Classifying pathologies of vertebral column using machine learning algorithms

Pronob Barman

---

1 Vertebral Column Data Set Analysis

Introduction

The data set used for this project is publicly available through the UCI Machine Learning Repository (Vertebral Column Data Set). This data set contains two types of datasets for Vertebral Column; one is Column 2 and the other is Column 3. We are interested in Column 2 dataset in our analysis. Our interested dataset contains 7 variables. And this dataset contains sample size n=310. In our dataset we have 6 continuous biomechanical variables and one categorical variable class. Here, class is our interested response variable with 2 categories- Normal and Abnormal. Below are the variables that used in our analysis:

- Class(Normal and Abnormal): response variable = Y.
- Pelvic Incidence = X_1.
- Pelvic Tilt = X_2.
- Lumbar Lordosis Angle = X_3.
- Sacral Slope = X_4.
- Pelvic Radius = X_5.
- Degree Spondylolisthesis = X_6.

We examined the distribution of each variable and found that X_1, X_4, X_4, and X_6 were approximately normally distributed while X_2 and X_3 had a right skew.

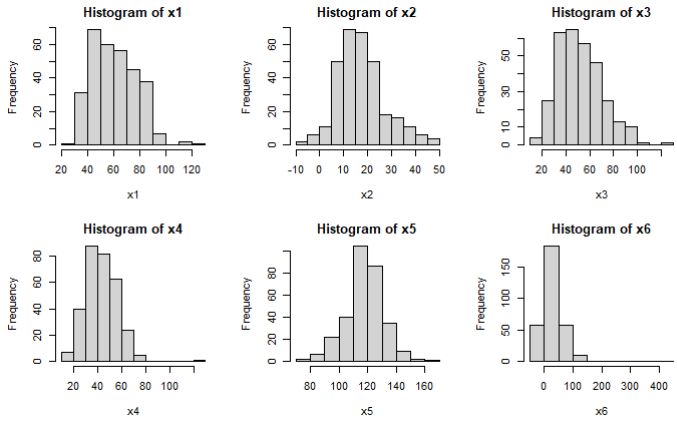

Figure 1: Distribution of Explanatory Variables

Data Analysis

1.1 Partitioning the dataset

Before we performed our data analysis we randomly partitioned our dataset into two parts: Training and Validation. Data was randomly assigned to either group with probabilities of $\frac{2}{3}$ and $\frac{1}{3}$ respectively. For partitioning the dataset we used two R packages: tidyverse [5] and caret [2] [1].

[1] Also, we used foreign [3] package to read the dataset

1.2 Perform linear discriminant analysis (LDA)

Before performing LDA [2], we needed to examine whether the eigenvalues of the sample variance/covariance matrix (consisting of six continuous variables) are close to 0. We found that one of the eigenvalues was close to 0. Therefore, we needed to drop one of the variables for performing LDA. To decide which one of the variables we should drop, we looked at the correlation matrix of the six continuous variables. We found that X_1 *and* X_4 are correlated with each other, with $R^2 = 0.81$. Furthermore, X_1 was more strongly correlated with the other predictors than X_4, on average. Therefore, we chose to drop X_1 from our LDA model before continuing.

The results from our LDA analysis are found in Table 1. The caret package[2] was used to generate these estimates, given the output of the LDA analysis.

	Training	Validation
Correct Classification Rate	0.8413	0.8431
Sensitivity	0.8649	0.8442
Specificity	0.7833	0.8400
Positive Predictive Value	0.9078	0.9420
Negative Predictive Value	0.7015	0.6364

Table 1: LDA model performance: Sensitivity, Specificity, PPV, and NPV

1.3 Perform quadratic discriminant analysis (QDA)

Next, we performed QDA, once again omitting the X_1 term from the model. The results of QDA can be found in Table 2.

	Training	Validation
Correct Classification Rate	0.8173	0.8725
Sensitivity	0.9725	0.9118
Specificity	0.6465	0.7941
Positive Predictive Value	0.7518	0.8986
Negative Predictive Value	0.9552	0.8182

Table 2: QDA model performance: Sensitivity, Specificity, PPV, and NPV

1.4 Perform logistic regression analysis

We performed the ordinary logistic regression analysis with all of the variables and found the results shown in Table 3.

