

# TITLE: AI BASED DIABETES PREDICTION SYSTEM

## PROGRAM:

```
Import random
```

```
# Define sigmoid function for logistic regression
```

```
Def sigmoid(x):
```

```
    Return 1 / (1 + (2.71828 ** -x))
```

```
# Initialize model parameters (theta values)
```

```
Theta = [random.uniform(-1, 1) for _ in range(8)] # 8 features
```

```
# Define a function to make predictions
```

```
Def predict(features):
```

```
    Z = sum([theta[i] * features[i] for i in range(8)]) # Calculate the weighted sum
```

```
    Return sigmoid(z)
```

```
# Define a function to train the model
```

```
Def train_model(data, labels, learning_rate, num_epochs):
```

```
    For epoch in range(num_epochs):
```

```
        For i in range(len(data)):
```

```
            X = data[i]
```

```
            Y = labels[i]
```

```
            Prediction = predict(x)
```

```
            Error = y - prediction
```

```
            For j in range(len(theta)):
```

```
                Theta[j] += learning_rate * error * x[j]
```

```
# Example data (you should replace this with your dataset)
```

```
Data = [  
    [0.2, 0.5, 0.1, 0.8, 0.6, 0.3, 0.4, 0.7],  
    # Add more data points…  
]
```

```
Labels = [1, 0, 1, 0] # Replace with actual labels
```

```
Learning_rate = 0.01  
Num_epochs = 100
```

```
# Train the model  
Train_model(data, labels, learning_rate, num_epochs)
```

```
# Make predictions for new data  
New_data_point = [0.3, 0.4, 0.2, 0.7, 0.5, 0.1, 0.6, 0.8]  
Prediction = predict(new_data_point)
```

```
# Print the prediction  
Print( “Prediction: ” , prediction)
```