Implementation of Explainable Machine Learning in Diabetes Diagnosis

Tracy Liu
Biomedical Engineering
The University of Texas at Austin
Austin, USA
ytliu.tracy@utexas.edu

HyoJeong Kim
School of Information
The University of Texas at Austin
Austin, USA
hkim92@utexas.edu

I. INTRODUCTION

Once you receive or access your health report from the laboratory, it may not be easy for you to read or understand, leaving you with more questions than answers. To understand what your report means, you may tend to professionals or online resources to get the information. However, limited doctor consultation time and barriers to accessing professional health services led to an increasing number of people in the US using online searching for health information [1]. Searching through the internet can spend extensive time and get broad and mixed information especially if you have no idea what the right term is to search. Even finally knowing what those medical values in the report mean in terms of the health condition, people may still be concerned about how much each medical feature contributed to the outcome? Without knowing the fraction of each medical feature's contribution, people may not be fully convinced with the result. Nowadays, healthcare applications mainly focus on keeping track of physiological values and knowing their health status or disease diagnosis [2]-[5]. The trust issue of diagnosis results remains unsolved.

There is numerous information in the health report, considering the scope of the course project, we will focus on diabetes specifically. We propose to build a model to help predict whether a patient/person has diabetes or not based on the related medical features in the health report including the number of pregnancies the patient has had, BMI, insulin level, age, skin thickness, blood pressure, glucose concentration, and diabetes pedigree function. After the prediction, we will implement SHAP to provide an explanation of how much each feature contributes to the predicted outcome. This course project aims to predict whether a patient/person has diabetes or not based on the related medical features and provide the contribution composition of these features to the outcome. We are trying to tackle the problems of health report interpretation and trust issues with the outcome interpretation when there is no or not enough professional health support.

II. DATA SET

We will use the Pima Indians Diabetes Database which is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. The dataset consists of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, skin thickness, blood pressure, glucose concentration, and diabetes pedigree function. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset.

III. METHOD

There are five major steps in this project, including data preprocessing, feature engineering, modeling, SHAP implementing, and application development.

A. Data preprocessing

First, we will do descriptive analysis to know the data distribution and data types, then identify associations among variables. Data may have many irrelevant and missing parts. To handle this part, data cleaning, and data imputation will be done. For data imputation, we will use both mean and KNN to see which performs better.

B. Feature engineering

This is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. We intend to compare the performance among original features, PCA, and LDA.

C. Modeling

We will compare the prediction performance by using typical machine learning models such as SVM, Logistic regression, Decision tree, Random Forest, and Adaboost.

D. SHAP Implementing

SHAP (SHapley Additive exPlanations) is a game-theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions. After prediction, we will implement SHAP on our prediction model to get the explanation result.

E. Evaluation

We will use performance metrics like Confusion Matrix, Precision-Recall, and F1-Score along with Accuracy while evaluating our model. We are using these evaluation metrics because we are predicting whether a participant has diabetes or not based on the health record. For example, in confusion metrics, there are two types of error. Type I error is the test result says you have diabetes, but you actually don't. While Type II error is the test result says you don't have diabetes, but you actually do. For our model, type II error is very risky. If a diabetes patient is mis predicted, the condition will become worse without treatment.

IV. RESULT

In this section, the findings for each step in this project are presented. There are eight features, one outcome variable and 768 data points in total. Throughout the training/testing process, we randomly select 33% of data points as testing data and the remaining as training data.

1. *Descriptive Analysis*

In this study, the outcome is the dependent variable, and the remaining 8 variables are independent variables. The null values do not exist in the dataset. But there are lots of zero values in the dataset (Table 1).

TABLE 1. COUNT AND PERCENTAGE OF ZEROS IN EACH FEATURE

Feature	Count	%
Pregnancies	111	14.5
Glucose	5	0.7
Blood Pressure	35	4.6
Skin Thickness	227	29.6
Insulin	374	48.7
BMI	11	1.4
Diabetes Pedigree Function	0	0
Age	0	0

An association between variables to study bivariate relationship is analyzed. Participants with higher glucose levels tend to have a higher chance to be diagnosed with Diabetes. There are also higher chances of Diabetes as BMI increases. Age alone is not a clear indicator of Diabetes. The result is shown in Figure 1.

2. *Data Imputation*

From the previous section, we can observe that there is no data point missing in the dataset. However, there are some of the measurements (Glucose, Blood Pressure, Skin Thickness, Insulin and BMI) have zero values, which is either rare or not possible for a living human. Instead of zero values, we think this should be more like losing valuable information. Below is the detailed explanation for features that cannot be zero:

- Glucose: Even after fasting glucose levels would not be as low as 0. Therefore, 0 is an invalid reading.
- Blood Pressure: A living person cannot have a diastolic blood pressure of 0.
- Skin thickness: For normal people, skin fold thickness can’t be less than 10 mm better yet 0.
- Insulin: In a rare situation a person can have 0 insulin but by observing the data, we can find that there is a total of 374 counts.
- BMI: Should not be 0 or close to 0 unless the person is really underweighting which could be life-threatening.

