Diagnosing Anaemia Disease using Partial Least Square Models and Support Vector Machines

Assist. Lect. Soran Husen Mohamad
Soran.abdulrahman@univsul.edu.iq

Assist. Prof.Dr.Mohammad Mahmood Faqe Hussein
Mohammad.faqe@univsul.edu.iq

Department of Statistics- College of Administration and Economy- University of Sulaimany

Abstract

A severe worldwide health concern that impacts millions of individuals is anaemia, which is characterized as a deficiency of red blood cells or haemoglobin. A precise and prompt diagnosis is necessary to treat and manage the illness effectively. The usefulness of two machine learning methods for diagnosing anaemia is investigated in this work: support vector machines and partial least squares models. The study is based on a large dataset of 220 patients that includes information on blood pressure, white blood cell count, haemoglobin level, hematocrit test results, gestational age, age, education level, and iron storage levels in addition to the presence of chronic disease and physical exhaustion index. Initially, Partial Least Square (PLS), which is commonly used in prognostic modelling, is applied to the dataset to identify significant forecasters of anaemia. PLS regression is then used to develop a predictive model capable of accurately diagnosing anaemia founded on these features. Subsequently, Support Vector Machines (SVM), a powerful supervised learning algorithm, are employed to classify anaemia cases. The SVMs are trained on the same dataset, and their act is evaluated in terms of precision, sensitivity, and specificity. The comparative analysis highlights the strengths and weaknesses of both PLS models and SVMs in diagnosing anaemia. While PLS models demonstrate strong predictive capabilities and interpretability, the variables in the study can be ranked by importance as follows: chronic illness, body mass index, blood pressure, education level, ferritin levels in the serum, gestational age, mean corpuscular haemoglobin, white blood cell count, and hematocrit. These results offer important new information to medical professionals, enabling them to make better decisions and enhance patient care plans for anemic patients.

Keywords: Partial Least Square, Support Vector Machine, Principal Component Analysis, Coefficient of determination, and MSE.

Recieved: 2/9/2024
Accepted: 26/9/2024

1 Introduction

Regression analysis uses two prominent methods to predict the connection between the dependent variable and one or more independent variables: Partial Least Squares and Support Vector Machine regression. These procedures remain frequently used in many different domains, such as chemometrics, machine learning, and statistics. They are predicated on distinct ideas and presumptions even if they have the same goal.[4][5][8]

SVM regression is a supervised learning algorithm that aims to minimize the difference between expected and actual values by finding the optimal hyperplane in a high-dimensional space for grouping data points into discrete categories. The fundamental idea behind this method is to calculate a hyperplane that maximizes the margin between support vectors, or the data points nearest to the decision boundary. Since SVM regression does not rely on distributional assumptions like standard regression methods do, it is a good strategy for controlling nonlinear relationships between variables.[3]

On the other hand, PLSR is a statistical way that uses the identification of latent variables, or components, to optimize the variance in both the dependent variable and the predictors in order to describe the association between the dependent variable and some independent variables. The existence of a linear association between the predictor and responder variables is a fundamental tenet of PLS. By removing orthogonal components, it effectively manages multicollinearity among predictors.[1][3][12]

SVM regression and PLS differ primarily in their underlying theories and approaches to optimization. PLS seeks to optimize the covariance between predictors and the response variable, whereas SVM regression places more emphasis on optimizing the margin between support vectors. In addition, SVM regression works especially well in high-dimensional spaces and is resistant to overfitting, while PLS regression is easier to understand and better suited for scenarios in which there are more predictors than observations.[5][7]

To compare their performance, consider a simulation where both SVM regression and PLS are applied to a dataset with complex nonlinear relationships. SVM regression might deliver superior predictive accuracy, particularly in high-dimensional contexts, while PLS could provide deeper insights into the data's underlying structure, making it a better choice when interpretability is crucial.[9]

Regarding their advantages, SVM regression is highly capable of handling high-dimensional data and is resilient to overfitting. In contrast, PLS offers greater interpretability and effectively manages multicollinearity. However, SVM regression can be computationally demanding, especially with large datasets, and selecting the appropriate kernel function can be challenging.[6]

PLS's performance is highly responsive to the number of components chosen, and it runs the danger of overfitting when there are more predictors than explanations.[6]

2 The goals of this research

This study aims to compare the effectiveness of partial least squares and support vector machines, two machine learning techniques, in diagnosing anaemia.

