In [40]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [ ]:
```
Upload Dataset
```

In [41]:
```
 df = pd.read_csv("diabetes.csv")
```

In [63]:
```
# show uploaded dataset
df.head(10)
```

Out[63]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	A
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
5	5	116	74	0	0	25.6	0.201	
6	3	78	50	32	88	31.0	0.248	
10	4	110	92	0	0	37.6	0.191	
11	10	168	74	0	0	38.0	0.537	
14	5	166	72	19	175	25.8	0.587	
16	0	118	84	47	230	45.8	0.551	

In [43]:
```
df.shape
```

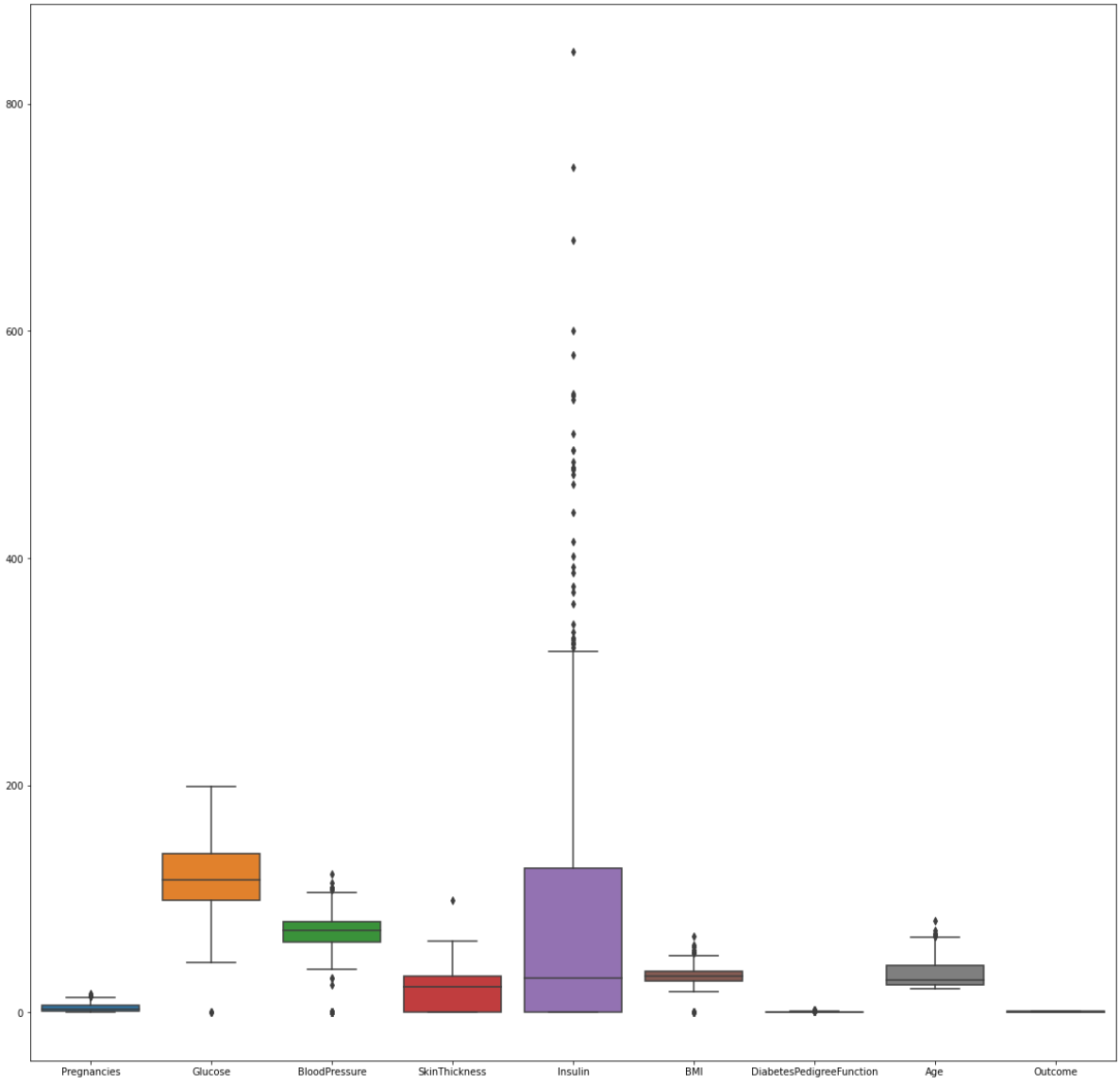
Out[43]:
```
(768, 9)
```

In [ ]:
```
Feature Engineering
```

In [44]:
```
# check null entry of any attribute in dataset
df.isnull().sum()
```

