**Title:** Missing Value and Normalization

Gautham M (15CO118)

**Introduction:**

Many existing, industrial and research data sets contain Missing Values. They are introduced due Due to various reasons, such as manual data entry procedures, equipment errors and incorrect measurements. Hence, it is usual to find missing data in most of the information sources used. The detection of incomplete data is easy in most cases, looking for Null values in a data set. However, this is not always true, since Missing Values (MVs) can appear with the form of outliers or even wrong data (i.e. out of boundaries).

Normalization is scaling technique or a mapping technique where we can find new range from existing one range. There are many ways to predict or forecast but all of them vary a lot so to maintain the large variation of prediction and forecasting the normalization technique is required to make them closer. One of the first steps of pre processing the normalization of the data. This step is very important when dealing with parameters of different units and scales. For example, some data mining techniques use the Euclidean distance. Therefore, all parameters should have the same scale for a fair comparison between them.

**Handling missing values**

#### 1. Ignore the data row

This is usually done when the class label is missing (assuming your data mining goal is classification), or many attributes are missing from the row (not just one). However, you’ll obviously get poor performance if the percentage of such rows is high.

For example, let’s say we have a database of students enrolment data (age, SAT score, state of residence, etc.) and a column classifying their success in college to “Low”, “Medium” and “High”. Let’s say our goal is to build a model predicting a student’s success in college. Data rows who are missing the success column are not useful in predicting success so they could very well be ignored and removed before running the algorithm.

### Pros:

* Complete removal of data with missing values results in robust and highly accurate model
* Deleting a particular row or a column with no specific information is better, since it does not have a high weightage

### Cons:

* Loss of information and data
* Works poorly if the percentage of missing values is high (say 30%), compared to the whole dataset

#### 2. Use a global constant to fill in for missing values

Decide on a new global constant value, like “unknown“, “N/A” or minus infinity, that will be used to fill all the missing values.  
This technique is used because sometimes it just doesn’t make sense to try and predict the missing value.

For example, let’s look at the students enrollment database again. Assuming the state of residence attribute data is missing for some students. Filling it up with some state doesn’t really makes sense as opposed to using something like “N/A”.

### Pros:

* Less possibilities with one extra category, resulting in low variance after one hot encoding — since it is categorical
* Negates the loss of data by adding an unique category

### Cons:

* Adds less variance
* Adds another feature to the model while encoding, which may result in poor performance

#### 3. Use attribute mean,median or mode

Replace missing values of an attribute with the mean (or median if its discrete) value for that attribute in the database.

For example, in a database of US family incomes, if the average income of a US family is X you can use that value to replace missing income values.

### Pros:

* This is a better approach when the data size is small
* It can prevent data loss which results in removal of the rows and columns

### Cons:

* Imputing the approximations add variance and bias
* Works poorly compared to other multiple-imputations method

#### 4. Use attribute mean for all samples belonging to the same class

Instead of using the mean (or median) of a certain attribute calculated by looking at all the rows in a database, we can limit the calculations to the relevant class to make the value more relevant to the row we’re looking at.

Let’s say you have a cars pricing database that, among other things, classifies cars to “Luxury” and “Low budget” and you’re dealing with missing values in the cost field. Replacing missing cost of a luxury car with the average cost of all luxury cars is probably more accurate than the value you’d get if you factor in the low budget cars.

**Normalization techniques**

**Min-Max Normalization:**

Min-max normalization performs a linear alteration on the original data. The values are normalized within the given range. For mapping a v value, of an attribute A from range [minA,maxA] to a new range [new\_minA,new\_maxA], the computation is given by,

where v’ is the new value in the required range The benefit of Min-Max normalization is that all the values are annealed within certain range.

**Z-Score Normalization:**

Z score normalization, also called as Zero mean normalization. Here the data is normalized based on the mean and standard deviation. Then the formula is,

Where Mean(p) = sum of the all attribute values of P

Std(P)=Standard deviation of all values of P

**Decimal Scale Normalization:**

Decimal Scale Normalization based on the movement of decimal point of value of attribute. The decimal point numbers are moved depends on the maximum absolute values of attribute. The decimal Scale normalization formula is,

where, m is the smallest integer that max (| d′|)<1.

**Background Work.**

There have been many contributions to new techniques for data normalization and handling of missing data .