In this project, we will try the imputation technique such as mean imputation and KNN to see the performance.

2.1 Impute missing values with mean

We impute features (glucose, blood pressure, skin thickness, insulin and BMI) that cannot be zero with mean along each feature. Using random forest as classifier, we got cross validation accuracy 75.6%.

2.2 KNN Imputation

Each sample’s missing values are imputed using the mean value from k nearest neighbors found in the training set. A critical point here is that the KNN Imptuer is a distance-based imputation method, and it requires us to normalize our data. Otherwise, the different scales of our data will lead the KNN Imputer to generate biased replacements for the missing values. For simplicity, we will use Scikit-Learn’s MinMaxScaler which will scale our variables to have values between 0 and 1.

Figure 2 depicts the cross-validation accuracy with increasing k. It shows that there is not much improvement when using an increasing number of k to impute the data. There is no obvious change of trend in accuracy, mean and standard deviation. We got the best performance of about 77.6% accuracy with k=9.

TABLE 2. COMPARISON USING IMPOTED DATA OR NOT

Data	Cross validation score
Original data	77.9%
Imputed data with mean	75.6%
Imputed data with KNN	77.6%

Table 2 shows that the cross validation score using original data is higher than doing data imputation. It seems imputation do not help improve the prediction accuracy although we assume it should. In the following steps, we will use original data for feature engineering and modeling.

3. *Feature Engineering*

In this section, we use logistic regression as classifiers to evaluate the feature engineering options. All the figures and tables are using original data as input unless clearly specify as using imputed data.

3.1. Select K best

Select K best works by retaining the first k features of X with the highest scores. Figure 3 shows the cross-validation train accuracy and test accuracy with increasing number of features are retained using original dataset as input. From Figure 3, we got highest combination of cross-validation train accuracy = 77.8% and test accuracy = 74% with k = 6. While using imputed data as input, we got highest combination of cross-validation train accuracy = 77.8% and test accuracy = 74.4% with k = 7.

3.2. Principal Component Analysis (PCA)

PCA is used for the dimensionality reduction. It converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. Table 3 shows cross-validated classification accuracy with increasing number of principle components using original data and imputed data, respectively. Figure 4 shows the cross-validation train accuracy and test accuracy with increasing number of principal components. Figure 5 shows explained variance ratio with increasing number of principal components.

Table 3. Accuracy with increasing number of principal components

# of pc	CV score	CV score (Imputed data)
1	74.5%	76.1%
2	73.3%	76.3%
3	73.7%	76.3%
4	73.2%	76.5%
5	75.3%	75.9%
6	77.8%	76.8%
7	77.8%	76.6%
8	77.2%	77.6%

4. *Modeling*

Here we present the prediction performance using various machine learning models mentioned in method section. The performances of each model will be evaluated using Confusion Matrix, Precision-Recall, F1-score. All the results present in this section are being optimized.

4.1 Logistic Regression

Logistic regression is a classification algorithm, used when the value of the target variable is categorical in nature. Table 4 to Table 7 present the classification report using features like: all features, select k best, PCA, LDA. According to these tables, the results show similar performance with or without feature engineering.

Table 4. Classification report (All features)

	Precision	Recall	F1-score	Support
No diabetes	0.79	0.83	0.81	168
Diabetes	0.62	0.56	0.59	86
Accuracy			0.74	254

Table 5. Classification report (select k best, k=6)

	Precision	Recall	F1-score	Support
No diabetes	0.79	0.83	0.81	168
Diabetes	0.63	0.57	0.60	86
Accuracy			0.74	254

Table 6. Classification report (PCA, n_components=8)*

	Precision	Recall	F1-score	Support
No diabetes	0.80	0.81	0.80	168
Diabetes	0.61	0.59	0.60	86
Accuracy			0.74	254

Table 7. Classification report (LDA)

	Precision	Recall	F1-score	Support
No diabetes	0.80	0.80	0.80	168
Diabetes	0.61	0.60	0.61	86
Accuracy			0.74	254

Figure 6 and Figure 7 present the Confusion matrix and roc curve for PCA with logistic regression, respectively.

4.2 Support Vector Machine

SVM is a set of supervised learning methods for classification, regression, and outlier detection, effective in high-dimensional spaces. Table 8 to Table 11 present the classification report using features like: all features, select k best, PCA, LDA. According to these tables, the results show the best performance using all features with SVM.

