In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
```

In [2]:
```
df  = pd.read_csv("TSNE_data.csv")
df
```

Out[2]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	M	17.99	10.38	122.80	1001.0	0.11840
1	M	20.57	17.77	132.90	1326.0	0.08474
2	M	19.69	21.25	130.00	1203.0	0.10960
3	M	11.42	20.38	77.58	386.1	0.14250
4	M	20.29	14.34	135.10	1297.0	0.10030
...	...	...	...	...	...	...
564	M	21.56	22.39	142.00	1479.0	0.11100
565	M	20.13	28.25	131.20	1261.0	0.09780
566	M	16.60	28.08	108.30	858.1	0.08455
567	M	20.60	29.33	140.10	1265.0	0.11780
568	B	7.76	24.54	47.92	181.0	0.05263

569 rows × 31 columns

In [3]:
```
df1 = df.drop(['diagnosis'],axis=1)
```

In [4]:
```
df['diagnosis'].unique()
```

Out[4]:
```
array(['M', 'B'], dtype=object)
```

In [5]:
```
df_labels = df.diagnosis
```

In [6]:
```
scaler = StandardScaler()
x_scaled = scaler.fit_transform(df1)
x_scaled
```

Out[6]:
```
array([[ 1.09706398, -2.07333501,  1.26993369, ...,  2.29607613,
         2.75062224,  1.93701461],
       [ 1.82982061, -0.35363241,  1.68595471, ...,  1.0870843 ,
        -0.24388967,  0.28118999],
       [ 1.57988811,  0.45618695,  1.56650313, ...,  1.95500035,
         1.152255  ,  0.20139121],
       ...,
       [ 0.70228425,  2.0455738 ,  0.67267578, ...,  0.41406869,
        -1.10454895, -0.31840916],
       [ 1.83834103,  2.33645719,  1.98252415, ...,  2.28998549,
         1.91908301,  2.21963528],
       [-1.80840125,  1.22179204, -1.81438851, ..., -1.74506282,
        -0.04813821, -0.75120669]])
```

In [7]:
```
from sklearn.decomposition import PCA
pca=PCA(n_components=3,random_state = 42)
pca_data1=pca.fit_transform(x_scaled)
```

In [8]:
```
pca_data1
```

Out[8]:
```
array([[ 9.19283683,  1.94858333, -1.12316483],
       [ 2.3878018 , -3.76817147, -0.52929058],
