Machine Learning to Predict a Diabetic Diagnosis

Eduardo Jones

Using Machine Learning to Predict a Diabetic Diagnosis

According to the CDC, diabetes is a condition where the body does not effectively produce insulin. This condition affects roughly 1 in 10 Americans and is the seventh leading cause of death in the U.S. (CDC, 2022). For these reasons, it might be helpful to identify specific indicators that may determine if a person has diabetes.

To start, we obtained the Pima Indians Diabetes Dataset from the Kaggle website. The data, which were initially from the National Institute of Diabetes and Digestive and Kidney Diseases, contains eight predictor variables and one binary target variable. The predictors are the number of pregnancies, glucose, blood pressure, insulin, skin thickness, BMI, age, and a diabetes pedigree function that scores the likelihood of diabetes based on family history. The target variable is labeled "outcome" and indicates whether the person has diabetes. All patients in the dataset are at least 21 years of age, female, and have Pima Indian heritage. Our objective is to build a model that can predict a person's diabetic diagnosis using the explanatory variables.

We begin by exploring our predictors to understand their distributions and relationships. The predictors had a positive and nonlinear correlation, with some variables exhibiting moderate correlation (figure 11). We also learned that the predictors presented significant skewness and could benefit from pre-processing. Next, we examined the data for missing values. Skin thickness (29.6%) and insulin (48.7%) have the highest concentration of missing values relative to the other predictors. We initially chose to remove these variables; however, after running the data a second time without deleting any variables, we found that overall accuracy in all models improved by up to 2%. Furthermore, we chose to keep both variables in our analysis because the missing data for skin thickness and insulin is less than 70%. Therefore, the remaining processes mentioned in this paper will include all predictors.

We had several options to address the missing data, including using median values to make our estimates. However, we ultimately decided to handle the missing data with the Multivariate Imputation by Chained Equations (MICE) package in R. We used the Classification and Regression Trees (CART) method in our MICE function.

Our next step was to select a model. Because our outcome variable is categorical, we decided to use classification models. The data also had nonlinear characteristics; thus, we further narrowed our selection to nonlinear methods. Nevertheless, we still included linear models in our analysis since they have certain features that could prove advantageous to our cause. Accordingly, the following supervised models were used for our predictions: Logistic Regression, Partial Least Squares Discriminant Analysis regression (PLS-DA), Support Vector Machines (SVM), and Random Forest (RF).

The next stage involved splitting the data into training and testing sets. 80% of the data was dedicated to training, while the remaining 20% was used for testing. Both were pre-processed using various methods. Once transformations were complete, we began our analysis. Of the four models, RF was the superior model with an accuracy of 75% and a Kappa value of 42%. The remainder of this piece will describe the methods outlined above in greater detail.

This Diabetes dataset was originally from the National Institute of Diabetes and Digestive and Kidney Disease. It has a total of nine different variables and 768 observations. An overview of these variables is below.

(1)*Pregnancies*: Number of times pregnant
(2)*Glucose*: Plasma glucose concentration a 2 hrs in an oral glucose tolerance test
(3)*BloodPressure*: Diastolic blood pressure-mm Hg
(4)*Skin Thickness*: Triceps skin fold thickness-mm
(5)*Insulin*: 2-Hour serum insulin-muU/ml
(6)*BMI*: Body Mass Index - (weight in kg / (height in m)^2)
(7)*DiabetesPedigreeFunction*
(8)*Age*: years
(9)*Outcome*: Class variable 0 or 1

