4.1. Introduction

Data is a collection of data objects called entity and their attributes. Properties of a data object are represented by the corresponding data fields or attributes. Attributes are of following categories:

- **Numerical** – Set of values (continuous) eg. any real number like Age, Height, Weight etc.
- **Nominal** – A finite set of discrete values eg. Color, Days-of-Weak etc.
- **Binary** – It is nominal attribute type with only two values 0 and 1 eg. Marital-status {Married, Unmarried}, Sex {Male, Female} etc.
- **Ordinal** – It is also nominal attribute in which there is a meaningful order in the value. For example Size {Small, Medium, Large}, Employee-rank {A,B,C,D} etc.

Good quality of data should be needed to get good outcome as the presence of garbage in input data set may produce garbage output. In real world the data may be dirty due to following reason:

- **Presence of noise**
- **Some values are missing**
- **Data redundancy**
- **Inconsistent data**

Data are noisy due to outliers or erroneous data. For example Age = -12. Sometimes data may be incomplete as some values are missing due to several reasons. Data redundancy i.e. duplicate or irrelevant data should be removed to optimize time and...
space during analysis. Data are inconsistent when there are some discrepancies found in the data. As for example Age = 50 and Date-of birth = “05/08/2005”. For the above reasons data preprocessing is absolutely needed to prepare data of good quality.

4.2. Data Preprocessing

One of the important and critical steps of data mining process is data preprocessing and it has a great effect on the success of data mining project [1]. Data preprocessing is needed to analyze data more accurately and efficiently. Preprocessed data is in better condition for subsequent analysis. Major steps in data preprocessing are:

- Data cleaning
- Data integration
- Data transformation
- Data reduction

4.2.1. Data Cleaning

Major steps in data cleaning phase of data preprocessing involves:

- Handling missing values
- Handling noisy data
- Removing data redundancy
- Resolving data inconsistencies

Various techniques that can be applied to handle missing values have been discussed later in this chapter. Existence of outliers or random errors in a measured variable is the main reason for noisy data. Different methods to remove noise in the data are:

- Binning
- Clustering
- Regression
In binning data are sorted and placed into different bins of equal frequencies depending on the attribute values. In smoothing, data in a bin may be replaced by the mean value of the content of the bin or by the median value of the content of the bin. In another process of smoothing each data in a bin may be replaced by the closest boundary value (minimum or maximum value) of the bin.

Outliers can be identified by clustering easily, in which attribute values are separated into groups or clusters. Values, lying outside of these clusters are identified as outliers.

By applying different regression methods, function or a mathematical equation is generated to fit the data and also to predict the value of an attribute by knowing the value or values of other attribute or attributes. Thus regression helps in smoothing out the noise in the data.

Data redundancy can be removed by eliminating the duplicate records, which are also irrelevant and thus reduces the size of the data set. As a result the data set can be analyzed more efficiently.

Data inconsistencies can be removed either manually or by using tools which are designed to help in removing incorrect use of codes. Known data constraints can also be verified by using these tools.

### 4.2.2. Data Integration and Transformation

Data may be collected from different sources which may involve multiple databases and flat files. Entity identification problem is a real problem during data integration, which may be resolved by using meta-data (data about data). During this phase there may occur data redundancy and data inconsistency. By using different techniques such as correlation analysis and X2 (Chi-square) test data redundancy can be detected.

Sometimes data may be transformed to make it convenient for data analysis. Different methods that can be used to transform data may involve the following:

- **Generalization of data**: Data may be transformed to higher level concepts. For example numeric data height may be mapped to tall, medium, short etc.

- **Normalization of data**: In this case data are mapped within a small specified range. The range may be -1.0 to +1.0, 0.0 to 1.0 etc.
- Construction of attributes: Sometimes new attributes may help the analysis process which can be constructed from the given set of attributes in the data set.