       [ 5.73389628, -1.07517381, -0.55174723],
       ...,
       [ 1.25617928, -1.90229684,  0.56272972],
       [10.37479406,  1.67200998, -1.87702989],
       [-5.4752433 , -0.6706363 ,  1.49044698]])
```

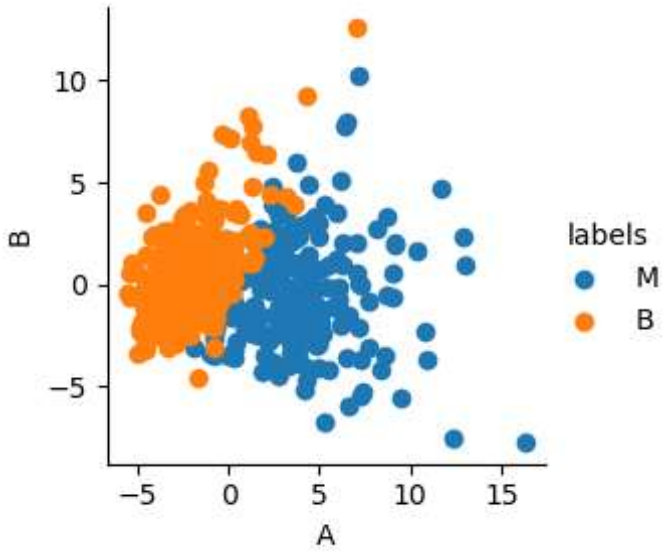
In [9]:
```
pca_data = np.column_stack((pca_data1, df_labels))
pca_df = pd.DataFrame(data=pca_data, columns=("A","B","C","labels"))
```

In [10]:
```
print(pca_df.head(10))
```

```
          A          B         C labels
0  9.192837   1.948583 -1.123165      M
1  2.387802  -3.768171 -0.529291      M
2  5.733896  -1.075174 -0.551747      M
3  7.122953  10.275589 -3.232788      M
4  3.935302  -1.948071  1.389769      M
5  2.380247   3.949929 -2.934877      M
6  2.238883  -2.690031 -1.639912      M
7  2.143299   2.340244 -0.871947      M
8  3.174924   3.391813 -3.119986      M
9  6.351747   7.727174 -4.341915      M
```

