Vivek Kaushal | CSE 4334/5334 Data Mining | October 31, 2015

Project 1 Report.

Classification techniques

# Problem Description

In this project we were asked to work on 3 different data sets and predict the class of the test examples based on the data we have. To solve this problem, we use Classification Techniques of Data Mining.

Classification is a technique which based on the data given to the model tries to predict the correct class of the test data using simple measures like distance from a given vertex, mean of the given data etc. as the similarity measures.

Each classification algorithm has these things in common:

1. Based on the data we give to the algorithm it builds a classifier model which is to be used to predict the outcomes.
2. Then based on the similarity measures like distance etc. it determines to which class the given Test Data belongs and classifies it accordingly.

# Classification Techniques Used

In this project I implemented 2 classification techniques which are

1. K-Nearest Neighbor
2. Support Vector Machines.

## K-Nearest neighbor

The K-Nearest Neighbor algorithm is the most basic algorithm of all classification techniques.

Steps of the algorithm:

1. First of all, we process our Dataset and split it into Training Data and Testing data.

We use the Test Set to conduct experiments on the Training Set and predict the outputs.

1. Then after splitting the dataset we compute the distance between the given Test Instance with each tuple of the training set and store all the distance values.

To compute the distance, we may use any of the following methods:

* Manhattan Distance
* Euclidean Distance
* Minkowski Distance

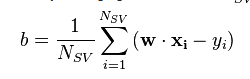
1. Then we use our determining parameter i.e. K to determine the nearest neighbors. We take K number of closest distances from our table and see their Classifying class.

* If the value of K is odd, then we go for MajorityVoting and decide the class of test set.
* If the value of K is even, we compute the weighted average of the distances and compute the similarity coefficient. The class with the lowest similarity coefficient is selected as our predicted class.

## support vector machines

A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

A parameter b is computed using the following formula to determine the center of this distribution of our support vectors:



For solving our problem, I implemented the K-NN algorithm using python 2.7 and used Weka GUI for implementing SVM.

# Datasets used:

## ATT Face data:

This dataset contains face data of 40 At&T customers. Each column of the data describes each pixel of the face. It is numeric dataset.

Dimensions: 645\*401

## IRIS Data:

It contains the details of petals of flowers based on which its IRIS is classified. It has 3 different classification classes. It has both numeric and nominal attributes.

Dimensions: 5\*151

## Blood transfusion data:

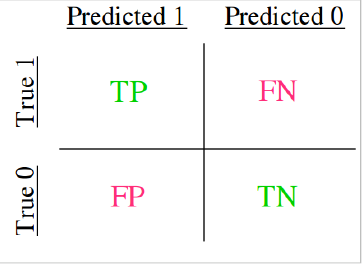
To demonstrate the RFMTC marketing model (a modified version of RFM), this study adopted the donor database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan. The center passes their blood transfusion service bus to one university in Hsin-Chu City to gather blood donated about every three months. To build a FRMTC model, we selected 748 donors at random from the donor database. These 748 donor data, each one included R (Recency - months since last donation), F (Frequency - total number of donation), M (Monetary - total blood donated in c.c.), T (Time - months since first donation), and a binary variable representing whether he/she donated blood in March 2007 (1 stand for donating blood; 0 stands for not donating blood).

Dimension: 5\*749

Source: [https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center](https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center%20)

# Evaluation of Result:

Evaluation of a result is done by creating a confusion matrix. Based on the values of confusion matrix following parameters are computed:



Confusion Matrix

1. Accuracy:

Accuracy is the total number of correctly identified instances over total number of instances tested i.e.

**Accuracy =**

In a multi-class problem, we consider the sum of the diagonals by the total instances to compute accuracy.

1. K-Fold Cross Validation:

It involved partitioning the given Dataset ‘d’ in K number of partitions in which K-1 partitions are used to train the model and 1 partition is used to test it.

We make K iterations to compute the sum of all the errors i.e. number of wrongly classified instances and the standard deviation of the distribution.

