P.E.S COLLEGE OF ENGINEERING, MANDYA

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

A Project Report on on

"Data Analytics"

submitted in partial fulfilment of the requirements for
the completion of the 6th Semester Project.
Bachelor of Engineering

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

SYEDA AFREEN - [4PS22CS172]
TEJASHWINI M P - [4PS22CS174]
SWATHI H U - [4PS22CS170]
UMME AIMAN - [4PS22CS181]
SHASHI NAGARATHNA M M - [4PS22CS149]
SNEHA K M - [4PS22CS156]

Submitted to

Dr.Deepika

Assistant Professor, Dept of CS&E,

PESCE, Mandya.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
P.E.S. College of Engineering, Mandya
2024-2025

P.E.S COLLEGE OF ENGINEERING
MANDYA-571401

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that SYEDA AFREEN [4PS22CS172], TEJASHWINI M P [4PS22CS174], SWATHI H U [4PS22CS170], UMME AIMAN [4PS22CS181], SHASHI NAGARATHNA M M [4PS22CS149], SNEHA K M [4 PS22CS156] student of 6th Semester B.E in Computer Science & Engineering of P.E.S College of Engineering, Mandya, has satisfactorily completed the Project on **Data Analytics** during the year 2024-25. The project report has been approved as it satisfies the academic requirements with respect to project work prescribed for the 6th Semester of Computer Science and Engineering discipline.

Signature of Guide
Dr. Deepika
Assistant Professor of CS&E,
Dept of CS&E, PESCE, Mandya.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without acknowledging the individuals who made it possible and provided constant guidance and encouragement throughout the process.

We would like to express my deep sense of gratitude to **Dr. N. L. Muralikrishna,** PESCE, Mandya for his timely help and inspiration during the course.

We extend our sincere gratitude to the activity coordinator and mentor**, Dr. Deepika**, Assistant Professor of CS&E, Department of CSE, for her encouragement and support throughout the project. We consider it our cardinal duty to express our deepest gratitude for the invaluable guidance she provided at every stage and in every possible way.

Finally, we are very thankful to all the faculty members of the Department of CSE, our seniors, friends, and our parents for their constant encouragement, support, and help throughout the period of the activity conduction.

ABSTRACT

Diabetes is a rapidly growing health concern worldwide, particularly due to its silent and chronic nature. With the emergence of machine learning, accurate prediction models can significantly improve early detection and treatment. This report presents a machine learning pipeline built using Python and Scikit-learn to predict diabetes based on various physiological and clinical features such as glucose level, BMI, insulin, and age. The model uses the Support Vector Machine (SVM) algorithm with a linear kernel, which is efficient for binary classification problems. After performing data cleaning, outlier treatment, standardization, and stratified train-test splitting, the model is trained and evaluated, achieving a test accuracy of approximately 77.27%. This result illustrates the model's potential to support clinical decisions, although further tuning and enhancements could improve performance.

Diabetes Prediction Using Support Vector Machine (SVM)

1.Introduction

Diabetes mellitus is a chronic, non-communicable disease characterized by elevated levels of blood glucose (sugar), which over time can lead to serious damage to the heart, blood vessels, eyes, kidneys, and nerves. There are mainly three types of diabetes: Type 1, Type 2, and gestational diabetes. Type 1 diabetes results from autoimmune destruction of insulin-producing beta cells in the pancreas. Type 2 diabetes, the most common form, occurs when the body becomes resistant to insulin or the pancreas fails to produce enough insulin. Gestational diabetes affects pregnant women and can lead to complications for both mother and child.

According to the International Diabetes Federation, more than 537 million adults were living with diabetes in 2021, and this number is expected to reach 783 million by 2045. What makes the disease more concerning is that many people remain undiagnosed until significant complications occur. Lifestyle changes, genetic predisposition, obesity, and lack of physical activity are some of the major risk factors contributing to the global diabetes epidemic.

Importance of Early Prediction

Early prediction and timely diagnosis of diabetes play a crucial role in preventing or delaying complications. Detecting diabetes at an early stage allows patients to receive appropriate medical intervention, adopt healthier lifestyles, and monitor their blood sugar levels more effectively. In clinical practice, early diagnosis typically relies on blood tests such as fasting glucose levels, oral glucose tolerance tests, and HbA1c. However, these tests require clinical infrastructure and may not always be feasible in low-resource settings.

