Diabetes Prediction with Logistic Regression

Angela Boakye Danquah, Yayi (Celine) Feng, Evan Mitchell
DS 5100 - Adam Tashman
July 18, 2021

Abstract

This project focuses on exploring the Pima Indians Diabetes dataset and predicting diabetes in Pima Indian women using a machine learning model based on the patients' medical metrics such as Glucose, BMI, age, etc. The dataset was retrieved from Kaggle and was processed to eliminate anomalies such as missing values. Through exploring and visualizing our dataset with tables and plots, we found that the glucose level of the patient was the most relevant in determining if the patient has diabetes. Moreover, our logistic regression model was able to predict with 77% accuracy whether a Pima Indian woman would be diagnosed with diabetes given the metrics in the dataset, and it was revealed that the glucose level was the most important feature in predicting diabetes in our model.

Introduction

Diabetes is a chronic health condition that occurs when the blood glucose level in the body is too high. Having diabetes can potentially lead to many health concerns and problems such as strokes, heart disease, kidney malfunction, and more. For this project, we are interested in working with a dataset that contains various diagnostic measurements of diabetes and the diagnosis with special concentration on the population of women of Pima Indian heritage. We would like to determine the accuracy of diagnosing diabetes in women of Pima Indian heritage with the given medical factors and the most important feature when making this prediction. Understanding the underlying factors that lead to diabetes is essential for prevention and timely treatment.

Data Description

Our Pima Indians Diabetes dataset originates from the National Institute of Diabetes and Digestive Kidney Diseases. It contains information about 768 women of Pima descent located near Phoenix, Arizona; all of whom are at least 21 years old.

The dataset contains various medical predictor variables and one target variable called Outcome. The predictor variables include age, number of pregnancies, BMI, glucose, blood pressure, insulin, skin thickness, BMI, and diabetes pedigree. The plasma glucose concentration and insulin concentration were measured by the two-hour oral glucose tolerance test and two-hour serum insulin test, respectively. The blood pressure column accounts for the diastolic blood pressure measurements, and the skin thickness column contains measurements of triceps skin fold thickness in millimeters. The diabetes pedigree column contains values which are scores that represent the likelihood of the patient developing diabetes based on family history. The Outcome column indicates whether an individual was diagnosed with diabetes. A value of one in the column represents a diabetic individual, while a value of zero represents an individual without diabetes.

Data Processing Methodology

The first step of our data cleaning process was to import the necessary libraries, such as Pandas, Numpy, Seaborn, and Matplotlib. We obtained our diabetes dataset from Kaggle, and then saved it as a CSV file called diabetes.csv. Next, we imported and loaded our dataset into a pandas dataframe called data, and used data.columns to list the features we could use to predict the diabetes outcome. Then, we made sure that our data does not contain any missing values by using pd.isnull(data).sum(). We noted that the dataset used zero to represent a missing data point; we assumed that variables with biologically impossible zero values represented missing data, while variables that could feasibly be zero, such as the number of pregnancies a woman has had, were assumed to be complete data. Finally, we were left with 392 rows, each representing a person for which we have complete data. We believe this sample size was sufficient for the following data analysis and visualization tasks.

Testing

Two unit tests were written to test out the functions we built. The first function was built to clean up a given dataset by rows with missing values, in this case, any value that is 0 except for columns where the value 0 has significant meaning (for example, the columns Pregnancies and Outcome in the diabetes dataset were passed into the exclusions parameter). A unittest helps to check whether there is any value 0 in a column (except for the columns in exclusions) by counting the appearance of 0 in that column and comparing that with the expected value.

```
[9]
    def cleanZeros(df, exclusions=None):
        exclusions = [] if exclusions is None else exclusions
        for (columnName, columnData) in df.iteritems():
            if (columnName not in exclusions):
                df = df[df[columnName]>0]
        return df
```

```
[5] import unittest

    def test_cleanZeros():

            cleaned_df = cleanZeros(data, exclusions = ['Pregnancies','Outcome'])
            counts = 0
            for i in cleaned_df['Glucose']:
                if i == 0:
                    counts += 1

            expected = 0

            assert counts == expected

    if __name__ == "__main__":
        test_cleanZeros()
        print("Everything passed")
```

```
    Everything passed
```