Table 8. Classification report (All features)*

	Precision	Recall	F1-score	Support
No diabetes	0.81	0.83	0.82	168
Diabetes	0.65	0.62	0.63	86
Accuracy			0.76	254

Table 9. Classification report (select k best, k=6)

	Precision	Recall	F1-score	Support
No diabetes	0.81	0.82	0.81	168
Diabetes	0.64	0.62	0.63	86
Accuracy			0.75	254

Table 10. Classification report (PCA, n_components=8)

	Precision	Recall	F1-score	Support
No diabetes	0.81	0.82	0.81	168
Diabetes	0.64	0.62	0.63	86
Accuracy			0.75	254

Table 11. Classification report (LDA)

	Precision	Recall	F1-score	Support
No diabetes	0.79	0.83	0.81	168
Diabetes	0.64	0.57	0.60	86
Accuracy			0.74	254

Figure 8 and Figure 9 show the confusion matrix and roc curve for all features using SVM, respectively. Comparing the result in Figure 9 to Figure 7, we found that the precision is higher in SVM than in logistic regression as well as accuracy.

4.3 Decision Tree, Random Forest, AdaBoost, XGBoost

Table 12 to Table 15 present the best classification report of Decision Tree, Random Forest, AdaBoost, XGBoost with their corresponding features, respectively.

Table 12. Classification report (All features + Decision Tree), ROC: 70.04%

	Precision	Recall	F1-score	Support
No diabetes	0.81	0.74	0.77	168
Diabetes	0.56	0.66	0.61	86
Accuracy			0.71	254

TABLE 13. CLASSIFICATION REPORT (ALL FEATURES + RANDOM FOREST), ROC: 80.99%

	Precision	Recall	F1-score	Support
No diabetes	0.80	0.81	0.81	168
Diabetes	0.62	0.62	0.62	86
Accuracy			0.74	254

TABLE 14. CLASSIFICATION REPORT (ALL FEATURES + ADABOOST), ROC: 81.14%

	Precision	Recall	F1-score	Support
No diabetes	0.82	0.83	0.83	168
Diabetes	0.66	0.64	0.65	86
Accuracy			0.77	254

TABLE 15. CLASSIFICATION REPORT (ALL FEATURES + XGBOOST), ROC: 81.04%

	Precision	Recall	F1-score	Support
No diabetes	0.81	0.79	0.80	168
Diabetes	0.61	0.65	0.63	86
Accuracy			0.74	254

Figures 10 to 13 show the confusion matrix of using all features with Decision Tree, Random Forest, AdaBoost, and XGBoost, respectively. Among these figures, we observed that Figure 10 has the least specificity and highest recall, which indicate that DT is the weakest in correctly predicting if the subject does not have diabetes and the most powerful in correctly predicting if the subject has diabetes. Adaboost shows the most promising accuracy among all the models we implemented so far as well as the highest precision. Adaboost shows the smallest value among all models when comparing the probability that the models predict no, but the subjects actually have diabetes.

4.4 ANN

In this section, we tried to implement ANN to see if the performance can outperform the models we have used so far. First, we tried several different hidden units to test which kind of construction works best for the study (Table 16). From Table 16, we found that two hidden layers with 50 hidden units show the best performance both in train and test accuracy. To optimize the result, we tried different regularization value to see if we can reduce the overfitting in training data. In Table 17, as we increase the regularization value, there is no clear trend that training accuracy increases or decreases, while test accuracy tends to decrease. This indicates that 0.00001 may be the suitable regularization value in (50, 50) hidden layers. From Table 17, we also found that test accuracy reaches the maximum at epoch 86 when regularization value = 0.00001. Looking at Figures 13 and 14, as the epoch increases, the model training loss decreases and model training accuracy increases, while the model validation loss increases, and model validation accuracy drops. It seems like in this model more epochs do not lead to better performance, this model requires an early stop to ensure the performance.

Using all features as input to ANN (regularization = 0.00001, epoch = 86), we get the classification accuracy = 79%, precision = 73% (Table 18, Figure 15). ANN outperforms other models so far in both testing accuracy and precision.

TABLE 16. ANN CLASSIFICATION ACCURACY WITH DIFFERENT HIDDEN UNITS

Hidden layer	Train	Test
(4, 4)	74.51%	73.62%
(50, 50)	99.2% / 80.35%	72.05% / 79.13%
(3, 1, 2)	78.21%	78.35%
(10, 50, 10, 5)	85.21%	74.02%

TABLE 17. ANN CLASSIFICATION ACCURACY WITH DIFFERRENT REGULARIZATION PENALTY

Regularization	Train	Test
0.00001 (epoch=86)	80.35%	79.13%
0.00001 (epoch=1000)	99.2%	72.05%
0.00005	94.36%	70.47%
0.0001	88.33%	72.05%
0.00015	89.88%	68.9%

TABLE 18. CLASSIFICATION REPORT (ALL FEATURES + ANN), REGULARIZATION=0.00001, EPOCH=86

	Precision	Recall	F1-score	Support
No diabetes	0.82	0.88	0.85	168
Diabetes	0.73	0.62	0.67	86
Accuracy			0.79	254

V. SHAP IMPLEMENTATION

The goal of SHAP is to explain the prediction of instance x by computing the contribution of each feature to the prediction. To see how a single feature affects the output of the model, we can plot the SHAP value for that feature against the feature values for all samples in the dataset. Figure 16 shows that the greater the SHAP in glucose, the greater the predictive value of the outcome, implying a predisposition to diabetes. Conversely, lower SHAP values in glucose predicted a lower value for the outcome, implying a tendency to not have diabetes.

