VYSHNAVI VALLALA

2303A52429

BATCH-32

KNN

```
import pandas as pd
import numpy as np
```

```
data = pd.read_csv('/content/breast_cancer_survival (2).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-4-709dcaf1cf2f>:2: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future versior
  data.replace('MALE',1, inplace=True)
<ipython-input-4-709dcaf1cf2f>:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future versior
  data.replace('Positive',1, inplace=True)
<ipython-input-4-709dcaf1cf2f>:4: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future versior
  data.replace('Negative',0, inplace=True)
<ipython-input-4-709dcaf1cf2f>:6: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future versior
  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-5-fd5d96a82175>:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future versior
  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-6-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-7-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.shape
```

```
(510, 12)
```

```
y.shape
```

```
(510,)
```

```
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=0)
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
accuracy_list=[]
for i in range(1,101):
  bkn=KNeighborsClassifier(n_neighbors=i)
  bkn.fit(x_train,y_train)
  accuracy_list.append([bkn.score(x_test,y_test)])
```

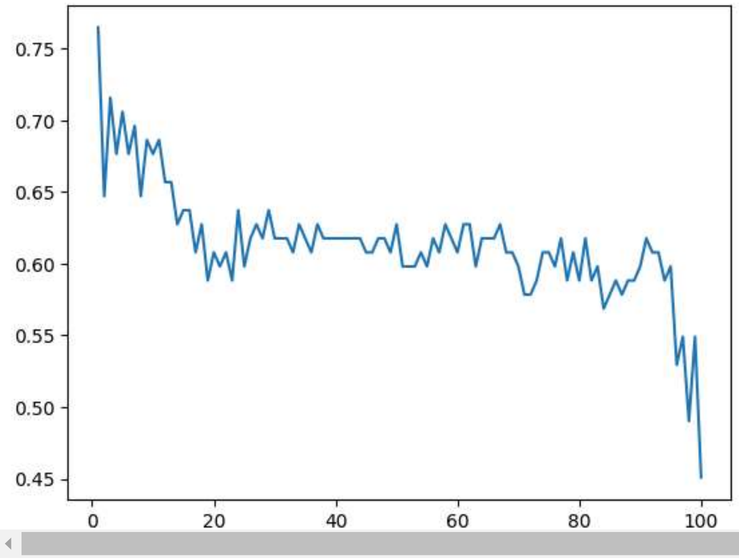
```
l=[]
for i in range(len(accuracy_list)):
  print(accuracy_list[i])
  l.append(i+1)
```

```
[0.6078431372549019]
[0.6274509803921569]
[0.5980392156862745]
[0.5980392156862745]
[0.5980392156862745]
[0.6078431372549019]
[0.5980392156862745]
[0.6176470588235294]
[0.6078431372549019]
[0.6274509803921569]
[0.6176470588235294]
[0.6078431372549019]
[0.6274509803921569]
[0.6274509803921569]
[0.5980392156862745]
[0.6176470588235294]
[0.6176470588235294]
[0.6176470588235294]
[0.6274509803921569]
[0.6078431372549019]
[0.6078431372549019]
[0.5980392156862745]
[0.5784313725490197]
[0.5784313725490197]
[0.5882352941176471]
[0.6078431372549019]
[0.6078431372549019]
[0.5980392156862745]
[0.6176470588235294]
[0.5882352941176471]
[0.6078431372549019]
[0.5882352941176471]
[0.6176470588235294]
[0.5882352941176471]
[0.5980392156862745]
[0.5686274509803921]
[0.5784313725490197]
[0.5882352941176471]
[0.5784313725490197]
[0.5882352941176471]
[0.5882352941176471]
[0.5980392156862745]
[0.6176470588235294]
[0.6078431372549019]
[0.6078431372549019]
[0.5882352941176471]
[0.5980392156862745]
[0.5294117647058824]
[0.5490196078431373]
[0.49019607843137253]
[0.5490196078431373]
[0.45098039215686275]
```

```
import matplotlib.pyplot as plt
```

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

```
[<matplotlib.lines.Line2D at 0x7b565220da50>]
```



SVC

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

```
l=[0.20,0.25,0.30,0.35]
```

```
from sklearn.svm import SVC
sm=SVC(kernel='linear')
```

```
accuracy_list1=[]
accuracy_list2=[]
reports=[]
metrics=[]
```

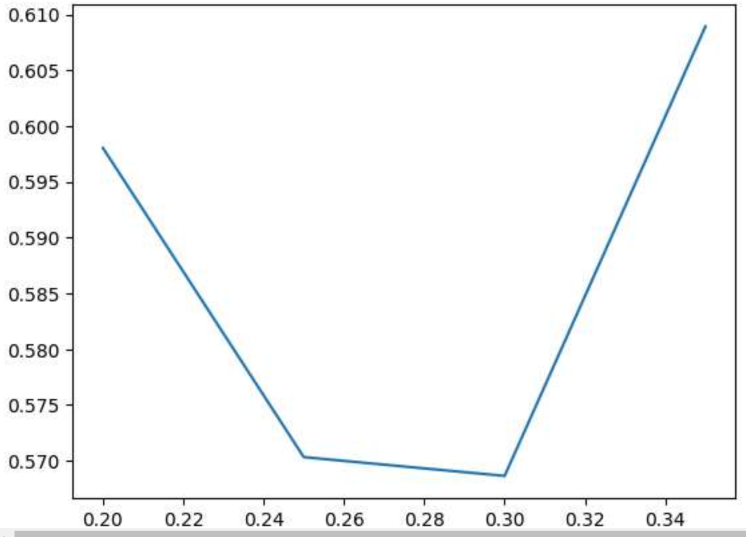
```
for i in l:
    x_train1,x_test1,y_train1,y_test1=train_test_split(x,y,test_size=i,random_state=42)
    sm.fit(x_train1,y_train1)
    y_pred=sm.predict(x_test1)
    accuracy_list2.append(accuracy_score(y_test1,y_pred))
    reports.append(classification_report(y_test1,y_pred))
    metrics.append(confusion_matrix(y_test1,y_pred))
    accuracy_list1.append([sm.score(x_test1,y_test1)])
```

```
print(accuracy_list1)
```

```
[[0.5980392156862745], [0.5703125], [0.5686274509803921], [0.6089385474860335]]
```

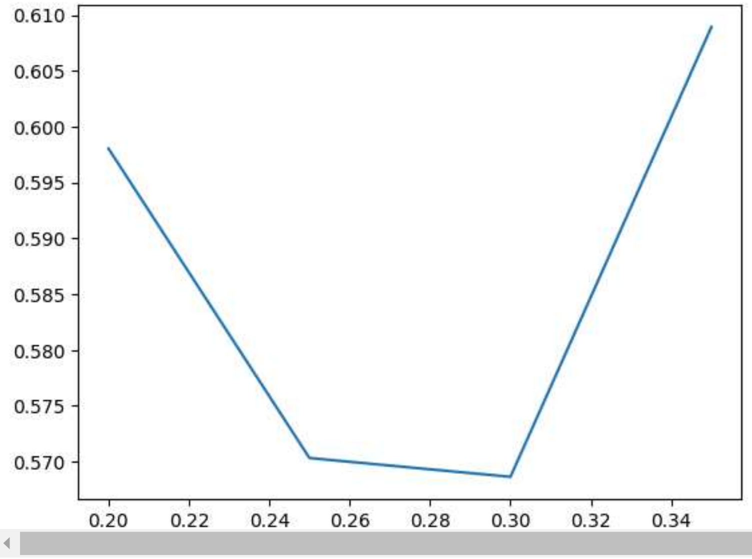
```
plt.plot(l,accuracy_list1)
```

```
[<matplotlib.lines.Line2D at 0x7b564f06b5e0>]
```



```
plt.plot(l,accuracy_list2)
```

```
[<matplotlib.lines.Line2D at 0x7b56530da2f0>]
```



```
print(reports[1])
```

```
              precision    recall  f1-score   support

         0.0       0.59      0.63      0.61        68
         1.0       0.55      0.50      0.52        60

    accuracy                           0.57       128
   macro avg       0.57      0.57      0.57       128
weighted avg       0.57      0.57      0.57       128
```