

Breast Cancer Prediction

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

```
df=pd.read_csv('Breast Cancer Prediction.csv')
```

```
df.shape
```

```
(569, 32)
```

```
df.head()
```

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

	smoothness_mean	compactness_mean	concavity_mean	concave
points_mean \				
0	0.11840	0.27760	0.3001	
0.14710				
1	0.08474	0.07864	0.0869	
0.07017				
2	0.10960	0.15990	0.1974	
0.12790				
3	0.14250	0.28390	0.2414	
0.10520				
4	0.10030	0.13280	0.1980	
0.10430				

	symmetry_mean	...	texture_worst	perimeter_worst	area_worst	\
0	0.2419	...	17.33	184.60	2019.0	
1	0.1812	...	23.41	158.80	1956.0	
2	0.2069	...	25.53	152.50	1709.0	
3	0.2597	...	26.50	98.87	567.7	
4	0.1809	...	16.67	152.20	1575.0	

	smoothness_worst	compactness_worst	concavity_worst	concave
points_worst \				
0	0.1622	0.6656	0.7119	
0.2654				
1	0.1238	0.1866	0.2416	
0.1860				
2	0.1444	0.4245	0.4504	
0.2430				

3	0.2098	0.8663	0.6869
0.2575			
4	0.1374	0.2050	0.4000
0.1625			

	symmetry_worst	fractal_dimension_worst	diagnosis
0	0.4601	0.11890	M
1	0.2750	0.08902	M
2	0.3613	0.08758	M
3	0.6638	0.17300	M
4	0.2364	0.07678	M

[5 rows x 32 columns]

df.columns

```
Index(['id', 'radius_mean', 'texture_mean', 'perimeter_mean',
      'area_mean',
      'smoothness_mean', 'compactness_mean', 'concavity_mean',
      'concave points_mean', 'symmetry_mean',
      'fractal_dimension_mean',
      'radius_se', 'texture_se', 'perimeter_se', 'area_se',
      'smoothness_se',
      'compactness_se', 'concavity_se', 'concave points_se',
      'symmetry_se',
      'fractal_dimension_se', 'radius_worst', 'texture_worst',
      'perimeter_worst', 'area_worst', 'smoothness_worst',
      'compactness_worst', 'concavity_worst', 'concave points_worst',
      'symmetry_worst', 'fractal_dimension_worst', 'diagnosis'],
      dtype='object')
```

df.tail()

	id	radius_mean	texture_mean	perimeter_mean	area_mean	\
564	926424	21.56	22.39	142.00	1479.0	
565	926682	20.13	28.25	131.20	1261.0	
566	926954	16.60	28.08	108.30	858.1	
567	927241	20.60	29.33	140.10	1265.0	
568	92751	7.76	24.54	47.92	181.0	

	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	\
564	0.11100	0.11590	0.24390		
0.13890					
565	0.09780	0.10340	0.14400		
0.09791					
566	0.08455	0.10230	0.09251		
0.05302					
567	0.11780	0.27700	0.35140		
0.15200					

```
568          0.05263          0.04362          0.00000
0.00000
```

```
      symmetry_mean ... texture_worst perimeter_worst area_worst \
564          0.1726 ...          26.40          166.10          2027.0
565          0.1752 ...          38.25          155.00          1731.0
566          0.1590 ...          34.12          126.70          1124.0
567          0.2397 ...          39.42          184.60          1821.0
568          0.1587 ...          30.37           59.16           268.6
```

```
      smoothness_worst compactness_worst concavity_worst \
564          0.14100          0.21130          0.4107
565          0.11660          0.19220          0.3215
566          0.11390          0.30940          0.3403
567          0.16500          0.86810          0.9387
568          0.08996          0.06444          0.0000
```

```
      concave points_worst symmetry_worst fractal_dimension_worst
diagnosis
564          0.2216          0.2060          0.07115
M
565          0.1628          0.2572          0.06637
M
566          0.1418          0.2218          0.07820
M
567          0.2650          0.4087          0.12400
M
568          0.0000          0.2871          0.07039
B
```

```
[5 rows x 32 columns]
```

```
df.sample(5)
```

```
      id radius_mean texture_mean perimeter_mean
area_mean \
456  9112366          11.630          29.29          74.87          415.1
360  901034302          12.540          18.07          79.42          491.9
182   873701          15.700          20.31          101.20          766.6
502   91505          12.540          16.32          81.25          476.3
505   915276           9.676          13.14          64.12          272.5
```

```
      smoothness_mean compactness_mean concavity_mean concave
points_mean \
456          0.09357          0.08574          0.071600
0.020170
```

360	0.07436	0.02650	0.001194
0.005449			
182	0.09597	0.08799	0.065930
0.051890			
502	0.11580	0.10850	0.059280
0.032790			
505	0.12550	0.22040	0.118800
0.070380			

	symmetry_mean	...	texture_worst	perimeter_worst	area_worst	\
456	0.1799	...	38.81	86.04	527.8	
360	0.1528	...	20.98	86.82	585.7	
182	0.1618	...	32.82	129.30	1269.0	
502	0.1943	...	21.40	86.67	552.0	
505	0.2057	...	18.04	69.47	328.1	

	smoothness_worst	compactness_worst	concavity_worst	\
456	0.14060	0.20310	0.292300	
360	0.09293	0.04327	0.003581	
182	0.14140	0.35470	0.290200	
502	0.15800	0.17510	0.188900	
505	0.20060	0.36630	0.291300	

	concave	points_worst	symmetry_worst	fractal_dimension_worst
diagnosis				
456		0.06835	0.2884	0.07220
B				
360		0.01635	0.2233	0.05521
B				
182		0.15410	0.3437	0.08631
M				
502		0.08411	0.3155	0.07538
B				
505		0.10750	0.2848	0.13640
B				

[5 rows x 32 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 569 entries, 0 to 568

Data columns (total 32 columns):

#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	id	569 non-null	int64
1	radius_mean	569 non-null	float64
2	texture_mean	569 non-null	float64
3	perimeter_mean	569 non-null	float64
4	area_mean	569 non-null	float64

5	smoothness_mean	569	non-null	float64
6	compactness_mean	569	non-null	float64
7	concavity_mean	569	non-null	float64
8	concave points_mean	569	non-null	float64
9	symmetry_mean	569	non-null	float64
10	fractal_dimension_mean	569	non-null	float64
11	radius_se	569	non-null	float64
12	texture_se	569	non-null	float64
13	perimeter_se	569	non-null	float64
14	area_se	569	non-null	float64
15	smoothness_se	569	non-null	float64
16	compactness_se	569	non-null	float64
17	concavity_se	569	non-null	float64
18	concave points_se	569	non-null	float64
19	symmetry_se	569	non-null	float64
20	fractal_dimension_se	569	non-null	float64
21	radius_worst	569	non-null	float64
22	texture_worst	569	non-null	float64
23	perimeter_worst	569	non-null	float64
24	area_worst	569	non-null	float64
25	smoothness_worst	569	non-null	float64
26	compactness_worst	569	non-null	float64
27	concavity_worst	569	non-null	float64
28	concave points_worst	569	non-null	float64
29	symmetry_worst	569	non-null	float64
30	fractal_dimension_worst	569	non-null	float64
31	diagnosis	569	non-null	object

dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB

df.dtypes

id	int64
radius_mean	float64
texture_mean	float64
perimeter_mean	float64
area_mean	float64
smoothness_mean	float64
compactness_mean	float64
concavity_mean	float64
concave points_mean	float64
symmetry_mean	float64
fractal_dimension_mean	float64
radius_se	float64
texture_se	float64
perimeter_se	float64
area_se	float64
smoothness_se	float64
compactness_se	float64
concavity_se	float64

```

concave points_se      float64
symmetry_se            float64
fractal_dimension_se   float64
radius_worst           float64
texture_worst          float64
perimeter_worst        float64
area_worst             float64
smoothness_worst       float64
compactness_worst      float64
concavity_worst        float64
concave points_worst   float64
symmetry_worst         float64
fractal_dimension_worst float64
diagnosis              object
dtype: object