Following are the formulas:

**Sum of Errors**

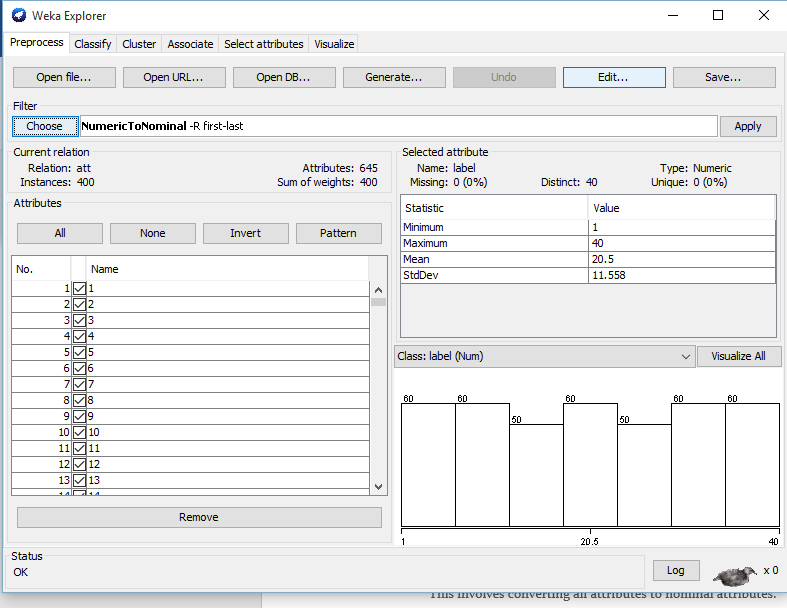
Where, t is the total number of iterations, n is the error in each prediction, m is the total number of predictions in each instance.

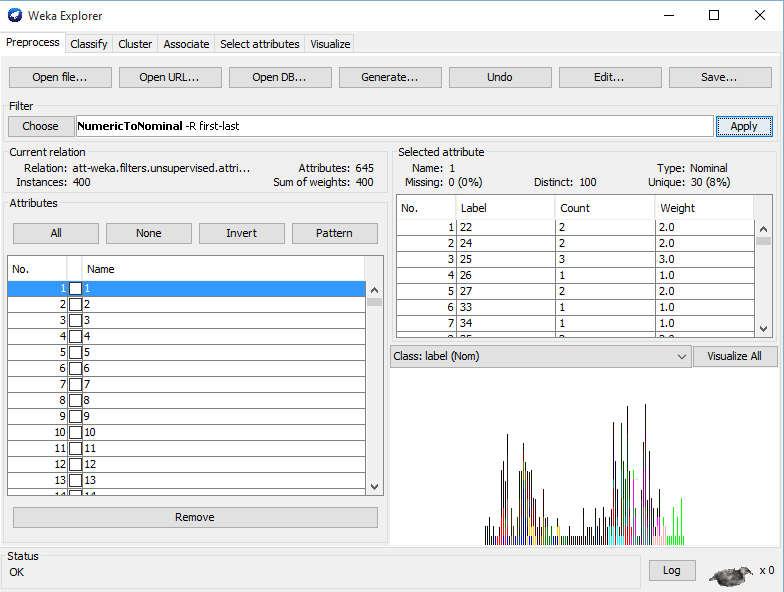
**Standard Deviation**

# Support Vector Machine Weka Implementation:

## Pre-process data:

This involves converting all attributes to nominal attributes.

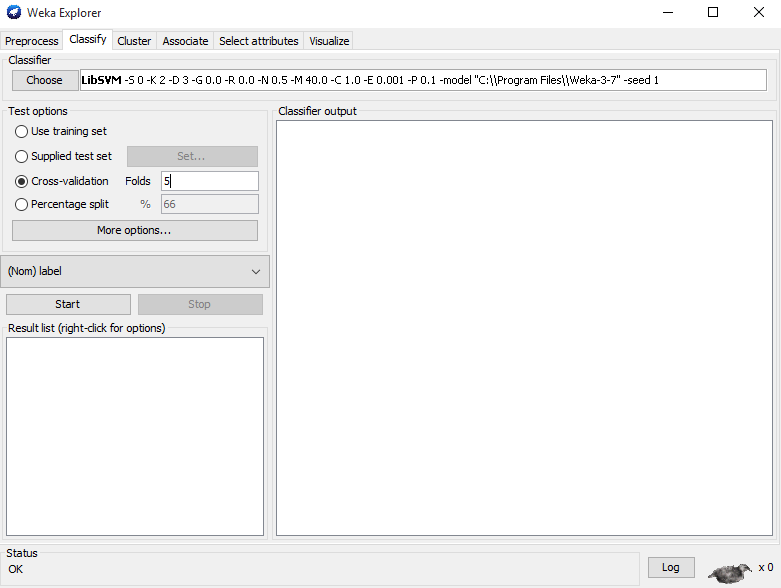


After pre-processing is done it looks like:

