Comparative Analysis of Naive Bayes, Logistic Regression, and Multi-Layer Neural Network Model for Predicting Type 2 Diabetes.

Anamika Nayak, Deana Jackson, Truman Daniels, Upasana Chaudhari
Portland State University
(Dated: March 22, 2024)

Acknowledgements

The project Comparative Analysis of Naive Bayes, Logistic Regression, and Multi-Layer Neural Network Network Models for Predicting Type 2 Diabetes wouldn't have been successful without the teamwork and expert advice we received. Anamika Nayak and Deana Jackson worked on the Multi-Layer Neural Network Model, while Truman Daniels and Upasana Chaudhari took care of the Naive Bayes and Logistic Regression Models. We all came together to write the final report. Also, a big thank you goes to Dr. Anthony Rhodes for his guidance and Minwei Lou for his technical help.

Introduction

Early detection of type 2 diabetes is crucial for managing the disease and preventing complications. Machine learning (ML) offers a promising approach to analyze patient data and predict the risk of developing type 2 diabetes. In this project, we aim to evaluate and compare the effectiveness of three machine learning models in predicting the occurrence of type 2 diabetes in patients. The models we will be examining are the

Naive Bayes Classifier, Logistic Regression, and a Multi-Layer Neural Network. Our goal is to understand how each model performs in terms of accuracy, precision, and recall when applied to a dataset of diagnostic measures related to type 2 diabetes. This comparison will help us identify which model is most reliable for this type of prediction task.

We'll be using a dataset from Kaggle containing patient information and their diabetes status. By training and testing the models on this data, we'll assess their accuracy, precision, and recall in correctly identifying patients with the disease. This comparison will help us understand which model might be more suitable for similar medical data and binary classification tasks in the future, potentially aiding in earlier diabetes detection and improved patient care.

Project Goal

The primary goal of this project is to assess and compare the performance of three machine learning models – Naive Bayes Classifier, Logistic Regression, and Multi-Layer Neural Network – in accurately predicting the presence of type 2 diabetes in patients. We aim to analyze these models based on their accuracy and calibration plots using a designated dataset of diagnostic measures for type 2 diabetes. The outcome will provide insight into which model is most effective and reliable for use in medical diagnostics and can serve as a guideline for selecting appropriate machine learning approaches in similar binary classification scenarios in the future.

Data Set Description

This dataset comes from the National Institute of Diabetes and Digestive and Kidney Diseases and is used to predict if a patient has diabetes based on various medical tests and measurements. It includes different medical factors, like the number of times a patient has been pregnant, their body mass index (BMI), weight, age, and blood sugar levels, which help in understanding and predicting diabetes. The main goal is to use these details to build a machine learning model that can accurately identify patients with diabetes. It comprises 390 entries, each representing individual patient data, with the following detailed attributes:

- Patient Number: A unique integer identifier assigned to each patient in the dataset, facilitating easy reference and analysis.
- Cholesterol: Measured in milligrams per deciliter, this value represents the total cholesterol level in the patient's blood, an important indicator of metabolic health and risk factor for diabetes.
- Glucose: The concentration of glucose in the patient's blood, also in milligrams per deciliter, critical for diagnosing diabetes as it indicates how well the body manages blood sugar.
- HDL Chol (High-Density Lipoprotein Cholesterol): Often referred to as "good" cholesterol, these levels are vital in assessing cardiovascular risk and metabolic health.

- Chol/HDL Ratio: The ratio of total cholesterol to HDL cholesterol, providing insight into the patient's risk profile for cardiovascular diseases and metabolic disorders.
- Age: The patient's age in years, an essential demographic factor affecting the risk and prevalence of diabetes.
- Gender: The patient's gender (male or female), as diabetes prevalence and risk factors can vary between genders.
- Height: The height of the patient in inches, used in calculating BMI and assessing overall health status.
- Weight: The weight of the patient in pounds, a crucial measure for evaluating obesity, which is a significant risk factor for diabetes.
- BMI (Body Mass Index): A calculated value from height and weight, BMI provides a standardized measure of obesity, directly correlated with the risk of diabetes.
- Systolic BP: The systolic blood pressure measurement in millimeters of mercury (mmHg), indicating the pressure in blood vessels during heartbeats.
- Diastolic BP: The diastolic blood pressure measurement, representing the pressure in blood vessels between heartbeats.
- Waist: The waist circumference in inches, indicative of central obesity, a key risk factor for type 2 diabetes.
- Hip: The hip circumference in inches, used to assess body fat distribution.

