Based data mining approach for quality control

Science, Computer Science

Classification-Based Data Mining Approach For Quality Control In Wine Production GUIDED BY: | | SUBMITTED BY:| Jayshri Patel| | Hardik Barfiwala| INDEX Sr No| Title| Page No. | 1| Introduction Wine Production| | 2| Objectives| | 3| Introduction To Dataset| | 4| Pre-Processing| | 5| Statistics Used In Algorithms| | 6| Algorithms Applied On Dataset| | 7| Comparison Of Applied Algorithm | | 8| Applying Testing Dataset| | 9| Achievements| | 1.

INTRODUCTION TO WINE PRODUCTION * Wine industry is currently growing well in the market since the last decade. However, the quality factor in wine has become the main issue in wine making and selling. * To meet the increasing demand, assessing the quality of wine is necessary for the wine industry to prevent tampering of wine quality as well as maintaining it. * To remain competitive, wine industry is investing in new technologies like data mining for analyzing taste and other properties in wine. Data mining techniques provide more than summary, but valuable information such as patterns and relationships between wine properties and human taste, all of which can be used to improve decision making and optimize chances of success in both marketing and selling. * Two key elements in wine industry are wine certification and quality assessment, which are usually conducted via physicochemical and sensory tests. * Physicochemical tests are lab-based and are used to characterize physicochemical properties in wine such as its density, alcohol or pH values. * Meanwhile, sensory tests such as taste preference are performed by human experts.

Taste is a particular property that indicates quality in wine, the success of wine industry will be greatly determined by consumer satisfaction in taste requirements. * Physicochemical data are also found useful in predicting

human wine taste preference and classifying wine based on aroma chromatograms. 2. OBJECTIVE * Modeling the complex human taste is an important focus in wine industries. * The main purpose of this study was to predict wine quality based on physicochemical data. * This study was also conducted to identify outlier or anomaly in sample wine set in order to detect ruining of wine. 3. INTRODUCTION TO DATASET

To evaluate the performance of data mining dataset is taken into consideration. The present content describes the source of data. * Source Of Data Prior to the experimental part of the research, the data is gathered. It is gathered from the UCI Data Repository. The UCI Repository of Machine Learning Databases and Domain Theories is a free Internet repository of analytical datasets from several areas. All datasets are in text files format provided with a short description. These datasets received recognition from many scientists and are claimed to be a valuable source of data. * Overview Of Dataset INFORMATION OF DATASET|

Title:| Wine Quality| Data Set Characteristics:| Multivariate| Number Of Instances:| WHITE-WINE : 4898 RED-WINE : 1599 | Area:| Business| Attribute Characteristic:| Real| Number Of Attribute:| 11 + Output Attribute| Missing Value:| N/A| * Attribute Information * Input variables (based on physicochemical tests) * Fixed Acidity: Amount of Tartaric Acid present in wine. (In mg per liter) Used for taste, feel and color of wine. * Volatile Acidity: Amount of Acetic Acid present in wine. (In mg per liter) Its presence in wine is mainly due to yeast and bacterial metabolism. * Citric Acid: Amount of Citric Acid present in wine. In mg per liter) Used to acidify wine that are too basic and as a flavor additive. * Residual Sugar: The concentration of sugar

remaining after fermentation. (In grams per liter) * Chlorides: Level of Chlorides added in wine. (In mg per liter) Used to correct mineral deficiencies in the brewing water. * Free Sulfur Dioxide: Amount of Free Sulfur Dioxide present in wine. (In mg per liter) * Total Sulfur Dioxide: Amount of free and combined sulfur dioxide present in wine. (In mg per liter) Used mainly as preservative in wine process. * Density: The density of wine is close to that of water, dry wine is less and sweet wine is higher. In kg per liter) * PH: Measures the quantity of acids present, the strength of the acids, and the effects of minerals and other ingredients in the wine. (In values) * Sulphates: Amount of sodium metabisulphite or potassium metabisulphite present in wine. (In mg per liter) * Alcohol: Amount of Alcohol present in wine. (In percentage) * Output variable (based on sensory data) * Quality (score between 0 and 10) : White Wine : 3 to 9 Red Wine : 3 to 8 4. PRE-PROCESSING * Pre-processing Of Data Preprocessing of the dataset is carried out before mining the data to remove the different lacks of the information in the data source.