4.2.3. Data Reduction

Size of data set stored in database or in data warehouse may be huge. Complex data analysis process may take long time to analyze such data set, which is undesirable. For this reason data reduction is needed to obtain reduced form of the data set in terms of size but produces almost same result during analysis. It is one of the primary steps of data pre-processing. The most common techniques which are used to obtain reduced data set are: numerosity reduction, dimensionality reduction and data compression.

4.2.3.1. Numerosity Reduction

In this technique actual data in the data set are replaced by alternative smaller forms. Data cube aggregation, sampling are examples of numerosity reduction.

Suppose a data set contains income per quarter for the year 2005 to 2017. This kind of data can be aggregated to represent the total income per year in place of per quarter. Thus the new data set will be smaller in size without losing any information. Multidimensional aggregated information are stored in data cubes.

Sampling is an important technique, which is applied for selection of data. The objective is to select a subset of data that represents the whole data set. Statistical sample is required as handling whole data set may be too expensive or time consuming. Different sampling methods are:

- Simple random sampling: In this method each item has an equal probability to be chosen. Performance of this kind of sampling may be poor in presence of skew.

- Sampling without replacement: In this method the object selected for sample has been removed from the original data set. For this reason there are no duplicate objects present in the sample.
• Sampling with replacement: In this method objects selected for sample are not removed from the original data set. This allows presence of duplicate objects in the sample.

• Adaptive sampling: In adaptive sampling selection criteria of sample has been adapted depending on previous result obtained during survey. As data collection continues the sampling process is modified in real time. Adaptive stratified or cluster sampling may be used for data reduction. Different adaptive sampling strategies are also proposed in literatures.

Figure 4.1 and Figure 4.2 show pictorial presentation of sampling processes.
4.2.3.2. Dimensionality Reduction

In recent years, a variety of techniques have been proposed in literatures for dimensionality reduction. Dimensionality of some real world data with high dimensionality such as ECG signals, digital image etc. should be reduced for convenience. Redundant and irrelevant features in the original data set should be removed to reduce the dimensionality of the data set. The objective is to select a feature subset from the original data set which is a meaningful representation with reduced dimensionality. Feature extraction and reduction is one of the important steps of classification since even the best classifier may perform poorly if the features are not well chosen [2], [3]. These reduced features may play vital roles during final decision making by the physicians. Many algorithms are available for dimensionality reduction. Methods which we have used in our study are:

- Correlation-based Feature Subset Selection (CFS) [4]
- Rough Set (RS) [5]
- Principal Component Analysis (PCA) [2]
- Genetic Algorithm (GA)

4.2.3.2.A. Correlation-based Feature Subset Selection (CFS)

“A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other” [4] – this is the central hypothesis of CFS.

A feature evaluation formula, based on ideas from test theory [6], provides an operational definition of the above hypothesis as follows:

\[
\tau_{fc} = \frac{k \overline{r_{fc}}}{\sqrt{k + k(k - 1) \overline{r_{ff}}}}
\]

(1)

where \( r_{fc} \) is the correlation between the summed features and the class variable, \( k \) is the number of features, \( \overline{r_{fc}} \) is the average of the correlation of between the features and the class variable, and \( \overline{r_{ff}} \) is the average inter-correlation between features.
To accommodate nominal or categorical as well as continuous or ordinal features in Eq. 1, continuous features are transformed to categorical features using the supervised discretisation method of [7] as a preprocessing step. The theory of information gain [8] is applied estimating the degree of associations between nominal features. Heuristic search strategies, such as best first and hill-climbing [9] are often used to search the feature subset in reasonable amount of time. Moreover, both filter type as well as wrapper types of feature selection methods use correlation-based approach in different applications [10], [11].