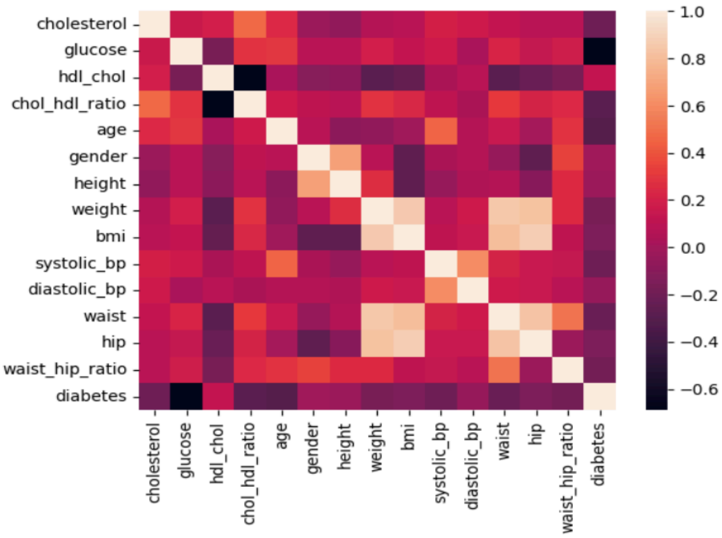
- Waist/Hip Ratio: A significant indicator of body fat distribution and risk of metabolic syndrome, calculated as the waist circumference divided by the hip circumference.
- Diabetes: This is the target variable, indicating whether the patient has been diagnosed with diabetes ("Diabetes" or "No diabetes").

This comprehensive dataset provides a multifaceted view of factors influencing the diagnosis of diabetes, including metabolic measurements, physical characteristics, and demographic information, which will be crucial for training and testing our machine learning models.