Following different process are carried out in the preprocessing reasons to make the dataset ready to perform classification process. * Data in the real world is dirty because of the following reason. * Incomplete: Lacking attribute values, lacking certain attributes of interest, or containing only aggregate data. * E. g. Occupation=“” * Noisy : Containing errors or outliers. * E. g. Salary=“-10” * Inconsistent : Containing discrepancies in codes or names. * E. g. Age=“ 42” Birthday=“ 03/07/1997” * E. g. Was rating “ 1, 2, 3”, Now rating “ A, B, C” * E. g. Discrepancy between duplicate records * No quality data, no quality mining results! Quality decisions must be based on

quality data. * Data warehouse needs consistent integration of quality data. * Major Tasks in done in the Data Preprocessing are, * Data Cleaning * Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies. * Data integration * Integration of multiple databases, data cubes, or files. * The dataset provided from given data source is only in one single file. So there is no need for integrating the dataset. * Data transformation * Normalization and aggregation * The dataset is in Normalized form because it is in single data file. * Data reduction Obtains reduced representation in volume but produces the same or similar analytical results. * The data volume in the given dataset is not very huge, the procedure of performing different algorithm is easily done on dataset so the reduction of dataset is not needed on the data set * Data discretization * Part of data reduction but with particular importance, especially for numerical data. * Need for Data Preprocessing in wine quality, * For this dataset Data Cleaning is only required in data pre-processing. * Here, NumericToNominal, InterquartileRange and RemoveWithValues filters are used for data pre-processing. * NumericToNominal Filter weka. filters. unsupervised. attribute. NumericToNominal) * A filter for turning numeric attribute into nominal once. * In our dataset, Class attribute " Quality" in both dataset (Red-wine Quality, White-wine Quality) have a type " Numeric". So after applying this filter, class attribute " Quality" convert into type " Nominal". * And Red-wine Quality dataset have class names 3, 4, 5 ... 8 and White-wine Quality dataset have class names 3, 4, 5 ... 9. * Because of classification does not apply on numeric type class field, there is a need for this filter. * InterquartileRange Filter (weka. filters. unsupervised. attribute.

InterquartileRange) A filter for detecting outliers and extreme values based on interquartile ranges. The filter skips the class attribute. * Apply this filter for all attribute indices with all default options. * After applying, filter adds two more fields which names are “ Outliers” and “ ExtremeValue”. And this fields has two types of label “ No” and “ Yes”. Here “ Yes” label indicates, there are outliers and extreme values in dataset. * In our dataset, there are 83 extreme values and 125 outliers in White-wine Quality dataset and 69 extreme values and 94 outliers in Red-wine Quality. * RemoveWithValues Filter (weka. filters. unsupervised. instance.

RemoveWithValues) * Filters instances according to the value of an attribute. * This filter has two options which are “ AttributeIndex” and “ NominalIndices”. * AttributeIndex choose attribute to be use for selection and NominalIndices choose range of label indices to be use for selection on nominal attribute. * In our dataset, AttributeIndex is “ last” and NominalIndex is also “ last”, so It will remove first 83 extreme values and then 125 outliers in White-wine Quality dataset and 69 extreme values and 94 outliers in Red-wine Quality. * After applying this filter on dataset remove both fields from dataset. * Attribute Selection

Ranking Attributes Using Attribute Selection Algorithm| RED-WINE| RANKED| WHITE-WINE| Volatile_Acidity(2)| 0. 1248| 0. 0406| Volatile_Acidity(2)| Total_sulfer_Dioxide(7)| 0. 0695| 0. 0600| Citric_Acidity(3)| Sulphates(10)| 0. 1464| 0. 0740| Chlorides(5)| Alcohal(11)| 0. 2395| 0. 0462| Free_Sulfer_Dioxide(6)| | | 0. 1146| Density(8)| | | 0. 2081| Alcohal(11)| * The selection of attributes is performed automatically by WEKA using Info Gain Attribute Eval method. * The method evaluates the worth of an attribute by

measuring the information gain withrespectto the class. 5. STATISTICS USED IN ALGORITHMS * Statistics Measures

There are Different algorithms that can be used while performing data mining on the different dataset using weka, some of them are describe below with the different statistics measures. * Statistics Used In Algorithms * Kappa statistic * The kappa statistic, also called the kappa coefficient, is a performance criterion or index which compares the agreement from the model with that which could occur merely by chance. * Kappa is a measure of agreement normalized for chance agreement. * Kappa statistic describe that our prediction for class attribute for given dataset is how much near to actual values. * Values Range For Kappa Range| Result| lt; 0| POOR| 0-0. 20| SLIGHT| 0. 21-0. 40| FAIR| 0. 41-0. 60| MODERATE| 0. 61-0. 80| SUBSTANTIAL| 0. 81-1. 0| ALMOST PERFECT| * As above range in weka algorithm evaluation if value of kappa is near to 1 then our predicted values are accurate to actual values so, applied algorithm is accurate. Kappa Statistic Values For Wine Quality DataSet| Algorithm| White-wine Quality| Red-wine Quality| K-Star| 0. 5365| 0. 5294| J48| 0. 3813| 0. 3881| Multilayer Perceptron| 0. 2946| 0. 3784| * Mean absolute error (MAE) * Mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by, Mean absolute Error For Wine Quality DataSet| Algorithm| White-wine Quality| Red-wine Quality| K-Star| 0. 1297| 0. 1381| J48| 0. 1245| 0. 1401| Multilayer Perceptron| 0. 1581| 0. 1576| * Root Mean Squared Error * If you have some data and try to make a curve (a formula) fit them, you can graph and see how close the curve is to the points. Another measure of how well the curve