## Classify:

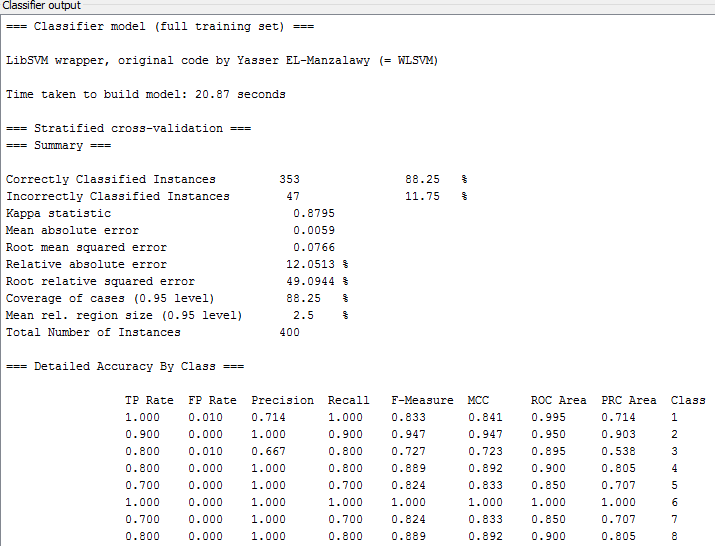
Then we go to Classify tab and select our class based on which we need to predict the output and select SVM as our classifier.

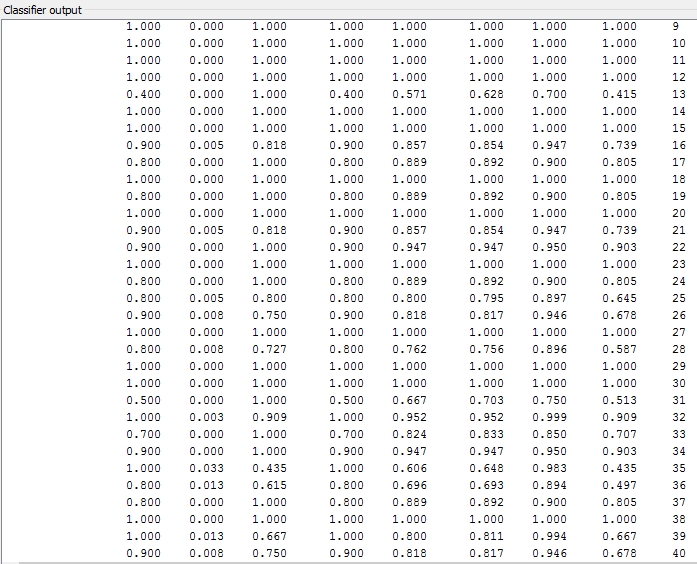
Here we have specified Cross Validation Folds = 5 and the data set being used is ATT.csv.