In [1], authors give a comparison of several approaches for solving the missingvalue problem, including deletion and single imputation techniques. Each method is briefly described, together with the advantage and disadvantage of the solution it provides. Simulation results are presented to compare the induced errors between predicted (imputed) valued and true values obtained by using different methods to fill in slots with missing values.

In paper [2] the authors , have compared two imputation methods for dealing with the missing data, namely k-NN imputation method and mean and median imputation method. As a result, they have found that both of the imputation methods are efficient and yield more or less the same accuracy.

In [3] the authors developed a novel method to estimate the values of missing data by the use of a weighted K-nearest neighbors algorithm. A weighting scheme that exploits the correlation between a ldquomissingrdquo dimension and available data values from other fields, which is quantified based on the support vector regression method.

In [4] missing value are handled using Dynamic Bayesian Network (DBN). DBN is a useful technique to maintain the relationships between attributes of data. The results of the prediction are used to fill in the missing values in the data. Support Vector Regression (SVR) algorithm is used for predicting the missing values.

In [5] ,authors explain and describe several previous studies about missing values handling methods or approach on time series data. The paper also discuss some plausible option of methods to estimate missing values to be used by other researchers in this field of study. The discussion's aim is to help them to figure out what method is commonly used now along with its advantages and drawbacks.

In [6] , authors propose a new multiple imputation approach based on sampling techniques to handle missing values problems, in order to improving the quality and efficiency of data mining process. The proposed method is favourably compared

with some imputation techniques and outperforms the existing approaches using an experimental benchmark on a large scale, waveform dataset taken from machine learning repository and different rate of missing values

In the paper [7], authors have analyzed the use normalization techniques in achieving privacy. They have compared the results of these techniques and from the experimental outcome it can be concluded that Min-Max normalization have minimum misclassification error.

In the paper[8] authors have performed different normalization techniques on different datasets. These techniques help in obtaining high training accuracy for classification. The classification is performed on these datasets using SVM.

**Algorithm:**

**Input:** *D*, a database of transactions;

entity containing missing values

case 1:(Ignore)

for each record if value of entity is missing

drop the record.

Case 2:( global constant)

Input: constant

for each record if value of entity is missing

replace value with constant.

Case 3:(replace with mean median or mode)

calculate global mean ,median ,mode

for each record if value of entity is missing

replace value with calculated mean median mode.

Case 4:(class wise mean/median/mode)

using classifier algo calculate mean/median/mode

for each record if value of entity is missing

replace value with calculated mean median mode.

**Output:**  modified database without missing values

*Min Max Normalization*

*Input: D, a database of transactions;*

*New Min ,New Max*

*Calculate current\_MIN, current\_MAX*

For Each record attribute

replace value with value calculated using above formula.

Z score Normalization

*Input: D, a database of transactions;*

*Calculate Mean of all values of attribute*

*Calculate Standard Deviation of the records*

For Each record attribute

replace value with value calculated using above formula.

Decimal Normalization

*Input: D, a database of transactions;*

*Calculate m*

For Each record attribute

replace value with value calculated using above formula

**Example Input:**

|  |  |
| --- | --- |
| Person\_ID | Income |
| 101 | 28000 |
| 102 | 32000 |
| 103 | 34000 |
| 104 | 36000 |
| 105 |  |
| 106 | 38000 |
| 107 |  |
| 108 | 42000 |
| 109 | 50000 |

**Step by step implementation with data:**

*Case 1: Ignore*

|  |  |
| --- | --- |
| Person\_ID | Income |
| 101 | 28000 |
| 102 | 32000 |
| 103 | 34000 |
| 104 | 36000 |
| 106 | 38000 |
| 108 | 42000 |
| 109 | 50000 |

*Case 2:Global constant*

*let the constant be 40000*

|  |  |
| --- | --- |
| Person\_ID | Income |
| 101 | 28000 |
| 102 | 32000 |
| 103 | 34000 |
| 104 | 36000 |
| 105 | 40000 |
| 106 | 38000 |
| 107 | 40000 |
| 108 | 42000 |
| 109 | 50000 |

