4 DATA PREPROCESSING

4.1 Data Normalization

4.1.1 Min-Max

4.1.2 Z-Score

4.1.3 Decimal Scaling

4.2 Data Imputation

4.2.1 Bayesian Principal Component Analysis

4.2.2 K Nearest Neighbor

4.2.3 Weighted K Nearest Neighbor

4.2.4 Local Least Square

4.2.5 Iterated Local Least Square

4.2.6 Regularized Expectation Maximization

4.3 Experimental Results

4.4 Chapter Summary
Data preprocessing is a fundamental building block of the KDD process. It prepares the data by removing outliers, smoothing noisy data and imputing the missing values in the dataset. Though most of the data mining techniques have predefined noise handling and imputing data mechanisms, preprocessing reduces the confusion during the learning process. In addition, the acquired datasets from the different data sources may undergo several data preprocessing techniques to produce a final result. The simplified and specialized data preprocessing techniques in the knowledge discovery process are listed as follows:

- **Data cleaning**
- **Data integration**
- **Data transformation**
- **Data reduction**
- **Data discrimination**

Data cleaning identifies the origin of errors that are detected in the dataset and using that information, it prevents the errors from recurring in the dataset. Thus, the inconsistency in the dataset is removed and data quality is improved. This preprocessing technique is extensively used in data warehouses. Data integration is a crucial problem in designing the decision support systems and data warehouses. Therefore, data from different data sources are merged together into an appropriate form that is suitable for mining the patterns. It is used to create a coherent data repository from data sources that include multiple databases, flat files or data cubes.

Data transformation consolidates the data into a specific format that helps to mine the feasible patterns easily. Data transformation can be performed using different techniques like smoothing, generalization, normalization and feature construction. This is depicted in Figure 4.1. Data reduction technique reduces the representation of an original dataset into a smaller subset. Usually data reduction
techniques can be applied to multidimensional data, where the data must be cubed and given as an input to the reduction algorithms. The input given to the reduction algorithms should be non-empty samples to reduce the approximation error. The reduced dataset should retain the integrity of an original dataset and produce almost the same experimental results.

![Figure 4.1 Taxonomy of Data Transformation techniques]

Data discrimination generates the discriminant rules that compare the feature values of the dataset between the two classes i.e. referred as target class and contrasting class. In discriminant analysis, multivariate instances with different classes are observed together to form the training data sample. Using the instance of training data the class label is known and it is used to classify the new data instances into one of the predefined classes. The following are the reasons where the different data preprocessing techniques are often applied to multiple data sources

- To apply data mining algorithms easily
- To enhance the performance and effectiveness of data mining algorithms
- To represent the data in an understandable format
- To retrieve the data from databases and warehouses quickly and
- To make the datasets suitable for an explicit data analysis
Chapter I

Introduction

The above listed data preprocessing techniques help in improving the accuracy and efficiency of the classification process. From the data analysis, the two techniques that are required to preprocess the considered datasets in this research work are data normalization and data imputation.

4.1 Data Normalization

Data normalization is a preprocessing technique where it groups the given data into a well refined format. The success of machine learning algorithm largely depends on the quality of the datasets chosen. Thus, data normalization is an important transformation technique where it can improve the accuracy and accomplish better performance in considered datasets. Realizing the significance of transformation techniques in data mining algorithms, normalization technique is used here to improve the generalization process and learning capability with minimum error.

Normally, the feature values in the dataset are in different scales of measurement. Some features may be integer values while others may be decimal values. The data normalization technique is used to manage and organize the feature values in the dataset. Also, it scales the feature values to the same specified range. Normalization is used in classification and clustering techniques, since the input data should not be overwhelmed by other data points in terms of distance metric. It minimizes bias and speeds up the training time in the classification process because each feature value starts in the same range.

From the literature, it is evident that the different types of normalization techniques are logarithmic, sigmoid, statistical column, median, min max, z-score and decimal scaling. Logarithmic normalization (Zavadskas and Turskis, 2008) normalizes the datasets where the vector component is skewed and distributed exponentially. This normalization technique is based on non-linear transformation that best represents the data values. If the input values in the dataset are clustered
around minimum values with few maximum values then this transformation can be applied to give better results.

