titanic-survival-prediction

December 10, 2023

Titanic Survival Prediction

Objective:

The aim of the project is to build a prediction model that predicts whether a passenger on the Titanic survived or not.

Importing the libraries and dataset

[ ]:

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv('/content/tested.csv')
df
```

[ ]:

```
     PassengerId  Survived  Pclass  \
0            892         0       3
1            893         1       3
2            894         0       2
3            895         0       3
4            896         1       3
..           …        …     …
413         1305         0       3
414         1306         1       1
415         1307         0       3
416         1308         0       3
417         1309         0       3

                                             Name     Sex   Age  SibSp  Parch  \
0                                Kelly, Mr. James    male  34.5      0      0
1                Wilkes, Mrs. James (Ellen Needs)  female  47.0      1      0
2                       Myles, Mr. Thomas Francis    male  62.0      0      0
3                                Wirz, Mr. Albert    male  27.0      0      0
4    Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1      1
..                                            …     …   …    …    …
413                            Spector, Mr. Woolf    male   NaN      0      0
414                  Oliva y Ocana, Dona. Fermina  female  39.0      0      0
415                  Saether, Mr. Simon Sivertsen    male  38.5      0      0
```

```
416                           Ware, Mr. Frederick    male   NaN      0      0
417                      Peter, Master. Michael J    male   NaN      1      1

                 Ticket      Fare Cabin Embarked
0                330911    7.8292   NaN        Q
1                363272    7.0000   NaN        S
2                240276    9.6875   NaN        Q
3                315154    8.6625   NaN        S
4               3101298   12.2875   NaN        S
..                  …       …   …      …
413           A.5. 3236    8.0500   NaN        S
414            PC 17758  108.9000  C105        C
415  SOTON/O.Q. 3101262    7.2500   NaN        S
416              359309    8.0500   NaN        S
417                2668   22.3583   NaN        C

[418 rows x 12 columns]
```

Datapreprocessing

[ ]:
```
df.head()
```

[ ]:
```
   PassengerId  Survived  Pclass  \
0          892         0       3
1          893         1       3
2          894         0       2
3          895         0       3
4          896         1       3

                                           Name     Sex   Age  SibSp  Parch  \
0                              Kelly, Mr. James    male  34.5      0      0
1              Wilkes, Mrs. James (Ellen Needs)  female  47.0      1      0
2                     Myles, Mr. Thomas Francis    male  62.0      0      0
3                              Wirz, Mr. Albert    male  27.0      0      0
4  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1      1

    Ticket     Fare Cabin Embarked
0   330911   7.8292   NaN        Q
1   363272   7.0000   NaN        S
2   240276   9.6875   NaN        Q
3   315154   8.6625   NaN        S
4  3101298  12.2875   NaN        S
```

[ ]:
```
df.tail()
```

[ ]:
```
     PassengerId  Survived  Pclass                          Name     Sex  \
413         1305         0       3            Spector, Mr. Woolf    male
```

```
414         1306         1        1  Oliva y Ocana, Dona. Fermina  female
415         1307         0        3  Saether, Mr. Simon Sivertsen    male
416         1308         0        3           Ware, Mr. Frederick    male
417         1309         0        3      Peter, Master. Michael J    male

      Age  SibSp  Parch              Ticket      Fare Cabin Embarked
413   NaN      0      0           A.5. 3236    8.0500   NaN        S
414  39.0      0      0            PC 17758  108.9000  C105        C
415  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN        S
416   NaN      0      0              359309    8.0500   NaN        S
417   NaN      1      1                2668   22.3583   NaN        C
```

```
[ ]: df.dtypes
```

```
[ ]: PassengerId      int64
     Survived         int64
     Pclass           int64
     Name            object
     Sex             object
     Age            float64
     SibSp            int64
     Parch            int64
     Ticket          object
     Fare           float64
     Cabin           object
     Embarked        object
     dtype: object
```

```
[ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Survived     418 non-null    int64
 2   Pclass       418 non-null    int64
 3   Name         418 non-null    object
 4   Sex          418 non-null    object
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64
 7   Parch        418 non-null    int64
 8   Ticket       418 non-null    object
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object
 11  Embarked     418 non-null    object
```