4.2.3.2.B. Rough Set Theory (RST)

Rough set theory was first presented by Pawlak in the year 1980’s [12]. Rough set is a formal approximation of a crisp set in terms of a pair of sets which give lower approximation with positive region and upper approximation with negative region. In between there is a boundary. Let there be an information system I = ( U,A ) (attribute – value system), where U be the universe of discourse and is a non-empty set of finite objects; A is a non-empty finite set of attributes. With any P ∈ A, there is an associated equivalence relation IND(P). The relation IND(P) is called P – indiscernibility relation. Let X ∈ U be a target set. The target set X can be approximated using only the information contained within P by constructing P-lower (P[sub]X[/sub]) and P-upper (P[sub]X[/sub]) approximation of X. The tuple (P[sub]X[/sub], P[sub]X[/sub]) is called a rough set. The accuracy of the rough-set representation of the set X can be given [13] by the following:

\[ k^P = \frac{|P_X|}{|P_X|} \]  

Rough set theory is an intelligent technique for managing uncertainties that is used for the discovery of data dependencies, to reduce redundancies, to evaluate the importance of attributes, to discover patterns in data and to classify objects. Several useful features of rough sets are:

- Extraction of rules from data sets in the form of if-then rules.
- It requires no external parameters unlike other intelligent techniques except the data itself.
- It can predict whether the data is complete or not.
- The computation of reduct and core using rough set theory is an important feature.
4.2.3.2.B.1. Feature Selection Using Rough Set [14]

Global and local feature selection methods can be clearly distinguished while using rough sets for selection of features. The whole data set are taken into consideration in global feature selection scheme while in local feature selection scheme only the descriptors of the form \((a, v)\), where \(a \in A\) and \(v \in V_a\) (set of values of attributes) are chosen for a given object. The relevant features for the object classification are searched in both strategies. In former case features are selected to define a partition of object universe, which along with other features are relevant for describing the approximation of partition (or part of it) as described by the decision attribute. In the local case descriptors are extracted for a given object with respect to a decision class.

One of the simplest approaches to select features using rough set is to calculate core for discrete attribute data set having strongly relevant features and reducts having a core with some additional weakly relevant features. Some feature selection criteria can be decided depending on a set of reducts. Bazan et al. (1994, 1998) proposed dynamic reducts for selecting robust feature subset. Based on cross validation method, this technique has been applied to extract features.

4.2.3.2.B.2. Reduct

Reducts are reduced subset of attributes which are obtained by removing superfluous attributes from the original data set. So this reduced system did not loss the classification performance. Each attribute in the reduct possess a particular property of the given information table. Attributes in the reduct are sufficient to describe the class attribute. Reducts are generated to reduce the number of attributes without losing the classification power provided by the all attributes of the original data set. For both positive, and boundary rule sets, this power may be interpreted by system and semantic properties [15].

Let us consider a system \(T = (U, A, C, D)\). If for all attributes \(c\) in \(C\) are indispensable, \(T\) is independent. A feature \(c\) is dispensable if \(\text{POS}_{(C-I)}(D) = \text{POS}_C(D)\). Taking this fact into consideration a reduct can be defined as a set of features \(R\) in \(C\), if \(T'=(U, A, R, D)\) is independent and \(\text{POS}_R(D) = \text{POS}_C(D)\). Moreover, no proper subset of \(R\) is same as \(\text{POS}_C(D)\). Thus a minimal set of attributes form a reduct, that preserves the above relation [16].
4.2.3.2.B.3. Core

The intersection of all reducts of an information table constitutes the core. All attributes in the core are indispensable in \( T = (U, A, C, D) \). So removal of any attribute from core will affect classification performance. Thus we can write:

\[ \text{CORE}(T) = \cap \text{RED}(T) \]

where \( \text{RED}(T) \) is the set of all reducts of \( T \). For the dependency between condition attributes \( C \) and decision attribute \( D \) (\( C \rightarrow D \)), if \( D \) depends on \( E \), where \( E \subset C \) then \( E \) is referred to as relative \( D \)-reduct of \( C \). Thus relative \( D \)-core of \( C \) is defined as:

\[ \text{CORE}_D(I) = \cap \text{RED}_D(I) \]

where \( \text{RED}_D(I) \) is set of all \( D \)-reducts of \( C \). Similarly \( \text{CORE}(I) \) represents set of all indispensable features in \( C \). \( E \) have \( \text{CORE}(I) = \text{RED}(I) \), where \( \text{RED}(I) \) denotes all the reducts of \( C \).