Out[44]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [45]:
```
# check outliear present in dataset
plt.figure(figsize=(20,20))
ax = sns.boxplot(data=df)
```



In [46]:
```
# find Zscore of dataset
from scipy import stats
z = np.abs(stats.zscore(df))
print(z)
```

```
     Pregnancies   Glucose  BloodPressure  SkinThickness   Insulin       BMI  \
0       0.639947  0.848324       0.149641       0.907270  0.692891  0.204013
1       0.844885  1.123396       0.160546       0.530902  0.692891  0.684422
2       1.233880  1.943724       0.263941       1.288212  0.692891  1.103255
3       0.844885  0.998208       0.160546       0.154533  0.123302  0.494043
4       1.141852  0.504055       1.504687       0.907270  0.765836  1.409746
..           ...       ...            ...            ...       ...       ...
763     1.827813  0.622642       0.356432       1.722735  0.870031  0.115169
764     0.547919  0.034598       0.046245       0.405445  0.692891  0.610154
765     0.342981  0.003301       0.149641       0.154533  0.279594  0.735190
766     0.844885  0.159787       0.470732       1.288212  0.692891  0.240205
767     0.844885  0.873019       0.046245       0.656358  0.692891  0.202129

     DiabetesPedigreeFunction       Age   Outcome
0                    0.468492  1.425995  1.365896
1                    0.365061  0.190672  0.732120
2                    0.604397  0.105584  1.365896
3                    0.920763  1.041549  0.732120
4                    5.484909  0.020496  1.365896
..                        ...       ...       ...
763                  0.908682  2.532136  0.732120
764                  0.398282  0.531023  0.732120
765                  0.685193  0.275760  0.732120
766                  0.371101  1.170732  1.365896
767                  0.473785  0.871374  0.732120

[768 rows x 9 columns]
```

In [47]:

```
# find those rows which third SD >3
threshold = 3
print(np.where(z>3))
```

```
(array([  4,   7,   8,   9,  13,  15,  45,  49,  49,  58,  60,  60,  75,
        78,  81,  81,  88, 111, 123, 145, 153, 159, 172, 177, 182, 186,
       193, 220, 222, 228, 228, 247, 261, 266, 269, 286, 298, 300, 330,
       332, 336, 342, 347, 349, 357, 370, 370, 371, 371, 395, 409, 415,
       426, 426, 430, 435, 445, 445, 453, 453, 455, 459, 468, 484, 486,
       494, 494, 502, 522, 522, 533, 535, 579, 584, 589, 593, 601, 604,
       619, 621, 643, 645, 655, 666, 673, 684, 684, 695, 697, 703, 706,
       706, 753], dtype=int64), array([6, 2, 4, 5, 4, 2, 6, 2, 5, 6, 2, 5, 1, 2, 2,
5, 0, 4, 7, 5, 4, 0,
       2, 5, 1, 4, 2, 4, 2, 4, 6, 4, 2, 2, 2, 4, 0, 2, 6, 2, 2, 1, 2, 1,
       2, 4, 6, 5, 6, 6, 4, 4, 2, 5, 2, 2, 5, 6, 2, 7, 0, 7, 2, 2, 4, 2,
       5, 1, 2, 5, 2, 2, 3, 4, 2, 6, 2, 2, 2, 6, 2, 4, 4, 7, 5, 5, 7, 4,
       2, 2, 2, 5, 4], dtype=int64))
```

In [48]:

```
# find interquartile
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3-Q1
print(IQR)
```

```
Pregnancies                 5.0000
Glucose                    41.2500
BloodPressure              18.0000
SkinThickness              32.0000
Insulin                   127.2500
BMI                         9.3000
DiabetesPedigreeFunction    0.3825
Age                        17.0000
Outcome                     1.0000
dtype: float64
```

