ITCS 6155: Knowledge-Based Systems
Project Report

Discretize and Classification Dataset

Venkata Naga Akash Ungarala	**800902992**
Mani Venkata Rajesh Guttikonda	**800898839**
Snigdha Sree Paladugu	**800903113**
Kalyana Krishna Sampath Pendyala	**800907556**

Dataset

Data is collected from the URL: http://www.saatchiart.com/paintings/fine-art

Saatchi is an art gallery which offers an unparalleled selection of paintings, drawings, sculpture and photography in a range of prices, and it provides artists from around the world with an expertly curated environment in which to exhibit and sell their work.

Tools Used

Python Compiler
WEKA 3.6 (stable version)

Python (Data Extraction)

We have used Python code to extract structured data set from the website.

WEKA

Weka is an open source and popular suite of machine learning workbench that contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. The original non-Java version of Weka was a TCL/TK front-end to (mostly third-party) modeling algorithms implemented in other programming languages, plus data preprocessing utilities in C, and a Make file-based system for running machine learning experiments. This original version was primarily designed as a tool for analyzing data from agricultural domains but the more recent fully Java-based version (Weka 3), is now used in many different application areas, in particular for educational purposes and research.

Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining, but there is separate software for converting a collection of linked database tables into a single table that is suitable for processing using Weka. Another important area that is currently not covered by the algorithms included in the Weka distribution is sequence modeling.

Work Flow

Extraction of Structured Dataset and Pre-Processing

- ✓ We have extracted 480+ paintings by running python code to access the web data. The extracted data set is cleaned and refined by deleting the details of the paintings with either Size or Price is 0 or Name or Artist having any special characters.
- ✓ The final data set contains 336 paintings. The first 9 features (Name, Artist, Size, Price, Views, Favorites, Date, Subject, Medium) are mandatory for the project, the other 4 features (Material, Category, Country, Crafts Availability) are the additional features added.
- ✓ Once the data is clean, it is loaded into WEKA Explorer.

Discretize

We have Price as the decision feature. In the data set, the price column is sorted. Since we have to split the Price into 3 intervals, we can split it as the levels Low, Medium and High. We need to choose the split to get a maximum entropy gain for the decision feature considered. In order to have higher entropy, the records should be distributed with nearly equal frequency in all the 3 ranges, so we choose the Equal Frequency Binning. We had the count as below

Low 115 Medium 112 High 109

Entropy Gain for Price

$E(D) = -((115/336)*\log_3(115/336)+(112/336)*\log_3(112/336)+(109/336)*\log_3(109/336))$
$E(D) = -0.3422619*(-0.97593942519) - 0.3333333*(-1) - 0.32440476*(-1.0247139006)$
$E(D) = 0.999782249$

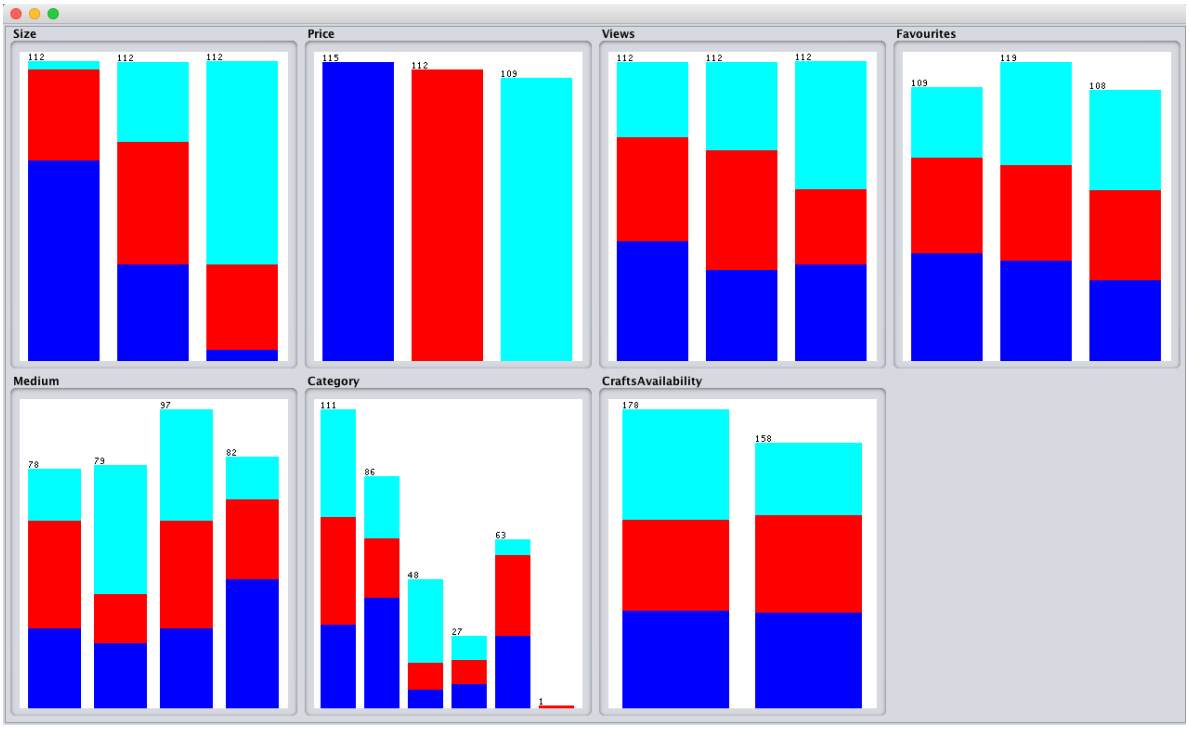
We have calculated the initial entropy and made the split based on the entropy gain of the split. We have chosen the split which has higher Entropy Gain and decided to have the price split into 3 intervals as below