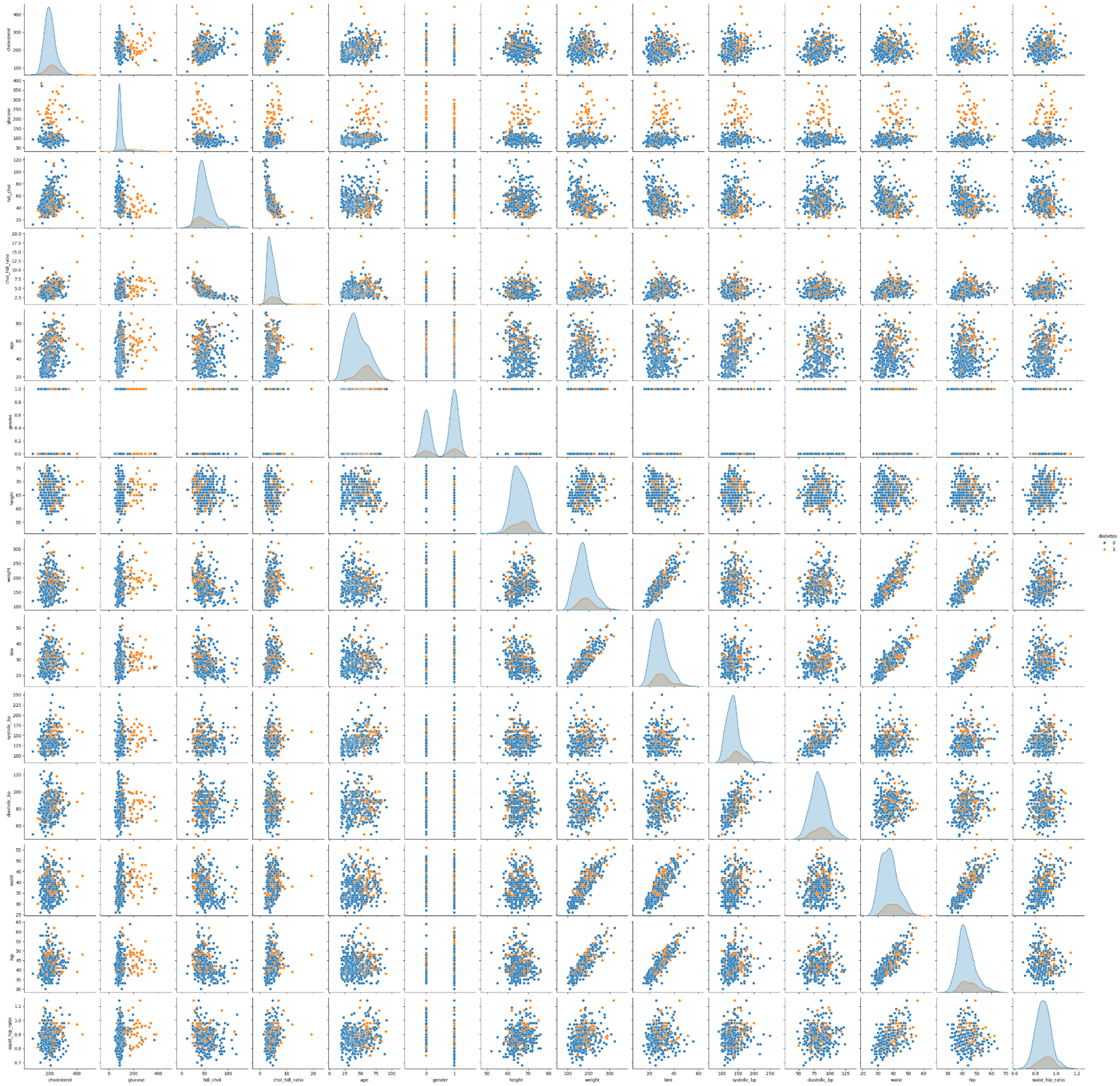
Machine Learning Models Used

Data Visualization for Every Model

- ○ A heatmap is generated to visualize the correlation between different features in the dataset.



- ○ This pairwise plot shows joint and marginal distributions for all pairwise relationships and for each variable, and the color shows diabetes or not diabetes.



Data Preparation

- Load the dataset into a pandas DataFrame.

- All decimal values with ‘,’ are converted to ‘.’ in order to cast them as float types.
- Convert the target variable diabetes into a binary categorical format (0 for "No diabetes", 1 for "Diabetes").
- Convert the gender values into a binary categorical format (male = 0, female = 1).
- Stratified the dataset during splitting such that the proportion of Diabetes vs. non-diabetes remains constant between the training set and test set.
    - The function assert_stratification() performs this check.
    - Training on 80% of the data and testing on 20%.

Naive Bayes Classifier

The Naive Bayes Classifier is based on the principle of Bayes' Theorem, which uses prior knowledge to predict the likelihood of outcomes. It is called "naive" because it assumes that all the features in the dataset are independent of each other, which simplifies the computation but is often not true in real-world data. Despite this, Naive Bayes is effective in many scenarios, especially in text classification and spam detection. It calculates the probability of each class (such as having type 2 diabetes or not) based on the input features and selects the class with the highest probability as the prediction.

Flow of the Naive Bayes Model

- **Model Training**
    - Because the dataset contains a mix of both discrete and continuous features, we use a special Python package: MixedNB classifier (Mixed Naive Bayes) which “implements categorical (multinoulli) and Gaussian naive Bayes algorithms (hence mixed naive Bayes). This means that we

are not confined to the assumption that features (given their respective y's) follow the Gaussian distribution, but also the categorical distribution”. The classifier is trained using the training data.

- **Model Evaluation**
  - The trained model is used to make predictions on the test set. Accuracy is computed to evaluate the model's performance.

Logistic Regression

Logistic Regression is a statistical method used for binary classification. We use the scikit-learn implementation, which includes regularization by default (also used in our model). Although Logistic Regression predicts a probability, we can convert the probability of the target variable (the presence of diabetes, in this case) to a binary classifier by applying a 0.5 threshold.

Flow of Logistic Regression

Model Training

- Train the LogisticRegression model on the training set.

Model Evaluation

- Make predictions on the test set using the trained logistic regression model.
- Calculate and print the accuracy of the model.
- Generate a confusion matrix based on the model's predictions.