The sigmoid normalization technique (Jayalakshmi and Santhakumaran, 2011) scales the dataset in the range of 0-1 or (+1, -1). There are different kinds of non-linear sigmoid based normalization techniques. Among these, tan sigmoid normalization technique is feasible since it estimates the parameters from the noisy data. Statistical column normalization technique (Jayalakshmi and Santhakumaran, 2011) normalizes each data value by normalizing its column value. In median based normalization (Jayalakshmi and Santhakumaran, 2011), each sample is normalized by the median of input values in the dataset. It can be applied when there is a requirement, to ascertain the ratio between two samples. It is also used in the datasets that perform the distribution between the input samples.

In this classification framework, three kinds of data normalization techniques that can enhance support vector machines are applied for the binary and multiclass datasets. By applying and comparing these techniques, a best one is identified. The three data normalization techniques that are used in the classification framework are as follows:

**4.1.1 Min-Max**

The min-max normalization technique (Kotsiantis et.al. 2006) normalizes the dataset using linear transformation and transforms the input data into a new fixed range. Min-max technique preserves the associations between the original input value and the scaled value. Also, an out of bound error is encountered when the normalized values deviate from the original data range. This technique ensures that extreme input values are constrained within a specific range. Min-max normalization transforms transforms a value $X_0$ to $X_n$ which fits in the specified range and it is given by the equation (4.1)
Chapter I

Introduction

A Framework for Admissible Kernel Function in Support Vector Machines using Lévy Distribution

\[ X_n = \frac{X_0 - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \]  

(4.1)

where \( X_n \) is a new value for variable \( X \), \( X_0 \) is a current value for variable \( X \), \( X_{\text{min}} \) is the minimum data point in the dataset and \( X_{\text{max}} \) is the maximum data point in the dataset.

4.1.2 Z-Score

Z-score normalization (Kotsiantis et al. 2006) is also known as zero-mean normalization. Z-score normalization technique normalizes the input values in the dataset using mean and standard deviation. The mean and standard deviation for each feature vector is calculated across the training dataset. This normalization technique determines whether an input value is below or above the average value. It will be very useful to normalize the dataset when the attribute's maximum or minimum values are unknown and outliers dominate the input values. This technique transforms a value \( v \) to \( v' \) by the equation (4.2)

\[ v' = ((v - \bar{A}) / \sigma_A) \]  

(4.2)

where \( v' \) is a new value of an attribute, \( v \) is an old value of an attribute, \( \bar{A} \) is the mean of an attribute value \( A \) and \( \sigma \) is the standard deviation of an attribute value \( A \).

4.1.3 Decimal Scaling

Decimal scaling normalization (Jayalakshmi and Santhakumaran, 2011) is the simplest transformation technique that normalizes an attribute by moving the decimal point of the input values. Maximum absolute value of an input attribute decides the number of decimal points to be moved in a value. It is shown in the equation (4.3)

\[ v' = (v / 10^j) \]  

(4.3)

where \( v' \) is the new value, \( v \) is an old value and \( j \) is the smallest integer value such that \( \text{Max} \ (|v'| < 1) \).
4.2 Data Imputation

Missing data is an unrelenting problem in all areas of recent empirical research. This problem should be treated carefully since data plays a key role in every domain analysis. If this missing data problem is handled improperly, then it will produce biased results and distort the data analysis. Even though there are various techniques available in the literature to overcome the missing data problem, data imputation is a technique that imputes the missing data approximately and reduces the estimation error.

The main objective of data imputation technique is to create an inclusive dataset, where it can be analyzed by an inferential method. Data imputation is broadly categorized into two types. They are single imputation and multiple imputation. However, choosing the most reliable imputation technique to fill the missing data is a challenging issue for the researchers. Figure 4.2 depicts the different techniques that are used to overcome the missing data problem.
Single value imputation is a simple technique which imputes a single value for a missing data. Single value based imputation has a disadvantage that it reproduces an additional uncertainty in dataset. This disadvantage is replaced by a new technique i.e. multiple imputation, proposed by (Rubin, 1976). In this technique, imputation takes place repeatedly to create multiple imputed dataset. Each imputed dataset is analyzed statistically and generates multiple result where all the results are combined to present an overall result. Multiple imputation is an attractive choice for researchers who deal with real time problems. It also performs favorably by producing unbiased results.

Single/Multiple Imputation techniques are classified into three types. They are global based imputation, neighbor based imputation and model based imputation. Global based imputation technique imputes the missing data using eigen vectors and the techniques related to global imputation are partial least squares, singular value decomposition and Bayesian Principal Component Analysis (BPCA). Neighbor based imputation technique uses a distance measure to impute a missing data. least square analysis, K Nearest Neighbor (KNN), Weighted K Nearest Neighbor (Wt. KNN), Least Square (LLS) and Iterated Local Least Square (It. LLS) are some of the methods in this category. In model based imputation, a predictive model is created to estimate a missing value. The techniques are maximum likelihood, expectation Maximization and Regularized Expectation Maximization (Reg. EM).

