**Pima Indians Diabetes**

Michael Zelaya, [mzelaya@bellarmine.edu](mailto:mzelaya@bellarmine.edu)

**ABSTRACT**

This paper is half explanatory data analysis (EDA) and half logistic regression in predicting whether or not a patient has diabetes. The first part I did was the EDA process, which helped me better understand the data by creating several visuals about each specific medial measurement. In addition, the second part of this project implemented a logistic regression to predict whether a patient has diabetes as accurately as possible.

1. **INTRODUCTION**

For my final project, I am interested in exploring the Prima Indians Diabetes dataset I found on Kaggle. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes based on specific measurements. In this project, I aim to build a machine-learning model by using logistic regression to accurately predict whether a patient in the dataset has diabetes or does not have diabetes.

1. **BACKGROUND**

In the dataset, there are 768 patients and nine columns. The dataset includes several medical variables and one target variable: the outcome. For instance, the dataset contains columns for the number of pregnancies the patient has had, glucose level, blood pressure measurement, skin thickness, insulin, BMI, diabetes pedigree function, age, and outcome. In addition, all patients in the dataset are females at least 21 years old and of Pima Indian heritage.

1. **EXPLORATORY ANALYSIS**

This data set contains 768 samples with nine columns, with most data types being quantitative and just one column being binary (0 or 1).

**A graph with blue bars

Description automatically generated**

**Figure 1: Demonstrates a right-skewed distribution.**

This figure demonstrates a right-skewed distribution for each histogram's skin thickness, insulin levels, and diabetes pedigree function.

A graph of a tall building

Description automatically generated with medium confidence

**Figure 2: Demonstrates a relatively normal distribution.**

This second diagram demonstrates a normal distribution for each histogram's glucose concentration, blood pressure, and body mass index (BMI).

A comparison of a graph

Description automatically generated

**Figure 3: Demonstrates a right-skewed distribution.**

These box plots demonstrate a right-skewed distribution for the number of pregnancies and age in years. As you can see on the box plot on the left, the value count decreases as the number of pregnancies increases. In addition, the box plot on the right demonstrates that the dataset contains mostly younger patients.

A blue and orange pie chart

Description automatically generated

**Figure 4: Shows a pie chart of the percentage of patients who either have diabetes or not.**

As you can see, the pie chart demonstrates that approximately 36% of the patients have diabetes. On the other hand, 65% of the patients did not have diabetes.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| **Pregnancies** | int64 |
| **Glucose** | int64 |
| **Blood Pressure** | int64 |
| **Skin Thickness** | int64 |
| **Insulin** | int64 |
| **BMI** | float64 |
| **Diabetes Pedigree Function** | float64 |
| **Age** | int64 |
| **Outcome** | int64 |

1. **METHODS**

In this section, describe how you prepared the data for your model and performed multiple experiments using different parameters for the model(s).

* 1. *Data Preparation*

First, I imported the necessary libraries like NumPy, pandas, matplotlib, and seaborn that I would use through the EDA analysis and logistic regression process. Second, luckily, there were no missing data values, so I didn't have to do anything. Then, I was ready to start the EDA process, in which I used histograms, bar plots, and a pie chart for visualization purposes to learn and better understand the data. Next, I split my dependent variable (Outcome) and other independent variables for my prediction. I finally trained my linear regression model to predict if a patient has diabetes.

* 1. *Experimental Design*

x2=dataset[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI',

'DiabetesPedigreeFunction', 'Age']]

y2=dataset['Outcome']

Table X: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | All eight (8) raw features with 80/20/20 split for train, validate, and test |
| 2 | All eight (8) raw features with 70/15/15 split for train, validate, and test |
| 3 | All eight (8) raw features with 90/5/5 split for train, validate, and test |

* 1. *Tools Used*

Describe all of the software tools you used to perform your data preparation and model implementation. For example:

The following tools were part of this analysis: Python v3.5.2 running the Anaconda 4.3.22 environment for Apple Macintosh computer to help with all analysis and implementation. In addition to base Python, the following libraries were part of the process: Pandas 0.18.1, NumPy 1.11.3, Matplotlib 1.5.3, Seaborn 0.7.1, SKLearn 0.18.1, and import warnings.

Provide a brief explanation of why you chose these tools.

I chose these tools because they were the exact tools we used throughout the semester and were the ones I was most familiar with and comfortable working with.

1. **RESULTS**
   1. *Classification Measures/ Accuracy measure*

*A chart of a blue yellow and purple box

Description automatically generated with medium confidence*

A number of numbers in a row

Description automatically generated with medium confidence

**Figure 5: Experiment 1 (80/20/20 split for train, validation, and test.)**

This first experiment demonstrates an accuracy of 82%. In addition, the model predicted 84% for patients who did not have diabetes, while on the other hand, the model only predicted 76% accuracy for patients who did have diabetes.

A chart with different colored squares

Description automatically generated

A number of numbers in a row

Description automatically generated with medium confidence

**Figure 6: Experiment 2 (70/15/15 split for train, validation, and test.)**

In my second experiment, the model predicted a total accuracy of 78%, which was worse than in my first experiment. The model indicated that 80% did not have diabetes, while it only proved that 71% of patients had diabetes.

A chart of a blue yellow and purple box

Description automatically generated with medium confidence

A number of numbers in a row

Description automatically generated with medium confidence

**Figure 7: Experiment 3 (90/5/5 split for train, validation, and test.)**

In my third experiment, the model had an accuracy of 87%, which was the best out of the other two experiments I had done before. In addition, it had an accuracy of 87% in detecting patients with diabetes and an 86% accuracy with patients who did not have diabetes.

* 1. *Discussion of Results*

Experiment 3, all eight (8) raw features with 90/5/5 split for train, validation, and test, was my best model with an accuracy of 87%. On the other hand, in experiment 2, all eight (8) raw features with 70/15/15 split for train, validate, and test was my worst model with an accuracy of 78. I believe this is because the higher the split for the train, the more accurate the model becomes, but at the same time, I'm not sure if that is true or not.

* 1. *Problems Encountered*

Luckily, my project didn't have many problems; however, one problem occurred during the EDA process, specifically the heat map visualization. In general, the colors and results of both heat maps looked accurate. Still, for some reason, I was missing most of the values, which prevented me from further examining the correlation of each variable.

* 1. *Limitations of Implementation*

My model did pretty well to accurately predict diabetes, ranging from the lowest accuracy of 78% to the highest of 87%. However, there can always be room for improvement to improve the model.

* 1. *Improvements/Future Work*

I would like to improve my model's accuracy to be higher for future work. I might try different models to see if that improves the model's accuracy. In addition, I would want to experiment by removing variables one by one to see if that also helps the models' prediction accuracy.

1. **CONCLUSION**

In this project, I found 0 columns with missing values for the data processing. Then, I made the outcome independent for the logistic regression model and left the other variables dependent. Next, I split the dataset into training and test sets. Next, I trained the logistic regression model and later performed using a confusion matrix, confusion matrix display, and classification report, and predicted new results for each case (row). The best classification report out of my three experiments demonstrates an 87% accuracy in patients who didn't have diabetes and an 86% accuracy in patients who did have diabetes. Lastly, the model did pretty well in detecting whether a patient has diabetes. Still, there is always more room to improve the mode's prediction accuracy.

**REFERENCES**

Link: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>