In [62]:

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

In [63]:

```
df=pd.read_csv("/Users/bob/Downloads/8_BreastCancerPrediction.csv")
df.fillna(0,inplace=True)
df
```

Out[63]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smootl
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 [64]:

```
df.head()
```

Out[64]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
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	

5 rows × 33 columns

In [65]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              569 non-null    float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

In [66]:

```
df.columns
```

Out[66]:

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimete
r_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concav
ity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_
mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoo
thness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'sym
metry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_wor
st',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 3
2'],
      dtype='object')
```

In [*]:

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

In [*]:

```
sns.displot(df['fractal_dimension_worst'])
```

In [52]:

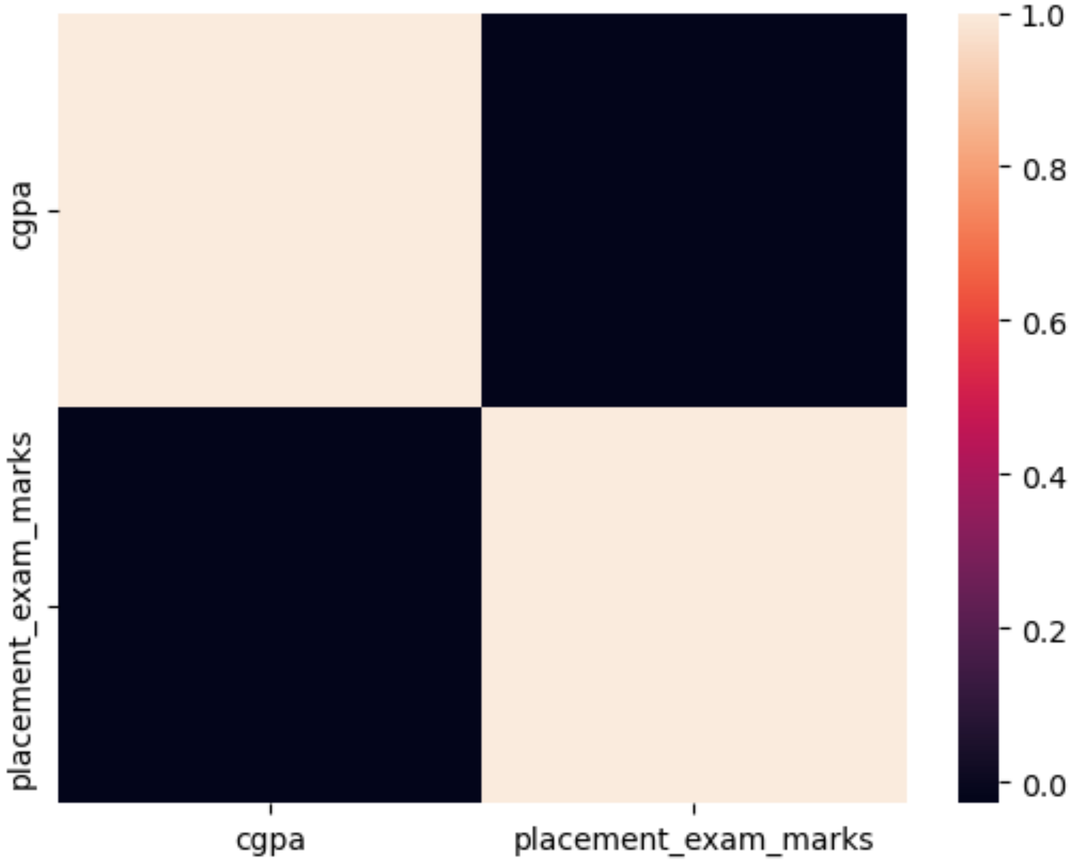
```
df1=df.drop(['placed'],axis=1)
df1
df1=df1.drop(df1.index[1537:])
df1.isna().sum()
```

Out[52]:

```
cgpa                   0
placement_exam_marks   0
dtype: int64
```

In [53]:
```
sns.heatmap(df1.corr())
```

Out[53]:
```
<Axes: >
```



In [54]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [55]:
```
df1.isna().sum()
```

Out[55]:
```
cgpa                    0
placement_exam_marks    0
dtype: int64
```

In [56]:

```
y=df1['placement_exam_marks']
x=df1.drop(['placement_exam_marks'],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

```
     cgpa
206  7.50
191  6.63
702  6.24
467  6.65
507  6.77
..    ...
277  6.45
710  7.56
587  7.58
441  7.80
74   6.16

[700 rows x 1 columns]
```

In [57]:

```
model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

Out[57]:

```
36.78350248626957
```

In [58]:

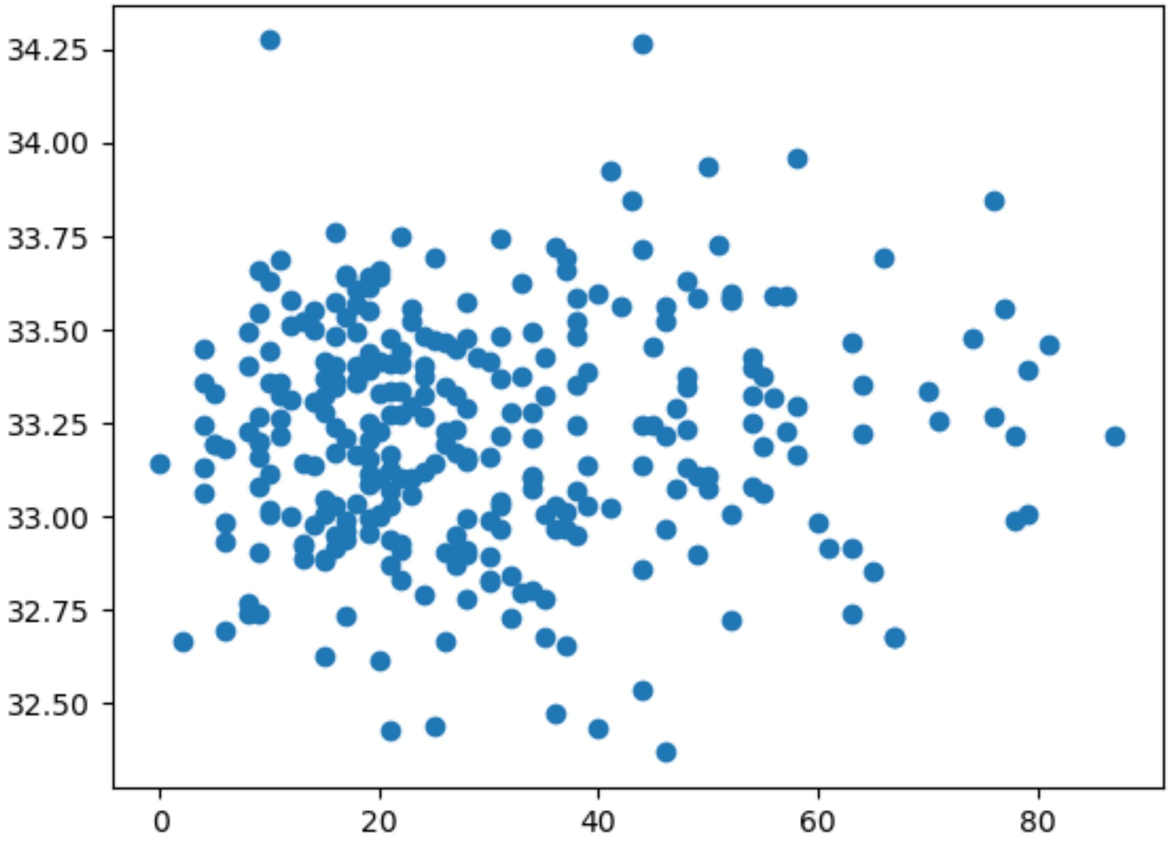
```
model.coef_
```

Out[58]:

```
array([-0.5123083])
```

In [59]:
```
prediction=model.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[59]:
```
<matplotlib.collections.PathCollection at 0x7f8ef4ba2fb0>
```



In [60]:
```
model.score(x_test,y_test)
```

Out[60]:
```
-0.031875624215878195
```

In [ ]:
```

```