```

```
df.describe().T
```

	count	mean	std
min \			
id	569.0	3.037183e+07	1.250206e+08
8670.000000			
radius_mean	569.0	1.412729e+01	3.524049e+00
6.981000			
texture_mean	569.0	1.928965e+01	4.301036e+00
9.710000			
perimeter_mean	569.0	9.196903e+01	2.429898e+01
43.790000			
area_mean	569.0	6.548891e+02	3.519141e+02
143.500000			
smoothness_mean	569.0	9.636028e-02	1.406413e-02
0.052630			
compactness_mean	569.0	1.043410e-01	5.281276e-02
0.019380			
concavity_mean	569.0	8.879932e-02	7.971981e-02
0.000000			
concave points_mean	569.0	4.891915e-02	3.880284e-02
0.000000			
symmetry_mean	569.0	1.811619e-01	2.741428e-02
0.106000			
fractal_dimension_mean	569.0	6.279761e-02	7.060363e-03
0.049960			
radius_se	569.0	4.051721e-01	2.773127e-01
0.111500			
texture_se	569.0	1.216853e+00	5.516484e-01
0.360200			
perimeter_se	569.0	2.866059e+00	2.021855e+00
0.757000			
area_se	569.0	4.033708e+01	4.549101e+01
6.802000			

smoothness_se	569.0	7.040979e-03	3.002518e-03
0.001713			
compactness_se	569.0	2.547814e-02	1.790818e-02
0.002252			
concavity_se	569.0	3.189372e-02	3.018606e-02
0.000000			
concave points_se	569.0	1.179614e-02	6.170285e-03
0.000000			
symmetry_se	569.0	2.054230e-02	8.266372e-03
0.007882			
fractal_dimension_se	569.0	3.794904e-03	2.646071e-03
0.000895			
radius_worst	569.0	1.626919e+01	4.833242e+00
7.930000			
texture_worst	569.0	2.567722e+01	6.146258e+00
12.020000			
perimeter_worst	569.0	1.072612e+02	3.360254e+01
50.410000			
area_worst	569.0	8.805831e+02	5.693570e+02
185.200000			
smoothness_worst	569.0	1.323686e-01	2.283243e-02
0.071170			
compactness_worst	569.0	2.542650e-01	1.573365e-01
0.027290			
concavity_worst	569.0	2.721885e-01	2.086243e-01
0.000000			
concave points_worst	569.0	1.146062e-01	6.573234e-02
0.000000			
symmetry_worst	569.0	2.900756e-01	6.186747e-02
0.156500			
fractal_dimension_worst	569.0	8.394582e-02	1.806127e-02
0.055040			

	25%	50%	75%	\
id	869218.000000	906024.000000	8.813129e+06	
radius_mean	11.700000	13.370000	1.578000e+01	
texture_mean	16.170000	18.840000	2.180000e+01	
perimeter_mean	75.170000	86.240000	1.041000e+02	
area_mean	420.300000	551.100000	7.827000e+02	
smoothness_mean	0.086370	0.095870	1.053000e-01	
compactness_mean	0.064920	0.092630	1.304000e-01	
concavity_mean	0.029560	0.061540	1.307000e-01	
concave points_mean	0.020310	0.033500	7.400000e-02	
symmetry_mean	0.161900	0.179200	1.957000e-01	
fractal_dimension_mean	0.057700	0.061540	6.612000e-02	
radius_se	0.232400	0.324200	4.789000e-01	
texture_se	0.833900	1.108000	1.474000e+00	
perimeter_se	1.606000	2.287000	3.357000e+00	
area_se	17.850000	24.530000	4.519000e+01	

smoothness_se	0.005169	0.006380	8.146000e-03
compactness_se	0.013080	0.020450	3.245000e-02
concavity_se	0.015090	0.025890	4.205000e-02
concave points_se	0.007638	0.010930	1.471000e-02
symmetry_se	0.015160	0.018730	2.348000e-02
fractal_dimension_se	0.002248	0.003187	4.558000e-03
radius_worst	13.010000	14.970000	1.879000e+01
texture_worst	21.080000	25.410000	2.972000e+01
perimeter_worst	84.110000	97.660000	1.254000e+02
area_worst	515.300000	686.500000	1.084000e+03
smoothness_worst	0.116600	0.131300	1.460000e-01
compactness_worst	0.147200	0.211900	3.391000e-01
concavity_worst	0.114500	0.226700	3.829000e-01
concave points_worst	0.064930	0.099930	1.614000e-01
symmetry_worst	0.250400	0.282200	3.179000e-01
fractal_dimension_worst	0.071460	0.080040	9.208000e-02

	max
id	9.113205e+08
radius_mean	2.811000e+01
texture_mean	3.928000e+01
perimeter_mean	1.885000e+02
area_mean	2.501000e+03
smoothness_mean	1.634000e-01
compactness_mean	3.454000e-01
concavity_mean	4.268000e-01
concave points_mean	2.012000e-01
symmetry_mean	3.040000e-01
fractal_dimension_mean	9.744000e-02
radius_se	2.873000e+00
texture_se	4.885000e+00
perimeter_se	2.198000e+01
area_se	5.422000e+02
smoothness_se	3.113000e-02
compactness_se	1.354000e-01
concavity_se	3.960000e-01
concave points_se	5.279000e-02
symmetry_se	7.895000e-02
fractal_dimension_se	2.984000e-02
radius_worst	3.604000e+01
texture_worst	4.954000e+01
perimeter_worst	2.512000e+02
area_worst	4.254000e+03
smoothness_worst	2.226000e-01
compactness_worst	1.058000e+00
concavity_worst	1.252000e+00
concave points_worst	2.910000e-01
symmetry_worst	6.638000e-01
fractal_dimension_worst	2.075000e-01


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

```
id                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
diagnosis         0
dtype: int64
```

```
df.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
...
564    False
565    False
566    False
567    False
568    False
Length: 569, dtype: bool
```

```
df.nunique()
```

```
id                569
radius_mean       456
texture_mean      479
perimeter_mean    522
area_mean         539
smoothness_mean   474
compactness_mean  537
concavity_mean    537
concave points_mean 542
symmetry_mean     432
fractal_dimension_mean 499
radius_se         540
texture_se        519
perimeter_se      533
area_se           528
smoothness_se     547
compactness_se    541
concavity_se      533
concave points_se 507
symmetry_se       498
fractal_dimension_se 545
radius_worst      457
texture_worst     511
perimeter_worst   514
area_worst        544
smoothness_worst  411
compactness_worst 529
concavity_worst   539
concave points_worst 492
symmetry_worst    500
fractal_dimension_worst 535
diagnosis         2
dtype: int64
```

```
df.drop('id',axis=1,inplace=True)
```

```
df.head()
```

	radius_mean	texture_mean	perimeter_mean	area_mean
0	17.99	10.38	122.80	1001.0
0.11840				
1	20.57	17.77	132.90	1326.0
0.08474				
2	19.69	21.25	130.00	1203.0
0.10960				
3	11.42	20.38	77.58	386.1
0.14250				

4	20.29	14.34	135.10	1297.0
0.10030				
	compactness_mean	concavity_mean	concave	points_mean
symmetry_mean \				
0	0.27760	0.3001		0.14710
0.2419				
1	0.07864	0.0869		0.07017
0.1812				
2	0.15990	0.1974		0.12790
0.2069				
3	0.28390	0.2414		0.10520
0.2597				
4	0.13280	0.1980		0.10430
0.1809				
	fractal_dimension_mean	...	texture_worst	perimeter_worst
area_worst \				
0	0.07871	...	17.33	184.60
2019.0				
1	0.05667	...	23.41	158.80
1956.0				
2	0.05999	...	25.53	152.50
1709.0				
3	0.09744	...	26.50	98.87
567.7				
4	0.05883	...	16.67	152.20
1575.0				
	smoothness_worst	compactness_worst	concavity_worst	concave
points_worst \				
0	0.1622	0.6656		0.7119
0.2654				
1	0.1238	0.1866		0.2416
0.1860				
2	0.1444	0.4245		0.4504
0.2430				
3	0.2098	0.8663		0.6869
0.2575				
4	0.1374	0.2050		0.4000
0.1625				
	symmetry_worst	fractal_dimension_worst	diagnosis	
0	0.4601	0.11890		M
1	0.2750	0.08902		M
2	0.3613	0.08758		M
3	0.6638	0.17300		M
4	0.2364	0.07678		M

