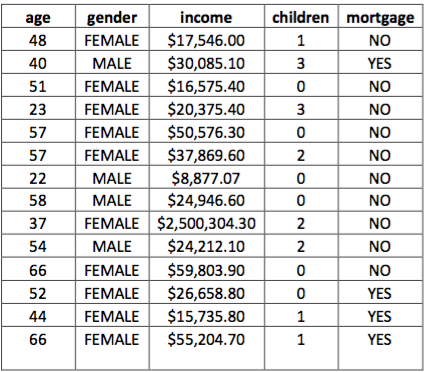
Felix Singerman

7970742

Assignment 2

 Mortgage is the target class

1. Show all the steps you would follow to determine the first attribute to split on for the data included in the table, using the information gain criterion.

Since mortgage is the target class we will be classifying it as;

Class P: has\_mortgage = “YES”

Class N: has\_mortgage = “NO”

We need to select the attribute with the highest information gain. Since attribute ‘income’ and ‘age’ (truncated to whole number, but concept of age is continuous) are continuous we need to determine the best split point for the attribute. We do this by sorting them in increasing order. The midpoint between each pair of adjacent values is considered as a possible split point for the attributes. The point with the minimum expected information requirement for the attribute is selected as the split-point for that attribute. We then split it by taking D1 as the set of tuples in D where A <= split-point, and D2 as the set of tuples in D where A > split-point.

We let pi be the probability that an arbitrary tuple in D belongs to class Ci estimated by |Ci,D|/|D|. For each discrete attribute we will take the entropy needed to classify a tuple and subtract it by information needed in order to get the information gained.

Entropy = Info(D) = I(4/10) = - 4/14 log(4/14) – 10/14 log(10/14)

(I abbreviate the next steps by not including the log calculation and just leaving the I(x/X))

We then take the Info X(D) for each attribute (i.e.: gender, children). For example Info gender(D) = 10/14 I(3, 7) + 4/14 I(1,3)

Info chidren(D) = 6/14 I(1, 5) + 3/14 I(2,1) + 3/14 I(0,3) + 2/14 I(1,1)

So we sort income and get the midpoint which is 25802.7

So Info Income(D) = 7/14 I(1,6) + 7/14 I(3 ,4)

We do the same for age median 51.5:

Indo Age(D) = 7/14 I(2,5) + 7/14 I(2,5)

Gain(gender) = Info(D) – Info gender(D)

Gain(children) = Info(D) – Info children(D)

Gain(income) = Info(D) – Info income(D)

Gain(age) = Info(D) – Info age(D)

The highest result will be our first attribute split.

1. Suppose that we remove the “Mortgage” class label from the dataset. Show all the steps you would follow when applying the k-means cluster analysis algorithm to the data, with k = 3.

The algorithm will be implemented in 4 steps. Partition objects into k = 3 nonempty subsets. Compute seed points as the centroids of the clusters of the current partitioning. Assign each object to the cluster with the nearest seed point. Go back to step 2 and stop when the assignment does not change.

We need to partition our database D of 14 objects into 3 clusters, such that the sum of squared distances is minimized (where ci is the centroid). Each cluster is represented by the center of the cluster.

Use the Customer.CSV file attached to this assignment to answer the following questions.

1. Apply the C4.5 (J48 in WEKA) decision tree algorithm to this data. Show your resultant confusion matrix and comment on the accuracy, precision, recall, sensitivity and specificity of the model you constructed.

For the following questions I used a 10-fold cross-validation: divide dataset into 10 parts, hold out each part in turn, average the results, each data point used once for testing, 9 times for training. I used an unpruned J48 tree.

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 386 64.3333 %

Incorrectly Classified Instances 214 35.6667 %

Kappa statistic -0.0126

Mean absolute error 0.4437

Root mean squared error 0.4832

Relative absolute error 99.7921 %

Root relative squared error 102.5088 %

Total Number of Instances 600

=== Detailed Accuracy By Class ===

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

0.940 0.950 0.664 0.940 0.778 -0.020 0.521 0.671 NO

0.050 0.060 0.294 0.050 0.085 -0.020 0.521 0.349 YES

Weighted Avg. 0.643 0.653 0.541 0.643 0.547 -0.020 0.521 0.563

=== Confusion Matrix ===

a b <-- classified as

376 24 | a = NO

190 10 | b = YES

The precision and recall and F-Measure were much higher for class NO\_Mortgage. The results of the YES class were extremely low. Whereas the NO class was pretty very accurate. 190 of the errors took place in misclassify YES’s and NO’s compared to misclassifying only 24 NO’s as YES’s.

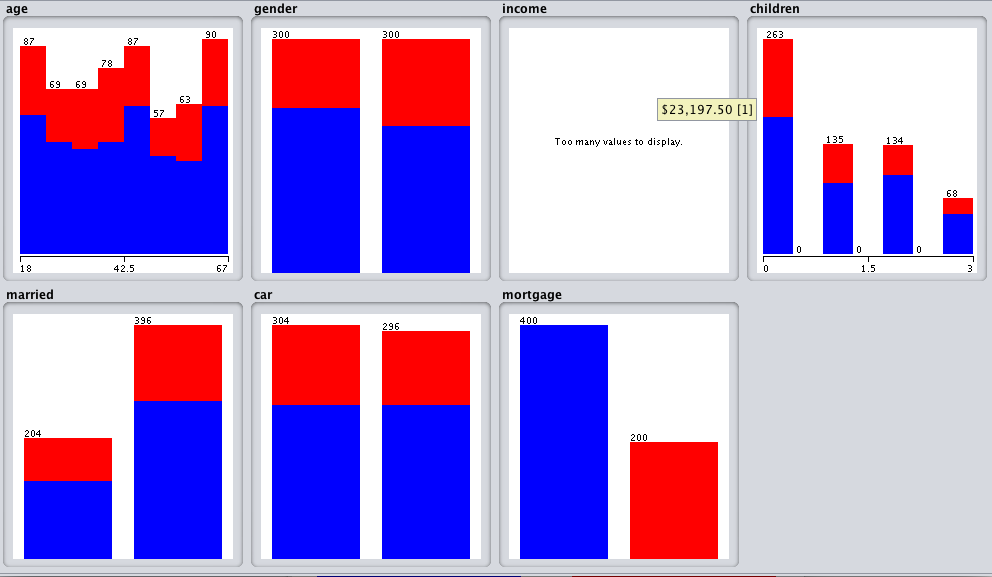
**Describe more!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!:**

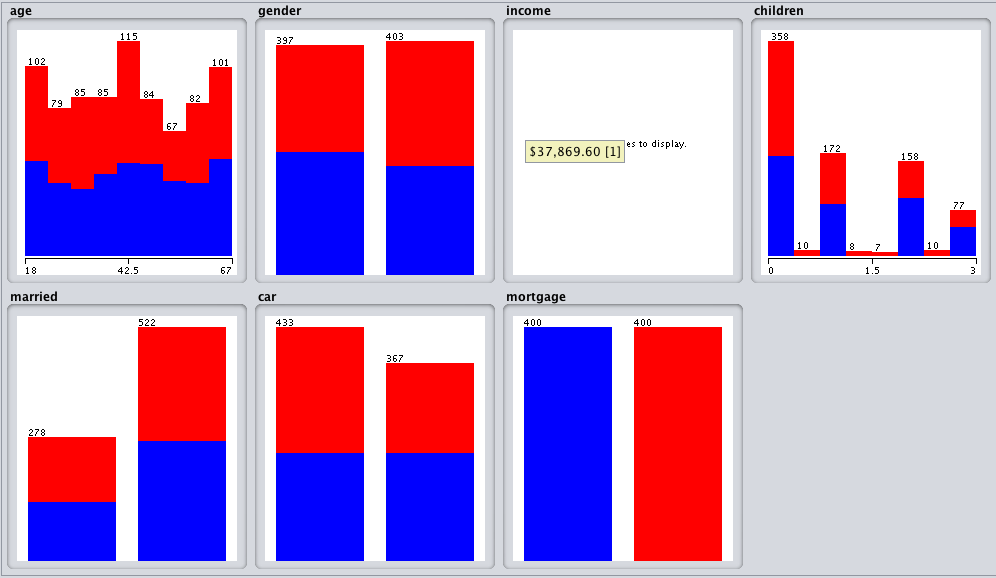
1. This dataset is imbalanced and the initial results are therefore poor. Explain how you would address this issue during preprocessing, training and model evaluation.