fits the data is Root Mean Squared Error. * For each data point, CalGraph calculates the value of y from the formula. It subtracts this from the data's y-value and squares the difference. All these squares are added up and the sum is divided by the number of data. * Finally CalGraph takes the square root. Written mathematically, Root Mean Square Error is Root Mean Squared Error For Wine Quality DataSet| Algorithm| White-wine Quality| Red-wine Quality| K-Star| 0. 2428| 0. 2592| J48| 0. 3194| 0. 3354| Multilayer Perceptron| 0. 2887| 0. 3023| * Root Relative Squared Error * The root relative squared error is relative to what it would have been if a simple predictor had been used. More specifically, this simple predictor is just the average of the actual values. Thus, the relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the simple predictor. * By taking the square root of therelative squared error one reduces the error to the same dimensions as the quantity being predicted. * Mathematically, the root relative squared error Ei of an individual program i is evaluated by the equation: * where P(ij) is the value predicted by the individual program i for sample case j (out of n sample cases); Tj is the target value for sample case j; andis given by the formula: * For a perfect fit, the numerator is equal to 0 and Ei = 0.

So, the Ei index ranges from 0 to infinity, with 0 corresponding to the ideal. Root Relative Squared Error For Wine Quality DataSet| Algorithm| White-wine Quality| Red-wine Quality| K-Star| 78. 1984%| 79. 309 %| J48| 102. 9013 %| 102. 602 %| Multilayer Perceptron| 93. 0018 %| 92. 4895 %| * Relative Absolute Error * The relative absolute error is very similar to the relative squared error in the sense that it is also relative to a simple predictor, which

is just the average of the actual values. In this case, though, the error is just the total absolute error instead of the total squared error. Thus, the relative absolute error takes the total absolute error and normalizes it by dividing by the total absolute error of the simple predictor. Mathematically, the relative absolute error Ei of an individual program i is evaluated by the equation: * where P(ij) is the value predicted by the individual program i for sample case j (out of n sample cases); Tj is the target value for sample case j; andis given by the formula: * For a perfect fit, the numerator is equal to 0 and Ei = 0. So, the Ei index ranges from 0 to infinity, with 0 corresponding to the ideal.

Relative Absolute Squared Error For Wine Quality DataSet| Algorithm| White-wine Quality| Red-wine Quality| K-Star| 67. 2423 %| 64. 5286 %| J48| 64. 577 %| 65. 4857 %| Multilayer Perceptron| 81. 9951 %| 73. 6593 %| * Various Rates * There are four possible outcomes from a classifier. * If the outcome from a prediction is p and the actual value is also p, then it is called a true positive (TP). * However if the actual value is n then it is said to be a false positive (FP). * Conversely, a true negative (TN) has occurred when both the prediction outcome and the actual value are n. And false negative (FN) is when the prediction outcome is n while the actual value is p. * Absolute Value | P| N| TOTAL| p'| True positive| false positive| P'| n'| false negative| True negative| N'| Total| P| N| | * ROC Curves * While estimating the effectiveness and accuracy of data mining technique it is essential to measure the error rate of each method. * In the case of binary classification tasks the error rate takes and components under consideration. * The ROC

analysis which stands for Receiver Operating Characteristics is applied. * The sample ROC curve is presented in the Figure below.

The closer the ROC curve is to the top left corner of the ROC chart the better the performance of the classifier. * Sample ROC curve (squares with the usage of the model, triangles without). The line connecting the square with triage is the benefit from the usage of the model. * It plots the curve which consists of x-axis presenting false positive rate and y-axis which plots the true positive rate. This curve model selects the optimal model on the basis of assumed class distribution. * The ROC curves are applicable e. g. in decision tree models or rule sets. * Recall, Precision and F-Measure There are four possible results of classification. * Different combination of these four error and correct situations are presented in the scientific literature on topic. * Here three popular notions are presented. The introduction of these classifiers is explained by the possibility of high accuracy by negative type of data. * To avoid such situation recall and precision of the classification are introduced. * The F measure is the harmonic mean of precision and recall. * The formal definitions of these measures are as follow : PRECSION = TPTP+FP RECALL = TPTP+FN

F-Measure = 21PRECSION+1RECALL * These measures are introduced especially in information retrieval application. * Confusion Matrix * A matrix used to summarize the results of a supervised classification. * Entries along the main diagonal are correct classifications. * Entries other than those on the main diagonal are classification errors. 6. ALGORITHMS * K-Nearest Neighbor Classifiers * Nearest neighbor classifiers are based on learning by analogy. * The training samples are described by n-dimensional numeric

attributes. Each sample represents a point in an n-dimensional space. In this way, all of the training samples are stored in an n-dimensional pattern space. When given an unknown sample, a k-nearest neighbor classifier searches the pattern space for the k training samples that are closest to the unknown sample. * These k training samples are the k-nearest neighbors of the unknown sample. “ Closeness" is defined in terms of Euclidean distance, where the Euclidean distance between two points, , * The unknown sample is assigned the most common class among its k nearest neighbors. When k = 1, the unknown sample is assigned the class of the training sample that is closest to it in pattern space. Nearest neighbor classifiers are instance-based or lazy learners in that they store all of the training samples and do not build a classifier until a new (unlabeled) sample needs to be classified. * Lazy learners can incur expensive computational costs when the number of potential neighbors (i. e. , stored training samples) with which to compare a given unlabeled sample is great. * Therefore, they require efficient indexing techniques. As expected, lazy learning methods are faster at training than eager methods, but slower at classification since all computation is delayed to that time.