	Training	Validation
Correct Classification Rate	0.8365	0.8627
Sensitivity	0.8794	0.8767
Specificity	0.7463	0.8276
Positive Predictive Value	0.8794	0.9275
Negative Predictive Value	0.7463	0.7273

Table 3: Ordinary logistic regression performance: Sensitivity, Specificity, PPV, and NPV

1.5 Perform sequence of logistic regression analysis using forward selection

We used stepwise forward selection method to select our best models. Our best models are given below. After the 4th model, adding additional terms did not improve AIC and so X_1 and X_3 were

[2]Used MASS [4] package to run lda

not included in any of the models tested. Based on the AIC values of these models, Model 3, including X_6, x_5, and X_4, is the simplest model that offers good performance on logistic regression. The performance of each model on the training data can be found in Table 5, while the performance on validation data can be found in Table 6. Model 3 is slightly worse than model 4 at correctly classifying the training dataset, while performance for both models is similar for the validation dataset.

Model	AIC
$M_1 : logit[\pi(y)] = X_6$	180.25
$M_2 : logit[\pi(y)] = X_6 + X_5$	164.11
$M_3 : logit[\pi(y)] = X_6 + X_5 + X_4$	142.97
$M_4 : logit[\pi(y)] = X_6 + X_5 + X_4 + X_2$	142.9

Table 4: Forward Selection Models

Where, $M_i = ith\ model\ \ and\ i = 1, 2, 3, 4$

	M_1	M_2	M_3	M_4
Correct Classification Rate	0.7212	0.7837	0.8558	0.8606
Sensitivity	0.7345	0.7927	0.8581	0.8636
Specificity	0.6452	0.7500	0.8491	0.8519
Positive Predictive Value	0.9220	0.9220	0.9433	0.9433
Negative Predictive Value	0.2985	0.4925	0.6716	0.6866

Table 5: Forward selection models: Sensitivity, Specificity, PPV, and NPV for training dataset

	M_1	M_2	M_3	M_4
Correct Classification Rate	0.7059	0.7941	0.8627	0.8725
Sensitivity	0.7191	0.7791	0.8667	0.8784
Specificity	0.6154	0.8750	0.8519	0.8571
Positive Predictive Value	0.9275	0.9710	0.9420	0.9420
Negative Predictive Value	0.2424	0.4242	0.6970	0.7273

Table 6: Forward selection models: Sensitivity, Specificity, PPV, and NPV for validation dataset

1.6 Perform logistic regression with shrinkage using ridge regression

We used glmnet[1] to perform ridge regression. After performing logistic regression with shrinkage using ridge regression we got the results summarized in Table 7, with a selection of values of λ, corresponding to $log(\lambda)$ values of approximately -3.8 to -1.2 . The ridge trace from the ridge regression can be seen in Figure 1.

	$\lambda = 0.303$	$\lambda = 0.209$	$\lambda = 0.144$	$\lambda = 0.075$	$\lambda = 0.022$
Correct Classification Rate	0.6971	0.6923	0.7115	0.7115	0.7163
Sensitivity	0.9333	0.9326	0.9551	0.9551	0.9767
Specificity	0.5169	0.5126	0.5294	0.5294	0.5328
Positive Predictive Value	0.5957	0.5887	0.6028	0.6028	0.5957
Negative Predictive Value	0.9104	0.9104	0.9403	0.9403	0.9701

Table 7: Ridge regression performance: Sensitivity, Specificity, PPV, and NPV for training dataset