To get an overview of which features are most important for a model we can plot the SHAP values of every feature for every sample. Figure 17 sorts features by the sum of SHAP value magnitudes over all samples and uses SHAP values to show the distribution of the impacts each feature has on the model output. Each point on the summary plot is a Shapley value for a feature and an instance. The color represents the feature value (red high, blue low). The features are ordered according to their importance. In Figure 17, we observe that the ranking importance orders of features are glucose, age, BMI, diabetes pedigree function, insulin, skin thickness, blood pressure, and pregnancies. Glucose, age, and BMI play important roles in diabetes prediction. We can also just take the mean absolute value of the SHAP values for each feature to get a standard bar plot (Figure 18).

Figures 19 and 20 show each feature that helps push the model output from the base value (the average model output on the training dataset we passed) to the model output. Each Shapley value is an arrow that pushes to increase (positive value, shown in red) or decrease (negative value, shown in blue) the prediction. Age, BMI, glucose, and insulin are powerful features pushing to increase the prediction of diabetes in patients 586 (Figure 19). On the contrary, diabetes pedigree function is a powerful factor in the tendency to pull back. In Figure 20, glucose is a powerful feature that pushes to decrease the prediction, which means not having diabetes. Glucose is so powerful that even the total strength of the skin thickness and age is not enough to compare.

VI. CONCLUSION

In this study, we successfully predict whether a patient/person has diabetes or not based on the related medical features and provide the contribution composition of these features to the outcome. According to the SHAP results, the features that matter most for predicting diabetes and no diabetes are not quite the same. Diabetes pedigree function, BMI, and age are the powerful factors that contribute to the diabetes result. Glucose is a strong factor when predicting no diabetes results.

During the data imputation step, we would have thought that imputation should significantly improve the prediction performance, but this is not the case. In this dataset, the samples without diabetes are twice as many as those with diabetes. Data imbalance may be more influential than missing values for predicted outcomes. In future steps, perform down sampling on the samples with no diabetes to avoid the data imbalance while training may show promising prediction improvement.

REFERENCES

[1] Amante, Daniel J et al. “Access to care and use of the Internet to search for health information: results from the US National Health Interview Survey.” Journal of medical Internet research vol. 17,4 e106. 29 Apr. 2015, doi:10.2196/jmir.4126

[2] Naz, Huma, and Sachin Ahuja. “Deep learning approach for diabetes prediction using PIMA Indian dataset.” Journal of diabetes and metabolic disorders vol. 19,1 391-403. 14 Apr. 2020, doi:10.1007/s40200-020-00520-5

[3] Li, Wei, et al. "A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare systems." *Mobile Networks and Applications* 26.1 (2021): 234-252

[4] Tariq, Hassan, et al. "Performance Analysis of Deep-Neural-Network-Based Automatic Diagnosis of Diabetic Retinopathy." Sensors 22.1 (2021): 205.

[5] Schick, Timo, Sahana Udupa, and Hinrich Schütze. "Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp." *Transactions of the Association for Computational Linguistics* 9 (2021): 1408-1424.