Figure 1 *Clean missing value function and its unit test*

The second function was built to find the mean value of the predictor variable given a specific outcome. This function is helpful in the early phase of data preprocessing because it can reveal whether there is a general difference between the two different outcomes. For the unit test of this function, a simple data frame was generated and used to check the validity of this function.

```
[6] def find_mean_outcome(df, predictor_variable, outcome_variable):
      new_df = df.groupby(outcome_variable).mean()
      return new_df[predictor_variable]

    test_df = pd.DataFrame({'a':[1,2,3,4], 'b':[0,1,0,1]})

    def test_find_mean_outcome():

      test_mean_df = find_mean_outcome(test_df, 'a', 'b')
      test_mean_count = test_mean_df.iloc[0]

      expected = 2

      assert test_mean_count == expected

    if __name__ == "__main__":
        test_find_mean_outcome()
        print("Everything passed")
```

```
    Everything passed
```

Figure 2 *Group by mean function and its unit test*

Results

In order to get an initial insight into our data, we started by looking at the mean value of each predictor variable with the different outcomes using the .groupby function followed by the .mean() function. We can see that the diabetic population has on average higher values for all eight medical predictor variables.

Outcome	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	2.721374	111.431298	68.969466	27.251908	130.854962	31.750763	0.472168	28.347328
1	4.469231	145.192308	74.076923	32.961538	206.846154	35.777692	0.625585	35.938462

However, the mean values we obtained above only provided a very general understanding of the relationship between the predictors and outcome. To build additional insight into our data, we started by finding the correlations between every pair of predictor variables and the outcome variable using a heat map. A heat map is a data visualization tool that shows the magnitude of the relationship or correlation as color in two dimensions. The color intensity can give visual cues about how they vary over space.

Figure 3 *Heat map visualization of correlation between predictor variables and target variable*

One of the main assumptions for our logistic regression is that there should not be any multicollinearity between the independent variables. There are two pairs of independent variables that are slightly correlated: Age/Pregnancies and Skin Thickness/BMI. However, the VIF scores for these variables were all low (below 2.05), so we will still be able to complete a logistic regression despite minor multicollinearity. Then, we looked at how much each predictor variable correlates with the Outcome variable by sorting the correlation values of each attribute with Outcome in descending order:

```
Outcome                     1.000000
Glucose                     0.515703
Age                         0.350804
Insulin                     0.301429
BMI                         0.270118
Pregnancies                 0.256566
SkinThickness               0.255936
DiabetesPedigreeFunction    0.209330
BloodPressure               0.192673
Name: Outcome, dtype: float64
```

Figure 4 *VIF scores*

The correlation coefficient ranges from -1 to 1; thus, when it is closer to 1, it demonstrates a strong positive correlation, and when it is closer to -1, it means there is a strong negative correlation. When the coefficient is closer to 0, it means there is no linear correlation. From the table above, we can see that the top four predictors of the outcome of diabetes are Glucose, Age, Insulin, and BMI.

In order to examine the relationship between each predictor variable and the outcome variable further, we created several box and whisker plots. The first figure consists of box and whisker plots with the four predictor variables that have the highest correlation with the outcome variable, while the latter figure consists of the plots with the four lower correlation values. Comparing the content of the plots, we can see that the median glucose level of diabetic people is higher than the median glucose level of nondiabetic people, while the other four predictor variables have less significant differences between the two groups.

