# MACHINE LEARNING TUTORIAL

# Table of Contents

# 1. Introduction

Machine learning has become one of the most innovative systems in healthcare since it developed different models for forecasting, diagnosis, therapeutic methods, and results (Javaid *et al.,* 2022). Incorporating the Internet into the healthcare sector has the potential to improve its ability to diagnose and monitor chronic conditions such as diabetes, which has cut across millions of the world's population. This report aims to work through the diabetes dataset to show the possibility of applying machine learning techniques. Logistic Regression, Random Forest, and Support Vector Machines (SVM) choices comprised the nature of this report as an educational guide on how to implement these methods for determining the probability of the likelihood of diabetes in a given individual.

# 2. Background and Techniques Overview

Machine learning has transformed the functionality of healthcare by improving the provider's ability to diagnose diseases, as well as the general management of health plans (Haleem *et al.,* 2022). Drawing from big data, ML accurately diagnoses diseases such as diabetes, heart disease, and cancer, thereby improving patients' quality of life and financial savings. Logistic Regression is a supervised learning algorithm used for binary classification tasks, predicting outcomes such as disease presence (diabetes). Random Forest is another method of constructing learners from decision trees where many decision trees work in an ensemble. It avoids overfitting problems and has better performance in handling missing data (Khosravi *et al.,* 2020). Support Vector Machine (SVM) is a known algorithm to classify data by finding a hyperplane that is far from the nearest data points of both classes.

# 3. Dataset Overview

## *3.1 Dataset Description*

Diabetes is the most used dataset for the evaluation of machine learning models in healthcare on the Kaggle platform. This database has 768 rows and 10 column entities, which include the dependent variable. It has useful information regarding potential precursors to diabetes.

## *3.2 Features*

1. The number of pregnancies the patient has had in the past, whether full-term or not.
2. Level of fasting and 2 hours plasma glucose after an oral glucose tolerance test.
3. The last value of the blood pressure measurement is the lower number (mm Hg).
4. The thickness of the skin folds in the Triceps in millimetres.

5. Serum insulin level data used in the analysis include the 2-hour insulin level expressed in mu U/ml.
6. Body Mass Index or BMI is the weight in kilograms divided by the square of height in meters.
7. A function highlighting the conditions promoting diabetes in relation to family heritage.
8. Age of the patient (years).
9. The outcome is the dependent variable, which is the patient's presence of diabetes. 1 if the patient has diabetes and 0 if the patient does not.

## 4. Data Preprocessing

- Although Kaggle does not have any missing records, some of the features, such as Glucose, blood pressure, and skin thickness, have zeros that are not possible. These are then dealt with as the missing data. They imputed the mean of feature mean for simplicity and completeness of data since there was no skewed distribution in the dataset.
- The attributes in the dataset are standardised, so, for instance, Glucose varies from 0 to 200 +, while the DiabetesPedigreeFunction is below 2.0 normally. The resulting scales can be problematic for machine learning algorithms which are sensitive to the scale, the example being Support Vector Machines (SVM). To normalise features, the StandardScaler from sklearn was used to scale features and bring all of them to the value range of [0,1] (Testas, 2023). This makes certain that all characteristics of the model are, in the same way, committed to the decision boundary.
- For the purpose of testing the performance of the model, the dataset was split into Training and Testing Sets, whereby 70% of the data was used for developing the models while 30% was used in the testing of the models. The splitting of data was done by using the function train_test_split from the sklearn category (Zollanvari, 2023). The splitting closed the discrepancy of the class distribution regarding the target variable between both sets.

```python
# Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix,
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.metrics import roc_curve, auc
```

**Figure 1: Importing libraries**

This code imports essential Python libraries for data analysis and machine learning tasks, including NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, and more.

## Loading the dataset

```
data = pd.read_csv('/content/diabetes.csv')
```

## Explore the dataset

```
data.head()
```

| | 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: Exploring Dataset**

This code reads the CSV "diabetes.csv" file into a Pandas DataFrame named "data" and displays the first few rows of the data using the head() method.

