```
from google.colab import drive
drive.mount('/gdrive')
```

```
Mounted at /gdrive
```

```
ls
```

```
ls: cannot access 'drive': Transport endpoint is not connected
drive/  sample_data/
```

```
fname="/gdrive/MyDrive/diabetes_two (1).csv"
```

data preprocessing_1

import all the necessary libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import math
```

Passing the dataset into the pandas DataFrame and show first 100 rows with head() command and show last 100 rows with tail command

```
df=pd.read_csv(fname)
#df.head(100)
#df.tail(100)
#df.head()
df
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	mus stiffr
0	40.0	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	
1	58.0	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	
2	NaN	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	
3	45.0	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	
4	600.0	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
515	39.0	Female	Yes	Yes	Yes	No	Yes	No	No	Yes	No	Yes	Yes	
516	48.0	Female	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	
517	58.0	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	
518	32.0	Female	No	No	No	Yes	No	No	Yes	Yes	No	Yes	No	
519	42.0	Male	No	No	No	No	No	No	No	No	No	No	No	

Next steps: View recommended plots

```
df[299:320] # for custom range 299th to 319th
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	musc stiffne
299	43.0	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	\
300	35.0	Female	Yes	Yes	No	Yes	No	No	Yes	No	No	No	No	
301	47.0	Female	No	No	Yes	Yes	Yes	No	No	No	No	No	No	\
302	61.0	Female	Yes	No	No	No	Yes	No	No	No	Yes	No	No	
303	58.0	Female	Yes	No	Yes	No	Yes	No	No	No	Yes	No	No	\
304	69.0	Female	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	No	\
305	40.0	Male	No	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes	\
306	28.0	Male	No	No	Yes	No	No	No	No	No	No	No	No	
307	37.0	Male	No	No	No	No	No	No	No	No	No	No	No	
308	34.0	Male	No	No	No	No	No	No	No	No	No	No	No	
309	30.0	Male	No	No	No	No	No	No	No	No	No	No	No	
310	67.0	Male	Yes	No	No	Yes	Yes	No	Yes	Yes	Yes	Yes	No	\
311	60.0	Male	No	No	No	Yes	No	No	No	No	No	No	Yes	\
312	58.0	Male	No	No	No	No	Yes	No	No	Yes	No	Yes	No	
313	54.0	Male	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	
314	43.0	Male	No	No	Yes	No	No	Yes	No	No	No	Yes	No	
315	33.0	Female	No	No	No	No	No	No	No	No	No	No	No	
316	55.0	Female	No	No	No	Yes	No	Yes	No	Yes	No	Yes	Yes	
317	36.0	Female	No	No	No	Yes	No	No	No	No	Yes	No	No	
318	28.0	Female	No	No	No	No	Yes	No	No	No	No	No	Yes	

```
df["Gender"] #to see any particular column
```

```
0        Male
1        Male
2        Male
3        Male
4        Male
        ...
515    Female
516    Female
517    Female
518    Female
519      Male
Name: Gender, Length: 520, dtype: object
```

```
df["Gender"].head()
df.loc[df.Gender=='Male']
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	mus stiffr
0	40.0	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	
1	58.0	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	
2	NaN	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	
3	45.0	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	
4	600.0	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
509	54.0	Male	No	No	No	No	No	No	No	No	No	No	No	
510	67.0	Male	No	No	No	Yes	No	No	No	Yes	No	Yes	No	
511	66.0	Male	No	No	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes	
512	43.0	Male	No	No	No	No	No	No	No	No	No	No	No	
519	42.0	Male	No	No	No	No	No	No	No	No	No	No	No	

```
df.loc[(df.Gender=="Male")&(df.Age>25)] #all above 25 years and male shows
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	mus stiffr
0	40.0	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	
1	58.0	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	
3	45.0	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	
4	600.0	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
5	55.0	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
509	54.0	Male	No	No	No	No	No	No	No	No	No	No	No	
510	67.0	Male	No	No	No	Yes	No	No	No	Yes	No	Yes	No	
511	66.0	Male	No	No	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes	
512	43.0	Male	No	No	No	No	No	No	No	No	No	No	No	
519	42.0	Male	No	No	No	No	No	No	No	No	No	No	No	