[-infinity … 825] [825 … 2350] [2350 … +infinity]

Classification using WEKA

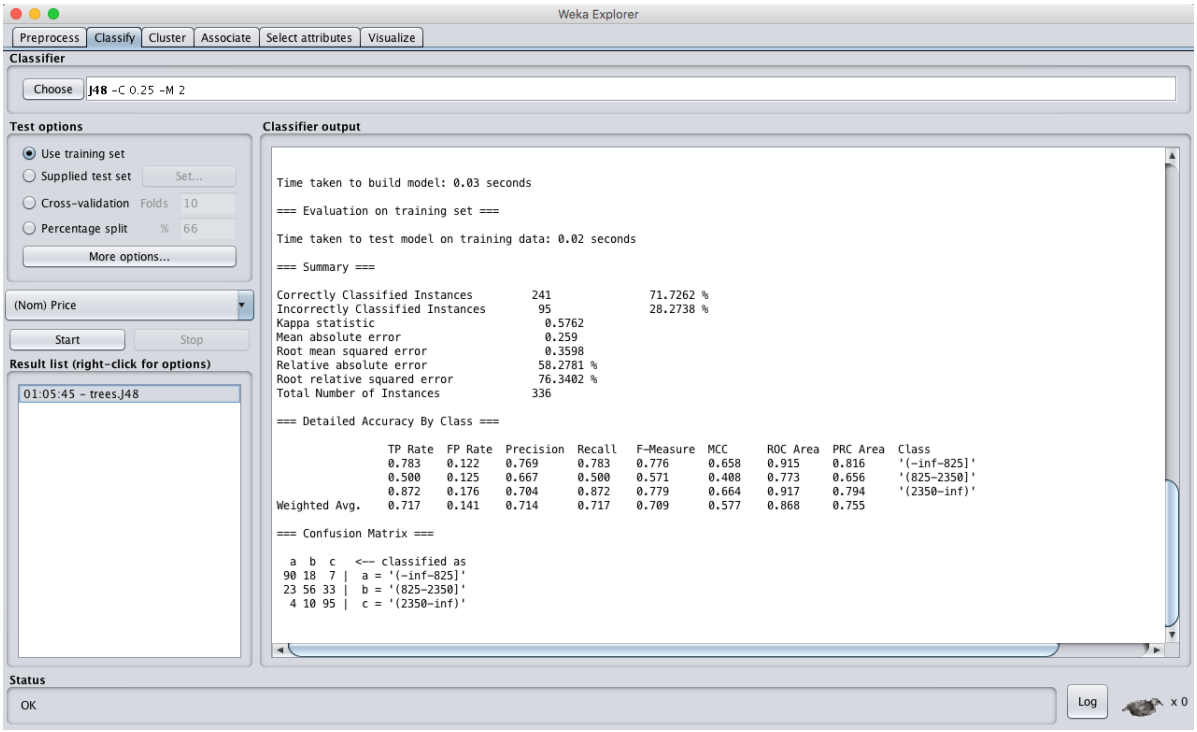
Since classification should not depend on the features Painting Name and Artist, remove them before performing classification. We also removed certain unwanted attributes which we felt doesn't impact the sales of the paintings. Select Price as the decision feature and apply different algorithms to compare the results and choose the algorithm that gives the best number of correctly classified records.

