Comparative analysis on efficacy of machine learning models in predicting type 2 diabetes

Tanvi Das

Abstract:

Different indicators and symptoms of diabetes, although researched, are always necessary to understand. According to the World Economic Forum's report in 2019, approximately 463 million people worldwide, aged between 20 and 79, were affected by diabetes. This number is projected to increase to 700 million by the year 2045. In the Americas, around 11.3% of the population is diagnosed with diabetes, followed by the Middle East with the next highest percentage. The goal of this paper is to investigate and model the different predictors of diabetes using mathematical and machine learning methods to get a better understanding of the disease and to propose a process that can be used by hospitals and practitioners to predict diabetes and intervene before its onset. A detailed review of all algorithms, models, and procedures can be seen below. The purpose of this study was to determine if there was a significant difference in the performance of various models, namely support vector machine (SVM), linear classifier, decision tree classifier, and artificial neural network for diabetes prediction. The comparative results show that the decision tree model outperforms both SVM and linear classifier; the decision tree gave a classification accuracy of 76.66%, the SVM gave a classification accuracy of 75.33%, and the linear classifier gave an accuracy of 67.67%. This result indicates that the accuracy of decision trees and SVM is better than linear classifiers for predicting diabetes in a patient population.

Introduction

Diabetes, affecting about one in ten adults, is a chronic health condition that involves the human body's ability to make energy and insulin. Type 1 diabetes is diagnosed when the pancreas makes little or no insulin, while Type II diabetes, the more common of the two, affects the way in which the body produces glucose. Diabetes is prevalent in 40 million and 13% of Americans along with 463 million people worldwide and has well-established risk factors including BMI, weight, and age, that can help predict its presence. Although many factors can be used to predict one's diabetic health, the search for new

breakthroughs, observations, and findings in biological research is never-ending. The purpose of this study is to explore the different indicators that work hand-in-hand to diagnose potential patients with Type I or Type II diabetes using mathematical modeling and machine learning. Using linear regression and neural networks, along with several other machine learning methods, it can be concluded that certain attributes and machine modeling techniques are better at predicting diabetes than others. Currently, machine learning modeling is used to predict patients' probability of having diabetes. This, at scale, has proven effective in accurately predicting diabetes for large populations. The goal is to arrive at a reasonably accurate model that is performant in a clinical setting and can assist doctors to predict a percentage of at-risk patients before the onset of diabetes and treat those specific patients. Other papers, such as Julia Noguez's systematic review of type 2 diabetes, have also discussed machine learning techniques for predicting diabetes [1]. This paper focuses on features included in the datasets used to create models, the optimal machine learning technique to create a predictive model, and the optimal validation metrics to compare the models' accuracy. The procedures of Noguez's research were replicated and we also found key features of the data sample and compared each model. In fact, Noguez concludes that "the structure of the dataset is relevant to the accuracy of the models, regardless of the selected features that are heterogeneous," which was also found to be the case from our analysis. However, this paper builds on Noguez's research by using additional classifiers and models. Another related work used for background and inspiration was Rajiv Singla's Artificial Intelligence/Machine Learning in Diabetes Care [2]. It explains the definition of AI and ML and their usefulness as techniques in helping to manage chronic diseases, especially diabetes. It highlights different aspects

of diabetes care using AI/ML and provides examples of studies and research on AI algorithms used in the management of type I and type II diabetes. The document also acknowledges the limitations in the applicability of AI/ML approaches in India and suggests the need for data collection efforts and research. Singla explains the concept of reasoning from knowledge and its components, presenting a detailed review of the role of AI in diabetes management and the potential benefits it could bring. The idea of using machine learning for patient management in Singla's paper is also present in this research but with a focus on specific predictors and the correlation between them.

Data

The data used in this report came from the Kaggle Diabetes Health Indicators Dataset, which is made up of reported patient statistics on their pregnancies, glucose, blood pressure, skin thickness, BMI, and their diabetes diagnosis [3]. The Diabetes Dataset Modeling objective "is to predict based on diagnostic measurements whether a patient has diabetes." Kaggle cites the source of this dataset as the *National Institute of Diabetes and Digestive and Kidney Diseases*. The original dataset had 768 patient records with nine columns of indicators. After reviewing the data, it could be seen that some patient records had null results in certain indicators, meaning.... For example, in Figure 1, patients 0, 1, and 2 show an insulin level of 0. Assuming this lack of data was simply not recorded for those patients, the data was used as is.

Figure 1: Snapshot of Patient Data

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

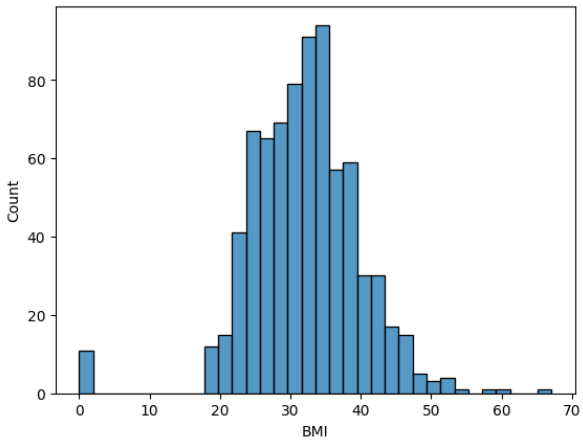


Figure 2: Age Distribution of Sample

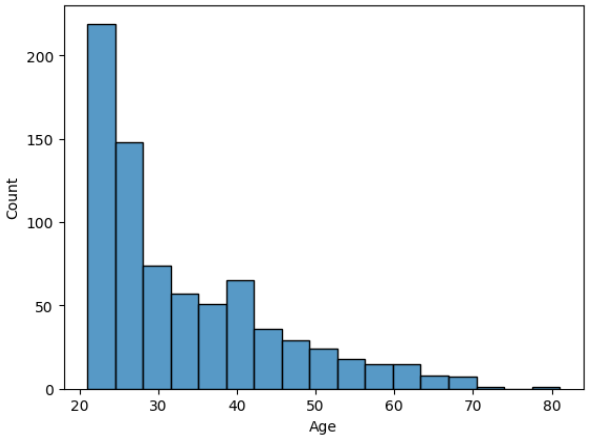


Figure 3: BMI distribution of Sample

The distribution in Figure 2 shows patients' groupings by age. In further analysis, it was noteworthy to compare the age distribution of the data set with census data available on census.gov [4]. It was found that the study dataset had

a lot more patients (64%) in the age group below 35 versus the current distribution of (45%) for the US population. So, it should be noted that this data should be used only for model evaluation rather than to predict actual diabetes outcomes for age groups not represented uniformly in this set. For future experimentation, sampling a more representative dataset of the population could make the predictions more useful for that population. For example, predictions for a population of children should be sampled from a similar distribution of child data.

The data in Figure 4 shows a range of glucose levels that are centered mostly around 100, while the average glucose level is usually between 70 to 130. There are a few glucose levels at 0, so we followed the process of imputing the missing data using the mean average of the available data. Figure 5 shows the distribution of insulin data. The first spike of distribution indicates that many patients in this sample had a low to normal insulin level. However, the data also shows patients with insulin levels above 200.

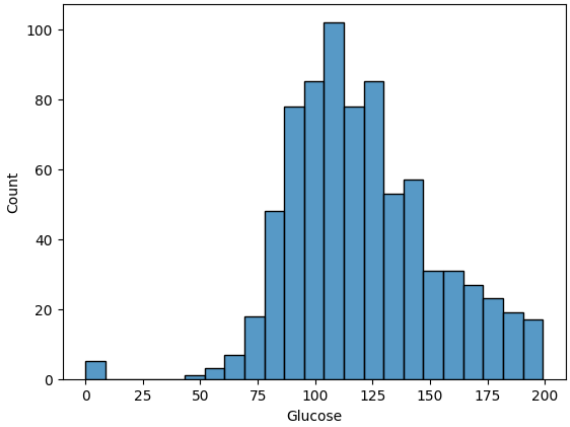


Figure 4: Glucose Data

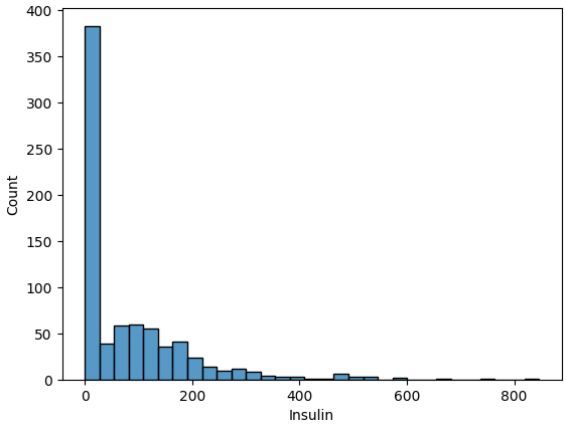


Figure 5: Insulin Data

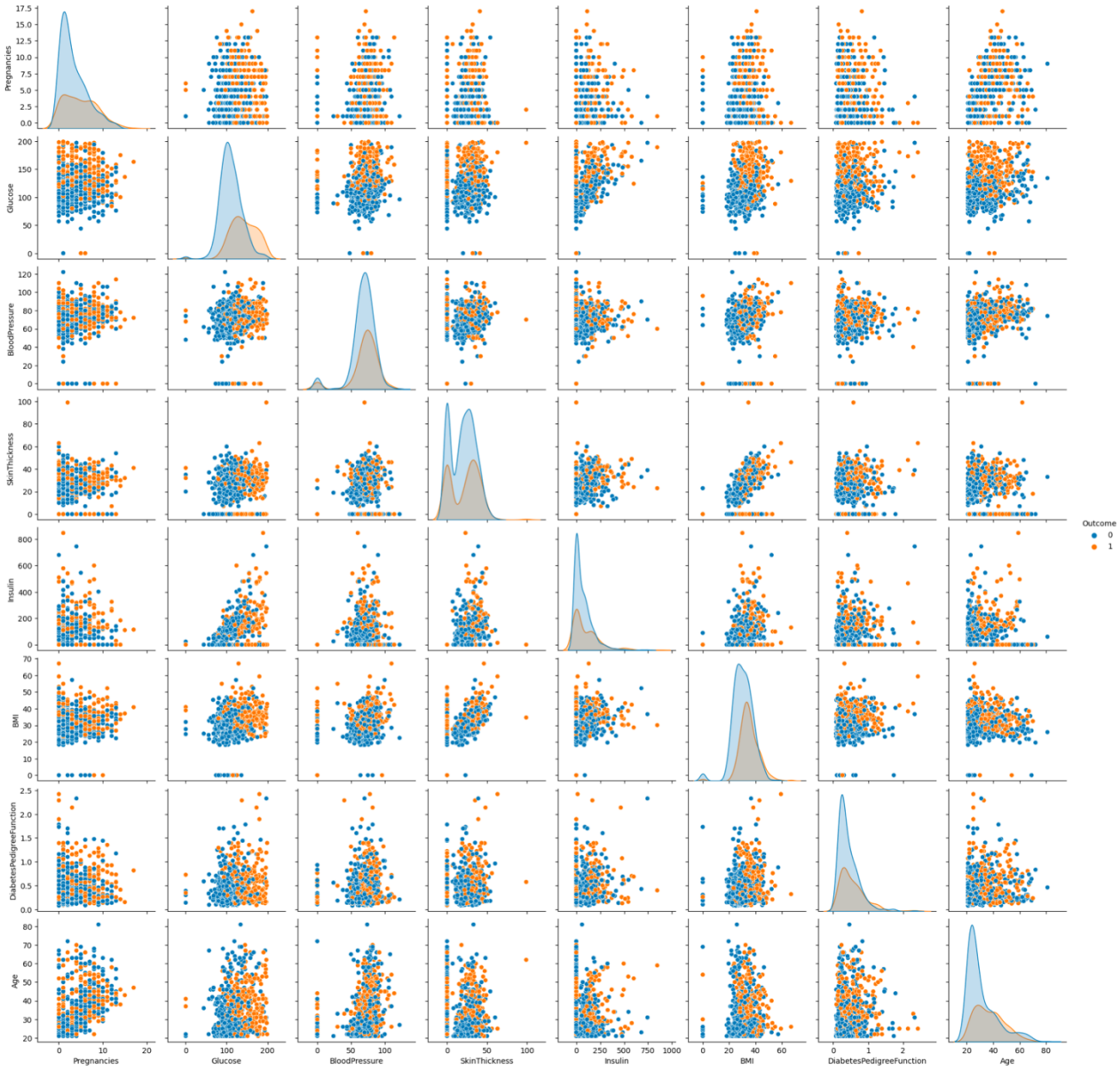


Figure 6: 2 Parameter Data Pair Plots- This gives us a visual representation of how each pair of features is related to one another along with the diabetes classification of the patient for that particular datapoint. The orange coloring represents those with diabetes and the blue represents those without diabetes. **Training and Evaluation**

A testing accuracy score is one that evaluates the number of correct predictions made by a model in comparison to the total number of predictions made. In this case, the accuracy was measured by dividing the number of correctly diabetes-diagnosed patients by the total number of patients. Separating the data

into a training set and a test set is a method used to evaluate the accuracy of the model itself. It is called a “split” as the data is split into 80% training and 20% testing, as seen in the code below:

Overfitting

Overfitting is a machine learning problem in which the model gives accurate predictions for the training data, but not test data. A model can't be considered reliable when this happens as the model cannot generalize to provide useful predictions to new data and only models the training data. Overfit models take into account the data points that fall out of a clear pattern in the data. In this way, overfitting models learn the noise in a distribution, rather than the overall trend. In the diabetes dataset, the model did not overfit the training dataset. For each of the three models, the training accuracy and testing accuracy were similar, indicating that the model generalized well. This outcome is good as it allows the model to perform well on unseen data.

Methods

Support Vector Classifier

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression, and outlier detection. The advantages of support vector machines include their effectiveness in high-dimensional spaces and in cases where the number of dimensions is greater than the number of samples. SVMs use a subset of training points in the decision function called support vectors. These models gain versatility because unique Kernel functions can be specified for the decision function [5]. The kernel functions supported are linear, polynomial, radial basis function (RBF) and sigmoid. The study was performed using both linear and non-linear kernels. The results tabulated are mainly for the radia basis function (RBF). This kernel provides approximation methods that can be tuned to make models faster and more scalable. When training an SVM with the *Radial Basis Function* (RBF) kernel, two

```
[ ]  x = data.drop(['Outcome'] , axis = 1).values
     y = data['Outcome'].values
     x_train , x_test , y_train , y_test = train_test_split(x,y , test_size= 0.2)
```

parameters must be considered: C and gamma. The parameter C, common to all SVM kernels, trades off misclassification of training examples against the simplicity of the decision surface. A low C value makes the decision surface smooth, while a high C aims at classifying all training examples correctly. The parameter gamma defines how much influence a single training example has. In the default setting of the library used, C is set to 1 and gamma is set to 0.1. Lower values of gamma result in models with lower accuracy. Intermediate values of gamma give a model with good decision boundaries. The choice to have gamma at intermediate levels was to get reasonable accuracy.

Linear Classifier

A linear classifier is a model that classifies a set of data points into discrete classes based on a linear combination of its explanatory variables. The effectiveness of these models lies in their ability to find this mathematical combination of features that groups data points together when they have the same class and separate them when they have different classes, providing us with clear boundaries for classification. Linear classification is initially an extension of linear regression models. The goal is to find a set of coefficients for our features that when summed together, will provide us with an accurate measure of our target variable. For our study, we chose a linear classifier with stochastic gradient descent (SGD) training. SGD is computationally more efficient compared to traditional batch gradient descent methods. It updates the model parameters based on a random subset of the training data at each iteration, which makes it well-suited for large datasets. There are over 10 parameters in the SGD classifier that can be used to fine-tune the behavior of the classifier. The most important ones that we used were:

1) Loss, the loss function to be optimized during training. Options include 'hinge' for linear SVM, 'log' for logistic regression, and 'modified_huber'. Since we already have done the SVM separately, we chose 'modified_huber' for smoothed loss. 2) Penalty, the regularization term to be applied during training to prevent overfitting. Common options are 'l2' for L2 regularization and 'l1' for L1 regularization. We set this to L1 regularization.

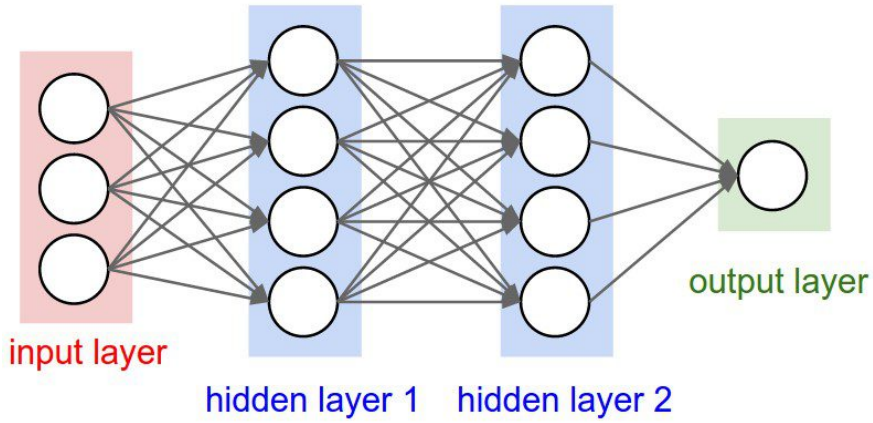
Decision Tree

Decision Trees are a non-parametric supervised learning method used for classification and regression [6,7]. The goal of this classifier is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data's features. A tree can be seen as a piecewise constant approximation and have advantages as they are simple to understand and interpret. Trees can also be visualized and require little data preparation. Other techniques often require data normalization, dummy variables, and blank values to be removed. However, it is important to note that this module does not support missing values. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the decision tree [8]. Decision trees are able to handle both numerical and categorical data while other techniques are usually specialized in analyzing datasets that have one or the other. In a decision tree, given an observation, the explanation for the condition can be easily explained by Boolean logic, even though it has many outputs. By contrast, in a black box model (e.g. in an artificial neural network), results may be more difficult to interpret (it is possible to validate a model using statistical tests). In our experiment, we set the criterion to be 'entropy' and max depth to be 3. The model would create nodes only at three levels instead of creating leaf nodes until the end.

Artificial Neural Network

(image source: https://cs231n.github.io/neural-networks-1/)

An Artificial Neural Network (ANN) is a computational model inspired by the human brain's neural structure. It consists of interconnected nodes (neurons) organized into layers. Information flows through



these nodes, and the network adjusts the connection strengths (weights) during training to learn from data, enabling it to recognize patterns, make predictions, and solve various tasks in machine learning and artificial intelligence. There are three layers in the network architecture: the input layer, the hidden layer (often more than one), and the output layer. It is possible to think of the hidden layer as a "distillation layer," which extracts some of the most relevant patterns from the inputs and sends them on to the next layer for further analysis. It accelerates and improves the efficacy of the network by recognizing the most important information from the inputs and discarding the redundant information. The activation function is important because it introduces non-linearity for the model to learn more complex relationships and decision boundaries. It also contributes to the conversion of the input into a more usable output.

The most important hyperparameters are hidden_layer_sizes, which defines the architecture of the neural network, and activation, which specifies the activation function to be used in the hidden layers. In our experiment, we created a neural network with three hidden layers having eight neurons each, respectively, and used the ReLU activation function. One can adjust the hidden_layer_sizes and other hyperparameters based on the dataset and task requirements.

Results

	SVC	Linear Classifier	Decision Tree	ANN
Accuracy (test)	0.753	0.68	0.766	0.766
Precision	0.655	0.56	0.686	0.686
Recall	0.655	0.58	0.636	0.636
F1	0.655	0.57	0.660	0.660
Training Acc.	77.524	69.5	77.361	77.361
Testing Acc.	76.623	68.8	75.324	75.324

Table 1: Results of Classifiers

The decision tree and neural network models returned the highest accuracy of 0.766. Additionally, it is important to note training accuracy and test accuracy of each classifier. Table 1 shows the accuracy, precision, recall, and F1 score of each classifier and depicts how each classifier had similar training and testing accuracies, which indicates that the model is not overfitting. The performance on the training set is similar to how well the model's tasks can be recreated. If a training accuracy is significantly larger than a testing accuracy then it is likely that the model overfit, and vice versa with underfitting if the training accuracy is less than the testing accuracy.

Correlation

Figure 6 shows the results of two predictors. The plots include one predictor plotted in orange and the latter plotted in blue. The relationship between the two parameters on each graph varies significantly. For example, some orange and blue plots are scattered, while other plots have distinct blue and orange separation. The simple heatmap in Figure 7 expands on Figure 6 in detail. It gives us the importance of each feature towards predicting diabetes. It is observed that these four features are the most important indicators for diabetes: (1) Glucose (2) BMI (3) Age (4) Pregnancy. Interestingly, insulin, a common predictor for diabetes, did not yield a high prediction accuracy. This lower pairwise correlation could be due to the fact that insulin is a nonlinear diabetes predictor. Glucose, however, is a linear predictor as the higher glucose a patient has, the more likely it is that they have diabetes. Patients' insulin level can vary significantly between those of the same diagnosis. When one's insulin is high, the body attempts to bring the insulin level down. The dataset also has many null insulin values and does not take into account whether patients are on medication or have other health problems that affect the insulin level.
Two ways that the importance of a feature can be determined are pairwise correlation to the class outcome and the weight of the coefficients of the linear classifier. For both models, it can be seen that glucose and BMI are the most reliable predictors of diabetes.

