In [34]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import train_test_split
```

In [35]:

```
df = pd.read_csv("C:/Users/HP/Downloads/breast_canceer.csv")
df
```

Out[35]:

| | id | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothı |
|---|---|---|---|---|---|---|---|
| **0** | 842302 | M | 17.99 | 10.38 | 122.80 | 1001.0 | |
| **1** | 842517 | M | 20.57 | 17.77 | 132.90 | 1326.0 | |
| **2** | 84300903 | M | 19.69 | 21.25 | 130.00 | 1203.0 | |
| **3** | 84348301 | M | 11.42 | 20.38 | 77.58 | 386.1 | |
| **4** | 84358402 | M | 20.29 | 14.34 | 135.10 | 1297.0 | |
| **...** | ... | ... | ... | ... | ... | ... | |
| **564** | 926424 | M | 21.56 | 22.39 | 142.00 | 1479.0 | |
| **565** | 926682 | M | 20.13 | 28.25 | 131.20 | 1261.0 | |
| **566** | 926954 | M | 16.60 | 28.08 | 108.30 | 858.1 | |
| **567** | 927241 | M | 20.60 | 29.33 | 140.10 | 1265.0 | |
| **568** | 92751 | B | 7.76 | 24.54 | 47.92 | 181.0 | |

569 rows × 33 columns

In [36]:

```
1 df.isna().sum()
```

Out[36]:

```
id                          0
diagnosis                   0
radius_mean                 0
texture_mean                0
perimeter_mean              0
area_mean                   0
smoothness_mean             0
compactness_mean            0
concavity_mean              0
concave points_mean         0
symmetry_mean               0
fractal_dimension_mean      0
radius_se                   0
texture_se                  0
perimeter_se                0
area_se                     0
smoothness_se               0
compactness_se              0
concavity_se                0
concave points_se           0
symmetry_se                 0
fractal_dimension_se        0
radius_worst                0
texture_worst               0
perimeter_worst             0
area_worst                  0
smoothness_worst            0
compactness_worst           0
concavity_worst             0
concave points_worst        0
symmetry_worst              0
fractal_dimension_worst     0
Unnamed: 32               569
dtype: int64
```

In [37]:

```
1 df.dropna(inplace=True,axis=1)
```

In [38]:

```
# here we are label encoding M=1 and B=0 values ,
from sklearn.preprocessing import LabelEncoder
labelencoder_ = LabelEncoder()
df['diagnosis'] = labelencoder_.fit_transform(df['diagnosis'])
df.head()
```

Out[38]:

| | id | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothnes |
|---|---|---|---|---|---|---|---|
| **0** | 842302 | 1 | 17.99 | 10.38 | 122.80 | 1001.0 | |
| **1** | 842517 | 1 | 20.57 | 17.77 | 132.90 | 1326.0 | |
| **2** | 84300903 | 1 | 19.69 | 21.25 | 130.00 | 1203.0 | |
| **3** | 84348301 | 1 | 11.42 | 20.38 | 77.58 | 386.1 | |
| **4** | 84358402 | 1 | 20.29 | 14.34 | 135.10 | 1297.0 | |

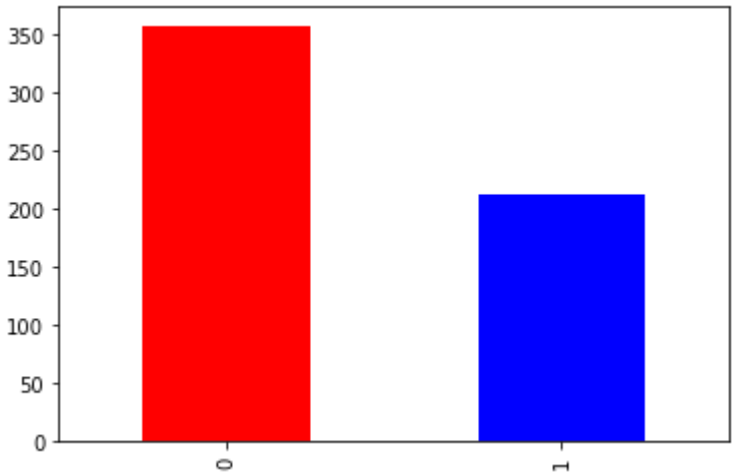
5 rows × 32 columns

In [39]:

```
# lets find count of M , B
df['diagnosis'].value_counts().plot(kind='bar' , color=['red','blue'])
```

Out[39]:

```
<AxesSubplot:>
```

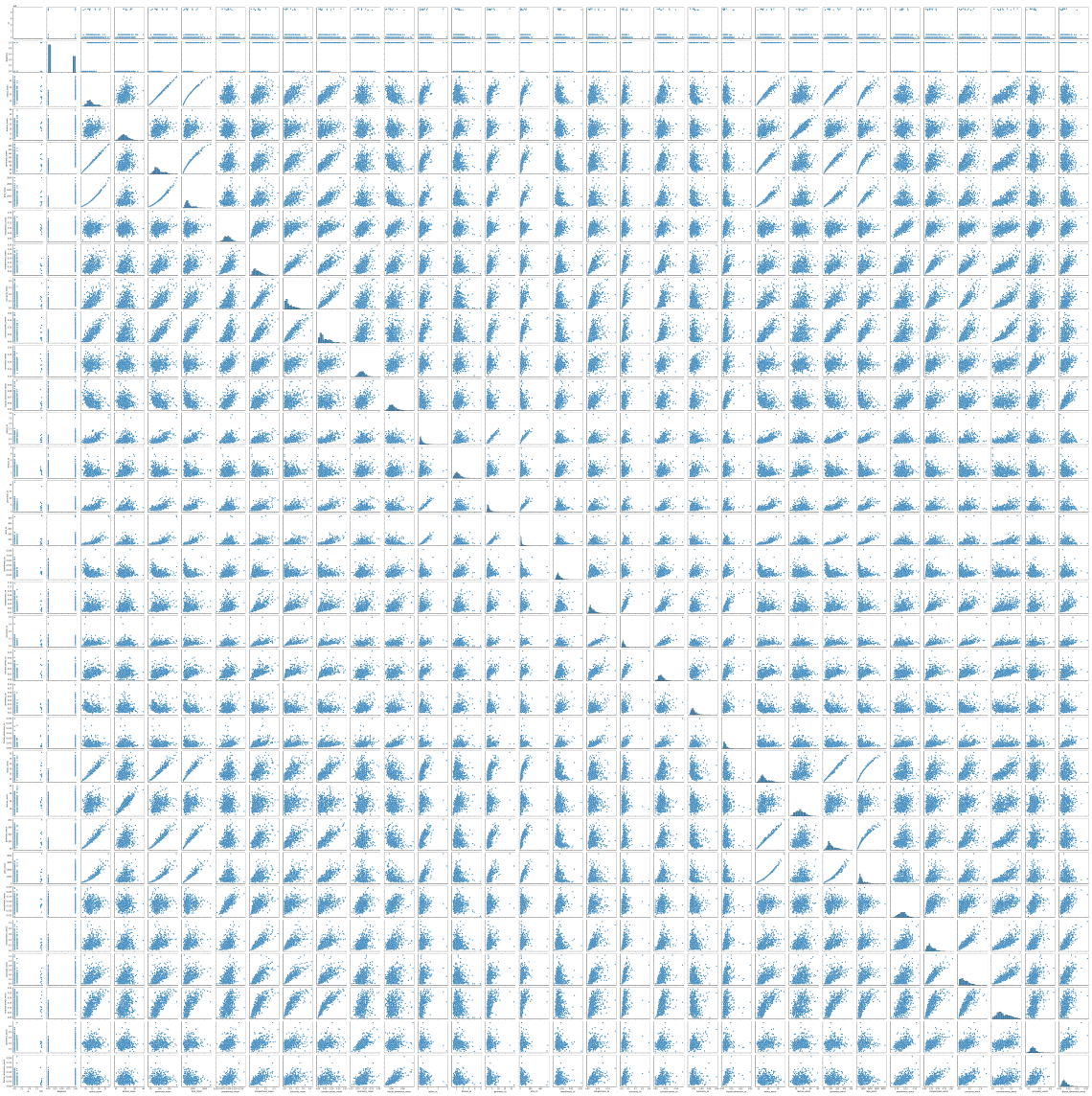


In [40]:

```
1 sns.pairplot(df)
```

Out[40]:

```
<seaborn.axisgrid.PairGrid at 0x18fa07b67f0>
```

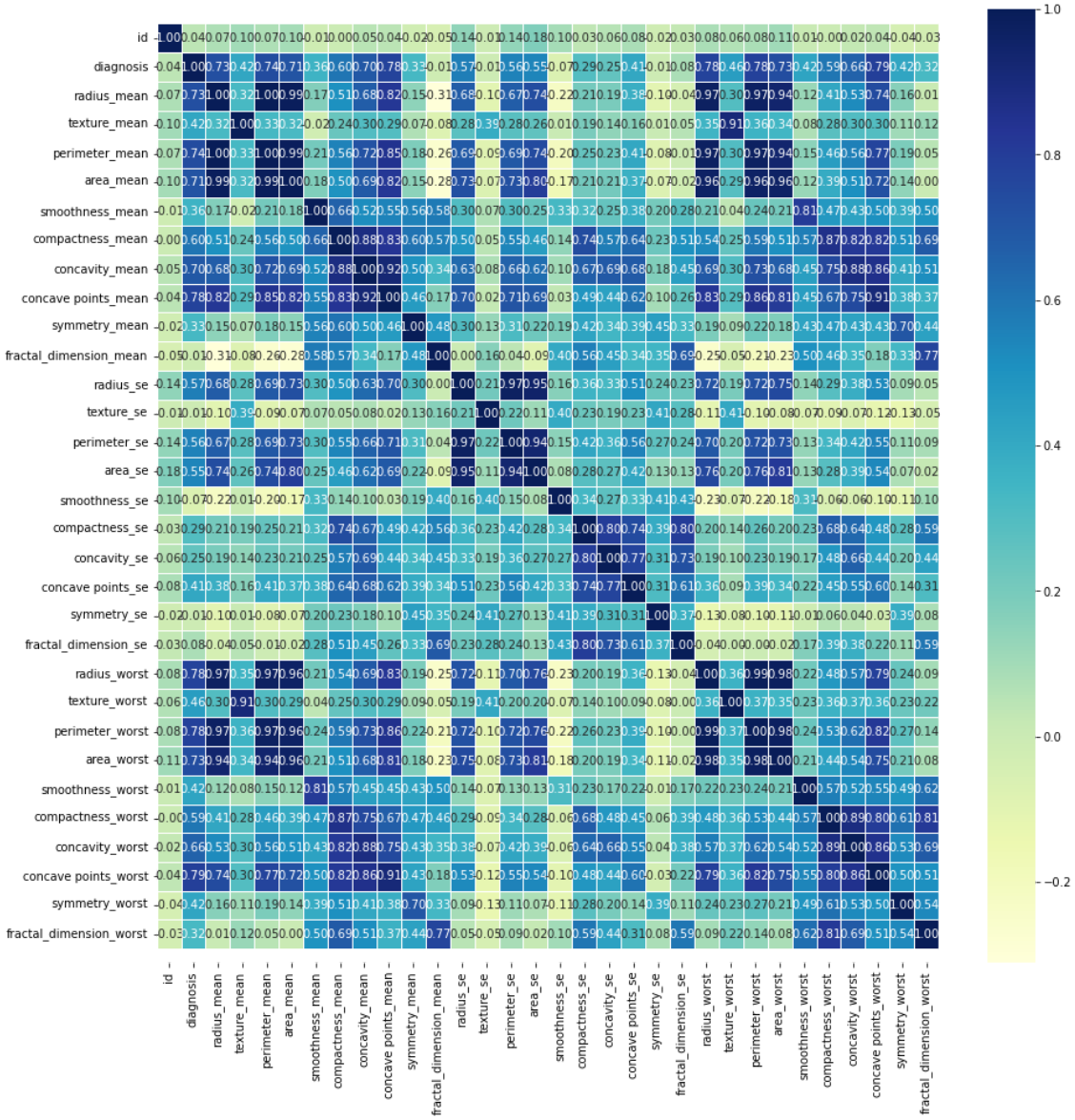


In [41]:

```
# finding correrelation matrix
corr_matrix = df.corr()
fig,ax = plt.subplots(figsize=(15,15))
ax = sns.heatmap(corr_matrix,annot = True, linewidths =0.5,
                 cmap="YlGnBu", fmt = ".2f")
bottom , top = ax.get_ylim()
ax.set_ylim(bottom + 0.5 ,top -0.5 )
```

Out[41]:

(32.5, -0.5)




In [42]:

```
# Data preparation for model training
x = df.drop("diagnosis" , axis=1)

y= df["diagnosis"]
```

In [43]:

```
1 x.head()
```

Out[43]:

| | id | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | c |
|---|---|---|---|---|---|---|---|
| **0** | 842302 | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | |
| **1** | 842517 | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | |
| **2** | 84300903 | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | |
| **3** | 84348301 | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | |
| **4** | 84358402 | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | |

5 rows × 31 columns

In [44]:

```
1 y.head()
```

Out[44]:

```
0    1
1    1
2    1
3    1
4    1
Name: diagnosis, dtype: int32
```

In [45]:

```
1 #performing train_test split
2 xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.3)
```

# Model training

In [ ]:

```
1 Decison Tree
2 Adaboost algo
```

In [61]:

```
1 DT = DecisionTreeClassifier()
2 DT.fit(xtrain,ytrain)
3 ypred=DT.predict(xtest)
4 score= accuracy_score(ytest,ypred)
5 print(score)
```

```
0.9532163742690059
```

In [ ]:

```
# using adaboost for accuracy score checking
```

In [67]:

```
abc = AdaBoostClassifier(n_estimators=100, learning_rate=1, random_state=0)
abc.fit(xtrain,ytrain)
y_pred=abc.predict(xtest)
score1 = accuracy_score(ytest,y_pred)
print(score1)
```

```
0.9649122807017544
```