Our response variable, 'Outcome,' is a factor variable with two classes: 0, meaning the patient does not have diabetes, and 1, meaning the patient does have diabetes. We re-coded these classes into the following: 0 = "No" and 1 = "Yes." After a deeper look into our dataset, we concluded that different pre-processing methods had to occur. First, we looked at the summary of our data set. As shown in figure 1 below, it initially seems that we have no missing values. However, looking closer, we can see that 'Glucose,' 'BloodPressure,' 'SkinThickness,' 'Insulin,' and 'BMI' all have minimum values of 0, which is an unlikely result for those predictors.

```
 Pregnancies        Glucose      BloodPressure    SkinThickness      Insulin           BMI
Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0   Min.   : 0.00
1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0   1st Qu.:27.30
Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5   Median :32.00
Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8   Mean   :31.99
3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2   3rd Qu.:36.60
Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0   Max.   :67.10
DiabetesPedigreeFunction      Age           Outcome
Min.   :0.0780           Min.   :21.00   Min.   :0.000
1st Qu.:0.2437           1st Qu.:24.00   1st Qu.:0.000
Median :0.3725           Median :29.00   Median :0.000
Mean   :0.4719           Mean   :33.24   Mean   :0.349
3rd Qu.:0.6262           3rd Qu.:41.00   3rd Qu.:1.000
Max.   :2.4200           Max.   :81.00   Max.   :1.000
```

Figure 1: Dataset summary output

Data Imputation

To correct the issue, we transformed our data set by changing any value of 0 from all predictors (except pregnancies) to a value of "NA." We applied the technique to age because there were no values of 0, so age would not be affected by imputation. After converting values, we can see in figure 2 that "Insulin" and "SkinThickness" have the highest percentage of missing values. "Insulin" at 48.7% and "SkinThickness" at 29.6%. This finding was initially concerning, but after further evaluation, we decided to keep these two predictors for further modeling. Since the missing percentages were below 70%, we imputed the predictors' values instead of removing them completely. To do this, we used the CART method from the Mice package in R. CART has

many desirable characteristics that prove relevant to our dataset. It is effective against outliers, can deal with multicollinearity and skewness, and has the flexibility to fit nonlinear data (van Buuren, n.d.). The Mice function then uses a predictive algorithm to determine how a given variable's values can be predicted based on other values.

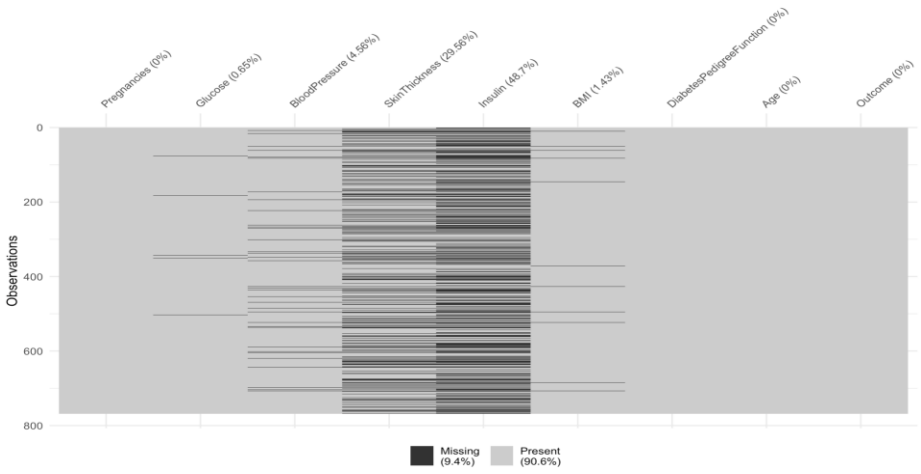

Figure 2: Visual of missing data

Splitting the Data

Before splitting our data, we looked for imbalances in our response variable. There were 268 "Yes" outcomes and 500 "No" outcomes. Because there were more than double the "No" outcomes, we split our data set using the stratified random split method. This method splits the data proportionally using the createDataPartition() function. We used 80% of our data for training our model, and the remaining 20% of data was used to test the model accuracy.

Pre-processing the Data

After performing the final splits, we continued pre-processing the training and testing data sets using various methods. As shown in our distribution plot in figure 4, several variables suffered from being right-skewed; therefore, we used the Center and Scaling methods to fix this issue.

Lastly, we see in figure 5 that insulin had several outliers. Since this could cause problems in our analysis, we applied the "BoxCox" transformation in R. Other transformations used were "nzv" and "spatialSign." The final results of our pre-processing are shown in figure 6.

