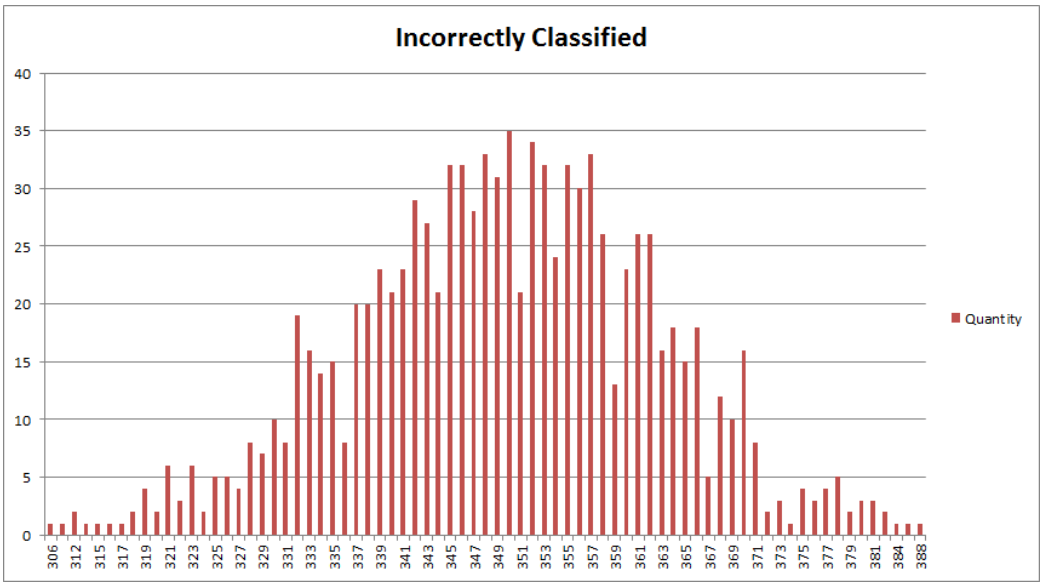

Figure 6: Histogram using ZeroR

4 Experimental Results

The 5 figures on the First Appendix represent 5 tables of the Experimental Results. Each table represents the results of one classifier in 22 different dataset. For analysis, the Mean Absolute Error, the Average Error and the Standard Deviation of the each classifier are shown for each dataset. Analysing the table, some number can be found as rather strange. For example, some zeros can be noted as the value of Average Error and Standard Deviation. Another strange variation of the numbers that can be noted is some high Standard Deviation that appears (0.12 can be already considered as a high Standard Deviation). These strange variations occur when the dataset is very small. For example, the dataset named weather on the table has only 14 instances.

A First Appendix

Classifiers	Naïve Bayes		
Data Set	Average	Standart Deviation	Mean Absolute Error
BC	0.300461538	0.020465042	0.301223266
BreastCancer	0.301440559	0.020483476	0.301223266
creditg	0.282055	0.010779619	0.282098648
diabetes	0.281308594	0.011611086	0.281096355
glass2	0.360208589	0.015484661	0.359765113
HE	0.147612903	0.015315258	0.147705968
heart-train	0.158071429	0.018021339	0.158003043
HO	0.211192935	0.011746779	0.211130741
HY	0.02301644	0.001367098	0.022917607
ionosphere	0.166997151	0.00712296	0.166749963
labor	0.04877193	0.02132194	0.048102248
labordiscretized	0.052298246	0.02227622	0.052773843
monk3	0.195311475	0.031926821	0.19531096
spambase	0.204092806	0.000579423	0.204088912
sunburn	0.212	0.138790355	0.220497279
supermarket	0.462361573	0.007167272	0.462407036
tictactoe	0.361791232	0.01359394	0.362099158
titanic	0.320064516	0.008262441	0.320500866
unbalanced	0.091042056	0.006565479	0.091391125
V1	0.12622069	0.005416082	0.126050467
vote	0.097593103	0.004988294	0.097475891
weather	0.281285714	0.109975102	0.279767513

Figure 7: Table of Experimental Results using Naive Bayes

Classifiers	J48		
Data Set	Average	Standart Deviation	Mean Absolute Error
BC	0.36756993	0.026105823	0.000261058
BreastCancer	0.366272727	0.024963776	0.000249638
creditg	0.231118	0.010541784	0.000105418
diabetes	0.238263021	0.012501084	0.000125011
glass2	0.144515337	0.020562764	0.000205628
HE	0.1278	0.021602531	0.000216025
heart-train	0.103964286	0.019086424	0.000190864
HO	0.238274457	0.018455449	0.000184554
HY	0.011056592	0.001291986	1.29199E-05
ionosphere	0.005267806	0.002766824	2.76682E-05
labor	0.196789474	0.043159189	0.000431592
labordiscretized	0.331526316	0.055662598	0.000556626
monk3	0.111319672	0.02133702	0.00021337
spambase	0.052053249	0.002434675	2.43468E-05
sunburn	0.29375	0.137767621	0.001377676
supermarket	0.462681867	0.007417129	7.41713E-05
tictactoe	0.091705637	0.006906449	6.90645E-05
titanic	0.309377101	0.008406887	8.40689E-05
unbalanced	0.027660047	0.003980248	3.98025E-05
V1	0.106485057	0.011654892	0.000116549
vote	0.052154023	0.00798436	7.98436E-05
weather	0	0	0