4.2.3.2.C. Principal Component Analysis (PCA)

It is one of the techniques used for feature extraction. The main advantage of PCA is – dimension will be reduced by avoiding redundant information, (Daugman, 1993) without much loss. Better understanding of PCA is through statistics and some of the mathematical techniques such as Eigen values, Eigen vectors.

PCA is a mathematical procedure that uses linear transformation to map data from high dimensional space to low dimensional space. The low dimensional space can be determined by Eigen vectors of the covariance matrix.

The steps involved in PCA include:

- The mean value of the given data set \( S \) is found.
- Subtract the mean value from \( S \). From these values a new matrix is obtained, say \( A \).
- Covariance (\( C \)) is obtained from this matrix i.e. \( C = AA^T \). Eigen values are obtained from the covariance matrix are \( V_1V_2V_3\ldots\ldots\ldotsV_n \).
- Finally Eigen vectors are calculated for covariance matrix \( C \).
- Sort Eigen vectors in decreasing order of Eigen values.
• Only Eigen vectors for largest Eigen values are kept to form lower dimensional data set.

• Once principal components (Eigen vectors) have been chosen, a feature vector has been formed by taking the transpose of the vector and multiplying it on the left of the original data set, transposed.

\[
\text{Final data} = \text{Row Feature Vector} \times \text{Row Data Adjust}
\]

where, Row Feature Vector is the matrix with the Eigen vectors in the columns transposed so that the Eigen vectors are in the rows, with the most significant Eigen vector at the top and Row Data Adjust is the mean-adjusted data transposed i.e. the data items are in each column, with each row holding a separate dimension.

### 4.2.3.2.D. Gene\(\text{c}\) Algorithm (GA)

Genetic Algorithms mimic Darwinian forces of natural selection to find optimum values of some function (Mitchell, 1998). Many studies used this algorithm to select effective features from data set. GA calculates an initial set of candidate solutions referred to as population and their corresponding fitness values. Larger fitness values are better. Each solution is known as an individual. Next population is made up of producing offspring by randomly combining the individuals with the best fitness values. To accomplish solution these individuals are selected and undergo crossover and also are subject to random mutations. By repeating the process many generations are created to produce better solution.

For feature selection, the individuals are subsets of predictors that are represented as binary; a feature is either included or not in the subset. The fitness values are some measure of model performance.

If the data set contains n number of attributes, there exist \(2^n\) chromosomes. Each chromosome is a binary vector of size n-bits, where bit 0 means the corresponding feature is not selected and bit 1 means the corresponding feature is selected. If n is large, the search space will be large enough to find out a suitable subset of features. GA is useful in solving large-scale problems to evaluate optimal or nearly optimal solution [17].

Initially a set of chromosomes i.e. population is generated randomly. Choosing the number of chromosomes in initial population is an important factor for performance of GA. Genetic diversity requires large population, which converges slowly.
Fitness function is the measure of quality of chromosomes of the population. GA is used to find out the chromosome (feature subset) with maximum fitness value.

Genetic operators, such as selection, crossover and mutation are used to converge into solution. Individuals (chromosomes) are selected for crossover. Crossover operator generates population for next generation. Each individual has a probability to mutate. Percent of bits of the selected individuals to be flipped in the mutation phase may be chosen randomly.

4.2.4. Missing Value Handling

4.2.4.A. Introduction

Missing data, a persistent problem in most scientific research, should be handled very carefully, as role of data are vital in every analysis. Mishandling missing values may cause distorted analysis or may generate biased results. Valid and reliable models require good data preparation. Dozens of techniques have been proposed by methodologists to address the problem. Appropriate method should be taken into consideration for a particular study in order to achieve efficient and valid analysis. In most scientific research domain like Biology [18], Medicine [19] missing data are common problems. One of the most challenging decisions confronting researcher is to choose the most appropriate method to handle missing data. Numerous methods are used in literature to handle missing data. Moreover handling missing data are not typically addressed in most literature. Unfortunately most of the statistical packages implement old standby techniques which are prone to statistical bias. There are different methods which are being used by people:

- Delete the records containing missing data;
- Use attribute mean;
- Use attribute median;
- Use a global constant to fill in for missing values which seem not relevant to the decision attribute;
- Use a data mining method.