*Case 3:Replace by mean*

*calculate Mean:(28000+32000+34000+36000+38000+42000+50000)/7*

*=26000*

|  |  |
| --- | --- |
| Person\_ID | Income |
| 101 | 28000 |
| 102 | 32000 |
| 103 | 34000 |
| 104 | 36000 |
| 105 | 26000 |
| 106 | 38000 |
| 107 | 26000 |
| 108 | 42000 |
| 109 | 50000 |

*Case 4:Replace by Class mean*

let even numbers be class A and odd numbers be class B

Class A Mean=(32000+36000+38000+42000)/4

=37000

Class B Mean=(28000+34000+50000)/3

=37333

|  |  |
| --- | --- |
| Person\_ID | Income |
| 101 | 28000 |
| 102 | 32000 |
| 103 | 34000 |
| 104 | 36000 |
| 105 | 37333 |
| 106 | 38000 |
| 107 | 37333 |
| 108 | 42000 |
| 109 | 50000 |

Normalization

min max normalization

min=0 max =1

|  |  |
| --- | --- |
| Person\_ID | Income |
| 101 | 0 |
| 102 | 0.181818182 |
| 103 | 0.272727273 |
| 104 | 0.363636364 |
| 105 | 0.424227273 |
| 106 | 0.454545455 |
| 107 | 0.424227273 |
| 108 | 0.636363636 |
| 109 | 1 |

Z Score Normalization

Standard Deviation=6234.11

mean=37185.11

|  |  |
| --- | --- |
| Person\_ID | Income |
| 101 | -1.4333 |
| 102 | -0.8317 |
| 103 | −0.5109 |
| 104 | −0.1901 |
| 105 | 0.0237 |
| 106 | 0.1307 |
| 107 | 0.0237 |
| 108 | 0.7723 |
| 109 | 2.0556 |

Decimal Scale Normalization:

m calculated=5 since 50000/100000=0.5<1

|  |  |
| --- | --- |
| Person\_ID | Income |
| 101 | 0.28000 |
| 102 | 0.32000 |
| 103 | 0.34000 |
| 104 | 0.36000 |
| 105 | 0.26000 |
| 106 | 0.38000 |
| 107 | 0.26000 |
| 108 | 0.42000 |
| 109 | 0.50000 |

**Data set Description:**

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes.

Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

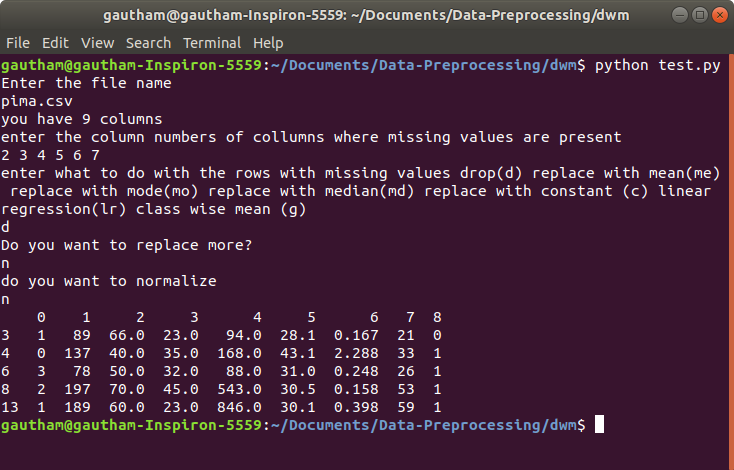
* Pregnancies: Number of times pregnant
* Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
* BloodPressure: Diastolic blood pressure (mm Hg)
* SkinThickness: Triceps skin fold thickness (mm)
* Insulin: 2-Hour serum insulin (mu U/ml)
* BMI: Body mass index (weight in kg/(height in m)^2)
* DiabetesPedigreeFunction: Diabetes pedigree function
* Age: Age (years)
* Outcome: Class variable (0 or 1)