```
df.iloc[[1,100,200,300,400,500]] #show the numbered index
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	musc stiffne
1	58.0	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	
100	48.0	Female	Yes	Yes	Yes	No	Yes	Yes	No	No	Yes	Yes	No	Y
200	40.0	Male	No	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Y
300	35.0	Female	Yes	Yes	No	Yes	No	No	Yes	No	No	No	No	
400	44.0	Male	Yes	No	Yes	Yes	No	Yes	No	Yes	No	Yes	No	

```
df.loc[df.Age.isnull()] #show Age column if it has any NaN
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness

```
df.Gender.iloc[[1,10,100]] #show 1st, 10th and 100th row of Gender column
df.Gender.iloc[1:100] #show 1st to 99th rows of gender columns
```

```
1       Male
2       Male
3       Male
4       Male
5       Male
       ...
95    Female
96    Female
97    Female
98    Female
99    Female
Name: Gender, Length: 99, dtype: object
```

```
df.columns #Give the columns name
```

```
Index(['Age', 'Gender', 'Polyuria', 'Polydipsia', 'sudden weight loss',
       'weakness', 'Polyphagia', 'Genital thrush', 'visual blurring',
       'Itching', 'Irritability', 'delayed healing', 'partial paresis',
       'muscle stiffness', 'Alopecia', 'Obesity', 'class'],
      dtype='object')
```

```
len(df)
print(len(df.axes[0]),'X',len(df.axes[1])) #show how many rows and columns present
```

```
520 X 17
```

```
df[1:200].loc[(df.Gender=='Male')] #give the 1st to 199th values of Gender column
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	mus stiffr
1	58.0	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	
2	NaN	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	
3	45.0	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	
4	600.0	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
5	55.0	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
192	64.0	Male	No	Yes	No	No	No	No	No	No	Yes	Yes	No	
193	36.0	Male	Yes	No	No	Yes	No	Yes	Yes	Yes	No	Yes	No	
194	43.0	Male	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	
195	31.0	Male	Yes	No	No	No	Yes	No	Yes	No	No	No	Yes	
196	66.0	Male	No	No	No	No	Yes	No	Yes	No	No	No	Yes	

```
df.describe() # describe the data set and works only for numerical datas
df.std()
```

```
<ipython-input-16-6c672d69e3a9>:2: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a future version,
  df.std()
Age    27.104027
dtype: float64
```

Handeling Missing Values

```
df.isnull() #gives in boolean table if null present return True else False
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	mus stiffn
0	False	False	False	False	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	False	False	False	False	F
2	True	False	False	False	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
515	False	False	False	False	False	False	False	False	False	False	False	False	False	F
516	False	False	False	False	False	False	False	False	False	False	False	False	False	F
517	False	False	False	False	False	False	False	False	False	False	False	False	False	F
518	False	False	False	False	False	False	False	False	False	False	False	False	False	F
519	False	False	False	False	False	False	False	False	False	False	False	False	False	F

```
df.isnull().sum() #Returns which colum has how many colums containing null value
```

```
Age                 1
Gender              1
Polyuria            0
Polydipsia          0
sudden weight loss  0
weakness            1
Polyphagia          0
Genital thrush      0
visual blurring     0
Itching             0
Irritability        0
delayed healing     0
partial paresis     0
muscle stiffness    0
Alopecia            0
Obesity             0
class               0
dtype: int64
```

```
df.info()  #gives general information of the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 520 entries, 0 to 519
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Age                 519 non-null    float64
 1   Gender              519 non-null    object
 2   Polyuria            520 non-null    object
 3   Polydipsia          520 non-null    object
 4   sudden weight loss  520 non-null    object
 5   weakness            519 non-null    object
 6   Polyphagia          520 non-null    object
 7   Genital thrush      520 non-null    object
 8   visual blurring     520 non-null    object
 9   Itching             520 non-null    object
 10  Irritability        520 non-null    object
 11  delayed healing     520 non-null    object
 12  partial paresis     520 non-null    object
 13  muscle stiffness    520 non-null    object
 14  Alopecia            520 non-null    object
 15  Obesity             520 non-null    object
 16  class               520 non-null    object
dtypes: float64(1), object(16)
memory usage: 69.2+ KB
```

```
df1=df #keeping copies of the dataset
```

Droping the intire row tha has a missing value

```
df1.isnull().sum() #Returns which colum has how many colums containing null value
```

```
Age                 1
Gender              1
Polyuria            0
Polydipsia          0
sudden weight loss  0
weakness            1
Polyphagia          0
Genital thrush      0
visual blurring     0
Itching             0
Irritability        0
delayed healing     0
partial paresis     0
muscle stiffness    0
Alopecia            0
Obesity             0
class               0
dtype: int64
```