Visualize all the features



Trees-J48 as Classifier

Classifier	J48 -C 0.25 –M 2	**Test options**	Use training set
Unpruned	True	**Decision feature**	Price



Classifier Output

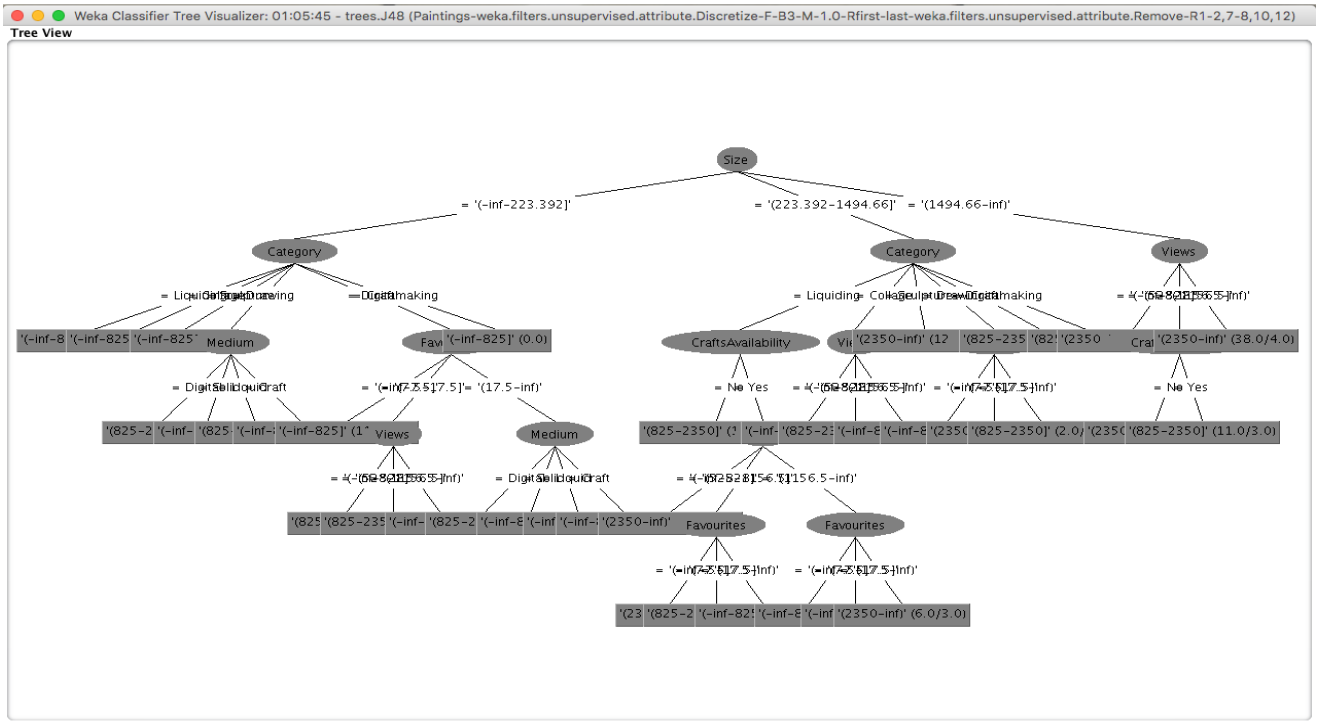
Correctly Classified Instances	241	71.7262 %
Incorrectly Classified Instances	95	28.2738 %
Kappa statistic	0.5762	
Mean absolute error	0.259	
Root mean squared error	0.3598	
Relative absolute error	58.2781 %	
Root relative squared error	76.3402 %	
Total Number of Instances	336	

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.783	0.122	0.769	0.783	0.776	0.658	0.915	0.816	'(-inf-825]'
0.500	0.125	0.667	0.500	0.571	0.408	0.773	0.656	'(825-2350]'
0.872	0.176	0.704	0.872	0.779	0.664	0.917	0.794	'(2350-inf)'