[5 rows x 31 columns]

```

df.shape
(569, 31)
df.columns
Index(['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
      'smoothness_mean', 'compactness_mean', 'concavity_mean',
      'concave points_mean', 'symmetry_mean',
      'fractal_dimension_mean',
      'radius_se', 'texture_se', 'perimeter_se', 'area_se',
      'smoothness_se',
      'compactness_se', 'concavity_se', 'concave points_se',
      'symmetry_se',
      'fractal_dimension_se', 'radius_worst', 'texture_worst',
      'perimeter_worst', 'area_worst', 'smoothness_worst',
      'compactness_worst', 'concavity_worst', 'concave points_worst',
      'symmetry_worst', 'fractal_dimension_worst', 'diagnosis'],
      dtype='object')

df['diagnosis'] = df['diagnosis'].map({'M': 0, 'B': 1})
df.head()

```

	radius_mean	texture_mean	perimeter_mean	area_mean
0	17.99	10.38	122.80	1001.0
1	20.57	17.77	132.90	1326.0
2	19.69	21.25	130.00	1203.0
3	11.42	20.38	77.58	386.1
4	20.29	14.34	135.10	1297.0

	compactness_mean	concavity_mean	concave points_mean
0	0.27760	0.3001	0.14710
1	0.07864	0.0869	0.07017
2	0.15990	0.1974	0.12790
3	0.28390	0.2414	0.10520
4	0.13280	0.1980	0.10430

	fractal_dimension_mean	...	texture_worst	perimeter_worst
--	------------------------	-----	---------------	-----------------

area_worst \				
0	0.07871	...	17.33	184.60
2019.0				
1	0.05667	...	23.41	158.80
1956.0				
2	0.05999	...	25.53	152.50
1709.0				
3	0.09744	...	26.50	98.87
567.7				
4	0.05883	...	16.67	152.20
1575.0				

	smoothness_worst	compactness_worst	concavity_worst	concave
points_worst \				
0	0.1622	0.6656	0.7119	
0.2654				
1	0.1238	0.1866	0.2416	
0.1860				
2	0.1444	0.4245	0.4504	
0.2430				
3	0.2098	0.8663	0.6869	
0.2575				
4	0.1374	0.2050	0.4000	
0.1625				

	symmetry_worst	fractal_dimension_worst	diagnosis
0	0.4601	0.11890	0
1	0.2750	0.08902	0
2	0.3613	0.08758	0
3	0.6638	0.17300	0
4	0.2364	0.07678	0

[5 rows x 31 columns]

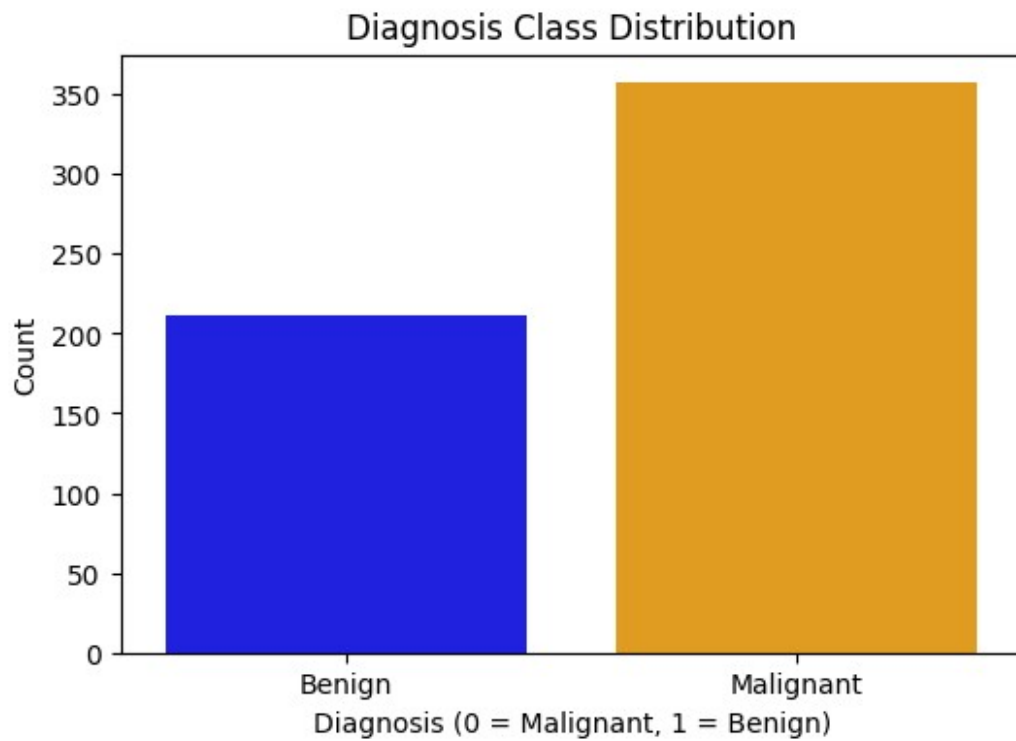
#Countplot

```
plt.figure(figsize=(6,4))
sns.countplot(x=df['diagnosis'],palette=["blue","orange"])
plt.xticks(ticks=[0,1],labels=['Benign','Malignant'])
plt.title('Diagnosis Class Distribution')
plt.xlabel('Diagnosis (0 = Malignant, 1 = Benign)')
plt.ylabel('Count')
plt.show()
```

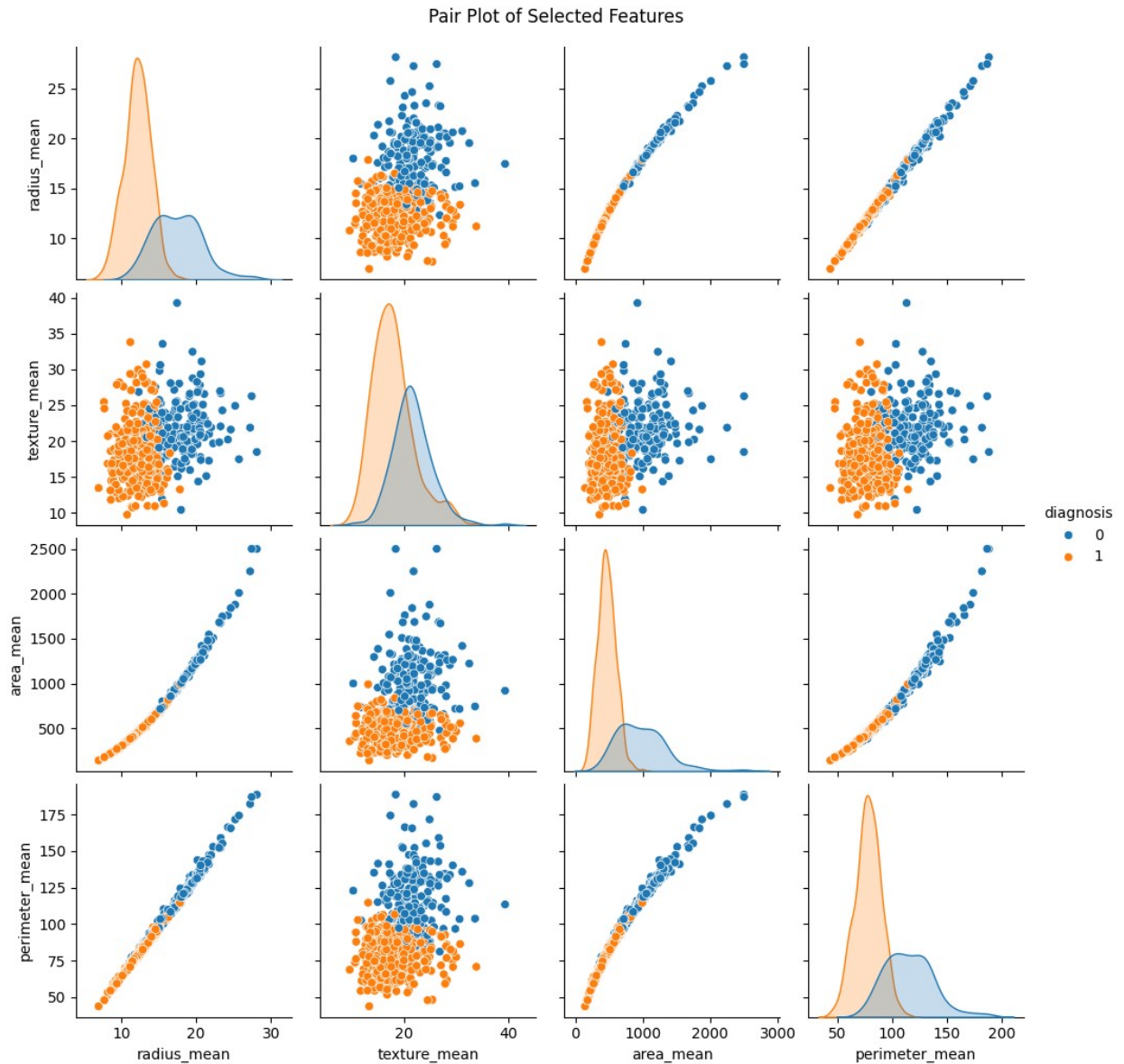
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_376\3898453000.py:3:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(x=df['diagnosis'],palette=["blue","orange"])
```



```
#Pairplot
sns.pairplot(df[['radius_mean', 'texture_mean', 'area_mean',
'perimeter_mean', 'diagnosis']], hue='diagnosis')
plt.suptitle('Pair Plot of Selected Features', y=1.02)
plt.show()
```



```
df.corr()
```

	radius_mean	texture_mean	perimeter_mean
area_mean \			
radius_mean	1.000000	0.323782	0.997855
0.987357			
texture_mean	0.323782	1.000000	0.329533
0.321086			
perimeter_mean	0.997855	0.329533	1.000000
0.986507			
area_mean	0.987357	0.321086	0.986507
1.000000			
smoothness_mean	0.170581	-0.023389	0.207278
0.177028			

compactness_mean 0.498502	0.506124	0.236702	0.556936	
concavity_mean 0.685983	0.676764	0.302418	0.716136	
concave points_mean 0.823269	0.822529	0.293464	0.850977	
symmetry_mean 0.151293	0.147741	0.071401	0.183027	
fractal_dimension_mean 0.283110	-0.311631	-0.076437	-0.261477	-
radius_se 0.732562	0.679090	0.275869	0.691765	
texture_se 0.066280	-0.097317	0.386358	-0.086761	-
perimeter_se 0.726628	0.674172	0.281673	0.693135	
area_se 0.800086	0.735864	0.259845	0.744983	
smoothness_se 0.166777	-0.222600	0.006614	-0.202694	-
compactness_se 0.212583	0.206000	0.191975	0.250744	
concavity_se 0.207660	0.194204	0.143293	0.228082	
concave points_se 0.372320	0.376169	0.163851	0.407217	
symmetry_se 0.072497	-0.104321	0.009127	-0.081629	-
fractal_dimension_se 0.019887	-0.042641	0.054458	-0.005523	-
radius_worst 0.962746	0.969539	0.352573	0.969476	
texture_worst 0.287489	0.297008	0.912045	0.303038	
perimeter_worst 0.959120	0.965137	0.358040	0.970387	
area_worst 0.959213	0.941082	0.343546	0.941550	
smoothness_worst 0.123523	0.119616	0.077503	0.150549	
compactness_worst 0.390410	0.413463	0.277830	0.455774	
concavity_worst 0.512606	0.526911	0.301025	0.563879	
concave points_worst 0.722017	0.744214	0.295316	0.771241	
symmetry_worst 0.143570	0.163953	0.105008	0.189115	
fractal_dimension_worst	0.007066	0.119205	0.051019	

0.003738			
diagnosis	-0.730029	-0.415185	-0.742636
0.708984			

	smoothness_mean	compactness_mean
concavity_mean \		
radius_mean	0.170581	0.506124
0.676764		
texture_mean	-0.023389	0.236702
0.302418		
perimeter_mean	0.207278	0.556936
0.716136		
area_mean	0.177028	0.498502
0.685983		
smoothness_mean	1.000000	0.659123
0.521984		
compactness_mean	0.659123	1.000000
0.883121		
concavity_mean	0.521984	0.883121
1.000000		
concave points_mean	0.553695	0.831135
0.921391		
symmetry_mean	0.557775	0.602641
0.500667		
fractal_dimension_mean	0.584792	0.565369
0.336783		
radius_se	0.301467	0.497473
0.631925		
texture_se	0.068406	0.046205
0.076218		
perimeter_se	0.296092	0.548905
0.660391		
area_se	0.246552	0.455653
0.617427		
smoothness_se	0.332375	0.135299
0.098564		
compactness_se	0.318943	0.738722
0.670279		
concavity_se	0.248396	0.570517
0.691270		
concave points_se	0.380676	0.642262
0.683260		
symmetry_se	0.200774	0.229977
0.178009		
fractal_dimension_se	0.283607	0.507318
0.449301		
radius_worst	0.213120	0.535315
0.688236		
texture_worst	0.036072	0.248133

0.299879		
perimeter_worst	0.238853	0.590210
0.729565		
area_worst	0.206718	0.509604
0.675987		
smoothness_worst	0.805324	0.565541
0.448822		
compactness_worst	0.472468	0.865809
0.754968		
concavity_worst	0.434926	0.816275
0.884103		
concave points_worst	0.503053	0.815573
0.861323		
symmetry_worst	0.394309	0.510223
0.409464		
fractal_dimension_worst	0.499316	0.687382
0.514930		
diagnosis	-0.358560	-0.596534
0.696360		

	concave points_mean	symmetry_mean \
radius_mean	0.822529	0.147741
texture_mean	0.293464	0.071401
perimeter_mean	0.850977	0.183027
area_mean	0.823269	0.151293
smoothness_mean	0.553695	0.557775
compactness_mean	0.831135	0.602641
concavity_mean	0.921391	0.500667
concave points_mean	1.000000	0.462497
symmetry_mean	0.462497	1.000000
fractal_dimension_mean	0.166917	0.479921
radius_se	0.698050	0.303379
texture_se	0.021480	0.128053
perimeter_se	0.710650	0.313893
area_se	0.690299	0.223970
smoothness_se	0.027653	0.187321
compactness_se	0.490424	0.421659
concavity_se	0.439167	0.342627
concave points_se	0.615634	0.393298
symmetry_se	0.095351	0.449137
fractal_dimension_se	0.257584	0.331786
radius_worst	0.830318	0.185728
texture_worst	0.292752	0.090651
perimeter_worst	0.855923	0.219169
area_worst	0.809630	0.177193
smoothness_worst	0.452753	0.426675
compactness_worst	0.667454	0.473200
concavity_worst	0.752399	0.433721
concave points_worst	0.910155	0.430297

symmetry_worst	0.375744	0.699826
fractal_dimension_worst	0.368661	0.438413
diagnosis	-0.776614	-0.330499

	fractal_dimension_mean	...	texture_worst \
radius_mean	-0.311631	...	0.297008
texture_mean	-0.076437	...	0.912045
perimeter_mean	-0.261477	...	0.303038
area_mean	-0.283110	...	0.287489
smoothness_mean	0.584792	...	0.036072
compactness_mean	0.565369	...	0.248133
concavity_mean	0.336783	...	0.299879
concave points_mean	0.166917	...	0.292752
symmetry_mean	0.479921	...	0.090651
fractal_dimension_mean	1.000000	...	-0.051269
radius_se	0.000111	...	0.194799
texture_se	0.164174	...	0.409003
perimeter_se	0.039830	...	0.200371
area_se	-0.090170	...	0.196497
smoothness_se	0.401964	...	-0.074743
compactness_se	0.559837	...	0.143003
concavity_se	0.446630	...	0.100241
concave points_se	0.341198	...	0.086741
symmetry_se	0.345007	...	-0.077473
fractal_dimension_se	0.688132	...	-0.003195
radius_worst	-0.253691	...	0.359921
texture_worst	-0.051269	...	1.000000
perimeter_worst	-0.205151	...	0.365098
area_worst	-0.231854	...	0.345842
smoothness_worst	0.504942	...	0.225429
compactness_worst	0.458798	...	0.360832
concavity_worst	0.346234	...	0.368366
concave points_worst	0.175325	...	0.359755
symmetry_worst	0.334019	...	0.233027
fractal_dimension_worst	0.767297	...	0.219122
diagnosis	0.012838	...	-0.456903

	perimeter_worst	area_worst	smoothness_worst
\			
radius_mean	0.965137	0.941082	0.119616
texture_mean	0.358040	0.343546	0.077503
perimeter_mean	0.970387	0.941550	0.150549
area_mean	0.959120	0.959213	0.123523
smoothness_mean	0.238853	0.206718	0.805324
compactness_mean	0.590210	0.509604	0.565541

concavity_mean	0.729565	0.675987	0.448822
concave points_mean	0.855923	0.809630	0.452753
symmetry_mean	0.219169	0.177193	0.426675
fractal_dimension_mean	-0.205151	-0.231854	0.504942
radius_se	0.719684	0.751548	0.141919
texture_se	-0.102242	-0.083195	-0.073658
perimeter_se	0.721031	0.730713	0.130054
area_se	0.761213	0.811408	0.125389
smoothness_se	-0.217304	-0.182195	0.314457
compactness_se	0.260516	0.199371	0.227394
concavity_se	0.226680	0.188353	0.168481
concave points_se	0.394999	0.342271	0.215351
symmetry_se	-0.103753	-0.110343	-0.012662
fractal_dimension_se	-0.001000	-0.022736	0.170568
radius_worst	0.993708	0.984015	0.216574
texture_worst	0.365098	0.345842	0.225429
perimeter_worst	1.000000	0.977578	0.236775
area_worst	0.977578	1.000000	0.209145
smoothness_worst	0.236775	0.209145	1.000000
compactness_worst	0.529408	0.438296	0.568187
concavity_worst	0.618344	0.543331	0.518523
concave points_worst	0.816322	0.747419	0.547691
symmetry_worst	0.269493	0.209146	0.493838
fractal_dimension_worst	0.138957	0.079647	0.617624
diagnosis	-0.782914	-0.733825	-0.421465

	compactness_worst	concavity_worst \
radius_mean	0.413463	0.526911
texture_mean	0.277830	0.301025
perimeter_mean	0.455774	0.563879
area_mean	0.390410	0.512606
smoothness_mean	0.472468	0.434926
compactness_mean	0.865809	0.816275
concavity_mean	0.754968	0.884103
concave points_mean	0.667454	0.752399
symmetry_mean	0.473200	0.433721
fractal_dimension_mean	0.458798	0.346234
radius_se	0.287103	0.380585
texture_se	-0.092439	-0.068956
perimeter_se	0.341919	0.418899
area_se	0.283257	0.385100
smoothness_se	-0.055558	-0.058298
compactness_se	0.678780	0.639147
concavity_se	0.484858	0.662564
concave points_se	0.452888	0.549592
symmetry_se	0.060255	0.037119
fractal_dimension_se	0.390159	0.379975
radius_worst	0.475820	0.573975
texture_worst	0.360832	0.368366
perimeter_worst	0.529408	0.618344
area_worst	0.438296	0.543331
smoothness_worst	0.568187	0.518523
compactness_worst	1.000000	0.892261
concavity_worst	0.892261	1.000000
concave points_worst	0.801080	0.855434
symmetry_worst	0.614441	0.532520
fractal_dimension_worst	0.810455	0.686511
diagnosis	-0.590998	-0.659610

	concave points_worst	symmetry_worst \
radius_mean	0.744214	0.163953
texture_mean	0.295316	0.105008
perimeter_mean	0.771241	0.189115
area_mean	0.722017	0.143570
smoothness_mean	0.503053	0.394309
compactness_mean	0.815573	0.510223
concavity_mean	0.861323	0.409464
concave points_mean	0.910155	0.375744
symmetry_mean	0.430297	0.699826
fractal_dimension_mean	0.175325	0.334019
radius_se	0.531062	0.094543
texture_se	-0.119638	-0.128215
perimeter_se	0.554897	0.109930
area_se	0.538166	0.074126
smoothness_se	-0.102007	-0.107342

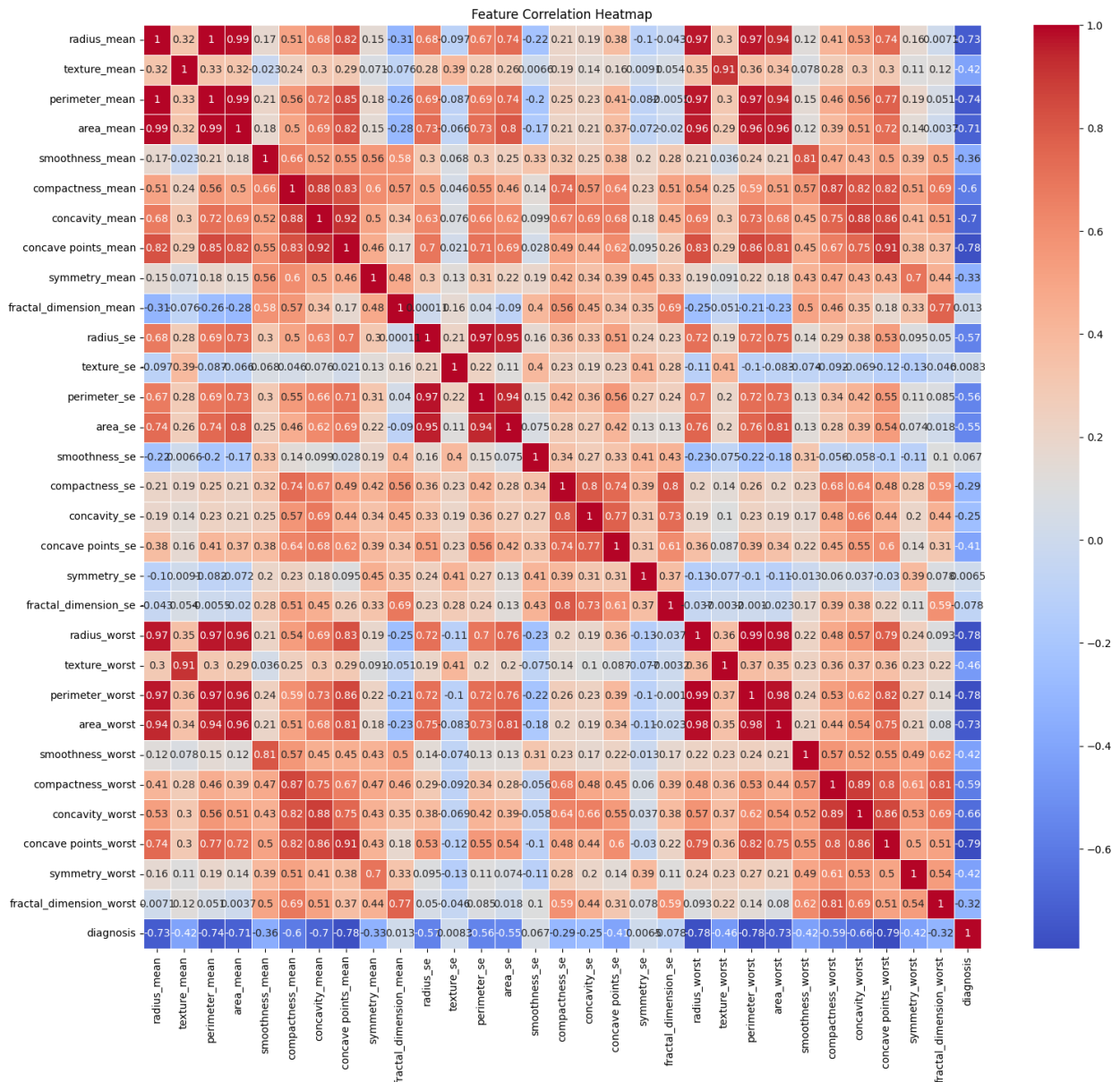
compactness_se	0.483208	0.277878
concavity_se	0.440472	0.197788
concave points_se	0.602450	0.143116
symmetry_se	-0.030413	0.389402
fractal_dimension_se	0.215204	0.111094
radius_worst	0.787424	0.243529
texture_worst	0.359755	0.233027
perimeter_worst	0.816322	0.269493
area_worst	0.747419	0.209146
smoothness_worst	0.547691	0.493838
compactness_worst	0.801080	0.614441
concavity_worst	0.855434	0.532520
concave points_worst	1.000000	0.502528
symmetry_worst	0.502528	1.000000
fractal_dimension_worst	0.511114	0.537848
diagnosis	-0.793566	-0.416294

	fractal_dimension_worst	diagnosis
radius_mean	0.007066	-0.730029
texture_mean	0.119205	-0.415185
perimeter_mean	0.051019	-0.742636
area_mean	0.003738	-0.708984
smoothness_mean	0.499316	-0.358560
compactness_mean	0.687382	-0.596534
concavity_mean	0.514930	-0.696360
concave points_mean	0.368661	-0.776614
symmetry_mean	0.438413	-0.330499
fractal_dimension_mean	0.767297	0.012838
radius_se	0.049559	-0.567134
texture_se	-0.045655	0.008303
perimeter_se	0.085433	-0.556141
area_se	0.017539	-0.548236
smoothness_se	0.101480	0.067016
compactness_se	0.590973	-0.292999
concavity_se	0.439329	-0.253730
concave points_se	0.310655	-0.408042
symmetry_se	0.078079	0.006522
fractal_dimension_se	0.591328	-0.077972
radius_worst	0.093492	-0.776454
texture_worst	0.219122	-0.456903
perimeter_worst	0.138957	-0.782914
area_worst	0.079647	-0.733825
smoothness_worst	0.617624	-0.421465
compactness_worst	0.810455	-0.590998
concavity_worst	0.686511	-0.659610
concave points_worst	0.511114	-0.793566
symmetry_worst	0.537848	-0.416294
fractal_dimension_worst	1.000000	-0.323872
diagnosis	-0.323872	1.000000

[31 rows x 31 columns]

#Heatmap