Unlike decision tree induction and back propagation, nearest neighbor classifiers assign equal weight to each attribute. This may cause confusion when there are many irrelevant attributes in the data. * Nearest neighbor classifiers can also be used for prediction, i. e. to return a real-valued prediction for a given unknown sample. In this case, the classifier returns the average value of the real-valued labels associated with the k nearest neighbors of the unknown sample. * In weka the previously described

algorithm nearest neighbor is given as Kstar algorithm in classifier -> lazy tab. The Result Generated After Applying K-Star On White-wine Quality Dataset Kstar Options : -B 70 -M a | Time Taken To Build Model: 0. 02 Seconds| Stratified Cross-Validation (10-Fold)| * Summary | Correctly Classified Instances | 3307 | 70. 6624 %| Incorrectly Classified Instances| 1373 | 29. 3376 %| Kappa Statistic | 0. 5365| | Mean Absolute Error | 0. 1297| | Root Mean Squared Error| 0. 2428| | Relative Absolute Error | 67. 2423 %| | Root Relative Squared Error | 78. 1984 %| | Total Number Of Instances | 4680 | | * Detailed Accuracy By Class | TP Rate| FP Rate | Precision | Recall | F-Measure | ROC Area | PRC Area| Class| | 0 | 0 | 0 | 0 | 0. 583 | 0. 004 | 3| | 0. 211 | 0. 002 | 0. 769 | 0. 211 | 0. 331 | 0. 884 | 0. 405 | 4| | 0. 672 | 0. 079 | 0. 777 | 0. 672 | 0. 721 | 0. 904 | 0. 826 | 5| | 0. 864 | 0. 378 | 0. 652 | 0. 864 | 0. 743 | 0. 84 | 0. 818 | 6| | 0. 536 | 0. 031 | 0. 797 | 0. 536 | 0. 641 | 0. 911 | 0. 772 | 7| | 0. 398 | 0. 002 | 0. 883 | 0. 398 | 0. 548 | 0. 913 | 0. 572 | 8| | 0 | 0 | 0 | 0 | 0 | 0. 84 | 0. 014 | 9| Weighted Avg. | 0. 707 | 0. 2 | 0. 725 | 0. 707 | 0. 695 | 0. 876 | 0. 787| | * Confusion Matrix| A | B | C | D | E | F| G | | Class| 0 | 0 | 4 | 9 | 0| 0 | 0 | | | A= 3| 0| 30| 49| 62| 1 | 0 | 0| | | B= 4| 0 | 7 | 919| 437| 5 | 0 | 0 | | | C= 5| 0 | 2 | 201| 1822| 81 | 2 | 0 | || D= 6| 0 | 0 | 9 | 389 | 468 | 7 | 0| || E= 7| 0 | 0 | 0 | 73 | 30 | 68 | 0 | || F= 8| 0 | 0 | 0 | 3 | 2 | 0 | 0 | || G= 9| * Performance Of The Kstar With Respect To A Testing Configuration For The White-wine Quality Dataset

Testing Method| Training Set| Testing Set| 10-Fold Cross Validation| 66% Split| Correctly Classified Instances| 99. 6581 %| 100 %| 70. 6624 %| 63. 9221 %| Kappa statistic| 0. 9949| 1| 0. 5365| 0. 4252| Mean Absolute Error| 0. 0575| 0. 0788| 0. 1297| 0. 1379| Root Mean Squared Error| 0. 1089| 0.