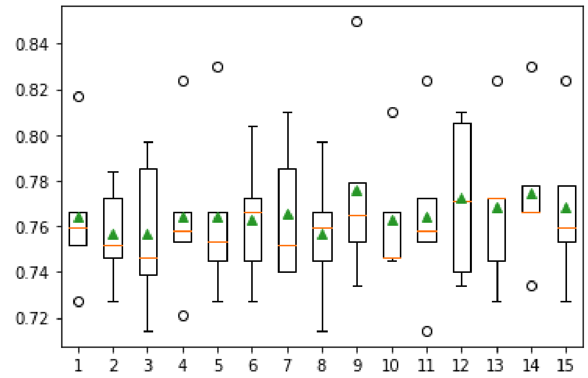


Figure 2. Cross validation accuracy with increasing k

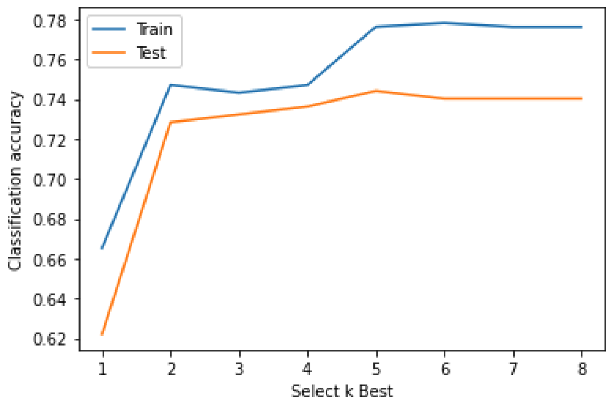


Figure 3. Classification accuracy as number of selected k features increases

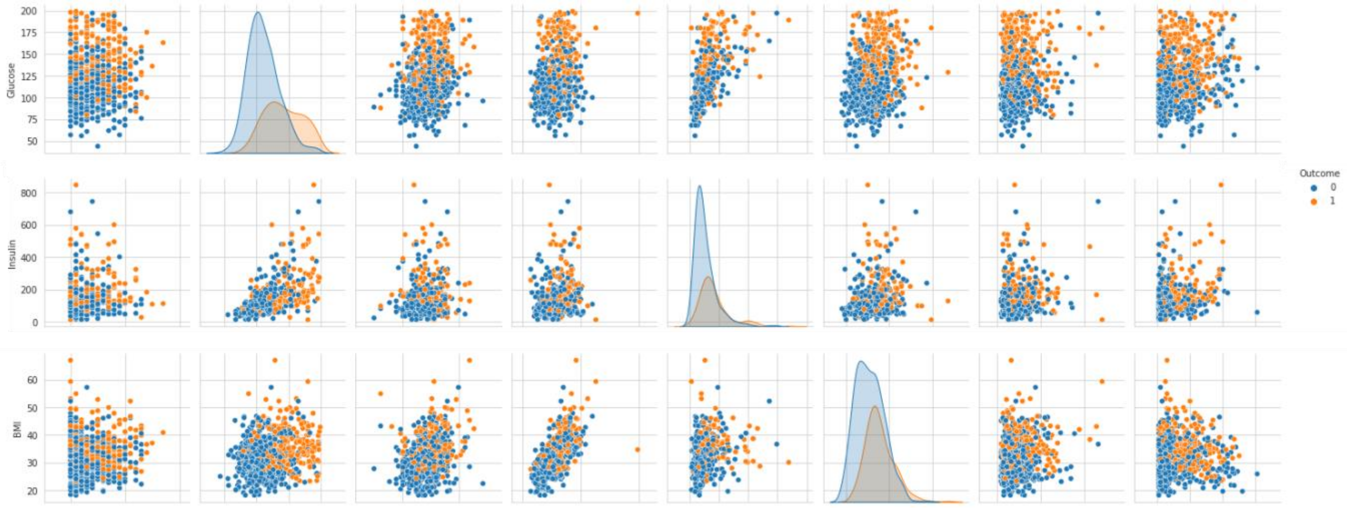


Figure 1. An association between variables to study bivariate relationship

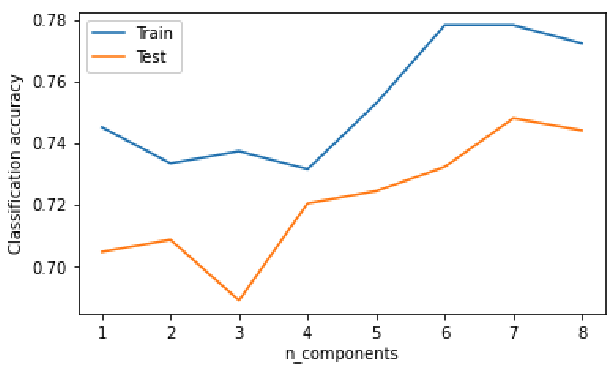


Figure 4. Classification accuracy with increasing number of principal components

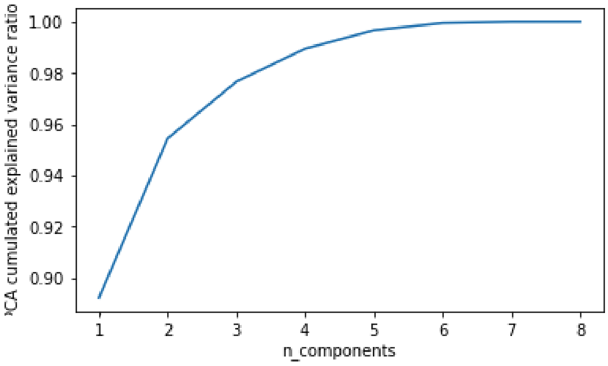


Figure 5. Explained variance ratio with increasing number of principal components