Figure 3: Relationship between predictor variables

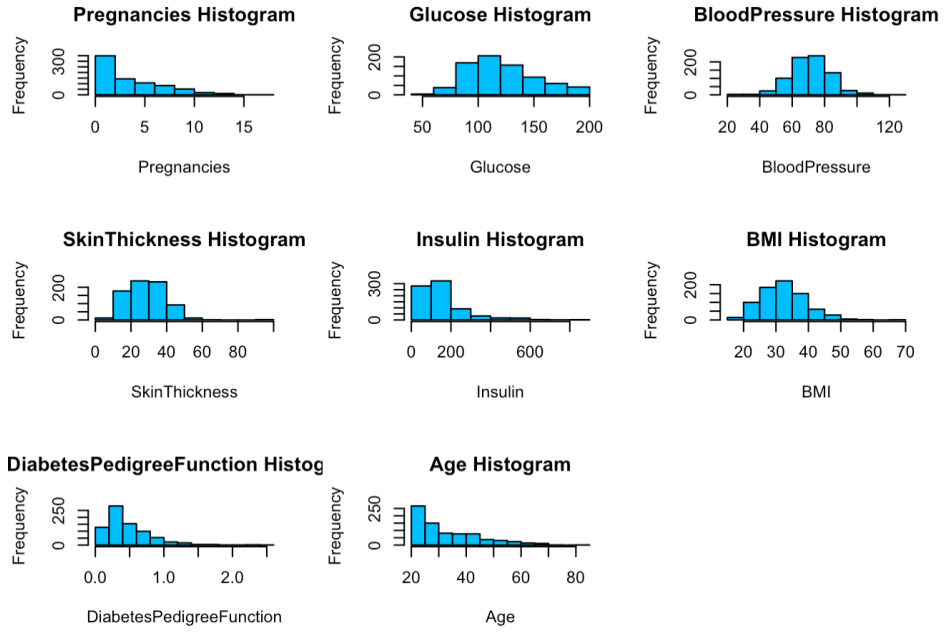

Figure 4: Variable distribution before transformations

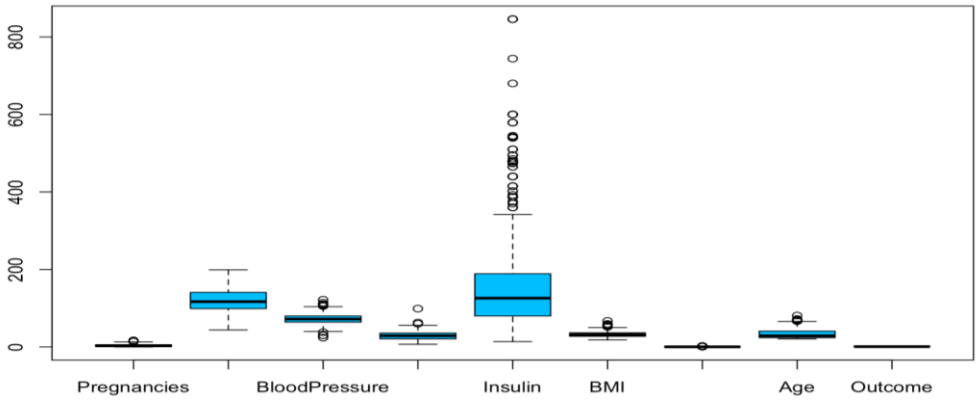

Figure 5: Box plot shows a significant number of outliers for insulin

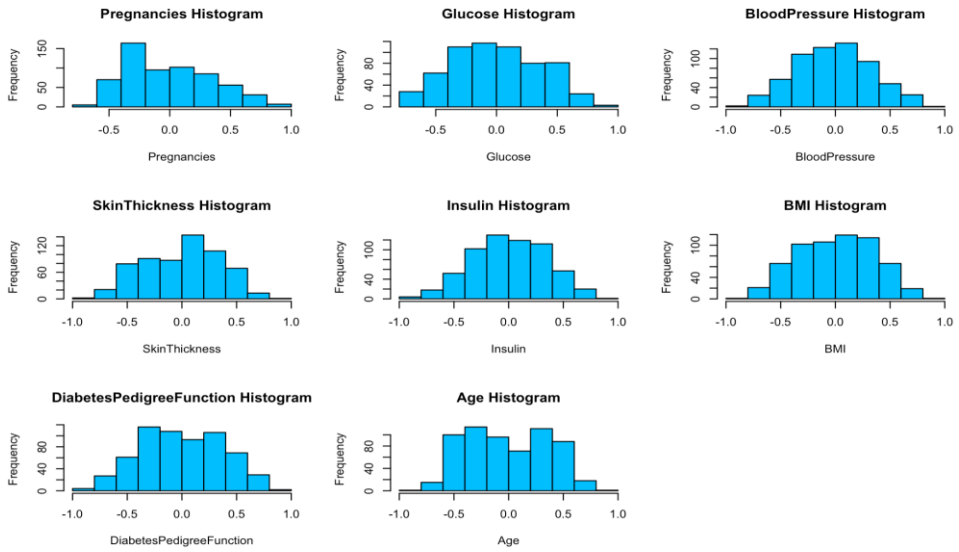

Figure 6: Variable distributions after transformations

Statistical Learning Method(s)

We selected categorical methods to make our predictions. While many models can handle categorical data, the models we selected each possess valuable characteristics that make them ideal for our analysis. Among the chosen models were the Logistic Regression model, Partial Least Square Discriminant Analysis (PLS-DA), Support Vector Machines (SVM), and Random Forest (RF). For reasons expanded upon below, we believe these models will help achieve our goal of accurately predicting a diabetic diagnosis.

Models

Logistic regression is a well-known classification method. Although logistic regression models, by default, generate linear classification boundaries (Kuhn & Johnson, 2018, p. 48), their usefulness for categorical data and simplicity made it one of our top picks. Other

advantages of logistic regression include that results are easy to interpret, generally have high accuracy, and do not take long to run.

PLS-DA is also a linear classification model and is typically used when data is highly correlated. Although our data was only moderately correlated, a method that specializes in multicollinearity could be helpful. Additionally, PLS-DA does well with outliers, such as those seen in our insulin variable, further making it an ideal choice. Interestingly, the results (figure 8) from this model and the logistic regression did not differ significantly from their nonlinear counterparts.

