VYSHNAVI VALLALA

2303A52429

BATCH-32

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

```
data = pd.read_csv('/content/breast_cancer_survival.csv')
data.head()
```

	Age	Gender	Protein1	Protein2	Protein3	Protein4	Tumour_Stage	Histology	ER status	PR status	HER2 status	Surgery_type	Date_of_Surgery
0	42	FEMALE	0.95256	2.15000	0.007972	-0.048340	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	20-May-18
1	54	FEMALE	0.00000	1.38020	-0.498030	-0.507320	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	26-Apr-18
2	63	FEMALE	-0.52303	1.76400	-0.370190	0.010815	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Lumpectomy	24-Aug-18
3	78	FEMALE	-0.87618	0.12943	-0.370380	0.132190	I	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	16-Nov-18
4	42	FEMALE	0.22611	1.74910	-0.543970	-0.390210	II	Infiltrating Ductal Carcinoma	Positive	Positive	Positive	Lumpectomy	12-Dec-18

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data.replace('FEMALE',0, inplace=True)
data.replace('MALE',1, inplace=True)
data.replace('Positive',1, inplace=True)
data.replace('Negative',0, inplace=True)
data.replace('Dead',0, inplace=True)
data.replace('Alive',1, inplace=True)
```

```
<ipython-input-3-709dcaf1cf2f>:2: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version
  data.replace('MALE',1, inplace=True)
<ipython-input-3-709dcaf1cf2f>:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version
  data.replace('Positive',1, inplace=True)
<ipython-input-3-709dcaf1cf2f>:4: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version
  data.replace('Negative',0, inplace=True)
<ipython-input-3-709dcaf1cf2f>:6: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version
  data.replace('Alive',1, inplace=True)
```

```
data.replace('II',2, inplace=True)
data.replace('III',3, inplace=True)
data.replace('I',1, inplace=True)
```

```
<ipython-input-4-fd5d96a82175>:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version
  data.replace('I',1, inplace=True)
data.replace('Infiltrating Ductal Carcinoma',1, inplace=True)
data.replace('Infiltrating Lobular Carcinoma',2, inplace=True)
data.replace('Mucinous Carcinoma',3, inplace=True)
```

```
<ipython-input-5-ecb44d251b39>:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future versior
  data.replace('Mucinous Carcinoma',3, inplace=True)
```

```
data.replace('Other',0, inplace=True)
data.replace('Lumpectomy',1, inplace=True)
data.replace('Modified Radical Mastectomy',2, inplace=True)
data.replace('Simple Mastectomy',3, inplace=True)
```

```
<ipython-input-6-f9216a2b26c7>:4: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future versior
  data.replace('Simple Mastectomy',3, inplace=True)
```

```
data.head()
```

	Age	Gender	Protein1	Protein2	Protein3	Protein4	Tumour_Stage	Histology	ER status	PR status	HER2 status	Surgery_type	Date_of_Surgery	D
0	42	0	0.95256	2.15000	0.007972	-0.048340	2	1	1	1	0	0	20-May-18	
1	54	0	0.00000	1.38020	-0.498030	-0.507320	2	1	1	1	0	0	26-Apr-18	
2	63	0	-0.52303	1.76400	-0.370190	0.010815	2	1	1	1	0	1	24-Aug-18	
3	78	0	-0.87618	0.12943	-0.370380	0.132190	1	1	1	1	0	0	16-Nov-18	
4	42	0	0.22611	1.74910	-0.543970	-0.390210	2	1	1	1	1	1	12-Dec-18	

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
x=data.drop(['Patient_Status','Date_of_Surgery','Date_of_Last_Visit'],axis=1)
y=data['Patient_Status']
```

```
y.isnull().sum()
y.fillna(0,inplace=True)
```

```
from imblearn.over_sampling import SMOTE
smote=SMOTE()
x,y=smote.fit_resample(x,y)
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

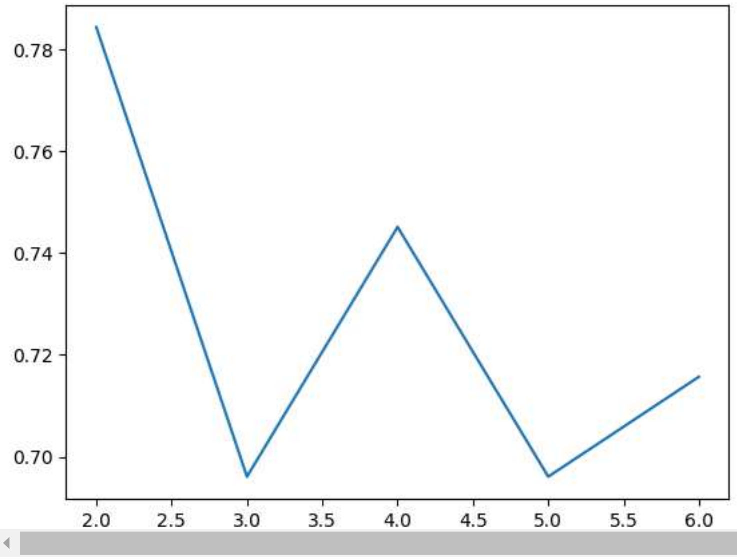
before PCA

KNN

```
accuracy_list=[]
l=[]
for i in range(1,6):
  bkn=KNeighborsClassifier(n_neighbors=i)
  bkn.fit(x_train,y_train)
  accuracy_list.append([bkn.score(x_test,y_test)])
  l.append(i+1)
```

```
plt.plot(l,accuracy_list)
```

```
[<matplotlib.lines.Line2D at 0x7d619e5060e0>]
```



SVM

```
import torch
# Check for GPU
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
print(f"Using device: {device}")
```

```
Using device: cpu
```

```
svc=SVC()
# Initialize the model
#svc.to(device)
svc.fit(x_train,y_train)
```

```
▾ SVC
SVC()
```

```
y_pred=svc.predict(x_test)
accuracy_svc=accuracy_score(y_test,y_pred)
```

```
print(accuracy_svc)
```

```
0.49019607843137253
```

```
c_report =classification_report(y_test,y_pred)
print(c_report)
```

```
              precision    recall  f1-score   support

         0.0       0.48      0.24      0.32        51
         1.0       0.49      0.75      0.59        51

    accuracy                           0.49       102
   macro avg       0.49      0.49      0.45       102
weighted avg       0.49      0.49      0.45       102
```

```
c_m=confusion_matrix(y_test,y_pred)
print(c_m)
```

```
[[12 39]
 [13 38]]
```

LOGISTIC REGRESSION

```
lg=LogisticRegression()
lg.fit(x_train,y_train)
```

```
▾ LogisticRegression ⓘ ?
LogisticRegression()
```

```
y_pred=lg.predict(x_test)
accuracy_lg=accuracy_score(y_test,y_pred)
```

```
print(accuracy_lg)
```

```
0.5980392156862745
```

```
l_report =classification_report(y_test,y_pred)
print(l_report)
```

```
              precision    recall  f1-score   support

         0.0       0.61      0.55      0.58        51
         1.0       0.59      0.65      0.62        51

    accuracy                           0.60       102
   macro avg       0.60      0.60      0.60       102
weighted avg       0.60      0.60      0.60       102
```

```
cml=confusion_matrix(y_test,y_pred)
print(cml)
```

```
[[28 23]
 [18 33]]
```

PCA

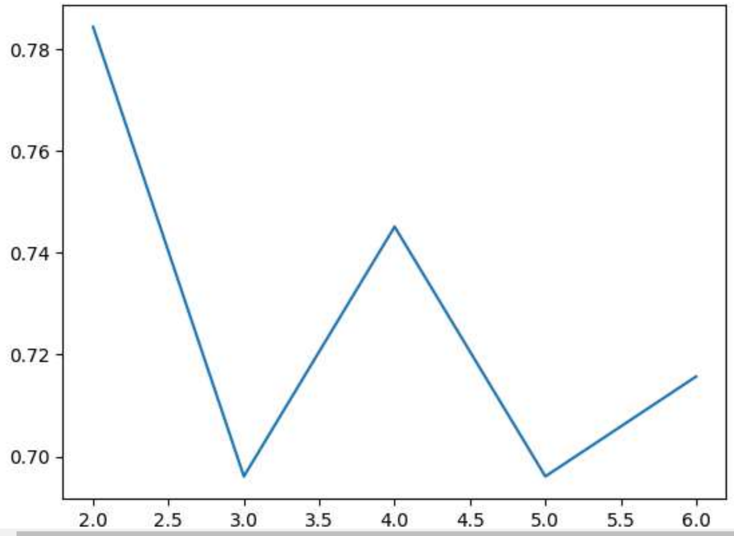
```
from sklearn.decomposition import PCA
pca=PCA(n_components=10)
x_train_pca=pca.fit_transform(x_train)
x_test_pca=pca.transform(x_test)
```

KNN

```
accuracy_list=[]
l=[]
for i in range(1,6):
  bkn=KNeighborsClassifier(n_neighbors=i)
  bkn.fit(x_train_pca,y_train)
  accuracy_list.append([bkn.score(x_test_pca,y_test)])
  l.append(i+1)
```

```
plt.plot(l,accuracy_list)
```

```
[<matplotlib.lines.Line2D at 0x7d6104965a50>]
```



SVM

```
svc=SVC()
# Initialize the model
#svc.to(device)
svc.fit(x_train_pca,y_train)
```

```
▾ SVC ⓘ ?
SVC()
```

```
y_pred=svc.predict(x_test_pca)
accuracy_svc=accuracy_score(y_test,y_pred)
```

```
print(accuracy_svc)
```

```
0.5784313725490197
```

```
c_report =classification_report(y_test,y_pred)
print(c_report)
```

```
              precision    recall  f1-score   support

         0.0       0.57      0.61      0.59        51
         1.0       0.58      0.55      0.57        51

    accuracy                           0.58       102
   macro avg       0.58      0.58      0.58       102
weighted avg       0.58      0.58      0.58       102
```

```
c_m=confusion_matrix(y_test,y_pred)
print(c_m)
```

```
[[31 20]
 [23 28]]
```

LOGISTIC REGRESSION

```
lg=LogisticRegression()
lg.fit(x_train_pca,y_train)
```

```
▾ LogisticRegression ⓘ ?
LogisticRegression()
```

```
y_pred=lg.predict(x_test_pca)
accuracy_lg=accuracy_score(y_test,y_pred)
```

```
print(accuracy_lg)
```

```
0.5980392156862745
```

```
cl_report =classification_report(y_test,y_pred)
print(cl_report)
```

```
              precision    recall  f1-score   support

         0.0       0.61      0.55      0.58        51
         1.0       0.59      0.65      0.62        51

    accuracy                           0.60       102
   macro avg       0.60      0.60      0.60       102
weighted avg       0.60      0.60      0.60       102
```