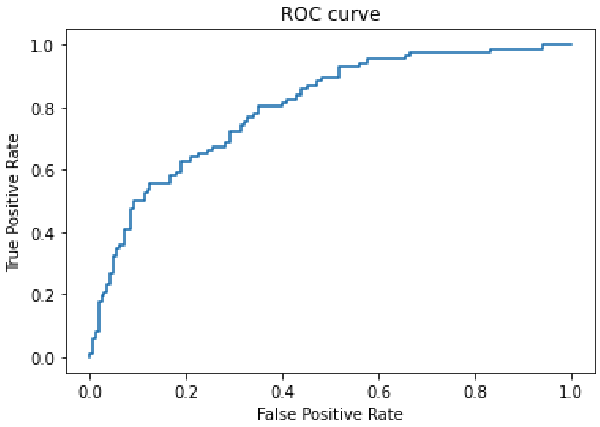
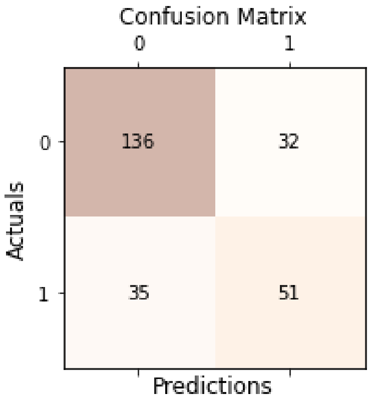


Figure 6. ROC curve. AUC = 80%. Combination of PCA and Logistic Regression.



	0	1
0	136	32
1	35	51

Figure 7. Confusion matrix. Combination of PCA and Logistic Regression

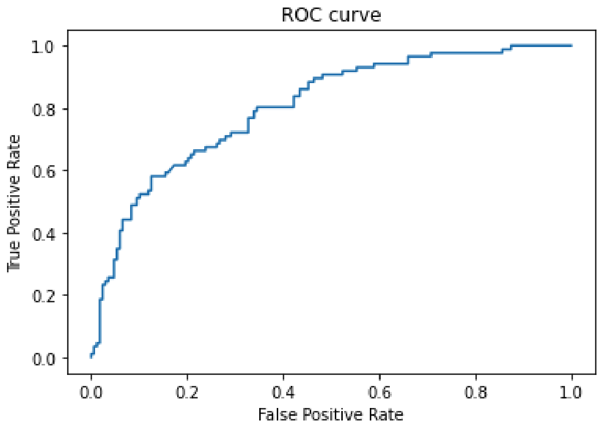
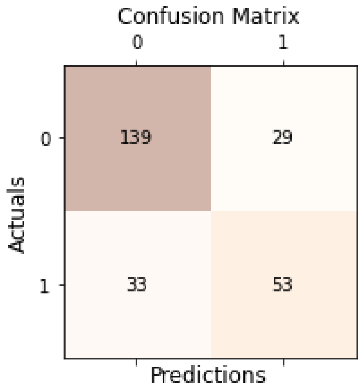
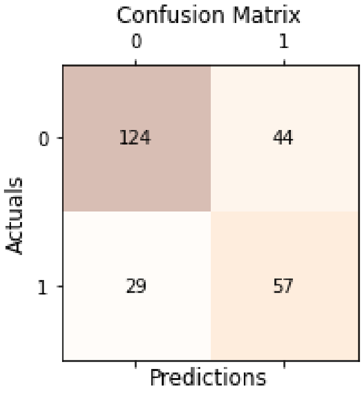


Figure 8. ROC curve. AUC = 80.3%. Combination of all features and SVM.