145| 0. 2428| 0. 2568| Relative Absolute Error| 29. 8022 %| | 67. 2423 %| 71. 2445 %| * The Result Generated After Applying K-Star On Red-wine Quality Dataset Kstar Options : -B 70 -M a | Time Taken To Build Model: 0 Seconds| Stratified Cross-Validation (10-Fold)| * Summary | Correctly Classified Instances | 1013 | 71. 379 %| Incorrectly Classified Instances| 413 | 28. 9621 %| Kappa Statistic | 0. 5294| | Mean Absolute Error | 0. 1381| | Root Mean Squared Error | 0. 2592| | Relative Absolute Error | 64. 5286 %| | Root Relative Squared Error | 79. 309 %| | Total Number Of Instances | 1426 | | * Detailed Accuracy By Class | | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area | PRC Area| Class| | 0 | 0. 001 | 0 | 0 | 0 | 0. 574 | 0. 019 | 3| | 0 | 0. 003 | 0 | 0 | 0 | 0. 811 | 0. 114 | 4| | 0. 791| 0. 176 | 0. 67| 0. 791| 0. 779 | 0. 894 | 0. 867 | 5| | 0. 769 | 0. 26 | 0. 668 | 0. 769 | 0. 715 | 0. 834 | 0. 788 | 6| | 0. 511 | 0. 032 | 0. 692 | 0. 511 | 0. 588 | 0. 936 | 0. 722 | 7| | 0. 125 | 0. 001 | 0. 5 | 0. 125 | 0. 2 | 0. 896 | 0. 142 | 8| Weighted Avg. | 0. 71| 0. 184| 0. 685| 0. 71| 0. 693| 0. 871| 0. 78| | * Confusion Matrix | A | B | C | D | E | F| | Class| 0 | 1 | 4| 1 | 0 | 0 | | | A= 3| 1 | 0 | 30| 17 | 0 | 0| | | B= 4| 0 | 2| 477| 120 | 4 | 0| | | C= 5| 0 | 1 | 103 | 444| 29 | 0| || D= 6| 0 | 0 | 8 | 76 | 90 | 2 | || E= 7| 0 | 0 | 0 | 7 | 7 | 2| || F= 8| Performance Of The Kstar With Respect To A Testing Configuration For The Red-wine Quality Dataset Testing Method| Training Set| Testing Set| 10-Fold Cross Validation| 66% Split| Correctly Classified Instances| 99. 7895 %| 100 % | 71. 0379 %| 70. 7216 %| Kappa statistic| 0. 9967| 1| 0. 5294| 0. 5154| Mean Absolute Error| 0. 0338| 0. 0436| 0. 1381| 0. 1439| Root Mean Squared Error| 0. 0675| 0. 0828 | 0. 2592| 0. 2646| Relative Absolute Error| 15. 8067 %| | 64. 5286 %| 67. 4903 %| * J48 Decision Tree * Class for generating a pruned or unpruned C4. 5 decision

tree. A decision tree is a predictive machine-learning model that decides the target value (dependent variable) of a new sample based on various attribute values of the available data. * The internal nodes of a decision tree denote the different attribute; the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable. * The attribute that is to be predicted is known as the dependent variable, since its value depends upon, or is decided by, the values of all the other attributes.

The other attributes, which help in predicting the value of the dependent variable, are known as the independent variables in the dataset. * The J48 Decision tree classifier follows the following simple algorithm: * In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly. * This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained. * For the other cases, we then look for another attribute that gives us the highest information gain. Hence we continue in this manner until we either get a clear decision of what combination of attributes gives us a particular target value, or we run out of attributes.

In the event that we run out of attributes, or if we cannot get an unambiguous result from the available information, we assign this branch a target value that the majority of the items under this branch possess. * Now that we have the decision tree, we follow the order of attribute selection as we have obtained for the tree. By checking all the respective attributes and their values with those seen in the decision tree model, we can assign or predict the target value of this new instance. * The Result Generated After Applying J48 On White-wine Quality Dataset Time Taken To Build Model: 1. 4 Seconds| Stratified Cross-Validation (10-Fold) | * Summary| | | Correctly Classified Instances| 2740 | 58. 547 %| Incorrectly Classified Instances | 1940 | 41. 453 %| Kappa Statistic | 0. 3813| | Mean Absolute Error | 0. 1245| | Root Mean Squared Error | 0. 3194| | Relative Absolute Error | 64. 5770 %| | Root Relative Squared Error| 102. 9013 %| | Total Number Of Instances | 4680| | * Detailed Accuracy By Class| | TP Rate| FP Rate| Precision| Recall| F-Measure| ROC Area| Class| | 0| 0. 002| 0| 0| 0| 0. 30| 3| | 0. 239| 0. 020| 0. 270| 0. 239| 0. 254| 0. 699| 4| | 0. 605| 0. 169| 0. 597| 0. 605| 0. 601| 0. 763| 5| | 0. 644| 0. 312| 0. 628| 0. 644| 0. 636| 0. 689| 6| | 0. 526| 0. 099| 0. 549| 0. 526| 0. 537| 0. 766| 7| | 0. 363| 0. 022| 0. 388| 0. 363| 0. 375| 0. 75| 8| | 0| 0| 0| 0| 0| 0. 496| 9| Weighted Avg. | 0. 585 | 0. 21 | 0. 582 | 0. 585 | 0. 584 | 0. 727| | * Confusion Matrix | A| B| C| D| E| F| G| || Class| 0| 2| 6| 5| 0| 0| 0| || A= 3| 1| 34| 55| 44| 6| 2| 0| || B= 4| 5| 50| 828| 418| 60| 7| 0| || C= 5| 2| 32| 413| 1357| 261| 43| 0| || D= 6| | 7| 76| 286| 459| 44| 0| || E= 7| 1| 1| 10| 49| 48| 62| 0| || F= 8| 0| 0| 0| 1| 2| 2| 0| || G= 9| * Performance Of The J48 With Respect To A Testing Configuration For The White-wine Quality Dataset Testing Method| Training Set| Testing Set| 10-Fold Cross Validation| 66%