Data imputation technique helps to fill the missing data with a feasible value, but before substituting the missing value the type of missingness should be identified. There are two reasons to distinguish the type of missingness in datasets. First, it helps to check how well the relation between the attribute values are represented (Schafer and Graham, 2002). Next, it identifies the missing data patterns that need to be imputed. There are three different kinds of missingness (Little and Rubin, 1987) and they are as follows
Missing completely at random (MCAR)

Missing at random (MAR)

Missing not at random (MNAR)

Missing completely at random

Missing completely at random is one type of missingness where the probability of missing data is totally due to the unrelated events and not because of the attributes in a dataset (Schafer and Graham, 2002; Streiner, 2002). This type of missingness occurs rarely so that it is better to categorize the type of missing data and impute the values.

Missing at random

In missing at random, the missingness occurs by removing the data that may be interrelated to the other attribute values in the dataset (Schafer and Graham, 2002; Streiner, 2002).

Missing not at random

Missing not at random is a missingness that often arises in the datasets. The reason for MNAR missingness is removing the outcome of one or more attribute values and it has an organized pattern (Pigott, 2001; Schafer and Graham, 2002).

Usually MCAR and MAR based missingness can be ignored but MNAR cannot be ignored because missing values due to MNAR are not recoverable. Missing data problem has a major impact in the feature selection and classification process, so data imputation technique is used here to make the datasets reliable to the classification framework. Based on the literature, six different data imputation techniques are considered and examined using the binary and multiclass datasets. These techniques can also improve the accuracy and robustness of the kernel based classifier framework. Following are the imputation techniques that are used in this framework.
4.2.1. Bayesian Principal Component Analysis

Bayesian principal component analysis (Oba et al. 2003) uses statistical procedure to impute the arbitrary missing data. BPCA imputation presents an accurate and suitable estimation for missing values. Basically BPCA is dependent on probabilistic principal component and it uses a Bayes technique that iteratively estimates the posterior distribution for missing data until it converges. The three primary processes that are involved in BPCA are

✓ Principal component regression
✓ Bayesian estimation and
✓ Expectation maximization like repetitive algorithm.

4.2.2. K Nearest Neighbor

The KNN imputation technique (Sun et al. 2009) is used to estimate and fill the missing values in the dataset. The key factor of KNN imputation technique is distance metric and it is a lazy learner. In KNN imputation, missing values are imputed by combining the columns of K nearest attribute values in a dataset based on the similarity metric. Here, similarity metric calculates the distance between complete record and incomplete record. The three strategies that are required to estimate KNN imputation are as follows

✓ Value of K should be decided
✓ Need training data with labeled classes
✓ Metric that measures closeness property

4.2.3. Weighted K Nearest Neighbor

Imputing the dataset using K nearest neighbor sometimes leads to loss of information. So weighted K nearest neighbor is introduced (Troyanskaya et al.
Chapter I

Introduction

A Framework for Admissible Kernel Function in Support Vector Machines using Lévy Distribution

2001). The only difference between K nearest neighbor and weighted K nearest neighbor is Wt. KNN imputes the dataset using a dynamically assigned K value.

4.2.4. Local Least Square

In local least square imputation (Kim et al. 2004), an absolute value of pearson correlation coefficient is defined as similarity metric to select the k attribute values which results in a local least square pearson correlation based imputation. Instead of Pearson correlation, L2 norm is used as a similarity metric where it improves the results. Also, the missing data is imputed as a linear combination of missing value attributes. After defining the similarity metric, the missing value is imputed as a linear combination of consequent values of the attribute.

4.2.5. Iterated Local Least Square

Iterated Local Least Square imputation (Cai et al. 2005) is used to impute the missing data more accurately. It is often used to impute the microarray gene expression data. Iterated Local Least Square based imputation technique consists of three steps. They are

✓ Similarity threshold value is used to estimate the known attribute value
✓ Next, the threshold value is used in local least square based imputation
✓ Several iterations are performed to obtain an estimate value for missing data.