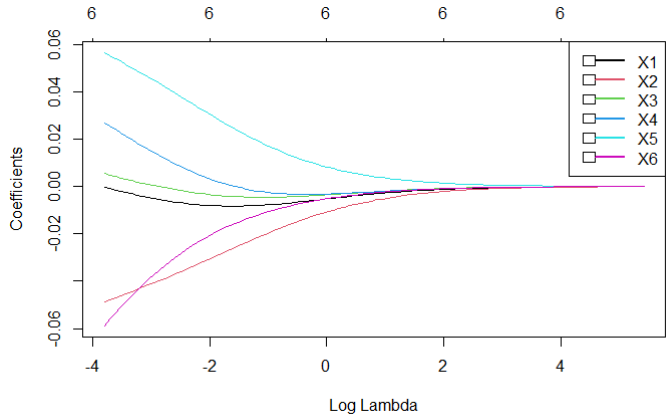

Figure 2: Ridge trace from ridge regression

	$\lambda = 0.303$	$\lambda = 0.209$	$\lambda = 0.144$	$\lambda = 0.075$	$\lambda = 0.022$
Correct Classification Rate	0.7647	0.7549	0.7451	0.7451	0.7647
Sensitivity	0.9592	0.9583	0.9574	1.0000	1.0000
Specificity	0.5849	0.5741	0.5636	0.5593	0.5789
Positive Predictive Value	0.6812	0.6667	0.6522	0.6232	0.6522
Negative Predictive Value	0.9394	0.9394	0.9394	1.0000	1.0000

Table 8: Ridge regression performance: Sensitivity, Specificity, PPV, and NPV for validation dataset

1.7 Perform six LDA's in greedy sequence

After performing the forward selection method we had a series of five models[3], summarized in Table 9. In each step of the selection process, we added the term that resulted in an LDA with the best correct classification rate on the training data.

Model	Terms	Correct Classification Rate
1	X_6	0.75
2	$X_6 + X_5$	0.8029
3	$X_6 + X_5 + X_4$	0.8462
4	$X_6 + X_5 + X_4 + X_3$	0.8462
5	$X_6 + X_5 + X_4 + X_3 + X_2$	0.8413

Table 9: Forward selection models created using LDA

We present the performance of each model on training and validation data, below, in table 10. After model 3, with $X_6 + X_5 + X_4$, there was no evidence for further performance improvements with additional terms.

Measure	Model 1	Model 2	Model 3	Model 4	Model 5
Training					
Correct Classification Rate	0.7500	0.8029	0.8462	0.8462	0.8413
Sensitivity	0.7730	0.8623	0.8707	0.8707	0.8649
Specificity	0.6667	0.6857	0.7869	0.7869	0.7833
Positive Predictive Value	0.8936	0.8440	0.9078	0.9078	0.9078
Negative Predictive Value	0.4478	0.7164	0.7164	0.7164	0.7015
Validation					
Correct Classification Rate	0.7647	0.8431	0.8627	0.8431	0.8431
Sensitivity	0.7711	0.8354	0.8571	0.8533	0.8442
Specificity	0.7368	0.8696	0.8800	0.8148	0.8400
Positive Predictive Value	0.9275	0.9565	0.9565	0.9275	0.9420
Negative Predictive Value	0.4242	0.6061	0.6667	0.6667	0.6364

Table 10: Forward selection with LDA: Performance on training and validation datasets by model

1.8 Cross validation for logistic regression [4]

To perform 10-fold cross validation, we used the trainControl function of the caret package caret [2]. Using this pacakge, we obtained the correct classification rate (CCR) of each of the models previously shown in Table 4. The results are presented in Table 11, with the 10-fold cross validation results compared to the validation results from Table 6.

Model	Cross-validation CCR	Validation CCR
M_1	0.7644	0.7059
M_2	0.7789	0.7941
M_3	0.8606	0.8627
M_4	0.8520	0.8725

Table 11: Comparison of cross validation and validation performance on logistic regression with forward selection. Additional details for each model can be found in Table 4. CCR = Correct Classification Rate. Validation CCR was previously presented in Table 6.

From these results, it is clear that cross-validation outperforms the standard approach to validation for the simplest model, while standard validation performs better for the more complex

[3]Because of collinearity presence we have only five LDA's in the greedy sequence, we dropped X_1 to reduce computation time

[4]We all contributed to this portion

models. However, it should be noted that applying cross-validation to models previously chosen through subset selection can inflate estimates of performance. This is because we selected the logistic regression models using a training set of data, while we are performing cross-validation with different, partially-overlapping, sets of data. To improve our cross-validation approach we would need to perform subset selection during cross validation, on each of the 10 folds, rather than on a training dataset. Cross validation is intended to be used when it is not practical to hold back validation data. While it does appear that the cross validation models perform slightly worse than the validation models, this may be a reasonable cost to pay if validation data is unavailable.

