CHAPTER -5
CHAPTER -5

IMPROVISED PROPHECY USING REGULARIZATION METHOD

5.1 INTRODUCTION

Patients with TD boast continuously increasing because of excessive growth of thyroid gland and its hormones. Automatic classification tools may reduce the burden on doctors. This module evaluates the selected algorithms for predicting TDD. The algorithms considered here are regularization methods (RM) of MLA. The analysis report generated by the proposed work suggests the best algorithm for predicting the exact levels of TDD. This work is a comparative study of MLA on UCI thyroid datasets (UCITD). The developed system deals with RM i.e., ridge regression algorithm (RRA) & least absolute shrinkage and selection operator algorithm (LASSO). The above algorithms personage produces at most 79% accuracy by RRA and 98.99% accuracy by LASSO. This module shows the importance of LASSO along with an example for parameter generation. The decisive factors (DF) also suggest the accuracy rate of LASSO is much better when compared with RRA.

5.2 PROPOSED METHODOLOGY

The work is done by capturing the attention of MLA i.e., RM (Martin, 2014) and SMS. The system developed is used for offline analysis on TDD. Originally the system is developed by using a SMS. When the SMS fails to identify the relevant data in the knowledge, then automatically the system invokes to get the optimistic disease by using two individual methods of RM which belongs to MLA namely (i) RRA (ii) LASSO . The method of RRA on TDD produced a result of maximum accuracy 79%, where as the LASSO on TDD produced 98.99%. As per the comparisons made by DF, it is understood that LASSO performance is best in prophecy. Hence, the system is further enhanced with DF values namely meticulousness (mT), exactitude (eX), compassion (cP) and rigour (rI) in order to provide the disease name with its prevention and curing methods. Here, we obtain the diagnostic methods for different symptoms entered by the user dynamically. If the data entered by the user is sufficient and if it matches the
knowledge, then the proposed system displays the actual disease with which the human is suffering, or else it displays the dialogue box stating that the knowledge is insufficient. For the purpose of calculating missing attribute values (Kluwer et al., 2012), RM is introduced. The nearby related diseases of TD can be determined based on the raised values. If the exact disease is unavailable in the knowledge of the expert, then the proposed system identically shows the probabilistic disease with which the human is suffering from. The LASSO method provides the actual disease by considering the values generated by the function LASSO_K. If the raised value satisfies the RM then LASSO (Bezdek, 1981) is responsible for providing the optimized disease which could be predicted.

5.3 REGULARIZATION METHOD

RM is used to solve problems by adding on data when necessary, this RM is used to solve any kind of structured data oriented problems related to multiple datasets. In this approach using RM, the limitations are provided for obtaining accuracy of the classifier.

The novelty which you can observe in this proposed RM algorithms is, these are used to calculate the missing values. Since, it is already informed that the data of this thyroid is inconsistent and vague. As per the representation, the pseudo code which is implemented can show you the exact working style of RRA in RM. The incessant values which are obtained are used for prophecy of TDD. The method of obtaining the continuous numerical values is mentioned in the pseudo codes of Section 5.3.1 & 5.3.2.

5.3.1 RIDGE REGRESSION ALGORITHM

RRA used for analyzing data which consists of multi co-linearity (Adam and Justin, 2014). When this multi co linearity occurs then the restrictions are automatically avoided. RRA reduces standard errors. But, the net effect of the solutions is reliable.

Pseudo code of the RRA developed:

Step 1: Input vector, $X = (S_1, S_2, \ldots, S_p)$, where ‘p’ value ranges from 1 to 28 considering the pristine attribute values, which is the count of symptoms. Let $\{S_j | j = 1, 2, \ldots, J\}$ be a set of $J$ samples. The outcome of each sample $S_j$ is denoted by the trait $t_j$. 
**Step 2:**

Output $Y$ is real-valued. For the problem defined above, the $I$-dimensional binary vector $C = \{b_1, b_2, \ldots, b_I\}$ where each bit indicates (‘1’ and ‘0’) based on the availability in the algorithm.

**Step 3:**

Predict $Y$ from $X$ by $f(X)$ so that the expected loss function $E(L(Y, f(X)))$ is minimized. To define the objective function, we use RRA to feature subset selection (Bin and Steve, 2012). Through regression coefficients by forcing a penalty on the size of subset. RRA is able to smoothly approach the solution with less variance compared with forward and backward stepwise selection methods.