4.2.6. Regularized Expectation Maximization

Regularized expectation maximization imputation technique (Schneider, 2001) has the same steps as in expectation maximization. But, expectation maximization algorithm cannot be applied for datasets where the number of variables exceed the input size. Due to this shortcoming, expectation maximization imputation technique revised as regularized to impute the missing data. The three
steps that are involved in regularized expectation maximization algorithm are as follows

- Compute the regression parameters from the estimates of the mean and covariance
- Impute the missing values with their conditional expectation values
- Iterate the EM algorithm until it imputes all the missing values

### 4.3 Experimental Results

The experimental results are carried out using binary and multiclass datasets that are taken from UCI machine learning repository. The dataset description is given inclusively in the previous chapter. The performance of data normalization and data imputation techniques are examined and recorded for evaluation. Performance metrics that are used to evaluate the data normalization techniques are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Squared Error with Regularization (MSEREG) and time. They are given by the equations (4.4 - 4.6). Tables 4.1 and 4.2 depict the performance of data normalization techniques for binary and multiclass datasets.

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 
\]  \hspace{1cm} (4.4)

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2} 
\]  \hspace{1cm} (4.5)

\[
MSEREG = \gamma \cdot MSE + (1 - \gamma) \cdot MSW, \text{ where } \ MSW = \frac{1}{n} \sum_{j=1}^{n} w_j^2 
\]  \hspace{1cm} (4.6)

where \( Y_i \) is a true value and \( \hat{Y}_i \) is an estimated value of an attribute.
Table 4.1 Performance of Normalization techniques for Binary datasets

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Normalization Technique</th>
<th>MSE</th>
<th>RMSE</th>
<th>MSEREG</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>Min-Max</td>
<td>1.3028</td>
<td>1.1414</td>
<td>1.1725</td>
<td>0.0018</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9933</td>
<td>0.9966</td>
<td>0.8940</td>
<td>0.0033</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.1596</td>
<td>0.3996</td>
<td>0.1437</td>
<td>0.0007</td>
</tr>
<tr>
<td>Liver</td>
<td>Min-Max</td>
<td>1.053</td>
<td>1.0262</td>
<td>0.9477</td>
<td>0.0018</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9971</td>
<td>0.9985</td>
<td>0.8973</td>
<td>0.0053</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.3073</td>
<td>0.5543</td>
<td>0.2766</td>
<td>0.0009</td>
</tr>
<tr>
<td>Heart</td>
<td>Min-Max</td>
<td>0.923</td>
<td>0.9607</td>
<td>0.8307</td>
<td>0.0011</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9963</td>
<td>0.9981</td>
<td>0.8966</td>
<td>0.0064</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.8351</td>
<td>0.9138</td>
<td>0.7516</td>
<td>0.0016</td>
</tr>
<tr>
<td>Diabetes</td>
<td>Min-Max</td>
<td>0.1968</td>
<td>0.4437</td>
<td>0.1771</td>
<td>0.0011</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9987</td>
<td>0.9993</td>
<td>0.8988</td>
<td>0.0104</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.5437</td>
<td>0.7373</td>
<td>0.4893</td>
<td>0.0010</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>Min-Max</td>
<td>0.2024</td>
<td>0.4499</td>
<td>0.1821</td>
<td>0.0036</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9985</td>
<td>0.9992</td>
<td>0.8987</td>
<td>0.0032</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.1532</td>
<td>0.3914</td>
<td>0.1379</td>
<td>0.0012</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>Min-Max</td>
<td>2.1467</td>
<td>1.4651</td>
<td>1.932</td>
<td>0.0012</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9935</td>
<td>0.9967</td>
<td>0.8941</td>
<td>0.0046</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.1701</td>
<td>0.4125</td>
<td>0.1531</td>
<td>0.0011</td>
</tr>
<tr>
<td>Ripley</td>
<td>Min-Max</td>
<td>0.1249</td>
<td>0.3553</td>
<td>0.1019</td>
<td>0.0018</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.999</td>
<td>0.9995</td>
<td>0.8991</td>
<td>0.0036</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.0028</td>
<td>0.0531</td>
<td>0.0025</td>
<td>0.0009</td>
</tr>
</tbody>
</table>
Chapter I

Introduction

Metrics that are used to evaluate the data imputation techniques are MSE, RMSE, MSEREG, Mean Absolute Error (MAE) and time. They are given by the equations (4.7-4.10). Tables 4.3 and 4.4 represent the performance analysis of data imputation techniques for binary and multiclass datasets.