```
df1.dropna(subset=['Age'],inplace=True) #removing any row
df1.dropna(subset=['Gender'],inplace=True)
df1.dropna(subset=['weakness'],inplace=True)
```

```
df1.isnull().sum()
```

```
Age                 0
Gender              0
Polyuria            0
Polydipsia          0
sudden weight loss  0
weakness            0
Polyphagia          0
Genital thrush      0
visual blurring     0
Itching             0
Irritability        0
delayed healing     0
partial paresis     0
muscle stiffness    0
Alopecia            0
Obesity             0
class               0
dtype: int64
```

```
df2=df
df2.columns
df2.drop('Polyuria', axis=1,inplace=True) #removing the column altogather
df2.head()
```

	Age	Gender	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopeci
0	40.0	Male	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Ye
1	58.0	Male	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Ye
3	45.0	Male	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	N
4	600.0	Male	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Ye

Next steps: View recommended plots

```
df2.drop(['delayed healing','Alopecia','muscle stiffness','Irritability'],axis=1,inplace=True)
df2
```

	Age	Gender	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	partial paresis	Obesity	class
0	40.0	Male	Yes	No	Yes	No	No	No	Yes	No	Yes	Positive
1	58.0	Male	No	No	Yes	No	No	Yes	No	Yes	No	Positive
3	45.0	Male	No	Yes	Yes	Yes	Yes	No	Yes	No	No	Positive
4	600.0	Male	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Positive
5	55.0	Male	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	Positive
...	...	...	...	...	...	...	...	...	...	...	...	...
515	39.0	Female	Yes	Yes	No	Yes	No	No	Yes	Yes	No	Positive
516	48.0	Female	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	Positive
517	58.0	Female	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Positive
518	32.0	Female	No	No	Yes	No	No	Yes	Yes	No	No	Negative
519	42.0	Male	No	No	No	No	No	No	No	No	No	Negative

517 rows × 12 columns

Next steps: View recommended plots

```
df3=df
df3['Age']=df3['Age'].fillna(df3['Age'].mean()) #fill nan value with mean of the column
#df3.iloc[[2]]
df3.Age
```

```
0       40.0
1       58.0
3       45.0
4      600.0
5       55.0
       ...
515     39.0
516     48.0
517     58.0
518     32.0
519     42.0
Name: Age, Length: 517, dtype: float64
```

```
df3['Gender']=df3['Gender'].fillna(df3['Gender'].median)
print(df3['Gender'].isnull())
```

```
0      False
1      False
3      False
4      False
5      False
       ...
515    False
516    False
517    False
518    False
519    False
Name: Gender, Length: 517, dtype: bool
```

```
df4=df3['weakness'].fillna(method='ffill', inplace=True) #fill the nan value with previous row value
```

```
df.isnull().sum()
```

```
Age                   0
Gender                0
Polydipsia            0
sudden weight loss    0
weakness              0
Polyphagia            0
Genital thrush        0
visual blurring       0
Itching               0
partial paresis       0
Obesity               0
class                 0
dtype: int64
```

```
import pandas as pd
DF=pd.read_csv(fname)
DF.dropna(how="any", inplace=True)
DF.isnull().sum()
```

```
Age                  0
Gender               0
Polyuria             0
Polydipsia           0
sudden weight loss   0
weakness             0
Polyphagia           0
Genital thrush       0
visual blurring      0
Itching              0
Irritability         0
delayed healing      0
partial paresis      0
muscle stiffness     0
Alopecia             0
Obesity              0
class                0
dtype: int64
```

```
A=DF['Gender'].mode() #find the most frequent value
A
```

```
0    Male
Name: Gender, dtype: object
```

```
DF1=DF
DF1
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	mus stiffr
0	40.0	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	
1	58.0	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	
3	45.0	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	
4	600.0	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
5	55.0	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
515	39.0	Female	Yes	Yes	Yes	No	Yes	No	No	Yes	No	Yes	Yes	
516	48.0	Female	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	
517	58.0	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	
518	32.0	Female	No	No	No	Yes	No	No	Yes	Yes	No	Yes	No	
519	42.0	Male	No	No	No	No	No	No	No	No	No	No	No	

