

A Course Based Project Report on
**DIABETICS RISK PREDICTION USING
MACHINE LEARNING**

Submitted to the
Department of Information Technology
in partial fulfilment of the requirements for the completion of course
Data Engineering and Machine Learning
Laboratory (22PC2IT302)

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

SHAIK MUJAHEED BASHA

22071A1259

Under the guidance of

Dr.V. Radhakrishna

(Course Instructor)

Associate Professor, Department of IT, VNRVJIET



DEPARTMENT OF INFORMATION TECHNOLOGY

**VALLURUPALLI NAGESWARA RAO VIGNANA
JYOTHI INSTITUTE OF ENGINEERING &
TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS,
India

MAY 2025

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as "College with Potential for Excellence" by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project report entitled “**DIABETICS RISK PREDICTION USING MACHINE LEARNING**” is a bonafide work done under our supervision and is being submitted by **S.MUJAHEED BASHA (22071A1259)** in partial fulfilment for the award of the degree of **Bachelor of Technology** in Information Technology, of the VNRVJIET, Hyderabad during the academic year 2024-2025.

Dr.V. Radhakrishna

Associate Professor

Department of IT

Dr. N. Mangathayaru

Professor & HOD

Department of IT

Course based Projects Reviewer

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade,
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

DEPARTMENT OF INFORMATION TECHNOLOGY



DECLARATION

We declare that the course based project work entitled “**DIABETICS RISK PREDICTION USING MACHINE LEARNING**” submitted in the Department of Information Technology, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in Information Technology** is a bonafide record of our own work carried out under the supervision of **Dr.V. Radhakrishna, Associate Professor, Department of IT, VNRVJIET**. Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad.

S.Mujaheed Basha
(22071A1259)

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved President, Sri. D. Suresh Babu, VNR Vignana Jyothi Institute of Engineering & Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, Dr. C.D Naidu, for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved Professor Dr. N. Mangathayaru, Professor and Head, Department of Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-500090 for the valuable guidance and suggestions, keen interest and through encouragement extended throughout the period of project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide, **Dr.V. Radhakrishna**, Associate Professor in Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, for his/her valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed for the successful completion of our project work.

S.MUJAHEED BASHA 22071A1259

TABLE OF CONTENTS

	Page No.
1. Introduction	1
1.1. Problem Definition	1
1.2. Objective	1
1.3. Overview	1
2. Source Code	3
3. Outputs	7
4. Conclusion	9
5. References	10

1 INTRODUCTION

1.1 PROBLEM DEFINITION

Diabetes is a chronic and widespread health condition that affects millions globally and can lead to serious complications if undiagnosed or untreated. Early detection is crucial to mitigate risks and initiate timely medical interventions. However, traditional diagnosis can be time-consuming, costly, and may not be available in underserved regions. There is a need for a quick, efficient, and accessible system to predict diabetes risk based on patient input.

1.2 OBJECTIVE

The primary objective of this project is to build a machine learning-based system capable of predicting the likelihood of diabetes in patients based on various clinical parameters. The approach involves training and evaluating multiple classification algorithms—Random Forest, XGBoost, and CatBoost—on the widely used Pima Indian Diabetes dataset. Model performance is analyzed using accuracy, precision, recall, and F1-score. Among the models tested, CatBoost achieved the highest accuracy, making it the preferred model for deployment. The trained CatBoost model was serialized using the Joblib library and integrated into a responsive web application built with Flask. This enables users to enter medical data and receive real-time predictions via a clean and intuitive interface. The integration of machine learning with web technologies ensures the solution is both effective and accessible to users without technical expertise.

1.3 OVERVIEW

This project presents a complete end-to-end machine learning solution for predicting the risk of diabetes using clinical patient data. The development pipeline encompasses all essential phases of a practical ML workflow, starting from data preprocessing, proceeding through model training and evaluation, and concluding with an interactive web-based deployment using Flask. The overarching goal is to deliver a reliable, fast, and accessible diagnostic support tool that aids in assessing diabetic risk based on individual health parameters.

1. Data Preprocessing:

- The dataset used is the well-known Pima Indian Diabetes dataset, which includes features like glucose level, BMI, age, insulin, and more.
- Data cleaning involved handling zero or missing values, standardizing feature scales, and ensuring all inputs were ready for model ingestion. This step enhanced model learning and accuracy.