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{4.7}
\]

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2} \tag{4.8}
\]

\[
MSEREG = \gamma \cdot MSE + (1 - \gamma) \cdot MSW, \text{ where } \ MSW = \frac{1}{n} \sum_{j=1}^{n} w_j^2 \tag{4.9}
\]

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \hat{Y}_i| \tag{4.10}
\]

where \(Y_i\) is a true value and \(\hat{Y}_i\) is an estimated value of an attribute. Though the different data normalization techniques minimize the estimation error, the empirical results from Tables 4.1 and 4.2 indicate that the decimal scaling based normalization produce the best result with minimum mean squared error, root mean squared error, mean squared error with regularization and time for the considered binary and multiclass datasets.

From the Tables 4.3 and 4.4, it is known that the K nearest neighbor decreases the mean squared error, root mean squared error, mean squared error with regularization, mean absolute error and time when compared to the other techniques for the binary and multiclass datasets used in the experiments. The data preprocessing techniques that refine the results and improve the reliability of the datasets are used in this classification framework. Also, the experimental results has shown that the performance of the classification framework depends on the data preprocessing techniques.
Table 4.2 Performance of Normalization techniques for Multiclass datasets

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Technique</th>
<th>MSE</th>
<th>RMSE</th>
<th>MSEREG</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>Min-Max</td>
<td>1.3028</td>
<td>1.1414</td>
<td>1.1725</td>
<td>0.0018</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9933</td>
<td>0.9966</td>
<td>0.8940</td>
<td>0.0033</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.1596</td>
<td>0.3996</td>
<td>0.1437</td>
<td>0.0007</td>
</tr>
<tr>
<td>Glass</td>
<td>Min-Max</td>
<td>51.296</td>
<td>22.649</td>
<td>46.166</td>
<td>0.0011</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.8957</td>
<td>0.9464</td>
<td>0.8062</td>
<td>0.0012</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.0616</td>
<td>0.2482</td>
<td>0.0554</td>
<td>0.0009</td>
</tr>
<tr>
<td>E-Coli</td>
<td>Min-Max</td>
<td>0.5217</td>
<td>0.7223</td>
<td>0.4696</td>
<td>0.0018</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.8724</td>
<td>0.9340</td>
<td>0.7851</td>
<td>0.0015</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.0027</td>
<td>0.0524</td>
<td>0.0024</td>
<td>0.0010</td>
</tr>
<tr>
<td>Wine</td>
<td>Min-Max</td>
<td>1.5191</td>
<td>1.2325</td>
<td>1.3672</td>
<td>0.0025</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9943</td>
<td>0.9971</td>
<td>0.8949</td>
<td>0.0040</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.0514</td>
<td>0.2268</td>
<td>0.0463</td>
<td>0.0015</td>
</tr>
<tr>
<td>Balance Scale</td>
<td>Min-Max</td>
<td>0.6875</td>
<td>0.8291</td>
<td>0.6187</td>
<td>0.0022</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9984</td>
<td>0.9992</td>
<td>0.8985</td>
<td>0.0044</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.11</td>
<td>0.3316</td>
<td>0.099</td>
<td>0.0007</td>
</tr>
<tr>
<td>Lenses</td>
<td>Min-Max</td>
<td>2.3132</td>
<td>1.5209</td>
<td>2.0819</td>
<td>0.0010</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9583</td>
<td>0.9789</td>
<td>0.8625</td>
<td>0.0010</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.0354</td>
<td>0.1884</td>
<td>0.0319</td>
<td>0.0007</td>
</tr>
<tr>
<td>Pentagon</td>
<td>Min-Max</td>
<td>0.0859</td>
<td>0.2931</td>
<td>0.0773</td>
<td>0.0018</td>
</tr>
<tr>
<td></td>
<td>Z-Score</td>
<td>0.9899</td>
<td>0.9949</td>
<td>0.8909</td>
<td>0.0037</td>
</tr>
<tr>
<td></td>
<td>Decimal Scaling</td>
<td>0.0026</td>
<td>0.0510</td>
<td>0.0023</td>
<td>0.0006</td>
</tr>
</tbody>
</table>
### Table 4.3 Performance of Imputation techniques for Binary datasets