Multi-Layer Neural Network

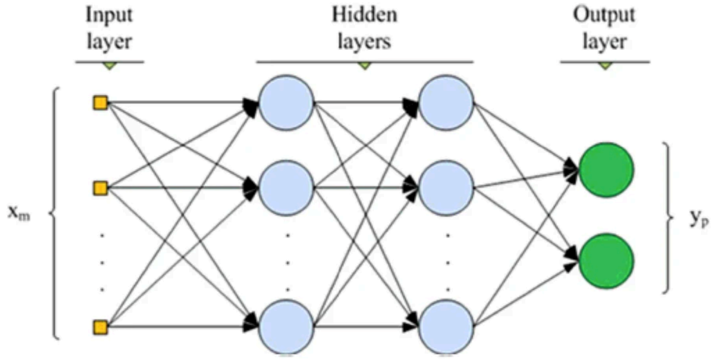
A Multi-Layer Neural Network (MLNN) is part of a broader family of neural networks and is known for its ability to model complex non-linear relationships. It consists of an input layer, several hidden layers, and an output layer. Each layer contains neurons, or nodes, connected to neurons in the subsequent layer. The connections, or weights, between neurons are adjusted during the training process. Neural networks can capture intricate patterns in large datasets, making them suitable for a wide range of applications, including image and speech recognition, and complex classification tasks like predicting diabetes. The depth and complexity of the network allow it to learn detailed hierarchies of features, contributing to its robustness and accuracy in predictions.

Flow of Neural Network

Model Preparation

- The model has 2 hidden layers of 128 and 64 neurons respectively both using tanh activation functions. The output layer is composed of a single neuron since this is a binary classification problem. The sigmoid activation function was used for the output layer.
- The Adam optimizer is used to adjust the weights after each batch. This method adapts the learning rate for each weight.
- Binary cross entropy is used to calculate and minimize the loss between labels and predictions since this is a binary classification problem.

From: A comprehensive review of machine learning techniques on diabetes detection



General representation of a neural network, x_m shows the input weights and y_p is the output weights

Model Training

- Train the Neural Network model on the training set with 50 epochs and a batch size of 10.

Model Evaluation

- Make predictions on the test set using the trained Neural Network model.
- Calculate and print the accuracy of the model.
- Generate a confusion matrix based on the model's predictions.

Results

Evaluation Metrics and Plots Explanation

Accuracy

- Accuracy is a metric used to measure the performance of a classification model. It is defined as the proportion of correct predictions (both true positives and true negatives) made by the model out of all predictions. In simple terms, it tells us what fraction of the total number of cases were correctly identified by the model.

- The formula to calculate accuracy is: Accuracy = Number of Correct Predictions / Total Number of Predictions

Confusion Matrix

- A confusion matrix is plotted to show the number of correct and incorrect predictions.
- The matrix differentiates between true positives, true negatives, false positives, and false negatives for the classes "No Diabetes" and "Diabetes."
- This plot helps in understanding the model's performance in terms of sensitivity (true positive rate) and specificity (true negative rate).

Calibration Plot

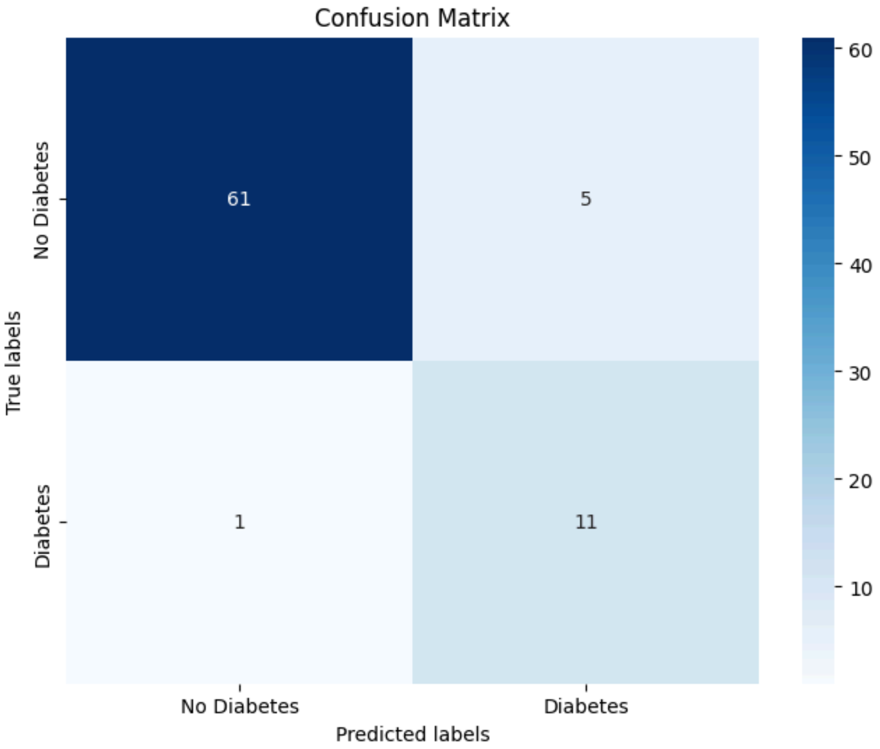
- A calibration plot is created to assess the reliability of the predicted probabilities.
- The plot shows the relationship between the mean predicted probability and the actual accuracy in each bin.
- The ideal line (diagonal gray line) represents perfect calibration where the predicted probabilities match the observed outcomes.
- This plot helps to evaluate whether the model's predicted probabilities of having diabetes are well-calibrated with the actual outcomes.

