logistic regression practise

February 11, 2023

```
[1]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import warnings
     warnings.filterwarnings('ignore')
```

```
[5]: df=pd.read_csv('diabetes.csv')
```

```
[6]: df.head()
```

```
[6]:    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
```

```
[15]: df.isna().sum()
```

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

```
[52]: df['BMI'].replace([np.inf,-np.inf],np.nan,inplace=True)
```

```
[44]: df['BloodPressure'].unique()
```

```
[44]: array([4.27666612, 4.18965474, 4.15888308, 3.68887945, 4.30406509,
             3.91202301, 4.26694579, 4.24849524, 4.56434819, 4.52178858,
             4.38202663, 4.09434456, 4.4308168 , 3.40119738, 4.47733681,
             4.49980967, 4.54329478, 4.33073334, 4.40671925, 4.31748811,
             4.06044301, 4.35670883, 4.21950771, 4.70048037, 4.02535169,
             4.12713439, 4.44265126, 4.4543473 , 3.87120101, 3.78418963,
             4.17438727, 4.68213123, 4.00733319, 4.80402104, 3.98898405,
             3.95124372, 4.58496748, 4.6443909 , 4.55387689, 3.8286414 ,
             4.62497281, 4.60517019, 4.11087386, 3.17805383, 3.63758616,
             4.66343909, 4.73619845])
```

```
[31]: from sklearn.impute import SimpleImputer
```

```
[32]: si=SimpleImputer()
```

```
[53]: df[['BMI']]=si.fit_transform(df[['BMI']])
```

```
[7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

```
[8]: df.describe()
```

```
[8]:        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
```

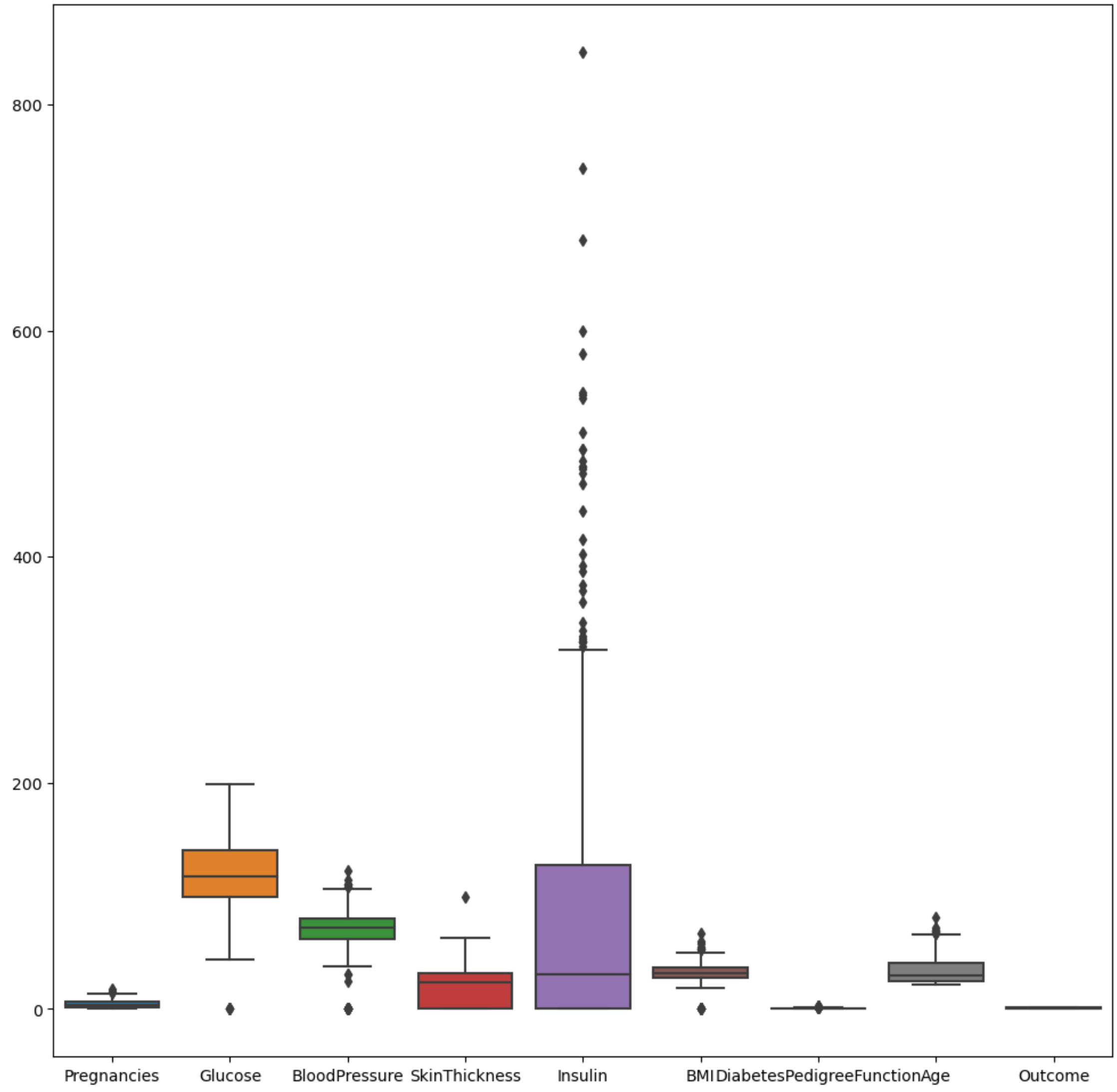
```
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              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
```

[11]:

```
plt.figure(figsize=(12,12))
sns.boxplot(data=df)
```

[11]:

```
<AxesSubplot:>
```



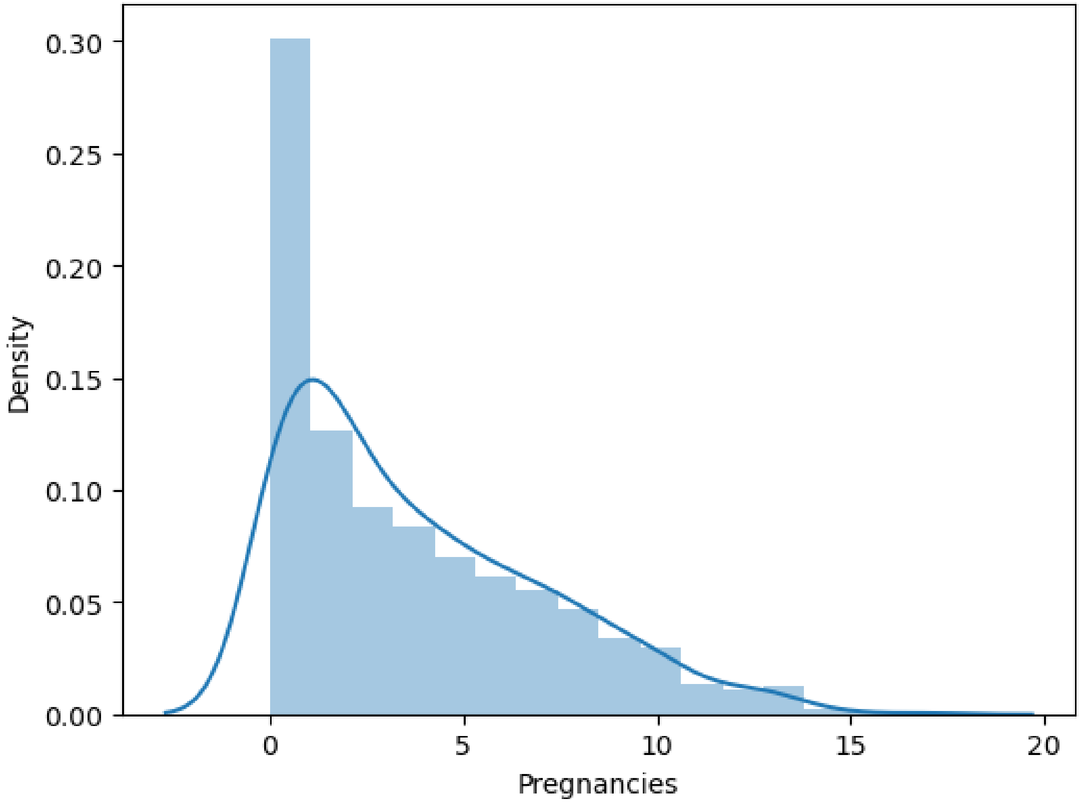
```
[14]: df.corr().style.background_gradient()
```