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Technique</th>
<th>MSE</th>
<th>RMSE</th>
<th>MSEREG</th>
<th>MAE</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>BPCA</td>
<td>0.1590</td>
<td>0.3987</td>
<td>0.1431</td>
<td>0.3464</td>
<td>0.0453</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.1591</td>
<td>0.3988</td>
<td>0.1432</td>
<td>0.3466</td>
<td>0.0207</td>
</tr>
<tr>
<td></td>
<td>Itr. LLS</td>
<td>0.1590</td>
<td>0.3987</td>
<td>0.1431</td>
<td>0.3464</td>
<td>0.0402</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td><strong>0.1587</strong></td>
<td><strong>0.3984</strong></td>
<td><strong>0.1429</strong></td>
<td><strong>0.3464</strong></td>
<td><strong>0.003</strong></td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.1591</td>
<td>0.3989</td>
<td>0.1432</td>
<td>0.3470</td>
<td>0.0034</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.1589</td>
<td>0.3986</td>
<td>0.1430</td>
<td>0.3465</td>
<td>0.0467</td>
</tr>
<tr>
<td>Liver</td>
<td>BPCA</td>
<td>0.3068</td>
<td>0.5539</td>
<td>0.2761</td>
<td>0.4281</td>
<td>0.0839</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.3067</td>
<td>0.5538</td>
<td>0.276</td>
<td>0.4283</td>
<td>0.0035</td>
</tr>
<tr>
<td></td>
<td>Itr. LLS</td>
<td>0.3066</td>
<td>0.5537</td>
<td>0.276</td>
<td>0.428</td>
<td>0.078</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td><strong>0.3066</strong></td>
<td><strong>0.5536</strong></td>
<td><strong>0.278</strong></td>
<td><strong>0.4279</strong></td>
<td><strong>0.003</strong></td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.3065</td>
<td>0.5537</td>
<td>0.2759</td>
<td>0.4281</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.3068</td>
<td>0.5539</td>
<td>0.2761</td>
<td>0.4281</td>
<td>0.0423</td>
</tr>
<tr>
<td>Heart</td>
<td>BPCA</td>
<td>0.8355</td>
<td>0.8355</td>
<td>0.7519</td>
<td>0.4612</td>
<td>0.1365</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.8316</td>
<td>0.9119</td>
<td>0.7484</td>
<td>0.461</td>
<td>0.031</td>
</tr>
<tr>
<td></td>
<td>Itr. LLS</td>
<td>0.8356</td>
<td>0.9141</td>
<td>0.752</td>
<td>0.4613</td>
<td>0.1249</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.8334</td>
<td>0.9129</td>
<td>0.7501</td>
<td>0.4606</td>
<td>0.0041</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td><strong>0.8316</strong></td>
<td><strong>0.9119</strong></td>
<td><strong>0.7484</strong></td>
<td><strong>0.4598</strong></td>
<td><strong>0.0032</strong></td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.8355</td>
<td>0.914</td>
<td>0.7519</td>
<td>0.4612</td>
<td>0.0982</td>
</tr>
<tr>
<td>Diabetes</td>
<td>BPCA</td>
<td>0.543</td>
<td>0.7369</td>
<td>0.4888</td>
<td>0.4499</td>
<td>0.1321</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.5431</td>
<td>0.7370</td>
<td>0.4887</td>
<td>0.45</td>
<td>0.0047</td>
</tr>
<tr>
<td></td>
<td>Itr. LLS</td>
<td>0.5432</td>
<td>0.7370</td>
<td>0.4888</td>
<td>0.4499</td>
<td>0.1358</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td><strong>0.5429</strong></td>
<td><strong>0.7368</strong></td>
<td><strong>0.4886</strong></td>
<td><strong>0.4498</strong></td>
<td><strong>0.0038</strong></td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.5428</td>
<td>0.7367</td>
<td>0.4885</td>
<td>0.4497</td>
<td>0.0034</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.543</td>
<td>0.7369</td>
<td>0.4887</td>
<td>0.4498</td>
<td>0.0598</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>BPCA</td>
<td>0.1529</td>
<td>0.391</td>
<td>0.1376</td>
<td>0.1072</td>
<td>0.0472</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.1529</td>
<td>0.391</td>
<td>0.1376</td>
<td>0.1073</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>Itr. LLS</td>
<td>0.1528</td>
<td>0.391</td>
<td>0.1375</td>
<td>0.1071</td>
<td>0.1623</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td><strong>0.1528</strong></td>
<td><strong>0.391</strong></td>
