```
import numpy as np
import pandas as pd

class Perceptron:

    def __init__(self, learning_rate, epochs):
        self.weights = None
        self.bias = None
        self.learning_rate = learning_rate
        self.epochs = epochs

    def activation(self, z):
        return np.heaviside(z, 0) # haviside(z) heaviside ->
activation

    # Perceptron training
    def fit(self, X, y):
        n_features = X.shape[1]

        # Initializing weights and bias
        self.weights = np.zeros((n_features))
        self.bias = 0

        # Iterating until the number of epochs
        for epoch in range(self.epochs):

            # Traversing through the entire training set
            for i in range(len(X)):
                z = np.dot(X, self.weights) + self.bias # Finding the
dot product and adding the bias
                y_pred = self.activation(z) # Passing through an
activation function

                #Updating weights and bias
                self.weights = self.weights + self.learning_rate *
(y[i] - y_pred[i]) * X[i]
                self.bias = self.bias + self.learning_rate * (y[i] -
y_pred[i])

        return self.weights, self.bias

    def predict(self, X):
        z = np.dot(X, self.weights) + self.bias
        return self.activation(z)


diabetes_df = pd.read_csv(r"C:\Users\student\Desktop\Aks\
diabetes.csv")

diabetes_df.head()
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin
BMI  \
0            6      148             72             35        0  33.6

1            1       85             66             29        0  26.6

2            8      183             64              0        0  23.3

3            1       89             66             23       94  28.1

4            0      137             40             35      168  43.1


   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

```
from sklearn.model_selection import train_test_split

X =
diabetes_df[['Pregnancies','Glucose','BloodPressure','SkinThickness','
Insulin','BMI','DiabetesPedigreeFunction','Age']].values
y = diabetes_df['Outcome'].values

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.5, random_state=42)

perceptron = Perceptron(0.001, 100)
perceptron.fit(X_train, y_train)
pred = perceptron.predict(X_test)

from sklearn.metrics import accuracy_score

print("Accuracy Score : ",(accuracy_score(pred, y_test))*100,"%")
```

```
Accuracy Score :  66.92708333333334 %
```

```
from sklearn.metrics import classification_report

report = classification_report(pred, y_test, digits=2)
print(report)
```

```
              precision    recall  f1-score   support

         0.0       0.93      0.69      0.79       343
         1.0       0.17      0.54      0.26        41

    accuracy                           0.67       384
   macro avg       0.55      0.61      0.52       384
```

```
weighted avg       0.84      0.67      0.73       384
```