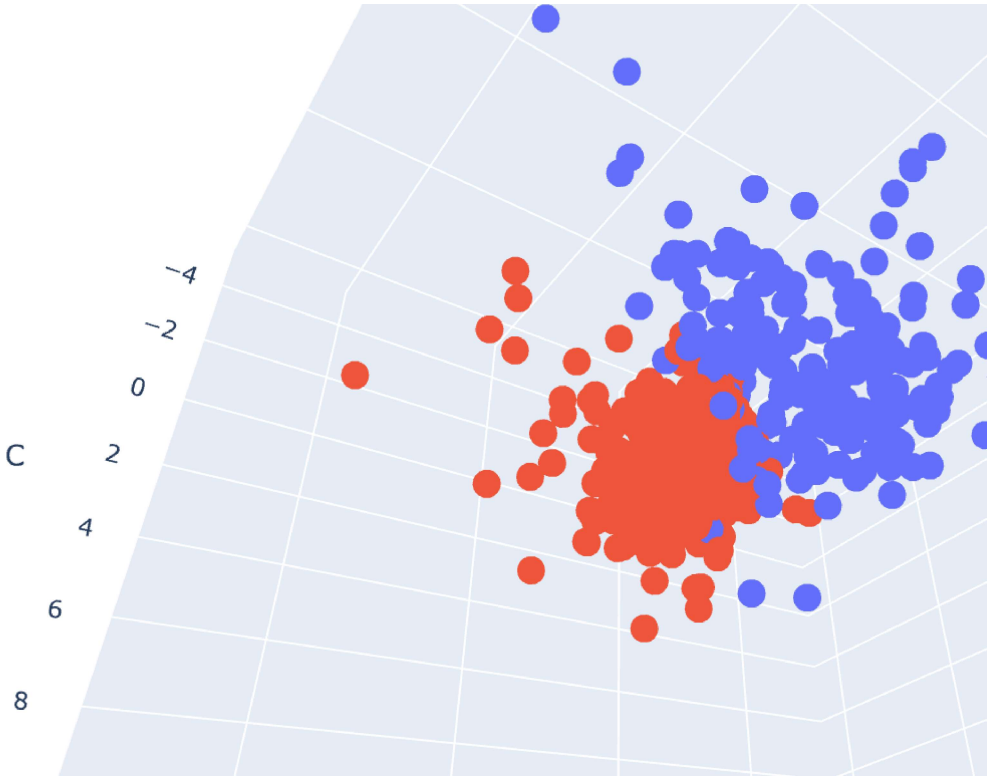
In [11]:

```
sns.FacetGrid(pca_df, hue="labels").map(plt.scatter, 'A', 'B').add_legend()
plt.show()
```



In [12]:
```
import plotly.express as px

fig = px.scatter_3d(pca_df, x=pca_df['A'], y=pca_df['B'], z=pca_df['C'],
                    color= pca_df['labels'])
fig.show()
```



T-SNE

In [13]:
```
t_sne = TSNE()
```

In [14]:
```
t_data1 = TSNE(n_components=3,perplexity=30).fit_transform(x_scaled)
```

In [15]:
```
t_data1
```

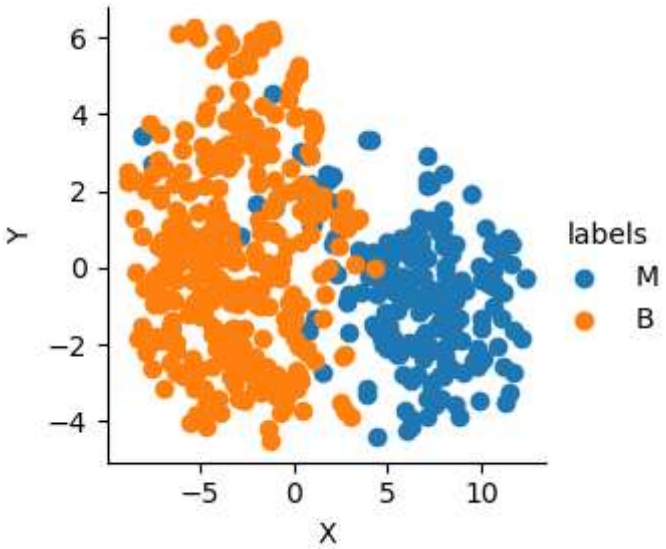
Out[15]:
```
array([[10.08635   , -2.3840137 ,  0.3878798 ],
       [ 5.8013787 ,  0.67179567,  4.693045  ],
       [ 8.852475  , -1.3143523 ,  1.7740929 ],
       ...,
       [ 3.9376345 ,  3.347712  ,  3.1025887 ],
       [10.872616  , -0.5450061 ,  0.23185869],
       [-7.1865973 ,  3.4996672 ,  3.3574657 ]], dtype=float32)
```

In [16]:
```
t_data = np.column_stack((t_data1, df_labels))