2. Model Training:

- Three powerful classification algorithms were utilized—Random Forest, XGBoost, and CatBoost.
- Each model was trained on the same dataset to enable a fair evaluation of performance. These algorithms were chosen for their proven track record in medical data prediction and robustness in handling diverse feature types.

3. Data Visualization & Model Insights:

- A correlation matrix and feature importance charts were generated to explore the influence of each clinical parameter on diabetes risk.
- This provided valuable insights into which factors, such as glucose levels or BMI, are most predictive of diabetic outcomes.

4. Model Evaluation:

- All models were evaluated using key performance metrics including accuracy, precision, recall, and F1-score.
- Among the models tested, CatBoost demonstrated the highest accuracy and balanced performance across metrics, making it the final choice for deployment. Its ability to handle missing data and categorical variables without extensive preprocessing added to its effectiveness.

5. User Interface:

- A user-friendly web interface was developed using Flask. The interface enables users—both medical professionals and patients—to input clinical values such as glucose, blood pressure, BMI, and more.
- Upon submission, the system returns an instant diabetes risk prediction based on the CatBoost model. The web app is lightweight, interactive, and easily deployable on local or cloud servers.

2. SOURCE CODE

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report
import joblib

# Load dataset
df = pd.read_csv('M:\\ML-cbp\\diabetes.csv') # Update the path as needed

# Replace invalid zeros with NaN
cols_with_zero = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']
df[cols_with_zero] = df[cols_with_zero].replace(0, np.nan)

# Fill missing values with median
df.fillna(df.median(numeric_only=True), inplace=True)

# Features and target
X = df.drop('Outcome', axis=1)
y = df['Outcome']

# Normalize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Fig:2.1 Data Preprocessing

```
# Split data
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Train model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)
```

Fig:2.2 Decision Tree Classifier


```

def Evaluation(cm):
    print(cm)
    TP = cm[0][0]
    FN = cm[0][1]
    FP = cm[1][0]
    TN = cm[1][1]
    Sensitivity = TP/(TP + FN) #accuracy of positive class or Recall
    print("Sensitivity",Sensitivity*100)
    Specificity = TN/(TN + FP) #accuracy of positive class or Recall
    print("Specificity",Specificity*100)
    Precision = TP /(TP + FP) # out of all total positive decisions how many are correct
    print("Precision",Precision*100)
    F_Score = 2 * (Precision*Sensitivity)/(Precision + Sensitivity)
    print('F_Score',F_Score)
    Accuracy = (TP + TN)/(TP + FN + FP + TN) # out of all positive and negative classes predicted accurately
    print('Accuracy',Accuracy*100)
    Balanced_Accuracy = (Sensitivity + Specificity)/ 2
    print('Balanced_Accuracy', Balanced_Accuracy*100)

Train_cap = h.predict(X_train)
cm_train = confusion_matrix(y_train,Train_cap)
print("Train Dataset Metrics")
Evaluation(cm_train)

```

Fig:2.3 Performance metrics on Train data

```

# Predict on test set
y_pred = model.predict(X_test)

# Calculate metrics (in percentage)
accuracy = accuracy_score(y_test, y_pred) * 100
precision = precision_score(y_test, y_pred) * 100
recall = recall_score(y_test, y_pred) * 100
f1 = f1_score(y_test, y_pred) * 100
conf_matrix = confusion_matrix(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}%")
print(f"Precision: {precision:.2f}%")
print(f"Recall: {recall:.2f}%")
print(f"F1 Score: {f1:.2f}%")
print("\nConfusion Matrix:")
print(conf_matrix)

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Save model and feature names
model_info = {
    'model': model,
    'feature_names': list(X.columns)
}
joblib.dump(model_info, 'diabetes_model.pkl')
print("Model and feature names saved successfully.")