**Implementation Tools:**

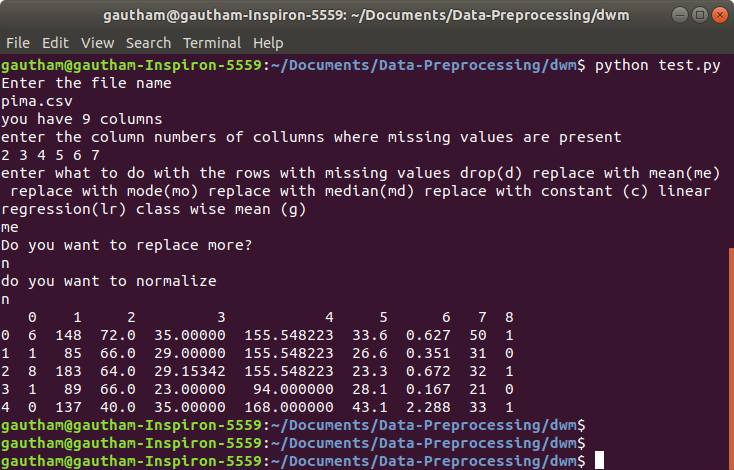
Implementation using python

**Screenshots :**

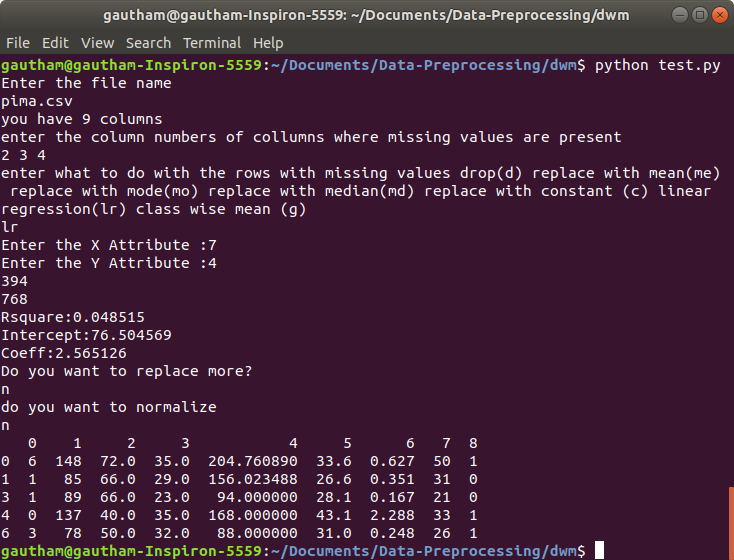
Drop rows with missing data



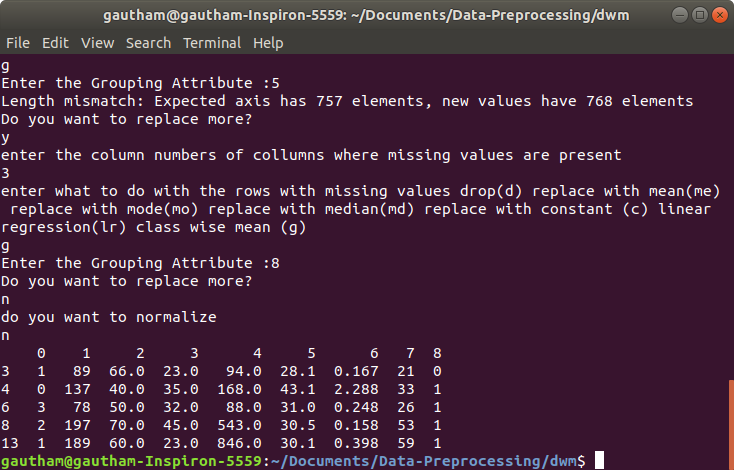
Replace value with global mean



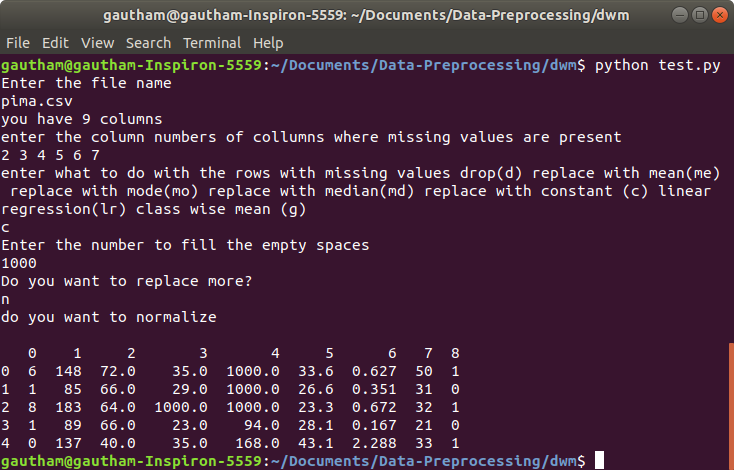
Replace value using Linear Regression



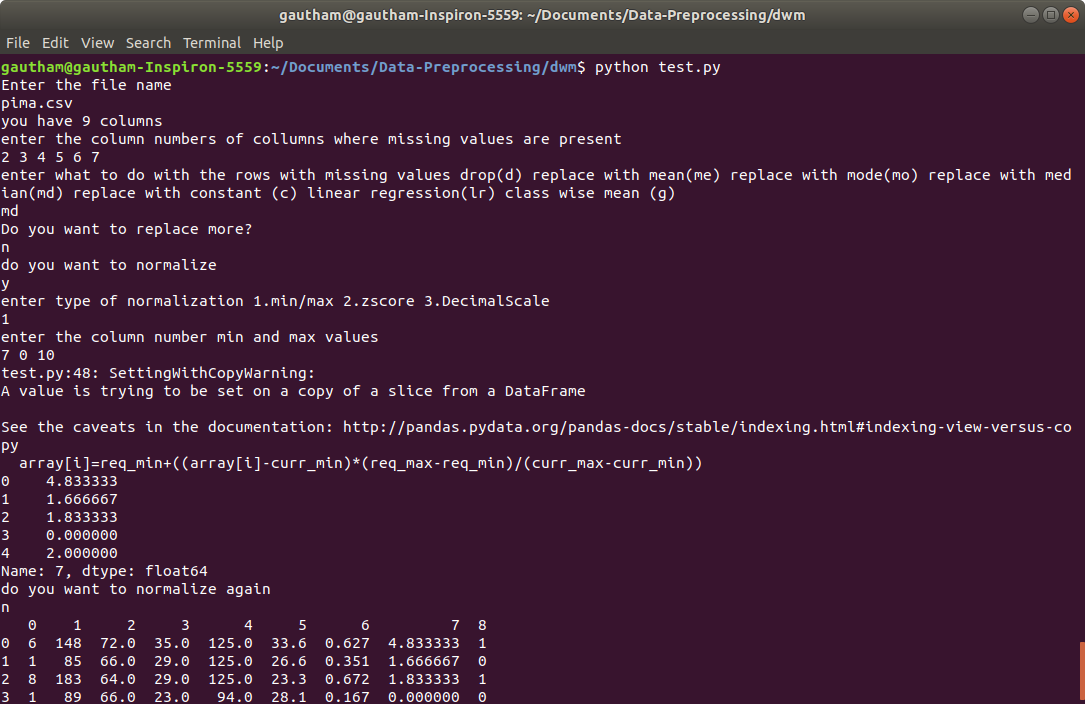
Replace values with class wise mean



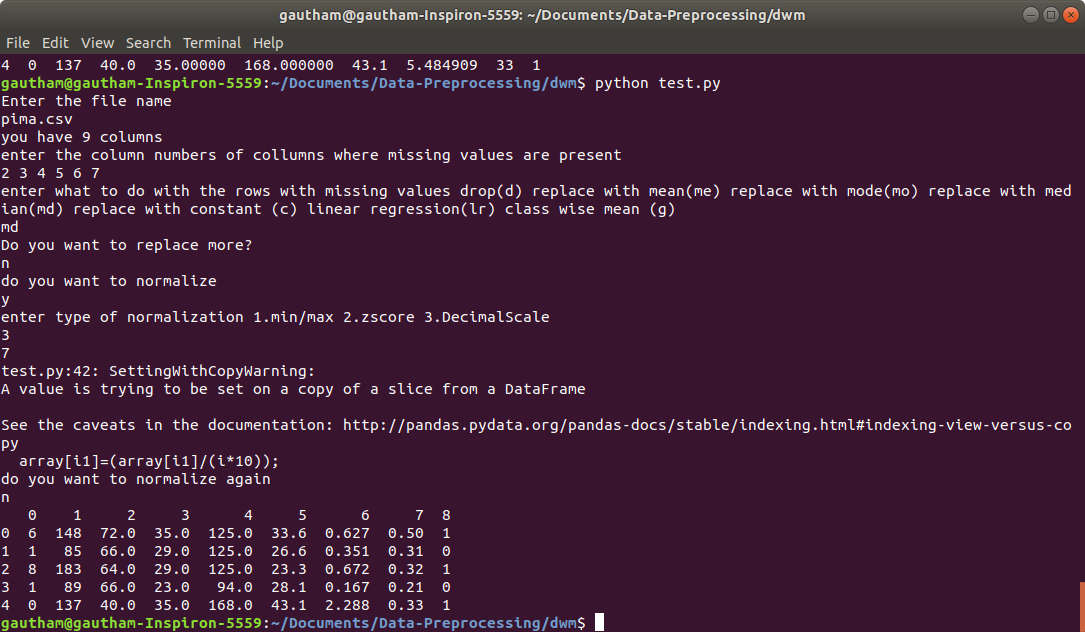
Replace value with constant



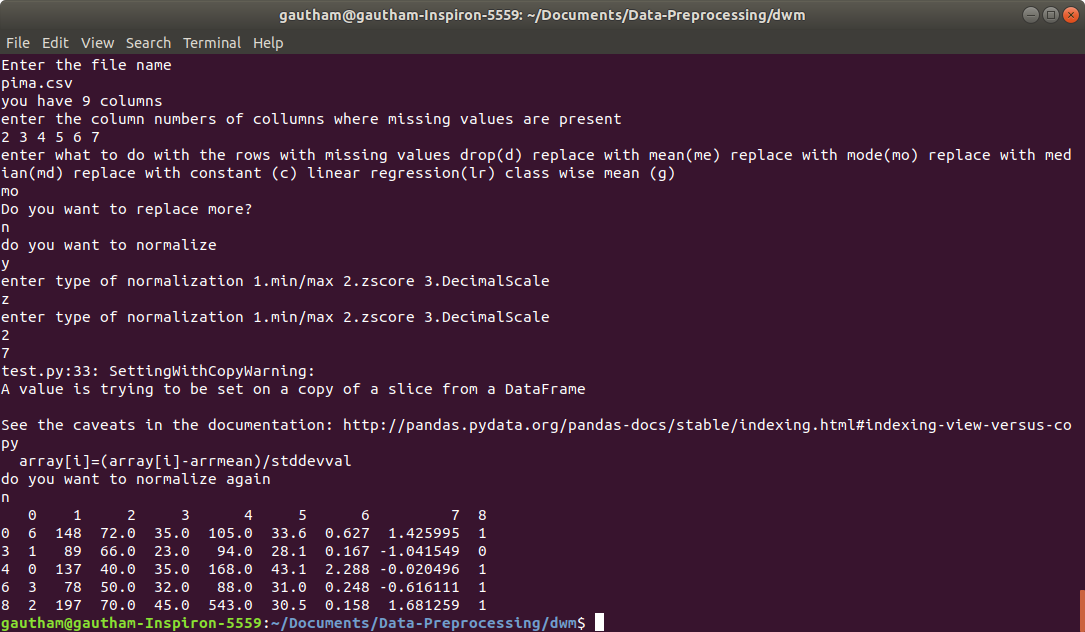
Normalize with min max normalization



Normalize with Decimal Normalization



Normalize with Z-Score Normalization



**References:**

1.A comparison of approaches for dealing with missing values Chih-Feng Liu. ThaoTsen Chen ,Shie-Jue Lee

2.Techniques to deal with missing data Jadran Sessa ; Dabeeruddin Syed

3.Estimation of Missing Values Using a Weighted K-Nearest Neighbors Algorithm Wang Ling ; Fu Dong-Mei

4.Imputation of missing value using dynamic Bayesian network for multivariate time series data Steffi Pauli Susanti ; Fazat Nur Azizah

5.A review of missing values handling methods on time-series data Irfan Pratama Adhistya Erna Permanasari Igi Ardiyanto Rini Indrayani

6.Handling Missing Data Problems with Sampling Methods Rima Houari ; Ahcène Bounceur ; A Kamel Tari ; M Tahar Kecha

7.A Study on Normalization Techniques for Privacy Preserving Data Mining A Study on Normalization Techniques C.Saranya, G.Manikandan

8.Classification using Different Normalization Techniques in Support Vector MachinePriti Sudhir Patki ,Vishakha V.Kelkar