In [49]:

```
# drop that rows which z>=3
```

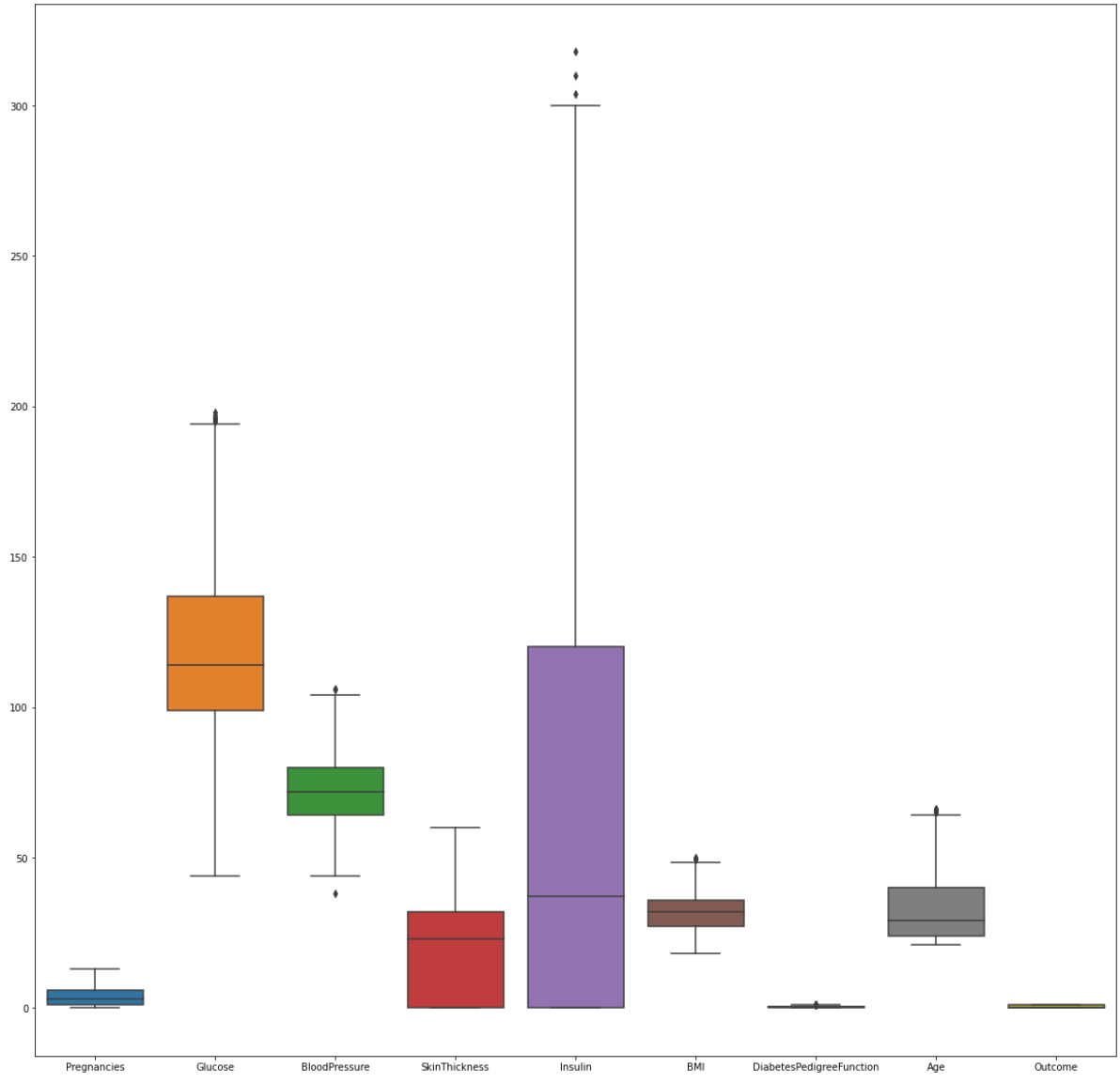
```
df = df[(z<3).all(axis = 1)]
df.shape
```

Out[49]:
```
(688, 9)
```

In [50]:
```
# set lower and upper boundary in dataset
df = df[~((df<(Q1 - 1.5*IQR))|(df>(Q3 + 1.5*IQR))).any(axis = 1)]
df.shape
```