3 Methodology

Partial Least Squares 3.1

PLS uses latent variables to maximize the association between responses and predictions. In the late 1960s, World devised it, first for econometrics. PLSR is a widely used multivariate modelling technique for Near-Infrared spectral data when creating calibrations. It combines MLR and Principal Component Analysis. The model considers the spectrum data represented by the variable matrix X as well as the properties of interest represented by the variable matrix Y. The nutrients in sugarcane leaves are estimated by the latent variables that come from the PLSR model. Next, it builds a regression model to predict response variables simultaneously, Y. the capacity to distinguish between data and noise in the system being studied. This information feature

compression yields more practically significant results by condensing the explanatory variable X and accounting for its association with the predicted variable Y. Concurrently derives the first latent variable, u1, from the variable set Y, guaranteeing maximum correlation between t_1 and u_1, and extracts the first latent variable, t_1, from the variable set X, capturing maximum variation information. Regression equations with Y and t_1 and with X and t1 are then established. The algorithm extracts the second latent variable, t_2, from the residual information interpreted by t_1 of X, and u_2 from the residual information interpreted by t_1 of Y, until it reaches the appropriate precision, or iterates until the accuracy requirements are met. The PLSR model integrates principal components analysis, canonical correlation analysis, and LRM within the modelling process, enhancing its effectiveness in predictive modelling and analysis of NIR spectral data.[9]

3.2 Mathematical applied of PLS

A brief overview of PLS mathematics is given in this section. PLS is essentially a dimension reduction strategy combined with a regression model. In contrast to comparable methods like PCR, the latent components derived by PLS are selected while considering the regression's dependent variable. Let's say we wish to use p continuous predictor variables $z_1,\ldots,z_p z_1,\ldots,z_p$ to predict q continuous dependent variables $Y_1,\ldots,Y_q Y_1,\ldots,Y_q.$ $(z_i',y_i')_i = 1,2,\ldots.,n (z_i',y_i')_i = 1,2,\ldots.,n$ Represents the presented data sample with n observations, which $z_i' \& y_i' z_i' \& y_i'$ stand for the ith observation of the dependent variable and the predictor variable, correspondingly. The prime represents un-centred fundamental data, i.e. Their elimination denotes the sample average's subtraction, i.e.[12]

$$z_i = z_i' - \frac{\sum_{s=1}^{n} z_s'}{n}$$

$$y_i = y_i' - \frac{\sum_{s=1}^{n} y_s'}{n}$$

The $z_i = (z_{i1}, z_{i2}, \ldots, z_{ip})'_{n\times p} z_i = (z_{i1}, z_{i2}, \ldots, z_{ip})'_{n\times p}$ of matrix Z. Likewise, $Y_{n\times p} Y_{n\times p}$ containing the $y_i = (y_{i1}, y_{i2}, \ldots, y_{ip})' y_i = (y_{i1}, y_{i2}, \ldots, y_{ip})'$

$$Z = \begin{bmatrix} z_1' \\ \vdots \\ z_n' \end{bmatrix} \quad \text{And} \quad Y = \begin{bmatrix} y_1' \\ \vdots \\ y_n' \end{bmatrix}$$

When $n < pn < p$, the LRM, which is often represented as OLS, cannot be applied $Cov(Z'Z)$ $Cov(Z'Z)$ (which can have a maximum rank $n-1n-1$) is singular. In contrast, PLS may be applied also to cases in which $n < pn < p$. PLS regression is based on the basic latent component decomposition:

$$Y = TQ' + FY = TQ' + F \qquad (1)$$

$$Y = TP' + EY = TP' + E \qquad (2)$$

Where $T_{n\times c} T_{n\times c}$ is a matrix giving the latent components for the n observations, $P_{p\times c} P_{p\times c}$ and $Q_{q\times c}$ $Q_{q\times c}$ are matrices of coefficients and $E_{n\times p} E_{n\times p}$ and $F_{n\times q} F_{n\times q}$ are matrices of random errors. Note that if the given matrices T, P and Q satisfy Equations (1) and (2), then so do $T^* = TM, P^* = P(M^{-1})'$

$T^* = TM, P^* = P(M^{-1})'$ and $Q^* = Q \times (M^{-1})' Q^* = Q \times (M^{-1})'$ for any non-singular $c \times cc \times c$ matrix M. Thus, the space spanned by the columns of T is more important than the columns of T themselves.

PLS as well as principal component regression and reduced rank regression can all be seen as methods to construct a matrix of latent components T as a linear transformation of ZZ:

$$T = Z \times W$$

Where $W_{p\times c} W_{p\times c}$ matrix of weights. In the remainder of the article, the columns of W and T are denoted as $w_i = (w_{1i}, \ldots, w_{pi})' w_i = (w_{1i}, \ldots, w_{pi})'$ and $t_i = (t_{1i}, \ldots, t_{ni})' t_i = (t_{1i}, \ldots, t_{ni})'$, respectively, for $i = 1,2,3, \ldots . ci = 1,2,3, \ldots . c$. For a fixed matrix W, the random variables obtained by forming the corresponding linear transformations $Z_1, \ldots, Z_p Z_1, \ldots, Z_p$ are denoted as $T_1, \ldots, T_c$ $T_1, \ldots, T_c$:

$$T_1 = W_{11} Z_1 + \cdots + W_{p1} Z_p,$$
$$\ldots . =$$
$$T_c = W_{1c} Z_1 + \cdots + W_{pc} Z_p$$