```
[14]: <pandas.io.formats.style.Styler at 0x200ef588fa0>
```

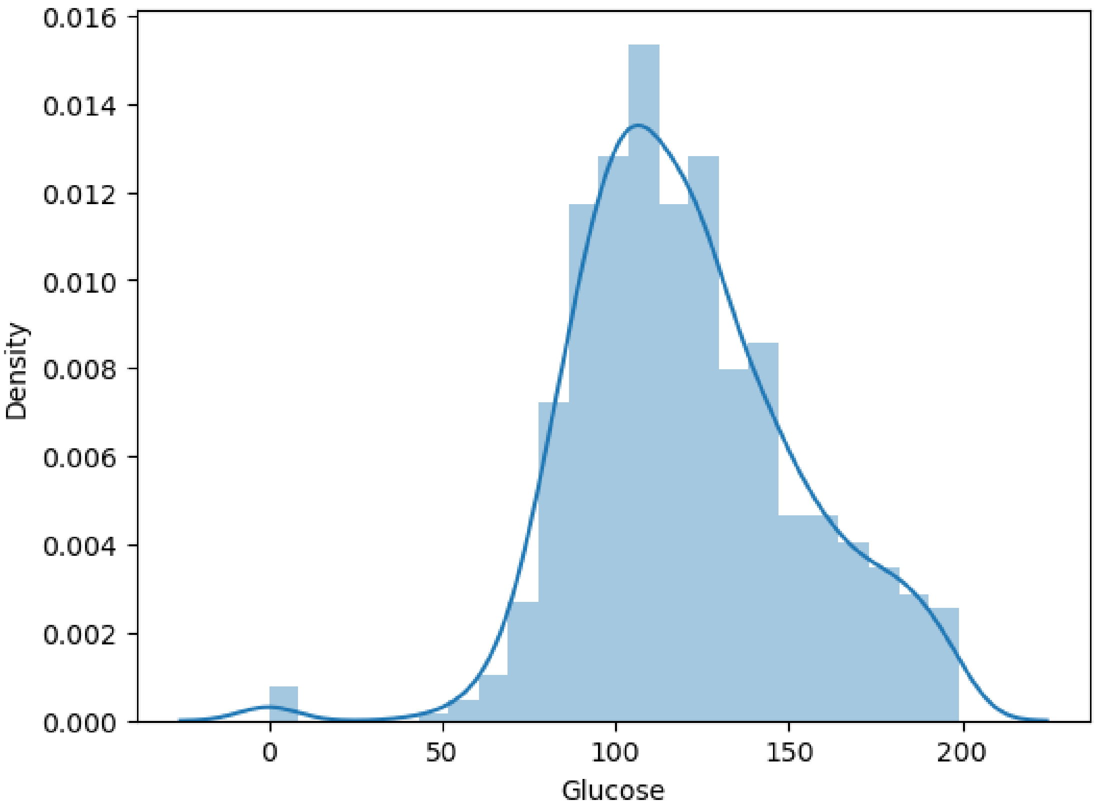
```
[16]: from scipy.stats import skew
```