Results

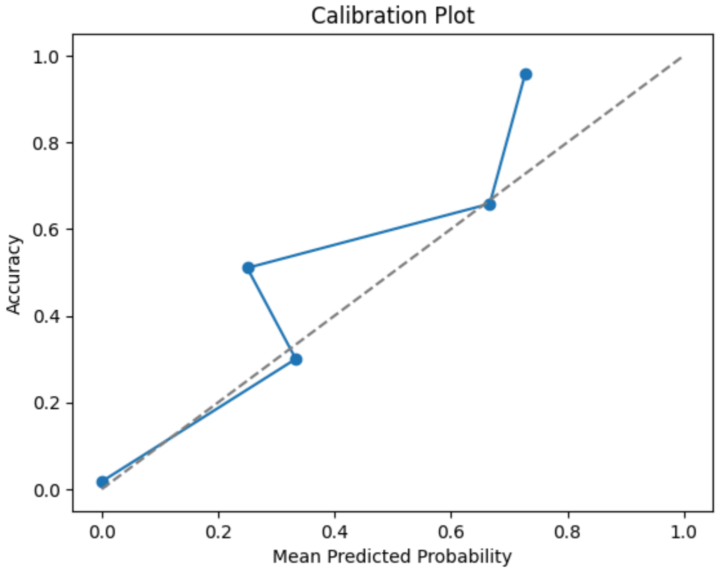
Naive Bayes Classifier

- **Accuracy: 0.9230769230769231**
- **Confusion Matrix:**

- True Negatives (Top-Left): The model correctly predicted 61 instances where the patients did not have diabetes.
- False Positives (Top-Right): There were 5 instances where the model incorrectly predicted diabetes when the patients were actually non-diabetic.
- False Negatives (Bottom-Left): The model incorrectly predicted 1 instances where the patients were diabetic, but the model predicted them as non-diabetic.
- True Positives (Bottom-Right): The model correctly identified 11 instances of diabetes.
- The model thus seems to have a higher number of true negatives and positives, indicating a relatively high accuracy.