It is important to note that the next 2 models in our analysis are Non-Linear classification models. We decided to train our dataset with these nonlinear models, because if you refer back to figure 3 of the plotted relationship between predictors. The relationship between many predictors tends to not follow a linear relationship. This non linear relationship can cause linear models to have less Accuracy , kappa , Sensitivity , and Specificity results. However there are still a few predictors that have a linear relationship. Therefore after consideration our best approach was to consider training and testing on both linear and non linear models.

SVM was our first nonlinear method. It is also important to note that the maximal classifier and the support vector classifier are considered linear classification boundaries. Therefore, to extend the linear nature of the model to nonlinear classification boundaries, we consider kernel function instead of the simple linear relationship. There are three popular kernels for the non-linear classification boundaries which are polynomial, radial basis function, and hyperbolic tangent.

Our third overall model and second nonlinear model was Random Forest. This method uses a classification tree analysis. Tree-based methods are simple and can be really useful when having to interpret. However, this classification tree based methods can often improve dramatic prediction accuracy, but will suffer from simple interpretation of the model. In conclusion, a quick overview of Random Forest and how it works, is by growing multiple trees which are then combined to yield a single consensus prediction.

After performing our Random Forest modeling, figure 7 down below shows which predictors were the most important when trying to predict diabetes by Yes or No. As you can see Glucose , Age, DiabetesPedigreeFuntion, and a bit of BMI were the most significant predictors in our analysis.