Figure 8: Table of Experimental Results using J48

Classifiers	ZeroR		
Data Set	Average	Standart Deviation	Mean Absolute Error
BC	0.419685315	0.02758769	0.418317793
BreastCancer	0.417118881	0.026478884	0.418317793
creditg	0.419853	0.014463699	0.420159681
diabetes	0.45513151	0.016935469	0.454491342
glass2	0.497319018	0.039644811	0.497750511
HE	0.330374194	0.032403518	0.32985412
heart-train	0.495928571	0.041140795	0.495070423
HO	0.463711957	0.025392128	0.466157462
HY	0.091195384	0.003812487	0.09117936
ionosphere	0.460641026	0.025917868	0.4604489
labor	0.454596491	0.062887102	0.457032412
labordiscretized	0.455807018	0.065145214	0.457032412
monk3	0.498991803	0.043861655	0.499867795
spambase	0.477639426	0.007035349	0.477556736
sunburn	0.4835	0.171229582	0.475
supermarket	0.462342555	0.007022805	0.462407036
tictactoe	0.454100209	0.015827441	0.453007568
titanic	0.43755702	0.009864623	0.437423628
unbalanced	0.028992991	0.004134822	0.028745398
V1	0.473970115	0.023532018	0.474220784
vote	0.475082759	0.022766997	0.474220784
weather	0.466	0.129888703	0.464285714

Figure 9: Table of Experimental Results using ZeroR

Classifiers	Logistic		
Data Set	Average	Standart Deviation	Mean Absolute Error
BC	0.322398601	0.02387838	0.322266541
BreastCancer	0.320356643	0.022763151	0.322266541
creditg	0.291668	0.012294345	0.292050575
diabetes	0.305903646	0.013950195	0.306294456
glass2	0.343196319	0.03117829	0.344123273
HE	0.139864516	0.021955911	0.140301083
heart-train	0.182885714	0.025470972	0.182842424
HO	0.186130435	0.015987547	0.185841284
HY	0.025482453	0.002063806	0.025506437
ionosphere	0.094247863	0.011999064	0.09459784
labor	0	0	1.60342E-07
labordiscretized	0	0	2.55632E-09
monk3	0.121836066	0.023359398	0.123495732
spambase	0.115720713	0.003750877	0.115683595
sunburn	0	0	2.13219E-07
supermarket	0.46239075	0.007295862	0.462390787
tictactoe	0.025588727	0.003523053	0.025809621
titanic	0.324998183	0.008397747	0.325281826
unbalanced	0.019617991	0.003409268	0.019770457
V1	0.101255172	0.010953387	0.101596176
vote	0.043006897	0.006997726	0.042771009
weather	0.206071429	0.086757057	0.206590091

Figure 10: Table of Experimental Results using Logistic

Classifiers	Decision Table		
Data Set	Average	Standart Deviation	Mean Absolute Error
BC	0.328325175	0.024118977	0.329306202
BreastCancer	0.328153846	0.025900329	0.329306202
creditg	0.338747	0.01263862	0.337436053
diabetes	0.321699219	0.015193082	0.322303567
glass2	0.256239264	0.028619795	0.255847476
HE	0.265851613	0.031414692	0.265128918
heart-train	0.318064286	0.034162004	0.317178229
HO	0.220638587	0.018649319	0.219555526
HY	0.01887828	0.001945652	0.018944378
ionosphere	0.162641026	0.016705415	0.16223143
labor	0.184403509	0.045011067	0.185386762
labordiscretized	0.287807018	0.054035263	0.286988304
monk3	0.175778689	0.031876815	0.176255507
spambase	0.197901326	0.005134208	0.197798323
sunburn	0.470375	0.169127433	0.458333333
supermarket	0.325142857	0.005983465	0.325049466
tictactoe	0.278204593	0.013356329	0.277485036
titanic	0.310183099	0.008571156	0.310322537
unbalanced	0.027869159	0.004159919	0.027628437
V1	0.172347126	0.014764888	0.173099059
vote	0.091257471	0.011387488	0.091033609
weather	0.458928571	0.128643207	0.452380952

Figure 11: Table of Experimental Results using Decision Table

References

[1] Evaluation of classifiers performance, April 2013.

[2] Peter A Flach. Roc analysis. In *Encyclopedia of Machine Learning*, pages 869–875. Springer, 2010.

[3] Eibe Frank. Threshold selector.

[4] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, nov 2009.

[5] José Hernández-Orallo, Peter Flach, and Cesar Ferri. A unified view of performance metrics: translating threshold choice into expected classification loss. *Journal of Machine Learning Research*, 13:2813–2869, 2012.

[6] George John and Pat Langley. *Estimating Continuous Distributions in Bayesian Classifiers*. Morgan Kaufmann, 1995.

[7] The University of Waikato. Weka.

[8] DMW Powers. Evaluation: From precision , recall and f-measure to roc., informedness, markedness &. *Journal of Machine*, 2(1):37–63, 2011.

[9] Lior Rokach and Oded Maimon. Top-down induction of decision trees classifiers-a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(4):476–487, 2005.