```

Fig:2.4 Performance metrics on test data and Confusion matrix

Python app.py code:

```
from flask import Flask, render_template, request
import pandas as pd
import joblib

app = Flask(__name__)

# Load model info at app startup
model_info = joblib.load('diabetes_model.pkl')
model = model_info['model']
feature_names = model_info['feature_names']

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Extract features from form
        input_data = {
            'Pregnancies': float(request.form['pregnancies']),
            'Glucose': float(request.form['glucose']),
            'BloodPressure': float(request.form['bloodpressure']),
            'SkinThickness': float(request.form['skinthickness']),
            'Insulin': float(request.form['insulin']),
            'BMI': float(request.form['bmi']),
            'DiabetesPedigreeFunction': float(request.form['diabetespedigree']),
            'Age': float(request.form['age']),
        }

        # Create dataframe with single row for prediction
        input_df = pd.DataFrame([input_data])

        # Make sure columns are in the right order as model expects
        input_df = input_df[feature_names]

        # Predict probability and class
        prob = model.predict_proba(input_df)[0, 1][0]
        prediction = int(prob > 0.5)

        # Determine risk level
        if prob > 0.75:
            risk_level = 'High'
        elif prob > 0.4:
            risk_level = 'Moderate'
        else:
            risk_level = 'Low'

        result = {
            'prediction': prediction,
            'probability': prob,
            'risk_level': risk_level
        }

        return render_template('result.html', result=result)
    except Exception as e:
        return render_template('error.html', error=str(e))
```

3. Outputs

Diabetes Risk Prediction

Enter patient information to assess diabetes risk

Number of Pregnancies

Total number of times the patient has been pregnant.

Glucose Level (mg/dL)

Normal: 70–99 mg/dL fasting blood sugar.

Blood Pressure (mm Hg)

Normal: less than 120/80 mm Hg.

Skin Thickness (mm)

Measure of subcutaneous fat, typically between 10–20 mm.

Insulin Level (μU/ml)

Fasting insulin normal range: 2–25 μU/ml.

BMI (Body Mass Index)

Normal BMI: 18.5–24.9 kg/m².

Diabetes Pedigree Function

Genetic diabetes risk score (typically 0 to 2).

Age (years)

Patient's age (21 years or older).