We discuss different methods to handle missing data and compare three imputation methods: Arithmetic Mean Imputation, Regression Imputation and Multiple Imputation using EMB algorithm, performed on three data sets from UCI repository.
under the assumption of MAR, based on Root Mean Square Error (RMSE) as an evaluation criterion.

### 4.2.4.B. Missing Data Mechanisms

Rubin [20] defined missing data based on three missingness mechanisms [21] – Missing at Random (MAR), Missing Completely at Random (MCAR), and Missing Not at Random (MNAR).

Data are missing at random when there is a relation between the probability of missing data for a variable to some other measured variable or variables, but not to the values of itself. MAR as its name does not imply missing in haphazard fashion, but it actually means that the probability of missing data is systematically related to other variable.

Data are missing completely at random when the probability of missing data for a variable is unrelated to any other measured variable and to the values of itself. MCAR implies missing completely in haphazard fashion. MCAR is a more restrictive condition than MAR as it assumes that missingness is completely unrelated to the data [22].

Data are missing not at random when the probability of missing data for a variable is related to the values of itself, even after controlling for other variable.

### 4.2.4.C. Techniques to Handle Missing Values

Dealing with missing data includes – removing the cases with missing values or imputing the missing values. Dozens of techniques have been found in literature to handle missing data problem. Some of these techniques are – List-wise deletion, Pair-wise deletion, Arithmetic Mean Imputation, Regression Imputation, Multiple Imputation with EMB approach.

#### 4.2.4.C.1. List-Wise Deletion

In list-wise deletion method data for any case which has one or more missing values are deleted. This is why the method is also known as complete-case analysis [23]. The main advantage of this method is that it is easy to implement and also available as standard option for statistical packages. In most situations the resulting reduced dataset as obtained by applying list-wise deletion may lead to decreased statistical
analysis power and also important knowledge may be missed. Another disadvantage is that this method assumes MCAR. If data are not in MCAR, list-wise deletion produces distorted result. In particular for large dataset where missing values are very minimal, this method may be appropriate.

4.2.4.C.2. Pair-Wise Deletion

To mitigate the loss of data that occurs in list-wise deletion, pair-wise deletion method eliminates cases on an analysis by analysis basis only on available cases. Pair-wise deletion uses the subset of cases with complete data for each pair of variables to compute correlation or covariance matrix. The strength of association between a pair of variables is measured by correlation. The correlation coefficients for each pair of variables for which data are available will take the data into account. Thus pair-wise deletion maximizes the use of data as much as possible, which increases the power of analysis. Pair-wise deletion method tends to be more powerful than list-wise deletion, particularly when the variables in a dataset have low to moderate correlations. The main advantage of pair-wise deletion is that it is easy to implement and also available in standard statistical packages.

The disadvantage of pair-wise deletion is that if the assumption of MCAR does not hold, it produces distorted result as it requires data in MCAR. In pair-wise deletion it is difficult to compute standard errors as average sample size is used to the entire correlation matrix. Thus it produces standard errors either underestimated or overestimated. Another disadvantage is that this technique may yield correlation outside [-1,1] which causes estimation problems for multivariate analyses that use correlation matrix as input.

4.2.4.C.3. Single Imputation

Single imputation methods impute data for unobserved values in the dataset prior to analysis. It replaces a single value for each missing value in the dataset. Out of many single imputation methods available we discussed two of them – Arithmetic Mean Imputation and Regression Imputation.

1) Arithmetic Mean Imputation: In this method the arithmetic mean of observed values for an attribute replaces all the missing values for that attribute. This is the simplest imputation method, but produces biased result. It increases the size of sample as well as the power of analysis. According to Rubin [21] mean substitution decreases the variability in the dataset, as mean that is the same value is used as a substitute for all the missing values.
2) **REGRESSION IMPUTATION:** It uses regression to predict missing values from other variables of known values. Variables containing missing data is assumed to be dependent while the other variables are considered as independent. If we consider bivariate dataset with attribute X and Y, missing values are computed from the regression equation (Eq. 1.):

\[
Y = b \times X + a
\]  
\tag{1}

Here we assume that value of dependent variable Y is to be predicted from independent variable X by estimating the regression with the available data of X and Y. The values of a and b are computed from the following formulae (Eq. 2. and Eq. 3.).

\[
a = \frac{\sum y \times \sum x^2 - \sum x \times \sum(x \times y)}{n \times \sum x^2 - (\sum x)^2} \]  
\tag{2}

\[
b = \frac{n \times \sum(x \times y) - \sum x \times \sum y}{n \times \sum x^2 - (\sum x)^2} \]  
\tag{3}

Regression imputation is better than mean imputation, but it also has predictable biases.

### 4.2.4.C.4. Multiple Imputations

A bootstrap-based EMB algorithm [24] performs multiple imputations for missing values. In multiple imputations, values are imputed for each missing value of the data set and completed m data sets are generated. In these imputed data sets with complete data, the known values remain same for each set but the imputed missing values may be different for each set. After imputation, analysis is done with each imputed data set and the results are combined. There are different combination techniques one can adopt [24, 25].

Figure 4.3 shows the schematic view of Multiple Imputation using EMB approach. Multiple imputations are found to produce more accurate results compared to list-wise deletion, arithmetic mean imputation. This technique reduces bias and increases efficiency. In this multiple imputation technique, MAR (missing at random) is assumed. It considers MAR, likelihood, law of iterated expectations, and a flat prior to compute posterior. From the posterior, it has to take draws. The EM [26] algorithm is
to find the mode of the posterior. This EMB algorithm uses the EM algorithm with bootstrap approach to take draws from this posterior. For each draw, the data is bootstrapped to simulate estimation uncertainty and then run EM algorithm to find the mode of the posterior for the bootstrapped data, which also gives fundamental uncertainty [27]. After having draws imputations are done using observed part \( D(\text{observed}) \) and unobserved part \( D(\text{missing}) \) as well as mean vector \( \mu \) and covariance matrix \( \Sigma \) with linear regression.

Figure 4.3. Schema View of Multiple Imputations

4.2.4.D. Principle of Analysis

Considering the variability of relative performance of different methods across datasets, results were generated based on three reference datasets: Breast Cancer dataset, Chronic Kidney Disease Dataset and Hepatitis Disease Dataset.
Chapter 4

Figure 4.4. Block Diagram of Principle of Analysis

Figure 4.4 shows the general principle of analysis. From the original data sets without missing values we produced bivariate data sets by selecting only two attributes from each data set and also introduced in the data a varying percentage of missing values (e.g., 10%, 20% and 30%) in such a way that MAR is assumed. From Breast Cancer data set we selected the attributes – Clump Thickness and Uniformity of Cell Size. The two attributes which are selected from Chronic Kidney Disease data set are – Albumin and Serum Creatinine. Similarly, from Hepatitis Disease data set we chose attributes – Albumin and Billirubin. The values of dependent variables (for Breast Cancer data set – Uniformity of Cell Size, for CKD data set - Serum Creatinine, for Hepatitis data set – Billirubin) are missing at random (MAR) as they are systematically missing as a function of respective independent variables (for Breast Cancer data set – Clump Thickness, for CKD data set – Albumin, for Hepatitis data set – Albumin). These simulated missing values are imputed using 3 methods - Arithmetic Mean, Regression and Multiple Imputation using EMB approach. Performances are measured by evaluating Root Mean Square Error (RMSE).
We compare three imputation methods on the basis of Root Mean Square Error (RMSE) (Eq. 4.), which measures the difference between imputed value and true value.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(x_{i}^{\text{obs}} - x_{i}^{\text{imputed}})^2}{n}} \quad (4)$$

### 4.2.4.E. Results

Results are summarized in Table 4.1, Table 4.2 and Table 4.3.