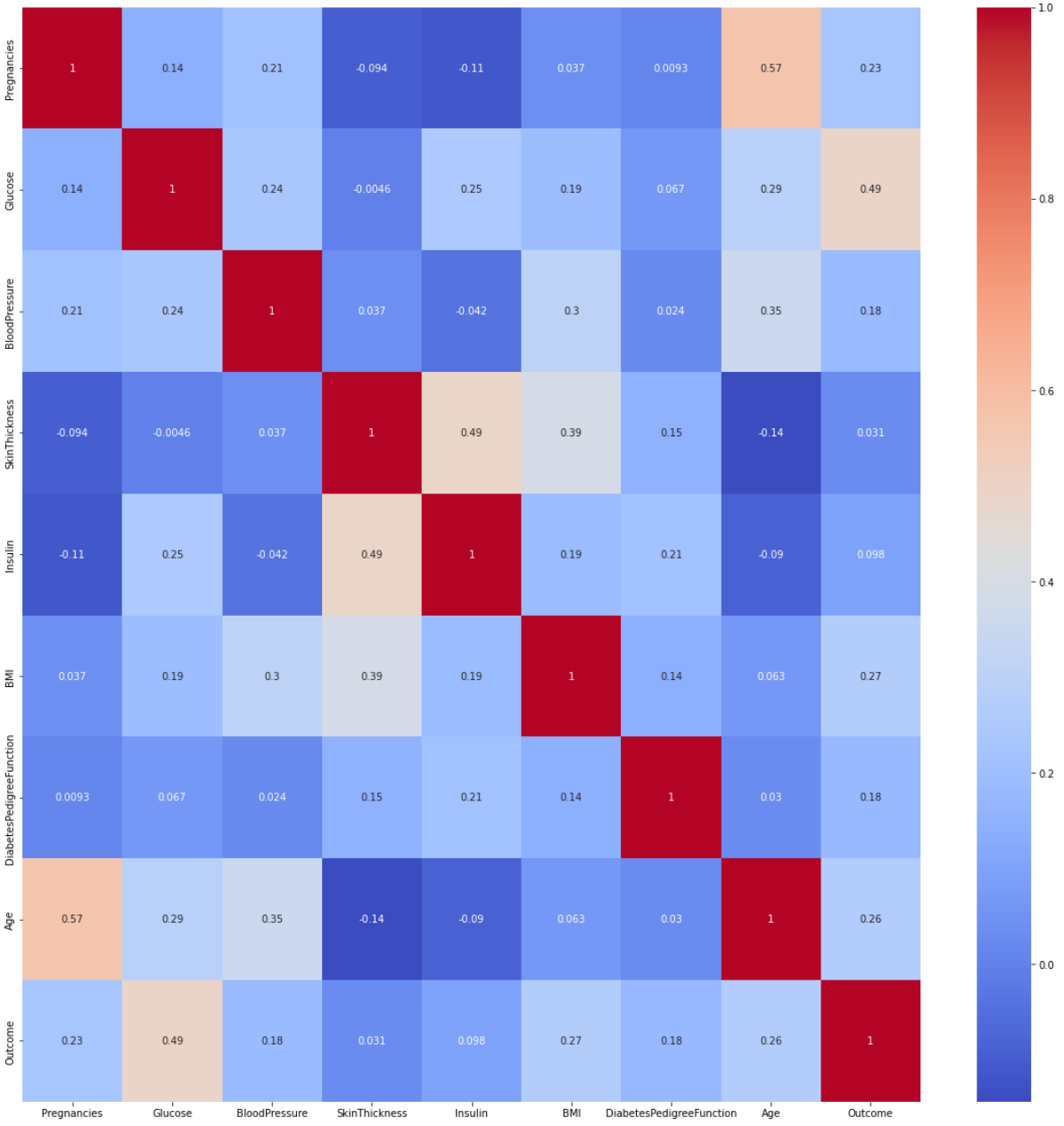
Out[50]:
```
(639, 9)
```

In [51]:
```
# check outliear present in dataset
plt.figure(figsize=(20,20))
ax = sns.boxplot(data=df)
```



In [ ]:
```
Feature Selection
```

In [56]:
```
# find correlations between attributes
plt.figure(figsize=(20,20))
d = sns.heatmap(df.corr(),cmap="coolwarm",annot=True)
```



In [ ]:
```
# if you wish to drop any coloum(attribute) from dataset
# df = df.drop(columms = "chol")
# df.head()
```

In [58]:
```
# statistical observation of various attributes in dataset
df.describe()
```

Out[58]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigr
count	639.000000	639.000000	639.000000	639.000000	639.000000	639.00000	
mean	3.804382	119.112676	72.120501	20.563380	65.931142	32.00579	
std	3.260995	29.162175	11.348686	15.339991	79.569482	6.43397	
min	0.000000	44.000000	38.000000	0.000000	0.000000	18.20000	
25%	1.000000	99.000000	64.000000	0.000000	0.000000	27.30000	
50%	3.000000	114.000000	72.000000	23.000000	37.000000	32.00000	
75%	6.000000	137.000000	80.000000	32.000000	120.000000	35.95000	
max	13.000000	198.000000	106.000000	60.000000	318.000000	50.00000	

In [ ]:
```
Visualisation
```

In [ ]:
```
sns.pairplot(df, hue = "target", height = 3, aspect =1)
```

In [ ]:
```

```