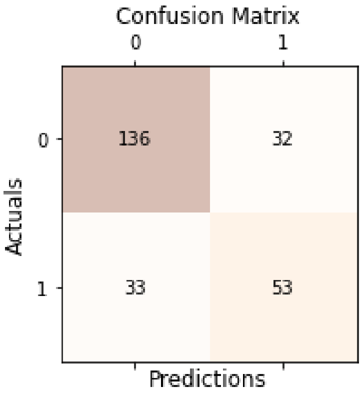
	0	1
0	139	29
1	33	53

Figure 9. Confusion matrix. Combination of all features and SVM



	0	1
0	124	44
1	29	57

Figure 10. Confusion matrix. Combination of all features and DT



	0	1
0	136	32
1	33	53

Figure 11. Confusion matrix. Combination of all features and RF

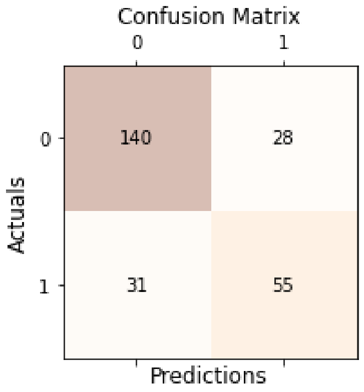


Figure 12. Confusion matrix. Combination of select all features and Adaboost

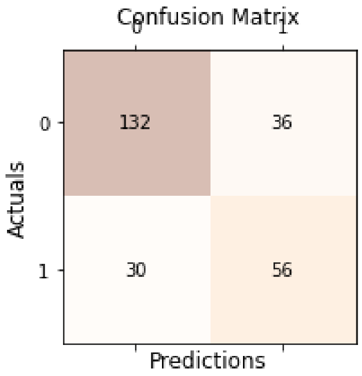


Figure 13. Confusion matrix. Combination of all features and XGBoost

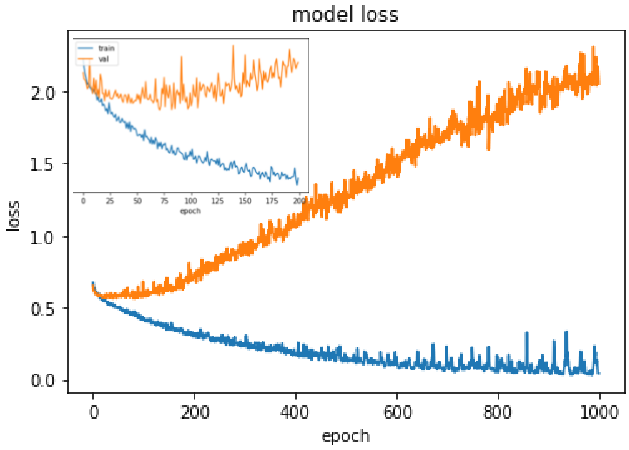


Figure 13. Model loss with hidden layers (50, 50). Regularization term=0.00001

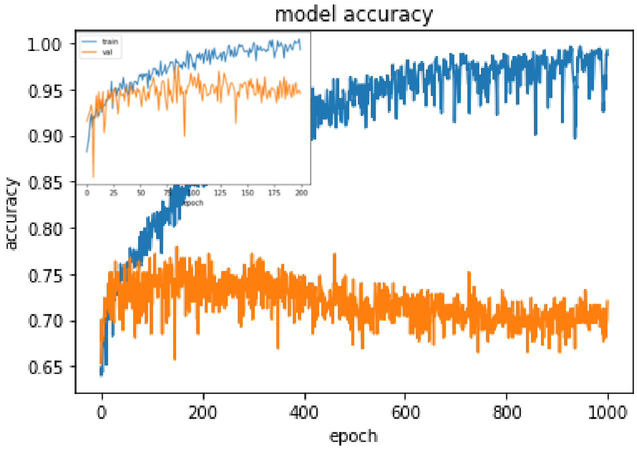


Figure 14. Model accuracy with hidden layers (50, 50). Regularization term=0.00001

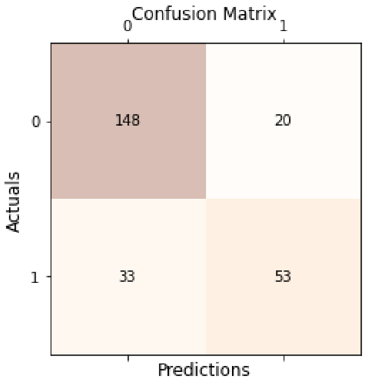


Figure 15. Confusion matrix. Combination of all features and ANN with hidden layers (50, 50)