<td><strong>0.1375</strong></td>
<td><strong>0.1071</strong></td>
<td><strong>0.0037</strong></td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.1520</td>
<td>0.3899</td>
<td>0.1371</td>
<td>0.1070</td>
<td>0.0054</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.1528</td>
<td>0.391</td>
<td>0.1375</td>
<td>0.1071</td>
<td>0.1390</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>BPCA</td>
<td>0.1841</td>
<td>0.4291</td>
<td>0.1657</td>
<td>0.1693</td>
<td>2.6188</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.1624</td>
<td>0.403</td>
<td>0.1461</td>
<td>0.15</td>
<td>0.0061</td>
</tr>
<tr>
<td></td>
<td>Itr. LLS</td>
<td>0.185</td>
<td>0.4301</td>
<td>0.1665</td>
<td>0.17</td>
<td>0.6117</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.1857</td>
<td>0.4309</td>
<td>0.1671</td>
<td>0.1614</td>
<td>0.017</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td><strong>0.1653</strong></td>
<td><strong>0.4066</strong></td>
<td><strong>0.1487</strong></td>
<td><strong>0.1514</strong></td>
<td><strong>0.0089</strong></td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.182</td>
<td>0.4266</td>
<td>0.1638</td>
<td>0.1691</td>
<td>0.6832</td>
</tr>
<tr>
<td>Ripley</td>
<td>BPCA</td>
<td>0.0028</td>
<td>0.0529</td>
<td>0.0025</td>
<td>0.0469</td>
<td>0.0263</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>3.9512</td>
<td>1.9878</td>
<td>3.0561</td>
<td>1.338</td>
<td>0.0031</td>
</tr>
<tr>
<td></td>
<td>Itr. LLS</td>
<td>1.4009</td>
<td>1.1836</td>
<td>1.2608</td>
<td>1.4134</td>
<td>0.0191</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td><strong>0.0026</strong></td>
<td><strong>0.0510</strong></td>
<td><strong>0.0024</strong></td>
<td><strong>0.0468</strong></td>
<td><strong>0.0039</strong></td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.0028</td>
<td>0.0531</td>
<td>0.0025</td>
<td>0.0470</td>
<td>0.0041</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.0028</td>
<td>0.0529</td>
<td>0.0025</td>
<td>0.0469</td>
<td>0.1069</td>
</tr>
<tr>
<td>Data Sets</td>
<td>Technique</td>
<td>MSE</td>
<td>RMSE</td>
<td>MSEREG</td>
<td>MAE</td>
<td>Time(s)</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>Iris</td>
<td>BPCA</td>
<td>0.1590</td>
<td>0.3987</td>
<td>0.1431</td>
<td>0.3464</td>
<td>0.0453</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.1591</td>
<td>0.3988</td>
<td>0.1432</td>
<td>0.3466</td>
<td>0.0207</td>
</tr>
<tr>
<td></td>
<td>It. LLS</td>
<td>0.1590</td>
<td>0.3987</td>
<td>0.1431</td>
<td>0.3464</td>
<td>0.0402</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.1587</td>
<td>0.3984</td>
<td>0.1429</td>
<td>0.3464</td>
<td>0.003</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.1591</td>
<td>0.3989</td>
<td>0.1432</td>
<td>0.3470</td>
<td>0.0034</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.1589</td>
<td>0.3986</td>
<td>0.1430</td>
<td>0.3465</td>
<td>0.0467</td>
</tr>
<tr>
<td>Glass</td>
<td>BPCA</td>
<td>0.0618</td>
<td>0.2486</td>
<td>0.0557</td>
<td>0.1127</td>
<td>1.0795</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.0612</td>
<td>0.2474</td>
<td>0.0551</td>
<td>0.1124</td>
<td>0.0041</td>
</tr>
<tr>
<td></td>
<td>It. LLS</td>
<td>0.0617</td>
<td>0.2484</td>
<td>0.0555</td>
<td>0.1126</td>
<td>0.1512</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.0611</td>
<td>0.2473</td>
<td>0.0550</td>
<td>0.1118</td>
<td>0.2192</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.0612</td>
<td>0.2474</td>
<td>0.0550</td>
<td>0.1120</td>
<td>0.2243</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.0617</td>
<td>0.2484</td>
<td>0.0555</td>
<td>0.1126</td>
<td>0.6941</td>
</tr>
<tr>
<td>E-Coli</td>
<td>BPCA</td>
<td>0.0028</td>
<td>0.0529</td>
<td>0.0025</td>
<td>0.0499</td>
<td>0.1151</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.0027</td>
<td>0.0523</td>
<td>0.0024</td>
<td>0.0499</td>
<td>0.0033</td>
</tr>
<tr>
<td></td>
<td>It. LLS</td>