```
plt.figure(figsize=(18, 16))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Feature Correlation Heatmap')
plt.show()
```



#Box-Plot

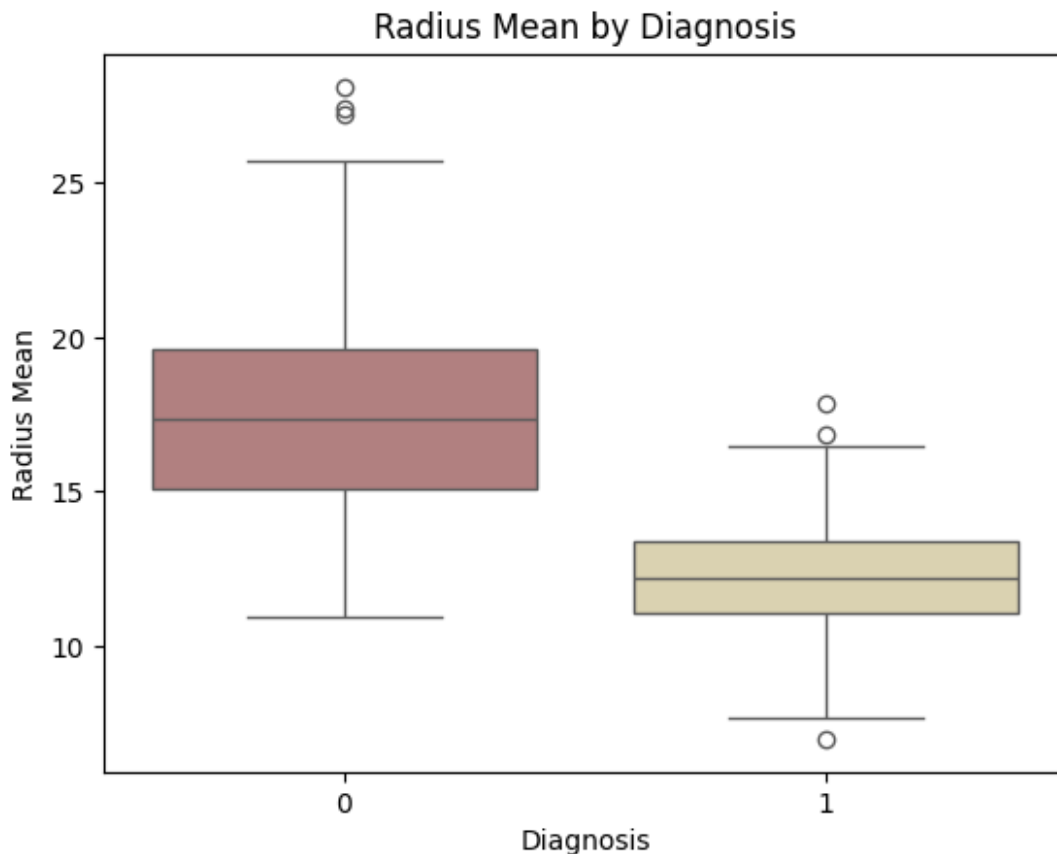
```
sns.boxplot(x='diagnosis', y='radius_mean', data=df, palette='pink')
plt.title('Radius Mean by Diagnosis')
plt.xlabel('Diagnosis')
```

```
plt.ylabel('Radius Mean')
plt.show()
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_376\410977824.py:2:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(x='diagnosis', y='radius_mean', data=df,palette='pink')
```



```
X=df.iloc[:, :-1]
y=df['diagnosis']
```

```
print(X)
```

	radius_mean	texture_mean	perimeter_mean	area_mean
smoothness_mean \				
0	17.99	10.38	122.80	1001.0
0.11840				
1	20.57	17.77	132.90	1326.0
0.08474				

2	19.69	21.25	130.00	1203.0
0.10960				
3	11.42	20.38	77.58	386.1
0.14250				
4	20.29	14.34	135.10	1297.0
0.10030				
...
...				
564	21.56	22.39	142.00	1479.0
0.11100				
565	20.13	28.25	131.20	1261.0
0.09780				
566	16.60	28.08	108.30	858.1
0.08455				
567	20.60	29.33	140.10	1265.0
0.11780				
568	7.76	24.54	47.92	181.0
0.05263				

	compactness_mean	concavity_mean	concave	points_mean
symmetry_mean \				
0	0.27760	0.30010		0.14710
0.2419				
1	0.07864	0.08690		0.07017
0.1812				
2	0.15990	0.19740		0.12790
0.2069				
3	0.28390	0.24140		0.10520
0.2597				
4	0.13280	0.19800		0.10430
0.1809				
...
...				
564	0.11590	0.24390		0.13890
0.1726				
565	0.10340	0.14400		0.09791
0.1752				
566	0.10230	0.09251		0.05302
0.1590				
567	0.27700	0.35140		0.15200
0.2397				
568	0.04362	0.00000		0.00000
0.1587				

	fractal_dimension_mean	...	radius_worst	texture_worst	\
0	0.07871	...	25.380	17.33	
1	0.05667	...	24.990	23.41	
2	0.05999	...	23.570	25.53	
3	0.09744	...	14.910	26.50	

4	0.05883	...	22.540	16.67
..
564	0.05623	...	25.450	26.40
565	0.05533	...	23.690	38.25
566	0.05648	...	18.980	34.12
567	0.07016	...	25.740	39.42
568	0.05884	...	9.456	30.37

	perimeter_worst	area_worst	smoothness_worst	compactness_worst
\				
0	184.60	2019.0	0.16220	0.66560
1	158.80	1956.0	0.12380	0.18660
2	152.50	1709.0	0.14440	0.42450
3	98.87	567.7	0.20980	0.86630
4	152.20	1575.0	0.13740	0.20500
..
564	166.10	2027.0	0.14100	0.21130
565	155.00	1731.0	0.11660	0.19220
566	126.70	1124.0	0.11390	0.30940
567	184.60	1821.0	0.16500	0.86810
568	59.16	268.6	0.08996	0.06444

	concavity_worst	concave points_worst	symmetry_worst	\
0	0.7119	0.2654	0.4601	
1	0.2416	0.1860	0.2750	
2	0.4504	0.2430	0.3613	
3	0.6869	0.2575	0.6638	
4	0.4000	0.1625	0.2364	
..	
564	0.4107	0.2216	0.2060	
565	0.3215	0.1628	0.2572	
566	0.3403	0.1418	0.2218	
567	0.9387	0.2650	0.4087	
568	0.0000	0.0000	0.2871	

	fractal_dimension_worst
0	0.11890
1	0.08902
2	0.08758
3	0.17300

```
4          0.07678
..
564        0.07115
565        0.06637
566        0.07820
567        0.12400
568        0.07039
```

```
[569 rows x 30 columns]
```

```
print(y)
```

```
0      0
1      0
2      0
3      0
4      0
..
564    0
565    0
566    0
567    0
568    1
```

```
Name: diagnosis, Length: 569, dtype: int64
```

```
from sklearn.preprocessing import StandardScaler
```

```
scaler=StandardScaler()
ss=scaler.fit_transform(X)
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

```
print(X.shape,X_train.shape,X_test.shape)
```

```
(569, 30) (455, 30) (114, 30)
```

```
print(y.shape,y_train.shape,y_test.shape)
```

```
(569,) (455,) (114,)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import
accuracy_score,precision_score,recall_score,f1_score
```

```

models = {
    "Logistic Regression":
    LogisticRegression(max_iter=2000,solver='liblinear'),
    "KNN": KNeighborsClassifier(n_neighbors=5),
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "Random Forest": RandomForestClassifier(random_state=42),
    "SVM": SVC(kernel='linear')
}

# Dictionary to store accuracy results
results = {}

# Train and evaluate each model
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')

    # Store accuracy in the results dictionary
    results[name] = accuracy

    # Print metrics
    print(f"--- {name} ---")
    print(f"Accuracy : {accuracy:.4f}")
    print(f"Precision: {precision:.4f}")
    print(f"Recall    : {recall:.4f}")
    print(f"F1 Score  : {f1:.4f}")
    print("\n")

--- Logistic Regression ---
Accuracy : 0.9561
Precision: 0.9569
Recall    : 0.9561
F1 Score  : 0.9558

--- KNN ---
Accuracy : 0.9561
Precision: 0.9590
Recall    : 0.9561
F1 Score  : 0.9555

--- Decision Tree ---
Accuracy : 0.9474
Precision: 0.9474

```

```
Recall    : 0.9474
F1 Score  : 0.9474
```

```
--- Random Forest ---
```

```
Accuracy : 0.9649
Precision: 0.9652
Recall    : 0.9649
F1 Score  : 0.9647
```

```
--- SVM ---
```

```
Accuracy : 0.9561
Precision: 0.9569
Recall    : 0.9561
F1 Score  : 0.9558
```

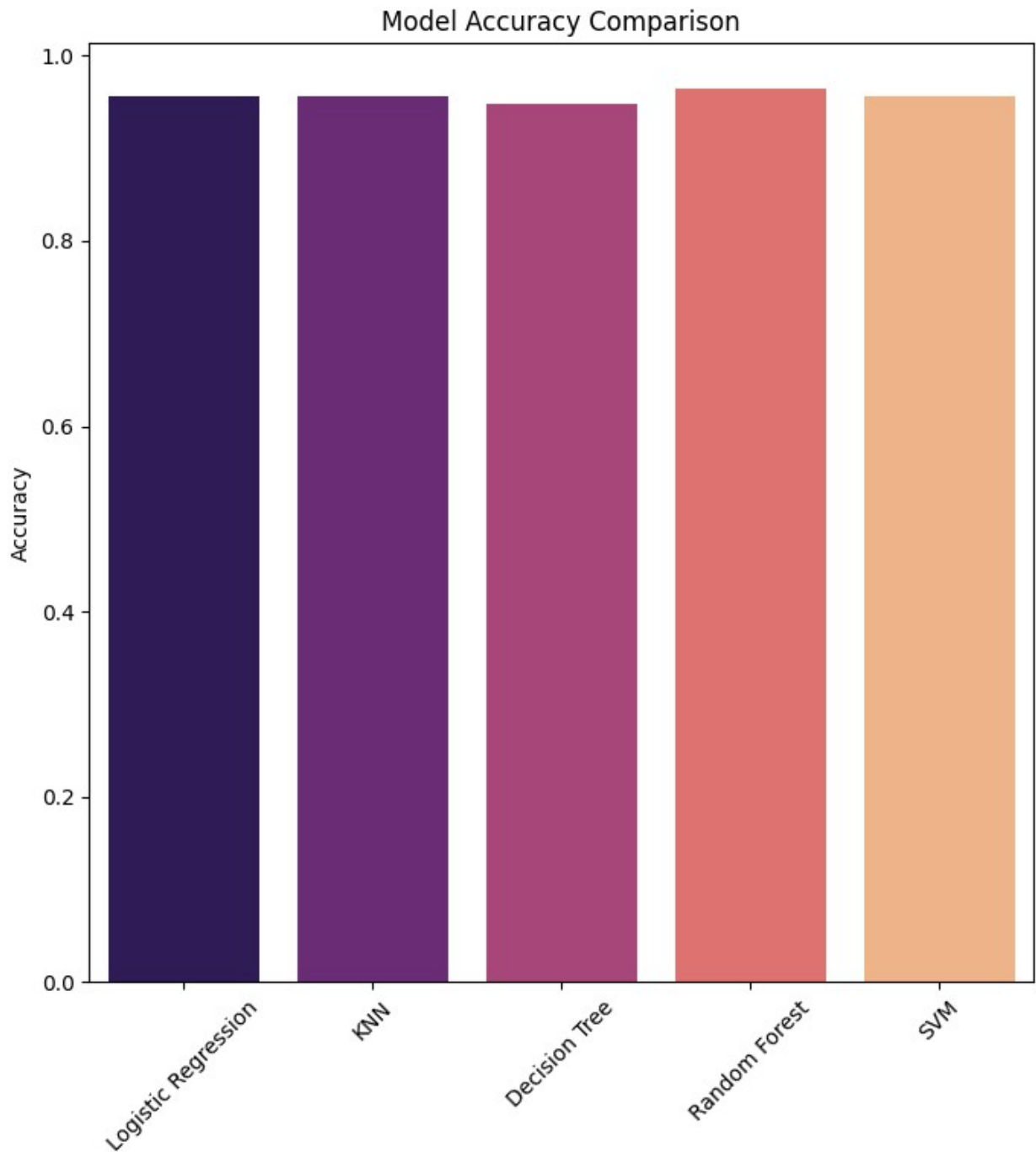
```
# Plotting the results
```

```
plt.figure(figsize=(8,8))
sns.barplot(x=list(results.keys()), y=list(results.values()),
palette='magma')
plt.title('Model Accuracy Comparison')
plt.ylabel('Accuracy')
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_376\2864430006.py:3:
FutureWarning:
```

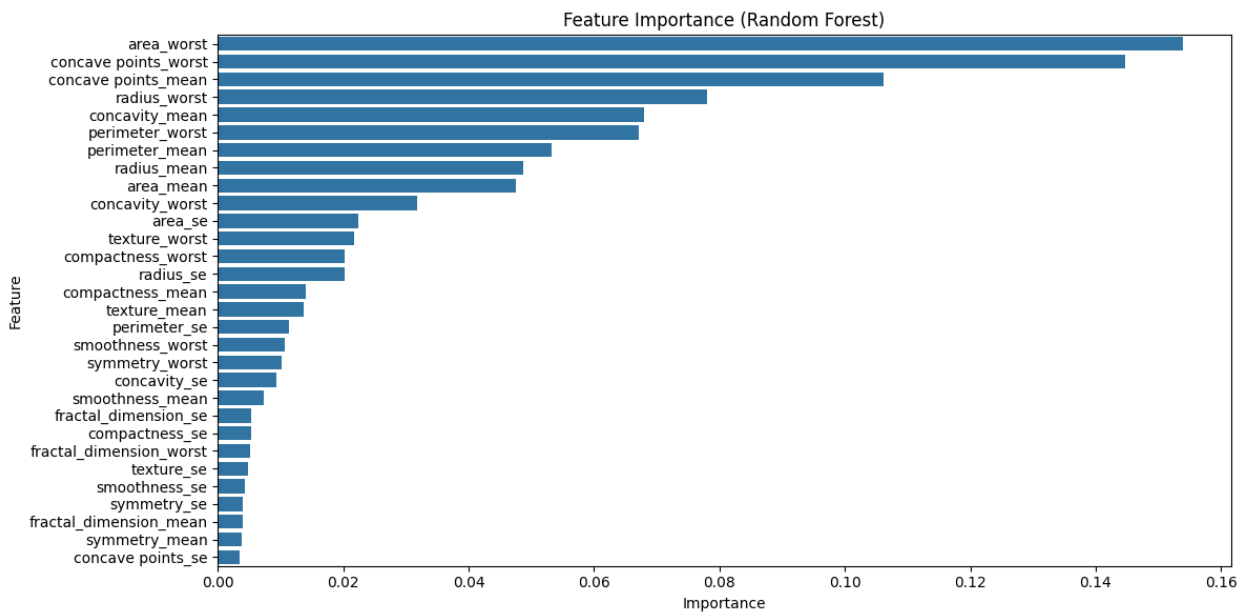
```
Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.
```

```
sns.barplot(x=list(results.keys()), y=list(results.values()),
palette='magma')
```



```
importances = models["Random Forest"].feature_importances_  
feature_names = X.columns  
indices = np.argsort(importances)[::-1]  
  
plt.figure(figsize=(12, 6))  
sns.barplot(x=importances[indices], y=feature_names[indices])  
plt.title('Feature Importance (Random Forest)')  
plt.xlabel('Importance')
```

```
plt.ylabel('Feature')
plt.tight_layout()
plt.show()
```



```
from sklearn.metrics import roc_auc_score, roc_curve

plt.figure(figsize=(8, 6))

for name, model in models.items():
    model.fit(X_train, y_train)

    # Check if model supports probability or decision scores
    if hasattr(model, "predict_proba"):
        y_scores = model.predict_proba(X_test)[:, 1] # probability
    for positive class
    elif hasattr(model, "decision_function"):
        y_scores = model.decision_function(X_test) # decision
    function scores
    else:
        print(f"{name} does not support probability estimates or
        decision function.")
        continue

    # Calculate AUC
    auc = roc_auc_score(y_test, y_scores)
    print(f"{name} AUC: {auc:.4f}")

    # Compute ROC curve
    fpr, tpr, thresholds = roc_curve(y_test, y_scores)

    # Plot ROC curve
```

```
plt.plot(fpr, tpr, label=f"{name} (AUC = {auc:.4f})")

# Plot the random chance line
plt.plot([0, 1], [0, 1], 'k--', label="Random Chance")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve Comparison')
plt.legend(loc='lower right')
plt.tight_layout()
plt.show()
```

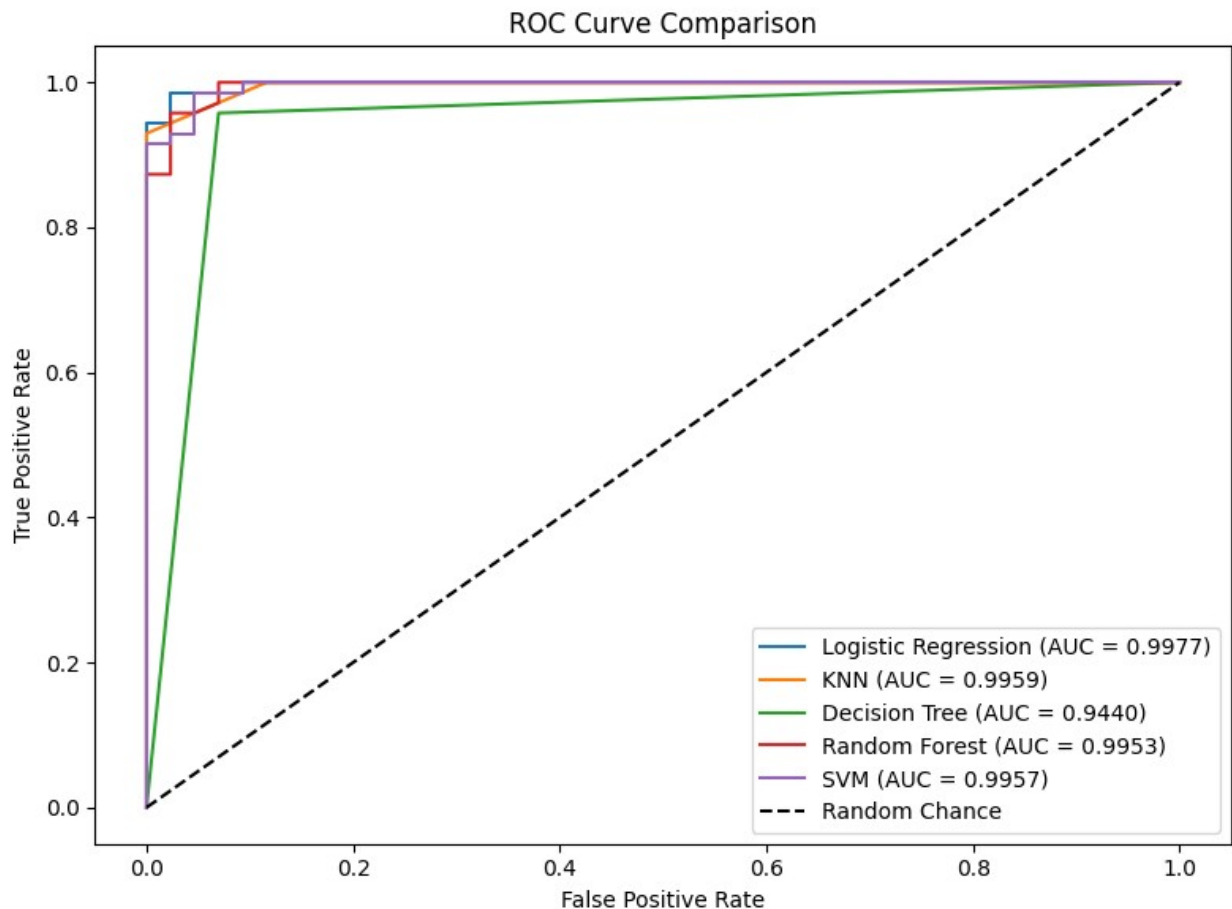
Logistic Regression AUC: 0.9977

KNN AUC: 0.9959

Decision Tree AUC: 0.9440

Random Forest AUC: 0.9953

SVM AUC: 0.9957




```
# Find the best model based on accuracy
best_model_name = max(results, key=results.get)
best_accuracy = results[best_model_name]

print(f"Best Model:{best_model_name}with Accuracy:
{best_accuracy:.4f}")
```

Best Model:Random Forestwith Accuracy:0.9649

```
# Save (pickle) best model
import pickle
with open('best_model_name.pkl','wb') as f:
    pickle.dump(best_model_name, f)
print("Best model saved as 'best_model_name.pkl'")
```

Best model saved as 'best_model_name.pkl'