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 |
| **mean** | 3.845052 | 120.894531 | 69.105469 | 20.536458 | 79.799479 | 31.992578 | 0.471876 | 33.240885 | 0.348958 |
| **std** | 3.369578 | 31.972618 | 19.355807 | 15.952218 | 115.244002 | 7.884160 | 0.331329 | 11.760232 | 0.476951 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.078000 | 21.000000 | 0.000000 |
| **25%** | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | 0.243750 | 24.000000 | 0.000000 |
| **50%** | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 30.500000 | 32.000000 | 0.372500 | 29.000000 | 0.000000 |
| **75%** | 6.000000 | 140.250000 | 80.000000 | 32.000000 | 127.250000 | 36.600000 | 0.626250 | 41.000000 | 1.000000 |
| **max** | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | 2.420000 | 81.000000 | 1.000000 |

**Figure 3: Summary Statistics of Diabetes Dataset**

This table provides a summary of the numerical features in the diabetes dataset, including count, mean, standard deviation, minimum, 25th percentile (Q1), 50th percentile (median), 75th percentile (Q3), and maximum values.

## Data Preprocessing

```
null_counts = data.isnull().sum()
print("Null values per column:")
print(null_counts)
```

```
Null values per column:
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

```
X = data.drop('Outcome', axis=1)
y = data['Outcome']
```

## Spliting the data into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.
```

## Standardizing the features

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

**Figure 4: Preprocessing steps**

The code demonstrates data preprocessing steps for a machine learning model. It checks for missing values, splits the data into training and testing sets, and standardises the features using StandardScaler. The goal is to prepare the data for model training and evaluation.

## 5. Exploratory Data Analysis (EDA)

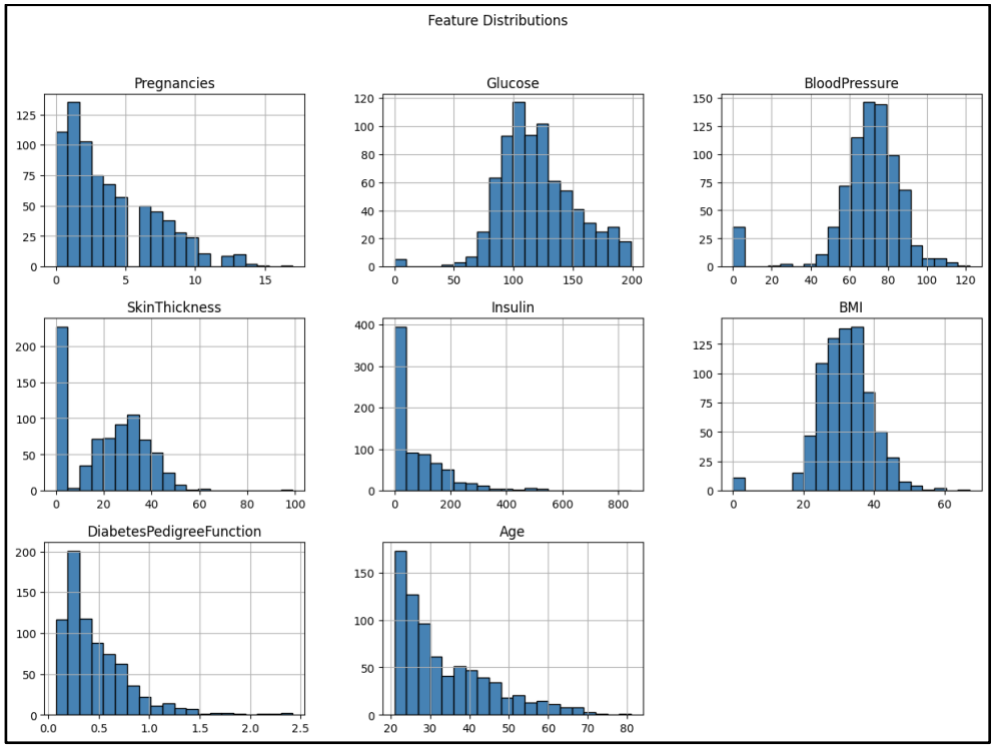


**Figure 5: Feature Distributions**

The distribution of each numerical feature in the diabetes dataset shows that most features have a right-skewed distribution, meaning they have a long tail towards the higher values. This indicates that there are some outliers in the data. Additionally, some features such as "SkinThickness" and "Insulin" have many 0 values, which might require further investigation.