Give the trait $t$ and $C$ leading to a subset of selected attributes.

$$X = \{x_{k1}, x_{k2}, x_{k3}, \ldots, x_{kn}\},$$

we fit the following pair wise interaction model as $\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots E(1)$. From $E(1)$, we can obtain selected attributes as displayed in $E(2)$, i.e.

$$t(C) = \beta_0 + \sum_{i=1}^{n} \beta_i x_{ki} \sum_{u=1}^{n-1} \sum_{v > u} \beta_{uv} x_{ku} x_{kv},$$

$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots E(2).$

**Step 4:**

Square loss: $L(Y, f(X)) = (Y - f(X))^2$. The optimal predictor $f^*(X) = \text{argmin}_{f(X)}E(Y - f(X))^2$. Then the ridge regression is to compute the coefficient set by minimizing the penalized residual sum of squares. Then the RRA (Seunghak and Eric, 2014) is to compute the coefficient set $\hat{\beta} = \{\beta_0, \beta_1, \beta_2, \ldots, \beta_n\}$ by minimizing the penalized residual sum of squares obtained from trait $t$ from $E(2)$.

Therefore,
\[ \hat{\beta} = \arg \min \beta \left\{ \sum_{j=1}^{J} \left( t_j - \beta_0 - n \sum_{i=1}^{n} \beta_i x_{ki} - \sum_{u=1}^{n-1} \sum_{v>u} \beta_{uv} x_{ku} x_{kv} \right)^2 \right\} + \lambda \left( \sum_{i=1}^{n} \beta_i^2 + \sum_{u=1}^{n} \sum_{v>u} \beta_{uv}^2 \right) \]

………………………….. Eq (3).

Where complexity parameter is denoted as \( \lambda \) in Eq (3).

Then, the objective function is defined as a multiple \( R^2 \) value, which is a decreasing function of the residual sum of squares from obtained Eq(3)

\[ R^2(c) = 1 - \frac{\sum_{j=1}^{J} \left( t_j - \hat{\beta}_0 - \sum_{i=1}^{n} \hat{\beta}_i x_{ki} - \sum_{u=1}^{n-1} \sum_{v>u} \hat{\beta}_{uv} x_{ku} x_{kv} \right)^2 + \Delta}{\sum_{j=1}^{J} (t_j - \bar{t})^2} \]

…………………………..Eq (4).

Where \( f^*(X) = E(Y \mid X) \) and “ \( \Delta \)” values in Eq(4) is defined as

\[ \Delta = \lambda \left( \sum_{i=1}^{n} \beta_i^2 + \sum_{u=1}^{n-1} \sum_{v>u} \beta_{uv}^2 \right) \]

………………………………………………………Eq (5).

\( \bar{t} = \frac{1}{J} \sum_{j=1}^{J} t_j \) is the mean outcome and \( \sum_{j=1}^{J} (t_j - \bar{t})^2 \) is the total outcome variation. The complexity parameter \( \lambda \geq 0 \) controls the eradication in accuracy levels. In the case of ordinary least squares regression \((\lambda = 0)\), \( R^2 \) lies between 0 and 1; \( R^2 = 1 \) indicates perfect model fit.

**Step 5:**

The function \( E(Y \mid X) \) is the regression function. The expected loss function \( E(L(Y, f(X))) \), is the additional data which acts as the restriction to predict the actual TDD (Karl, 2014). Choice of the ridge complexity parameter \( \lambda \) is a variant of model selection used to choose a value. In general, the value chosen should be the one associated with the largest \( R^2 \) value.
5.3.2. **LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR**

Lasso is the tool and considered to be an algorithm of regularization method. The major source of lasso when compared with RRA, LASSO (Fan et al., 2013) could produce astounded outcomes. Here, in lasso the major functionalities or features for identification increases. The ability of this LASSO is to handle any number of attributes without decreasing its parameter estimate to zero. The LASSO algorithm automatically chooses the relevant features and rejects the others. The rejection of the features in RRA is completely plotted to zero but in LASSO the concept of non-zero is not considered. A standard error rate is measured and thus work flow manages in focusing of attributes.

**Pseudo code of the LASSO implemented:**

**Step 1:** Consider an example, the extension of \( Y \) (OUTPUT) is proportional to the input given \( X \) (INPUT), i.e., \( S_1, S_2 \ldots S_{12} \) and

\[
Y = f(X, k) = kX, \quad \text{................................. Eq(6).}
\]

Constitutes the model, where \( X \) is the independent variable.