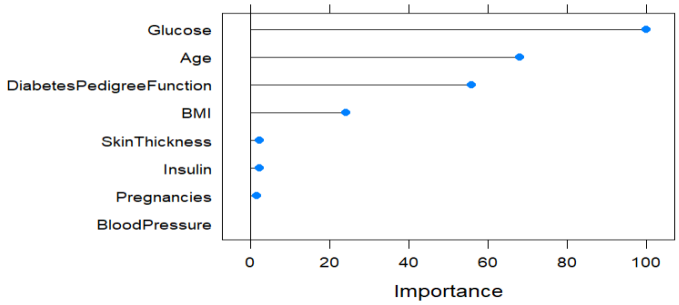

Figure 7: Variable importance graph. Using Random Forest

Analysis Results

While all models performed well, our results show that Random Forest outperformed the other three methods by a small margin. We used four measures to make this assessment: accuracy, which identifies relationships and patterns between variables; the Kappa statistic, a measure that functions similar to accuracy but also considers the accuracy that would be generated by chance; sensitivity, the true positive rate; and specificity, the true negative rate (Kuhn & Johnson, 2018, p.256).

The test results indicated that Random Forest predicted the outcome correctly with a respectable 75% accuracy, while the remaining models had an accuracy of 73% or less. It also had the highest values for Kappa (42%), sensitivity (83%), and specificity (58%). Additionally, we examined the tradeoff between sensitivity and specificity using the Receiver Operator Characteristic (ROC) curve. As shown in figure 9, the Random Forest curve is slightly closer to the top left corner than the other models; this indicates that RF has a higher model accuracy.

Models	Accuracy	Kappa	Sensitivity	Specificity
Logistic Regression	0.7124	0.3592	0.7900	0.5660
PLSDA	0.7190	0.3710	0.8000	0.5660
SVM	0.7320	0.4002	0.8100	0.5849
Random Forest	0.7451	0.4243	0.8300	0.5849

Figure 8: Analysis results for all four models. Logistic Regression, PLS-DA, SVM and RF

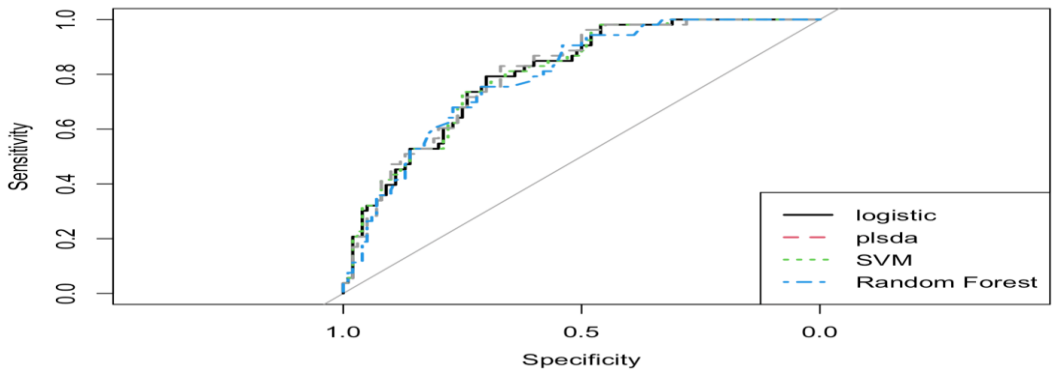

Figure 9: ROC for all four models

Finally, we created the confusion matrix shown in figure 10. We determined that the models with the best sensitivity are ranked in this order: Random Forest at 0.83, SVM at 0.81, and PLS-DA at 0.80. Next, we see that the best specificity is tied between Random Forest and SVM at 0.5849, followed by PLS-DA and Logistic at 0.5660.