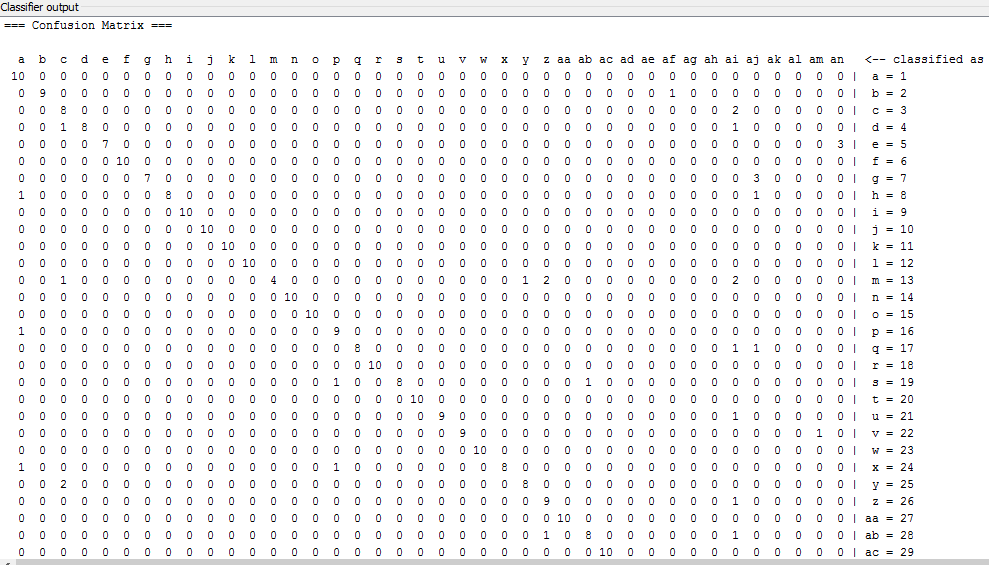
## Weka output and metrics for ATT.CSV:

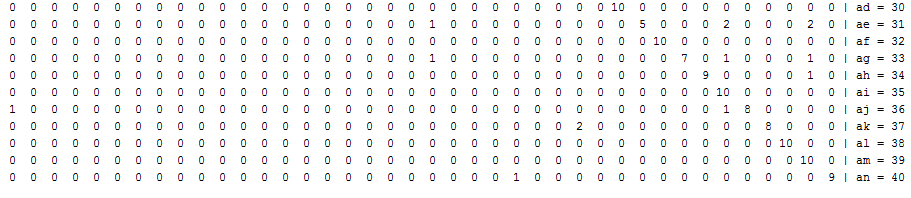
**When K = 5:**



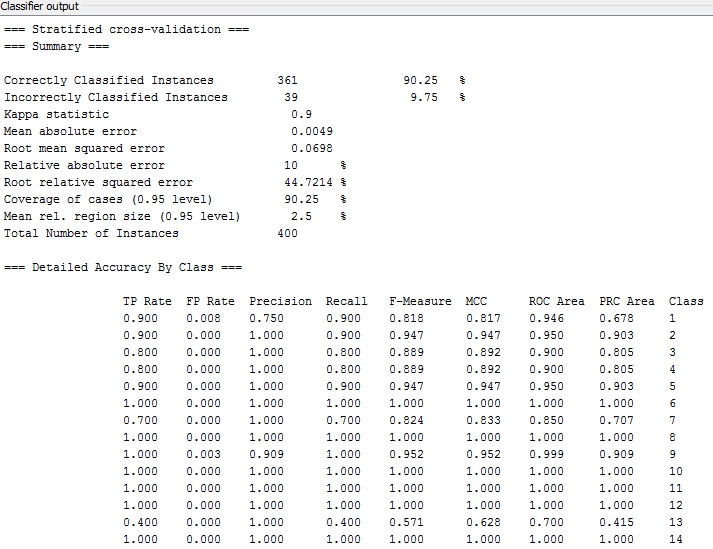


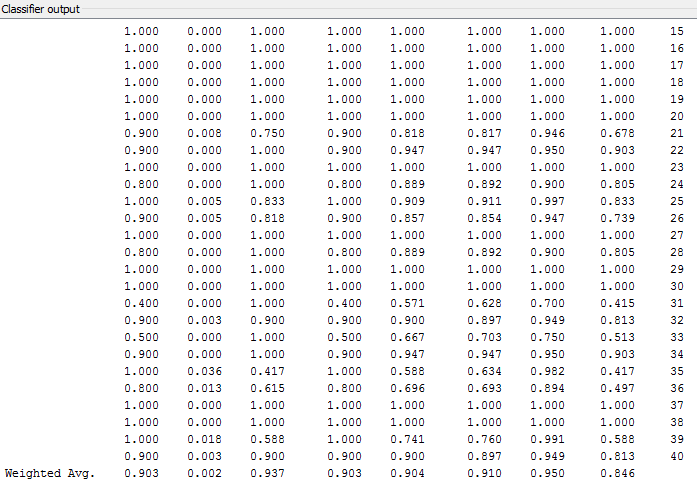
**Confusion Matrix:**



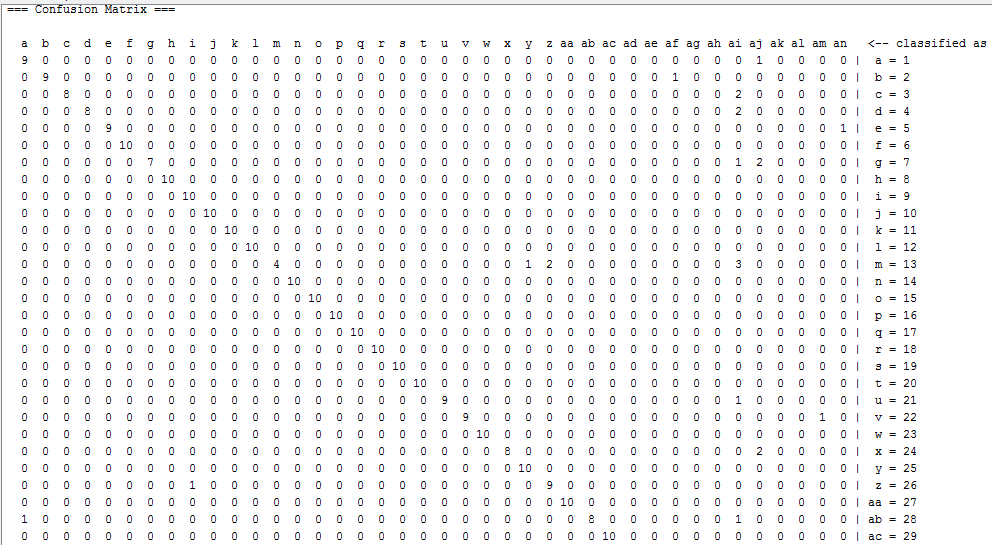


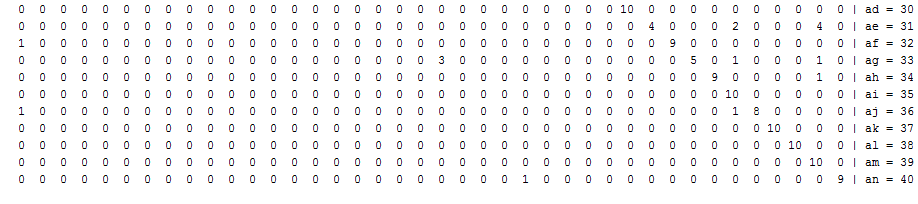
**For k = 10:**





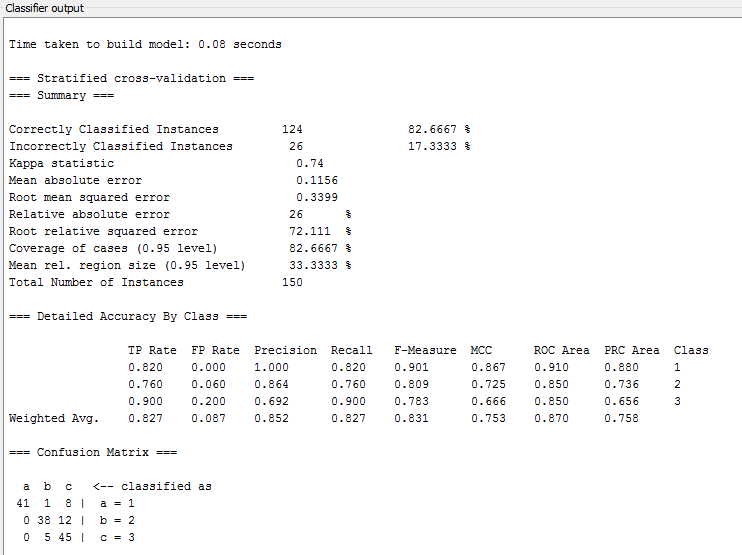
**Confusion Matrix:**



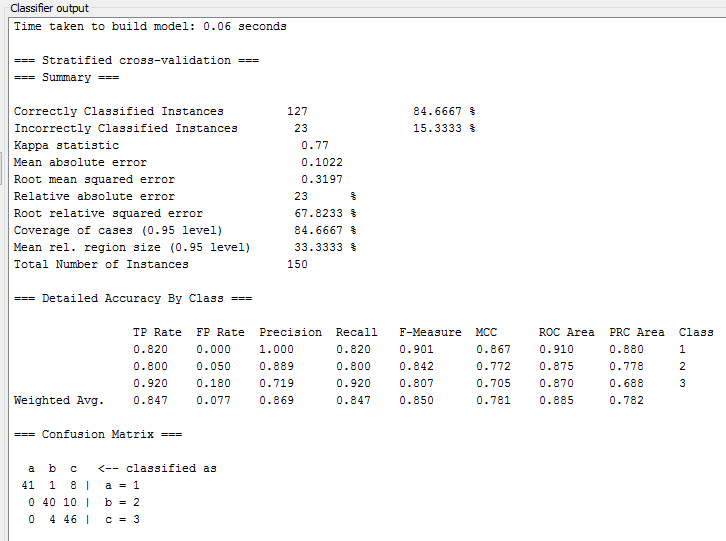


## Weka output and metrics for iris.csv

**For k = 10:**

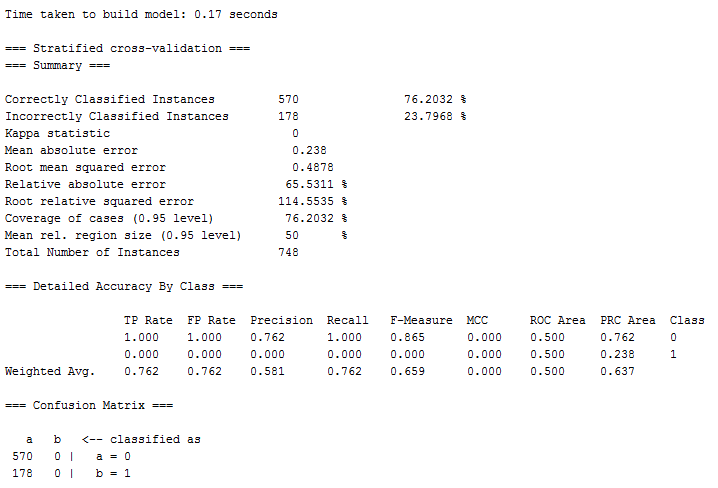