Split| Correctly Classified Instances| 90. 1923 %| 70 %| 58. 547 %| 54. 8083 %| Kappa statistic| 0. 854| 0. 6296| 0. 3813| 0. 33| Mean Absolute Error| 0. 0426| 0. 0961| 0. 1245| 0. 1347| Root Mean Squared Error| 0. 1429| 0. 2756| 0. 3194| 0. 3397| Relative Absolute Error| 22. 0695 %| | 64. 577 %| 69. 84 %| * The Result Generated After Applying J48 On Red-wine Quality Dataset Time Taken To Build Model: 0. 17 Seconds| Stratified Cross-Validation| * Summary| Correctly Classified Instances | 867 | 60. 7994 %| Incorrectly Classified Instances | 559 | 39. 2006 %| Kappa Statistic | 0. 3881| | Mean Absolute Error | 0. 1401| | Root Mean Squared Error | 0. 3354| | Relative Absolute Error | 65. 4857 %| | Root Relative Squared Error | 102. 602 %| |

Total Number Of Instances | 1426 | | * Detailed Accuracy By Class| | Tp Rate | Fp Rate | Precision | Recall | F-measure | Roc Area | Class| | 0 | 0. 004 | 0 | 0 | 0 | 0. 573 | 3| | 0. 063 | 0. 037 | 0. 056 | 0. 063 | 0. 059 | 0. 578 | 4| | 0. 721 | 0. 258 | 0. 672 | 0. 721 | 0. 696 | 0. 749 | 5| | 0. 57 | 0. 238 | 0. 62 | 0. 57 | 0. 594 | 0. 674 | 6| | 0. 563 | 0. 64 | 0. 553 | 0. 563 | 0. 558 | 0. 8 | 7| | 0. 063 | 0. 006 | 0. 1 | 0. 063 | 0. 077 | 0. 691 | 8| Weighted Avg. | 0. 608 | 0. 214 | 0. 606 | 0. 608 | 0. 606 | 0. 718 | | * Confusion Matrix | A | B | C | D | E | F | | Class| 0 | 2 | 1 | 2 | 1 | 0 | | | A= 3| 2 | 3 | 25 | 15 | 3 | 0 | | | B= 4| 1 | 26 | 435 | 122 | 17 | 2 | | | C= 5| 2 | 21 | 167 | 329 | 53 | 5 | | | D= 6| 0 | 2 | 16 | 57 | 99 | 2 | | | E= 7| 0 | 0 | 3 | 6 | 6 | 1 | | | F= 8| Performance Of The J48 With Respect To A Testing Configuration For The Red-wine Quality Dataset Testing Method| Training Set| Testing Set| 10-Fold Cross Validation| 66% Split| Correctly Classified Instances| 91. 1641 %| 80 %| 60. 7994 %| 62. 4742 %| Kappa statistic| 0. 8616| 0. 6875| 0. 3881| 0. 3994| Mean Absolute Error| 0. 0461| 0. 0942| 0. 1401| 0. 1323| Root Mean Squared Error| 0. 1518| 0. 2618|

0. 3354| 0. 3262| Relative Absolute Error| 21. 5362 %| 39. 3598 %| 65. 4857 %| 62. 052 %| * Multilayer Perceptron * The back propagation algorithm performs learning on a multilayer feed-forward neural network. It iteratively learns a set of weights for prediction of the class label of tuples. * A multilayer feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer. * Each layer is made up of units. The inputs to the network correspond to the attributes measured for each training tuple. The inputs are fed simultaneously into the units making up the input layer. These inputs pass through the input layer and are then weighted and fed simultaneously to a second layer of " neuronlike" units, known as a hidden layer. The outputs of the hidden layer units can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only one is used. The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given tuples. * The units in the input layer are called input units. The units in the hidden layers and output layer are sometimes referred to as neurodes, due to their symbolic biological basis, or as output units. * The network is feed-forward in that none of the weights cycles back to an input unit or to an output unit of a previous layer.

It is fully connected in that each unit provides input to each unit in the next forward layer. * The Result Generated After Applying Multilayer Perceptron On White-wine Quality Dataset Time taken to build model: 36. 22 seconds| Stratified cross-validation| * Summary| Correctly Classified Instances | 2598 | 55. 5128 %| Incorrectly Classified Instances | 2082 | 44. 4872 %| Kappa

statistic | 0. 2946| | Mean absolute error | 0. 1581| | Root mean squared error | 0. 2887| |