**Step 2:** To estimate the output constant, \( k \), a series of \( n \) measurements with different outputs will produce a set of data, \( (X_i, Y_i), i=1, 2 \ldots n \), where \( Y_i \) is a measured extension (Dhurandhar and Alin, 2013)

**Step 3:** Each experimental observation had an error. We denoted this error with \( \epsilon \), and specified an empirical model for our observations,

\[
Y_i = kX_i + \epsilon_i, \quad \text{................................. Eq(7).}
\]

**Step 4:** There are several methods we used to estimate the unknown parameter \( k \). Noting, that the \( n \) equations in the \( m \) variables in our data comprise an over determined system with one unknown and \( n \) equations, we had choose to estimate \( k \) using least squares. The sum of squares to be minimized as

\[
S = \sum_{i=1}^{n} (Y_i - kX_i)^2. \quad \text{................................. Eq(8).}
\]

This is obtained from \( Eq(7) \)

**Step 5:** The least squares estimate of the output constant, \( k \), is given by

\[
k = \frac{\sum X_i Y_i}{\sum X_i^2}. \quad \text{................................. Eq(9).}
\]

The above generated least square estimate is the combination obtained from \( Eq(7) \) & \( Eq(8) \).
Step 6: The output constant $k$ is the required identical value produced by LASSO, thus $k$ value by Eq(9).

5.3.3 Example code for obtaining LASSO function using thyroid datasets:

Function Lasso=lasso (Z, Y, k values)

% Lasso Function of Z (centered, explanatory)
% Y is the response,
// Z → Initial attribute value #0 ; Y → Final attribute value #1; k → Predicted attribute value #? (varies)
% k values are the values where to compute [n, p]=size(Z);

   ZpY=Z'*Y;
   ZpZ=Z'*Z;
   m=length(k values);

Lasso =ones(p,m);

   for k =1:m
      Lasso_K (:,k)=(ZpZ+diag(k values(k)))/ZpY;
   end

k values=(0:.05:.5)

Computation of k values as

Table 5.1: Entry of content into Questionnaire form.

<table>
<thead>
<tr>
<th>TSH</th>
<th>TSH-Value</th>
<th>T3</th>
<th>T3-Value</th>
<th>T4</th>
<th>TT4-Value</th>
<th>T4U</th>
<th>T4U-Value</th>
<th>FTI</th>
<th>FTI-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.5176</td>
<td>1</td>
<td>1.4622</td>
<td>1</td>
<td>1.4183</td>
<td>1</td>
<td>1.3827</td>
<td>1</td>
<td>1.3532</td>
</tr>
<tr>
<td>1</td>
<td>0.4775</td>
<td>1</td>
<td>0.4234</td>
<td>1</td>
<td>0.3806</td>
<td>1</td>
<td>0.3459</td>
<td>1</td>
<td>0.3172</td>
</tr>
<tr>
<td>1</td>
<td>0.0678</td>
<td>1</td>
<td>0.0113</td>
<td>1</td>
<td>-0.0334</td>
<td>1</td>
<td>-0.0696</td>
<td>1</td>
<td>-0.0997</td>
</tr>
<tr>
<td>1</td>
<td>-0.1762</td>
<td>1</td>
<td>-0.2292</td>
<td>1</td>
<td>-0.2713</td>
<td>1</td>
<td>-0.3054</td>
<td>1</td>
<td>-0.3336</td>
</tr>
<tr>
<td>1</td>
<td>0.0500</td>
<td>1</td>
<td>0.1500</td>
<td>1</td>
<td>0.2500</td>
<td>1</td>
<td>0.3500</td>
<td>1</td>
<td>0.4500</td>
</tr>
</tbody>
</table>
This table 5.1 contains the entered data through the questionnaire form, when the TSH, T3, TT4, T4U, FTI, TBG values are enabled “1”, then we can enter the related values of data obtained through the test dynamically and from thus, we can calculate the k values as shown above.

\[
\text{Lasso (Z, Y, (0:.05:.5))}
\]

\%Formula gives for choice of \( k \): \( \text{norm(Yhat-Yc)^2} \)

\text{Solution} \quad \text{47.8636} \quad >> \quad \text{47.8636}/(13-5)

\text{Thus, the k values is obtained as} \quad 5.9829

\% estimates the variance \( \sigma^2 \) >> \( bk0^*bk0 \)

\text{Solution} \quad 2.6973 \quad >> \quad k=(4*5.9829)/2.6973. \text{ Thus, the k values is obtained as 8.8724}

Therefore, the ‘k’ value obtained here could be a value nearer to Boltzman Constant, for obtaining optimistic value nearer upper and lower approximations of data could be considered by having Boltzman Constant into the data for securing the data.

**5.4 DECISIVE FACTORS**

These factors help in identifying the best algorithm in RM for classification and also give the accuracy and correctness for prophecy towards TDD. The term cumulative gives the total count of existing elements in the entered input string which are successful smacks or loss to the database, whereas term individual gives the occurrence as of read at particular iteration.