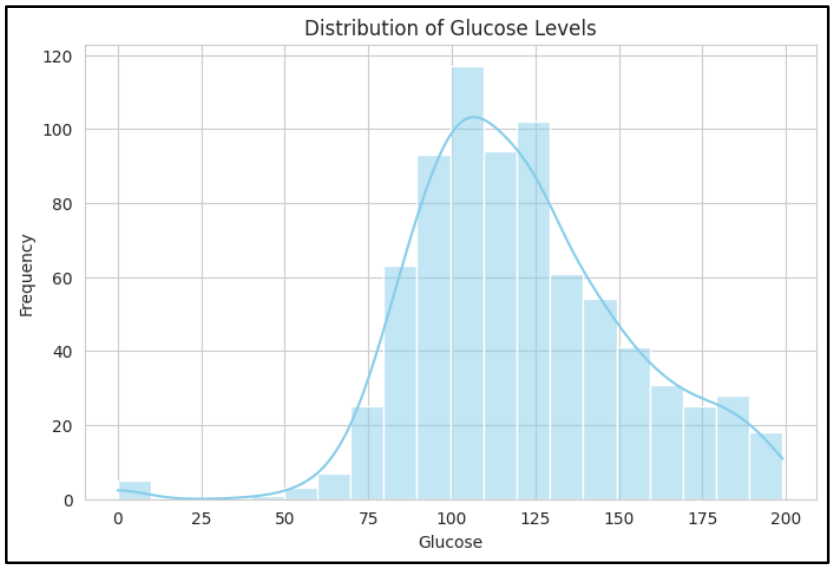


**Figure 6: Distribution of Glucose Levels**

The histogram shows the distribution of glucose levels in the dataset. The distribution is right-skewed, suggesting most individuals have lower glucose levels, with a few having significantly higher values. The peak around 100 suggests that this is the most common glucose level in the dataset.

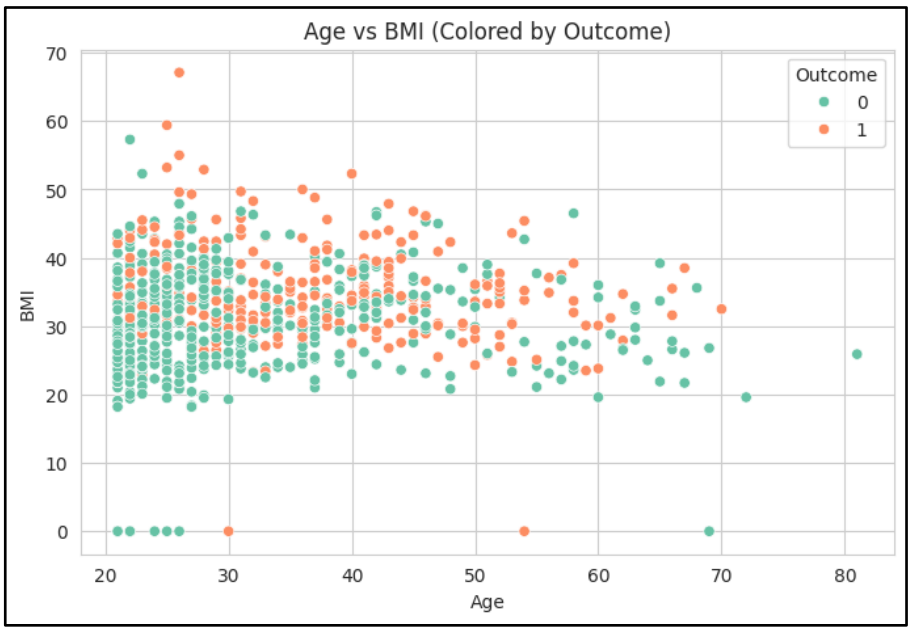


**Figure 7: Age vs BMI (Colored by Outcome)**

The scatter plot visualises the relationship between age and BMI, with the points coloured by the outcome variable (0 or 1). There is no clear linear relationship between age and BMI. However, individuals with higher BMI values are more likely to have an outcome of 1.