Next steps: View recommended plots

```
DF1['Gender'].replace ([np.nan],[A],inplace=True)
DF1
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	mus stiffr
0	40.0	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	
1	58.0	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	
3	45.0	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	
4	600.0	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
5	55.0	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
515	39.0	Female	Yes	Yes	Yes	No	Yes	No	No	Yes	No	Yes	Yes	
516	48.0	Female	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	
517	58.0	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	
518	32.0	Female	No	No	No	Yes	No	No	Yes	Yes	No	Yes	No	
519	42.0	Male	No	No	No	No	No	No	No	No	No	No	No	

Next steps: View recommended plots

```
DF1.isnull().sum()
```

```
Age                 0
Gender              0
Polyuria            0
Polydipsia          0
sudden weight loss  0
weakness            0
Polyphagia          0
Genital thrush      0
visual blurring     0
Itching             0
Irritability        0
delayed healing     0
partial paresis     0
muscle stiffness    0
Alopecia            0
Obesity             0
class               0
dtype: int64
```

```
DF.isnull().sum()
```

```
Age                 0
Gender              0
Polyuria            0
Polydipsia          0
sudden weight loss  0
weakness            0
Polyphagia          0
Genital thrush      0
visual blurring     0
Itching             0
Irritability        0
delayed healing     0
partial paresis     0
muscle stiffness    0
Alopecia            0
Obesity             0
class               0
dtype: int64
```

```
print(DF1['Gender'].replace (['Male'],['joe']))
```

```
0        joe
1        joe
3        joe
4        joe
5        joe
        ...
515    Female
516    Female
517    Female
```

```
518    Female
519       joe
Name: Gender, Length: 517, dtype: object
```

Handaling categorical value or encoding

```
!pip install -U scikit-learn
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
```

```
DF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 517 entries, 0 to 519
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Age                 517 non-null    float64
 1   Gender              517 non-null    object
 2   Polyuria            517 non-null    object
 3   Polydipsia          517 non-null    object
 4   sudden weight loss  517 non-null    object
 5   weakness            517 non-null    object
 6   Polyphagia          517 non-null    object
 7   Genital thrush      517 non-null    object
 8   visual blurring     517 non-null    object
 9   Itching             517 non-null    object
 10  Irritability        517 non-null    object
 11  delayed healing     517 non-null    object
 12  partial paresis     517 non-null    object
 13  muscle stiffness    517 non-null    object
 14  Alopecia            517 non-null    object
 15  Obesity             517 non-null    object
 16  class               517 non-null    object
dtypes: float64(1), object(16)
memory usage: 72.7+ KB
```

```
#its importanct to convert object datatypes to numerical values for device understanding
DF['Gender']=le.fit_transform(DF['Gender'])
DF['Polyuria']=le.fit_transform(DF['Polyuria'])
DF['Polydipsia']=le.fit_transform(DF['Polydipsia'])
DF['sudden weight loss']=le.fit_transform(DF['sudden weight loss'])
DF['weakness']=le.fit_transform(DF['weakness'])
DF['Polyphagia']=le.fit_transform(DF['Polyphagia'])
DF['Genital thrush']=le.fit_transform(DF['Genital thrush'])
DF['visual blurring']=le.fit_transform(DF['visual blurring'])
DF['Itching']=le.fit_transform(DF['Itching'])
DF['Irritability']=le.fit_transform(DF['Irritability'])
DF['delayed healing']=le.fit_transform(DF['delayed healing'])
DF['partial paresis']=le.fit_transform(DF['partial paresis'])
DF['muscle stiffness']=le.fit_transform(DF['muscle stiffness'])
DF['Alopecia']=le.fit_transform(DF['Alopecia'])
DF['Obesity']=le.fit_transform(DF['Obesity'])
DF['class']=le.fit_transform(DF['class'])
```

```
DF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 517 entries, 0 to 519
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Age                 517 non-null    float64
 1   Gender              517 non-null    int64
 2   Polyuria            517 non-null    int64
 3   Polydipsia          517 non-null    int64
 4   sudden weight loss  517 non-null    int64
 5   weakness            517 non-null    int64
 6   Polyphagia          517 non-null    int64
 7   Genital thrush      517 non-null    int64
```

```
 8   visual blurring     517 non-null    int64
 9   Itching             517 non-null    int64
 10  Irritability        517 non-null    int64
 11  delayed healing     517 non-null    int64
 12  partial paresis     517 non-null    int64
 13  muscle stiffness    517 non-null    int64
 14  Alopecia            517 non-null    int64
 15  Obesity             517 non-null    int64
 16  class               517 non-null    int64
dtypes: float64(1), int64(16)
memory usage: 72.7 KB
```