```
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

```
[ ]: df.columns
```

```
[ ]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
            'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
           dtype='object')
```

```
[ ]: df.describe()
```

```
[ ]:        PassengerId    Survived      Pclass         Age       SibSp  \
     count   418.000000  418.000000  418.000000  332.000000  418.000000
     mean   1100.500000    0.363636    2.265550   30.272590    0.447368
     std     120.810458    0.481622    0.841838   14.181209    0.896760
     min     892.000000    0.000000    1.000000    0.170000    0.000000
     25%     996.250000    0.000000    1.000000   21.000000    0.000000
     50%    1100.500000    0.000000    3.000000   27.000000    0.000000
     75%    1204.750000    1.000000    3.000000   39.000000    1.000000
     max    1309.000000    1.000000    3.000000   76.000000    8.000000

                 Parch        Fare
     count  418.000000  417.000000
     mean     0.392344   35.627188
     std      0.981429   55.907576
     min      0.000000    0.000000
     25%      0.000000    7.895800
     50%      0.000000   14.454200
     75%      0.000000   31.500000
     max      9.000000  512.329200
```

Data Cleaning

```
[ ]: df.duplicated().sum()
```

```
[ ]: 0
```

```
[ ]: df1=df.drop(['PassengerId','Name','Ticket','Cabin'],axis=1)
     df1
```

```
[ ]:      Survived  Pclass     Sex   Age  SibSp  Parch     Fare Embarked
     0           0       3    male  34.5      0      0   7.8292        Q
     1           1       3  female  47.0      1      0   7.0000        S
     2           0       2    male  62.0      0      0   9.6875        Q
     3           0       3    male  27.0      0      0   8.6625        S
     4           1       3  female  22.0      1      1  12.2875        S
     ..        …     …     …   …    …    …      …      …
     413         0       3    male   NaN      0      0   8.0500        S
```

```
414         1       1  female  39.0       0       0  108.9000        C
415         0       3    male  38.5       0       0    7.2500        S
416         0       3    male   NaN       0       0    8.0500        S
417         0       3    male   NaN       1       1   22.3583        C

[418 rows x 8 columns]
```