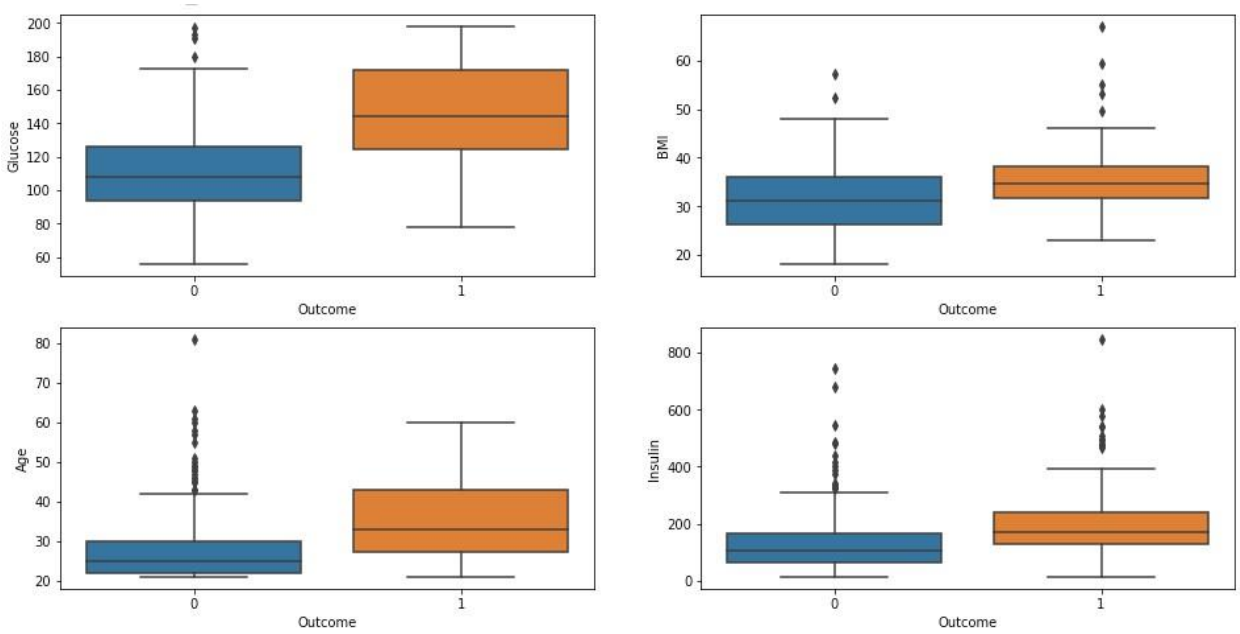

Figure 5.1 *Box and Whisker plot (I)*

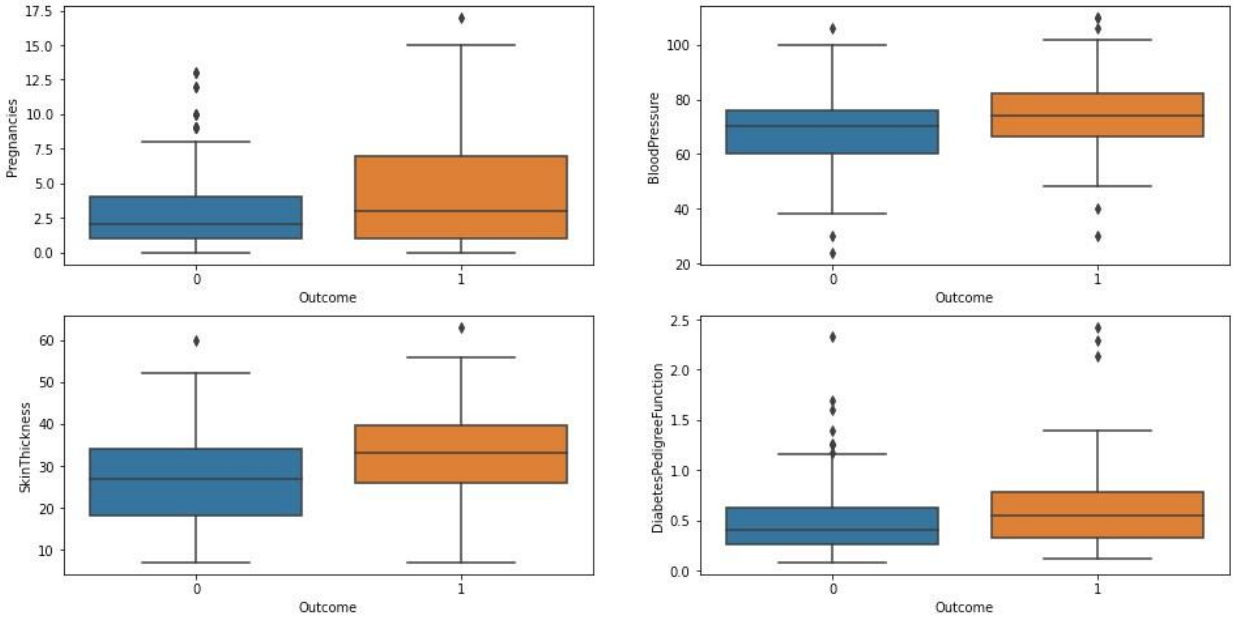

Figure 5.2 *Box and Whisker plot (II)*

In order to build a model to determine the accuracy of diabetes prediction, we began by separating the features and target variable and storing them as X and y, respectively. Then, we split the data into a training set and test set. There are 392 records in total after previous data cleaning. We used 274 records, or 70% of our data, to train the model and 118 to test. Next, we normalized our features, which is useful for understanding the importance of each feature later. Then, we used a machine learning model called logistic regression from the sklearn library to train our classification model, and then used our test data to find out the accuracy of the model.