```
[25]: for i in df[colname]:
          print(i)
          print(skew(df[i]))
          sns.distplot(df[i])
          plt.show()
```

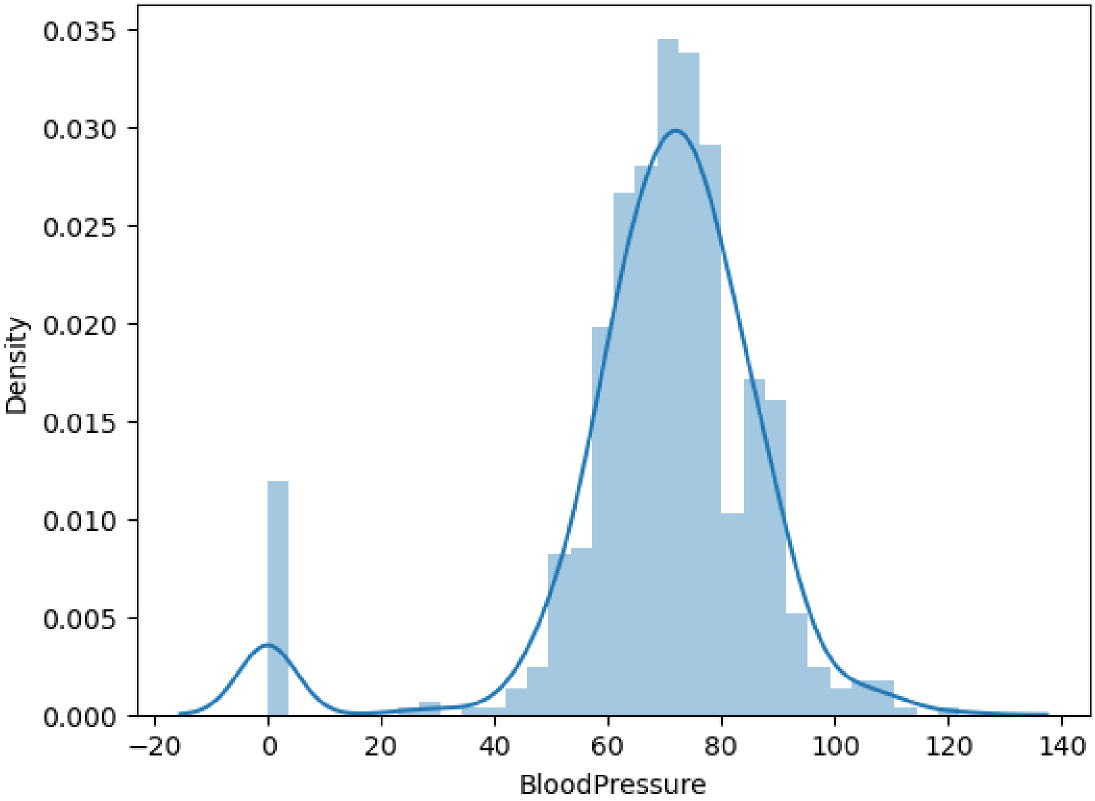
```
Pregnancies
0.8999119408414357
```



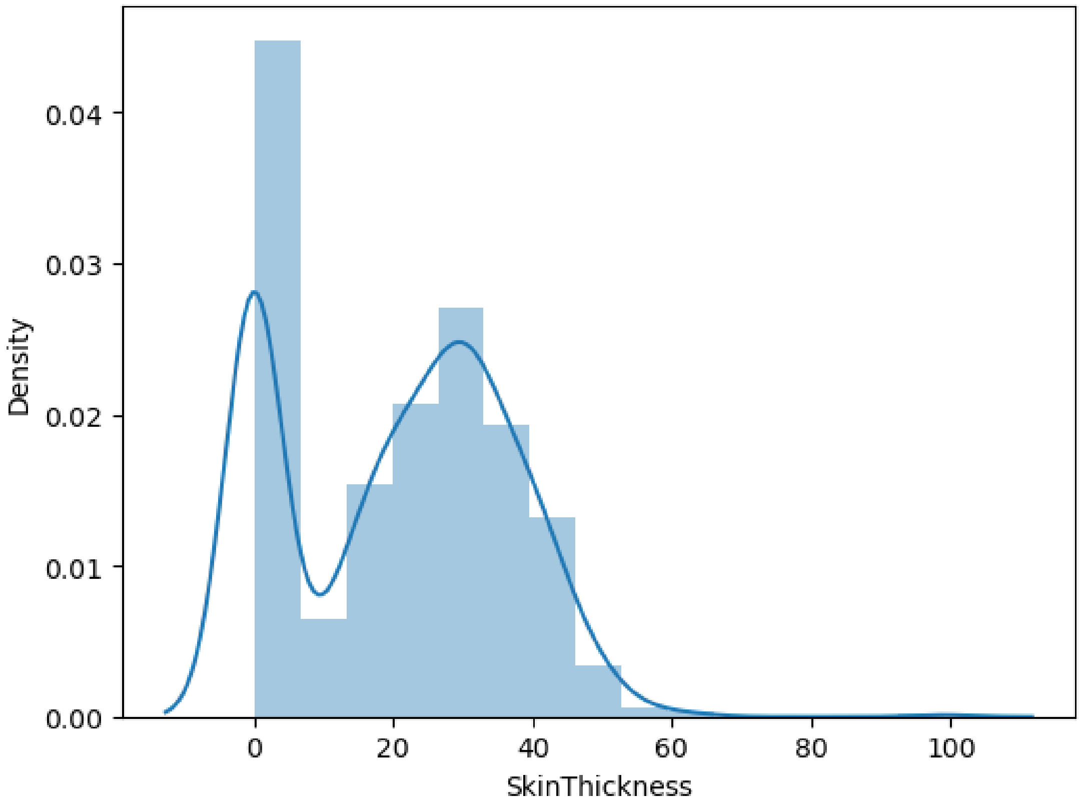
```
Glucose
0.17341395519987735
```