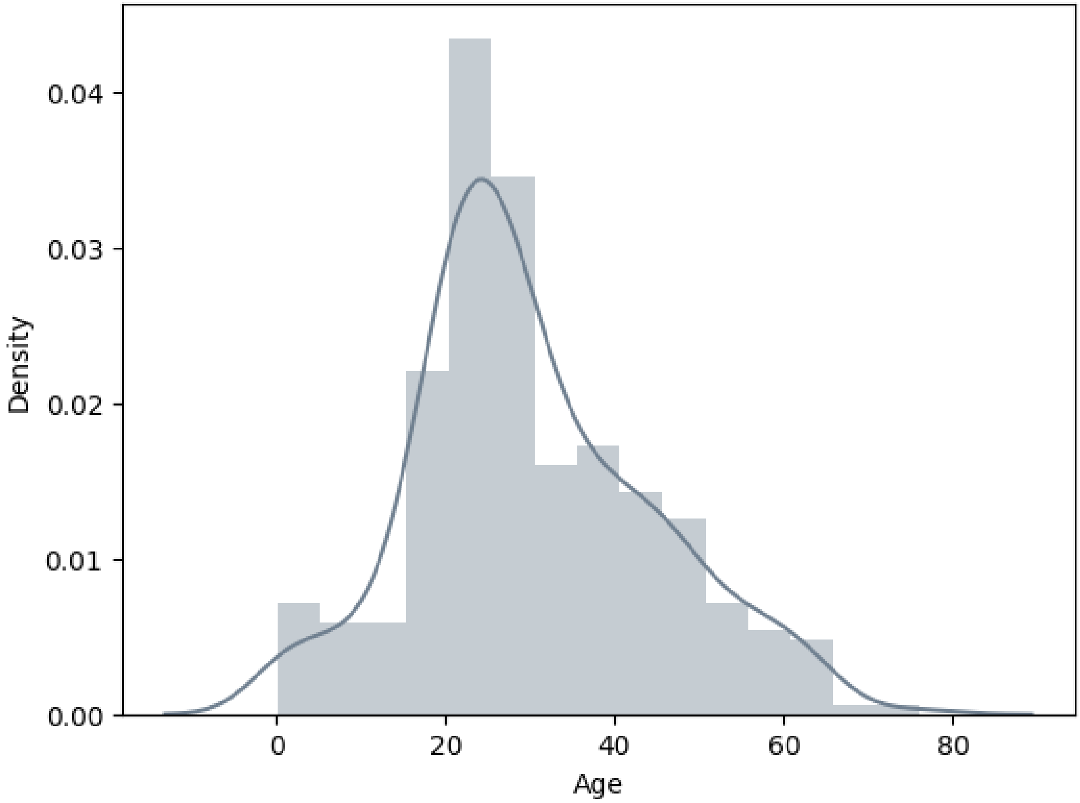
[ ]:
```
df1.isna().sum()
```

[ ]:
```
Survived     0
Pclass       0
Sex          0
Age         86
SibSp        0
Parch        0
Fare         1
Embarked     0
dtype: int64
```

[ ]:
```
sns.distplot(df1['Age'],color='slategray')
```

```
<ipython-input-11-d8f1a3847997>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df1['Age'],color='slategray')
```

[ ]:
```
<Axes: xlabel='Age', ylabel='Density'>
```



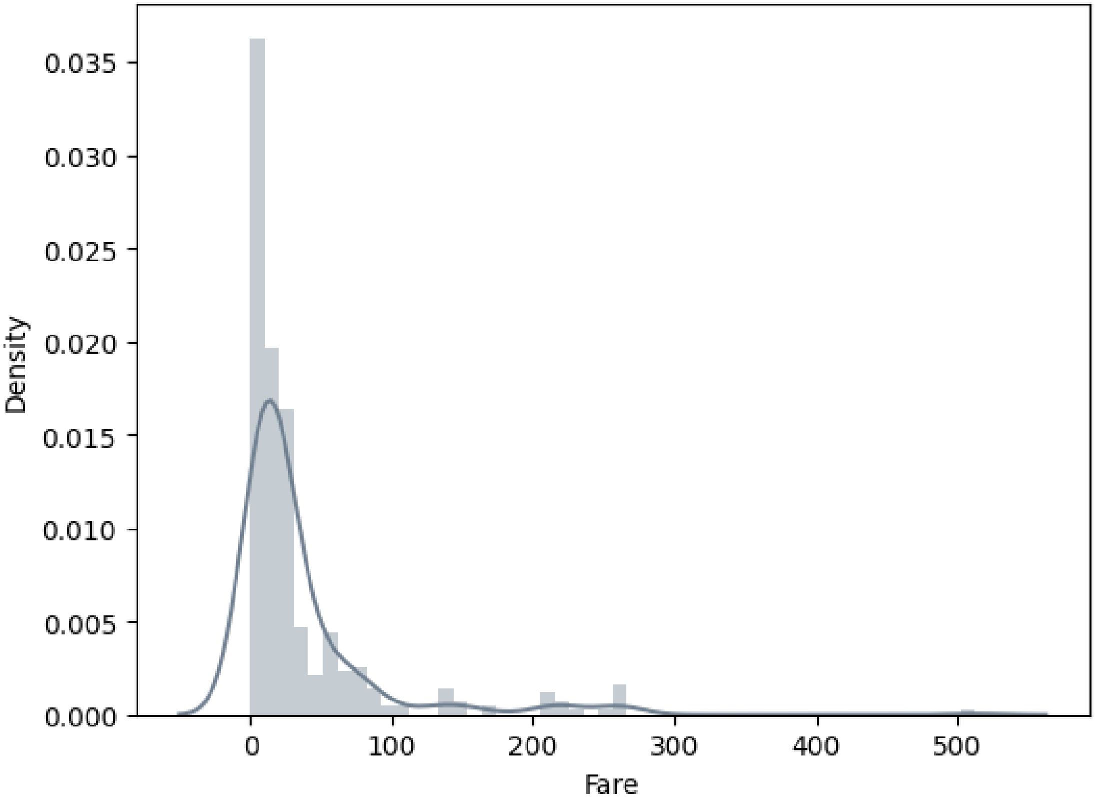
[ ]:
```
x=df1['Age'].mean()
print(x)
df1['Age'].fillna(x,inplace=True)
print(df1)
```

```
30.272590361445783
     Survived  Pclass     Sex       Age  SibSp  Parch      Fare Embarked
0           0       3    male  34.50000      0      0    7.8292        Q
1           1       3  female  47.00000      1      0    7.0000        S
2           0       2    male  62.00000      0      0    9.6875        Q
3           0       3    male  27.00000      0      0    8.6625        S
4           1       3  female  22.00000      1      1   12.2875        S
..        …       …       …        …    …      …     …       …
413         0       3    male  30.27259      0      0    8.0500        S
414         1       1  female  39.00000      0      0  108.9000        C
415         0       3    male  38.50000      0      0    7.2500        S
416         0       3    male  30.27259      0      0    8.0500        S
417         0       3    male  30.27259      1      1   22.3583        C

[418 rows x 8 columns]
```

[ ]:
```
sns.distplot(df1['Fare'],color='slategray')
```

```
<ipython-input-13-3d4b70d6366b>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df1['Fare'],color='slategray')
```

[ ]:
```
<Axes: xlabel='Fare', ylabel='Density'>
```



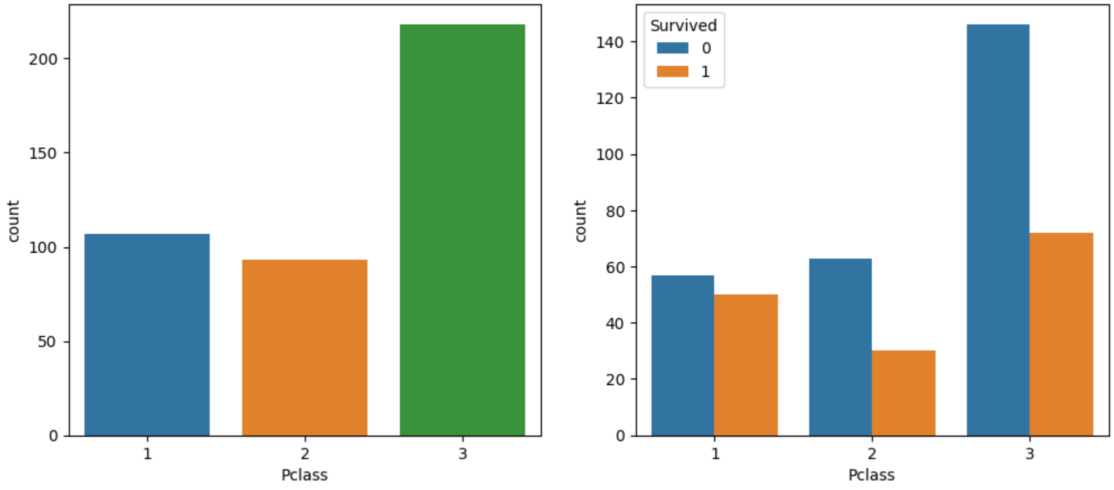
[ ]:
```
x=df1['Fare'].mean()
print(x)
df1['Fare'].fillna(x,inplace=True)
print(df1)
```

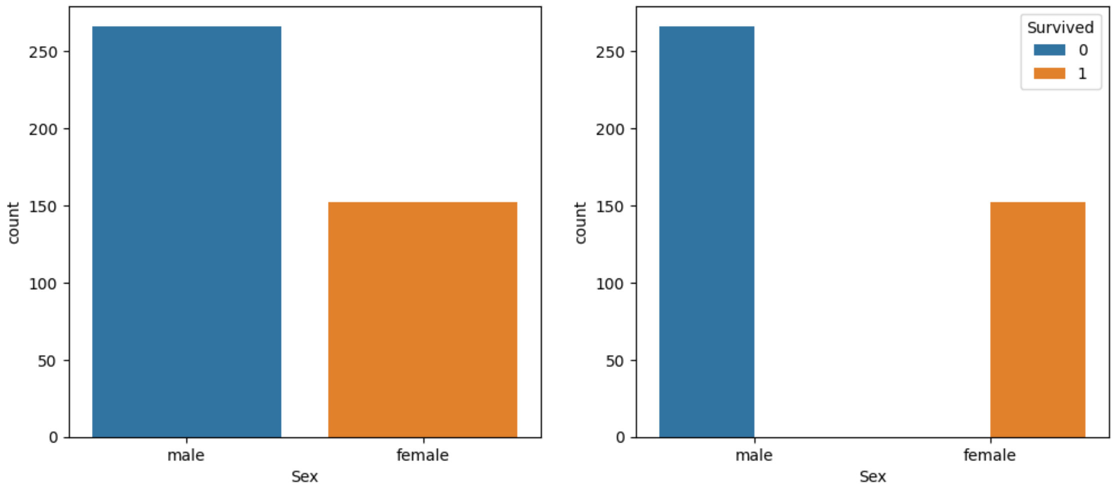
```
35.627188489208635
```

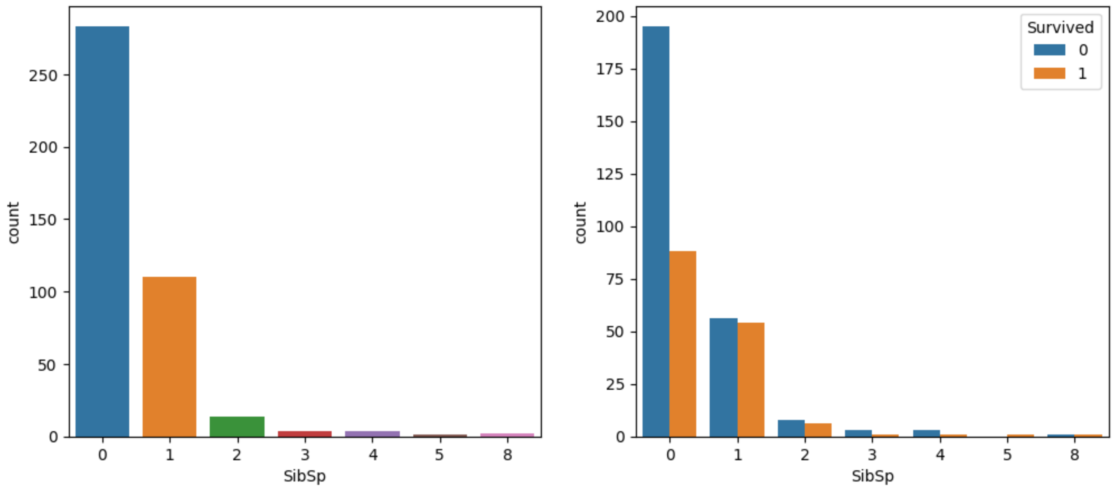
```
     Survived  Pclass     Sex       Age  SibSp  Parch      Fare Embarked
0           0       3    male  34.50000      0      0    7.8292        Q
1           1       3  female  47.00000      1      0    7.0000        S
2           0       2    male  62.00000      0      0    9.6875        Q
3           0       3    male  27.00000      0      0    8.6625        S
4           1       3  female  22.00000      1      1   12.2875        S
..        …       …       …         …      …      …       …      …
413         0       3    male  30.27259      0      0    8.0500        S
414         1       1  female  39.00000      0      0  108.9000        C
415         0       3    male  38.50000      0      0    7.2500        S
416         0       3    male  30.27259      0      0    8.0500        S
417         0       3    male  30.27259      1      1   22.3583        C

[418 rows x 8 columns]
```

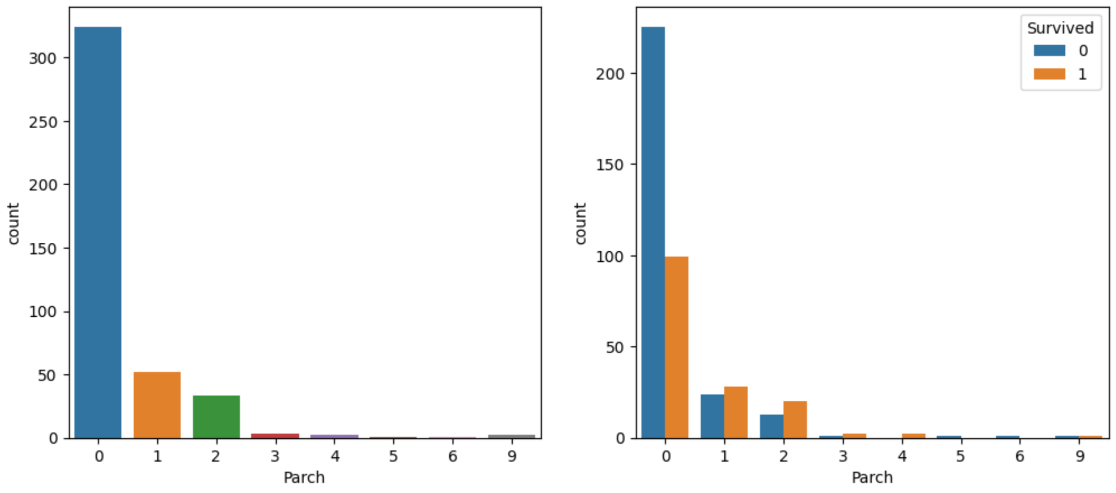
Data Visualization

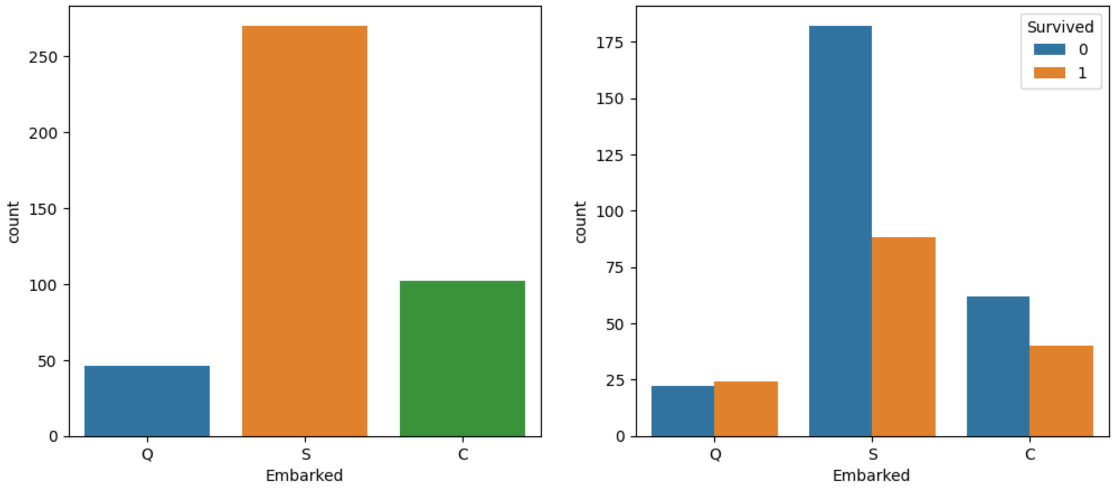
```
[ ]: for column in ['Pclass','Sex','SibSp','Parch','Embarked']:
       plt.figure(figsize=(12,5))
       plt.subplot(1,2,1)
       sns.countplot(data=df1,x=column)

       plt.subplot(1,2,2)
       sns.countplot(data=df1,x=column,hue='Survived')
       plt.show()
```




count
250
200
150
100
50
0
male
female
Sex
Survived
0
1
count
250
200
150
100
50
0
male
female
Sex