```
DF
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	mus stiffr
0	40.0	1	0	1	0	1	0	0	0	1	0	1	0	
1	58.0	1	0	0	0	1	0	0	1	0	0	0	1	
3	45.0	1	0	0	1	1	1	1	0	1	0	1	0	
4	600.0	1	1	1	1	1	1	0	1	1	1	1	1	
5	55.0	1	1	1	0	1	1	0	1	1	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
515	39.0	0	1	1	1	0	1	0	0	1	0	1	1	
516	48.0	0	1	1	1	1	1	0	0	1	1	1	1	
517	58.0	0	1	1	1	1	1	0	1	0	0	0	1	
518	32.0	0	0	0	0	1	0	0	1	1	0	1	0	
519	42.0	1	0	0	0	0	0	0	0	0	0	0	0	

Next steps: View recommended plots

```
DF_new=pd.read_csv(fname)
```

Imbalanced Dataset issue

```
DF['class'].unique()
```

```
array([1, 0])
```

```
DF['class'].nunique()
```

```
2
```

```
Value_counts= DF.groupby('class').size().reset_index(name='count') #find the numbers of unique value present
Value_counts
```

	class	count
0	0	200
1	1	317

Next steps: View recommended plots

```
#to solve Imbalanced dataset issue downsampling can be used . downsampling is basically deleting major class samples to match minor class sa
from sklearn.utils import resample
majority_class= DF[DF['class']==1]
minority_class= DF[DF['class']==0]

n_samples = len(minority_class)
majority_downsampled=resample(majority_class,replace=False,n_samples=n_samples, random_state=42)

balanced_df = pd.concat([minority_class,majority_downsampled])
```

```
Value_counts= balanced_df.groupby('class').size().reset_index(name='count')
Value_counts
```

	class	count
0	0	200
1	1	200

Next steps: View recommended plots

```
!pip uninstall scikit-learn --yes
```

```
!pip uninstall imblearn --yes
```

```
!pip install scikit-learn==1.2.2
```

```
!pip install imblearn
```

```
DF.head()
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscl stiffnes
0	40.0	1	0	1	0	1	0	0	0	1	0	1	0	
1	58.0	1	0	0	0	1	0	0	1	0	0	0	1	
3	45.0	1	0	0	1	1	1	1	0	1	0	1	0	
4	600.0	1	1	1	1	1	1	0	1	1	1	1	1	

Next steps: View recommended plots

```
#to solve Imbalanced dataset issue oversampling can be used . oversampling is basically increasing number of classes
from imblearn.over_sampling import SMOTE

x=DF.drop('class', axis=1)
y=DF['class']

smote=SMOTE(random_state=42)
X_resampled,Y_resampled = smote.fit_resample(x,y)

oversampled_df=pd.DataFrame(X_resampled,columns=x.columns)
oversampled_df['class'] = Y_resampled
```

```
Value_counts= oversampled_df.groupby('class').size().reset_index(name='count')
Value_counts
```

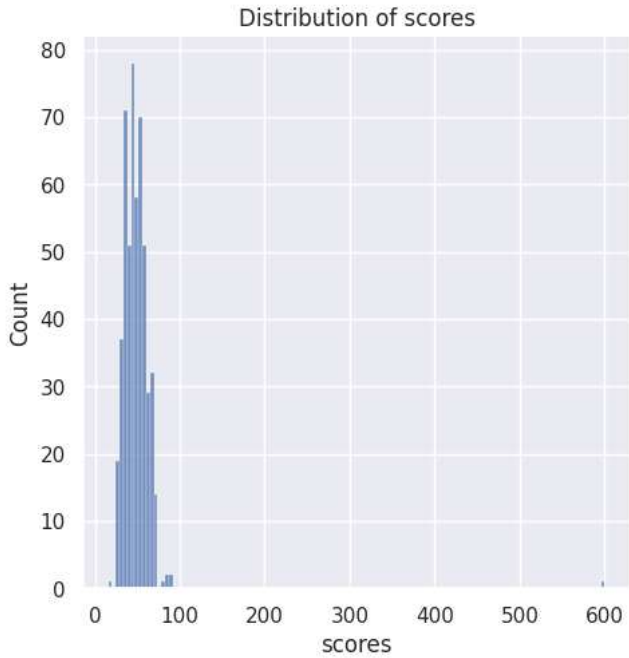
	class	count
0	0	317
1	1	317

Next steps: View recommended plots

Outlier finding and removing

```
sns.set_theme(color_codes="red")
sns.displot(data=DF['Age']).set(title="Distribution of scores", xlabel="scores")
```

```
<seaborn.axisgrid.FacetGrid at 0x7ae203ef62c0>
```




```
DF.describe()
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000
mean	49.112186	0.628627	0.493230	0.448743	0.417795	0.586074	0.456480	0.220503	0.450677	0.485493	0.239845
std	27.151937	0.483640	0.500438	0.497847	0.493674	0.493013	0.498585	0.414987	0.498043	0.500274	0.427402
min	16.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	39.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	48.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	57.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	0.000000
max	600.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

```
#find the z-score of x = (x-mean(x))/std(x)
z_score= (DF['Age']-DF['Age'].mean())/DF['Age'].std()
```

```
#z_score needs to be present in the range of -3 to 3
for i in z_score:
  if i<-3:
    print("outlier",i)
  elif i>3:
    print("outlier",i)
  else:
    continue
```

```
outlier 20.289079482050578
```

```
#find where outlier is present
index = z_score.index[z_score == 20.289079482050578][0]
index
```

```
4
```

```
DF['Age'][4]
```

```
600.0
```

```
DF['Age'][4]=DF['Age'].mean()
```

```
<ipython-input-79-d807d8be7b1c>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  DF['Age'][4]=DF['Age'].mean()
```

```
DF['Age'][4]
```

```
49.112185686653774
```

```
DF.head()
```

Handling Duplicate Samples

```
 #finding duplicates
duplicates=DF[DF.duplicated(keep='first')]
duplicates
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	musc stiffne
84	35.0	0	1	1	1	1	1	0	1	1	0	1	1	
159	38.0	0	1	1	1	1	1	0	1	1	1	1	1	
160	28.0	0	0	0	0	0	0	0	1	0	0	0	1	
161	68.0	0	1	1	0	1	1	0	1	1	0	1	1	
162	35.0	0	0	0	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
496	53.0	1	0	0	0	1	0	0	1	1	0	1	0	
497	47.0	1	0	0	0	0	0	0	0	0	1	0	1	
498	68.0	0	1	1	0	1	1	0	1	1	0	1	1	
499	64.0	1	0	0	0	1	1	0	1	1	1	1	0	
504	38.0	1	0	0	0	0	0	0	0	0	0	0	0	

Next steps: View recommended plots