Relative absolute error | 81. 9951 %| | Root relative squared error | 93. 0018 %| | Total Number of Instances | 4680 | | * Detailed Accuracy By Class | | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area | PRC Area | Class| | 0 | 0 | 0 | 0 | 0 | 0. 344 | 0. 002 | 3| | 0. 056 | 0. 004 | 0. 308 | 0. 056 | 0. 095 | 0. 732 | 0. 156 | 4| | 0. 594 | 0. 165 | 0. 597 | 0. 594 | 0. 595 | 0. 98 | 0. 584 | 5| | 0. 704 | 0. 482 | 0. 545 | 0. 704 | 0. 614 | 0. 647 | 0. 568 | 6| | 0. 326 | 0. 07 | 0. 517 | 0. 326 | 0. 4 | 0. 808 | 0. 474 | 7| | 0. 058 | 0. 002 | 0. 5 | 0. 058 | 0. 105 | 0. 8 | 0. 169 | 8| | 0 | 0 | 0| 0 | 0 | 0. 356 | 0. 001 | 9| Weighted Avg. | 0. 555 | 0. 279 | 0. 544 | 0. 555 | 0. 532 | 0. 728 | 0. 526| | * Confusion Matrix |

A | B | C | D | E | F | G | | Class| 0 | 0 | 5 | 7 | 1 | 0 | 0 | | | A= 3| 0 | 8 | 82 | 50 | 2 | 0 | 0 | | | B= 4| 0 | 11 | 812 | 532 | 12 | 1 | 0 | | | C= 5| 0 | 6 | 425 | 1483 | 188 | 6 | 0 | | | D= 6| 0 | 1 | 33 | 551 | 285 | 3 | 0 | | | E= 7| 0 | 0 | 3 | 98 | 60 | 10 | 0 | | | F= 8| 0 | 0 | 0 | 2 | 3 | 0 | 0 | | | G= 9| * Performance Of The Multilayer perceptron With Respect To A Testing Configuration For The White-wine Quality Dataset

Testing Method| Training Set| Testing Set| 10-Fold Cross Validation| 66% Split| Correctly Classified Instances| 58. 1838 %| 50 %| 55. 5128 %| 51. 3514 %| Kappa statistic| 0. 3701| 0. 3671| 0. 2946| 0. 2454| Mean Absolute Error| 0. 1529| 0. 1746| 0. 1581| 0. 1628| Root Mean Squared Error| 0. 2808| 0. 3256| 0. 2887| 02972| Relative Absolute Error| 79. 2713 %| | 81. 9951 %| 84. 1402 %| * The Result Generated After Applying Multilayer Perceptron On Red-wine Quality Dataset Time taken to build model: 9. 14 seconds|

Stratified cross-validation (10-Fold)| * Summary | Correctly Classified Instances | 880 | 61. 111 %| Incorrectly Classified Instances | 546 | 38. 2889 %| Kappa statistic | 0. 3784| | Mean absolute error | 0. 1576| | Root mean squared error | 0. 3023| | Relative absolute error | 73. 6593 %| | Root relative squared error | 92. 4895 %| | Total Number of Instances | 1426| | * Detailed Accuracy By Class | | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area | Class| | 0 | 0 | 0 | 0 | 0 | 0. 47 | 3| | 0. 42 | 0. 005 | 0. 222 | 0. 042 | 0. 070 | 0. 735 | 4| | 0. 723 | 0. 249 | 0. 680 | 0. 723 | 0. 701 | 0. 801 | 5| | 0. 640 | 0. 322 | 0. 575 | 0. 640 | 0. 605 | 0. 692 | 6| | 0. 415 | 0. 049 | 0. 545 | 0. 415 | 0. 471 | 0. 831 | 7| | 0 | 0 | 0 | 0 | 0 | 0. 853 | 8| Weighted Avg. | 0. 617 | 0. 242 | 0. 595 | 0. 617 | 0. 602 | 0. 758| | * Confusion Matrix | A | B | C | D | E | F | | Class| | 0 | 5 | 1 | 0 | 0| || A= 3| 0 | 2 | 34 | 11 | 1 | 0 | | | B= 4| 0 | 2 | 436 | 160 | 5 | 0 | | | C= 5| 0 | 5 | 156 | 369 | 47 | 0 | | | D= 6| 0 | 0 | 10 | 93 | 73 | 0 | | | E= 7| 0 | 0 | 0 | 8 | 8 | 0 | | | F= 8| * Performance Of The Multilayer perceptron With Respect To A Testing Configuration For The Red-wine Quality Dataset Testing Method| Training Set| Testing Set| 10-Fold Cross Validation| 66% Split| Correctly Classified Instances| 68. 7237 %| 70 %| 61. 7111 %| 58. 7629 %| Kappa statistic| 0. 4895| 0. 5588| 0. 3784| 0. 327| Mean Absolute Error| 0. 426| 0. 1232| 0. 1576| 0. 1647| Root Mean Squared Error| 0. 2715| 0. 2424| 0. 3023| 0. 3029| Relative Absolute Error| 66. 6774 %| 51. 4904 %| 73. 6593 %| 77. 2484 %| * Result * The classification experiment is measured by accuracy percentage of classifying the instances correctly into its class according to quality attributes ranges between 0 (very bad) and 10 (excellent). * From the experiments, we found that classification for red wine quality using Kstar algorithm achieved 71. 0379 % accuracy while J48