		Observed					
	Logistic	Yes	No	**SVM**	Yes	No
	Yes	30	21	Yes	31	19
Prediction	No	23	79	No	22	81
	PLSDA	Yes	No	**RF**	Yes	No
	Yes	30	20	Yes	31	17
	No	23	80	No	22	83

Figure 10: Confusion Matrix

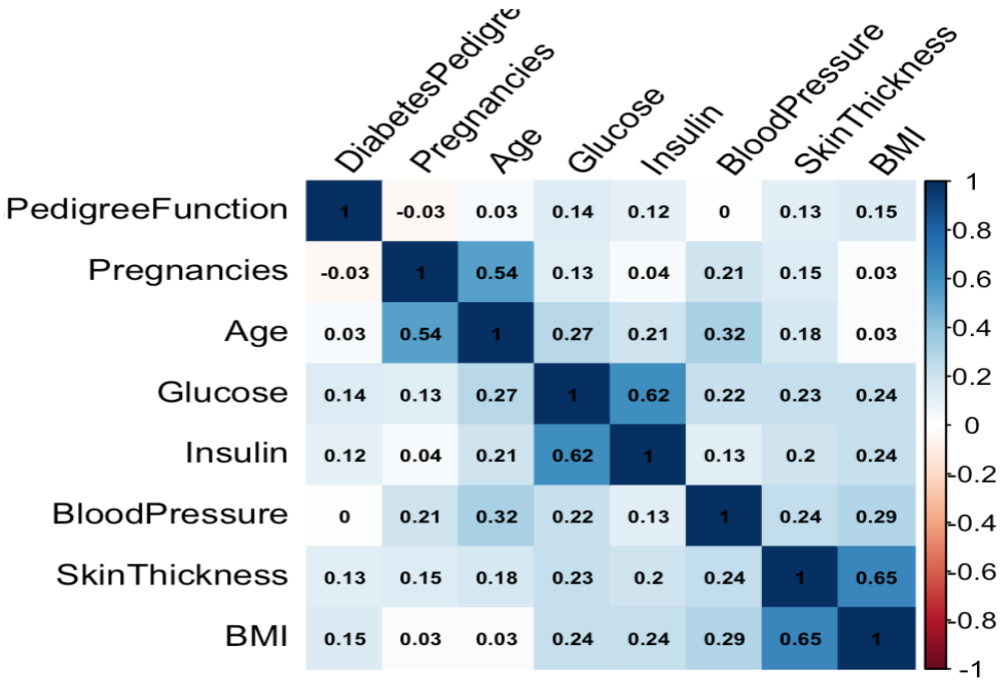

Figure 11: Correlation matrix. BMI and skin thickness have moderate correlation (0.65). Glucose and insulin also exhibit moderate correlation (0.62)

Conclusion

Our analysis aimed to build a model that can accurately predict whether a patient has diabetes given specific indicators. Before this could be done, we had to review our data to ensure it was suitable for testing. There were several steps involved in this phase.

First, we created a summary report, which indicated there were several missing values. We addressed this by using the MICE function and the CART method to impute those values.

Second, we needed to divide our data set into training and testing. We first checked for any imbalances in our target variable. Because there were imbalances, we split our data set using the stratified random split method. This method splits the data proportionally using the createDataPartition() function. 80% of our data was used to train our model, and the remaining 20% was used to test our model.

The third step was to prepare the data for modeling. A quick overview of the data revealed we had skewness and outliers. We transformed and cleaned our data using various pre-processing methods. This included BoxCox for handling outliers, scale and center to deal with skewness, and other pre-processing methods such as nzv and spatialSign.

Once our data was cleaned and pre-proceed, we used different classification models such as Logistic regression, PLS-DA, SVM, and Random Forest. After running each model, we determined that the Random Forest model was the best at predicting whether a patient had diabetes. It had the highest values for Accuracy and Kappa. The Kappa values were between 0.4 and 0.75, which are considered moderate to good.

References

CDC. The Facts, Stats, and Impacts of Diabetes. (2022, January 24). Centers for Disease Control and Prevention. https://www.cdc.gov/diabetes/library/spotlights/diabetes-facts-stats.html#:%7E:text=Key%20findings%20include%3A,t%20know%20they%20have%20it

Kuhn, M., & Johnson, K. (2018). Applied Predictive Modeling. Springer Publishing.

Van Buuren, S. (n.d.). Flexible imputation of missing data (2nd ed.). Taylor & Francis. https://stefvanbuuren.name/fimd/sec-cart.html