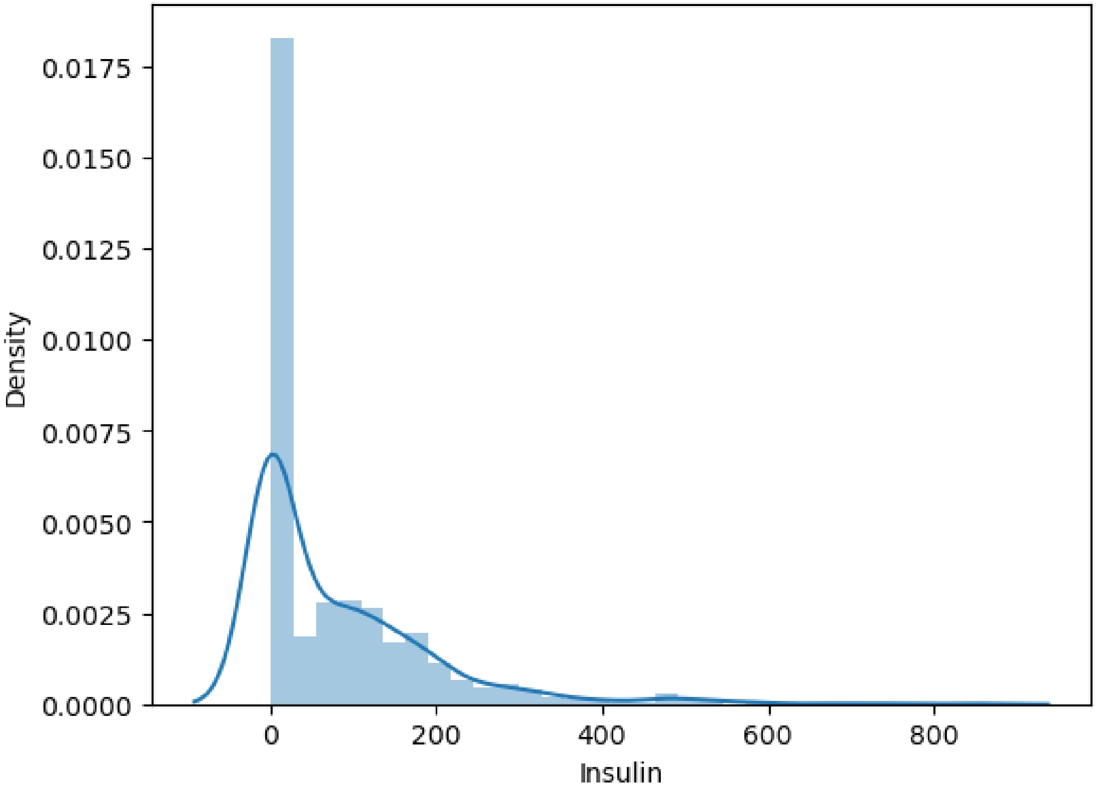
```
BloodPressure
-1.8400052311728738
```



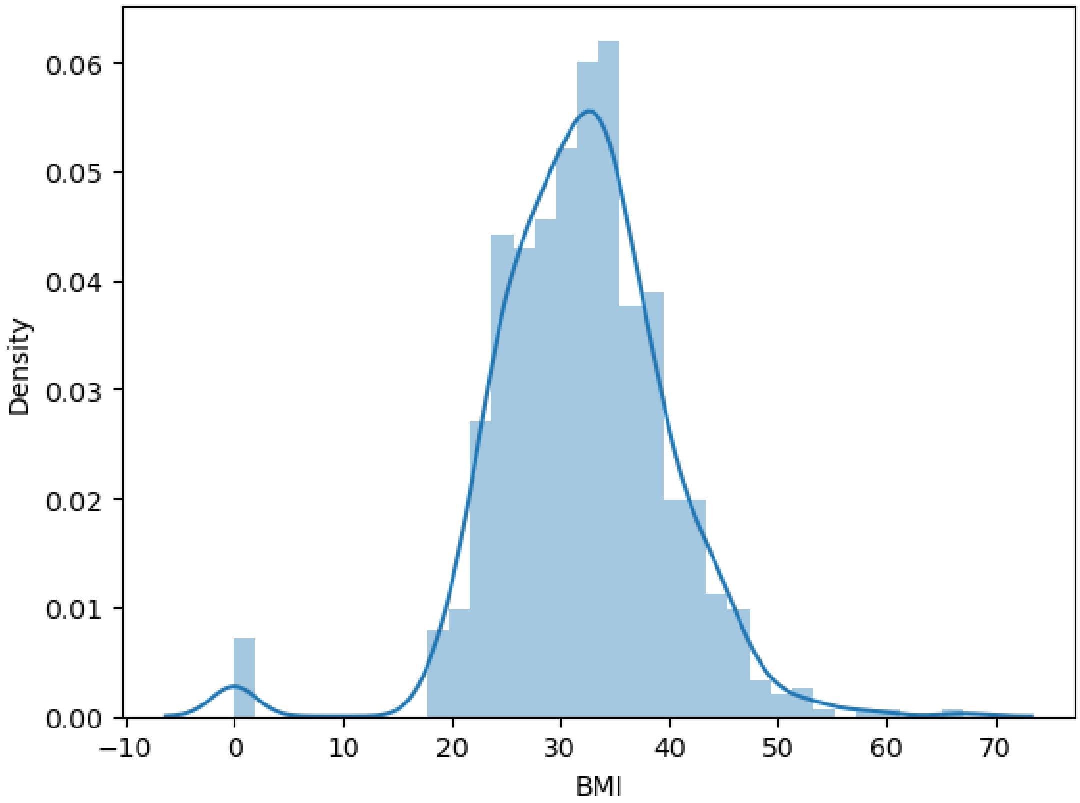
```
SkinThickness
0.109158762323673
```



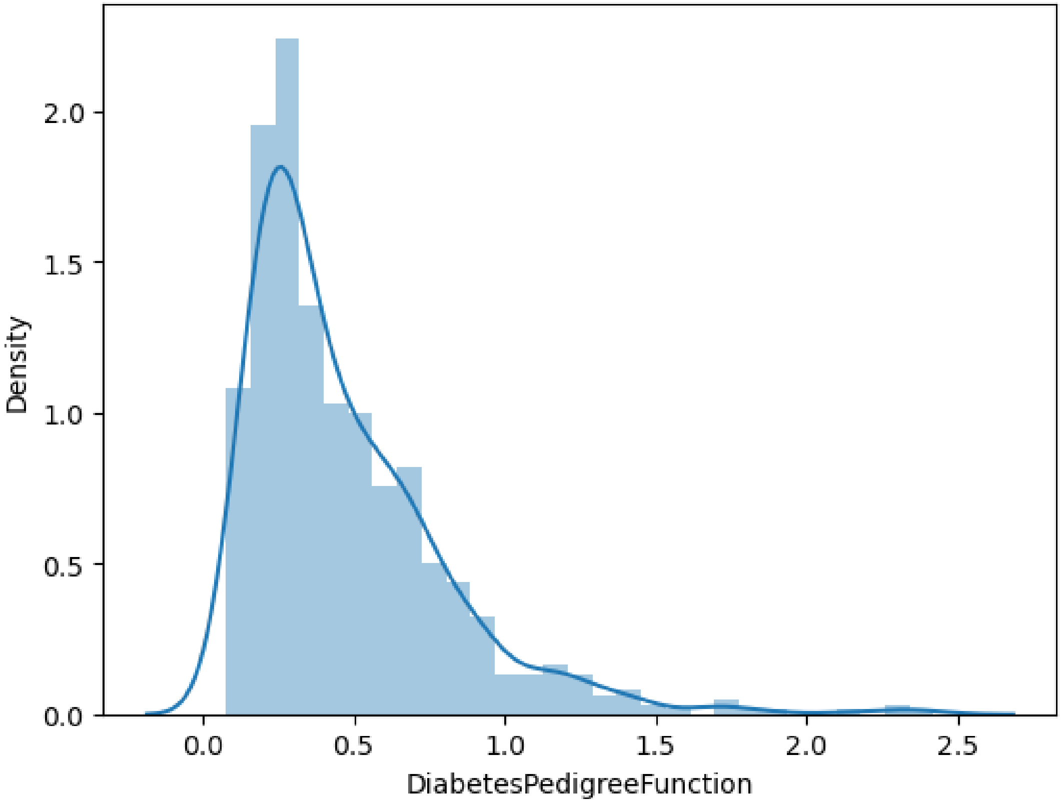
```
Insulin
2.2678104585131753
```



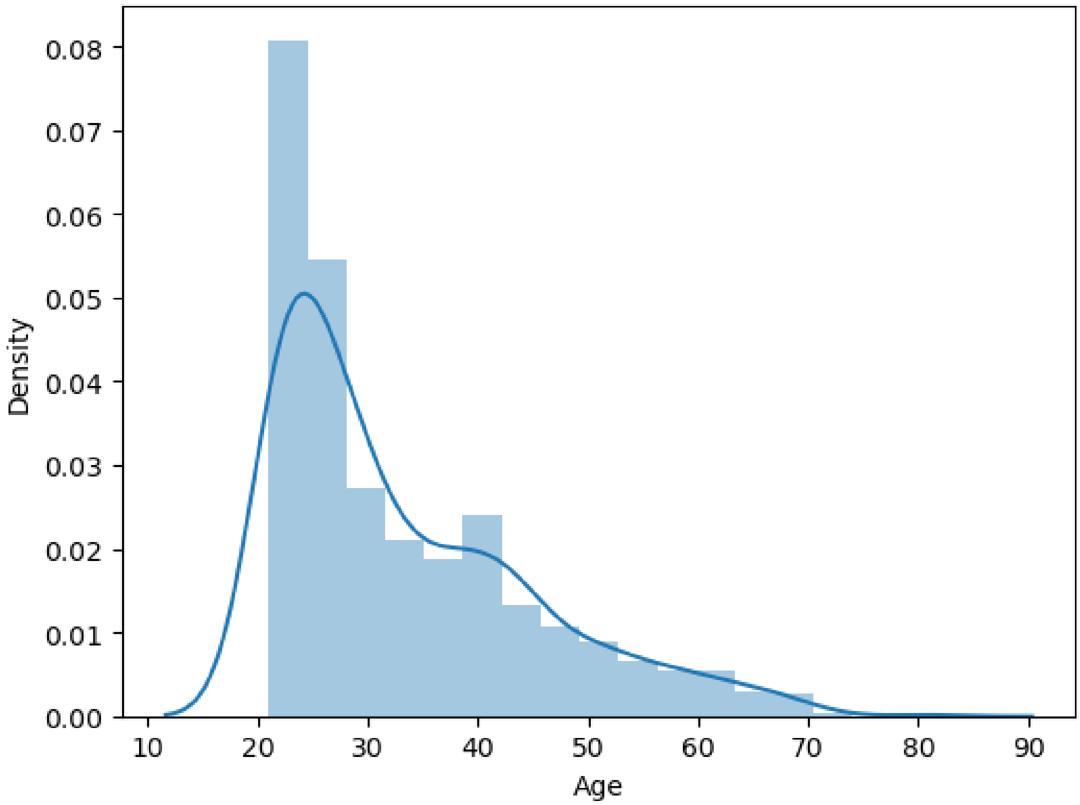
```
BMI
-0.42814327880861786
```