```
DFwithnoduplicate= DF.drop_duplicates()
```

```
duplicates_1= DFwithnoduplicate[DFwithnoduplicate.duplicated(keep='first')]
duplicates_1
```

	Age	Gender	Polyuria	Polydipsia	sudden weight	weakness	Polyphagia	Genital	visual	Itching	Irritability	delayed	partial	muscle

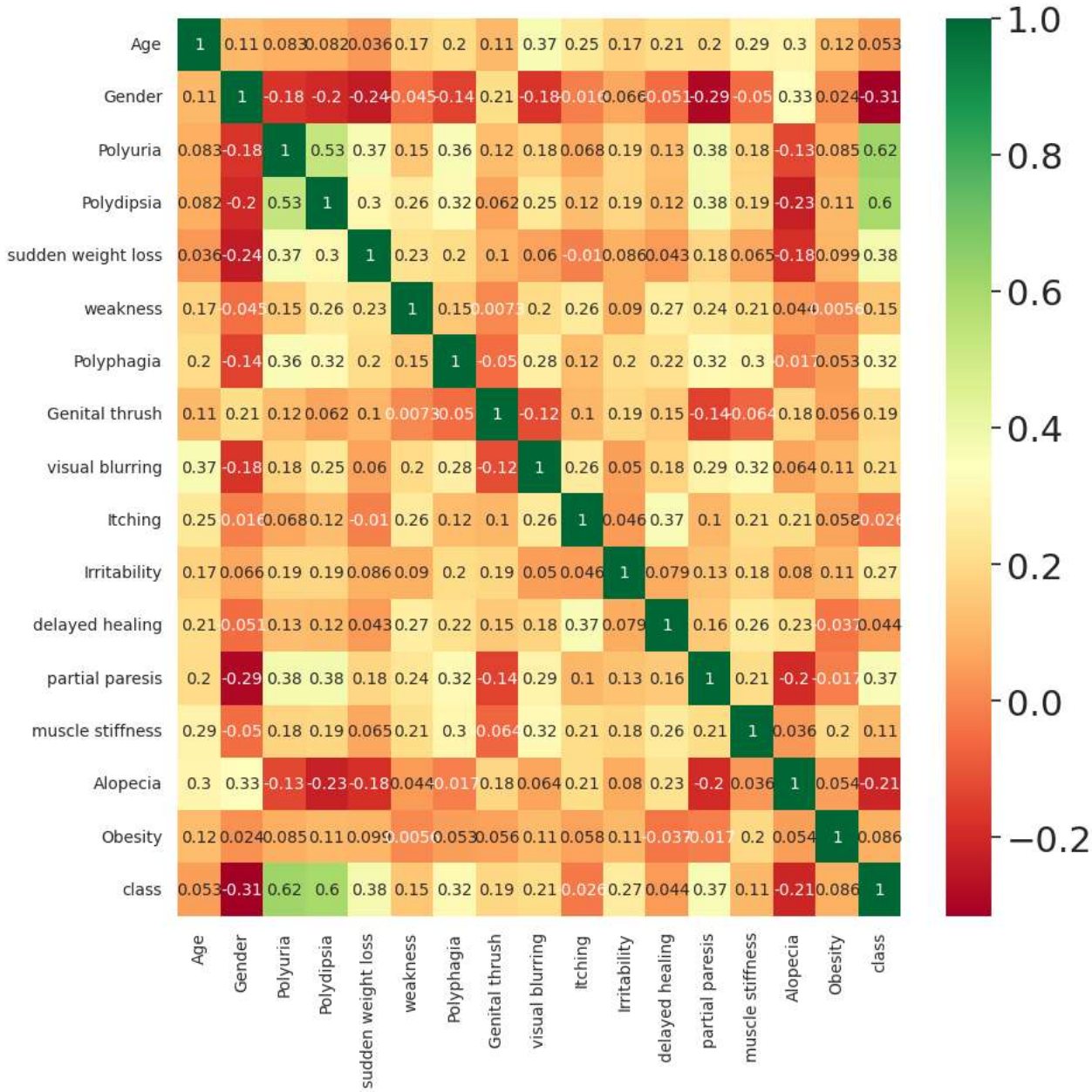
Feature Engineering

```
sns.set(font_scale=2)
plt.subplots(figsize=(10,10))
heat_plot= sns.heatmap(DFwithnoduplicate.corr(method='pearson'),annot=True,cmap='RdYlGn',annot_kws={'size':10})

plt.yticks(fontsize=10)
plt.xticks(fontsize=10)



plt.show()
```




```
correlations = DFwithnoduplicate.corr(method='pearson')
print(correlations['class'].sort_values(ascending=False).to_string())
```

```
class                 1.000000
Polyuria              0.619235
Polydipsia            0.598018
sudden weight loss    0.378853
partial paresis       0.366982
Polyphagia            0.318299
Irritability          0.268261
visual blurring       0.205055
Genital thrush        0.189799
weakness              0.146758
muscle stiffness      0.108025
Obesity               0.085882
Age                   0.053270
delayed healing       0.043818
Itching              -0.026500
Alopecia             -0.207104
Gender               -0.314769
```

Separating Feature and Target

```
DFwithnoduplicate
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	st
0	40.000000	1	0	1	0	1	0	0	0	1	0	1	0	
1	58.000000	1	0	0	0	1	0	0	1	0	0	0	1	
3	45.000000	1	0	0	1	1	1	1	0	1	0	1	0	
4	49.112186	1	1	1	1	1	1	0	1	1	1	1	1	
5	55.000000	1	1	1	0	1	1	0	1	1	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
515	39.000000	0	1	1	1	0	1	0	0	1	0	1	1	
516	48.000000	0	1	1	1	1	1	0	0	1	1	1	1	
517	58.000000	0	1	1	1	1	1	0	1	0	0	0	1	
518	32.000000	0	0	0	0	1	0	0	1	1	0	1	0	
519	42.000000	1	0	0	0	0	0	0	0	0	0	0	0	