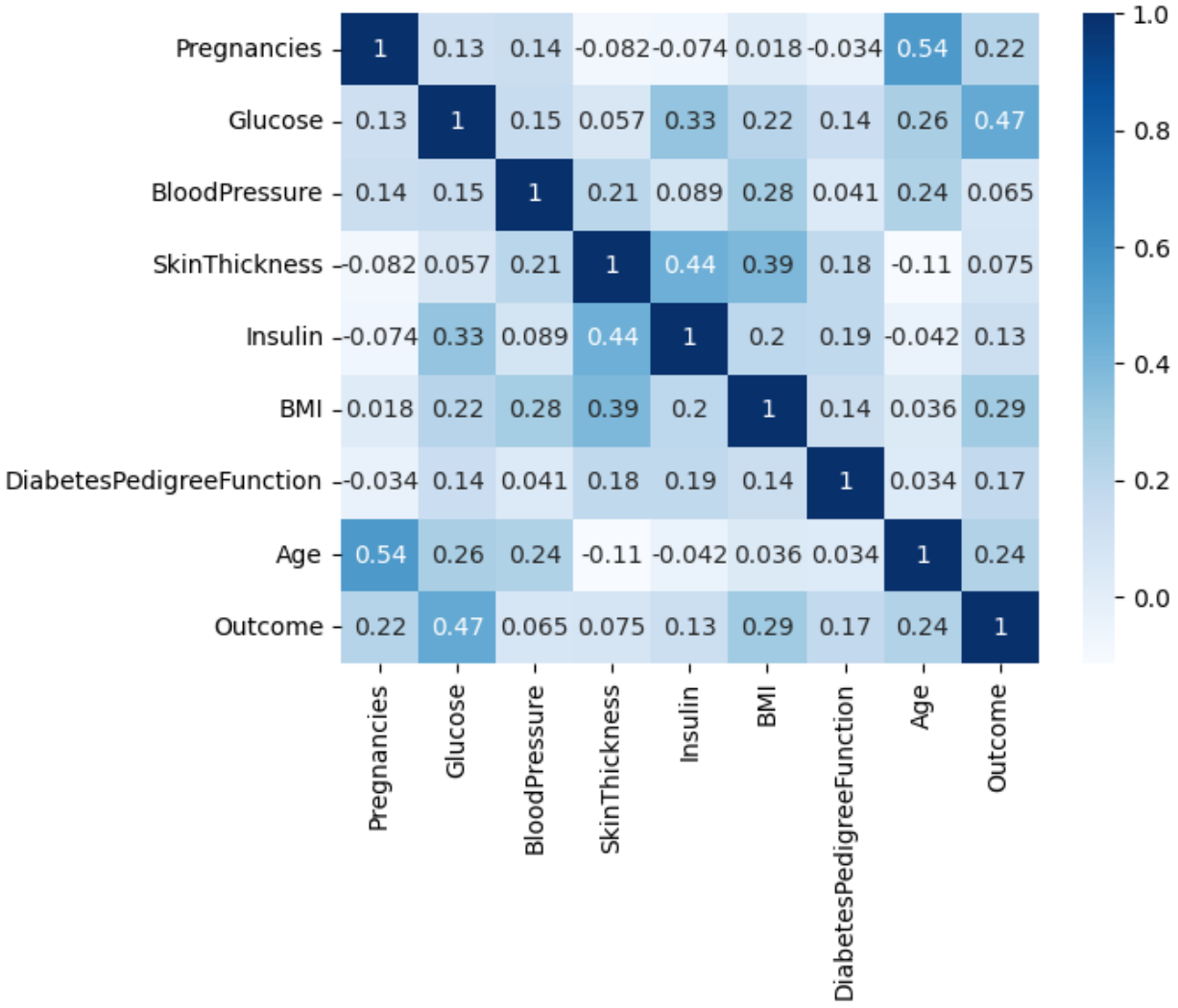


Figure 7: Cross-correlation matrix: This matrix contains the pairwise correlation coefficients between each feature in our dataset. This allows us to observe potential relationships between these features, including the outcome classification for each patient.

Conclusion

The results of this research found that the decision tree and SVM classifier were the most accurate in their capability to correctly predict diabetes. The predictions of these algorithms can be used in clinical settings such as predicting diabetes at scale for specific populations. For example, populations of similar

characteristics can be better treated as prediction models can accurately depict the percentage of patients that have diabetes. For example, health institutions can create diabetes-focused patient registries and sort the patients according to certain criteria (e.g. anyone over the age of 40, high glucose levels, etc.). After finding the patients at most risk for diabetes, doctors can follow up with those patients for diabetes screening according to the classification model [9]. This is just one way that classification models can be used to help improve patient care. The more accurate the data is, the more likely a better model can be made for patient diagnosis. A data-centric approach to modeling will yield better accuracy regardless of model used.

References

1. Fregoso-Aparicio, L., Noguez, J., Montesinos, L. *et al.* Machine learning and deep learning predictive models for type 2 diabetes: a systematic review. *Diabetol Metab Syndr* 13, 148 (2021). https://doi.org/10.1186/s13098-021-00767-9

2. Singla R, Singla A, Gupta Y, Kalra S. Artificial Intelligence/Machine Learning in Diabetes Care. Indian J Endocrinol Metab. 2019 Jul-Aug;23(4):495-497. doi: 10.4103/ijem.IJEM_228_19. PMID: 31741913; PMCID: PMC6844177

3. Teboul, Alex. "Diabetes Health Indicators Dataset." *Kaggle*, 8 Nov. 2021, www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset.

4. Bureau, US Census. "Census Bureau Releases New U.S. Population Estimates by Age and Sex." *Census.Gov*, 14 Apr. 2022, www.census.gov/newsroom/press-releases/2022/population-estimates-age-sex.html.

5. "1.4. Support Vector Machines." *Scikit*, scikit-learn.org/stable/modules/svm.html#svm-kernels. Accessed 21 July 2023.

6. "1.10. Decision Trees." *Scikit*, scikit-learn.org/stable/modules/tree.html#tree-classification. Accessed 29 July 2023.

7. "1.10. Decision Trees." *Scikit*, scikit-learn.org/stable/modules/tree.html#tree-regression. Accessed 29 July 2023.