```
DiabetesPedigreeFunction
1.9161592037386292
```



```
Age
1.127389259531697
```



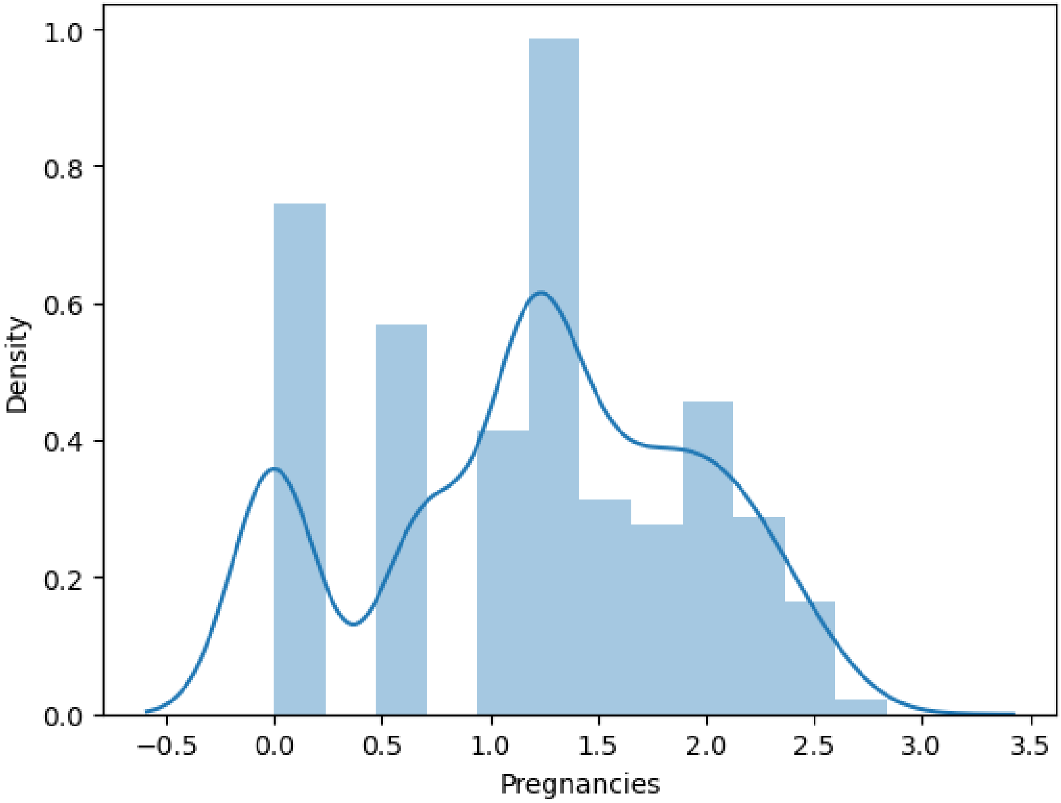
[26]:

```
df[colname]=np.log(df[colname])
```

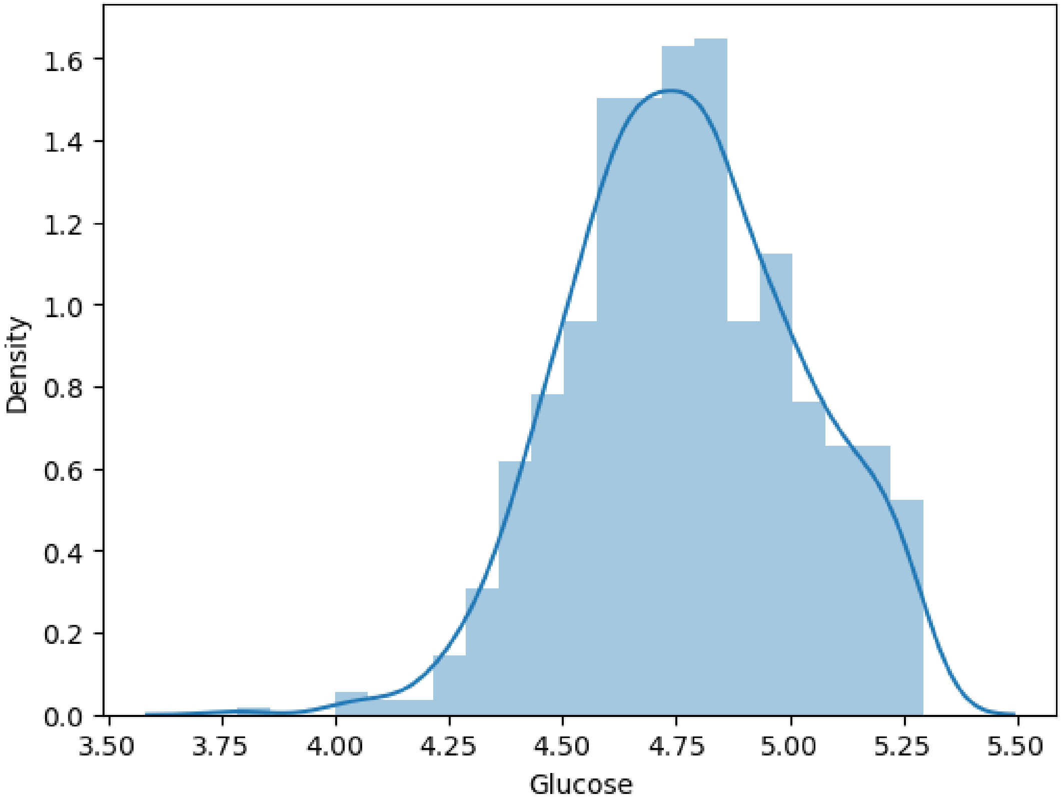
[54]:

```
for i in df[colname]:
    print(i)
    print(skew(df[i]))
    sns.distplot(df[i])
    plt.show()
```

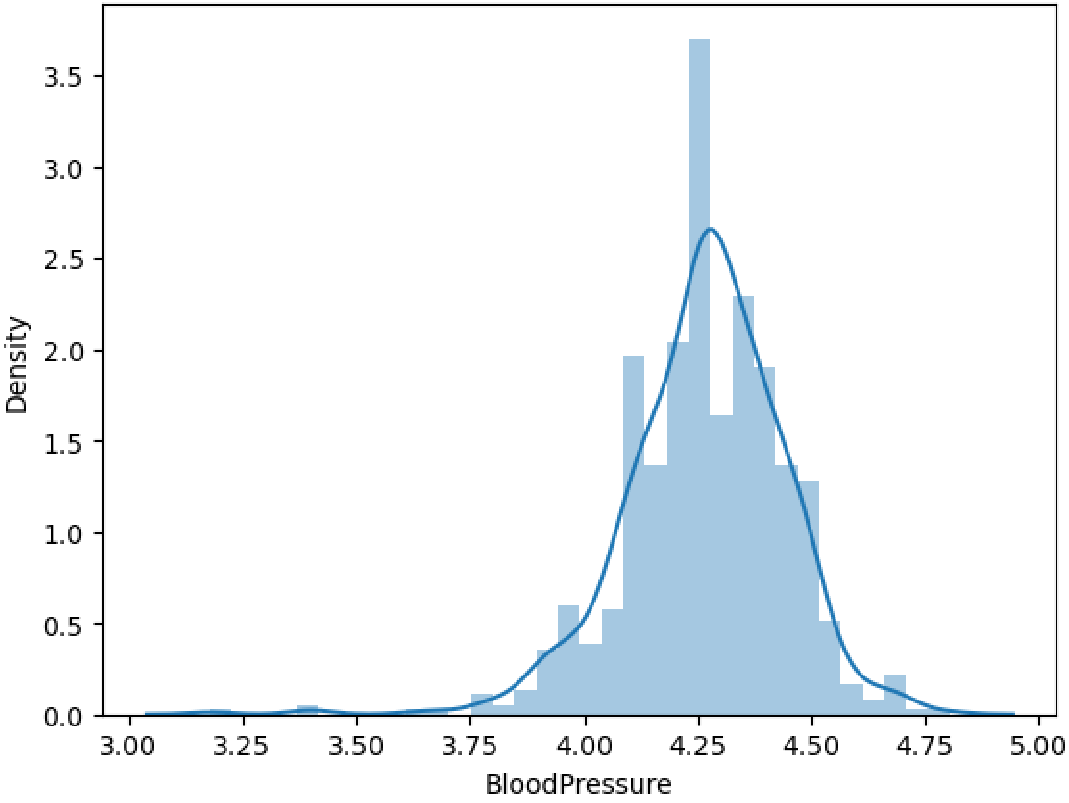
```
Pregnancies
-0.22092086734048239
```



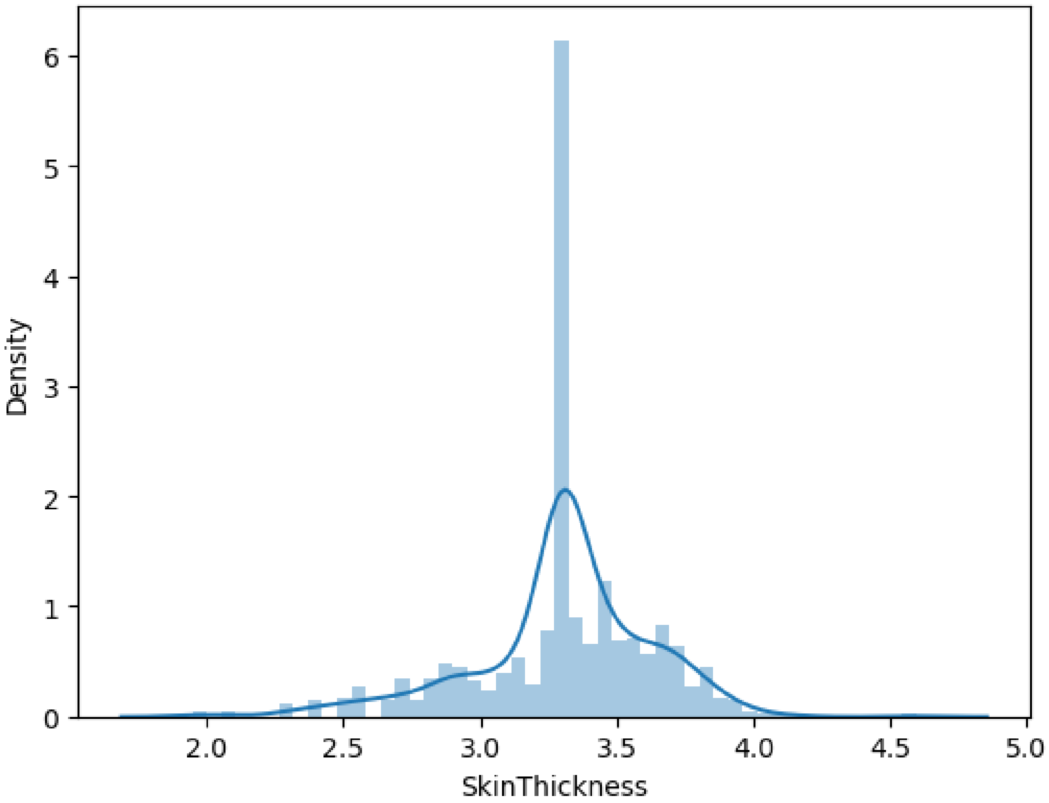
```
Glucose
-0.06522741519898129
```



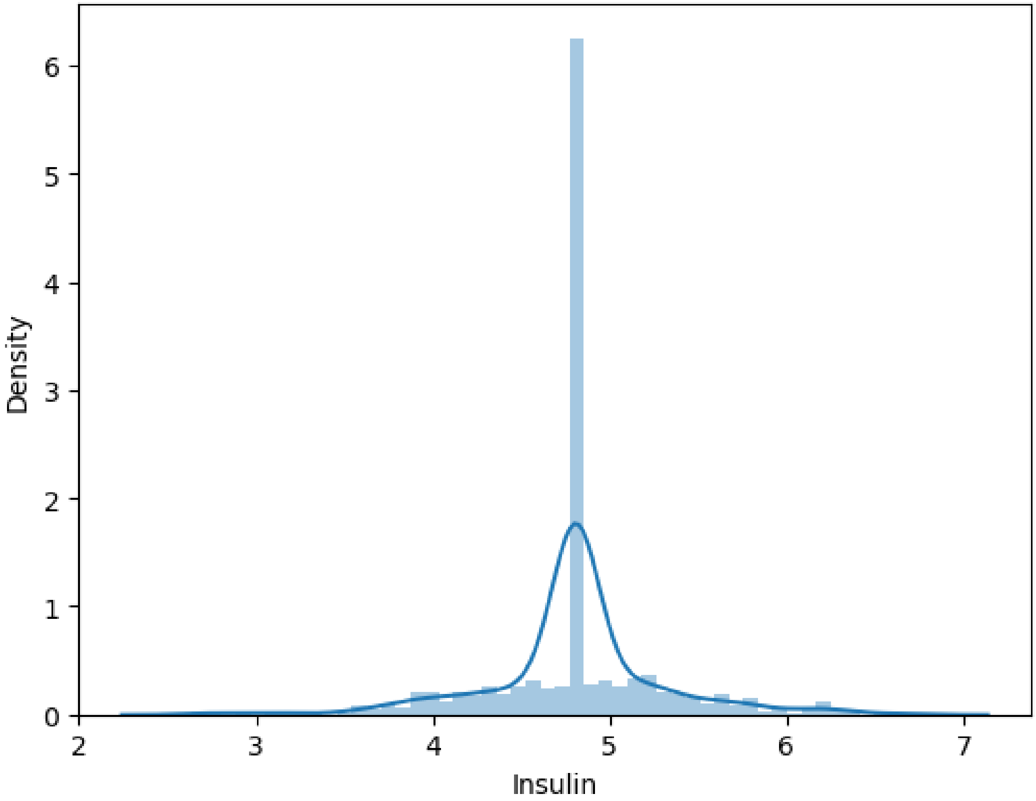