\( mT: \text{Meticulousness} \) of a classifier is the percentage of the test set tuples that are correctly predicted.

\[
mT = \frac{(NTP + NTN)}{(NTP + NFP + NFN + NTN)}
\]

\( eX: \text{Exactitude} \) is also referred as true positive rate i.e. the proportion of positive tuples that are correctly identified.

\[
eX = \frac{NTP}{(NTP+NFN)}
\]

\( cP : \text{Compassion} \) is defined as the proportion of the true positives against all the positive results.

\[
cP = \frac{NTP}{(NTP+NFP)}
\]

\( rI: \text{Rigour} \) is the true negative rate that is the proportion of negative tuples that are correctly identified.

\[
rI = \frac{NTN}{(NTN+NFP)}
\]
5.4.1 Procedure for calculating DF

Example: Consider a single tuple database of 12 symptoms or attribute values for calculating DF

Table 5.2: Smacks on data.

<table>
<thead>
<tr>
<th>SYMPTOMS</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>S10</th>
<th>S11</th>
<th>S12</th>
<th>Diagnose</th>
<th>Prevention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Set-1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Thyroid is Negative</td>
<td>SVHC (or) Normal Condition.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYMPTOMS</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>S10</th>
<th>S11</th>
<th>S12</th>
<th>Diagnose</th>
<th>Prevention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Set-1</td>
<td>0</td>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>0</td>
<td>1*</td>
<td>0~</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Thyroid is Negative</td>
<td>SVHC (or) Normal Condition.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Entered Data</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>S10</th>
<th>S11</th>
<th>S12</th>
<th>Diagnose</th>
<th>Prevention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Set-1 (Already Existing Knowledge)</td>
<td>0</td>
<td>1</td>
<td>1*</td>
<td>0</td>
<td>1*</td>
<td>0~</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Thyroid is Negative</td>
<td>SVHC (or) Normal Condition.</td>
</tr>
</tbody>
</table>

This table 5.2 is a sample database which is picked up from the original database of UCITD, to display the diagnose method along with its prevention and referral source as a suggestion. The entered data is the choice given by the user, if the entered data directly smacks the knowledge, then the SMS system produces 100% output, otherwise the system modifies itself by sending the same data to the classifiers, whether the smacks of the data is successful or not, the values which are required by the DF are updated and maintained as shown below.

Therefore, the obtained DF for the above considered data is

\[
\text{NTP} = 02 \quad \text{NTN} = 01 \quad \text{NFP} = 03 \quad \text{NFN} = 06
\]

\[
\text{mT} = \frac{(02+01)}{(02+01+03+06)} = \frac{03}{12} = 0.25,
\]

\[
\text{eX} = \frac{02}{02+06} = \frac{02}{08} = 0.25
\]

\[
\text{cP} = \frac{02}{02+03} = \frac{02}{05} = 0.4 \quad \text{and}
\]

\[
\text{rl} = \frac{01}{01+03} = \frac{01}{04} = 0.25
\]
The procedure discussed above is calculated for single tuple knowledge, but in the beginning it is informed that the UCITD are using additional tuples of data respectively, hence the matching and neighbouring elements trace would give more apex values for the DF (Bandyopadhy et al., 2007).

5.5 MODULE DESIGN

Fig 5.1: User Interface diagram for the proposed system.
In this stage i.e., figure 5.1, the DF are obtained by the input string smacks or the loss of object values to the knowledge and based on the strength of DF’s i.e., DF > 85%, the best resultant algorithm is announced.

5.6 RESULTS AND DISCUSSION

When the input string is given to the SMS, it processes the entered string with the knowledge automatically and then the diagnose is suggested, otherwise the MLA will predict the TD. After predicting the TD levels the DF calculations are made. As per the apex value obtained by the DF, the best classifier would be traced out and the result obtained by the classifier is finally displayed as a suggestion.

---

**Fig 5.2:** Input screen to identify hyperthyroid disease
In Fig 5.2, the resultant output is as followed below and this screen shot shows the diagnosis of hyperthyroid according to the symptoms fed by the user, the prediction and the referral source SVI is suggested to the user for further proceedings.

**Resultant output for the input Screen:**

Report generated as: Goiter. | 3523

Referral source specified: SVI

In Fig 5.3, the resultant output is as follows and this screen shot shows the diagnosis of goiter according to the symptoms fed by the user, and as the range of the test results fed. The prediction
is made and the referral source STMW is suggested to the user for further proceedings of treatment.