classifier achieved about 60. 7994% and Multilayer Perceptron classifier achieved 61. 7111% accuracy. For the white wine, Kstar algorithm yielded 70. 6624 % accuracy while J48 classifier yielded 58. 547% accuracy and Multilayer Perceptron classifier achieved 55. 5128 % accuracy. * Results from the experiments lead us to conclude that Kstar performs better in classification task as compared against the J48 and Multilayer Perceptron classifier. The processing time for Kstar algorithm is also observed to be more efficient and less time consuming despite the large size of wine properties dataset. 7. COMPARISON OF DIFFERENT ALGORITHM * The Comparison Of All Three Algorithm On White-wine Quality Dataset (Using 10-Fold Cross Validation) Kstar| J48| Multilayer Perceptron| Time (Sec)| 0| 1. 08| 35. 14| Kappa Statistics| 0. 5365| 0. 3813| 0. 29| Correctly Classified Instances (%)| 70. 6624| 58. 547| 55. 128| True Positive Rate (Avg)| 0. 707| 0. 585| 0. 555| False Positive Rate (Avg)| 0. 2| 0. 21| 0. 279| * Chart Shows The Best Suited Algorithm For Our Dataset (Measures Vs Algorithms) * In above chart, comparison of True Positive rate and kappa statistics is given against three algorithm Kstar, J48, Multilayer Perceptron * Chart describes algorithm which is best suits for our dataset. In above chart column of TP rate & Kappa statistics of Kstar algorithm is higher than other two algorithms. * In above chart you can see that the False Positive Rate and the Mean Absolute Error of the Multilayer Perceptron algorithm is high compare to other two algorithms. So it is not good for our dataset. * But for the Kstar algorithm these two values are less, so the algorithm having lowest values for FP Rate & Mean Absolute Error rate is best suited algorithm. * So the final we can make conclusion that the Kstar algorithm is best suited algorithm for

White-wine Quality dataset. The Comparison Of All Three Algorithm On Red-wine Quality Dataset (Using 10-Fold Cross Validation) | Kstar| J48| Multilayer Perceptron| Time (Sec)| 0| 0. 24| 9. 3| Kappa Statistics| 0. 5294| 0. 3881| 0. 3784| Correctly Classified Instances (%)| 71. 0379| 60. 6994| 61. 7111| True Positive Rate (Avg)| 0. 71| 0. 608| 0. 617| False Positive Rate (Avg)| 0. 184| 0. 214| 0. 242| * For Red-wine Quality dataset have also Kstar is best suited algorithm , because of TP rate & Kappa statistics of Kstar algorithm is higher than other two algorithms and FP rate & Mean Absolute Error of Kstar algorithm is lower than other algorithms. . APPLYING TESTING DATASET Step1: Load pre-processed dataset. Step2: Go to classify tab. Click on choose button and select lazy folder from the hierarchy tab and then select kstar algorithm. After selecting the kstar algorithm keep the value of cross validation = 10, then build the model by clicking on start button. Step3: Now take any 10 or 15 records from your dataset, make their class value unknown(by putting '? ' in the cell of the corresponding raw) as shown below. Step 4: Save this data set as . rff file. Step 5: From " test option" panel select " supplied test set", click on to the set button and open the test dataset file which was lastly created by you from the disk. Step 6: From " Result list panel" panel select Kstar-algorithm (because it is better than any other for this dataset), right click it and click " Re-evaluate model on current test set" Step 7: Again right click on Kstar algorithm and select " visualize classifier error" Step 8: Click on save button and then save your test model.

Step 9: After you had saved your test model, a separate file is created in which you will be having your predicted values for your testing dataset. Step 10: Now, this test model will have all the class value generated by model by

re-evaluating model on the test data for all the instances that were set to unknown, as shown in the figure below. 9. ACHIEVEMENT * Classification models may be used as part of decision support system in different stages of wine production, hence giving the opportunity for manufacturer to make corrective and additive measure that will result in higher quality wine being produced. From the resulting classification accuracy, we found that accuracy rate for the white wine is influenced by a higher number of physicochemistry attribute, which are alcohol, density, free sulfur dioxide, chlorides, citric acid, and volatile acidity. * Red wine quality is highly correlated to only four attributes, which are alcohol, sulphates, total sulfur dioxide, and volatile acidity. * This shows white wine quality is affected by physicochemistry attributes that does not affect the red wine in general. Therefore, I suggest that white wine manufacturer should conduct wider range of test particularly towards density and chloride content since white wine quality is affected by such substances. * Attribute selection algorithm we conducted also ranked alcohol as the highest in both datasets, hence the alcohol level is the main attribute that determines the quality in both red and white wine. * My suggestion is that wine manufacturer to focus in maintaining a suitable alcohol content, may be by longer fermentation period or higher yield fermenting yeast.