Confusion Matrix

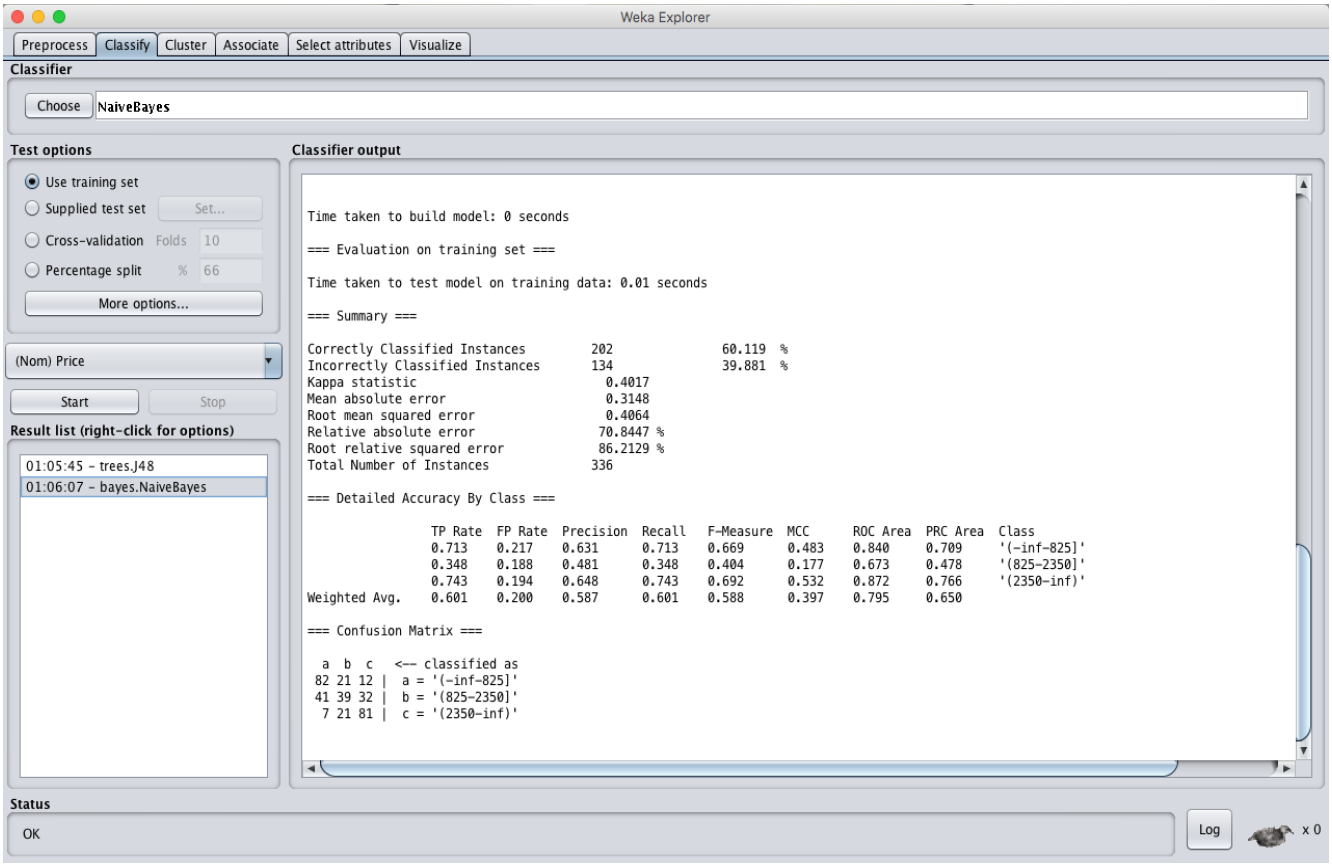
$$\begin{matrix} a & b & c \\ \begin{bmatrix} 90 & 18 & 7 \\ 23 & 56 & 33 \\ 4 & 10 & 95 \end{bmatrix} & \begin{matrix} a = '(-\inf - 825]' \\ b = '(825 - 2350]' \\ c = '(2350 - \inf)' \end{matrix} \end{matrix}$$

Tree Visualizer



Bayes-Naive Bayes as Classifier

Classifier	Naive Bayes	**Test options**	Use training set
Unpruned	True	**Decision feature**	Price



Classifier Output

Correctly Classified Instances	202	60.119 %
Incorrectly Classified Instances	134	39.881 %
Kappa statistic	0.4017	
Mean absolute error	0.3148	
Root mean squared error	0.4064	
Relative absolute error	70.8447 %	
Root relative squared error	86.2129 %	
Total Number of Instances	336	

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.713	0.217	0.631	0.713	0.669	0.483	0.840	0.709	'(-inf-825]'
0.348	0.188	0.481	0.348	0.404	0.177	0.673	0.478	'(825-2350]'
0.743	0.194	0.648	0.743	0.692	0.532	0.872	0.766	'(2350-inf)'

Confusion Matrix

$$\begin{matrix} a & b & c \\ \begin{bmatrix} 82 & 21 & 12 \\ 41 & 39 & 32 \\ 7 & 21 & 81 \end{bmatrix} & \begin{matrix} a = '(-\inf - 825]' \\ b = '(825 - 2350]' \\ c = '(2350 - \inf)' \end{matrix} \end{matrix}$$

Comparison of Algorithms and Result

Since the percentage of correctly classified instances for J48 and Naïve Bayes algorithms is 71.7262 and 60.119 respectively. This clearly shows that J48 (Trees) is best used to classify as it has high accuracy when compared with other algorithms like Naive Bayes (Bayes).