```
BloodPressure
-0.8117813252892095
```



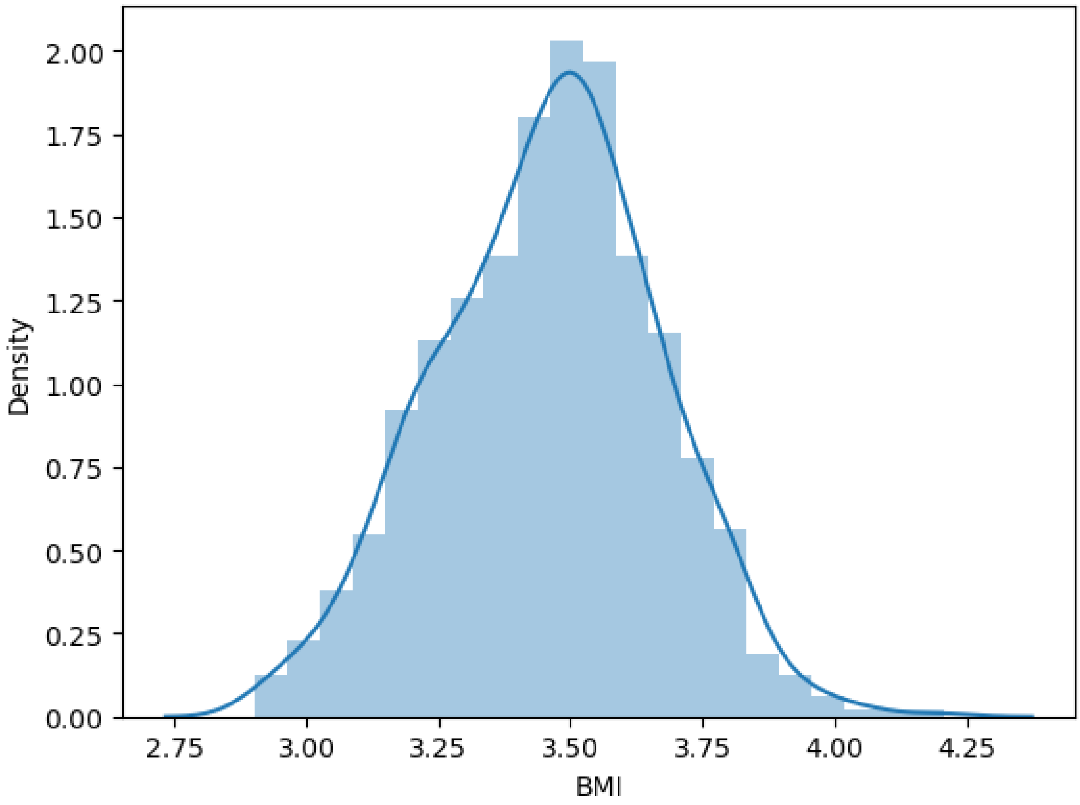
```
SkinThickness
-0.757853177699361
```



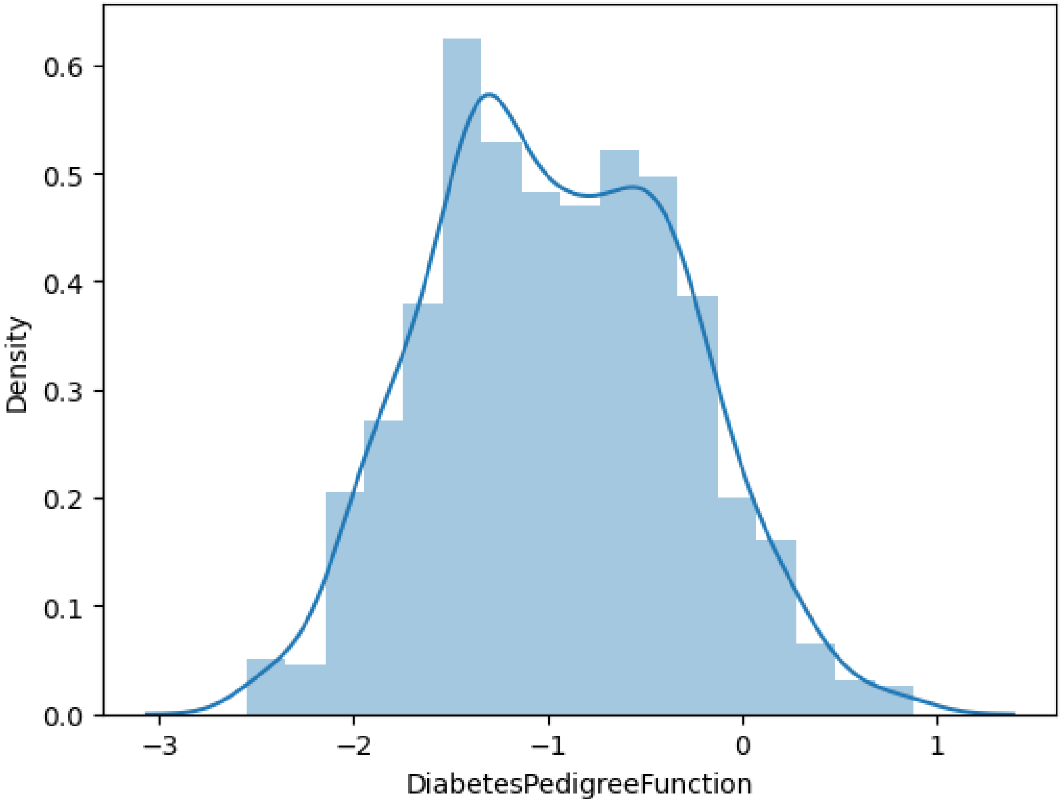
```
Insulin
-0.1591138900700523
```



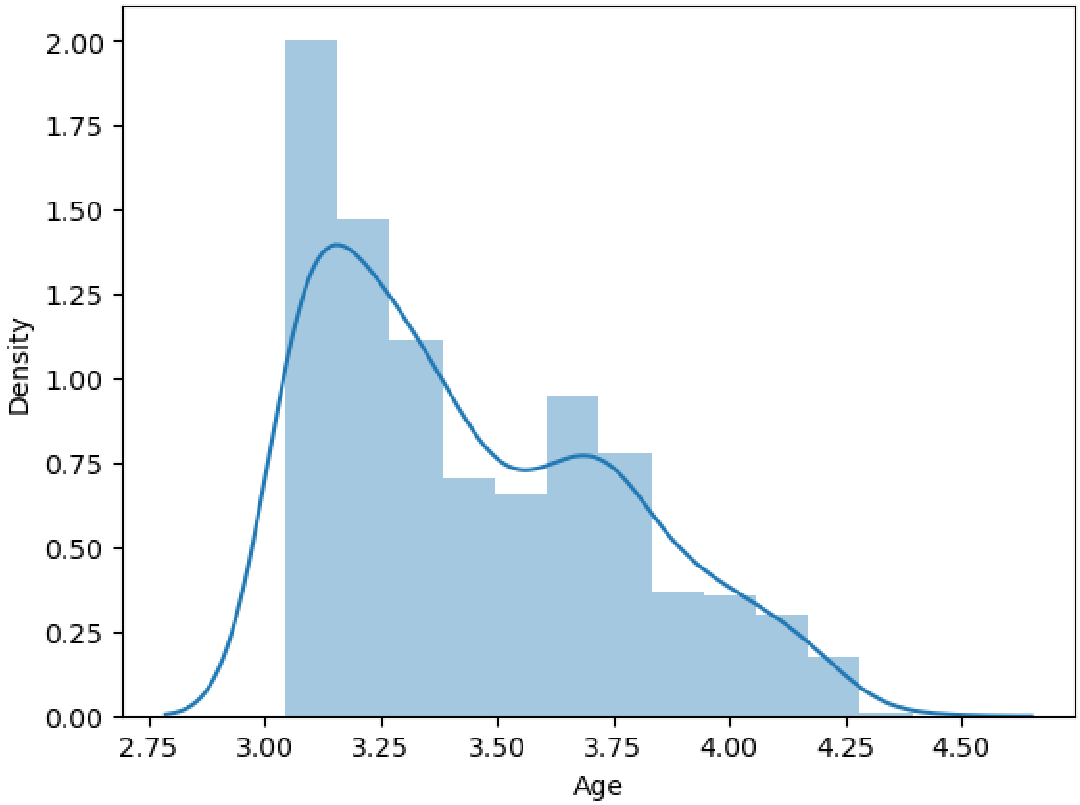
```
BMI
-0.05242667957133482
```



```
DiabetesPedigreeFunction
0.11395456387082803
```



```
Age
0.6005702138973051
```



```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[24]: df[colname]
```

```
[24]:      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
      ..           …        …              …              …        …     …
```

```
763            10      101             76             48       180  32.9
764             2      122             70             27         0  36.8
765             5      121             72             23       112  26.2
766             1      126             60              0         0  30.1
767             1       93             70             31         0  30.4

     DiabetesPedigreeFunction  Age
0                       0.627   50
1                       0.351   31
2                       0.672   32
3                       0.167   21
4                       2.288   33
..                        …    …
763                     0.171   63
764                     0.340   27
765                     0.245   30
766                     0.349   47
767                     0.315   23

[768 rows x 8 columns]
```

[23]:

```
colname
```

[23]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age'],
      dtype='object')
```

[59]:

```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

[60]:

```
colname=x.columns
```

[61]:

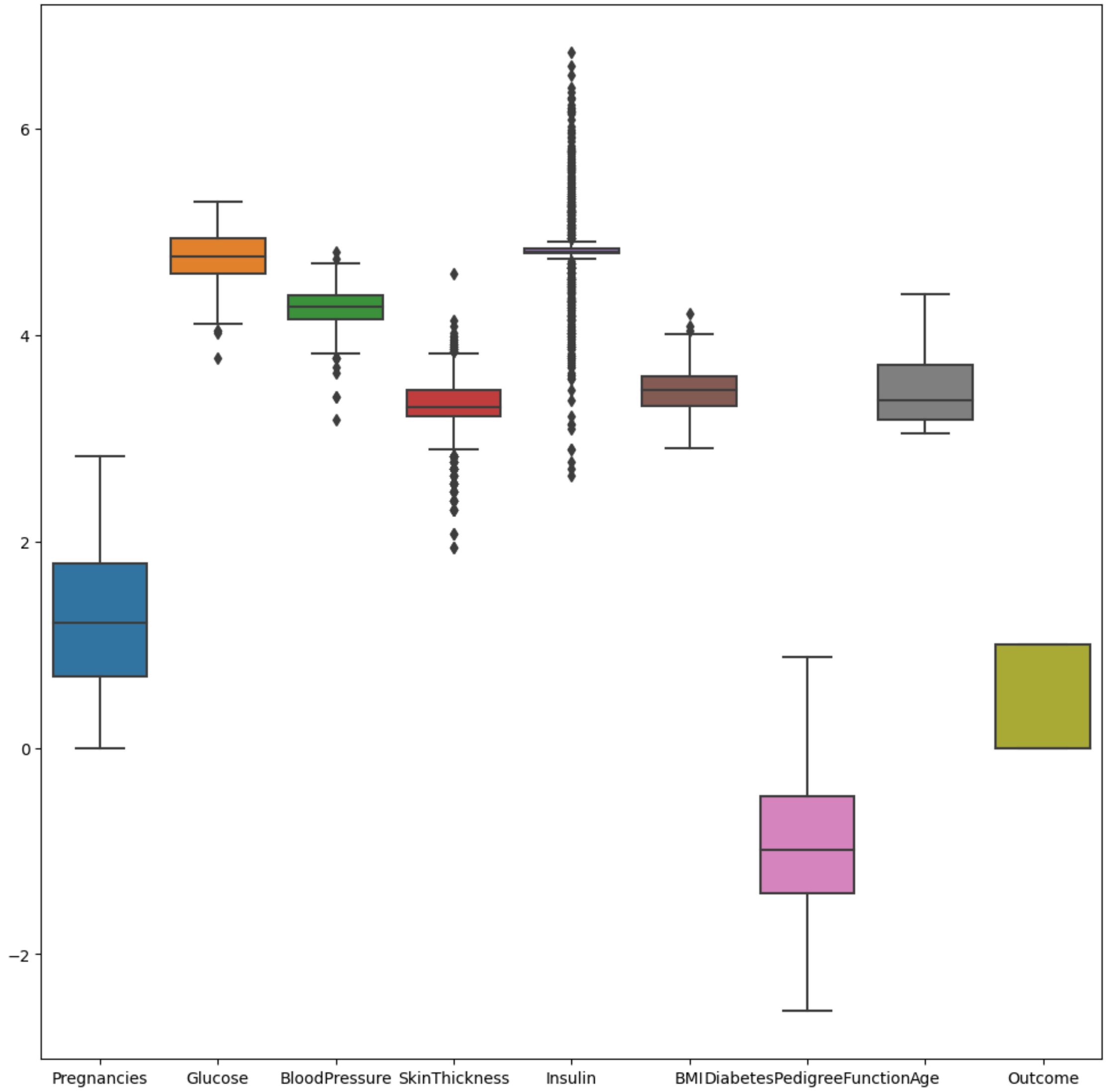
```
y
```

[61]:

```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

[55]:
```
plt.figure(figsize=(12,12))
sns.boxplot(data=df)
```

[55]:
```
<AxesSubplot:>
```



[58]:
```
from sklearn.model_selection import train_test_split
```

[63]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

[64]:
```
xtrain
```

[64]:
```
     Pregnancies   Glucose  BloodPressure  SkinThickness   Insulin       BMI  \
88      2.708050  4.912655       4.248495       3.465736  4.700480  3.613617
```

```
467     1.214556  4.574711       4.158883       3.583519  4.605170  3.605498
550     0.000000  4.753590       4.248495       3.332205  4.808038  3.310543
147     0.693147  4.663439       4.158883       3.555348  4.779123  3.417727
481     1.214556  4.812184       4.477337       3.610918  4.808038  3.561046
..           …       …             …             …       …       …
645     0.693147  5.056246       4.304065       3.555348  6.086775  3.673766
715     1.945910  5.231109       3.912023       3.496508  5.971262  3.523415
72      2.564949  4.836282       4.499810       3.302782  4.808038  3.770459
235     1.386294  5.141664       4.276666       3.302782  4.808038  3.775057
37      2.197225  4.624973       4.330733       3.610918  4.808038  3.493473

     DiabetesPedigreeFunction       Age
88                  -1.877317  3.761200
467                 -0.510826  3.218876
550                 -1.589635  3.044522
147                  0.336472  3.526361
481                 -1.624552  3.367296
..                        …       …
645                 -2.009915  3.401197
715                 -0.191161  3.526361
72                  -0.539568  3.737670
235                 -0.736055  3.258097
37                  -0.407968  3.828641

[537 rows x 8 columns]
```

```
[ ]:
```

```
[ ]:
```

```
[56]: from sklearn.linear_model import LogisticRegression
```

```
[57]: lr=LogisticRegression()
```

```
[65]: lr.fit(xtrain,ytrain)
```

```
[65]: LogisticRegression()
```

```
[66]: ypred=lr.predict(xtest)
```

```
[67]: ypred
```

```
[67]: array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
              1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
              0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
              1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0,
              0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0,
              0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
              1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0,
              1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0,
              1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
              0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
              0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0], dtype=int64)
```

```
[ ]:
```