The latent components are then used for prediction in place of the original variables: once T is constructed, $Q^T Q^T$ is obtained as the least squares solution of Equation (1):

$$Q^T = (T'T)'T'Y$$

Finally, the matrix B of regression coefficients for the model $Y = XB + FY = XB + F$ is given as

$$B = W \times Q' = W \times (T'T)'T'Y$$

The fitted response matrix $\widehat{YY}$ may be written as

$$\widehat{Y} = T \times (T'T)'T'Y$$

If we have a new (uncentered) raw observation $x_0' x_0'$, the prediction $\hat{y}_0' \hat{y}_0'$ of the response is given by

$$\hat{y}_0' = \frac{1}{n} \sum_{i=1}^{n} y_i' + B' \left(z_0 - \frac{1}{n} \sum_{i=1}^{n} z_i' \right) \hat{y}_0' = \frac{1}{n} \sum_{i=1}^{n} y_i' + B' \left(z_0 - \frac{1}{n} \sum_{i=1}^{n} z_i' \right) \quad (3)$$

PLS outputs the matrices W, T, P, and Q in addition to the matrix of regression coefficients B; consequently, PLS regression refers to the simultaneous performance of dimension reduction and regression in PLS. The columns of T are frequently referred to as “latent variables” or “scores” in the PLS literature. Since the columns of T in PLS are not observations of underlying random variables, but rather the outcome of a matrix decomposition, we prefer the term “latent components” for this study. ‘Z-loadings’ and ‘Y-loadings’ are common designations for P and Q, respectively.[11][12]

The response Y should be considered while building the components T, according to the fundamental principle of the PLS approach. More specifically, as described in the sections on “Univariate response” and “Multivariate response,” the components are defined so as to have significant covariance with the response. Because of this, PLS is referred to as a supervised approach as opposed to other methods like principal component analysis (PCA), which does not use the response to create new components. This characteristic explains why PLS typically outperforms PCA in prediction tasks. There are four levels at which the different PLS regression approaches can be characterized: The W matrix itself, the result matrix of the coefficients of regression $\beta\beta$,

the objective function that the W matrix maximizes, and the procedure used to compute W. The connections between these four distinct levels are as follows:

- Multiple objective functions can be maximized by the same W matrix. However, only one W-Matrix (and its opposite–W) often satisfies a given objective function.
- The W matrix may be produced via multiple algorithms.
- There is just one possible matrix of regression coefficients for a given W matrix. It is possible for two distinct matrices, W and W^*W^*, to provide identical regression coefficients, provided that an invertible matrix $M_{c\times c}M_{c\times c}$ exists, so that $W^* = W \times MW^* = W \times M$. Keep in mind that while W and W^*W^* produce the same forecast, they may not always meet the same goal function.

3.3 Support Vector Machine Regression

SVM is commonly used for regression and classification in machine learning. Vapnik introduced the SVM, which is a useful tool for pattern recognition and classification problems. Regression problems can be solved using SVMs by using a different loss function. SVM has drawn interest and seen widespread use as a result of its benefits and exceptional generalization performance over alternative approaches. Because SVM contains the structural risk minimization principle—which has been demonstrated to be superior to the conventional empirical risk minimization principle—it can provide global models that are frequently unique, resulting in excellent performance. Moreover, sparse solutions can be obtained and both linear and nonlinear regression can be carried out because of their particular formulation. However, because it necessitates the solution of several non-linear equations, determining the final SVM model can be quite computationally challenging.[2][11]

3.4 Mathematical Model of SVMR

This is a brief synopsis of SVM theory for regression. The fundamental idea behind SVMR is to solve a linear regression issue in higher dimensional feature space by mapping the novel data (z) nonlinearly. To learn the input-output connection from the data set $V = \{(z_i, y_i)\}_i^p V = \{(z_i, y_i)\}_i^p$, we first regress it using a linear function. Here, $z_i z_i$ represents the input vector to the SVR model, $y_i y_i$ is the output value, and p is the total number of data patterns. The following is an expression for the SVMR model:

$$f(z) = b + w.\varphi(z) f(z) = b + w.\varphi(z) \qquad (4)$$

where the high dimensional kernel-induced feature space is represented by the function $\varphi(z)\varphi(z)$. The parameters w and b, which represent a support vector weight vector and a bias term, are determined by minimizing the regularized risk function as follows:

$$R(c) = \frac{\|w\|^2}{2} + \frac{c}{p}\sum_{i=1}^{P}\rho_\varepsilon(y_i, f(z_i))R(c) = \frac{\|w\|^2}{2} + \frac{c}{p}\sum_{i=1}^{P}\rho_\varepsilon(y_i, f(z_i)) \qquad (5)$$

where **CC** represents a cost function measuring the empirical risk. $\frac{\|w\|^2}{2}\frac{\|w\|^2}{2}$ Denotes the regularization term. $\rho_\varepsilon(y_i, f(z_i))\rho_\varepsilon(y_i, f(z_i))$ is the so-called $\varepsilon\varepsilon$-insensitive loss function, which is defined as

$$\rho_\varepsilon(y_i, f(z_i)) = \begin{bmatrix} |y_i - f(z_i)| - \varepsilon, & |y_i - f(z_i)| \geq \varepsilon \\ 0, & \text{Otherwise} \end{bmatrix}$$

$$\rho_\varepsilon(y_i, f(z_i)) = \begin{bmatrix} |y_i - f(z_i)| - \varepsilon, & |y_i - f(z_i)| \geq \varepsilon \\ 0, & \text{Otherwise} \end{bmatrix} \qquad (6)$$

In Equation (6), if the predicting error is less than ε, the loss equals 0; if not, the loss equals a value greater

than ε.