Conclusions

Our test data yielded an overall accuracy score of 77%, which means that our logistic regression model could predict whether a patient has diabetes given the same features with 77% accuracy. As shown in Figure 6, the model was effective at producing true negatives, which occurs when an individual is correctly predicted to not have diabetes.We achieved a true positive rate of 57% and a true negative rate of 86%. The low true positive rate indicates that we produced many false positives, which occur when our model incorrectly predicts that an individual has diabetes. It is very important to minimize false positives because in medicine can often result in unnecessary treatment and an incorrect diagnosis.

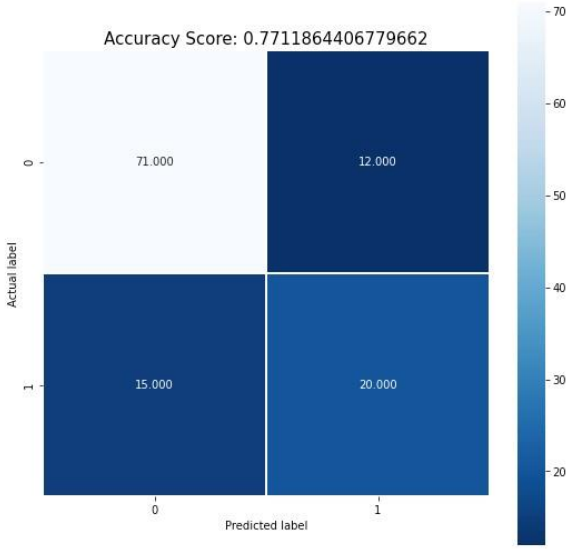

Figure 6 *Heat map visualization of confusion matrix*

To get a better sense of how the logistic regression model works, we can visualize how our model uses the features and their importance. Looking at Figure 7, the most important feature for predicting whether a woman of Pima Indian heritage has diabetes is glucose, followed by age and family history (represented by diabetes pedigree function scores).

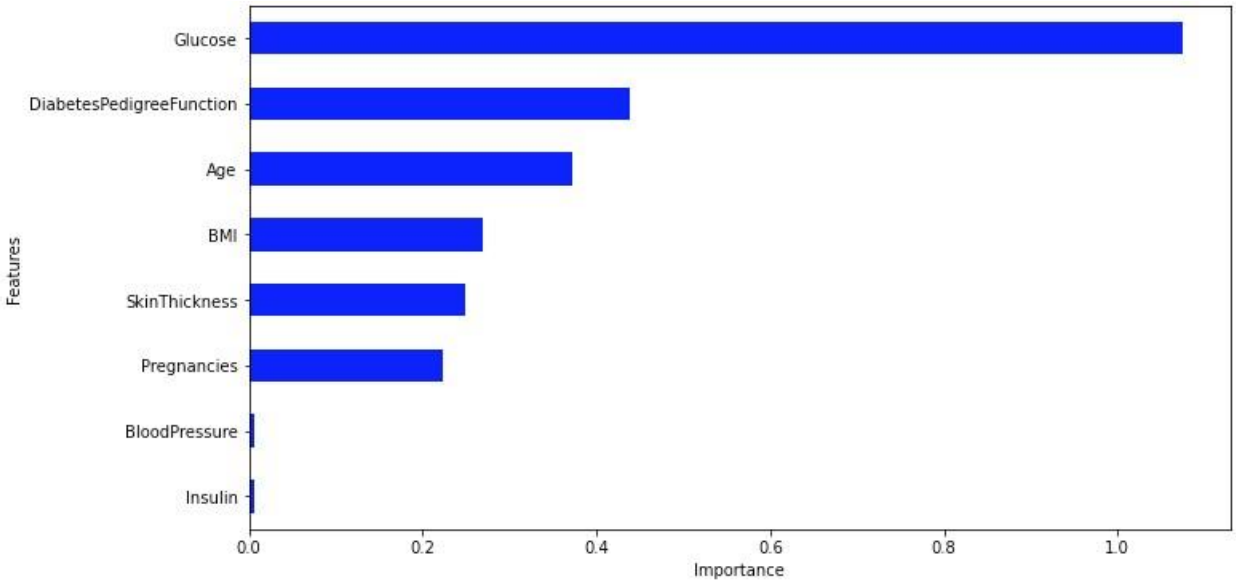

Figure 7 *Feature Importance Chart*

The AUC - ROC curve shown in Figure 8 measures the performance of a classification model. ROC is a probability curve that plots the true positive rate against the false positive rate. AUC stands for area under the curve, which measures separability. The closer the AUC is to 1, the

better the model is at distinguishing between the two classes. Our model's AUC is .855 which means there is an 85.5% chance our model will be able to distinguish between our two classes.