tsne_df = pd.DataFrame(data=t_data, columns=("X", "Y","Z" ,"labels"))
```

In [17]:
```
print(tsne_df.head(10))
```

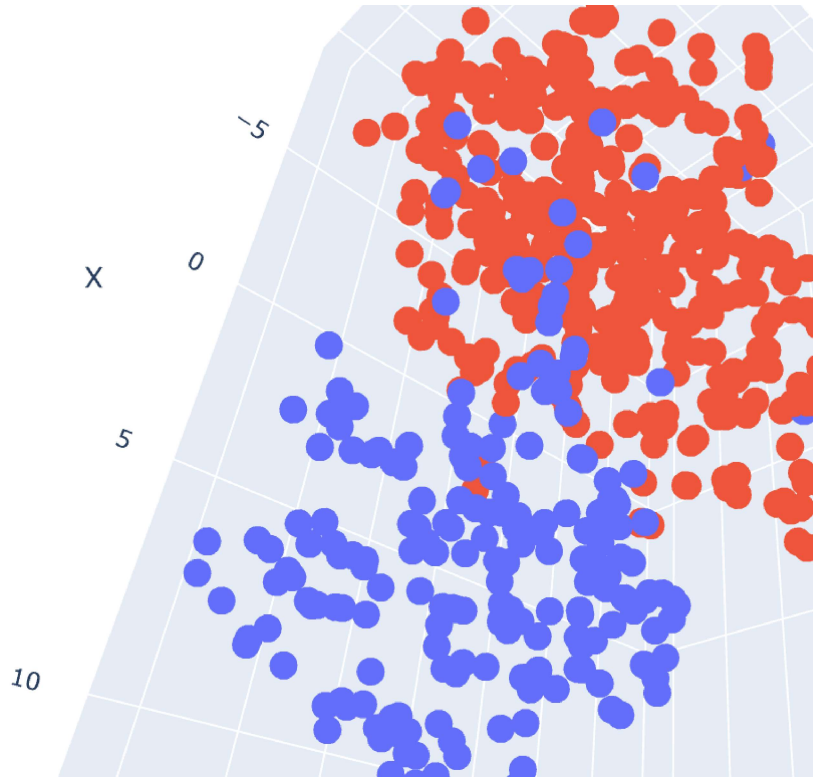
```
          X         Y         Z labels
0  10.08635 -2.384014   0.38788      M
1  5.801379  0.671796  4.693045      M
2  8.852475 -1.314352  1.774093      M
3  8.106648 -2.284597 -4.001067      M
4  6.580023 -2.768508  3.192085      M
5  5.744499 -0.276084 -3.567405      M
6  5.474233  0.426576  3.133465      M
7  5.112227 -1.646257 -3.011196      M
8  6.100016 -0.293798 -2.979045      M
9   7.99544 -0.751353 -3.964393      M
```

In [18]:
```
sns.FacetGrid(tsne_df, hue="labels" ).map(plt.scatter, 'X', 'Y').add_legend
plt.show()
```



In [19]:
```
import plotly.express as px

fig = px.scatter_3d(tsne_df, x=tsne_df['X'], y=tsne_df['Y'], z=tsne_df['Z']
                    color= tsne_df['labels'])
fig.show()
```