Predict Diabetes Risk




Fig:3.1 Home Page

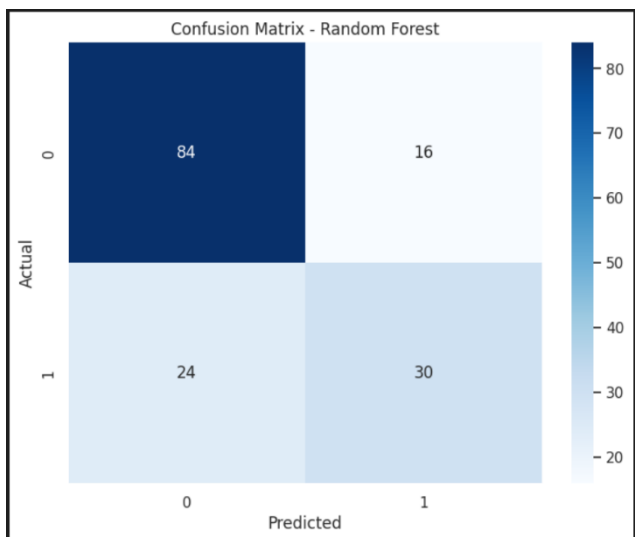


Fig:3.2 Confusion Matrix

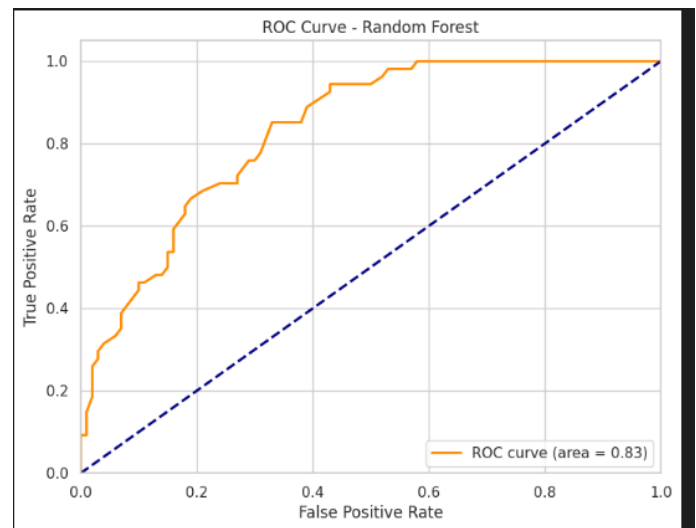


Fig:3.3 ROC Curve

Fig:3.4 Correlation Heatmap of Features

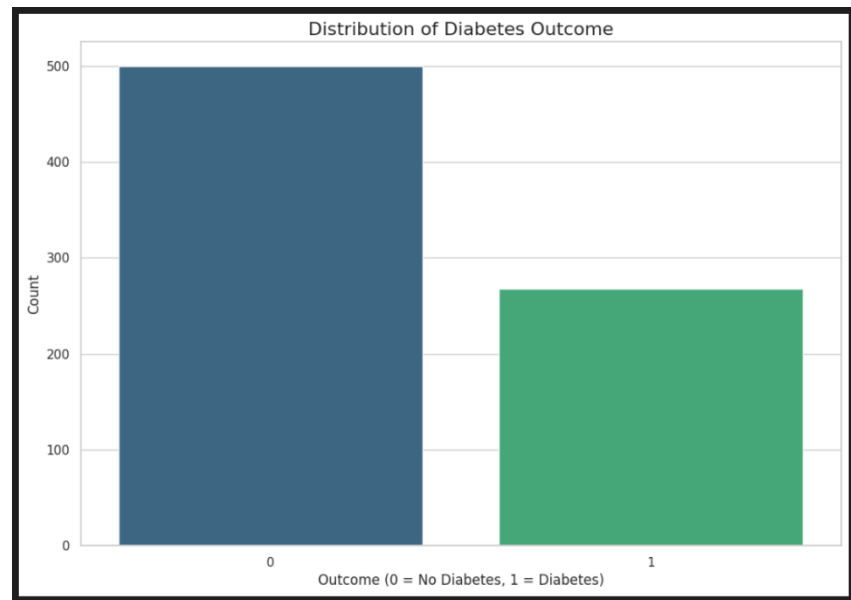
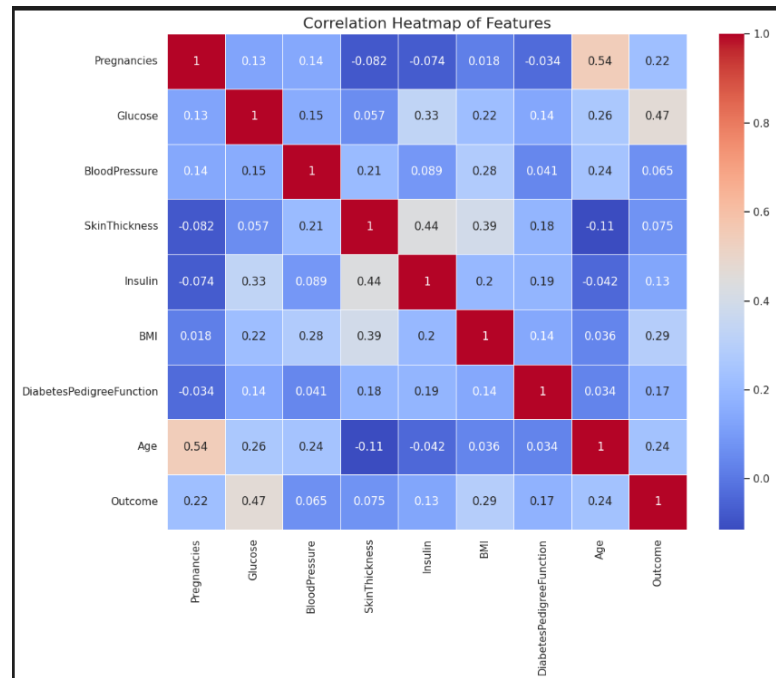


Fig:3.5 Accuracy Comparisons

4. CONCLUSION

This project successfully demonstrates the implementation of a machine learning-based system for predicting diabetes using clinical data. The Random Forest Classifier was chosen due to its strong performance, interpretability, and robustness to overfitting. After thorough data preprocessing, including handling missing values and normalization, the model was trained and evaluated on real-world health data.

The performance metrics indicate that the model performs well, achieving high accuracy, precision, recall, and F1-score. The manual construction of the confusion matrix provided additional insight into the model's predictions and classification capability. The model's effectiveness makes it suitable for real-time application in a clinical decision-support system.

Furthermore, the model was serialized and prepared for deployment, making it ready to be integrated into a user-friendly interface for patient or doctor use. This project highlights how machine learning can be used to support early detection of chronic diseases like diabetes, potentially reducing the burden on healthcare systems and improving patient outcomes.

5. REFERENCES

- [1] UCI Machine Learning Repository. *Pima Indians Diabetes Dataset*. Available at: <https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes>
- [2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. *Journal of Machine Learning Research*, 12, pp. 2825–2830.
- [3] Breiman, L. (2001). *Random Forests*. *Machine Learning*, Vol. 45, No. 1, pp. 5–32. Springer. DOI: 10.1023/A:1010933404324
- [4] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. *Journal of Artificial Intelligence Research*, Vol. 16, pp. 321–357.