Health Group
DATA 602
Fall Term 2020
University of Maryland

INTRODUCTION

For our group's final project, we set out to improve the accuracy and efficacy of diabetes diagnostics using a classification method, specifically, decision trees and logistic regression analysis. Our group chose this project because we all had a strong interest in healthcare, working in the field. We believed our project's outcomes would have a strong and positive impact on health care by increasing the accuracy and efficacy of diagnostics for diabetes.

BACKGROUND

Diabetes is a common and prevalent disease in the United States (CDC). With the information we gather from this project, we hope to understand the correlation between health conditions and a diabetes diagnosis. If such a correlation exists, it will allow us to determine if the healthcare industry can diagnose patients properly or if changes are needed. If our outcomes inform practice, it will enable patients to get adequate care. We also chose this project to close the often false negatives and false positives with diagnostics, making patient care difficult. Therefore, it is crucial to analyze how often this happens, so patients receive proper care regarding their diagnosis.

METHODOLOGY

Our group's architecture was Jupyter installed locally though some group members also used AWS Jupyter. Through the various architecture modalities, our group implemented many data science principles that were learned in this course, DATA 602. We learned how to deal with missing data and split data, which are fundamental principles used in data science that we will use to implement such principles in work settings.

Regarding how our group dealt with missing data, we were fortunate not to have any missing data. However, we have many anomalies, such as values labeled as zero when that value could not be zero. Some examples of these anomalies were some patients' 'Glucose,' 'Blood Pressure,' 'BMI,' and 'Skin Thickness' having zero variables when it is not possible for these values ever to have a value of zero. For example, someone cannot have zero blood pressure unless they are dead, not the target demographic for our project's population. Because of these anomalies, we had to accommodate for these values. We tried executing our code with those values having null values (empty), the values as the mean average of all the existing values for that variable, and ignoring the rows with zeros and the insulin variable altogether because our data would have condensed down significantly. When doing this, we found that our predictions have become more accurate with each plan. Our accuracy decreased by 9% for correctly predicting patients to have diabetes and decreased by 1% for predicting patients not to have it. Our false positives also had the same results: falsely expecting diabetes increased by 1%, and falsely predicting no diabetes increased by 9%. Although our accuracy for accurately predicting decreased and falsely predicting diabetes increased, this is still a good number considering we have condensed our data down significantly. We decided to remove the rows with the anomalies and insulin as our final decision.

Another principle of data science we utilized was splitting our data. Cross-validation is primarily used in applied machine learning to estimate a machine learning model's skill on unseen data. To use a limited sample to evaluate how the model performs in general when used to make predictions on data not used during the model's training.

Process:

1. Shuffled the dataset randomly
2. Split the dataset into five groups. (k=5)

3. For each unique group,

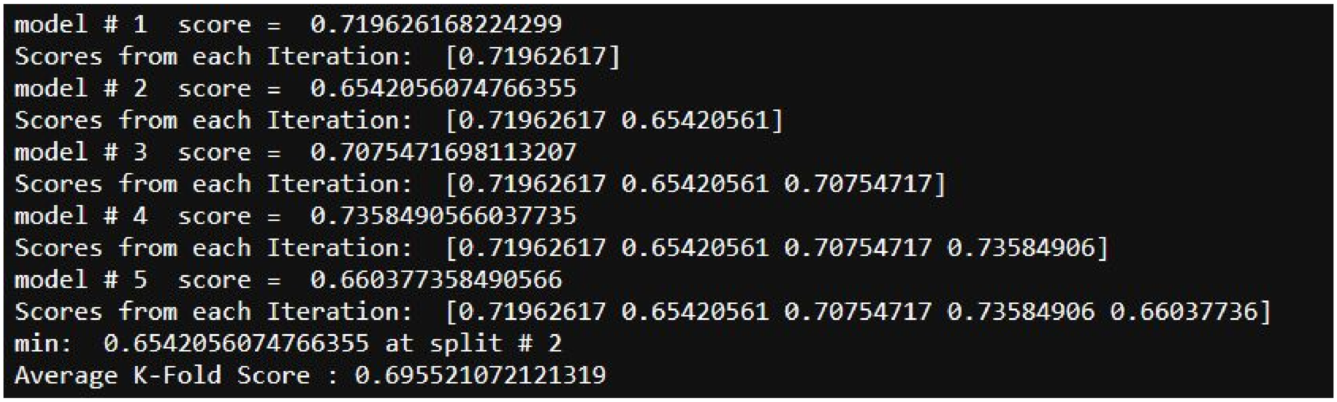
   I. Take that group as a test data set and the remaining groups as training data set