The deviation $(q_i, f(z_i))(q_i, f(z_i))$ from the edges of the ε-insensitive zone can be calculated using two positive slack variables, $\xi_i\xi_i$ and $\xi_i^*\xi_i^*$, for $, i = 1,2,3, \ldots., n, i = 1,2,3, \ldots., n$. Stated differently, they express the separation between the real values and the corresponding ε-insensitive zone boundary values. Eq. (5) is converted into the following constrained version by utilizing slack variables:

$$\text{Minimize}: \ \tau(w, \xi_i, |\ \xi_i^*) = c\sum_{i=1}^{P}(\xi_i + \xi_i^*) + \frac{\|w\|^2}{2}$$

$$\text{Minimize}: \ \tau(w, \xi_i, |\ \xi_i^*) = c\sum_{i=1}^{P}(\xi_i + \xi_i^*) + \frac{\|w\|^2}{2} \quad (7)$$

$$\text{subject to} \begin{bmatrix} -b + y_i - [w \times \varphi(z)] \le (\xi_i + \varepsilon) \\ -b + y_i - [w \times \varphi(z)] \le (\xi_i^* + \varepsilon) \\ \xi_i \ge 0\ , \xi_i^* \ge 0 \end{bmatrix}$$

$$\begin{bmatrix} -b + y_i - [w \times \varphi(z)] \le (\xi_i + \varepsilon) \\ -b + y_i - [w \times \varphi(z)] \le (\xi_i^* + \varepsilon) \\ \xi_i \ge 0, \xi_i^* \ge 0 \end{bmatrix}$$

By using Lagrangian multipliers and Karush-Kuhn-Tucker conditions to Eq. (7), it thus yields the following dual Lagrangian form

$$\text{Maximize}: \ H(\alpha, \alpha^*) = -\frac{1}{2}\sum_{i,j=1}^{P}(\alpha_i - \alpha_i^*)\left(\alpha_j - \alpha_j^*\right)$$

$$\text{Maximize}: \ H(\alpha, \alpha^*) = -\frac{1}{2}\sum_{i,j=1}^{P}(\alpha_i - \alpha_i^*)\left(\alpha_i - \alpha_j^*\right) \quad (8)$$

$$K(x_i, x_j) = \sum_{i=1}^{P} y_i(\alpha_i - \alpha_i^*) - \varepsilon\sum_{i=1}^{P}(\alpha_i - \alpha_i^*)$$

Subject to the constraints:

$$\sum_{i=1}^{P}(\alpha_i - \alpha_i^*) = 0\ , \alpha_i\ , \alpha_i^*\ \in [1, c],\ i\ \in [1, P]$$

$$\sum_{i=1}^{P}(\alpha_i - \alpha_i^*) = 0\ , \alpha_i\ , \alpha_i^*\ \in [1, c],\ i\ \in [1, P] \quad (9)$$

The LP in Eq. (9) satisfies the equality $\alpha_i \times \alpha_i^* = 0\alpha_i \times \alpha_i^* = 0$. The LP, $\alpha_i\alpha_i$ and $\alpha_i^*\alpha_i^*$, are calculated and an optimal desired weight vector of the regression hyperplane is obtained by

$$w^* = \sum_{i=1}^{P}(\alpha_i - \alpha_i^*) \times K(z_i, z_j)w^* = \sum_{i=1}^{P}(\alpha_i - \alpha_i^*) \times K(z_i, z_j)$$

Therefore, the SVM-based regression function's generic form can be expressed as

$$f(z) = b + \sum_{i=1}^{P}(\alpha_i - \alpha_i^*) \times K(z_i, z_j)f(z) = b + \sum_{i=1}^{P}(\alpha_i - \alpha_i^*) \times K(z_i, z_j) \quad (10)$$

In Eq. (10), $K(z_i, z_j)K(z_i, z_j)$ the function of kernel. The idea of the function of the kernel $K(z_i, z_j) = \Phi(z_i) \times \Phi(z)K(z_i, z_j) = \Phi(z_i) \times \Phi(z)$has been introduced to decrease the computational demand. [7][10]

4 Result and Discussion

The Children's Hospital in the Sulaymaniyah Governorate provided the study's data, which were gathered from a total of 220 patients. Test and training sets of data were separated apart. Age, education level, gestational age,

presence of chronic disease, physical exhaustion index, blood pressure, white blood cell count, haemoglobin level in the blood, hematocrit test, and iron storage test were the explanatory variables that were investigated, while the haemoglobin ratio was the response variable. Separate the dataset into training and testing sets.

4-1 PLS Algorithm steps

Partial Least Squares regression is useful when you have many predictor variables, and these predictors are highly collinear.

Z dimension: 219 10 Data:

Y dimension: 219 1

Fit method: kernel pls

Number of components considered: 10

First: PLS regression is a robust multivariate analysis technique used to model the relationships between predictor variables and response variables. It is especially beneficial when there are more predictor variables than observations or when the predictors are highly collinear.