USE DBSCAN FOR CLUSTERING

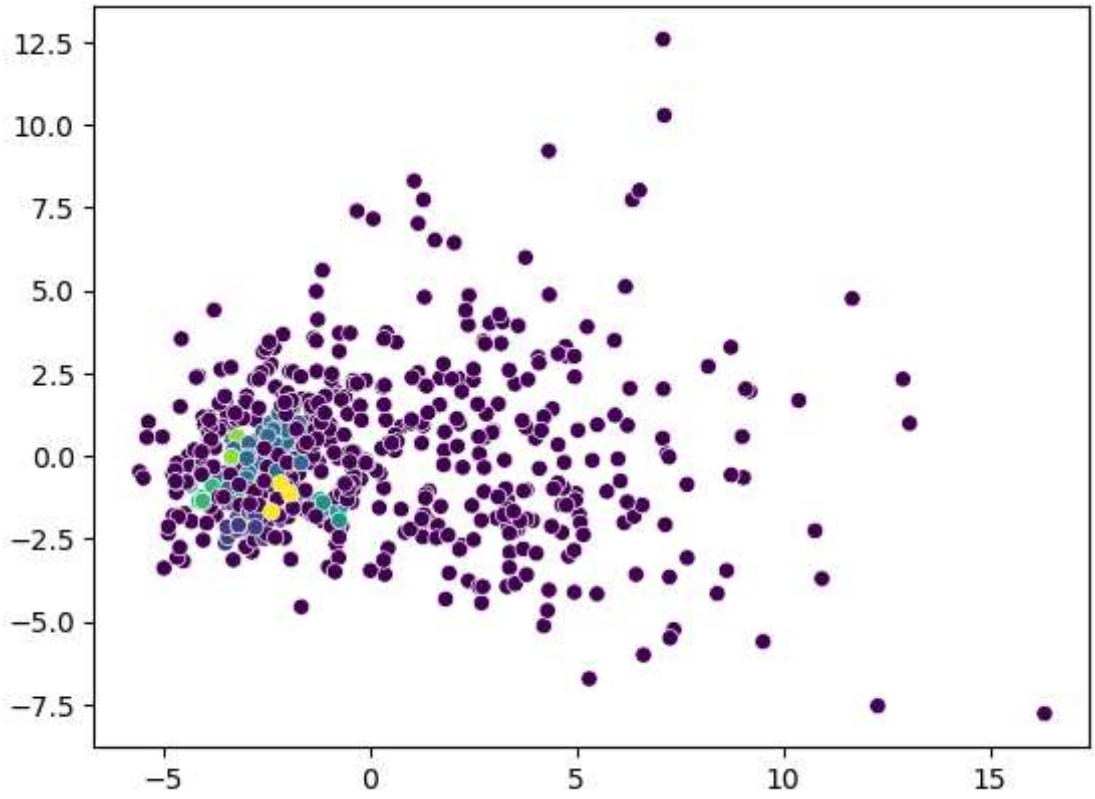
In [20]:
```
from sklearn.cluster import DBSCAN
dbcan=DBSCAN(eps=0.5)
dbcan.fit(pca_data1)
```

Out[20]:
```
DBSCAN
DBSCAN()
```

FOR PCA

In [21]:
```
import seaborn as sns
sns.scatterplot(x = pca_data1[:,0],y= pca_data1[:,1],c=dbcan.labels_)
```

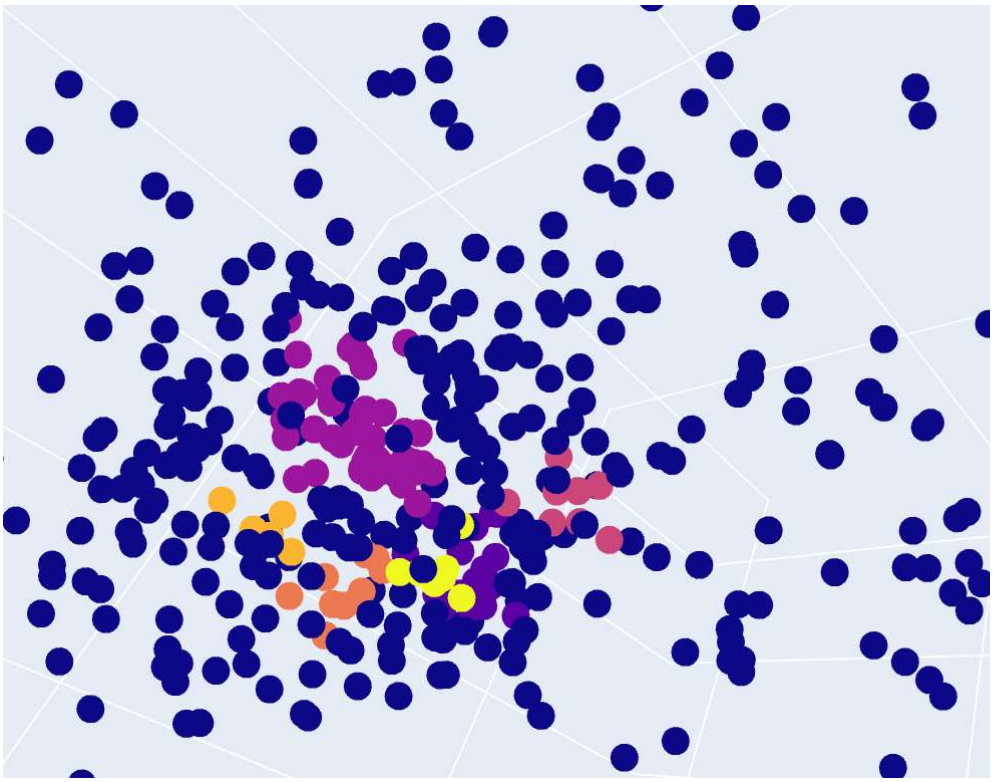
Out[21]: <Axes: >



In [22]:

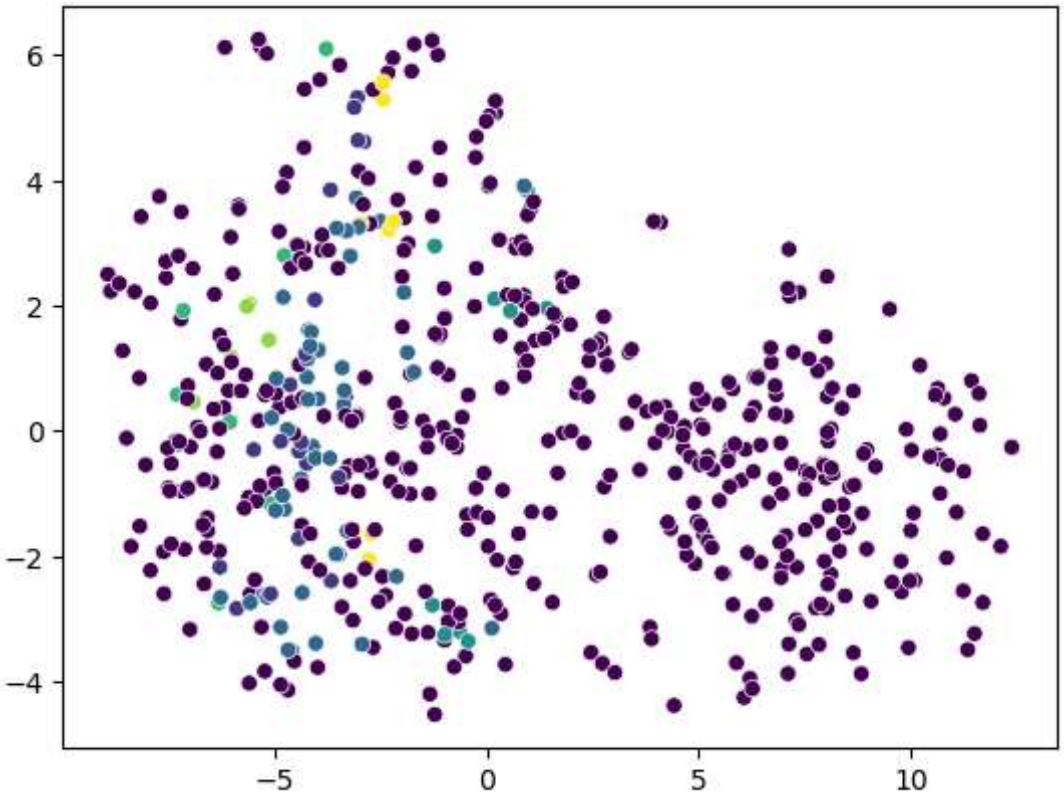
```
import plotly.express as px

fig = px.scatter_3d(pca_data1, x=pca_data1[:,0], y=pca_data1[:,1], z=pca_da
                    color=dbcan.labels_)
fig.show()
```

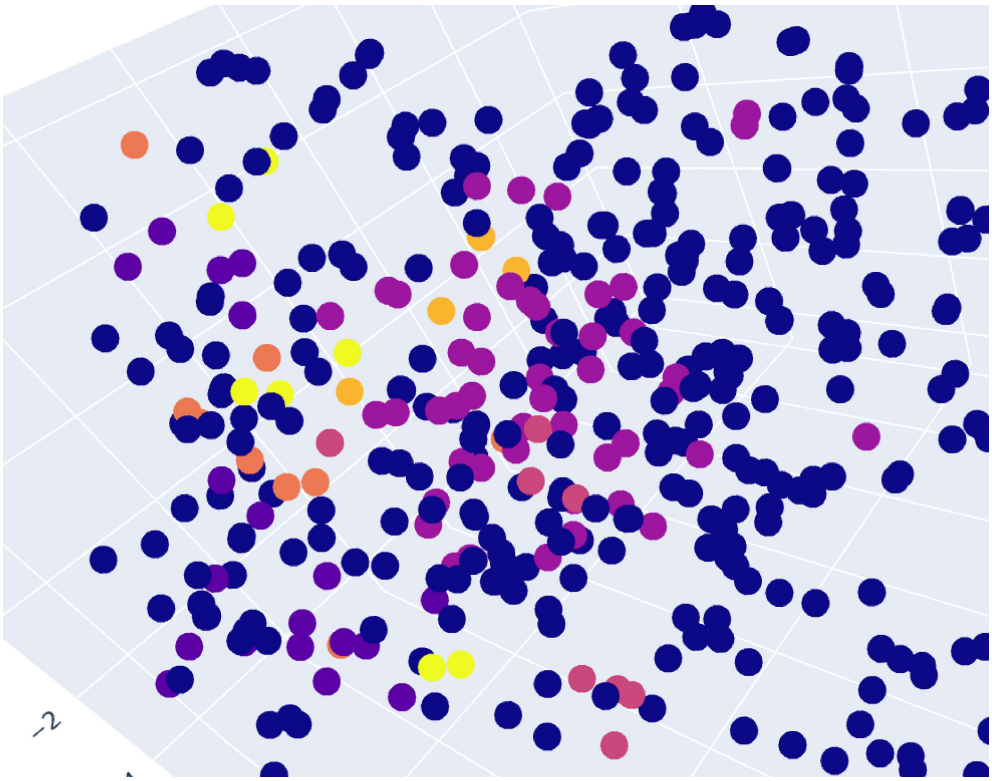


In [23]:
```
sns.scatterplot(x = t_data1[:,0],y= t_data1[:,1],c=dbcan.labels_)
```

Out[23]: <Axes: >



In [24]:
```
fig = px.scatter_3d(t_data1, x=t_data1[:,0], y=t_data1[:,1], z=t_data1[:,2
                    color=dbcan.labels_)
fig.show()
```



In [ ]: