Machine Learning to Identify Patients with Diabetes or Prediabetes

By Naama Bejerano

Introduction

Diabetes is a serious chronic disease in which individuals cannot effectively regulate glucose levels in the body. As a global health concern with significant impacts on individuals' quality of life, the national healthcare system, and economies, early detection and intervention is an essential tool in managing and mitigating its effect. I created a predictive model that uses a binary neural network classification algorithm to sort individuals into two categories: (non-diabetic or diabetic/pre-diabetic) based on various health indicators in the individual. The input to our algorithm is a series of health indicators, including but not limited to whether the individual has high blood pressure, whether they have healthcare, their age, and mental health score. I then used a binary neural network classification algorithm to output a predicted class between either non-diabetic and pre-diabetic/diabetic. Through a reliable, non-invasive tool for early detection of diabetes, I hope to significantly enhance patient outcomes and reduce healthcare costs. Of the 30.3 million Americans with diabetes, 7.2 remain undiagnosed because not everyone undergoes regular screening for high blood sugars, so this model can help to manage and mitigate effects (CDC).

Related Work

Diabetes affects over 500 million people globally and has a growing impact over health care systems worldwide. Because of all of the additional health concerns that accompany a diabetes diagnosis, scientists have been attempting to integrate machine learning algorithms into diagnostic approaches in various other ways. In a predictive model for early type 2 diabetes detection in Nigeria, the authors created a multivariate logistic regression model.[1] This study initially conducted a univariate study on each of the individual features to investigate the sole relationship between the feature and diabetic diagnosis. Then, they conducted a multivariate study to comprehensively analyze the diagnosis of each individual. I believe the initial univariate study is extremely clever in allowing the authors to intuitively understand the weight that each feature had over the classification. In an additional study on the application of machine learning for early detection of type II diabetes, they developed five of the most popular algorithms used in binary classification problems to determine the most accurate approach to diabetic diagnosis.[2] Overall, the K-NN model achieved the highest accuracy of 86% and proved to be easy to implement, however, it was extremely sensitive to outlier data points. The comparison between different algorithms shows meticulous work to optimize the accuracy of the machine learning classification. Lastly, the multifaceted nature of diabetes enables it to cause a plethora of other complications within individuals. One study investigated the relationship between data on diabetic patients and other complications including metabolic syndrome, neuropathy, and hypertension. I believe this study to be incredibly useful in additional development of diabetic classification because these complication factors can also be used in the initial classification. By analyzing health factors that result from a diabetes diagnosis, the study can also investigate how these complications could appear earlier in the diagnosis. This can provide a more comprehensive approach to the classification with extremely concrete features in the individuals.

Dataset

The dataset comes from The Behavioral Risk Factor Surveillance System (BRFSS) collected annually as a health related survey from the CDC, containing 253,680 rows of input (number of people as data points).[4] It outlines the health related risk behaviors, chronic health conditions, and foundational information associated with each data point as well as whether the individual is non-diabetic (0), pre-diabetic (1), or diabetic (2). In the pre-processing, I

combined the output y-values of classification groups 1 and 2 (diabetes and pre-diabetes). This enabled us to create an ultimately binary classification and flag *all* potential concerns for diabetes including the initial stages of the disease. For features such as high blood pressure, the data indicates a 1 for having high blood pressure and a 0 otherwise. Most of the features were able to be classified binarily as either yes (1) or no (0). However, for features that exist on a continuous scale, such as Body Mass Index, the data indicates the exact number. In our final model, there were 182,650 training data samples and 71,030 testing data samples, reflecting a 72% and 18% split respectively. The remaining 10% of the dataset was used in the validation set.

Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke
0.0	1.0	1.0	1.0	40.0	1.0	0.0
0.0	0.0	0.0	0.0	25.0	1.0	0.0
0.0	1.0	1.0	1.0	28.0	0.0	0.0
0.0	1.0	0.0	1.0	27.0	0.0	0.0
0.0	1.0	1.0	1.0	24.0	0.0	0.0
0.0	1.0	1.0	1.0	25.0	1.0	0.0
0.0	1.0	0.0	1.0	30.0	1.0	0.0
0.0	1.0	1.0	1.0	25.0	1.0	0.0
2.0	1.0	1.0	1.0	30.0	1.0	0.0
0.0	0.0	0.0	1.0	24.0	0.0	0.0

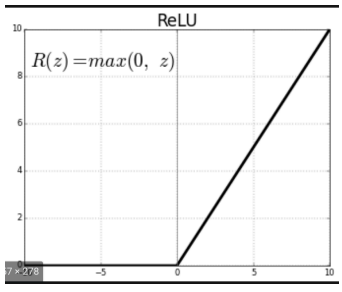
The first 10 rows of the dataset, outlining the numbers used to indicate different features about each patient.

Methods

The original algorithm I started with was a multiclass neural network classification algorithm. With three predicted classes 0, 1, and 2 corresponding to no diabetes, pre-diabetes, and diabetes respectively. Multi-class neural network classifications incorporate hidden layers connected to the input data. Then, the output layer synthesizes the outputs from the input layers to provide the final classification as either a 0, 1, or 2. This neural network has two layers that both use the ReLU activation function and the softmax function to classify the data.

The ReLU activation function works with positive input values, so a negative input value would result in an output of 0. The output has a range of 0 to infinity and it allows the learning algorithm to understand the complexities of the data better than a sigmoid function, which has a range of 0 to 1 and poses problems for outputs that are outside of that range.

ReLU Activation Function & Equation



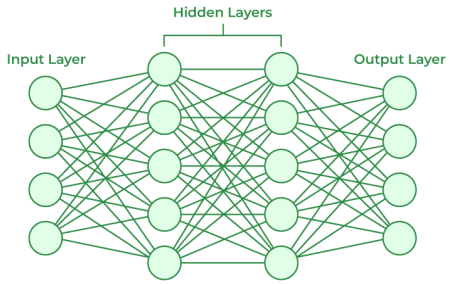
The softmax function transforms the raw outputs from the hidden layer ReLU activation equations into a vector of probabilities which gives a probability distribution over the classes. Because the output of our neural network is a classification, the function categorizes the inputs into one of three classes depending on the features associated with it.

Softmax Equation

$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

As a comparison point, I made a binary logistic regression model with two classes 0 and not 0 categorizing people as non-diabetic or either pre-diabetic or diabetic. This simple model is used to highlight the differences between the logistic regression model and the final neural network, which better fits the data. Logistic regression creates a linearly separated classification. This can be more difficult to fit to complex data, especially one with 21 features, since the decision boundary is linear and it struggles with high variance problems. As a result, I used a neural network to classify the data which is more flexible. The neural network takes in inputs through the input layer and analyzes the data through one or more hidden layers to curate the output.

Neural Network

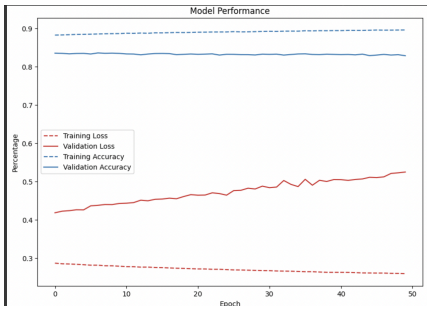


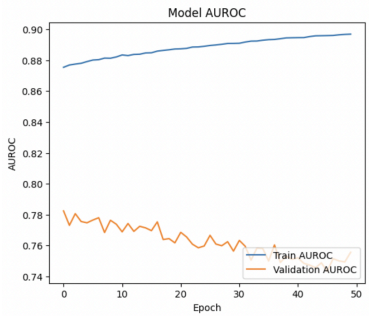
The final algorithm I have is a binary neural network classification algorithm. With two classes 0 and not 0 categorizing people as non-diabetic or either pre-diabetic or diabetic (reasons for this choice explained in methods). Here I also included cross validation into the model to help the model generalize and more accurately measure the models performance. I did not use mini batch gradient descent, which means that the models internal parameters are updated after each batch is processed as opposed to the entire data set. I chose to not use this as it would decrease the performance of the model and in this case with a model that is not being trained very often the need to reduce computational power is not necessary. This finds a balance between computational efficiency and getting precise updates for gradient descent. L1 regularization was not used since the model does not overfit the data therefore including L1 regularization unnecessarily reduced the performance of the model. The equation for L1 regularization is $L = L_0 + \lambda \sum_{i=1}^{n} |w_i|$ where L_0 is the original loss function λ is the regularization parameter and w_i represents the model's weights. I used AUROC which measures how well a classification model is working, by reflecting how much better the model is at classifying a test case than a random classifier. This measure of performance better fits the data than accuracy or an F1 score since this measure of performance accounts for the high probability of class 0 in our data set that allows for high accuracy with a model that is simply random or assigns all data points to class 0.

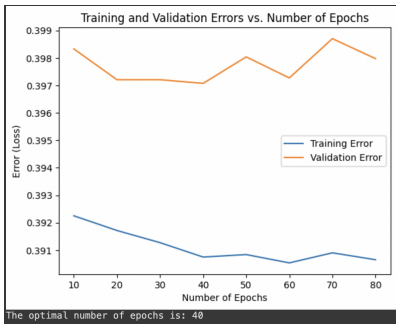
Experiments/Results/Discussion

Our initial neural network had two hidden layers, with widths 25 and 15, respectively. The activation function is ReLU. In the final layer, Softmax was used to produce probabilities over three classes. Performance was measured using loss, accuracy and F1 score (Test Loss: 0.398, Test Accuracy: 0.845, F1 Score: 0.389). After building a confusion matrix and further looking at the data I found that the model was never predicting a label of 1, corresponding to pre-diabetes. This makes sense since class one is only 1.8% of the data making it too small to be able to categorize well. I decided to combine classes 1 and 2 so that the model classifies into two options either class zero or not class zero since class one is 1.8% of the data, class zero is 84% of the data, and class 2 is 14% of the data. Before making the decision to combine the one and two classes, I ran extensive testing to determine the learning rate, number of layers, number of neurons, and epochs (seen at the bottom of the code) plotting the performance of the model with each variation. The results were 3 hidden layers, .001 learning rate, 40 epochs, widths of 10, 20, and 30 (for each layer respectively), ReLU activation functions, and softmax classification function. However, I ultimately did not use these results as our final model was a binary classification neural network.

I created a simple logistic regression model for the data to be able to compare the performance of the different models. The logistic regression model worked as a binary classifier, same as the final neural network. The model has a Training Accuracy: 84.81%, Training AUROC of 81.76%, Validation Accuracy of 84.71%, Validation AUROC 81.57%, Testing Accuracy of 84.85%, Testing AUROC: 81.82%.







For the rest of the process I used a binary classification neural network using Area Under the Receiver Operating Characteristic to measure the performance of the model since AUROC as opposed to accuracy to measure performance, to get a better understanding of how well the model was performing. I made sure to stratify the data before splitting into testing, training, and validation to ensure the classes were evenly distributed in each section of the data. The data was split to use 18% on testing, 72% on training, and 10% on cross validation. I ran extensive testing of different amounts of layers, learning rates, and epochs analyzing model performance after each version of the model to settle on the best model. The best model has 2 layers (both with a width of 150), ReLU activation functions, softmax function for dictating the final classes, 40 epochs, and a learning rate .001. I decided not to use L1 regularization. Further, I found that doing mini batch gradient descent improved issues with overfitting, which

led to the validation and training AUC to be significantly lower than the training AUC, specifically with a size of 150 which was found through testing. Following the training I plotted the performance of the training, validation, and testing using a number of factors however focusing on the AUROC as the primary measure of performance. I followed up with k-fold cross validation with a k value of 5 as this yielded the best performance and was thorough. Ultimately, our model has a training accuracy of 90.17%, training loss of 24.77%, training AUC 90.52%, validation accuracy of 83.07%, validation AUC of 79.39%, and test AUC of 79.04%. For this project I focused on performance while training to ensure the model worked well as opposed to the testing performance due to the skewed nature of the data.

Conclusion/Future Work

The report comprehensively analyzed several health indicators to determine the classification of an individual as either non-diabetic or pre-diabetic/diabetic. With a total of 21 features, the neural network algorithm was most successful in fitting the complex data with a wide range of factors. It was crucial for the algorithm to distinguish between heavily influencing features on the final classification as opposed to background information which may not be as strongly indicative of the diagnosis. The neural network was able to synthesize all of the features into a concise binary classification that would allow for early detection and easier diagnosis in diabetic patients.

With additional resources, including more time, more team members, and computational aid, I would investigate the more nuanced classification of pre-diabetes. The dataset contained only 1.8% of individuals who were classified as pre-diabetic, making it extremely difficult for the algorithm to tease out the differences between labels. As a result, I would like to collect more data on pre-diabetic patients to facilitate the algorithm's ability to identify pre-diabetic indicators. Considering the severe implications of a diabetes diagnosis, early detection in pre-diabetic patients is arguably the most important and could stifle the development of the disease instead of attempting to mitigate damages on the backend.

Works Cited

[1]Iparraguirre-Villanueva, Orlando et al. “Application of Machine Learning Models for Early Detection and Accurate Classification of Type 2 Diabetes.” *Diagnostics (Basel, Switzerland)* vol. 13,14 2383. 15 Jul. 2023, doi:10.3390/diagnostics13142383

[2]Ojurongbe, Taiwo Adetola et al. “Predictive model for early detection of type 2 diabetes using patients' clinical symptoms, demographic features, and knowledge of diabetes.” *Health science reports*vol. 7,1 e1834. 25 Jan. 2024, doi:10.1002/hsr2.1834

[3]Y. Jian, M. Pasquier, A. Sagahyroon and F. Aloul, "Using Machine Learning to Predict Diabetes Complications," 2021 4th International Conference on Bio-Engineeringfor Smart Technologies (BioSMART), Paris / Créteil, France, 2021, pp. 1-4, doi: 10.1109/BioSMART54244.2021.9677649. keywords: {Hypertension;Supportvector machines;Obesity;Retinopathy;Biological system modeling;Clusteringalgorithms;Predictive models;Diabetes Prediction;DiabetesComplications;Supervised Learning},

[4]Teboul, Alex. “Diabetes Health Indicators Dataset.” Kaggle, 8 Nov. 2021, www.kaggle.come/datasets/alexteboul/diabeteshealth-indicators-dataset/data.