With the rise of digital health records and biomedical datasets, there is a growing opportunity to develop computational models that can assess diabetes risk based on non-invasive or routine health metrics. Predictive analytics can assist clinicians in making faster decisions, prioritizing high-risk patients, and optimizing the allocation of medical resources. Moreover, such systems can be integrated into mobile health applications to make early diagnosis more accessible to the general population.

Role of Machine Learning

Machine learning (ML) has become a valuable tool in the medical field, particularly for predictive analytics. By learning patterns from historical health data, ML models can identify patients at risk of developing chronic diseases like diabetes. In this project, we apply a Support Vector Machine (SVM) with a linear kernel to predict diabetes using clinical features such as glucose, insulin, BMI, and age. This approach illustrates how ML can support healthcare professionals in making early and accurate diagnostic decisions.

2. Dataset Description

Features and Labels

The dataset used in this project is the **Pima Indians Diabetes Dataset**, a popular benchmark dataset in the machine learning community. It contains a total of **768 patient records** and **9 attributes**, where the first 8 attributes represent **independent features** (predictors), and the last attribute represents the **target label** (diabetes status: 0 for non-diabetic and 1 for diabetic). All patients are **female of Pima Indian heritage**, aged **21 years and older**.

Here is a breakdown of the features:

Column Name	Description
Pregnancies	Number of times the patient has been pregnant
Glucose	Plasma glucose concentration (mg/dL)
Diastolic	Diastolic blood pressure (mm Hg)
Triceps	Skinfold thickness (mm)
Insulin	2-hour serum insulin (mu U/ml)
BMI	Body Mass Index (weight in kg / (height in m)^2)
DPF	Diabetes Pedigree Function (family history influence)
Age	Age of the patient (in years)
Diabetes	Target label (0 = non-diabetic, 1 = diabetic)

These attributes represent both physiological and clinical risk factors associated with Type 2 diabetes. The goal of the machine learning model is to **classify** a new patient's data as diabetic or not based on these measurements.

Preliminary Data Checks

Before any model training, the dataset was subjected to basic validation and structure checks using Python and Pandas.

1. **Shape of the Dataset**

df.shape

Output: (768, 9) confirms 768 records with 9 columns.

2. **Check for Null Values**

df.isnull().sum()

Output: All columns return zero, indicating there are **no missing values**. However, fields like Glucose, Insulin, and Triceps contain biologically unrealistic zero values (e.g., glucose = 0), which are treated as **outliers** in later steps.

3. **Class Distribution**

df['diabetes'].value_counts()

o **500** non-diabetic (label 0)

o **268**diabetic (label 1)

This slight class imbalance is addressed using **stratified sampling** during the train-test split.

4. **Feature Means by Class**

df.groupby('diabetes').mean()

This helps identify how feature averages differ between diabetic and non-diabetic cases. For example, diabetics tend to have higher average glucose, insulin, and BMI values compared to non-diabetics.

3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is an essential step in understanding the structure, distribution, and quality of a dataset before applying machine learning models. In this project, EDA was performed using Python libraries such as Pandas, Seaborn, and Matplotlib to uncover patterns and irregularities in the Pima Indians Diabetes Dataset.

Summary Statistics

df.describe()

This generates key statistics such as:

- **Mean**: Average value of each feature
- **Std**: Standard deviation (spread)
- **Min & Max**: Range of values

- **25%, 50%, 75%**: Quartiles

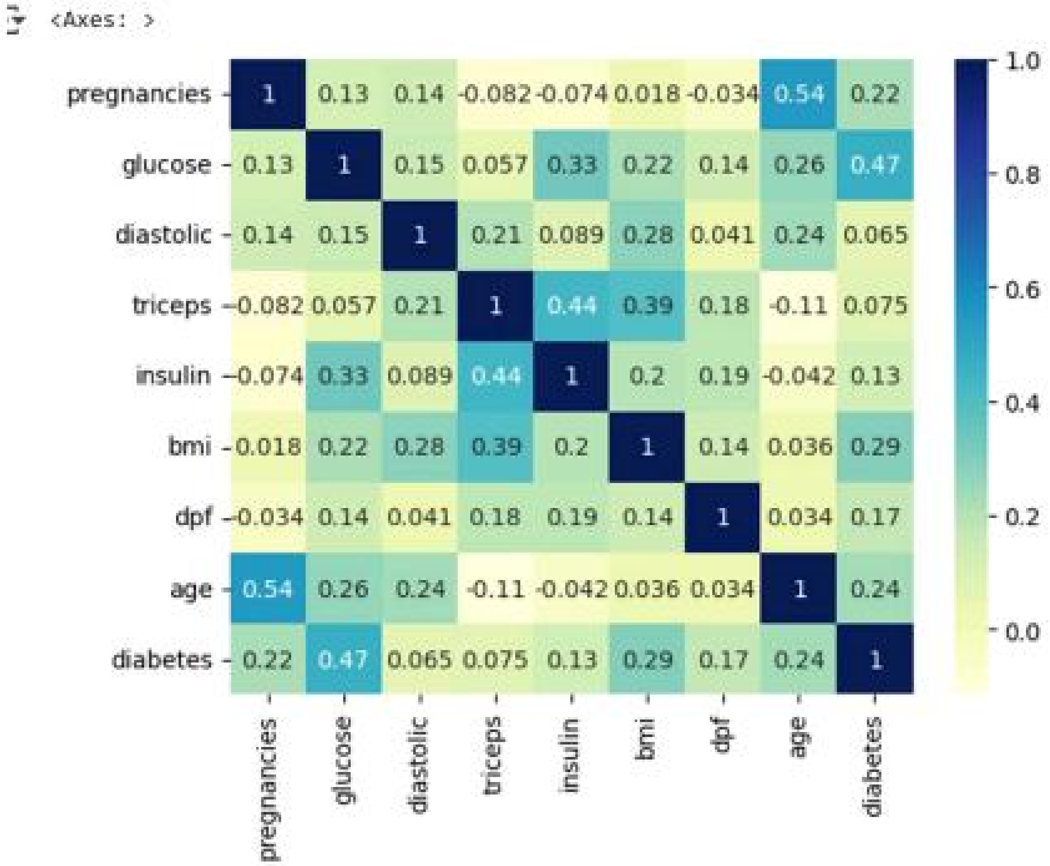
For example:

- Glucose: mean ≈ 120.9, min = 0, max = 199
- BMI: mean ≈ 32, range = 0 to 67.1
- Insulin: mean ≈ 79.8, with several zero values

These values highlight unrealistic medical entries (like glucose = 0) which are potential outliers to be corrected later.

Correlation Matrix

A Pearson correlation matrix was computed to examine linear relationships between features and the target variable:



```
<Axes: >
```

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age	diabetes
pregnancies	1	0.13	0.14	-0.082	-0.074	0.018	-0.034	0.54	0.22
glucose	0.13	1	0.15	0.057	0.33	0.22	0.14	0.26	0.47
diastolic	0.14	0.15	1	0.21	0.089	0.28	0.041	0.24	0.065
triceps	-0.082	0.057	0.21	1	0.44	0.39	0.18	-0.11	0.075
insulin	-0.074	0.33	0.089	0.44	1	0.2	0.19	-0.042	0.13
bmi	0.018	0.22	0.28	0.39	0.2	1	0.14	0.036	0.29
dpf	-0.034	0.14	0.041	0.18	0.19	0.14	1	0.034	0.17
age	0.54	0.26	0.24	-0.11	-0.042	0.036	0.034	1	0.24
diabetes	0.22	0.47	0.065	0.075	0.13	0.29	0.17	0.24	1

- ❖ **Glucose (0.47)** shows the strongest positive correlation with diabetes.
- ❖ **BMI, Age, and DPF** also show moderate correlations.
- ❖ **Triceps and Diastolic** show weaker relationships with the target.

4. Feature Scaling

Need for Scaling

Feature scaling is a critical step in preparing data for machine learning algorithms like Support Vector Machine (SVM), which are sensitive to the magnitude of input values. In datasets like the Pima Indians Diabetes Dataset, numerical features vary widely in range:

- **Glucose** ranges from 0 to 199
- **Insulin** from 0 to over 800
- **BMI** from 0 to 67
- **Diabetes Pedigree Function (DPF)** from 0.078 to 2.42

Without scaling, features with larger numerical values (e.g., Insulin or Glucose) can **dominate** those with smaller values (e.g., DPF), causing the model to become biased during training. Scaling ensures that **each feature contributes equally** to the decision-making process, leading to faster convergence and improved model performance.

StandardScaler Usage

To normalize the features, we used **StandardScaler** from Scikit-learn. This method **standardizes** data by removing the mean and scaling to unit variance:

```
from sklearn.preprocessing import StandardScaler

scale = StandardScaler()
x_scaled = scale.fit_transform(x)
```

Here's what happens under the hood for each feature:

z=x−μσz = \frac{x - \mu}{\sigma}z=σx−μ

Where:

- xxx is the original value
- μ\muμ is the mean
- σ\sigmaσ is the standard deviation
- zzz is the standardized value

After applying StandardScaler, each feature will have:

- **Mean ≈ 0**
- **Standard deviation ≈ 1**

This transformation is particularly suitable for algorithms that compute distances or dot products between feature vectors (like SVM).

5.Train-Test Split

Purpose and Method

To assess the generalization performance of a machine learning model, it is standard practice to split the dataset into training and testing subsets. The training set is used to build and tune the model, while the testing set evaluates how well the model performs on unseen data.

In this project, the dataset was split using Scikit-learn's train_test_split function:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(
    x_scaled, y, test_size=0.2, stratify=y, random_state=2
)
```

- Test size: 20% of the dataset
- Training size: 80% of the dataset
- Random state: 2 (for reproducibility)

Stratification Explanation

Stratification ensures that the proportion of diabetic and non-diabetic cases remains consistent across both training and test sets. Without stratification, a random split might result in an imbalanced subset, causing inaccurate evaluation or training.

For example:

Class Label	Count in Full Data	Count in Train Set	Count in Test Set
0 (No)	500	400	100
1 (Yes)	268	214	54

By stratifying on y, we preserve the original class distribution in both subsets, improving model reliability, especially for imbalanced datasets.

Train and Test Set Shapes

After splitting, the resulting datasets had the following shapes:

```
print(X_train.shape)  # (614, 8)
print(X_test.shape)   # (154, 8)
```

This means the model was trained on 614 records and tested on 154 records, with 8 features in each.

6.Visualization (3D Scatter Plot)

Plotting Glucose vs BMI vs Insulin

Axes Description

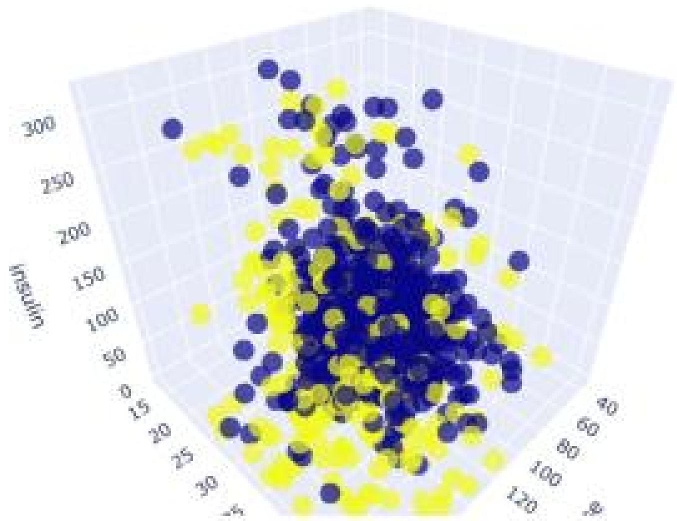
- X-axis: Glucose
- Y-axis: BMI (Body Mass Index)
- Z-axis: Insulin

These are three of the most influential features when predicting diabetes.

Color Coding

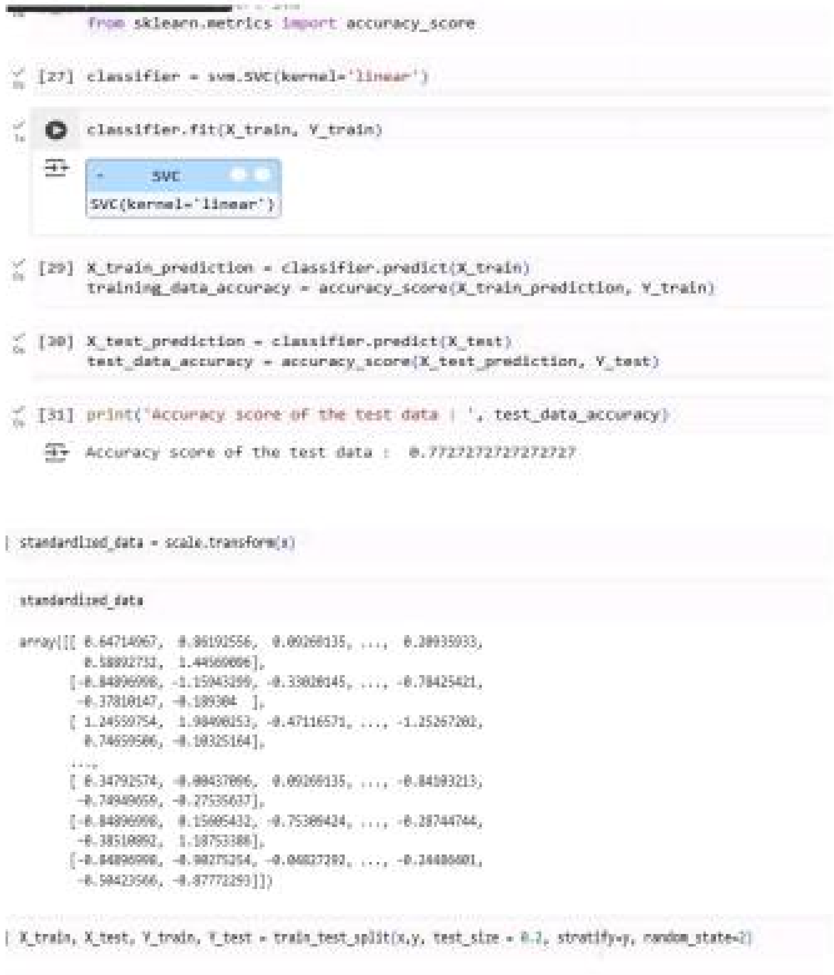
- Yellow Dots: Patients with diabetes (label = 1)
- Dark Blue Dots: Patients without diabetes (label = 0)

This color separation allows you to visually observe the distribution of diabetic vs. non-diabetic patients in the 3D feature space.



7.Linear Regression

A Support Vector Machine (SVM) classifier with a linear kernel was implemented to perform a classification task. The model was trained on the training dataset and evaluated using accuracy as the primary performance metric.



```
from sklearn.metrics import accuracy_score

[27] classifier = svm.SVC(kernel='linear')

     classifier.fit(X_train, Y_train)

     SVC
     SVC(kernel='linear')

[29] X_train_prediction = classifier.predict(X_train)
     training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

[30] X_test_prediction = classifier.predict(X_test)
     test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

[31] print('Accuracy score of the test data : ', test_data_accuracy)