<table>
<thead>
<tr>
<th>Percentage of missing value</th>
<th>Imputation Methods</th>
<th>Root Mean Square Error (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Arithmetic Mean</td>
<td>4.831</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>2.811</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>2.867</td>
</tr>
<tr>
<td>20</td>
<td>Arithmetic Mean</td>
<td>5.226</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>2.86</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>3.077</td>
</tr>
<tr>
<td>30</td>
<td>Arithmetic Mean</td>
<td>5.203</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>3.033</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>3.494</td>
</tr>
</tbody>
</table>
### Table 4.2. Results for Chronic Kidney Disease Data Set of UCI

<table>
<thead>
<tr>
<th>Percentage of missing value</th>
<th>Imputation Methods</th>
<th>Root Mean Square Error (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Arithmetic Mean</td>
<td>5.376</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>4.646</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>4.785</td>
</tr>
<tr>
<td>20</td>
<td>Arithmetic Mean</td>
<td>6.52</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>5.591</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>5.765</td>
</tr>
<tr>
<td>30</td>
<td>Arithmetic Mean</td>
<td>1.824</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>0.287</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>1.193</td>
</tr>
</tbody>
</table>

### Table 4.3. Results for Hepatitis Disease Data Set of UCI

<table>
<thead>
<tr>
<th>Percentage of missing value</th>
<th>Imputation Methods</th>
<th>Root Mean Square Error (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Arithmetic Mean</td>
<td>0.782</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>1.375</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>1.621</td>
</tr>
<tr>
<td>20</td>
<td>Arithmetic Mean</td>
<td>0.836</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>1.74</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>1.573</td>
</tr>
<tr>
<td>30</td>
<td>Arithmetic Mean</td>
<td>0.948</td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>0.838</td>
</tr>
<tr>
<td></td>
<td>Multiple Imputation using EMB</td>
<td>0.795</td>
</tr>
</tbody>
</table>
From the above tables it is observed that in almost all cases performance of Regression Imputation and Multiple Imputation using EMB are same, though in most of the cases regression imputation provides better result than the later. In case of Hepatitis Disease data set for 10% and 20% missing values imputation using Arithmetic Mean leads to better result as compared to other two methods, but for 30% missing values multiple imputations shows better performance.

4.3. Discussion

Missing data, a part of many studies, are handled by several alternative ways to overcome the drawbacks. Comparative studies are needed to ensure which imputation method should be well suited for a particular study. Only a few literatures address an evaluation of existing imputation methods.

In this work, we performed a neutral comparative study of three imputation methods based on three UCI data sets of various sizes under the assumption of MAR. We did not consider elimination processes like List-Wise deletion and Pair-Wise deletion, as these methods are applicable only for large data set with minimal number of missing values, otherwise there may be a chance of losing important information. So, we concentrated only on imputation methods. Imputation accuracy is measured by Root Mean Square Error (RMSE).

The limitation of our study is that the results are limited to data matrices of numerical values. Careful attention should be taken into consideration for other type of variables also [28].

In conclusion, it can be suggested that there is no universal imputation method performing best in every situation, but for bivariate data set if the data are missing at random, imputation using regression should be taken into consideration. For multivariate data set the regression imputation is somewhat complicated to implement. Regression imputation also requires data which are missing at random. So it is also suggested to consider multiple imputation approaches for multivariate data set which are in MAR or MCAR.
4.4. Conclusion

Data preprocessing is one of the most important issues in data mining as good data preparation is the first step to produce valid and reliable intelligent models. Summarization of descriptive data should be necessary to increase the performance of automated systems. Most of the data mining studies focus on data preprocessing to produce good quality data, which is a key to improve predictive performance of an intelligent model. High quality data produces high quality mining model. Different methods which are discussed include data cleaning, data integration, data transformation and data reduction. Though different related methods are studied, still it may be considered as an important area of further research.
References

1. Ham, Kamber, Data Pre-processing and Mining Algorithm, Knowledge and Data Mining and Pre-processing, 3rd Edition.