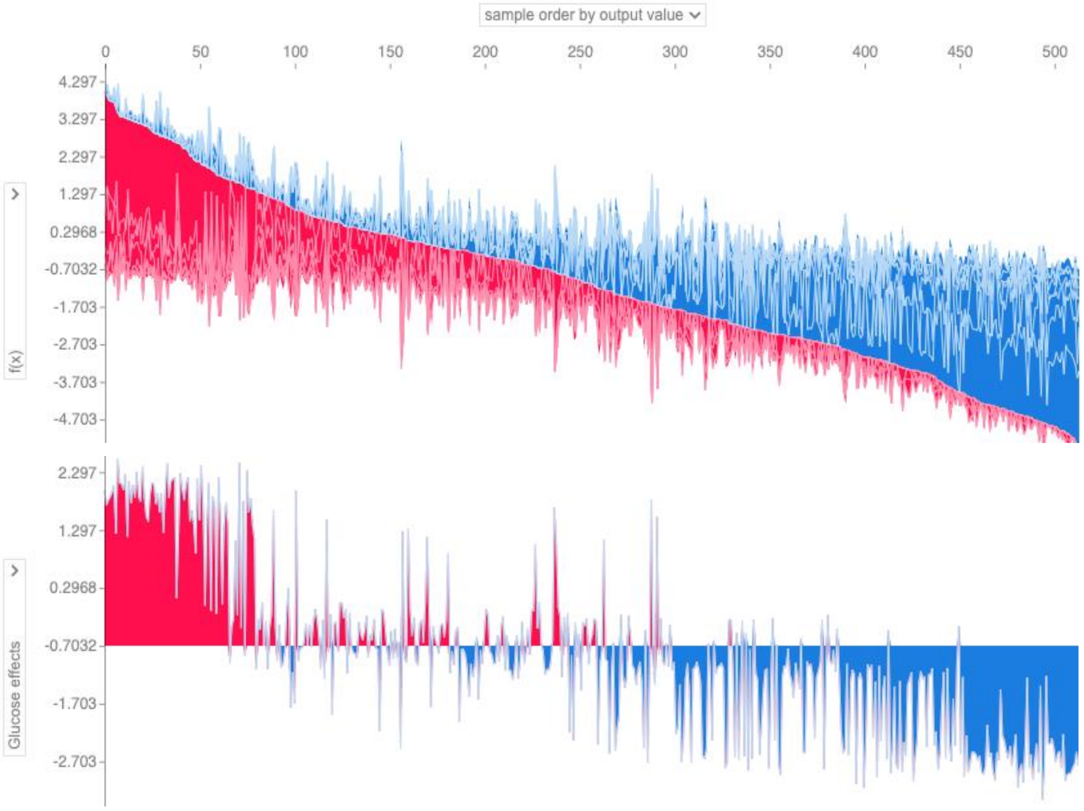


Figure 16. SHAP value of the features for all the examples in the training dataset. X-axis: training sample order by the outcome value. Each position on the X-axis is an instance of the data. Y-axis at the upper panel: Prediction in diabetes status outcomes interact with other features. Y-axis at the lower panel: Glucose effects. Red SHAP values increase the prediction, blue values decrease it.

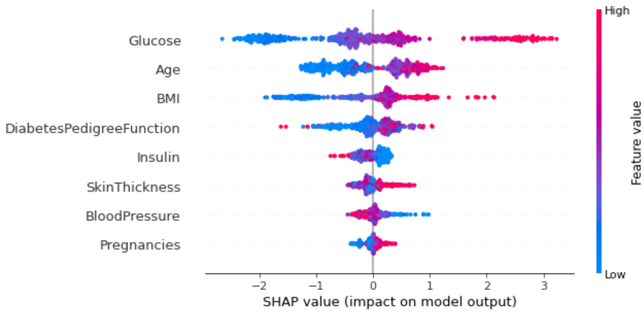


Figure 17. SHAP impact of each feature on model output over all training samples.

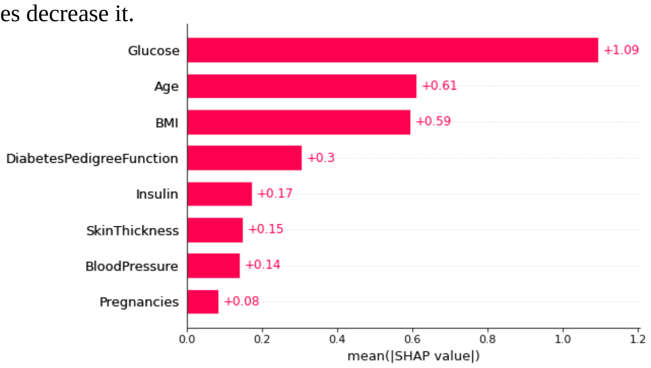


Figure 18. Mean absolute value of SHAP of each feature on model output

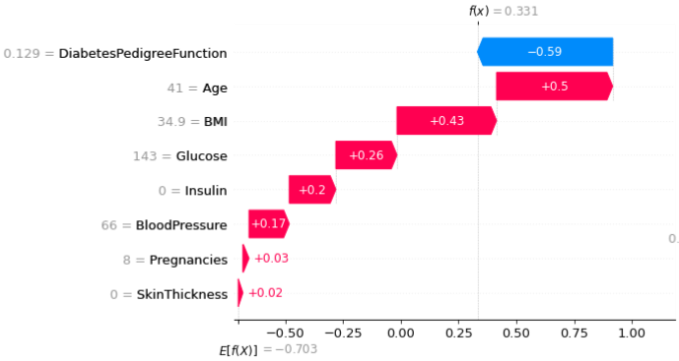


Figure 19. Feature contribution to the model prediction output on testing data. Patient 586 with diabetes.

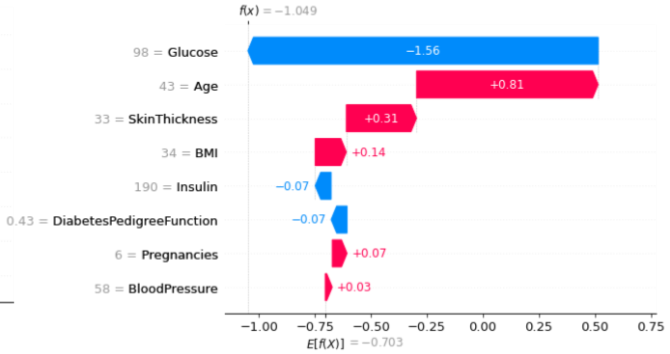


Figure 20. Feature contribution to the model prediction output on training data. Patient 668 with no diabetes.