**Calculate BMI - v3**

**By Begho L Joseph**

import pandas as pd  
import numpy as np  
import csv  
  
class hlthCalculator():  
  
 def assign\_health\_risk(df):  
 if df is not None:  
 if "BMI" in df.columns: ## check if this column exists in the dataFrame  
 df["Health risk"] = np.where( ## Create derived column that will hold the Health risk indicator  
 (df["BMI"] >= 18.4), 'Malnutrition risk',  
 np.where(  
 (df["BMI"] >= 18.5) & (df["BMI"] <= 24.9), 'Low risk',  
 np.where(  
 (df["BMI"] >= 25) & (df["BMI"] <= 29.9), 'Enhanced risk',  
 np.where(  
 (df["BMI"] >= 30) & (df["BMI"] <= 34.9), 'Medium risk',  
 np.where(  
 (df["BMI"] >= 35) & (df["BMI"] <= 39.9), 'High risk',  
 np.where((df["BMI"] >= 40), 'Very high risk', 'not in threshold'))))))  
 df["Count\_of\_overwight"] = len(  
 df[  
 (df["BMI"] >= 25) &  
 (df["BMI"] <= 29.9)  
 ]  
 )  
 return df  
 else:  
 return "BMI not processed yet"  
 else:  
 return ''  
  
 def calculate\_BMI(data):  
 if len(data) >0: ##Validating the input list of dictionary objects  
 df = pd.DataFrame(data)  
 df["BMI"] = round((df["HeightCm"] / (df["WeightKg"] \* df["WeightKg"])) \* 100, 2)  
 return df  
 else:  
 print("Health risk data not supplied. BMI could not be calculated. Contact the data supplier. Thanks.")  
  
 def return\_data():  
 data = [  
 {"Gender": "Male", "HeightCm": 171, "WeightKg": 96 },  
 { "Gender": "Male", "HeightCm": 161, "WeightKg": 85 },  
 { "Gender": "Male", "HeightCm": 180, "WeightKg": 77 },  
 { "Gender": "Female", "HeightCm": 166, "WeightKg": 62},  
 {"Gender": "Female", "HeightCm": 150, "WeightKg": 70},  
 {"Gender": "Female", "HeightCm": 167, "WeightKg": 82}  
 ]  
 return data  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 ## Get BMI value for each person  
 Bmi\_df = hlthCalculator.calculate\_BMI(hlthCalculator.return\_data())  
 ##Check health risk status  
 health\_risk\_df = hlthCalculator.assign\_health\_risk(Bmi\_df)  
 print(health\_risk\_df) ###Only for debuging  
  
 # save result set to local marchine  
  
 file\_name = 'HeartRisk\_processed.csv'  
 #path = 'C:\\Users\\Begho\\CalculatorProject\\'+file\_name  
 #health\_risk\_df.to\_csv(path)  
 print('')  
 print('file processed')  
  
 ##Clean up  
 del Bmi\_df  
 del health\_risk\_df

**Dataset result output**

A screenshot of a computer

Description automatically generated

Test code/Script

# Test calculator application.py  
from CalculateBMI import hlthCalculator as c  
  
class TestCalculatorAppInstance():  
 bmi\_df = c.calculate\_BMI(c.return\_data())  
 health\_risk\_df = c.assign\_health\_risk(bmi\_df)  
  
 def test\_weight\_for\_null\_values(self):  
 health\_risk\_df = self.health\_risk\_df  
 counter = health\_risk\_df['HeightCm'].isna().sum()  
 assert counter == 0  
  
 def test\_height\_for\_null\_values(self):  
 health\_risk\_df = self.health\_risk\_df  
 counter = health\_risk\_df['WeightKg'].isna().sum()  
 assert counter == 0  
  
 def test\_BMI\_for\_null\_values(self):  
 health\_risk\_df = self.health\_risk\_df  
 counter = health\_risk\_df['BMI'].isna().sum()  
 assert counter == 0  
  
 def test\_Count\_of\_overwight(self):  
 health\_risk\_df = self.health\_risk\_df  
 counter = len(health\_risk\_df[(health\_risk\_df["BMI"] >= 25) &(health\_risk\_df["BMI"] <= 29.9)])  
 assert counter == 0  
  
 def test\_source\_and\_dest\_volume(self):  
 s\_count = len(self.bmi\_df)  
 target\_count = len(self.health\_risk\_df)  
 assert s\_count == target\_count

Test Output:

A screenshot of a computer

Description automatically generated

**Hearth Risk calculation by Height by (weight - square)**

This above peace of code calculates BMI based on individual Height and weight Square.

1. There is one class that houses the functions
2. The are two functions created to perform the BMI measures and Health risk indicators and the counting of persons **overweight risks**
3. Three derived columns created according to requirements

**New derived columns created:**

1. BMI measure -> round((height / (weightCm \* weightCm ) ) \* 100, 2)
2. Count of overweight persons
3. Person hearth risk, based on BMI measure ranges

**Python code components**

**Python Modules**

1. Pandas
2. Numpy

**Python classes**

1. hlthCalculator

**Pyhton Functions**

1. calculate\_BMI(data) – health risk data must be supplied to this function as parameter
2. assign\_health\_risk(df) – this function assigns health risk indicators to each person. It counts the Overweight persons as well

**Data used for the project**

1. Supplied Test data

END