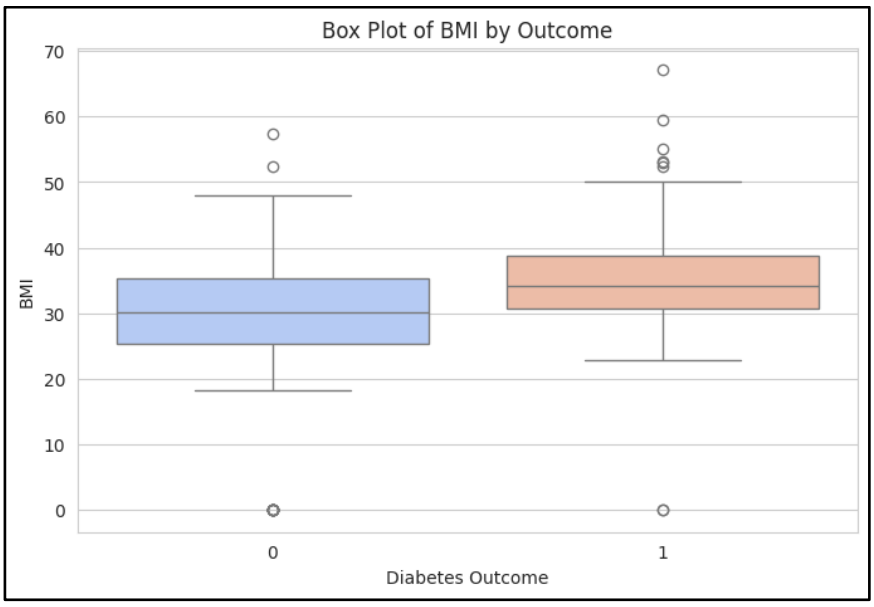


**Figure 8: Box Plot of BMI by Outcome**

The box plot compares the distribution of BMI values for individuals with diabetes (Outcome = 1) and those without diabetes (Outcome = 0). Individuals with diabetes tend to have higher BMI values on average. The box plot also shows that there is more variability in BMI among individuals with diabetes.

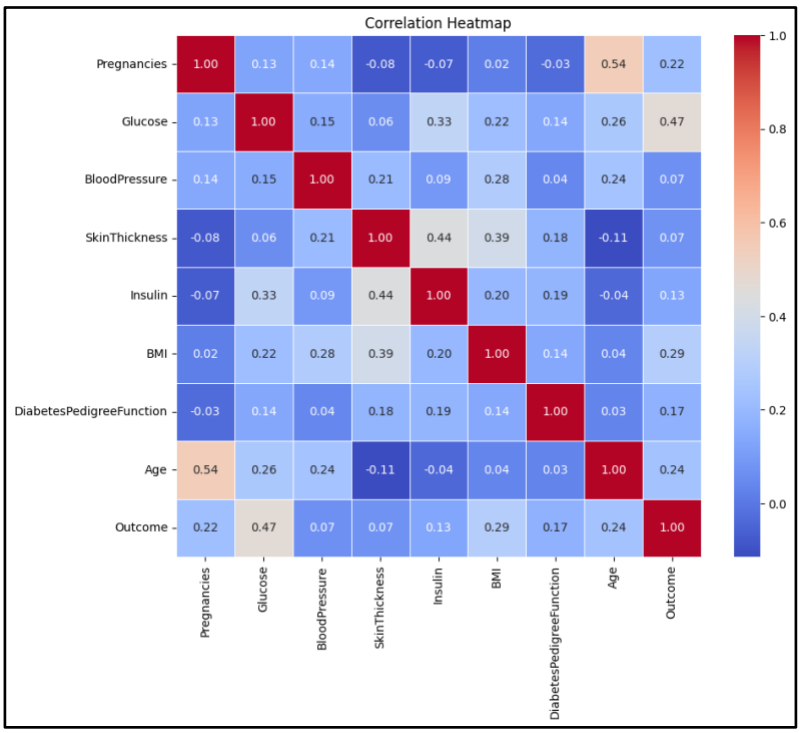


**Figure 9: Correlation Heatmap**

The correlation heatmap visualises the relationships between different features in the diabetes dataset. Glucose, BMI, and age show positive correlations with the outcome. Conversely, some features such as skin thickness have a negative correlation, suggesting that lower values of these features might be associated with a higher risk of diabetes. There are some moderate correlations between features such as glucose and BMI, indicating that these variables might be related to each other.

