In [1]:
```
import random
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

In [2]:
```
#access dataset
df=pd.read_csv("C:\\Users\\Administrator\\Downloads\\archive (3)\\Breast_cancer_data.csv")
df
```

Out[2]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
0	17.99	10.38	122.80	1001.0	0.11840	0
1	20.57	17.77	132.90	1326.0	0.08474	0
2	19.69	21.25	130.00	1203.0	0.10960	0
3	11.42	20.38	NaN	386.1	0.14250	0
4	20.29	14.34	NaN	1297.0	0.10030	0
...	...	...	...	...	...	...
564	NaN	22.39	142.00	NaN	0.11100	0
565	NaN	28.25	131.20	NaN	0.09780	0
566	16.60	28.08	108.30	NaN	0.08455	0
567	20.60	29.33	140.10	NaN	0.11780	0
568	7.76	24.54	47.92	NaN	0.05263	1

569 rows × 6 columns

In [3]:
```
#check the null value in dataset
df.isnull().sum()
```

Out[3]:
```
mean_radius          23
mean_texture         51
mean_perimeter       39
mean_area            29
mean_smoothness     440
diagnosis             0
dtype: int64
```

In [24]:
```
#fill null value using mean
df1=df.fillna(df.mean())
df1
```

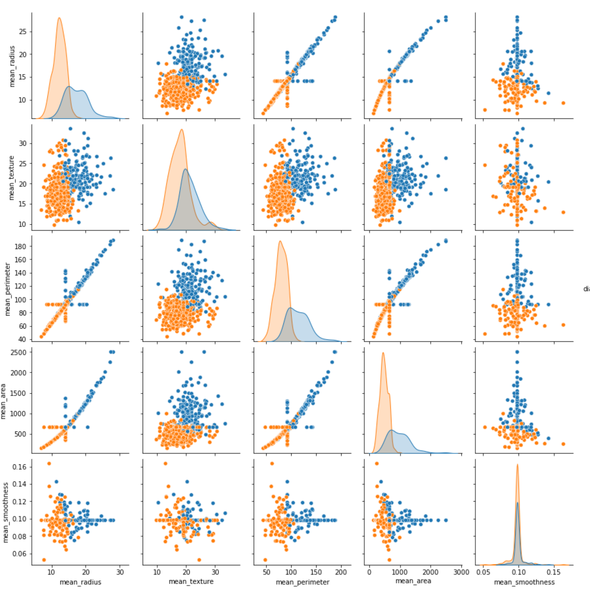
Out[24]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
0	17.990000	10.38	122.800000	1001.00000	0.11840	0
1	20.570000	17.77	132.900000	1326.00000	0.08474	0
2	19.690000	21.25	130.000000	1203.00000	0.10960	0
3	11.420000	20.38	92.095604	386.10000	0.14250	0
4	20.290000	14.34	92.095604	1297.00000	0.10030	0
...	...	...	...	...	...	...
564	14.076482	22.39	142.000000	658.94463	0.11100	0
565	14.076482	28.25	131.200000	658.94463	0.09780	0
566	16.600000	28.08	108.300000	658.94463	0.08455	0
567	20.600000	29.33	140.100000	658.94463	0.11780	0
568	7.760000	24.54	47.920000	658.94463	0.05263	1

569 rows × 6 columns

In [25]:
```
sns.pairplot(df1,hue='diagnosis')
```

Out[25]:
```
<seaborn.axisgrid.PairGrid at 0x2d0be40a9a0>
```



In [26]:
```
X=df1.drop('diagnosis',axis=1)
y=df1['diagnosis']
```

In [27]:
```
#train_test_split the data
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=51)
```

In [28]:
```
X_train.shape
```

Out[28]:
```
(455, 5)
```

In [29]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(X_train,y_train)
lr.score(X_test,y_test)
```

Out[29]:
```
0.49867309549817485
```

In [30]:
```
classifier=RandomForestClassifier(n_estimators=350,criterion='gini')
classifier.fit(X_train,y_train)
classifier.score(X_test,y_test)
```

Out[30]:
```
0.8596491228070176
```

In [31]:
```
classifier1=RandomForestClassifier(n_estimators=310,criterion='entropy')
classifier1.fit(X_train,y_train)
classifier1.score(X_test,y_test)
```

Out[31]:
```
0.868421052631579
```

In [32]:
```
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train,y_train)
knn.score(X_test,y_test)
```

Out[32]:
```
0.8947368421052632
```

In [33]:
```
from sklearn.svm import SVC
svc=SVC()
svc.fit(X_train,y_train)
svc.score(X_test,y_test)
```

Out[33]:
```
0.8771929824561403
```

In [34]:
```
knn.predict(X_test)
```

Out[34]:
```
array([1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1,
       0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1], dtype=int64)
```

In [35]:
```
y_test
```

Out[35]:
```
169    1
137    1
188    1
443    1
345    1
      ..
538    1
123    1
224    1
115    1
425    1
Name: diagnosis, Length: 114, dtype: int64
```

In [37]:
```
#this model gives 89% accuracy
```

In [ ]:
```

```