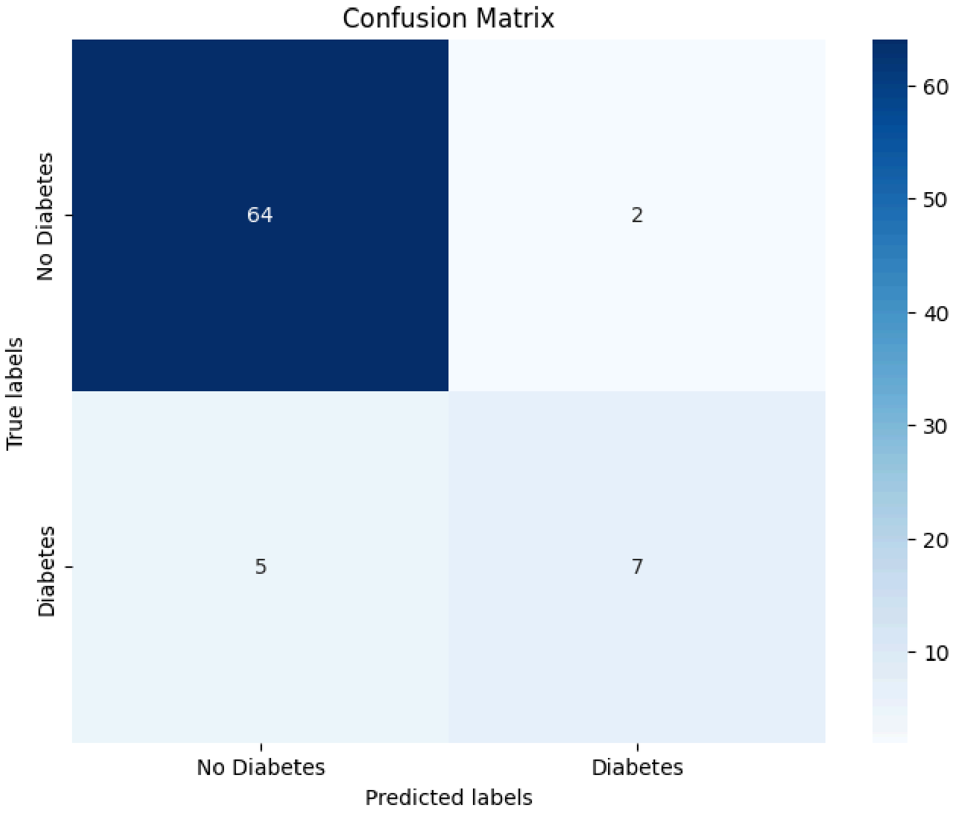


- **Calibration Plot**
  - The calibration plot shows how well the predicted probabilities of diabetes correlate with the actual outcomes. The perfect scenario is when the predicted probability (x-axis) equals the observed frequency of the outcome (y-axis), which would follow the dashed diagonal line.
  - The calibration curve (blue line) shows the model's performance in each bin of predicted probabilities.
  - The points along the blue line indicate the mean predicted probability for each bin on the x-axis and the proportion of positive outcomes (accuracy) for that bin on the y-axis.
  - The model seems well-calibrated in the extremes (close to 0 and 1), but it shows some deviation in the middle probabilities. The points in the middle suggest that the model might be overconfident or underconfident in its predictions for moderate probabilities.

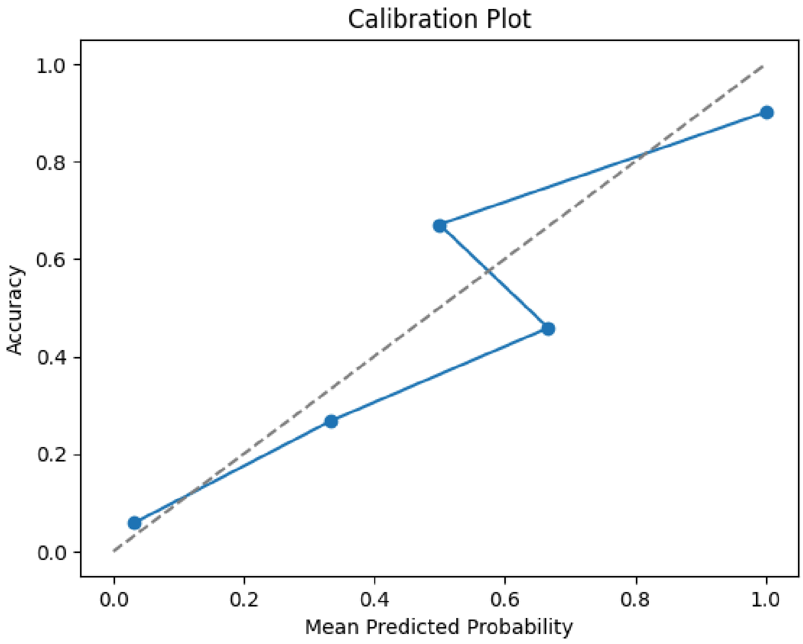


Logistic Regression

- **Accuracy: 0.9102564102564102**
- **Confusion Matrix:**
  - The confusion matrix displays the performance of the logistic regression model on the test dataset for predicting diabetes:
  - True Negatives (Top-Left): The model correctly predicted 'No Diabetes' for 64 instances.
  - False Positives (Top-Right): In only 2 instance, the model incorrectly predicted 'Diabetes' when the patient did not actually have the condition.
  - False Negatives (Bottom-Left): There were 5 instances where the model failed to detect 'Diabetes' (predicting 'No Diabetes' instead).
  - True Positives (Bottom-Right): The model correctly identified 'Diabetes' in 7 instances.
  - The confusion matrix is crucial for understanding not just the model's accuracy but also its sensitivity (ability to detect positives) and specificity (ability to detect negatives).