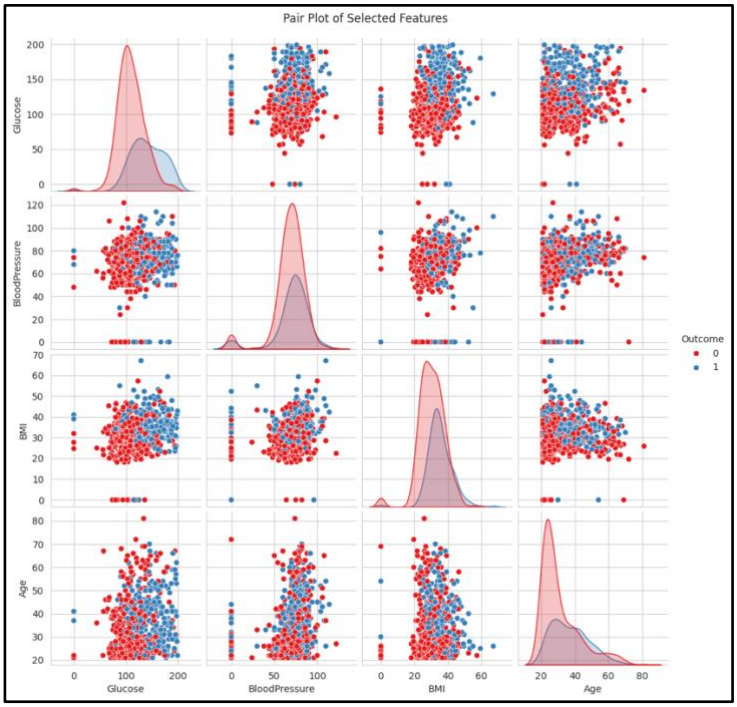


**Figure 10: Pair Plot of Selected Features**

The scatter plot illustrates the interactions of two features chosen out of glucose, blood pressure, BMI, age and diabetes. Glucose seems to be positively associated with the result, while the results indicating diabetes seem to present a higher glucose level for the subject. The positive impact of blood pressure is also revealed, but the effect is weaker than that of glucose. BMI is positively related and moderately related to the outcome.

### *5.1 Insights*

The results of the exploration of the diabetes dataset point out certain features of it. Quite a number of variables, including glucose levels, have a positive skewness, meaning that they have higher variability among the population than illustrated by the measures. Leverage values such as “SkinThickness” and “Insulin” include zeros. Hence, they require special attention when modelling. According to histograms and scatter plots, glucose has a positive impact on diabetic results, while BMI also plays a role in determining diabetic results, with high BMI having high diabetic results. The correlation heatmap for diabetes reveals high correlations in glucose level, BMI, and age and thus can be used when creating feature selection for the predictive models.

## 6. Model Implementation and Training

Logistic Regression Evaluation:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 0.90 | 0.85 | 157 |
| 1 | 0.71 | 0.53 | 0.60 | 74 |
| accuracy | | | 0.78 | 231 |
| macro avg | 0.76 | 0.71 | 0.73 | 231 |
| weighted avg | 0.77 | 0.78 | 0.77 | 231 |

**Figure 11: Logistic Regression**

The evaluation metrics for a logistic regression model show an accuracy of 78%. It performs better on class 0 with a precision of 0.80 and recall of 0.90. However, it struggles with class 1, having a lower precision of 0.71 and recall of 0.53. The F1-score, which balances precision and recall, is 0.85 for class 0 and 0.60 for class 1.

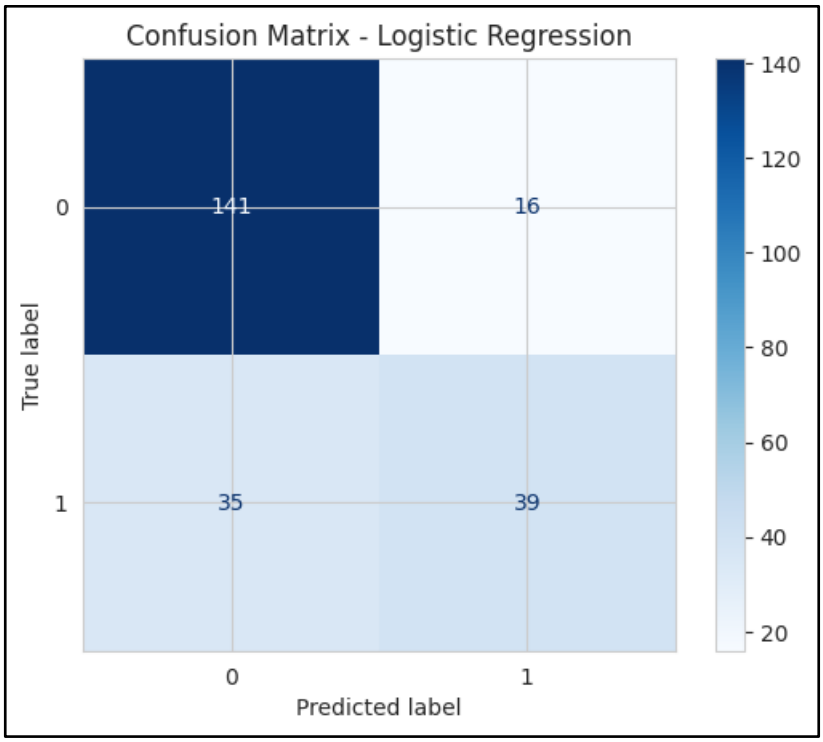


**Figure 12: Confusion matrix- Logistic regression**

The matrix visualises the performance of a logistic regression model on a binary classification task. The model correctly predicts 141 instances of class 0 and 39 instances of class 1. However, it misclassifies 16 instances of class 0 as class 1 and 35 instances of class 1 as class 0. This suggests that the model has better performance for class 0 compared to class 1.

Random Forest Evaluation:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 0.89 | 0.85 | 157 |
| 1 | 0.70 | 0.54 | 0.61 | 74 |
| accuracy | | | 0.78 | 231 |
| macro avg | 0.75 | 0.72 | 0.73 | 231 |
| weighted avg | 0.77 | 0.78 | 0.77 | 231 |

**Figure 13: Random Forest**

Random Forest model has achieved an overall accuracy of 78%. It performs better on class 0 with a precision of 0.80 and recall of 0.89. However, it struggles with class 1, similar to the previous data, having a lower precision of 0.70 and recall of 0.54. The F1-score, which balances precision and recall, is 0.85 for class 0 and 0.61 for class 1.

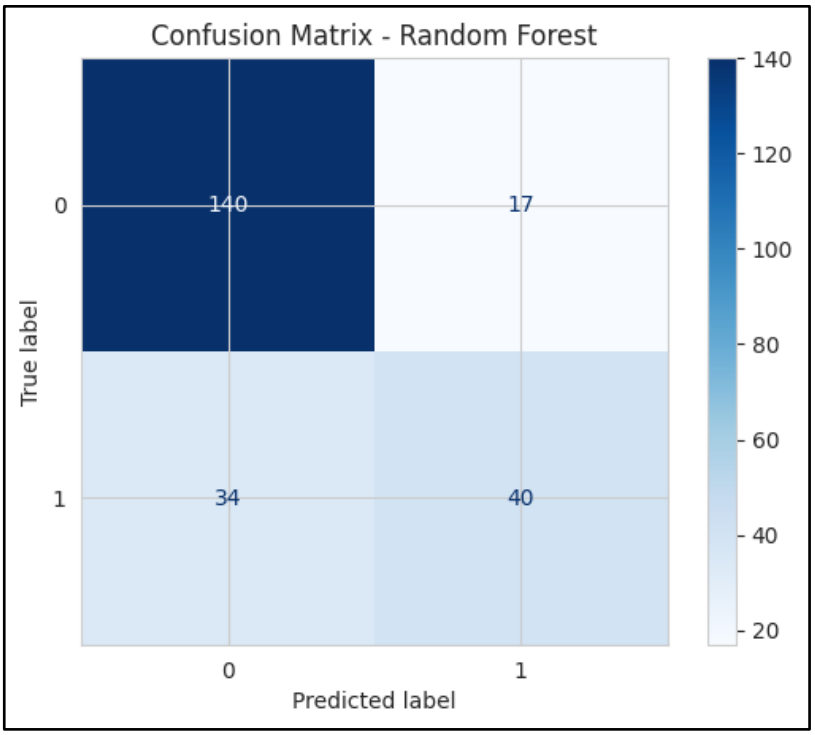

**Figure 14: Confusion matrix- Random Forest**

The Random Forest model on a binary classification task correctly predicts 140 instances of class 0 and 40 instances of class 1. However, it misclassifies 17 instances of class 0 as class 1 and 34 instances of class 1 as class 0. This suggests that the model has slightly better performance for class 0 compared to class 1.

Support Vector Machine Evaluation:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.90 | 0.85 | 157 |
| 1 | 0.71 | 0.54 | 0.62 | 74 |
| accuracy | | | 0.78 | 231 |
| macro avg | 0.76 | 0.72 | 0.73 | 231 |
| weighted avg | 0.78 | 0.78 | 0.77 | 231 |

**Figure 15: SVM**

The figure contains the evaluation results of the SVM model. The proposed model gains an effectiveness rate of 76% in the general classification. For class 0, it provides a higher precision of 0.81 and recall of 0.90. It is inefficient in class 1, reaching 0.71 of precision rates and 0.54 of recall rates. The average of precision and recall is 0.85 for class 0 and 0.62 for class 1.

