AI BASED DIABETES PREDICTION SYSTEM

TEAM MEMBER:

NAME: M RUDHRESHKUMAR

REGISTER NO: 410121243045

DOCUMENT SUBMISSION

PHASE 5 ---

Project Overview:

Introduction:

Diabetes is a chronic medical condition that affects millions of people worldwide. Early detection and management of diabetes are crucial for improving the quality of life and reducing complications. This project aims to develop an AI-based Diabetes Prediction System that can predict the risk of diabetes in individuals based on their health data and provide early warnings and recommendations.

Problem Statement:

The primary objective of this project is to create a predictive model that can analyze various health-related data points and provide an accurate prediction of an individual's risk of developing diabetes. The system will consider a range of risk factors, including age, gender, family history, lifestyle, and medical history, to provide personalized predictions.

Key Features and Components:

DATA COLLECTION:

Gather a diverse dataset of health records, including patient demographics, medical history, lifestyle data, and biomarker information.

DATA PROCESSING:

Clean and preprocess the data, handling missing values and outliers, and normalizing/standardizing features.

FEATURE SELECTION:

Identify and select the most relevant features that contribute to diabetes risk prediction.

MACHINE LEARNING MODELS :

Develop and train machine learning models, such as logistic regression, decision trees, random forests, or deep learning models, using the preprocessed data.

EVALUATION METRICS:

Implement evaluation metrics (e.g., accuracy, F1 score, ROC-AUC) to assess the performance of the predictive model.

USER INTERFACE:

Create a user-friendly interface for users to input their health data and receive risk predictions.

PERSONALISED RECOMMENDATIONS:

Provide personalized recommendations based on risk levels, such as lifestyle changes, diet modifications, and exercise routines.

PRIVACY AND SECURITY:

Implement strong security measures to protect user health data and ensure compliance with data privacy regulations.

BENEFITS:

Early Detection:

The system will assist in identifying individuals at risk of diabetes at an early stage, enabling timely intervention and treatment.

Personalized Recommendations:

Users will receive tailored recommendations to reduce their diabetes risk, promoting a healthier lifestyle.

Health Monitoring:

The system can be used for continuous health monitoring and provide regular updates on diabetes risk.

Target Audience:

Healthcare professionals for clinical use.

Individuals interested in monitoring their diabetes risk.

Future Enhancements:

Integration with wearable devices for real-time health data collection.

Expansion to include additional chronic diseases for prediction.

Collaboration with healthcare institutions for data sharing and research.

The AI-Based Diabetes Prediction System aims to leverage the power of artificial intelligence and machine learning to provide accurate and personalized predictions for diabetes risk. By implementing this system, individuals can take proactive steps towards preventing or managing diabetes, ultimately improving their overall health and well-being

PRIMARY GOALS:

- Early Detection: The system aims to identify individuals at risk of diabetes at an early stage, allowing for timely intervention and lifestyle modification.

- Personalized Risk Assessment: It provides a personalized risk assessment for each individual, taking into account their specific medical history, clinical parameters, and lifestyle factors.
- Improved Healthcare: By automating the prediction process, healthcare providers can allocate resources more efficiently,offer preventive care, and better mange diabetes patients.
- Data-Driven Insights: The system can offer valuable insights by analyzing patterns and relationships within the data, potentially

leading to a better understanding of diabetes risk factors.

Conclusion:

The AI-Based Diabetes Prediction System aims to leverage the power of artificial intelligence and machine learning to provide accurate and personalized predictions for diabetes risk. By implementing this system, individuals can take proactive steps towards preventing or managing diabetes, ultimately improving their overall health and well-being.

Machine learning algorithm

Program code:

```
# Import necessary libraries

import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, classification_report


# Load the diabetes dataset from scikit-learn

from sklearn.datasets import load_diabetes

data = load_diabetes()
```

```
X = data.data
y = (data.target > 140).astype(int)  # Binary classification: 1 if diabetes, 0 if not

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the feature data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Create and train a Logistic Regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate and print the accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

# Print a classification report
report = classification_report(y_test, y_pred)
print("Classification Report:\n", report)
```

output:

```
 Accuracy: 0.72
Classification Report:
```

	precision	recall	f1-score	support
0	0.78	0.72	0.75	39
1	0.64	0.71	0.67	29
accuracy			0.72	68
macro avg	0.71	0.72	0.71	68
weighted avg	0.72	0.72	0.72	68

Conclusion:

The AI-Based Diabetes Prediction System aims to leverage the power of artificial intelligence and machine learning to provide accurate and personalized predictions for diabetes risk. By implementing this system, individuals can take proactive steps towards preventing or managing diabetes, ultimately improving their overall health and well-being.

PROGRAM CODE :

```
# Step 1: Import necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings("ignore", category=UserWarning)

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
# Step 2: Load the dataset
df = pd.read_csv("/kaggle/input/diabetes-data-set/diabetes.csv")


# Step 3: Data Cleaning


# Check for Missing Values
missing_values = df.isnull().sum()
print("Missing Values:")
print(missing_values)


# Handle missing values (if any)
# For example, fill missing values with the mean of the column
mean_fill = df.mean()
df.fillna(mean_fill, inplace=True)


# Check for Duplicate Rows
duplicate_rows = df[df.duplicated()]
print("\nDuplicate Rows:")
print(duplicate_rows)


# Handle duplicate rows (if any)
# For example, drop duplicate rows
df.drop_duplicates(inplace=True)


# Step 4: Data Analysis


# Summary Statistics
summary_stats = df.describe()
print("\nSummary Statistics:")
print(summary_stats)


# Class Distribution (for binary classification problems)
```

```
class_distribution = df['Outcome'].value_counts()
print("\nClass Distribution:")
print(class_distribution)

# Step 5: Data Visualization
sns.pairplot(df, hue='Outcome')
plt.show()

# Step 6: Support Vector Machine (SVM) Modeling

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

# Split the dataset into a training and testing set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Initialize and train the SVM model
model = SVC(kernel='linear', random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

```
# Classification report and confusion matrix
print(classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d')
plt.show()
```

Missing Values:

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Duplicate Rows:

Empty DataFrame

Columns: [Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome]

Index: []

Summary Statistics:

```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
```

75%	6.000000	140.250000	80.000000	32.000000	127.250000
max	17.000000	199.000000	122.000000	99.000000	846.000000

	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	31.992578	0.471876	33.240885	0.348958
std	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.078000	21.000000	0.000000
25%	27.300000	0.243750	24.000000	0.000000
50%	32.000000	0.372500	29.000000	0.000000
75%	36.600000	0.626250	41.000000	1.000000
max	67.100000	2.420000	81.000000	1.000000

Class Distribution:

Outcome

0 500

1 268

Name: count, dtype: int64

Accuracy: 0.76

	precision	recall	f1-score	support
0	0.81	0.82	0.81	99
1	0.67	0.65	0.66	55
accuracy			0.76	154
macro avg	0.74	0.74	0.74	154
weighted avg	0.76	0.76	0.76	154