<td>0.0027</td>
<td>0.0523</td>
<td>0.0024</td>
<td>0.0499</td>
<td>0.0748</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.0027</td>
<td>0.0523</td>
<td>0.0024</td>
<td>0.0499</td>
<td>0.0008</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.0027</td>
<td>0.0523</td>
<td>0.0024</td>
<td>0.0499</td>
<td>0.0007</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.0029</td>
<td>0.0539</td>
<td>0.0026</td>
<td>0.0500</td>
<td>0.0746</td>
</tr>
<tr>
<td>Wine</td>
<td>BPCA</td>
<td>0.0512</td>
<td>0.2264</td>
<td>0.0461</td>
<td>0.0690</td>
<td>0.0524</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.0512</td>
<td>0.2263</td>
<td>0.0461</td>
<td>0.0692</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>It. LLS</td>
<td>0.0513</td>
<td>0.2265</td>
<td>0.0461</td>
<td>0.0691</td>
<td>0.0992</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.0511</td>
<td>0.2261</td>
<td>0.0460</td>
<td>0.0688</td>
<td>0.003</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.0511</td>
<td>0.2262</td>
<td>0.0460</td>
<td>0.0688</td>
<td>0.0026</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.0513</td>
<td>0.2265</td>
<td>0.0461</td>
<td>0.0691</td>
<td>0.0806</td>
</tr>
<tr>
<td>Balance</td>
<td>BPCA</td>
<td>0.1099</td>
<td>0.3315</td>
<td>0.0989</td>
<td>0.30</td>
<td>0.077</td>
</tr>
<tr>
<td>Scale</td>
<td>LLS</td>
<td>0.1098</td>
<td>0.3314</td>
<td>0.0988</td>
<td>0.2998</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>It. LLS</td>
<td>0.1098</td>
<td>0.3314</td>
<td>0.0988</td>
<td>0.2998</td>
<td>0.1782</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.1097</td>
<td>0.3312</td>
<td>0.0987</td>
<td>0.2995</td>
<td>0.0007</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.1098</td>
<td>0.3314</td>
<td>0.0988</td>
<td>0.2998</td>
<td>0.0008</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.1099</td>
<td>0.3315</td>
<td>0.0989</td>
<td>0.30</td>
<td>0.033</td>
</tr>
<tr>
<td>Lenses</td>
<td>BPCA</td>
<td>0.0352</td>
<td>0.1876</td>
<td>0.0317</td>
<td>0.1744</td>
<td>0.032</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.0355</td>
<td>0.1884</td>
<td>0.0319</td>
<td>0.1753</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>It. LLS</td>
<td>0.0349</td>
<td>0.1870</td>
<td>0.0314</td>
<td>0.1739</td>
<td>0.0211</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.0347</td>
<td>0.1865</td>
<td>0.0313</td>
<td>0.1729</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.0353</td>
<td>0.1879</td>
<td>0.1879</td>
<td>0.1747</td>
<td>0.0027</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.0352</td>
<td>0.1878</td>
<td>0.0317</td>
<td>0.1745</td>
<td>0.076</td>
</tr>
<tr>
<td>Pentagon</td>
<td>BPCA</td>
<td>0.0025</td>
<td>0.0503</td>
<td>0.0022</td>
<td>0.0445</td>
<td>0.0297</td>
</tr>
<tr>
<td></td>
<td>LLS</td>
<td>0.0025</td>
<td>0.0509</td>
<td>0.0023</td>
<td>0.0454</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>It. LLS</td>
<td>2.5253</td>
<td>1.5891</td>
<td>2.2727</td>
<td>2.5253</td>
<td>0.0073</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.0025</td>
<td>0.0509</td>
<td>0.0023</td>
<td>0.0454</td>
<td>0.0023</td>
</tr>
<tr>
<td></td>
<td>Wt. KNN</td>
<td>0.0025</td>
<td>0.0509</td>
<td>0.0023</td>
<td>0.0454</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>Reg. EM</td>
<td>0.0025</td>
<td>0.0503</td>
<td>0.0022</td>
<td>0.0446</td>
<td>0.0753</td>
</tr>
</tbody>
</table>
4.4 Chapter Summary

This chapter discusses the experimental results of data normalization and imputation techniques used for data preprocessing. Though all the techniques have their own merits and demerits, the assessment proposes few techniques for data preprocessing that best suits the considered binary and multiclass datasets in the classification framework. For data normalization, decimal scaling shows better results. For in the case of data imputation, KNN outperforms the other techniques.