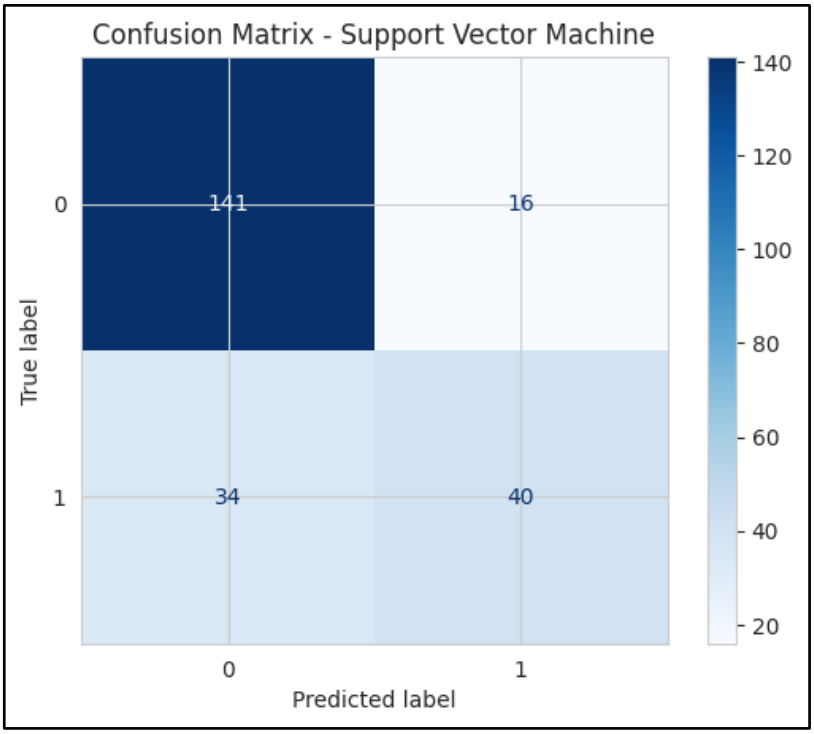


**Figure 16: Confusion matrix- SVM**

The model correctly predicts 141 instances of class 0 and 40 instances of class 1. However, it misclassifies 16 instances of class 0 as class 1 and 34 instances of class 1 as class 0. This suggests that the model has slightly better performance for class 0 compared to class 1.

## 7. Model Evaluation and Comparison

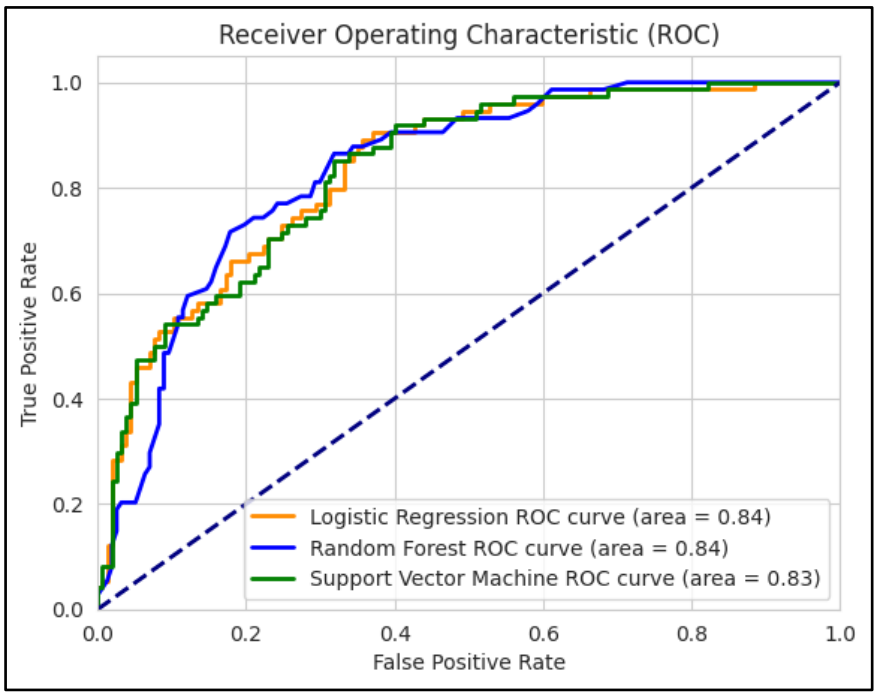


**Figure 17: Receiver Operating Characteristic (ROC) Curve**

This ROC curve compares the performance of three models: Logistic Regression, Random Forest, and Support Vector Machine. All models show good performance with AUC values close to 0.84. Logistic Regression and Random Forest have almost identical ROC curves, while SVM has a slightly lower AUC.

## 8. Discussion and Insights

The focus has been made on the machine learning models applied to predict diabetes. The usefulness of such characteristics as glucose level, BMI, and age is stressed. The accuracy of Logistic Regression, Random Forest, and SVM was found to be nearly similar, being around 76-78%. However, the cross-validation of the variables confirmed more precision for non-diabetic samples (Class 0) relative to the diabetic sample (Class 1). Both Logistic Regression and Random Forest had better precision and lower recall for class 1 than SVM. The ROC curves illustrate equivalent AUC, with slight superiority of Logistic Regression and Random Forest >SVM. These findings also stress the significance of feature engineering skills and the utilities of feature-balanced datasets for enhancing the model's performance.

## 9. Conclusion

The study provided preliminary evidence of the potential of applying Logistic Regression, Random Forest, and SVM in diabetic prediction that yielded a reasonably high accuracy and included glucose, BMI, and age as the critical factors. Yet, the models failed to perform well in determining diabetic cases, particularly indicating a need for a higher level of features such as selection of the features, oversampling or another set of data. The presented study highlights the potential of ML in the healthcare sector while identifying the directions for the improvement of prediction outcomes and class distribution.

## References


Abdul, A., Isiaka, R.M., Babatunde, R.S. and Ajao, J.F., 2021. An Improved Coronary Heart Disease Predictive System Using Random Forest. *Asian J. Res. Comput. Sci*, *11*(1), pp.17-27. https://www.academia.edu/download/74238879/56780.pdf

Alowais, S.A., Alghamdi, S.S., Alsuhebany, N., Alqahtani, T., Alshaya, A.I., Almohareb, S.N., Aldairem, A., Alrashed, M., Bin Saleh, K., Badreldin, H.A. and Al Yami, M.S., 2023. Revolutionizing healthcare: the role of artificial intelligence in clinical practice. *BMC medical education*, *23*(1), p.689. https://link.springer.com/article/10.1186/s12909-023-04698-z

Behl, R. and Kashyap, I., 2020. Machine learning classifiers. In *Big Data, IoT, and Machine Learning* (pp. 3-36). CRC Press. https://www.taylorfrancis.com/chapters/edit/10.1201/9780429322990-2/machine-learning-classifiers-rachna-behl-indu-kashyap

de Amorim, L.B., Cavalcanti, G.D. and Cruz, R.M., 2023. The choice of scaling technique matters for classification performance. *Applied Soft Computing*, *133*, p.109924. https://www.sciencedirect.com/science/article/pii/S1568494622009735

Javaid, M., Haleem, A., Singh, R.P., Suman, R. and Rab, S., 2022. Significance of machine learning in healthcare: Features, pillars and applications. *International Journal of Intelligent Networks*, *3*, pp.58-73. https://www.sciencedirect.com/science/article/pii/S2666603022000069

Khosravi, P., Vergari, A., Choi, Y., Liang, Y. and Broeck, G.V.D., 2020. Handling missing data in decision trees: A probabilistic approach. *arXiv preprint arXiv:2006.16341*. https://arxiv.org/abs/2006.16341

Mienye, I.D. and Sun, Y., 2022. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, *10*, pp.99129-99149. https://ieeexplore.ieee.org/abstract/document/9893798/

Ozsahin, D.U., Mustapha, M.T., Mubarak, A.S., Ameen, Z.S. and Uzun, B., 2022, August. Impact of feature scaling on machine learning models for the diagnosis of diabetes. In *2022*

*International Conference on Artificial Intelligence in Everything (AIE)* (pp. 87-94). IEEE. https://ieeexplore.ieee.org/abstract/document/9898687/

Shah, A.R., Mathew, A.A., Karthikeyan, V. and Shanmugasundaram, V., 2024, August. Patient Health Classification Using Various Machine Learning Algorithms. In *2024 Control Instrumentation System Conference (CISCON)*(pp. 1-6). IEEE. https://ieeexplore.ieee.org/abstract/document/10696145/

Shiwlani, A., Khan, M., Sherani, A.M.K., Qayyum, M.U. and Hussain, H.K., 2024. REVOLUTIONIZING HEALTHCARE: THE IMPACT OF ARTIFICIAL INTELLIGENCE ON PATIENT CARE, DIAGNOSIS, AND TREATMENT. *JURIHUM: Jurnal Inovasi dan Humaniora*, *1*(5), pp.779-790. http://jurnalmahasiswa.com/index.php/Jurihum/article/view/845

Testas, A., 2023. Support Vector Machine Classification with Pandas, Scikit-Learn, and PySpark. In *Distributed Machine Learning with PySpark: Migrating Effortlessly from Pandas and Scikit-Learn* (pp. 259-280). Berkeley, CA: Apress. https://link.springer.com/chapter/10.1007/978-1-4842-9751-3_10

Zollanvari, A., 2023. Supervised Learning in Practice: the First Application Using Scikit-Learn. In *Machine Learning with Python: Theory and Implementation* (pp. 111-131). Cham: Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-031-33342-2_4