**Resultant output for the input screen:**

Report generated as: Goiter.| 3523
Referral source specified: STMW

**CASE 1:**

SMS generates 100% output.
If symptoms in the knowledge base 100% matches, then the process of identifying the disease and providing the prevention or curing method is done through the knowledge.

**IF Symptoms**

S1=41, S2=F, S3=F, S4=F, S5=F, S6=F, S7=F, S8=F, S9=F, S10=F, S11=T, S12=F, S13=F, S14=F, S15=F, S16=F, S17=T, S18=0.2, S19=T, S20=3.8, S21=T, S22=253, S23=T, S24=1.24, S25=T, S26=204, S27=F, S28=0

**THEN**

Diagnose: Hyperthyroid.|2003
Referral Source is=STMW.

**CASE 2:**

RRA generates 79% & 98.99% output.

**IF Symptoms**

S1=41, S2=F, S3=T, S4=T, S5=F, S6=F, S7=F, S8=F, S9=F, S10=F, S11=T, S12=F, S13=F, S14=F, S15=T, S16=T, S17=T, S18=0.2, S19=T, S20=3.8, S21=T, S22=453, S23=T, S24=1.84, S25=T, S26=504, S27=F, S28=0

**THEN**

**SMS:** Relevant data not found.
Then the string is passed to RM for predicting TD and then the further process of DF are considered to tell which algorithm is the best.

**Output(s):**

/* These outputs depends upon the number of attributes which we are considering for prophecy*/

**Table 5.3: Performance of RRA for UCITD for a sample considered as CASE 1, 2 & 3.**

<table>
<thead>
<tr>
<th>CASE</th>
<th>RM</th>
<th>Meticulousness</th>
<th>Exactitude</th>
<th>Compassion</th>
<th>Rigour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Algorithms</td>
<td>UCITD</td>
<td>UCITD</td>
<td>UCITD</td>
<td>UCITD</td>
</tr>
<tr>
<td>Entry of Symptoms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASE-1</td>
<td>RRA</td>
<td>71.59</td>
<td>75.17</td>
<td>71.03</td>
<td>77.5</td>
</tr>
<tr>
<td></td>
<td>LASSO</td>
<td>95.94</td>
<td>97.71</td>
<td>90.34</td>
<td>98.93</td>
</tr>
<tr>
<td>CASE-2</td>
<td>RRA</td>
<td>74.42</td>
<td>73.47</td>
<td>76.86</td>
<td>79.2</td>
</tr>
<tr>
<td></td>
<td>LASSO</td>
<td>98.96</td>
<td>98.71</td>
<td>98.45</td>
<td>96.48</td>
</tr>
<tr>
<td>CASE-3</td>
<td>RRA</td>
<td>75.84</td>
<td>75.67</td>
<td>76.42</td>
<td>78.29</td>
</tr>
<tr>
<td></td>
<td>LASSO</td>
<td>95.63</td>
<td>98.41</td>
<td>96.44</td>
<td>95.67</td>
</tr>
</tbody>
</table>

The **Table 5.3** displays the decisive factor values obtained by regularization methods of UCITD. The tables also display the growth rate in LASSO compared with RRA.
Fig 5.4: This screen shot, compares the TDD obtained through several MLA developed using RRA, LASSO and the systems developed is working with result approximations up to 98.99% considering a 28 pre-eminent attributes in knowledge for identifying the TD. This is the comparison chart produced based on RRA & LASSO algorithms on evaluation methods of DF.

5.7 CONCLUSION

RM of MLA is considered for evaluating the performance in terms of DF in predicting TDD. The attributes for UCITD data are S1to S28 are crucial in deciding TD status, with the selected data. LASSO is found giving better results with all the features available. When the observations turned towards Table 5.3, it is observed that there is a growth rate of values in measure. As per the results obtained, it can be recommended that LASSO related to RM shows high performance rate compared with RRA. The efficiency in the parameters chose the best optimization algorithm which can lead to exact prophecy of the TDD. Successful implementation, i.e., (i) introduced all 28 pristine attributes along with its values for predicting TDD, (ii) comparison of two high flying algorithms belonging to same cadre of MLA are applied on common featured attributes of UCITD, which merely helped in obtaining predicting levels at 98.99% accuracy. Lastly,(iii) about 2096 tuples of data for predicting TD. The successful implementation & completion of this work can give young researchers the idea of how useful the “MLA” in prediction as well as its application in medical epoch.
This experimental study on thyroid using RM, has given good prophecy. As thyroid has large features which indirectly increase prophecy execution time as well as may lead to misclassification. MLA has proved to be a very useful pre-processing step to remove unwanted features and to resolve the missing data. Most importantly this experimental study showed more improved accuracy over the original data set with missed values.