**For k=5:**

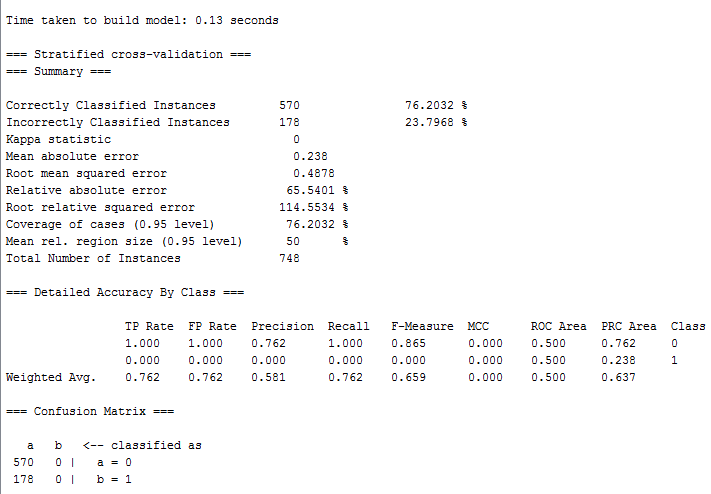


## Weka output and metrics for Transfusion.csv:

**For k = 5:**



**For k = 10:**



# References:

1. <https://en.wikipedia.org/wiki/Support_vector_machine>
2. <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>
3. <http://scikit-learn.sourceforge.net/dev/modules/cross_validation.html>
4. <http://scipy.org/>
5. Lecture notes by Prof. Shuai Zheng