   II. Fit them into a model

   III. Get the evaluation scores

4. Consider the test and train data from the split with the least score

OUTPUT



```
model # 1  score =  0.719626168224299
Scores from each Iteration:  [0.71962617]
model # 2  score =  0.6542056074766355
Scores from each Iteration:  [0.71962617 0.65420561]
model # 3  score =  0.7075471698113207
Scores from each Iteration:  [0.71962617 0.65420561 0.70754717]
model # 4  score =  0.7358490566037735
Scores from each Iteration:  [0.71962617 0.65420561 0.70754717 0.73584906]
model # 5  score =  0.660377358490566
Scores from each Iteration:  [0.71962617 0.65420561 0.70754717 0.73584906 0.66037736]
min:  0.6542056074766355 at split # 2
Average K-Fold Score : 0.695521072121319
```

It is important to note that we split our data to be 75% test data and 25% train data. We initially would split 50%, 50%, because it was easier than creating a new data set. However, this did not make the most sense in practice, so we decided to split our data this way instead. We got a relatively better 'least model score' with the 75%, 25% split.

The confusion matrix describes the performance of our classification model. There are two possible predicted classes, 0 and 1: predicted class 1 represented the presence of diabetes. Predicted class 0 conveyed that the patient does not have diabetes. The classifier made a total of 98 predictions. Out of those 98 cases, the classifier predicted yes 30 times and no 68 times. It showed that 34 of the patients had diabetes, also known as actual yes, and 64 did not have diabetes, also known as real no. The confusion matrix explained a few things. It explained that 25% of the time, when we predicted patients had diabetes, they did not have it, and 75% of the time, when we predicted patients did not have diabetes, they didn't have it. We also saw that 41% of the time, when we expected the patients did have it, they did have it. In other words, our model is very good at predicting if a person is not at risk of having diabetes.

CALCULATIONS

Accuracy: Overall, how often is the classifier correct?

TP+TN/Total = 0.6326530612244898

Precision: When the classifier predicts yes, how often the classifier is correct?

TP/TP + FP = 0.4666666666666667

Negative Predictive Value: When the classifier predicts no, what is the probability that the patient really does not have the disease?

TN/TN + FN = 0.7058823529411765

Sensitivity (Recall): When it's actually yes, how often does the classifier predict yes?

TP/TP+FN = 0.4117647058823529

Specificity (True Negative Rate): When it's actually no, how often does the classifier predict no?

TN/TN+FP = 0.75

False Positive Rate: When it's actually no, how often does the classifier predict yes?

FP/FP+TN = 0.25

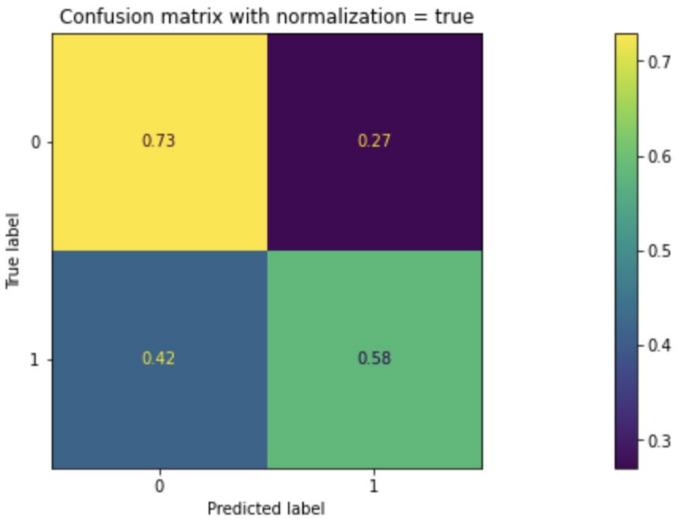
Misclassification Rate (Error Rate): Overall, how often the classifier is wrong?

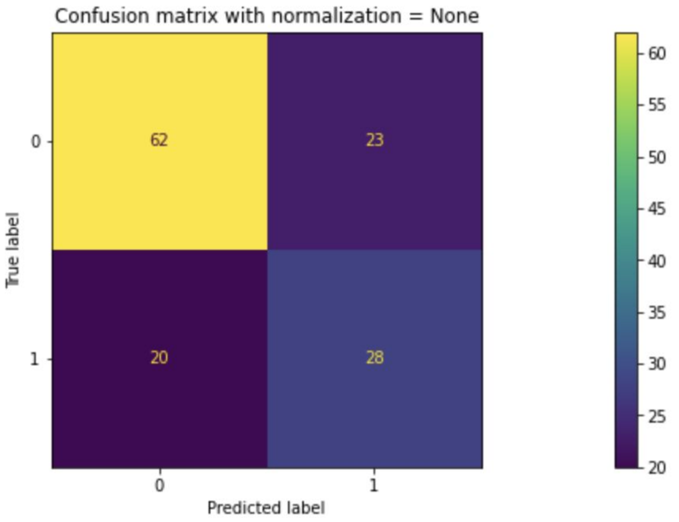
FP+FN/Total = 0.367346939

Prevalence: How often does the yes condition occur in the sample?

FN+TP/Total = 0.346938776

CONFUSION MATRIX





The logistic regression algorithm is used to fit the machine learning model and make predictions on new data instances that the model has not seen. We chose a logistic regression because our outcome variable was categorical, 0 for not having diabetes, and 1 for yes to having diabetes. Based on these criteria, we could not choose a linear regression. Our independent variables were pregnancy, glucose, blood pressure, skin thickness, BMI, diabetes pedigree function, and age. The accuracy of our logistic regression model on the test set is 74%.

TEAM MEMBERS AND CONTRIBUTIONS

Cooper Kidd: Project management, chose regression analysis, assisted with paper

Abbas Mohammad: Assisted with anomaly detection and handling, K-folds cross validation and decision tree

Ridgely Franklin: Wrote up the notes and the paper

Melody Dastanlee: Wrote up the notes and the paper

Shrey Patel: Anomaly detection and handling, data cleaning, k-folds cross-validation, decision tree

Shrey Nair: Picked the topic, added the idea of showing descriptive statistics for the dataset, assisted with the paper to add technical information.

Thu Cao: Helped with coding for plotting histogram, paper edits, shared insights on dataset, implemented logistic regression

Aina Ahanmisi: Helped with data cleaning and decision tree. Also assisted with paper and edits, assisted in picking and finalizing topic, matrix analysis

Renea Young: Helped with paper, wrote notes on the confusion matrix, calculations and logistic regression

CHALLENGES

Something that posed a challenge was determining how to handle the anomalies. Some variables had values of 0; some had over 300 fields that had 0 as a value. These anomalies posed a challenge for our group, as these values affect our outcomes. We debated between leaving the values as zero or filling the values in with the mean average for the variable, which could skew our accuracy and results, or not using the data with missing variables. We overcame this challenge by running our code for each situation and determining how it affected the outcome. In the end, we chose to run our data without the lines with anomalies because this way, we would not have to confirm if the missing data were not missing at random (NMAR), which could pose a challenge. Another challenging aspect of this project was determining how to create the test and train groups, which we eventually decided on based on the information we learned in class and project research.

Another challenge for us was choosing a dataset. Choosing a dataset was a challenge due to the narrowing down of our topic to a single health issue and finding a dataset that would be easy to input into Jupyter and work within the project's constraints. What was interesting about choosing a dataset was the number of datasets that there were to select. We knew that our group wanted to look into health-related data, which was a useful starting point, but beyond that, we did not know what direction we wanted to go in. So the sheer number of datasets was something our group found very interesting. Another thing about choosing a dataset that we found interesting were the styles that the many databases took. Some databases were very robust, while others were very slim because when manipulating data, the more data one has, the better.

AREAS FOR GROWTH

Our group would suggest having someone on the team who works with health data. Another person who might be helpful is someone who works with this type of data every day, such as CINs & ACOs. It is essential to have people on the team who have experience working with this type of data and projects because they can contribute most regarding the analyses of the data. We would also suggest using a data set with minimal missing values and intuitive to make any research easier.

DESCRIPTIVE STATISTICS

The describe() method is a quick way to get the usual univariate summary information. We are looking for anything unusual or unexpected at this stage, perhaps indicating a data entry error.

	Skewness	**Min Value**	**99% quartile Value**	**Max Value**
Pregnancies	mean > median Right skewed	N/A	13	17 (outlier)
Glucose	mean > median Right skewed	0	Values close to each other	
Blood Pressure	mean = median Gaussian Distribution	0	N/A	122
Skin Thickness	mean(0.47) > median (0.37) Right skewed	0	51	99 (outlier)
Insulin	mean(79) > median(30) Right skewed	0	519	846
BMI	mean(31) =~ median(32)	0	50	67

	Gaussian Distribution			
DPF	mean(0.47) > median (0.37 Right skewed	0.078	1.69	2.42
Age	mean(33)>median(29) Right skewed	21	67	2.42

ANOMALIES

Zeros ->

Total rows with zero glucose: 5

Total rows with zero blood pressure: 35

Total rows with zero Skin Thickness: 227

Total rows with zero BMI: 11

Total rows with zero Insulin: 374 # not used as a feature due to high volume of zeros / bad data

Outliers (excluding 0s) ->

column: Pregnancies : # of outliers: 4

column: Glucose : # of outliers: 0

column: BloodPressure : # of outliers: 10

column: SkinThickness : # of outliers: 1

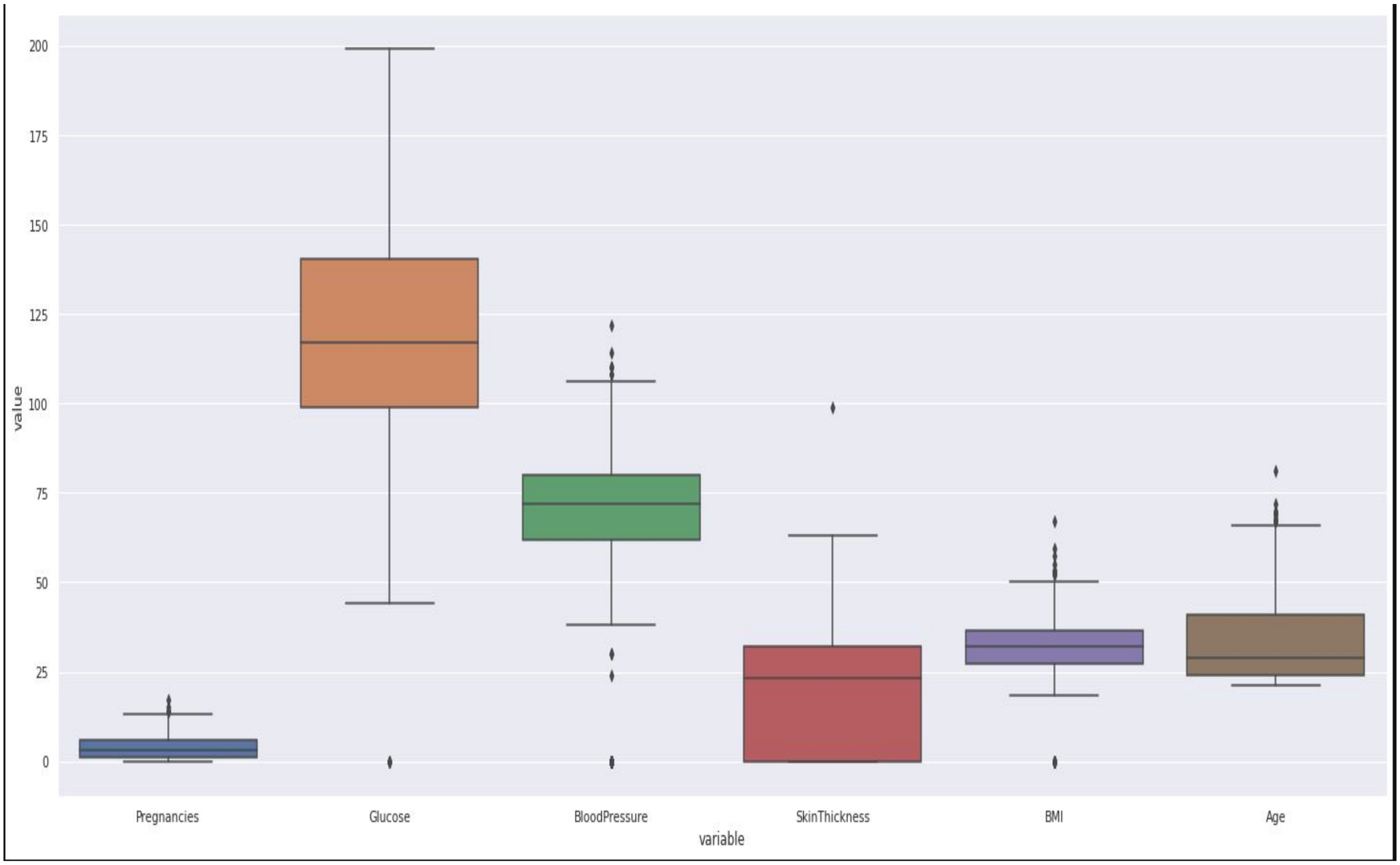
column: Insulin : # of outliers: 34

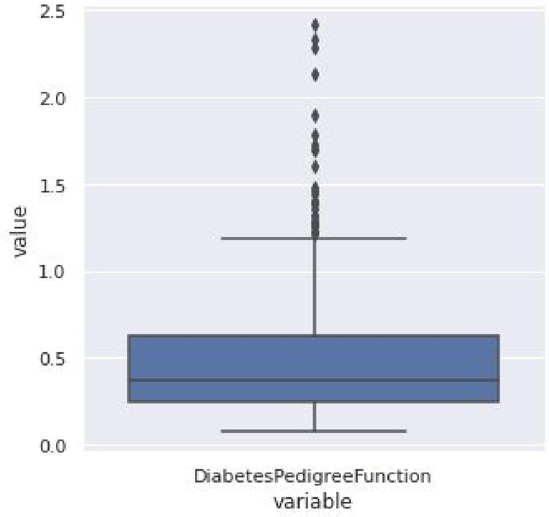
column: BMI : # of outliers: 8

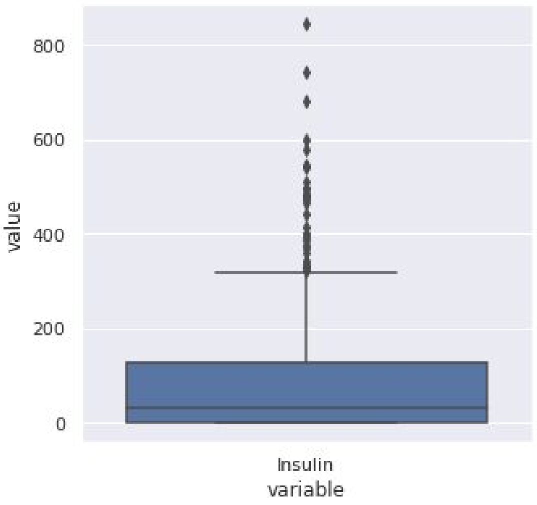
column: DiabetesPedigreeFunction : # of outliers: 29

column: Age : # of outliers: 9

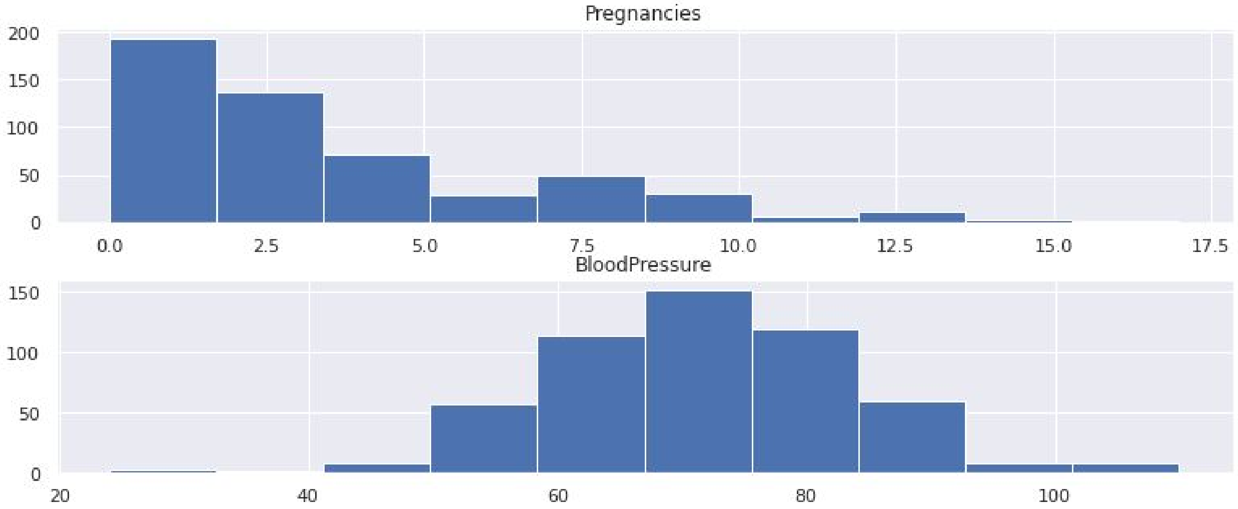
BOX PLOTS FOR ANOMALY DETECTION

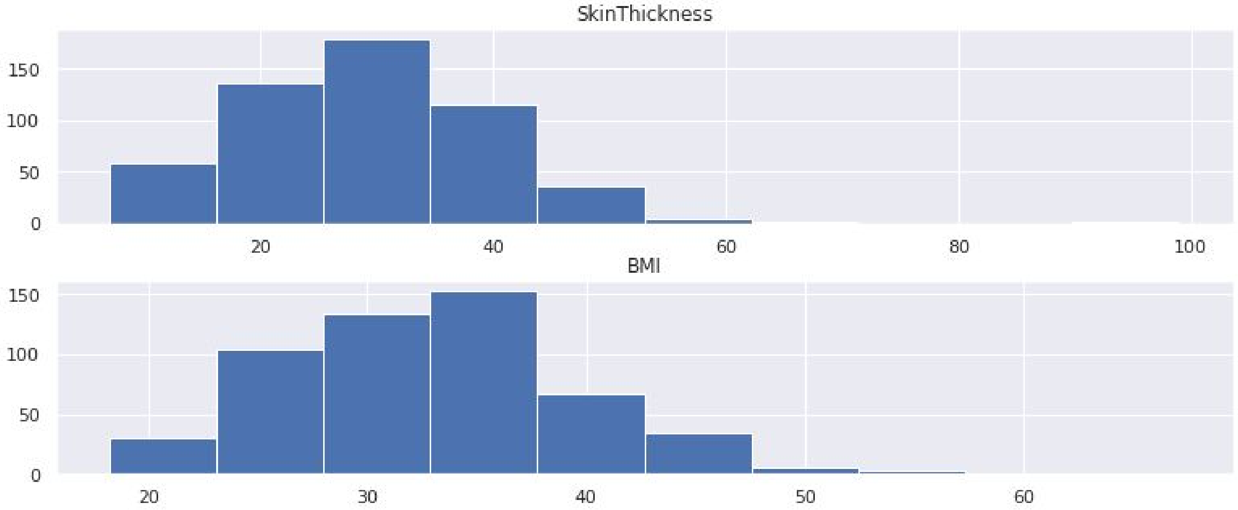


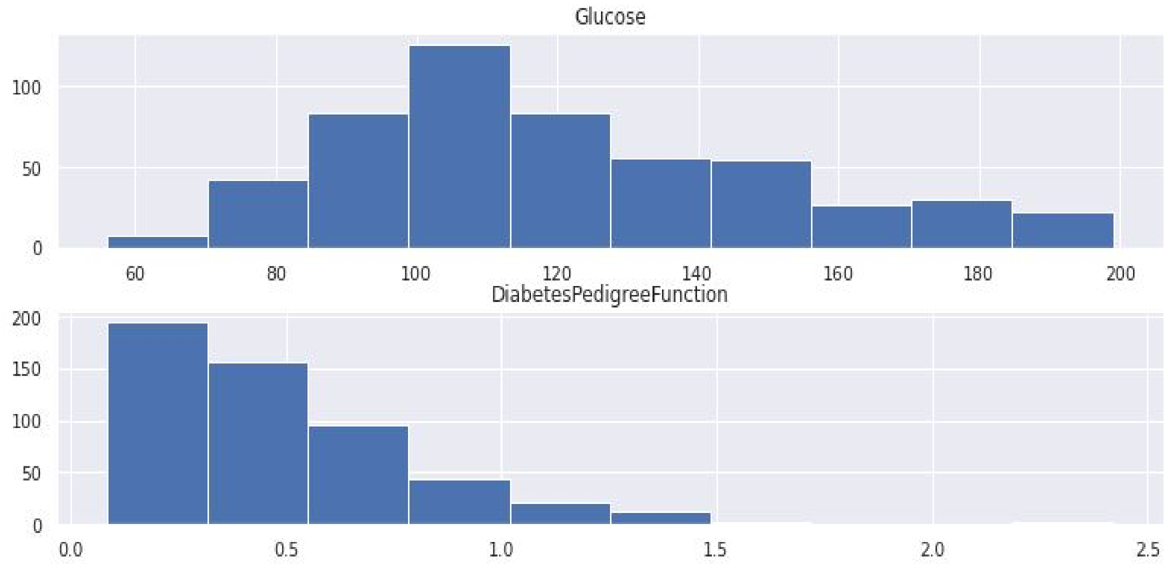

2.5
2.0
1.5
1.0
0.5
0.0
value
DiabetesPedigreeFunction
variable

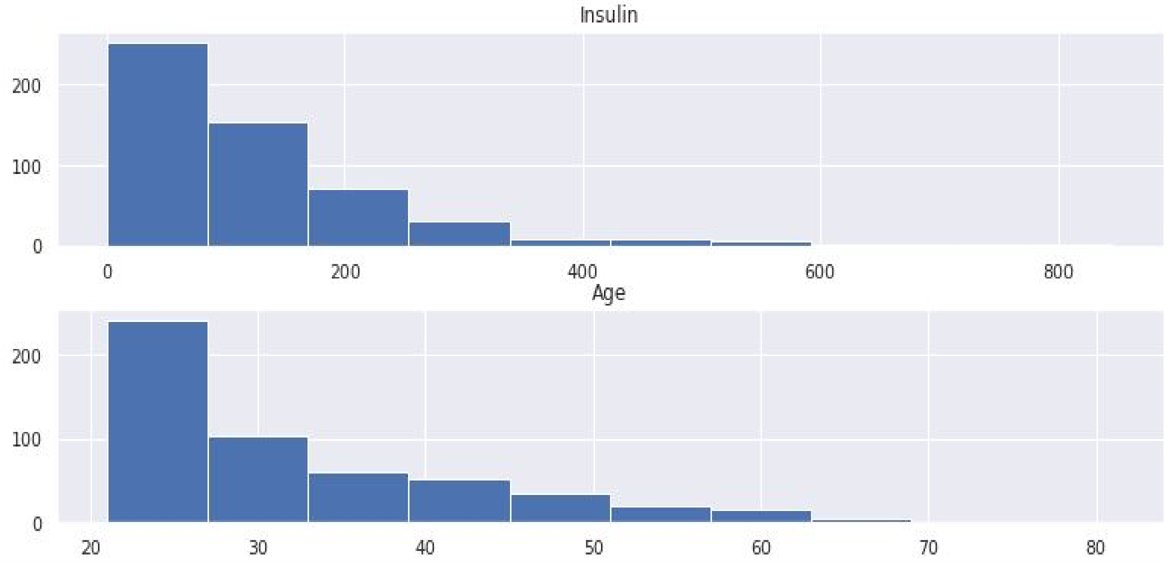

800
600
400
200
0
value
Insulin
variable

HISTOGRAMS AFTER HANDLING ANOMALIES









CONCLUSION

We learned that there is indeed a strong correlation between health conditions and diabetes. We also found that skin thickness plays a significant deciding factor in predicting according to the decision tree. There is always room for improvement in the healthcare industry to become better, but our model has shown that it can currently diagnose patients correctly. In terms of the model, we can learn that it is better to predict true negatives than true positives; therefore, our model had more specificity than sensitivity (recall). Specificity is when there is a true negative or actual no, and sensitivity is when there is a true positive or actual yes. In terms of our project, specificity would be the ability to predict a negative diabetes diagnosis and have a negative diabetes diagnosis. Sensitivity would be the ability to predict a positive diabetes diagnosis and have a positive diabetes diagnosis. Our model demonstrates that 75% of the time, there was a true negative diagnosis, and 41% of the time, there was a true positive diagnosis. We also learned that box plots are an excellent tool for demonstrating anomalies.

REFERENCE SECTION

Centers for Disease Control and Prevention. National Diabetes Statistics Report, 2020. Atlanta, GA: Centers for Disease Control and Prevention, U.S. Dept of Health and Human Services; 2020.