AI Revolution: Predicting diabetes for better health.

Logistic Regression & Random Forest model

Author:

Hassan Siddique (22007205)

Kevin Siagian (22043973)

Shafeeq Shuaib (22036203)

23 November 2024

Abstract

The purpose of this report is to explore how AI can help us predict diabetes more accurately. By using two different models, we studied how they performed under various conditions. Our focus was on making sure the models work well by balancing the dataset and carefully choosing features. We looked at different ways of splitting the data and how it affected the models' accuracy. The findings emphasis the importance of properly assessing these models and how they can be adapted for more effective healthcare. This study not only improves our understanding of diabetes prediction but also highlights the ever-changing landscape of healthcare technology, always aiming for better outcomes for patients.

Contents

Introduction

The way we handle medical issues is evolving due to artificial intelligence (AI), particularly complex ones like diabetes. Millions of people worldwide suffer from diabetes, a chronic health issue that presents difficulties for both patients and healthcare systems. The primary issue is the critical need for early diabetes detection and identification of high-risk individuals, both of which are challenging tasks. The complexity of diabetes, the volume of data involved, and the difficulty in identifying minuscule patterns that may indicate the earliest signs of the disease are the main causes of this problem.

AI uses sophisticated data analysis, pattern recognition, and predictive capabilities to address these problems. This holds great promise for more precise and swift interventions. In this report, we look at how AI is critical for both identifying those who are more likely to develop diabetes and for diagnosing the disease. This provides a clear picture of how AI is driving advancements in healthcare.

Background

A dataset, available on Kaggle and tailored for diabetes classification, originally encompassed 769 cases with 9 attributes. My modification process involved balancing the dataset by dividing it into nearly equal proportions of diabetic and non-diabetic cases, resulting in 637 instances to enhance fairness in analysis. The dataset's source is the National Institute of Diabetes and Digestive and Kidney Diseases, and it specifically involves females aged at least 21 of Pima Indian heritage.

This data collection is like a toolkit for predicting diabetes. It includes various measurements, and something called 'Outcome' that tells us the result. So, it's not just a random collection of numbers – it is a carefully chosen to set, that helps with making predictions.

Explaining parameters

- Pregnancies: The number of times someone has been pregnant can affect their chances of getting diabetes. If they had diabetes during pregnancy, it increases the risk of having diabetes later in life.
- Glucose: This refers to the sugar levels in your blood. High sugar levels, especially when fasting or after a glucose test, can indicate diabetes or a pre-diabetic condition.
- Blood Pressure: High blood pressure often goes hand in hand with diabetes. Keeping an eye on blood pressure is important for overall heart health, especially for those with diabetes.
- Skin Thickness: While not directly related to diabetes, it can be important in assessing the risk of complications associated with diabetes, such as skin problems.

- Insulin: Insulin is a hormone that regulates blood sugar. Problems with insulin levels or resistance to insulin can be signs of diabetes.
- BMI (Body Mass Index): BMI is a measure of body fat based on height and weight. A high BMI is a risk factor for type 2 diabetes.
- Diabetes Pedigree Function: This is a number that shows if diabetes runs in the family. Having relatives with diabetes can increase the risk.
- Age: Diabetes risk goes up as you get older, especially for type 2 diabetes, which is more common in older adults.

Database Table structure

No.	**Attribute**	**Domain**
1	Pregnancies	Weeks (0-52)
2	Glucose	Mg/dl (40-100) normal value
3	Blood pressure	MmHg (72-100) normal value
4	Skin thickness	Inches (0-45) normal value
5	Insulin	Units of insulin (0-543) normal value
6	BMI	(0-45.5) normal value
7	Diabetes pedigree function	Mg/dl (0-1) normal value
8	Age	From 21
9	Outcome	0 or 1

Methodology and Data

In this report, a python solution will be developed for the classification of diabetes based on 9 attributes. The solution will be using a Logistic regression model and a random forest model order. In addition, I will be analysing the models' categorisation rates to assess how well the solution performs. The project's goal is to use suitable techniques to achieve the maximum categorisation rate feasible.

Any rows in the dataset with missing values are removed. This is important to ensure that the model is trained on complete data. The X variable contains 8 independent features, that will be used to predict the outcome variable which is contained in the Y variable.

In our case, we used two different models, each with its own way of dealing with data to make predictions better. First, there is the Logistic Regression model, which makes use of normalized features. This indicates that the data is transformed to lie entirely inside a specific range, typically between -1 and 1. This improves the efficiency of the model by ensuring that each of the various variables (features) that we are examining is handled equally and that none has an excessive amount of effect.

The Random Forest model, which is the second model, approaches things a little bit differently. It makes advantage of features known as standardization. This indicates it modifies the data to have a standard distribution between 0 and 1, and a form of average of 0. This is useful when the several objects under examination have disparate scales. By ensuring that every feature, regardless of size, has equal weight, standardization benefits this approach. This can improve the model's performance, particularly when handling data with a wide range of sizes.

To sum it up, the choice between normalization and standardization depends on what kind of data we have and what models we are using. Normalisation suits models like Logistic Regression, where having a consistent scale is helpful. Standardization is good for models like Random Forest, which can handle different scales more effectively.

After looking at the data, we took an extra step to make our models better at predicting outcomes. Three features impacting the "outcome" were carefully chosen based on their scores. These scores serve as indicators of the features' relevance in influencing the target variable.

Initially to train both models we allocated 70% to the train set, 30% for the testing set. Later, the ratios were changed to different values to check how it has impacted the accuracy, which is covered in the next sections.

After assigning these values, the models predicted the probability of diabetes for each person in the dataset. We utilised various metrics, including accuracy, recall, F1 score, and precision, to assess the model's performance on the test dataset.

Lastly, we used a graph to see how well the models were doing. The graph showed us how different ways of scaling the features affected their accuracy. Interestingly, it pointed out that the model worked best when we used the Min-Max Scaling (Normalization) technique for scaling the features. This visual guide gave us important clues about how the model behaves, helping us thoroughly assess how it predicts the likelihood of diabetes in various situations.

Focus on evaluation – metrics.

When we are checking how well our models are doing, we look at a few important measures: accuracy, recall, F1 score, and precision.

- Accuracy: This tells us how often our model is right overall. It's like looking at all the predictions and seeing how many are correct compared to all of them. Formula used to calculate accuracy is: (TP+TN)/(TP+TN+FP+FN).
- Recall: This one is about catching all the important cases. We want to know how many of the actual important things our model is finding. Here is the formula: TP ÷ (TP + FN)
- F1 Score: It is a bit of a mix between precision and recall. It helps us see how good our model is at finding the right things without getting too many wrong. To calculate F1 Score, I have used: 2 * (Precision * Recall) / (Precision + Recall)
- Precision: Precision is about how accurate our positive predictions are. We want to ensure that when our model says something is true, it is true. So, the formula is: TP ÷ (TP + FP)

So, by looking at these measures, we can get a promising idea of how well our models are doing and where they might need some improvement in various aspects of making predictions.

Testing & Analysing

To make sure that the predictive models we created are strong and reliable, we tested them thoroughly. We exposed the models to different situations, like using different ratios to split the training and testing data instead of the usual 70-30 setup. This testing phase was meant to see how well the models could handle different types of data and if their predictive performance changed in any way.

We also did sensitivity analyses to see how the models reacted when we used different methods to scale the features, not just the Min-Max Scaling (Normalization) that we initially preferred. These tests helped us understand how well the models could adapt to changes and gave us insights into their overall performance and stability with various datasets.

By conducting these rigorous tests, we gained more confidence in the models' ability to predict accurately and confirmed that they can be relied upon in real-world situations.

Test Scenario:

For a specific test scenario, the train-test split ratio was adjusted to different values. The Logistic regression model, relying on normalized features, and the random forest model, employing standardized features, were both assessed under this modified configuration.

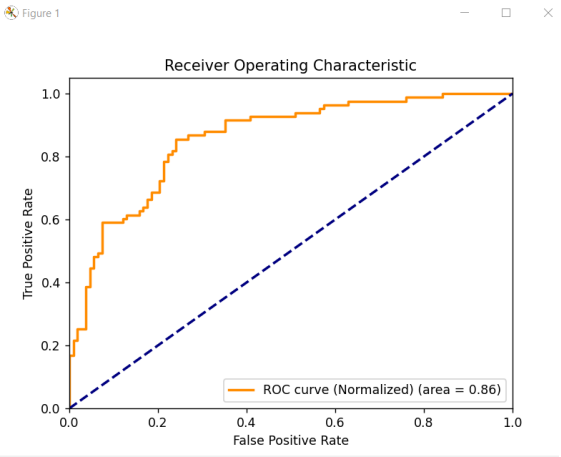
Initial 70-30 Split Results:

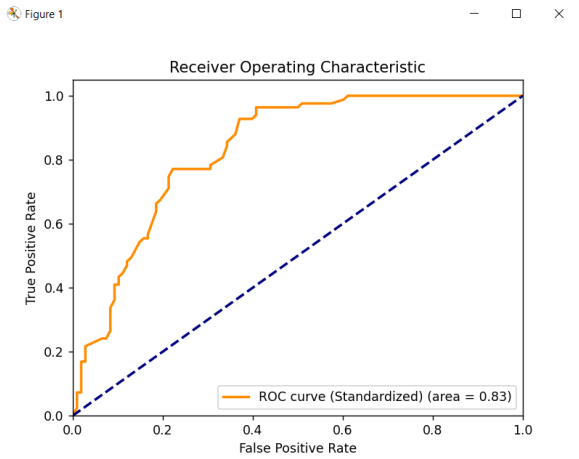
Logistic Regression Model Accuracy: 77%

```
Results for Logistic Regression Model with Normalized Feature
Accuracy: 0.774869109947644
Recall: 0.5903614457831325
Precision: 0.8448275862068966
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.92      0.82       108
           1       0.84      0.59      0.70        83

    accuracy                           0.77       191
   macro avg       0.79      0.75      0.76       191
weighted avg       0.79      0.77      0.77       191
```

Random Forest Model Accuracy: 72%

```
Results for Random Forest Model with Standardized Feature
Accuracy: 0.7277486910994765
Recall: 0.6024096385542169
Precision: 0.7246376811594203
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.82      0.77       108
           1       0.72      0.60      0.66        83

    accuracy                           0.73       191
   macro avg       0.73      0.71      0.72       191
weighted avg       0.73      0.73      0.72       191
```





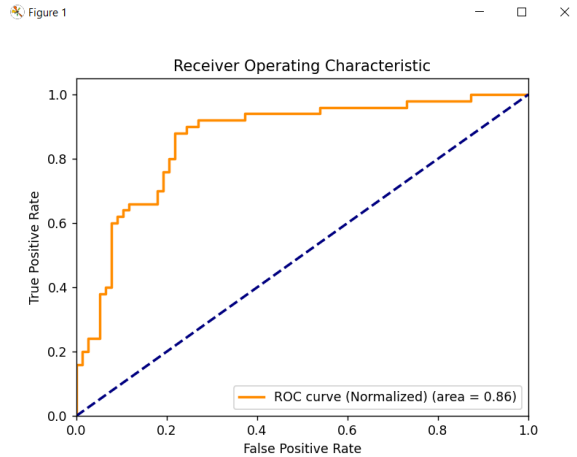
Modified 80-20 Split Results:

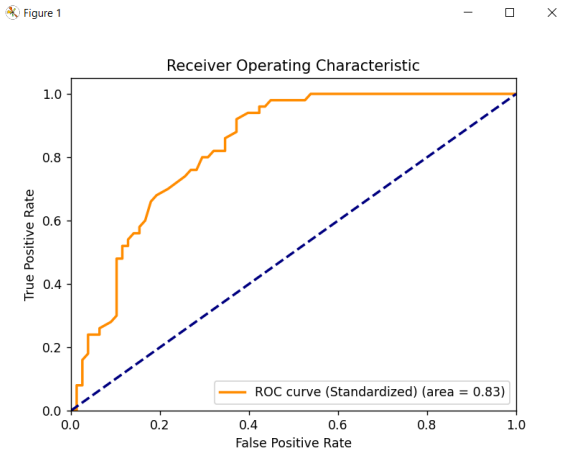
Logistic Regression Model Accuracy: 78%

```
Results for Logistic Regression Model with Normalized Feat
Accuracy: 0.7890625
Recall: 0.62
Precision: 0.7948717948717948
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.90      0.84        78
           1       0.79      0.62      0.70        50

    accuracy                           0.79       128
   macro avg       0.79      0.76      0.77       128
weighted avg       0.79      0.79      0.78       128
```

Random Forest Model Accuracy: 74%

```
Results for Random Forest Model with Standardized Feat
Accuracy: 0.7421875
Recall: 0.6
Precision: 0.6976744186046512
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.83      0.80        78
           1       0.70      0.60      0.65        50

    accuracy                           0.74       128
   macro avg       0.73      0.72      0.72       128
weighted avg       0.74      0.74      0.74       128
```



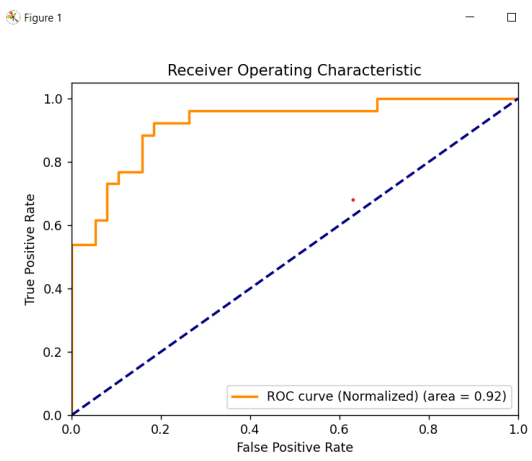


Modified 90-10 Split Results:

Logistic Regression Model Accuracy: 81%

Classification Report:

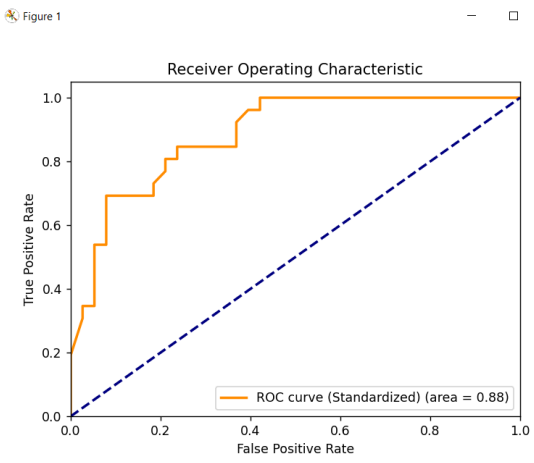
	precision	recall	f1-score	s
0	0.76	1.00	0.86	
1	1.00	0.54	0.70	
accuracy			0.81	
macro avg	0.88	0.77	0.78	
weighted avg	0.86	0.81	0.80	



Random Forest Model Accuracy: 81%

Classification Report:

	precision	recall	f1-score	su
0	0.81	0.89	0.85	
1	0.82	0.69	0.75	
accuracy			0.81	
macro avg	0.81	0.79	0.80	
weighted avg	0.81	0.81	0.81	



Conclusion

In wrapping up our study on using computer models to predict diabetes, we've learned a lot about how these models perform under different situations. The Logistic Regression and Random Forest models showed promise, especially in the initial 70-30 split, with accuracies reaching 77% and 72% respectively. Adjusting the split ratios, we saw adaptability in the models, hitting peak accuracies of 81% for Logistic Regression and Random Forest under a 90-10 split. Our analysis underscored the importance of how we scale features, with Min-Max Scaling (Normalization) turning out to be the best fit for our models.

Future Improvements

Looking ahead, there is room to make our models even better. We could experiment more with different split ratios and explore additional feature scaling methods to enhance their performance. Bringing in more advanced techniques and expanding our dataset to cover a broader range of people could also boost the accuracy of our predictions. Collaboration with healthcare experts and including

more domain knowledge in our models could make them more practical for real-world use, especially in early diabetes detection. As technology and medical knowledge advance, our approach remains flexible, ready to adapt for better healthcare outcomes.

References

- IBM (2022) What is random forest? Available at: https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems. (Accessed: 17/11/23).
- ScienceDirect (2007) Logistic regression Available at: https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on. (Accessed: 14/11/23).
- Kaggle (2022) Diabetes Dataset. Available at: https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset (Accessed: 11/11/23).

Code file instructions.

Project folder has been shared with both tutors.

In case of any difficulties, please use the link below to access the folder and please download the whole folder and read the file called “ReadMe.txt”:

Group 40

Team contribution

I'm in charge of leading the team (Hassan Siddique), and when the assignment came out, I tried reaching out to everyone through Canvas. Unfortunately, it seemed like no one saw my messages, so I informed the professor right away. Eventually, I managed to connect with the team, and we attempted to work on the project together. However, things didn't go smoothly. There were instances where team members didn't show up, and there were often excuses about miscommunication.

We only managed to sit down as a whole team twice during this period to actually get some work done. Two of the team members took on the introduction and background parts, but their writing had a lot of grammar issues and mistakes. I told them that I wouldn't work with them on the next group project if the quality didn't improve. One of the team members, Shafeeq Shuaib, took this to heart and created a completely new report in just two days. He demonstrated interest, but it was too late since I had already finished the code, completed the report, and received feedback from the tutors.

To be fair, I included their contributions in the final report, but unfortunately, there was no willingness or input from the other team member during last week, Kevin Siagian.