     Accuracy score of the test data :  0.7727272727272727
```

```
standardized_data = scale.transform(x)
```

```
standardized_data
```

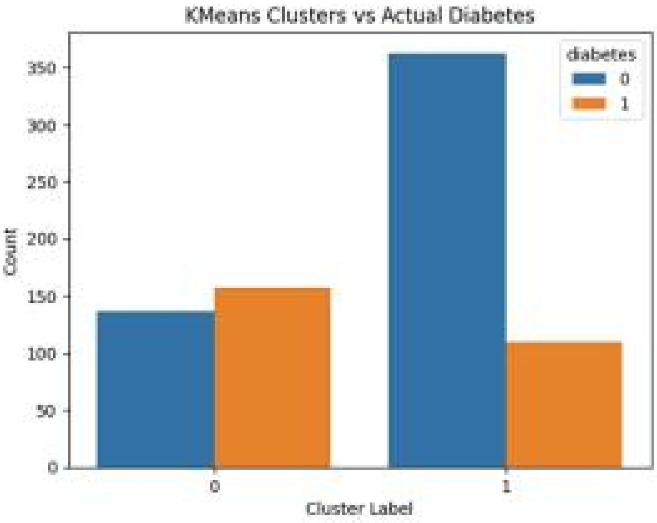
```
array([[ 0.64714067,  0.86192556,  0.09260135, ...,  0.20935933,
         0.58892732,  1.44569086],
       [-0.84896998, -1.15943299, -0.33020145, ..., -0.78425421,
        -0.37810147, -0.189304  ],
       [ 1.24559754,  1.98498253, -0.47116571, ..., -1.25267202,
         0.74659506, -0.10325164],
       ...,
       [ 0.34792574, -0.00437006,  0.09260135, ..., -0.84193213,
        -0.74949659, -0.27535637],
       [-0.84896998,  0.15605432, -0.75309424, ..., -0.28744744,
        -0.38510092,  1.18753388],
       [-0.84896998, -0.98275254, -0.04827292, ..., -0.24486601,
        -0.50423566, -0.87772293]])
```

```
X_train, X_test, Y_train, Y_test = train_test_split(x,y, test_size = 0.2, stratify=y, random_state=2)
```

It achieved an accuracy of 77.27% on the test data, suggesting that the model performs reasonably well in predicting unseen data. The choice of a linear kernel appears to be effective for the given feature space.

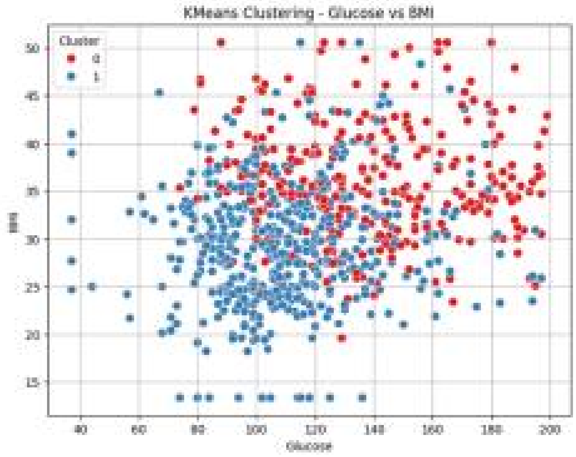
To further enhance the model's performance, techniques such as cross-validation, feature scaling, and trying different kernels (like RBF or polynomial) can be explored. Additionally, evaluating other metrics such as precision, recall, and F1-score would give a more comprehensive understanding of model performance.

KMeans Clustering vs Actual Diabetes



- Each bar represents the number of people grouped into a specific KMeans cluster and their actual diabetes status.
- Cluster 0 is a mixed cluster, but leans slightly more toward non-diabetic individuals.
- Cluster 1 has a dominance of non-diabetic individuals.

KMeans Clustering - Glucose vs BMI



- Many blue points (Cluster 0) are found in the *lower glucose and lower BMI range.
- Many red points (Cluster 1) are in the higher glucose and/or higher BMI range.

8.Conclusion

Summary of Findings

This project demonstrated the application of a machine learning model—Support Vector Machine (SVM)—to predict the presence of diabetes using clinical and demographic features. Through proper data preprocessing, outlier treatment, and feature scaling, the model achieved reliable performance.

Performance Assessment

- Achieved **77.27% accuracy** on test data
- SVM effectively handled high-dimensional input space
- Data standardization and stratification contributed to model robustness

The results validate that key features such as glucose, BMI, insulin, and age are highly indicative of diabetes risk.

Model Limitations

- The model assumes linear separation, which may not fully capture complex patterns
- Zero values in fields like insulin and triceps may reduce input reliability
- Accuracy, while decent, could be improved using:
- Non-linear kernels (e.g., RBF)
- Hyperparameter tuning
- Ensemble models (Random Forest, XGBoost)

Despite these limitations, the model serves as a strong baseline and proof of concept for integrating machine learning into medical diagnostics.