During preprocessing, we can do sampling which is rebalancing the dataset. We can do oversampling which is where instances of the minority classes are introduced into the training set. Past examples may be accumulated to augment the minority examples in the training sets. We can also do under-sampling where instances of the majority classes are removed. This can cause information loss. We can do a combination of the 2 as well. We can also used a k-means cluster where each cluster is oversampled such that all clusters of the same class have an equal number of instances and all classes have the same size. 

Since information gain measure is biased towards attributes with a large number of values we can use a gain ratio during training. GainRatio(A) = Gain(A)/SplitInfo(A). The attribute with the maximum gain ratio is selected as the splitting attribute.

In weka, I used a 5 nearest neighbors SMOTE on the data for balancing. Below is the data before and after the smote. The target class of the balancing is mortgage.





**During training: ???????????????????????????????**

**During model evaluation: ?????????????????????????????**

1. Reapply the C4.5 decision tree algorithm to your modified data and determine whether your modifications have improved the performance.

I again used 10-fold unpruned tree.

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 649 81.125 %

Incorrectly Classified Instances 151 18.875 %

Kappa statistic 0.6225

Mean absolute error 0.345

Root mean squared error 0.4168

Relative absolute error 69 %

Root relative squared error 83.3667 %

Total Number of Instances 800

=== Detailed Accuracy By Class ===

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

0.998 0.375 0.727 0.998 0.841 0.671 0.810 0.726 NO

0.625 0.003 0.996 0.625 0.768 0.671 0.810 0.810 YES

Weighted Avg. 0.811 0.189 0.861 0.811 0.804 0.671 0.810 0.768

=== Confusion Matrix ===

a b <-- classified as

399 1 | a = NO

150 250 | b = YES

**Explain results !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!**

1. Ensembles of classifiers, or so-called meta-leaners, are often used in order to improve the accuracy of base learners such as decision trees. Explore whether applying a Boosting ensemble, such as AdaBoost, to this dataset improve the performance. Show your results and discuss your findings.

I used AdaBoost as seen below: The AdaBoost used a J48 unpruned tree on the balanced data.

=== Summary ===

Correctly Classified Instances 584 73 %

Incorrectly Classified Instances 216 27 %

Kappa statistic 0.46

Mean absolute error 0.2951

Root mean squared error 0.4796

Relative absolute error 59.0223 %

Root relative squared error 95.9166 %

Total Number of Instances 800

=== Detailed Accuracy By Class ===

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

0.723 0.263 0.734 0.723 0.728 0.460 0.814 0.717 NO

0.738 0.278 0.727 0.738 0.732 0.460 0.814 0.859 YES

Weighted Avg. 0.730 0.270 0.730 0.730 0.730 0.460 0.814 0.788

=== Confusion Matrix ===

a b <-- classified as

289 111 | a = NO

105 295 | b = YES

AdaBoost is giving better results for the YES class, however, it has decreased for the NO class.

**Adaboost is overfitting the data so our results are lower. EXPLAIN MORE**

1. Explain what a global outlier is and suggest an algorithm that you could use to identify a global outlier in the dataset. List one global outlier that you found.

Global outliers also known as Point Anomaly, finds the object that significantly deviates from the rest of the data set.

We can use several methods in order to find outliers. We can use the Grubb’s test which is a statistical method under normal distribution and compute for each object its z-score. We could also use a histogram. One outlier I found is in the income attribute the value: 2,500,304.30. I could of also discretized the data in income then looked at the histogram and seen which one is an outlier. I used R to plot out the data in the income row. Screenshot below.