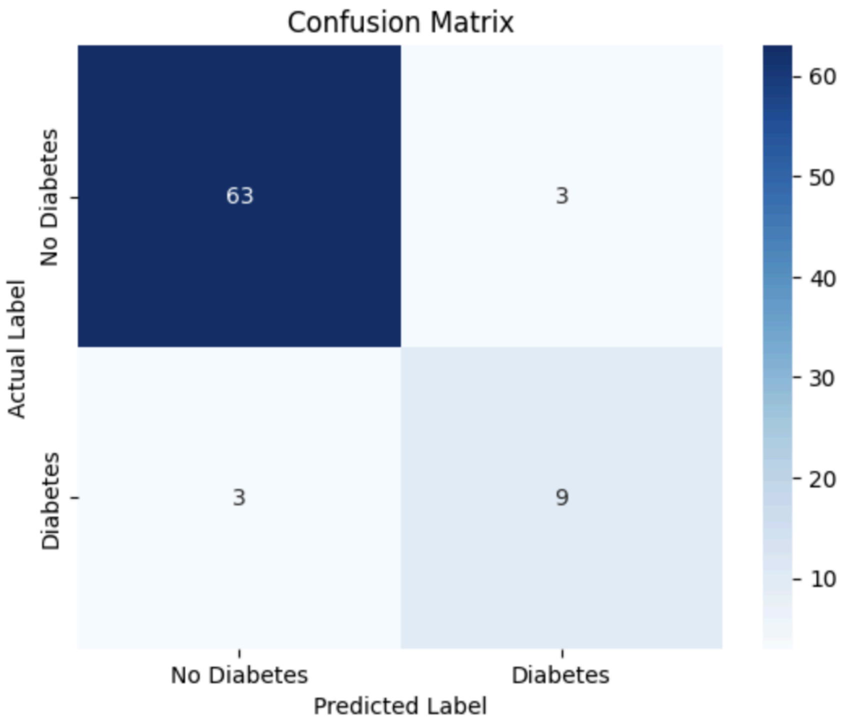


- **Calibration Plot:**
  - Mean Predicted Probability: This is the average probability predicted by the model for each bin of instances.
  - Accuracy: This represents the fraction of correct predictions in each bin.
  - Here we can see that for lower predicted probabilities, the model tends to under-estimate the chances of diabetes, since the blue line is below the diagonal, and for the higher probabilities, it tends to overestimate as the blue line goes above the diagonal.