Next steps: View recommended plots

```
from sklearn.utils import shuffle
shuffled_DF= shuffle(DFwithnoduplicate)
```

```
shuffled_DF
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	musc stiffne
17	67.0	1	0	1	0	1	1	0	1	0	1	1	1	
208	54.0	1	0	0	1	1	0	1	0	0	0	1	0	
247	53.0	1	0	0	0	1	1	0	1	1	0	1	1	
20	62.0	1	1	1	0	1	1	0	1	0	1	0	1	
116	30.0	0	1	1	1	0	1	0	0	0	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
30	57.0	1	1	1	1	1	1	0	1	0	0	0	1	
80	35.0	0	1	1	0	1	0	0	1	0	0	0	0	
243	35.0	1	0	0	0	1	0	0	0	0	0	0	0	
9	70.0	1	0	1	1	1	1	0	1	1	1	0	0	
12	35.0	1	1	0	0	0	1	1	0	0	1	1	0	

Next steps: View recommended plots

```
rearranged_DF=shuffled_DF.reset_index(drop=True) #re-arranging the index vales
```

```
rearranged_DF #see the index values
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	musc stiffne
0	67.0	1	0	1	0	1	1	0	1	0	1	1	1	
1	54.0	1	0	0	1	1	0	1	0	0	0	1	0	
2	53.0	1	0	0	0	1	1	0	1	1	0	1	1	
3	62.0	1	1	1	0	1	1	0	1	0	1	0	1	
4	30.0	0	1	1	1	0	1	0	0	0	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
244	57.0	1	1	1	1	1	1	0	1	0	0	0	1	
245	35.0	0	1	1	0	1	0	0	1	0	0	0	0	
246	35.0	1	0	0	0	1	0	0	0	0	0	0	0	
247	70.0	1	0	1	1	1	1	0	1	1	1	0	0	
248	35.0	1	1	0	0	0	1	1	0	0	1	1	0	

Next steps: View recommended plots

```
X= rearranged_DF.drop(columns=['class'])
Y= rearranged_DF['class']
```

```
X
```


	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	musc stiffne
0	67.0	1	0	1	0	1	1	0	1	0	1	1	1	
1	54.0	1	0	0	1	1	0	1	0	0	0	1	0	
2	53.0	1	0	0	0	1	1	0	1	1	0	1	1	
3	62.0	1	1	1	0	1	1	0	1	0	1	0	1	
4	30.0	0	1	1	1	0	1	0	0	0	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
244	57.0	1	1	1	1	1	1	0	1	0	0	0	1	
245	35.0	0	1	1	0	1	0	0	1	0	0	0	0	
246	35.0	1	0	0	0	1	0	0	0	0	0	0	0	
247	70.0	1	0	1	1	1	1	0	1	1	1	0	0	
248	35.0	1	1	0	0	0	1	1	0	0	1	1	0	

Next steps: View recommended plots


```
Y
```

```
0      1
1      0
2      0
3      1
4      1
      ..
244    1
245    1
246    0
247    1
248    1
Name: class, Length: 249, dtype: int64
```

Scalling data

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
```

```
#Scalling means have the data in a range
scaler1 = MinMaxScaler()
MinMax_scaled_DF=scaler1.fit_transform(X) #min-max scaling (0 to 1)
MinMax_scaled_DF
```

```
array([[0.68918919, 1.        , 0.        , ..., 1.        , 1.        ,
        1.        ],
       [0.51351351, 1.        , 0.        , ..., 0.        , 1.        ,
        0.        ],
       [0.5       , 1.        , 0.        , ..., 1.        , 1.        ,
        0.        ],
       ...,
       [0.25675676, 1.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.72972973, 1.        , 0.        , ..., 0.        , 1.        ,
        0.        ],
       [0.25675676, 1.        , 1.        , ..., 0.        , 1.        ,
        0.        ]])
```

```
scaler2=StandardScaler()
STD_scaled_DF=scaler2.fit_transform(X)
STD_scaled_DF
```

```
array([[ 1.44820523,  0.75891328, -1.0451971 , ...,  1.26243812,
         1.35260691,  2.15849274],
       [ 0.40966805,  0.75891328, -1.0451971 , ..., -0.79211803,
         1.35260691, -0.46328625],
       [ 0.32978057,  0.75891328, -1.0451971 , ...,  1.26243812,
         1.35260691, -0.46328625],
       ...,
       [-1.10819399,  0.75891328, -1.0451971 , ..., -0.79211803,
        -0.73931309, -0.46328625],
       [ 1.68786766,  0.75891328, -1.0451971 , ..., -0.79211803,
         1.35260691, -0.46328625],
       [-1.10819399,  0.75891328,  0.95675734,        -0.79211803
```