1.9 Removing Outliers [5]

For outlier removal, we first analyzed Mahalanobis distance within the training data, to determine if there were any datapoints that were outliers across the dimensions. We determined that Mahalanobis distances exceeding 20 appeared to be outliers, and we removed these from the training dataset. Three datapoints were removed in this way, with distances ranging from 27 to 50, as shown in Figure 3. We then reconstructed our LDA model with the revised training dataset and used it to make predictions on an unaltered validation dataset.

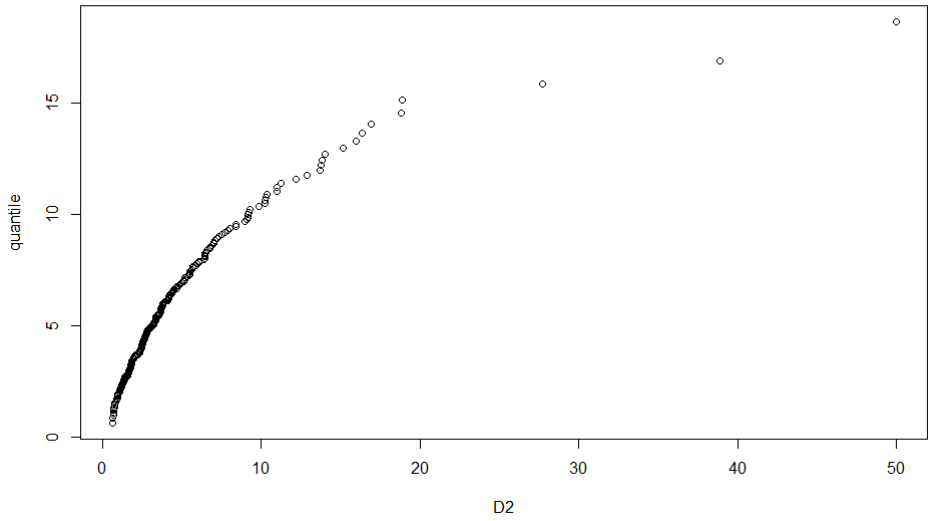

Figure 3: QQ plot of Mahalanobis distance vs quantile values of a chi-squared distribution

As shown in Table 12, our results were unaltered by the removal of outliers, with the same values being found as we previously reported in Table 1. This implies that removing outliers did not harm our analysis, but it also was not necessary. Outliers increase the variability in data, which can have a negative impact on the predictive validity of a model. Thus, it is permissible to remove outliers from the training data. However, it is not useful to remove outliers from the validation or testing data, as this would simply represent making the classification problem easier.

	Training	Validation
Correct Classification Rate	0.8413	0.8431
Sensitivity	0.8649	0.8442
Specificity	0.7833	0.8400
Positive Predictive Value	0.9078	0.9420
Negative Predictive Value	0.7015	0.6364

Table 12: Performance of Sensitivity, Specificity, PPV, and NPV

[5] We all want to pursue this part

1.10 Conclusion

In our work, we examined several methods of predicting whether a subject is "normal" or "abnormal". Based on our results, quadratic discriminant analysis (QDA) and forward selection resulted in the highest correct classification rate in the validation data of 0.8725. QDA resulted in a higher sensitivity than forward selection at the expense of lower specificity. Thus, we would select QDA as the best method to predict normal or abnormal status. Based on the results for forward selection and greedy sequence LDA, it is clear that adding additional variables to the model past the first improves our prediction. However, adding too many variables to the model causes the correct classification rate to drop in the validation data, indicating possible overfitting. Stepwise forward selection can help in preventing overfitting since it only adds covariates if they significantly improve the model.

References

[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. URL: https://www.jstatsoft.org/v33/i01/.

[2] Max Kuhn. "Building Predictive Models in R Using the caret Package". In: *Journal of Statistical Software, Articles* 28.5 (2008), pp. 1–26. ISSN: 1548-7660. DOI: 10.18637/jss.v028.i05. URL: https://www.jstatsoft.org/v028/i05.

[3] R Core Team. *foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...* R package version 0.8-81. 2020. URL: https://CRAN.R-project.org/package=foreign.

[4] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Fourth. ISBN 0-387-95457-0. New York: Springer, 2002. URL: https://www.stats.ox.ac.uk/pub/MASS4/.

[5] Hadley Wickham et al. "Welcome to the tidyverse". In: *Journal of Open Source Software* 4.43 (2019), p. 1686. DOI: 10.21105/joss.01686.