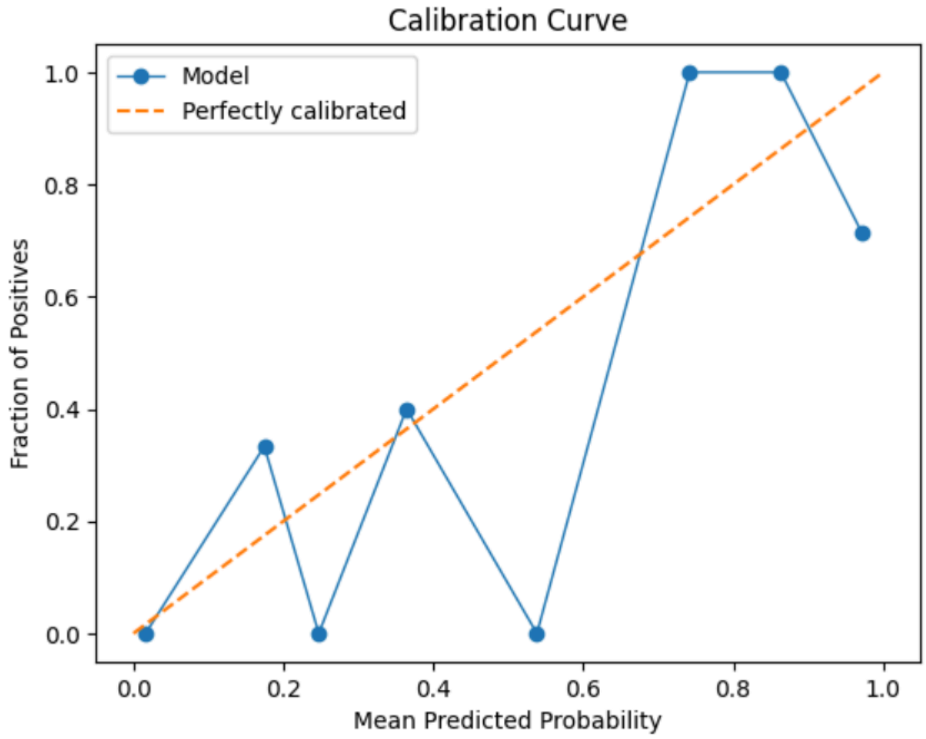


Multi-Layer Neural Network

- **Accuracy: 0.9230769276618958**
- **Confusion Matrix:**
  - ○ True Negatives (Top-Left): The model correctly predicted 'No Diabetes' for 63 instances.
  - ○ False Positives (Top-Right): In only 3 instances, the model incorrectly predicted 'Diabetes' when the patient did not actually have the condition.
  - ○ False Negatives (Bottom-Left): There were 3 instances where the model failed to detect 'Diabetes' (predicting 'No Diabetes' instead).
  - ○ True Positives (Bottom-Right): The model correctly identified 'Diabetes' in 9 instances.



- **Calibration Plot:**
  - In this case the model's predictions don't align very well with the observed event rates. The model performs especially poorly in the midrange, severely underestimating the rate of positive cases.
  - The model's predictions at the extremities do not perform well either. The best performance happens when the mean predicted probability of having type 2 diabetes is low.



Comparative Analysis

With this training and testing set, all three models had around 92% accuracy. The Naive Bayes model was marginally better at predicting true positives and predicted more false positives than the other two models. Both missed diagnoses and mis-diagnoses can lead to untreated conditions and increase chances for complications down the line. Both the Naive Bayes and the Logistic Regression models visually have more correct calibration plots than the Neural Network model which seems crucial in this context: a 55% likelihood of diabetes is very different than a 95% likelihood. In the end the data set was fairly small, with only 390 data points. In this case, Naive Bayes may be the best option in this context since not only is it an interpretable model–which is especially important in medical settings–it also had the best accuracy combined with being well calibrated. However, the probability of each of the parameters of the dataset certainly aren't independent of one another. All in all, it would be worthwhile to compare all the models with more data to see if more meaningful differences can be discovered.

Conclusion

Early detection of type 2 diabetes is critical in improving long term outcomes for patients with the condition. Machine learning models can help predict a patient's likelihood of having the condition making it a worthwhile endeavor to pursue a model that can accurately predict the presence of the illness. In this investigation we explored three machine learning models: Naive Bayes Classifier, Linear Regression, and Multi-Layer Neural Network. Although all three models scored around 92% accuracy on the testing data, because the Naive Bayes model had significantly better calibration and interpretability than the Neural Network and slightly higher accuracy than the Logistic regression, it appears to be the most appropriate model based on the limited dataset used.

Source Code

Google Colab Notebooks:

- Naive Bayes
  https://colab.research.google.com/drive/1FGzkOi9FayrQfc-wwzVzjgA2ifZdaorJ?usp=sharing

- Logistic Regression
  https://colab.research.google.com/drive/1GoDmu8iHZ_ZNqdUI0tBFe756cd1-neTh?usp=sharing

- Multi-Layer Neural Network
  https://colab.research.google.com/drive/1q2YcvIzLkzRbRYWB0tDyoJq4F5tqCx1v?usp=sharing

References

- Dataset used
  - https://www.kaggle.com/datasets/houcembenmansour/predict-diabetes-based-on-diagnostic-measures

- Calibration Plot
  - https://scikit-learn.org/stable/modules/calibration.html

- A comprehensive review of machine learning techniques on diabetes detection (used for Neural Network)
  - https://link.springer.com/article/10.1186/s42492-021-00097-7

- Naive Bayes Package Used
  - https://pypi.org/project/mixed-naive-bayes/

- Logistic Regression Documentation
  - https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression