In [1]:

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

In [2]:

```python
df = pd.read_csv('diabetes-data.csv')
```

In [3]:

```python
df
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
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	

768 rows × 9 columns

In [4]:

```python
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
```

In [5]:

```python
df.shape
```

Out[5]:

```
(768, 9)
```

In [6]:

```
df.describe()
```

Out[6]:

```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin         BMI  Diab
count   768.000000  768.000000     768.000000     768.000000  768.000000  768.000000
 mean     3.845052  120.894531      69.105469      20.536458   79.799479   31.992578
  std     3.369578   31.972618      19.355807      15.952218  115.244002    7.884160
  min     0.000000    0.000000       0.000000       0.000000    0.000000    0.000000
  25%     1.000000   99.000000      62.000000       0.000000    0.000000   27.300000
  50%     3.000000  117.000000      72.000000      23.000000   30.500000   32.000000
  75%     6.000000  140.250000      80.000000      32.000000  127.250000   36.600000
  max    17.000000  199.000000     122.000000      99.000000  846.000000   67.100000
```

In [7]:

```
df_copy = df.copy(deep=True)
```

In [8]:

```
df_copy
```

Out[8]:

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  DiabetesPedigreeFu
  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

768 rows × 9 columns
```

In [9]:

```
df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] =
```

In [10]:

```
df_copy
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
0	6	148.0	72.0	35.0	NaN	33.6	
1	1	85.0	66.0	29.0	NaN	26.6	
2	8	183.0	64.0	NaN	NaN	23.3	
3	1	89.0	66.0	23.0	94.0	28.1	
4	0	137.0	40.0	35.0	168.0	43.1	
...	...	...	...	...	...	...	
763	10	101.0	76.0	48.0	180.0	32.9	
764	2	122.0	70.0	27.0	NaN	36.8	
765	5	121.0	72.0	23.0	112.0	26.2	
766	1	126.0	60.0	NaN	NaN	30.1	
767	1	93.0	70.0	31.0	NaN	30.4	

768 rows × 9 columns

In [11]:

```
df_copy.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                   763 non-null    float64
 2   BloodPressure             733 non-null    float64
 3   SkinThickness             541 non-null    float64
 4   Insulin                   394 non-null    float64
 5   BMI                       757 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(6), int64(3)
memory usage: 54.1 KB
```

In [12]:

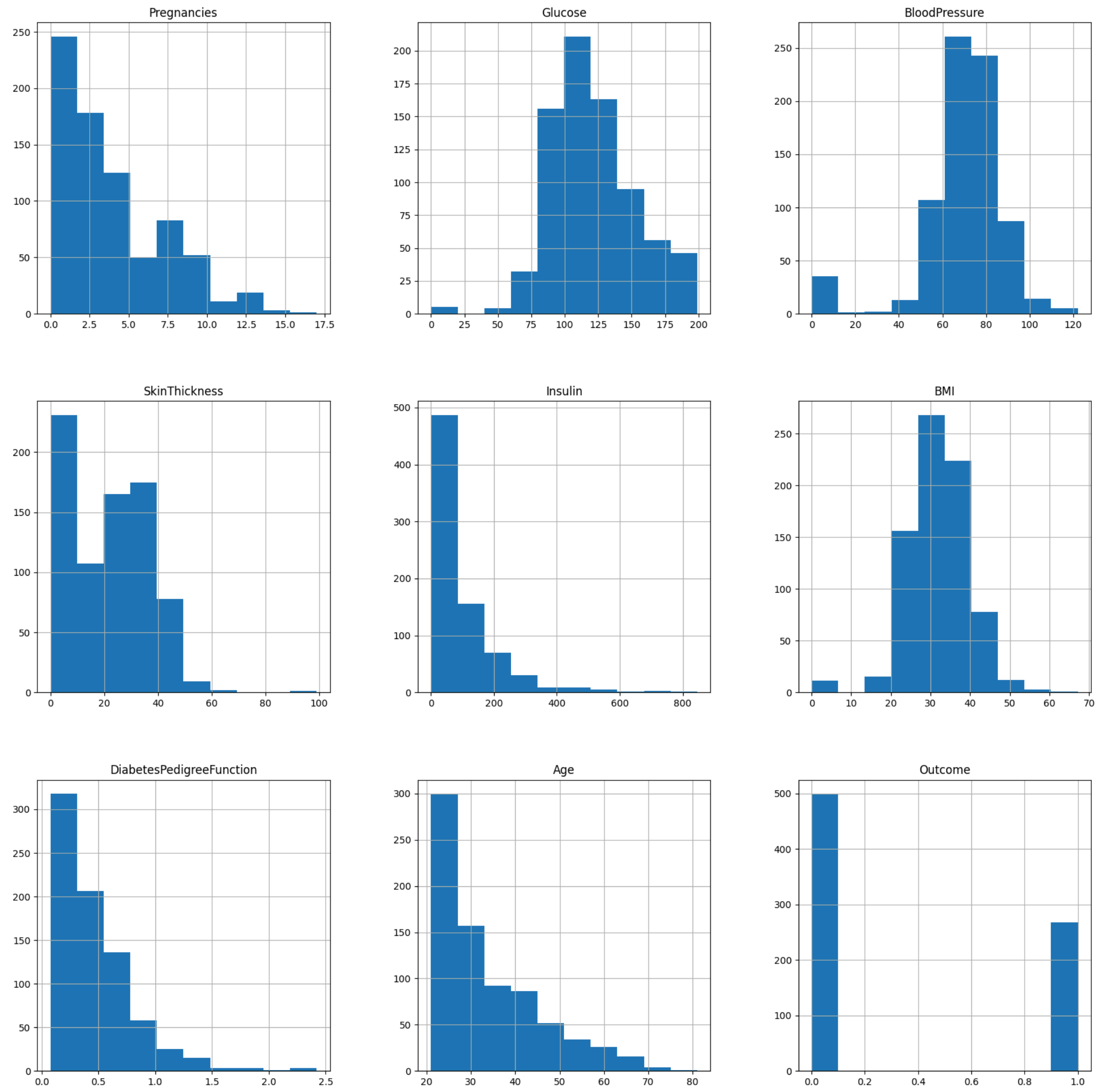
```
df_copy.isnull().sum()
```

Out[12]:

```
Pregnancies                   0
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
Outcome                       0
dtype: int64
```

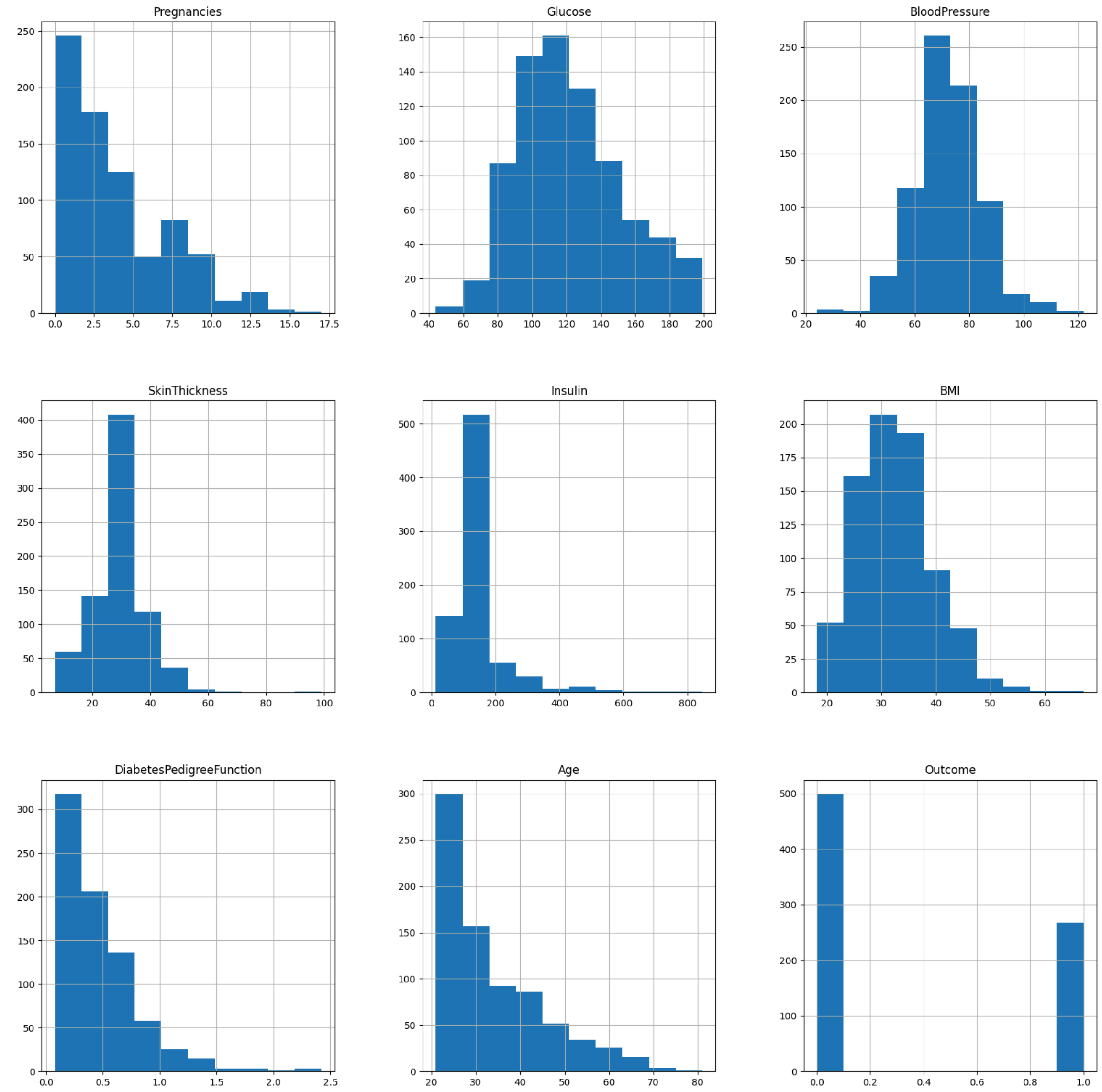
In [13]:

```
hplot = df.hist(figsize=(20,20))
```



In [14]:

```
df_copy['Glucose'].fillna(df_copy['Glucose'].mean(),inplace=True)
df_copy['BloodPressure'].fillna(df_copy['BloodPressure'].mean(),inplace
df_copy['SkinThickness'].fillna(df_copy['SkinThickness'].median(),inpla
df_copy['Insulin'].fillna(df_copy['Insulin'].median(),inplace=True)
df_copy['BMI'].fillna(df_copy['BMI'].median(),inplace=True)
hplot=df_copy.hist(figsize=(20,20))
```

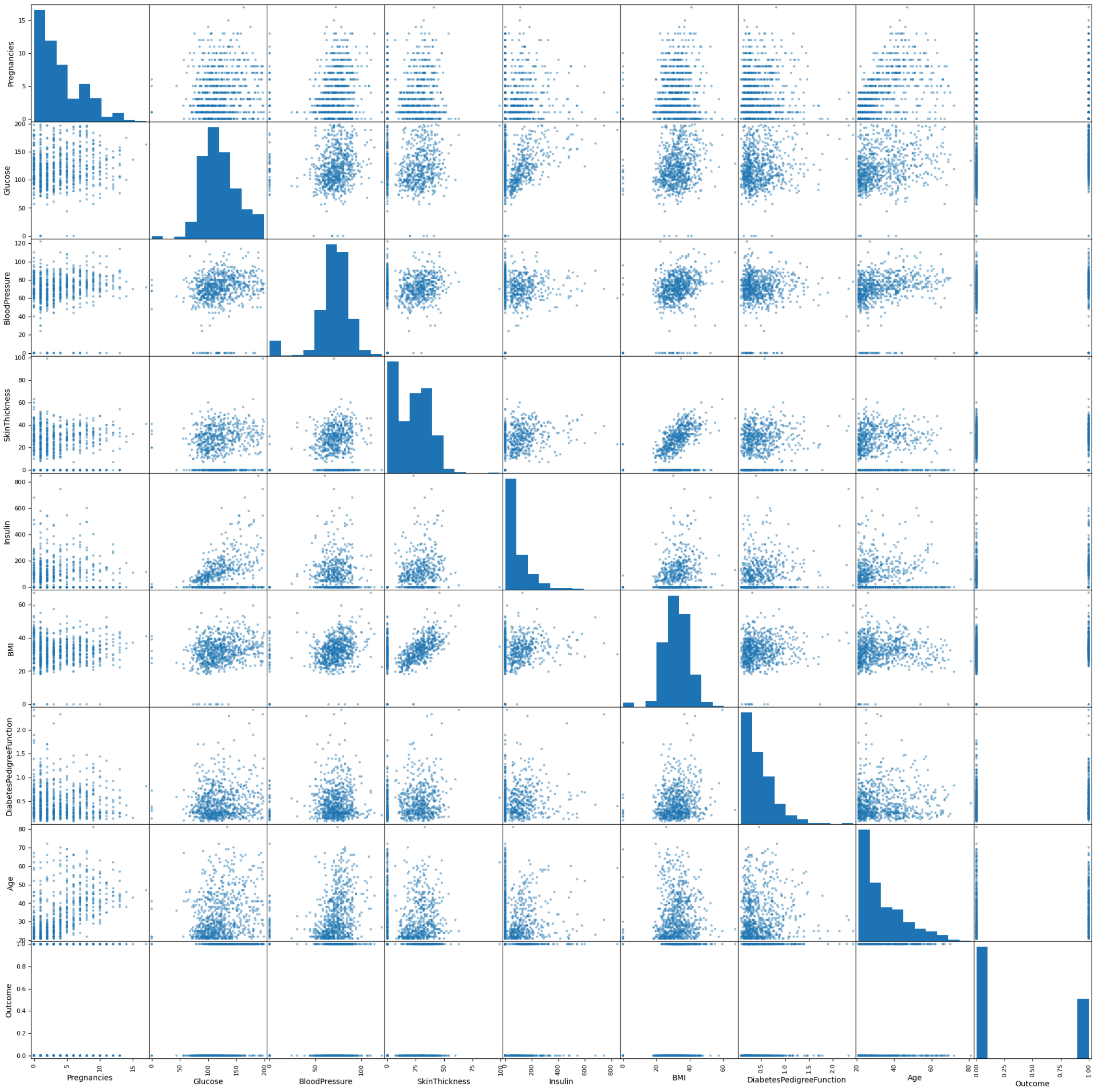


In [15]:

```
from pandas.plotting import scatter_matrix
```

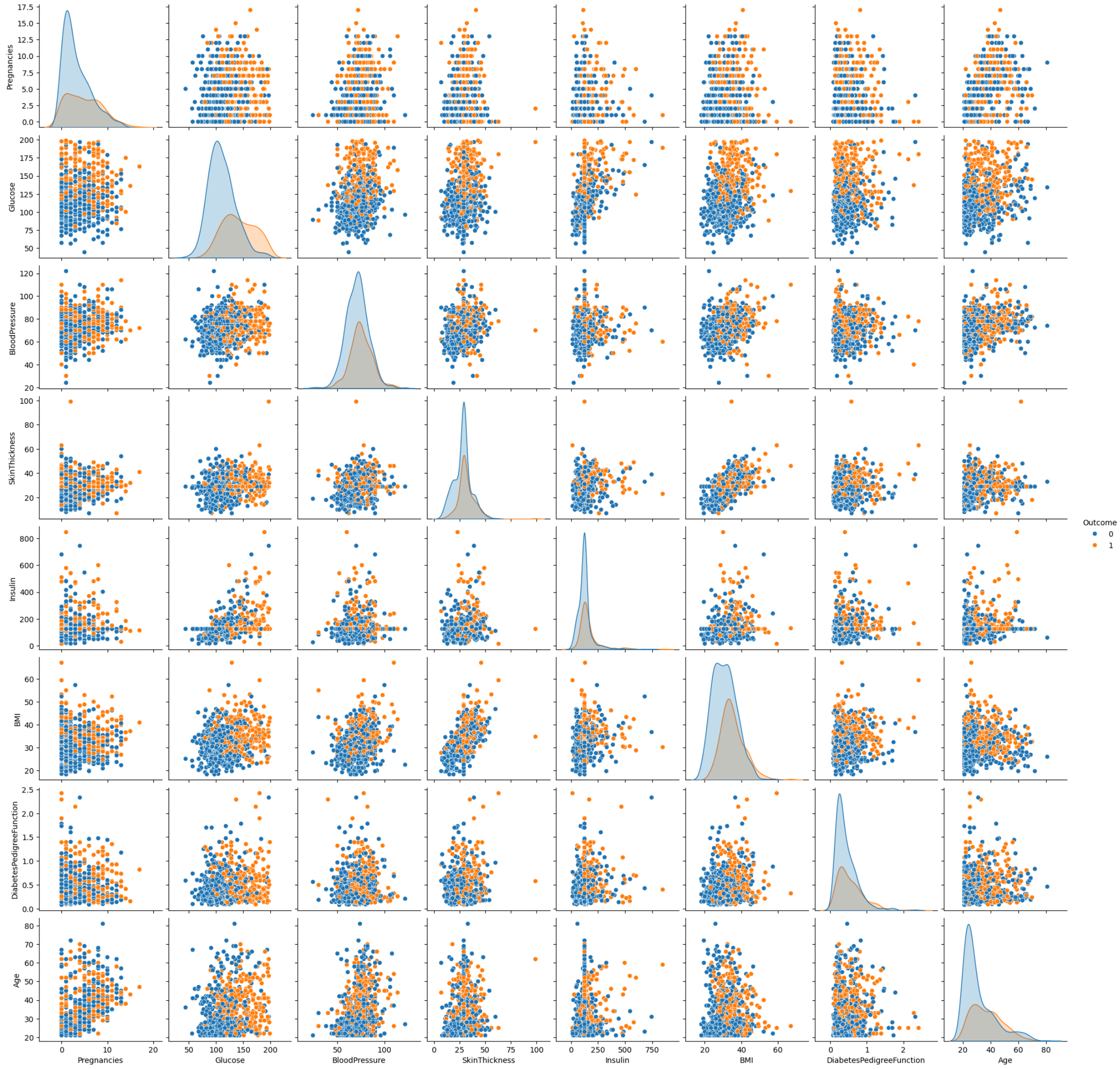
In [16]:

```
p = scatter_matrix(df,figsize=(25,25))
```

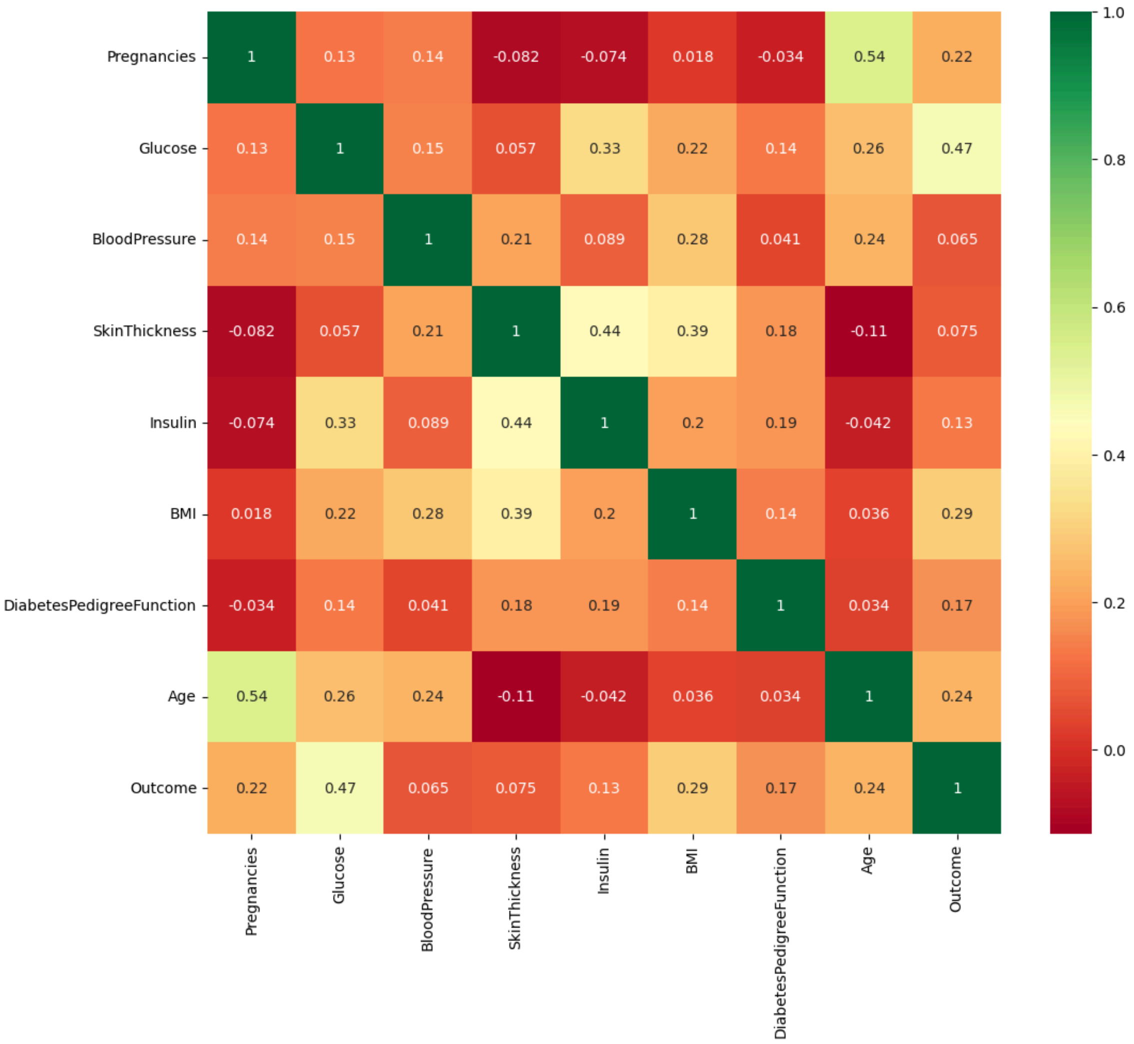


In [17]:

```
p = sns.pairplot(df_copy,hue='Outcome')
```



In [18]:
```python
plt.figure(figsize=(12,10))
p = sns.heatmap(df.corr(),annot=True,cmap='RdYlGn')
```



In [19]:
```python
from sklearn.preprocessing import StandardScaler
```

In [20]:
```python
scale_X = StandardScaler()
```

In [21]:
```python
X = scale_X.fit_transform(df_copy.drop(['Outcome'],axis=1),)
X = pd.DataFrame(X,columns=['Pregnancies','Glucose','BloodPressure','Sk
```

In [22]:
```python
X.head()
```

Out[22]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedi
0	0.639947	0.865108	-0.033518	0.670643	-0.181541	0.166619	
1	-0.844885	-1.206162	-0.529859	-0.012301	-0.181541	-0.852200	
2	1.233880	2.015813	-0.695306	-0.012301	-0.181541	-1.332500	
3	-0.844885	-1.074652	-0.529859	-0.695245	-0.540642	-0.633881	
4	-1.141852	0.503458	-2.680669	0.670643	0.316566	1.549303	

In [23]:

```
from sklearn.model_selection import train_test_split
```

In [24]:

```
y = df_copy.Outcome
```

In [25]:

```
X_train,X_test,Y_train,Y_test = train_test_split(X,y,test_size=1/3,ranc
```

In [26]:

```
#K-Nearest-Neighbors

from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
testing_score=[]
training_score=[]
for i in range(1,15):
    knn = KNeighborsClassifier(i)
    knn.fit(X_train,Y_train)
    training_score.append(knn.score(X_train,Y_train))
    testing_score.append(knn.score(X_test,Y_test))
```

In [27]:

```
max_training_score = max(training_score)
train_scores_ind = [i for i,v in enumerate(training_score) if v == max_
```

In [28]:

```
print('Max training score{}% and k ={}'.format(max_training_score*100,l

```

```
Max training score100.0% and k =[1]
```

In [29]:

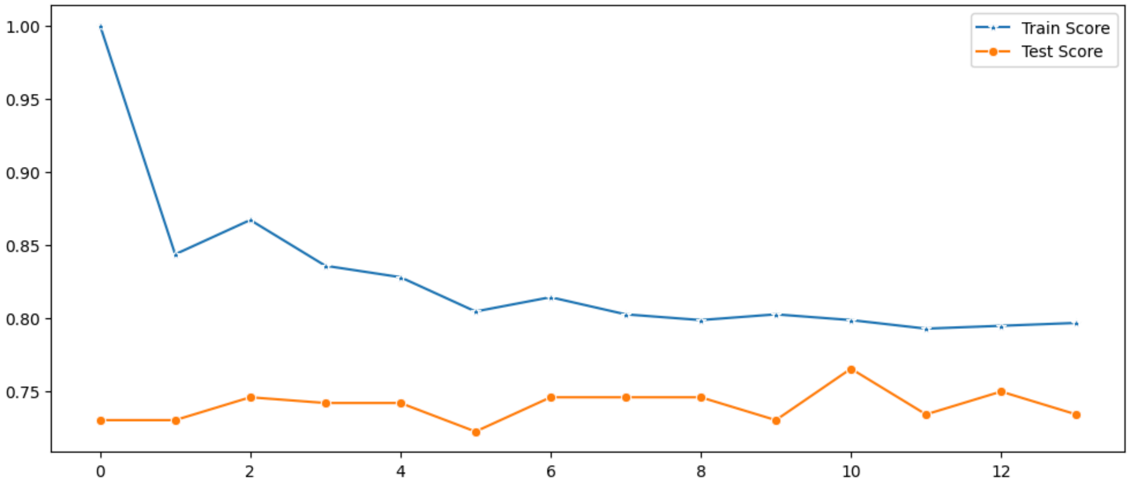
```
max_testing_score = max(testing_score)
test_scores_ind = [i for i,v in enumerate(testing_score) if v == max_te
print('Max training score{}% and k ={}'.format(max_testing_score*100,li
```

```
Max training score76.5625% and k =[11]
```

In [33]:

```
plt.figure(figsize=(12,5))

pplot = sns.lineplot(data=training_score,marker='*',label='Train Score'
pplot = sns.lineplot(data=testing_score,marker='o',label='Test Score')
```



In [34]:

```
knn = KNeighborsClassifier(11)
knn.fit(X_train,Y_train)
knn.score(X_test,Y_test)
```

Out[34]:

```
0.765625
```

In [ ]:

```

```