In [1]:

```
!pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\hp\anaconda3\lib\site-
packages (1.7.5)
Requirement already satisfied: scipy in c:\users\hp\anaconda3\lib\site-pa
ckages (from xgboost) (1.9.1)
Requirement already satisfied: numpy in c:\users\hp\anaconda3\lib\site-pa
ckages (from xgboost) (1.23.5)
```

In [2]:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
import xgboost as xgb
import numpy as np
```

In [3]:

```
df = pd.read_csv("C:\/Users/HP/Downloads/Oxygen Dataset Final.csv")
df.head()
```

Out[3]:

	age	gender	spo2	pr	c/nc	oxy_flow
0	27	0	74.0	72.0	1.0	6.0
1	53	1	NaN	110.0	NaN	28.0
2	56	0	99.0	98.0	1.0	NaN
3	26	1	NaN	110.0	1.0	4.0
4	52	0	69.0	84.0	1.0	0.0

In [4]:

```
df.shape
```

Out[4]:

```
(200000, 6)
```

In [5]:

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

Out[5]:

```
age             0
gender          0
spo2        26245
pr          32384
c/nc        26442
oxy_flow    37747
dtype: int64
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 6 columns):
 #   Column    Non-Null Count   Dtype  
---  ------    --------------   -----  
 0   age       200000 non-null  int64  
 1   gender    200000 non-null  int64  
 2   spo2      173755 non-null  float64
 3   pr        167616 non-null  float64
 4   c/nc      173558 non-null  float64
 5   oxy_flow  162253 non-null  float64
dtypes: float64(4), int64(2)
memory usage: 9.2 MB
```

In [7]:

```
np.unique(df['spo2'])
```

Out[7]:

```
array([35., 36., 37., 38., 39., 40., 41., 42., 43., 44., 45., 46., 47.,
       48., 49., 50., 51., 52., 53., 54., 55., 56., 57., 58., 59., 60.,
       61., 62., 63., 64., 65., 66., 67., 68., 69., 70., 71., 72., 73.,
       74., 75., 76., 77., 78., 79., 80., 81., 82., 83., 84., 85., 86.,
       87., 88., 89., 90., 91., 92., 93., 94., 95., 96., 97., 98., 99.,
       nan])
```

In [8]:

```
np.unique(df['pr'])
```

Out[8]:

```
array([ 40.,  41.,  42.,  43.,  44.,  45.,  46.,  47.,  48.,  49.,  50.,
        51.,  52.,  53.,  54.,  55.,  56.,  57.,  58.,  59.,  60.,  61.,
        62.,  63.,  64.,  65.,  66.,  67.,  68.,  69.,  70.,  71.,  72.,
        73.,  74.,  75.,  76.,  77.,  78.,  79.,  80.,  81.,  82.,  83.,
        84.,  85.,  86.,  87.,  88.,  89.,  90.,  91.,  92.,  93.,  94.,
        95.,  96.,  97.,  98.,  99., 100., 101., 102., 103., 104., 105.,
       106., 107., 108., 109., 110.,  nan])
```

In [9]:

```
np.unique(df['c/nc'])
```

Out[9]:

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

In [10]:

```
np.unique(df['oxy_flow'])
```

Out[10]:

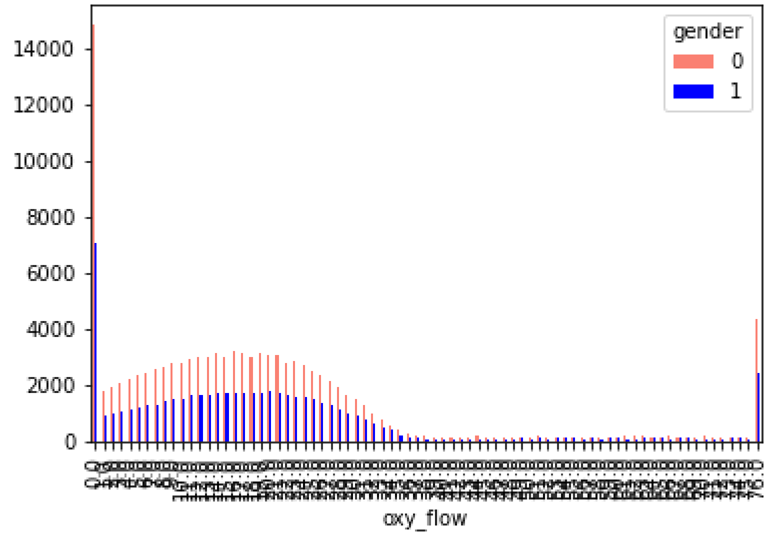
```
array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10., 11., 12.,
       13., 14., 15., 16., 17., 18., 19., 20., 21., 22., 23., 24., 25.,
       26., 27., 28., 29., 30., 31., 32., 33., 34., 35., 36., 37., 38.,
       39., 40., 41., 42., 43., 44., 45., 46., 47., 48., 49., 50., 51.,
       52., 53., 54., 55., 56., 57., 58., 59., 60., 61., 62., 63., 64.,
       65., 66., 67., 68., 69., 70., 71., 72., 73., 74., 75., 76., nan])
```

In [11]:

```
pd.crosstab(df['oxy_flow'],df['gender']).plot(kind='bar' , color=['salmon','blue'])
```

Out[11]:

```
<AxesSubplot:xlabel='oxy_flow'>
```

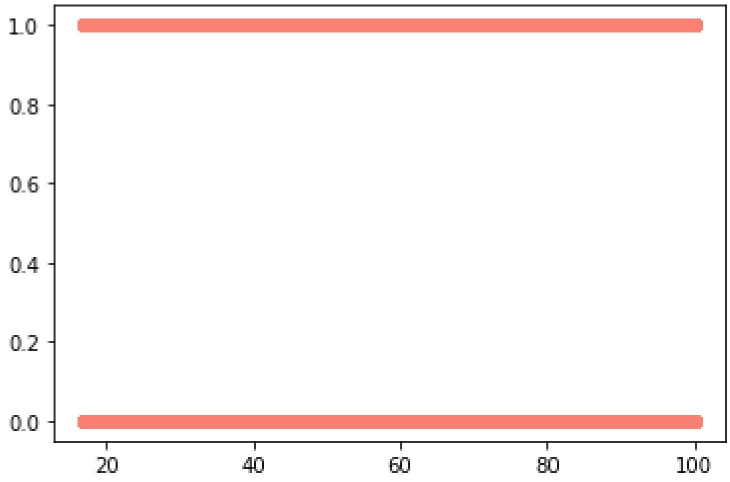


In [12]:

```
plt.scatter(df['age'],df['c/nc'],c='salmon')
```

Out[12]:

```
<matplotlib.collections.PathCollection at 0x1dcc9335a90>
```



In [13]:

```
df.columns
```

Out[13]:

```
Index(['age', 'gender', 'spo2', 'pr', 'c/nc', 'oxy_flow'], dtype='objec
t')
```

In [14]:

```
df.duplicated().sum()
```

Out[14]:

```
23486
```

In [15]:

```
df.drop_duplicates()
```

Out[15]:

	age	gender	spo2	pr	c/nc	oxy_flow
0	27	0	74.0	72.0	1.0	6.0
1	53	1	NaN	110.0	NaN	28.0
2	56	0	99.0	98.0	1.0	NaN
3	26	1	NaN	110.0	1.0	4.0
4	52	0	69.0	84.0	1.0	0.0
...	...	...	...	...	...	...
199992	47	1	96.0	89.0	1.0	16.0
199993	76	0	99.0	95.0	1.0	19.0
199996	48	1	99.0	NaN	1.0	5.0
199998	100	1	99.0	95.0	1.0	25.0
199999	22	1	99.0	82.0	0.0	32.0

176514 rows × 6 columns

In [16]:

```
df2 = df.dropna()
```

In [17]:

```
df2.describe()
```

Out[17]:

	age	gender	spo2	pr	c/nc	oxy_flow
count	98925.000000	98925.000000	98925.000000	98925.000000	98925.000000	98925.000000
mean	46.023644	0.322103	88.614577	92.515188	0.786919	18.581582
std	21.820753	0.467284	15.537629	16.025638	0.409486	17.887996
min	17.000000	0.000000	35.000000	40.000000	0.000000	0.000000
25%	28.000000	0.000000	84.000000	82.000000	1.000000	6.000000
50%	43.000000	0.000000	97.000000	96.000000	1.000000	16.000000
75%	61.000000	1.000000	99.000000	107.000000	1.000000	24.000000
max	100.000000	1.000000	99.000000	110.000000	1.000000	76.000000

In [18]:

```
df2.nunique()
```

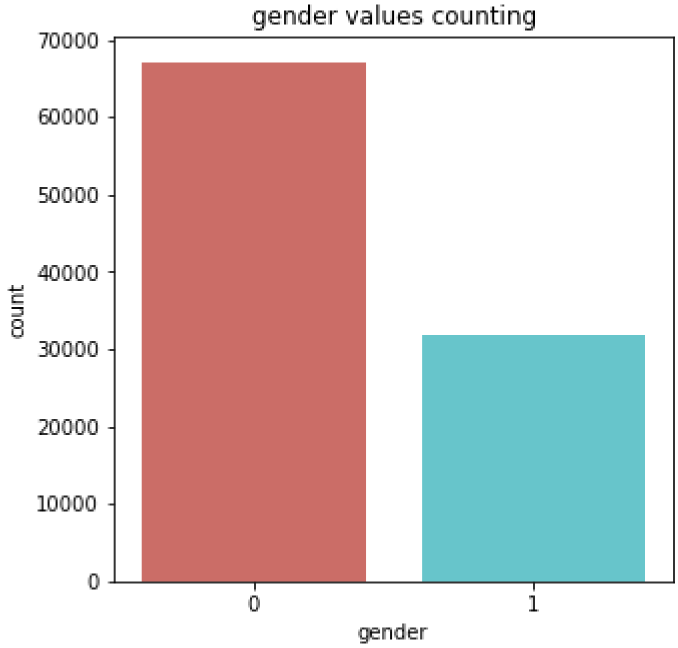
Out[18]:

```
age          84
gender        2
spo2         65
pr           71
c/nc          2
oxy_flow     77
dtype: int64
```

Visualizations

In [19]:

```
plt.figure(figsize=(5,5))
plt.title('gender values counting')
sns.countplot(x='gender',data=df2,palette='hls')
plt.show()
```



In [20]:

```
df2['c/nc'].unique()
```

Out[20]:

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

In [21]:

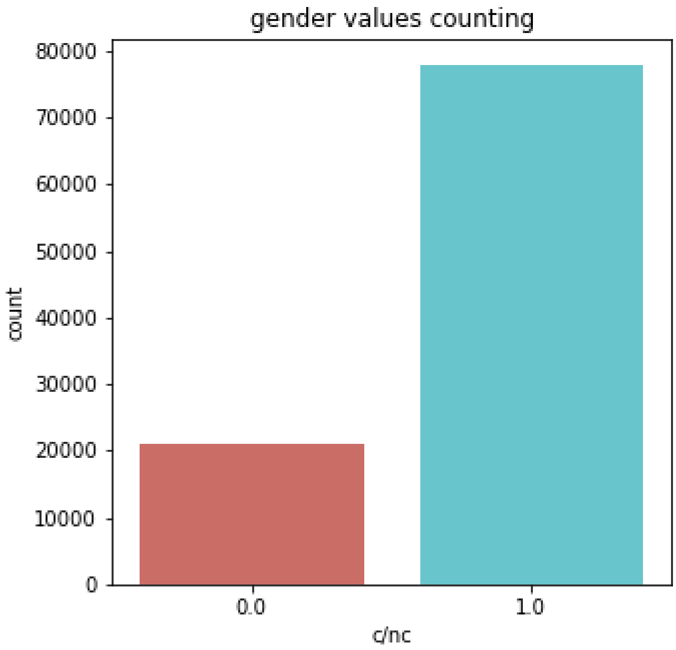
```
df2['c/nc'].value_counts()
```

Out[21]:

```
1.0    77846
0.0    21079
Name: c/nc, dtype: int64
```

In [22]:

```
plt.figure(figsize=(5,5))
plt.title('gender values counting')
sns.countplot(x='c/nc',data=df2,palette='hls')
plt.show()
```



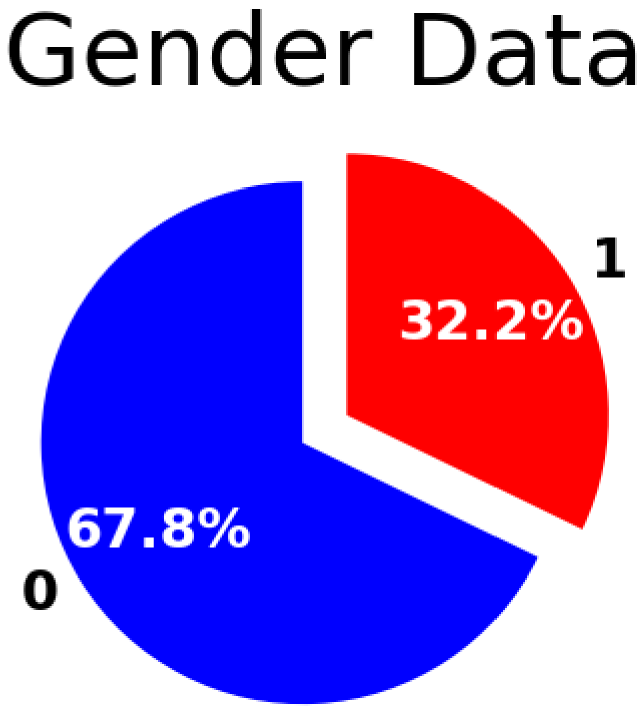
In [23]:

```
gender_data = df2['gender'].value_counts()
explode = (0.1,0.1)
plt.figure(figsize=(6,6))
patches,texts,pcts=plt.pie(gender_data,
                          labels=gender_data.index,
                          colors=['blue','red'],
                          pctdistance=0.65,
                          explode=explode,
                           startangle=90,
                           autopct='%1.1f%%',
                           textprops={'color': 'black',
                                     'weight':'bold',
                                     'fontsize': 27}
                          )

plt.setp(pcts,color='white')
plt.title('Gender Data',size=50)
```

Out[23]:

```
Text(0.5, 1.0, 'Gender Data')
```



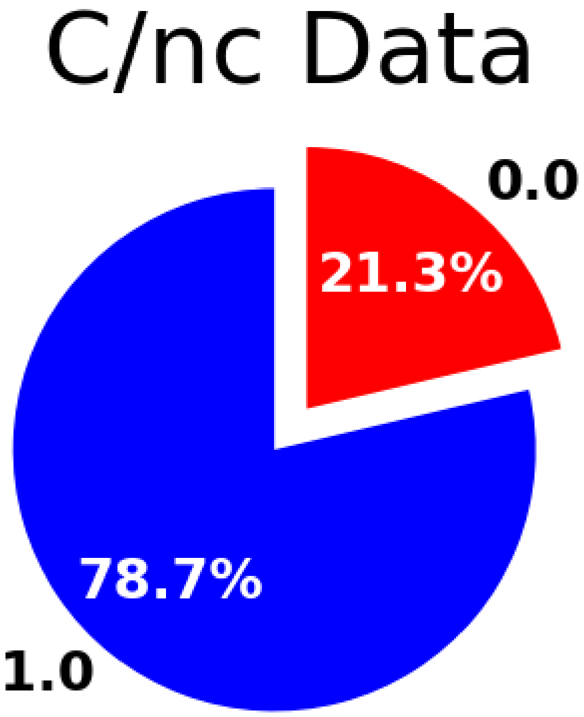
In [24]:

```
cnc_data = df2['c/nc'].value_counts()
explode = (0.1,0.1)
plt.figure(figsize=(6,6))
patches,texts,pcts=plt.pie(cnc_data,
                          labels=cnc_data.index,
                          colors=['blue','red'],
                          pctdistance=0.65,
                          explode=explode,
                           startangle=90,
                           autopct='%1.1f%%',
                           textprops={'color': 'black',
                                     'weight':'bold',
                                     'fontsize': 27}
                          )

plt.setp(pcts,color='white')
plt.title('C/nc Data',size=50)
```

Out[24]:

```
Text(0.5, 1.0, 'C/nc Data')
```

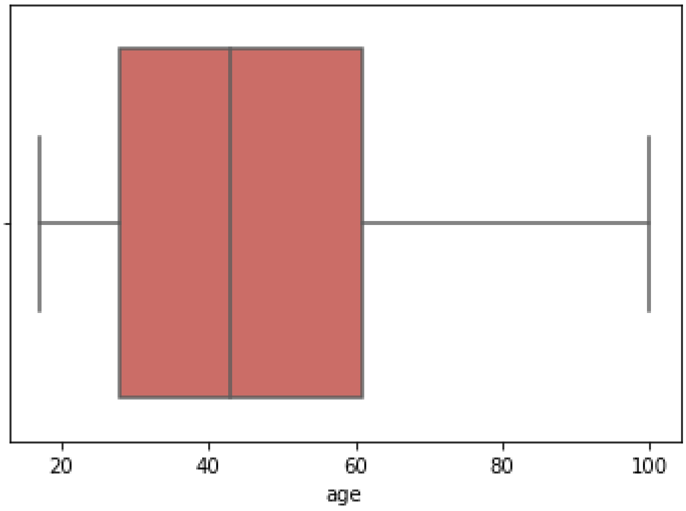


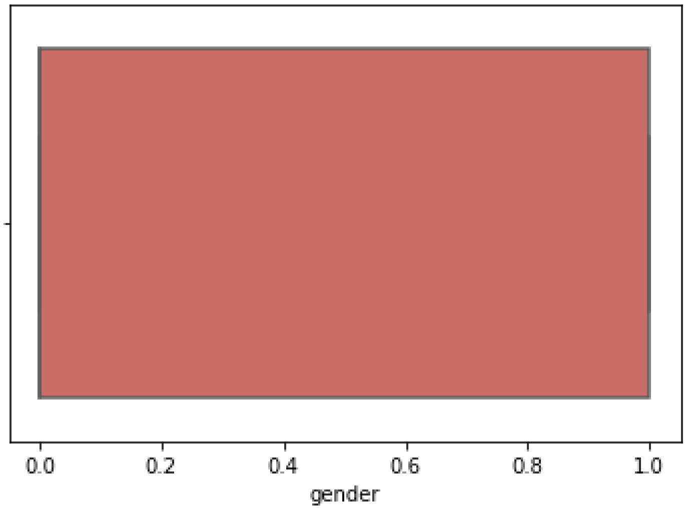
In [25]:

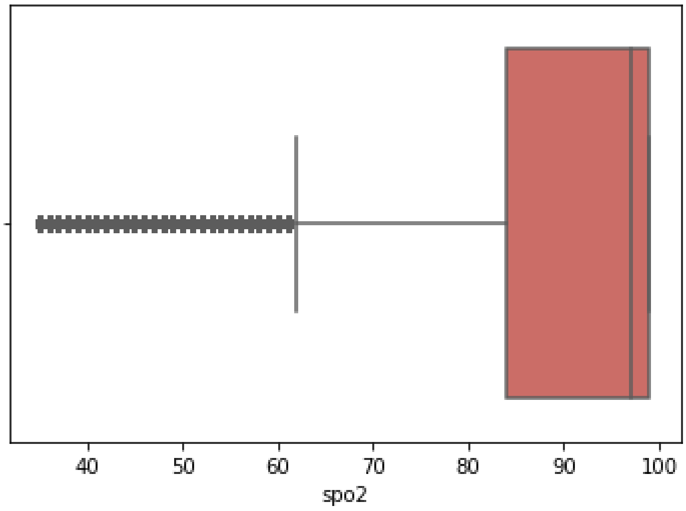
```
import warnings
warnings.filterwarnings('ignore')
```

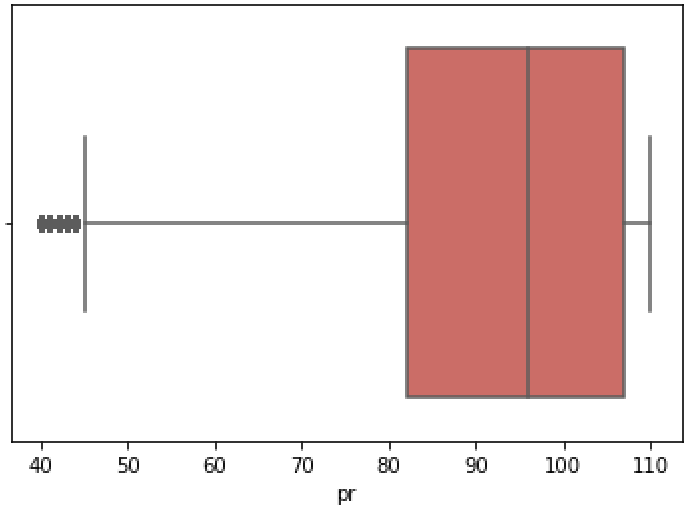
In [26]:

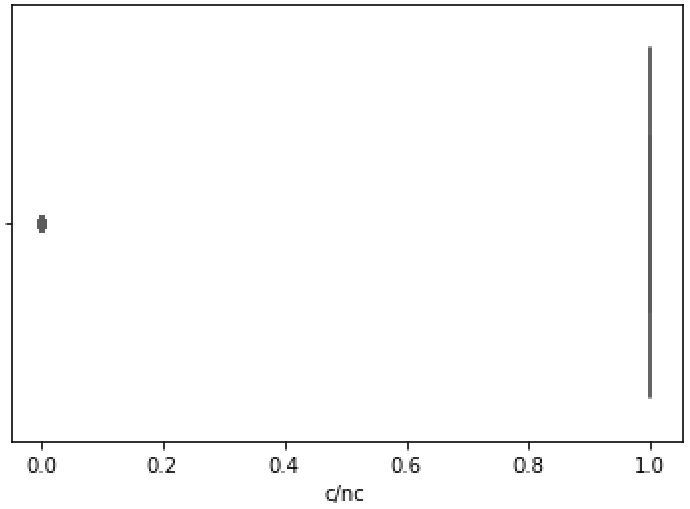
```
for  i in df2.columns:
    sns.boxplot(df2[i],palette='hls')
    plt.show()
```

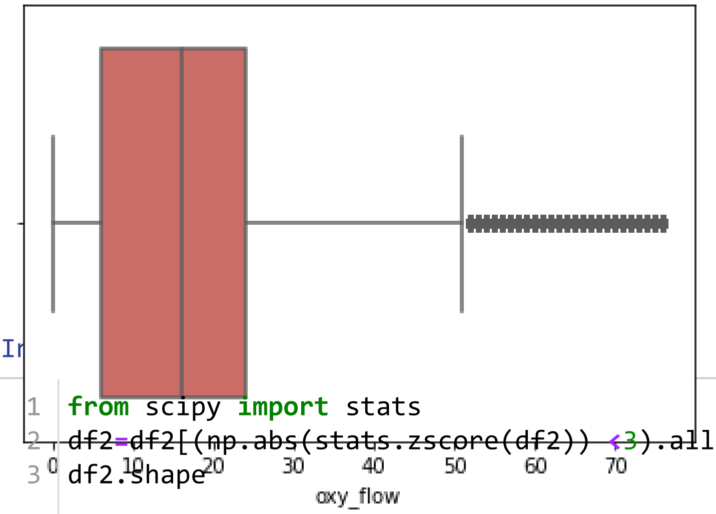












In

```
from scipy import stats
df2=df2[(np.abs(stats.zscore(df2)) <3).all(axis=1)]
df2.shape
```

Out[27]:

```
(91233, 6)
```

In [28]:

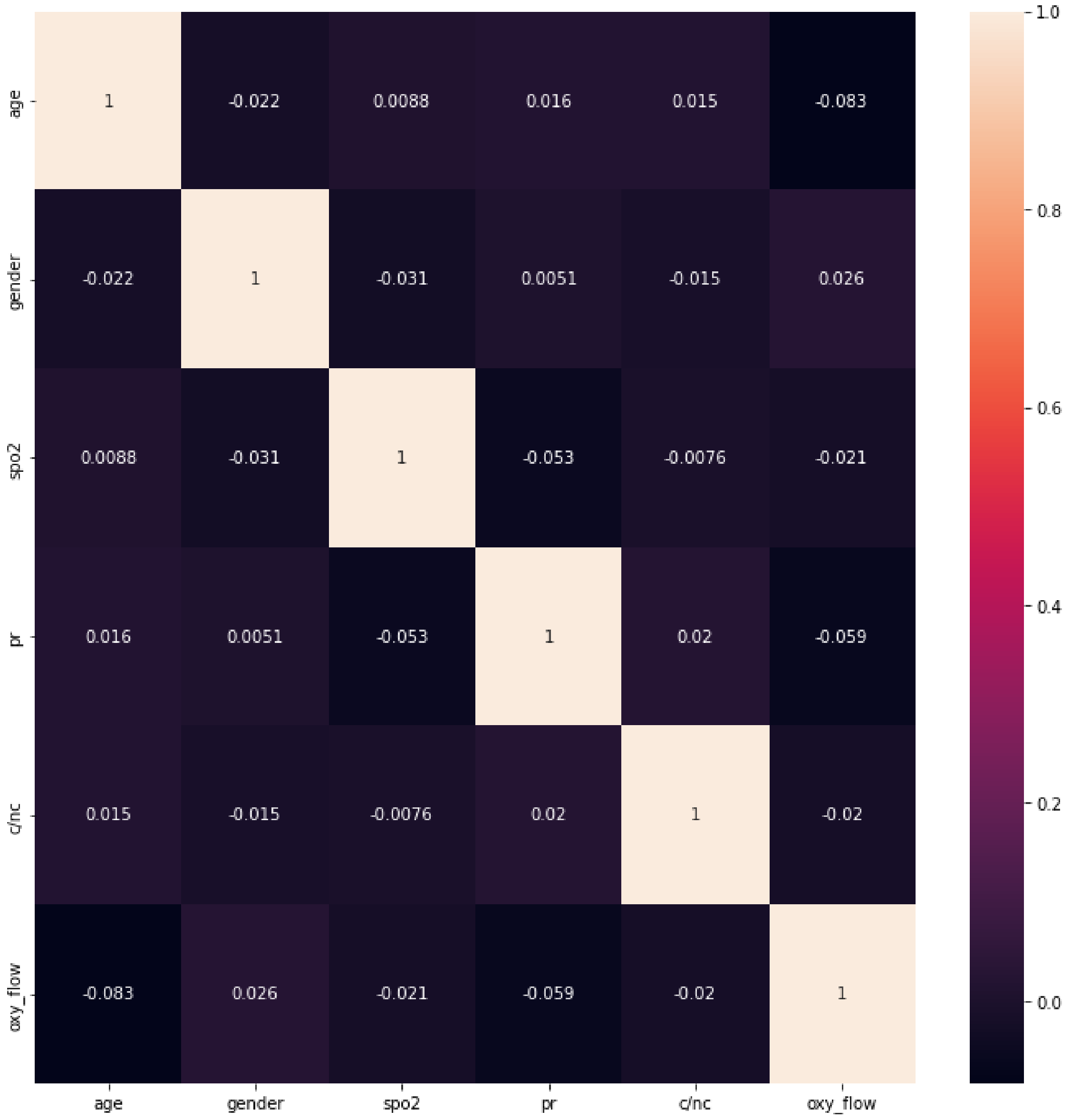
```
df2.head()
```

Out[28]:

	age	gender	spo2	pr	c/nc	oxy_flow
0	27	0	74.0	72.0	1.0	6.0
4	52	0	69.0	84.0	1.0	0.0
5	82	0	93.0	93.0	1.0	28.0
9	68	0	90.0	92.0	1.0	33.0
13	40	0	99.0	109.0	1.0	27.0

In [29]:

```
df_corr = df2.corr()
plt.figure(figsize=(12,12))
sns.heatmap(df_corr,annot=True)
plt.show()
```



In [30]:

```
x = df2.drop('oxy_flow',axis=1)
y = df2['oxy_flow']
```

In [31]:

```
from sklearn import preprocessing
scaler = preprocessing.StandardScaler()
x= scaler.fit_transform(x)
```

In [32]:

```
xtrain , xtest,ytrain,ytest = train_test_split(x,y , test_size=0.3)
xtrain.shape , xtest.shape ,ytrain.shape , ytest.shape
```

Out[32]:

```
((63863, 5), (27370, 5), (63863,), (27370,))
```

In [33]:

```
from sklearn.linear_model import LinearRegression
LR = LinearRegression()
LR.fit(xtrain,ytrain)
```

Out[33]:

```
LinearRegression()
```

In [34]:

```
ypred = LR.predict(xtest)
```

In [35]:

```
from sklearn.metrics import mean_absolute_error , mean_squared_error , r2_score
```

In [36]:

```
print('MAE',mean_absolute_error(ytest,ypred))
print('MSE',mean_squared_error(ytest,ypred))
print('RMSE',np.sqrt(mean_squared_error(ytest,ypred)))
```

```
MAE 9.885767702501937
MSE 169.9939685185366
RMSE 13.038173511598034
```

In [39]:

```
xgb_1 = xgb.XGBRegressor(n_estimators=10,
                          seed=123,
                          objective='reg:linear')
xgb_1.fit(xtrain,ytrain)
```

```
[01:02:25] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autos
caling-group-i-07593ffd91cd9da33-1\xgboost\xgboost-ci-windows\src\objecti
ve\regression_obj.cu:213: reg:linear is now deprecated in favor of reg:sq
uarederror.
```

Out[39]:

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=No
ne,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=N
one,
             interaction_constraints=None, learning_rate=None, max_bin=No
ne,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=Non
e,
             n_estimators=10, n_jobs=None, num_parallel_tree=None,
             objective='reg:linear', predictor=None, ...)
```

In [41]:

```
y_pred = xgb_1.predict(xtest)
```

In [42]:

```
print('MAE',mean_absolute_error(ytest,y_pred))
print('MSE',mean_squared_error(ytest,y_pred))
print('RMSE',np.sqrt(mean_squared_error(ytest,y_pred)))
```

```
MAE 9.880871639906772
MSE 170.8402491017598
RMSE 13.070587175095074
```