VALIDATION: RMSEP

Table (1): Represent Validation: RMSEP

Cross-validation using 10 random segments											
Measures	(Intercept)	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6	Comp 7	Comp 8	Comp 9	Comp 10
Cross Validation	1.54	0.8904	0.6055	0.4967	0.5026	0.5096	0.508	0.507	0.5078	0.5062	0.5063
Adj Cross Validation	1.54	0.7839	0.5855	0.4938	0.4995	0.5056	0.5041	0.5031	0.504	0.5025	0.5026

The test RMSE computed using k-fold cross-validation is displayed in the above table, which makes the following observations:

- With only the intercept term in the model, the test RMSE is 1.54.
- Adding the first PLS component reduces the test RMSE to 0.8904.
- Including the second PLS component further reduces the test RMSE to 0.6055.
- We observe that adding more PLS components increases test RMSE. Therefore, it seems optimal to use only three PLS components in the final model.

TRAINING: % variance explained

Table (2): Represent Training: % variance explained

Training: % variance explained										
Measures	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6	Comp 7	Comp 8	Comp 9	Comp10
X	10.26	76.35	95.89	98.84	99.18	99.5	99.59	99.8	99.88	100
y	87.71	89.19	91.36	91.41	91.54	91.59	91.61	91.62	91.62	91.62

The percentage of the response variable's variance that the PLS components account for is shown in this table. In particular:

- Of the variation in the response variable, 10.26% can be explained by the first PLS component alone.
- Upon adding the second PLS component, the explained variation rises to 76.35%.

Table (3): Measures of Criteria

Model	**Measure Criteria**	
	MSE	**R-Squared**
PLS	0.196758	0.916242

The Mean Squared Error (MSE) value of 0.196758 represents the average squared difference between the actual and predicted values. Whether this error is considered high or low depends on the data's scale, but generally, an MSE closer to 0 is preferred for practical applications.

The R^2 value of 91.62% indicates that about 91.62% of the variance in the response variable can be explained by the explanatory variables in the model. This high R^2 value suggests a strong fit of the model.

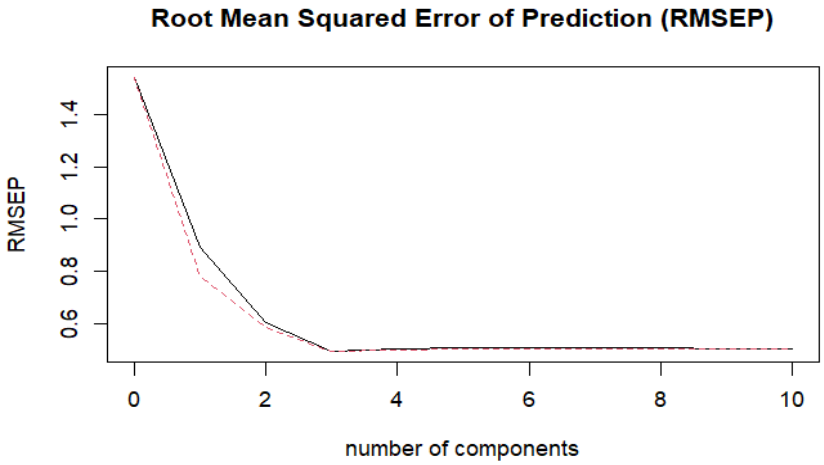

Figure (1) Represent root mean square error of prediction.

The statistics used to measure a predictive model's accuracy is the Root Mean Square Error of Prediction. It calculates the discrepancy between values that are observed and those that the model predicts. When evaluating model performance during cross-validation or when using a test dataset, RMSEP is quite helpful. The aforementioned figure suggests a good fit because it shows how well the model's predictions match the actual values.

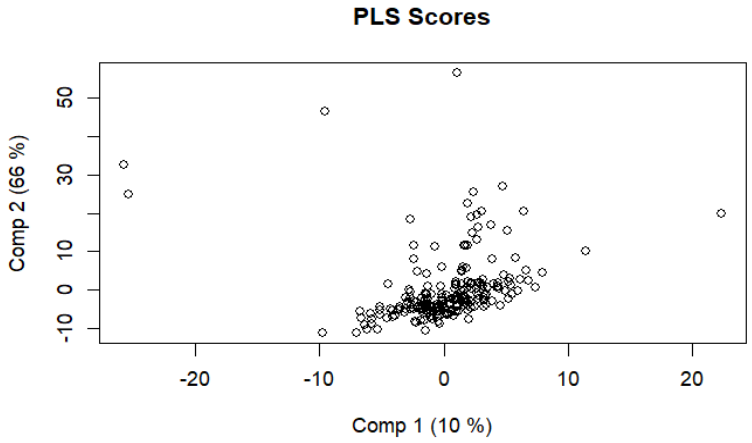

Figure (2) Represent the PLS scores.

Ten per cent of the variance in the predictor variables is explained by Component 1 (Comp1). This component's contribution is minimal, yet it might indicate some underlying structure in the data. Sixty-six per cent of the variance in the predictor variables is explained by Component 2 (Comp2). The most significant correlations between the predictors and the response variable are probably captured by this component, which is also much more significant.

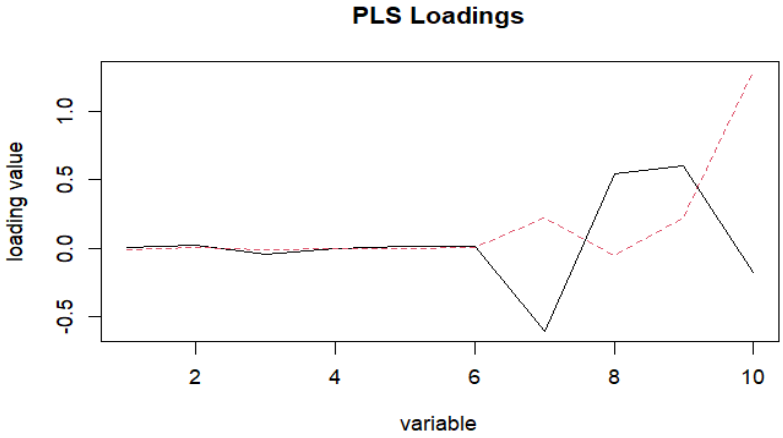

Figure (3) Represent the PLS loading

Table (4): Represent the ANOVA table

Sources	Degree of Freedom	Sum Square	Mean Square Error	F-Value	P-Value
Z_1	1	0.03	0.03	0.136	0.7123
Z^1	1	4.68	4.68	22.601	3.72E-06***
Z^2	1	29.60	29.6	142.868	< 2e-16***
Z^3	1	1.92	1.92	9.292	0.0026**
Z^4	1	18.21	18.21	87.918	< 2e-16***
Z^5	1	14.73	14.73	71.086	5.69E-15***
Z^6	1	69.65	69.65	336.229	< 2e-16***
Z^7	1	95.94	95.94	463.112	< 2e-16***
Z^8	1	234.53	234.53	1132.113	< 2e-16***
Z^9	1	2.07	2.07	9.997	0.0018**
Z^{10} Residual	208	43.09	0.21		
Total	218				

Since Z_1 is relatively high, there isn't much evidence to refute the null hypothesis. In particular, under typical significance thresholds like 0.05, the variable Z_1 (p-value of 0.7123 > 0.05) is not statistically significant. This suggests that, in comparison to the noise in the data, X1 does not substantially contribute to explaining the variance in the dependent variable. On the other hand, because their p-values are all less than 0.05, the remaining variables are all statistically significant.

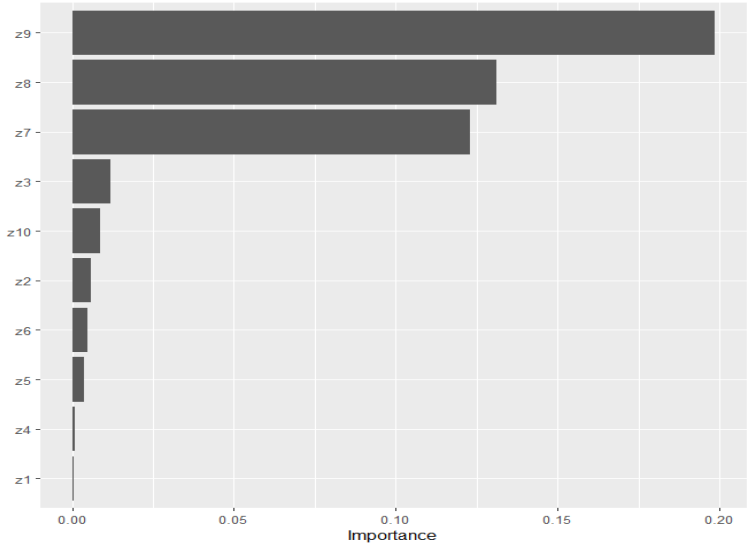

Figure (4) Represent the importance variable in PLS

From the above diagram, the influencing variables can be arranged as follows:

Z_4: Presence of Chronic Disease

Z_5: Body mass Index in kg/m2

Z_6: Blood Pressure

Z_2: Level of Education

Z_{10}: S.Ferritin

Z_3: Gestational Age

Z_7: WBC

Z_8: MCH [haemoglobin level in blood]

Z_9: HCT [Hematocrit Test]

Second: SVR is a regression problem-solving system that builds upon Support Vector Machine (SVM). Though SVR for classification relies on the same fundamental ideas as SVM, it adds some adjustments to make continuous value prediction easier.

Table (5): Comparison between different SVM-Kernel

SVR-Kernel	Number of SVR	R^2	MSE	Gamma	Epsilon
Polynomial	134	0.8273	0.3115	0.1	0.1
Radial	117	0.7877	0.3830	0.1	0.1

Polynomial Kernel Interpretation

- A good fit for the data is indicated by the model's coefficient of 0.8273, which shows that it accounts for 82.73% of the variance in the dependent variable.
- The average squared error of the model's predictions is equal to 0.3115. Better model performance is shown by a reduced mean squared error (MSE).

Radial Base Kernel Interpretation

- 0.7877: This indicates that, while marginally less than the previous number, the model still accounts for 78.77% of the variance, indicating a satisfactory fit.
- 0.3830: This MSE is greater than 0.3115, meaning that, on average, the model's predictions are less accurate than those of the model with an MSE of 0.3115.

In conclusion, a model with an R^2 of 0.8273 and an MSE of 0.3115 outperforms a model with an R^2 of 0.7877 and an MSE of 0.3830 in terms of

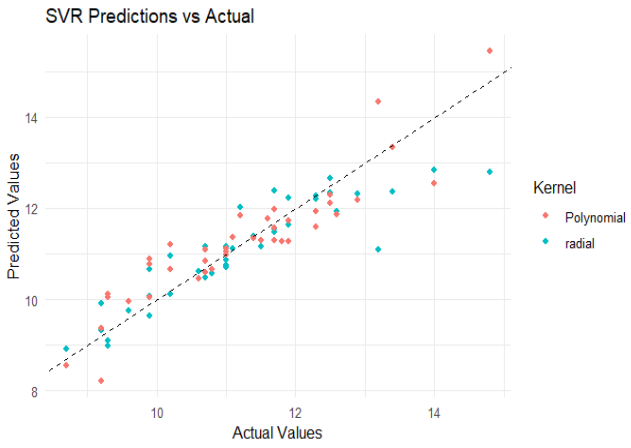

Figure (5) Represent the SVR predictions vs. actual

In summary, information on the SVR model's performance for each type of kernel may be obtained by looking at the scatter around the diagonal line. While polynomial kernels are excellent at modelling polynomial relationships, they may overfit if not properly adjusted. In contrast, radial kernels are good at handling non-linearities.

Third: Comparison between PLS and SVR Both methods have their strengths and can be powerful tools in regression analysis, dependent on the nature of the data and the specific requirements of the analysis. The result is as follows:

Table (6): Comparison between PLS and SVR

Model	MSE	R^2
Partial Least Square (PLS)	0.21	0.9162
Support Vector Regression Machine (SVR)	0.3115	0.8273

In this case, the Partial Least Squares (PLS) model outperforms the Support Vector Regression (SVR) model based on the MSE and R-Square values. With a reduced MSE and a greater root MSE, the PLS model exhibits superior prediction accuracy and explanatory power.

4 Conclusions and Recommendations

4-1 Conclusions

This is how the conclusion is paraphrased:

1. Investigating different kernel functions can increase the accuracy of classification; in biomedical contexts, radial basis functions frequently produce robust findings. Other kernel functions to

consider are linear, polynomial, and radial.

2. Partial Least Squares (PLS) is a useful technique for reducing dimensionality while maintaining important data required for the prediction of anaemia.

3. The PLS model performs better than the Support Vector Regression (SVR) model, as indicated by its Mean Squared Error (MSE) and R-Square values. A lower MSE and a larger R-Square indicate better prediction accuracy and explanatory power.

4. PLS provides important insights into the most important variables for predicting anaemia, such as the presence of chronic disease, body mass index (BMI), blood pressure, education level, S. ferritin, gestational age, WBC, MCH, and HCT.

4-2 Recommendations

To further improve the diagnostic process and model performance, consider the following recommendations:

1. To improve diagnostic performance, look into ensemble approaches that integrate PLS-SVM with additional machine-learning techniques.
2. Collaborate with medical experts to include the diagnostic model in clinical procedures, guaranteeing its applicability and influence in actual environments.
3. Provide procedures for the model's continual monitoring and updating to guarantee that its correctness is preserved when fresh patient data becomes available.

References

[1]Ahmed, D. H., Mohamad, S. H., & Karim, R. H. R. (2023). Using Single Exponential Smoothing Model and Grey Model to Forecast Corn Production in Iraq during the period (2022-2030). *University of Kirkuk Journal For Administrative and Economic Science*, *13*(3).

[2] Cortes, C., & Vapnik, V. (1995). "Support-Vector Networks." Machine Learning, 20(3), 273-297. This is the foundational paper on SVMs by Cortes and Vapnik.

[3] Hussein, M. M. F. (2024). Forecasting Price of Crude Oil Using the Weight Markov Chain (WMC) and ARIMA Model Techniques. Al-Ghary Journal of Economic and Administrative Sciences, 20(2), 42-64.

[4] Hussein, M. M. F., Saeed, A. A., & Mohamad, S. H. (2023). Comparison Markov Chain and Neural Network Models for forecasting Population growth data in Iraq. *University of Kirkuk Journal For Administrative and Economic Science*, *13*(4).

[5] Hussein, M. M. F. (2023). Predicting Number of People Living with Chronic HCV Using Gray-Weighted Markov Chain Model (GW-MCM). *University of Kirkuk Journal For Administrative and Economic Science*, *13*(4).

[6] Hamad, A. P. D. A. S., Faqe, A. P. D. M. M., & Mohamad, A. L. S. H. (2023). Forecasting Life-

Expectancy in Iraq During the Period (2022-2035) Using Fuzzy Markov Chain. *University of Fallujah, Journal of Business Economics for Applied Research*, *5*(3), 347-372.

[7] Liu, Y., & Li, Q. (2008). "A New Approach to Simultaneous Feature Selection and Support Vector Machine Classification." In: 2008 IEEE International Conference on Data Mining. This paper discusses integrating feature selection with SVM classification, relevant for combining with PLS.

[8] Omer, A., Faraj, S. M., & Mohamad, S. H. (2023). An application of two classification methods: hierarchical clustering and factor analysis to the plays PUBG. Iraqi Journal of Statistical Sciences, 20(1), 25-42.

[9] Rosipal, R., & Trejo, L.J. (2001). "Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space." Journal of Machine Learning Research, 2, 97-123. This paper explores the combination of PLS and SVM methodologies.

[10] Taha, A. A., & Mohammad, M. A. (2023). Correlated multistate model for the progression of chronic kidney disease with detecting risk factors effect. Revista Latinoamericana de Hipertension, 18(6).

[11] Vapnik, V. (1999). "An Overview of Statistical Learning Theory." IEEE Transactions on Neural Networks, 10(5), 988-999. This paper provides a broad overview of statistical learning theory, including SVMs.

[12] Wegelin, J.A. (2000). "A Survey of Partial Least Squares (PLS) Methods, with Emphasis on the Two-Block Case." Technical Report No. 371, University of Washington, Seattle. This survey paper covers various PLS methods, with a focus on the two-block case.

Appendix (A): Partial Least Square

```
install.packages("pls")
install.packages("caret")
library(pls)
library(caret)

# Dummy data
data =read.csv("C:/hp1.csv")
data
# Fit the PLS model
pls_model <- plsr(y ~ z1+z2+z3+z4+z5+z6+z7+z8+z9+z10, data = data, validation = "CV")

# Model summary
summary(pls_model)
```

```
# Calculate the MSE
predicted <- predict(pls_model, ncomp = pls_model$ncomp)
mse <- mean((data$y - predicted)^2)
print(paste("Mean Squared Error (MSE):", mse))

# Calculate R-Squared
r_squared <- 1 - sum((data$y - predicted)^2) / sum((data$y - mean(data$y))^2)
print(paste("R-Squared:", r_squared))
# Variable importance
vip <- vip(pls_model)
print(vip)
# Goodness of fit plot
plot(RMSEP(pls_model), main = "Root Mean Squared Error of Prediction (RMSEP)")

# Scores plot
plot(pls_model, plottype = "scores", main = "PLS Scores")

# Loadings plot
plot(pls_model, plottype = "loadings", main = "PLS Loadings")

# Variable importance plot
barplot(vip, main = "Variable Importance in Projection (vip)", col = "blue")

anova_model <- aov(y ~ z1+z2+z3+z4+z5+z6+z7+z8+z9+z10, data = data)
summary(anova_model)
```

Appendix (B): Support Vector Machines (SVM)

Support vector regression (RBF kernel, polynomial kernel,Y:response,Z_1:Independent,Z_2:independent ,Z_3:independent, MSE,R-Square ,variables significance, figure) use r programming

```
install.packages("e1071")
install.packages("caret")
install.packages("ggplot2")
library(e1071)
library(caret)
library(ggplot2)
```

```
data <- read.csv("C:/hp1.csv")
data
# Split data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(data$y, p = 0.8, list = FALSE)
trainIndex
trainData <- data[trainIndex,]
trainData
testData <- data[-trainIndex,]
testData
# Train SVR with RBF kernel
svr_rbf <- svm(y ~ ., data = trainData, kernel = "radial")
summary(svr_rbf)
# Predict on test data
predictions_rbf <- predict(svr_rbf, newdata = testData)
# Calculate MSE and R-squared for RBF kernel
mse_rbf <- mean((predictions_rbf - testData$y)^2)
rsq_rbf <- 1 - sum((predictions_rbf - testData$y)^2) / sum((mean(trainData$y) - testData$y)^2)
# Train SVR with polynomial kernel
svr_poly <- svm(y ~ ., data = trainData, kernel = "polynomial")
summary(svr_poly)
# Predict on test data
predictions_poly <- predict(svr_poly, newdata = testData)
# Calculate MSE and R-squared for polynomial kernel
mse_poly <- mean((predictions_poly - testData$y)^2)
rsq_poly <- 1 - sum((predictions_poly - testData$y)^2) / sum((mean(trainData$y) - testData$y)^2)
# Print results
cat("RBF Kernel:\n")
cat("MSE:", mse_rbf, "\n")
cat("R-squared:", rsq_rbf, "\n\n")
cat("Polynomial Kernel:\n")
cat("MSE:", mse_poly, "\n")
cat("R-squared:", rsq_poly, "\n")
# Combine predictions and actual values for plotting
plot_data_rbf <- data.frame(Actual = testData$y, Predicted = predictions_rbf, Kernel = "RBF")
```

```
plot_data_poly <- data.frame(Actual = testData$y, Predicted = predictions_poly, Kernel =
"Polynomial")
plot_data <- rbind(plot_data_rbf, plot_data_poly)
# Plot actual vs predicted values
ggplot(plot_data, aes(z = Actual, y = Predicted, color = Kernel)) +
geom_point() +
geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
theme_minimal() +
labs(title = "SVR Predictions vs Actual",
z = "Actual Values",
y = "Predicted Values")
```