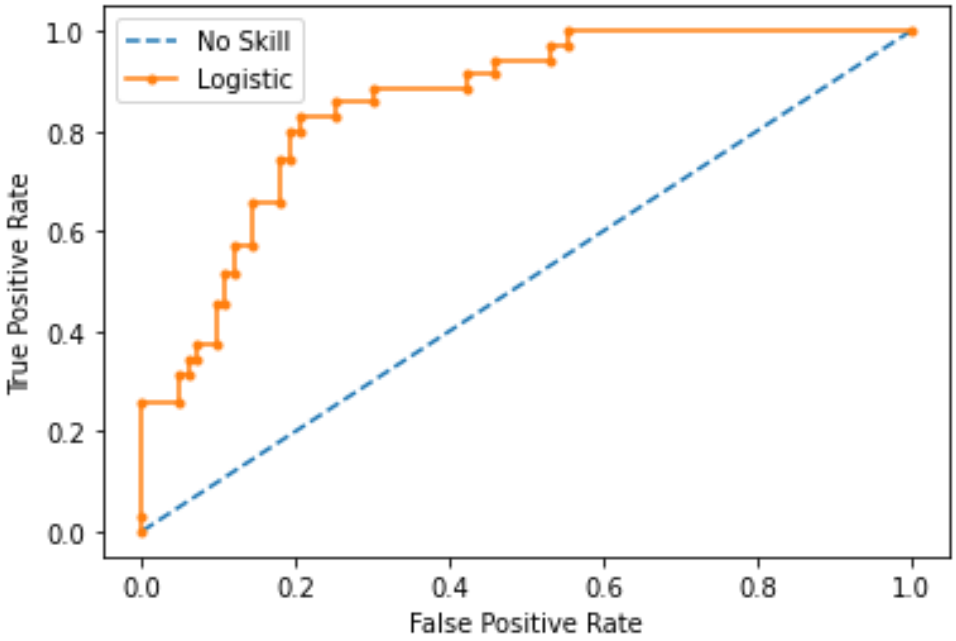

Figure 8 *AUC-ROC Curve*

There are many ways to improve our model. We faced some challenges with false positives in the model. These challenges may have been due to outliers in the dataset, which could have included individuals with high medical metrics, such as a high glucose level, although they are nondiabetic. We could have also improved our model by replacing our zero missing values with the mean or median instead of removing them from the dataset. Furthermore, additional relevant features would have also accounted for more of the data's variability, improving the strength of our model. One potential addition for future research would be to implement feature bucketing, which is converting continuous features into categorical features by dividing the range of values into different levels. Another potentially useful change would be to add new features which could be done by searching for other metrics that doctors rely on the most for diagnosis and identifying how well those features perform with the use of the logistic regression model. If possible, it would also be helpful to gather sample data from various other groups of people in order to make a model that can be used effectively to predict diabetes in a greater proportion of the population.

References

Centers for Disease Control and Prevention. (2017, January 10). *Native Americans with Diabetes - Vital Signs - CDC*. Centers for Disease Control and Prevention. https://www.cdc.gov/vitalsigns/aian-diabetes/index.html.

Centers for Disease Control and Prevention. (2020, February 11). *National Diabetes Statistics Report, 2020*. Centers for Disease Control and Prevention. https://www.cdc.gov/diabetes/library/features/diabetes-stat-report.html#:~:text=34.2%20million%20Americans%E2%80%94just%20over,1%20in%203%E2%80%94have%20prediab etes.

Centers for Disease Control and Prevention. (2020, June 11). *What is diabetes?* Centers for Disease Control and Prevention. https://www.cdc.gov/diabetes/basics/diabetes.html.

Learning, U. C. I. M. (2016, October 6). *Pima Indians Diabetes Database*. Kaggle. https://www.kaggle.com/uciml/pima-indians-diabetes-database/code.

Narkhede, S. (2021, June 15). *Understanding AUC - ROC Curve*. Medium. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.