**Diabetes Prediction**

<https://www.kaggle.com/datasets/kandij/diabetes-dataset?resource=download>

Evan Gousha, egousha@bellarmine.edu

Gabe Migliore, gmigliore[@bellarmine.edu](mailto:student@bellarmine.edu)

**ABSTRACT**

The prediction on whether a patient has diabetes now or could develop in the future could be used for prevention. Using Logistic Regression Analysis we set out to determine if certain diagnosable measurements could be used to predict it. We split the training into training, testing, and validation sets. 95% testing, 5% validation. Using Python we import the dataset using numpy and pandas. The dataset consists of several variables such as pregnancies, glucose levels, blood pressure, skin thickness, insulin levels, BMI, the diabetes pedigree function, and the age of the patient. We used logistic regression to determine the categorical likelihood insulin would soon be diagnosed. Our accuracy score was around 80% for both tests. Generally this was considered relatively accurate and the data found from this dataset and model could be insightful in determining developing diabetes in patients.

**INTRODUCTION**

Diabetes can be developed genetically or from certain behaviors. It forms when insulin doesn’t exist in your bloodstream, or the insulin is unable to bind with cells. Without insulin cells cannot receive energy and slowly die off. Our datasheet looks into whether patients have undiagnosed diabetes that can perhaps develop further in the future. It does this through several diagnosable measurements, such as pregnancies, glucose levels, blood pressure, skin thickness and more. The data was taken from Pima Indian women that are 21 or older.

**BACKGROUND**

*DATA SET DESCRIPTION*

Pregnancies: Number of pregnancies this person has had

Glucose: the average glucose concentration level of this person

Blood Pressure: the average blood pressure of this person

Skin thickness: how thick the fold of skin on tricep is in (mm)

Insulin: average level of the insulin hormone

BMI: the measurement of the Body-mass index for this person

Diabetes Pedigree Function: provided some data on diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient.

Age: How old the person is

Outcome: Whether the person had diabetes or not

*MACHINE LEARNING MODEL*

Regression model used:

Logistic Regression:Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

**EXPLORATORY ANALYSIS**

Our dataset is a relatively medium sized one, especially in comparison to our other project. Our data had 8 columns, with 7 independent variables and outcome as our dependent variable. Each column also had 768 entries, further adding to our data being a “medium sized” dataset.

*DATA TYPES*

| **VARIABLE NAME** | **VARIABLE TYPE** |
| --- | --- |
| Pregnancies | int64 |
| Glucose | int64 |
| Blood Pressure | int64 |
| Skin Thickness | int64 |
| Insulin | int64 |
| BMI | float64 |
| Diabetes Pedigree Function | float64 |
| Age | int64 |
| Outcome | int64 |

**METHODS**

*DATA PREPARATION*

We did not need to do much in regards to preparing the data for modeling. When looking for null values in each of our columns, we were unable to find any for any columns. Also when looking at the columns, all columns seemed necessary for modeling so we were unable to remove any. When looking at our variables, none were categorical so we did not have to use dummies in order to preprocess any of the variables.

*EXPERIMENTAL DESIGN*

In order to provide more accurate findings, we ran the logistic regression model twice with two different data splits to see whether one would be more accurate. The first one was an 80/15/5 split which we thought a 5 percent validation split was reasonable considering the size of our data, and that led to roughly 5 entries for validations. The second one was a 70/25/5 split using the same 5 validating entries.

*EXPERIMENTAL PARAMETER*

| **EXPERIMENT NUMBER** | **PARAMETERS** |
| --- | --- |
| **1** | Logistic Regression test using Python’s regression, 80/15/5 split for train, test, and validate |
| **2** | Logistic Regression test using Python’s regression, 70/25/5 split for train, test, and validate |

*TOOLS USED*

For this project we only had to use Python and Pandas in order to perform the logistic regression with the different validation splits, as well as matplotlib in order to plot the confusion matrix plots of all the results

**RESULTS**

*MEAN SQUARE ERROR AND R-SQUARED CALCULATION*

| **PROGRAM** | **SPLIT** | **ACCURACY SCORE** |
| --- | --- | --- |
| PYTHON | 80/15/5 | 0.8275 |
| PYTHON | 70/25/5 | 0.7968 |

*DISCUSSION OF RESULTS*

When looking at the two different models, the 80/15/5 split model had the higher accuracy score when predicting whether someone will have diabetes or not. When looking at the 80/15/5 model, there were 72 true negatives, meaning it predicted them not have diabetes and they actually didn’t. There were 6 false positives, meaning it predicted them to have diabetes and they actually didn’t. There were 14 false negatives, meaning it predicted them to not have diabetes when they actually did. There were 24 true positives, meaning it predicted them to have diabetes and they actually did. For the 70/25/5 model, we had 117 true negatives, 13 false positives, 26 false negatives, and 36 true positives. Because of the nature of the data the higher false negative rate leads us away from the 70/25/5 split and chose the 80/15/5 split as the more appropriate model for our data.

*PROBLEMS ENCOUNTERED*

When it comes to coding, not many problems were encountered that were not easily overcome by simple modifications to code. The only major problem when dealing with this data was trying to figure out what all the variables meant and then trying to gauge the accuracy of the data in the first place. Much of it was health science based information, something neither of us had much knowledge of going into it.

*LIMITATIONS OF IMPLEMENTATION*

Again, because of the nature of this data and the seriousness of dealing with diabetes, leaving one model to predict or diagnose something as serious as diabetes would not be very intelligible. Because of this, if one were to actually attempt to implement this work in the real world, much more would need to be taken into consideration and many more models and tests would need to be performed in order to provide more safe and sound results.

*IMPROVEMENT/FUTURE WORK*

In order to improve, I think just like with anything, performing more tests will always provide more accurate results so in the future I think diving into more and different models would be advantageous to our predictions. I think also having medical knowledge beforehand would probably have helped us understand the material and values better. Also testing more split sizes, such as an even higher training size.

**CONCLUSION**

For this project, we were prompted to use Logistic regression in order to predict whether a certain discrete value will happen or not. In our case, we chose a dataset that had many variables and then state whether the person developed diabetes or not, and so we used these variables to predict whether they developed diabetes or not. We used two different splits for the regression, with a 80/15/5 split and a 70/25/5 split. Our models produced relatively accurate data, but the 80/15/5 split was more accurate and had fewer false negatives so we chose that one as the more appropriate model for our data. Because we were dealing with medical data, relying on just one model is not an intelligent action to do, so if one were to actually use this method to predict diabetes, having more variables and more models would be needed to provide more accurate results. However despite the nature of the data, we believe this provided helpful insight, especially for the nature of the class and because of the high accuracy. However a higher accuracy would be needed to really apply this model.