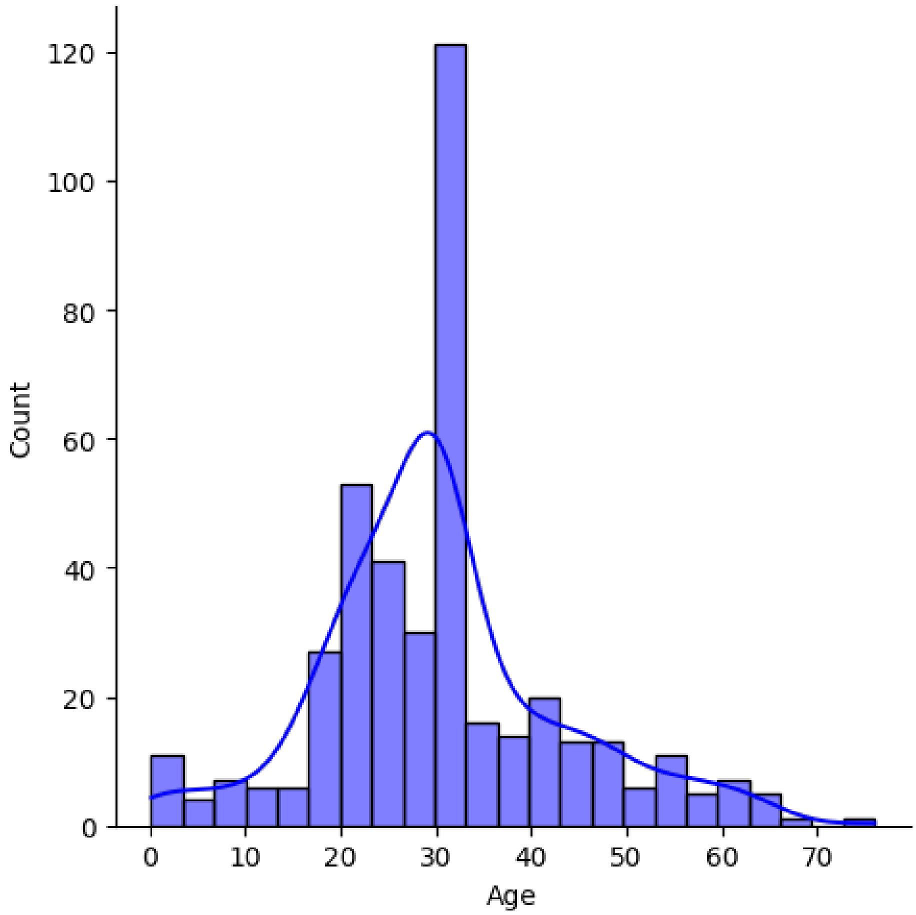

count
250
200
150
100
50
0
0
1
2
3
4
5
8
SibSp
Survived
0
1
count
200
175
150
125
100
75
50
25
0
0
1
2
3
4
5
8
SibSp


count
300
250
200
150
100
50
0
0
1
2
3
4
5
6
9
Parch
Survived
0
1
count
200
150
100
50
0
0
1
2
3
4
5
6
9
Parch



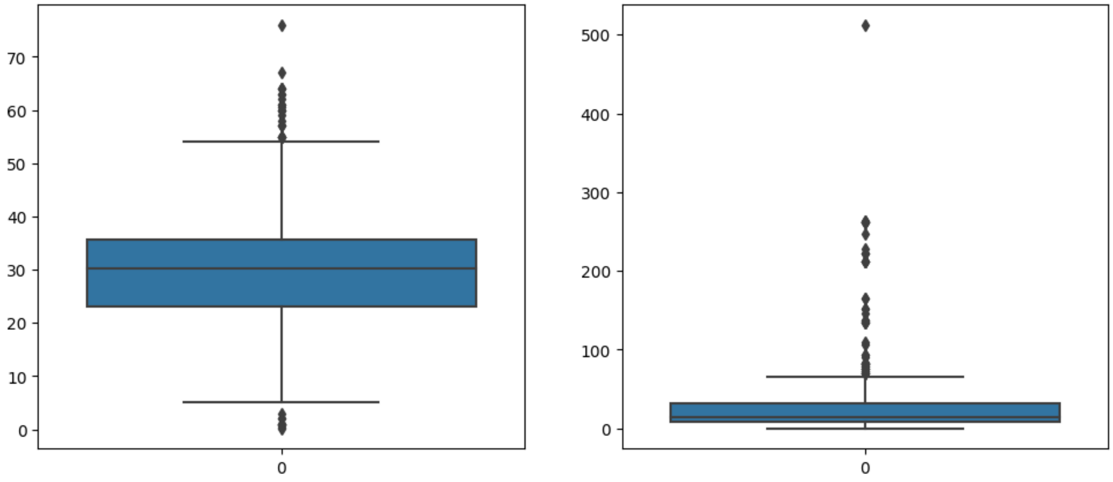
```
[ ]: sns.displot(df1['Age'],kde=True,color='Blue')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7e31e9cac340>
```



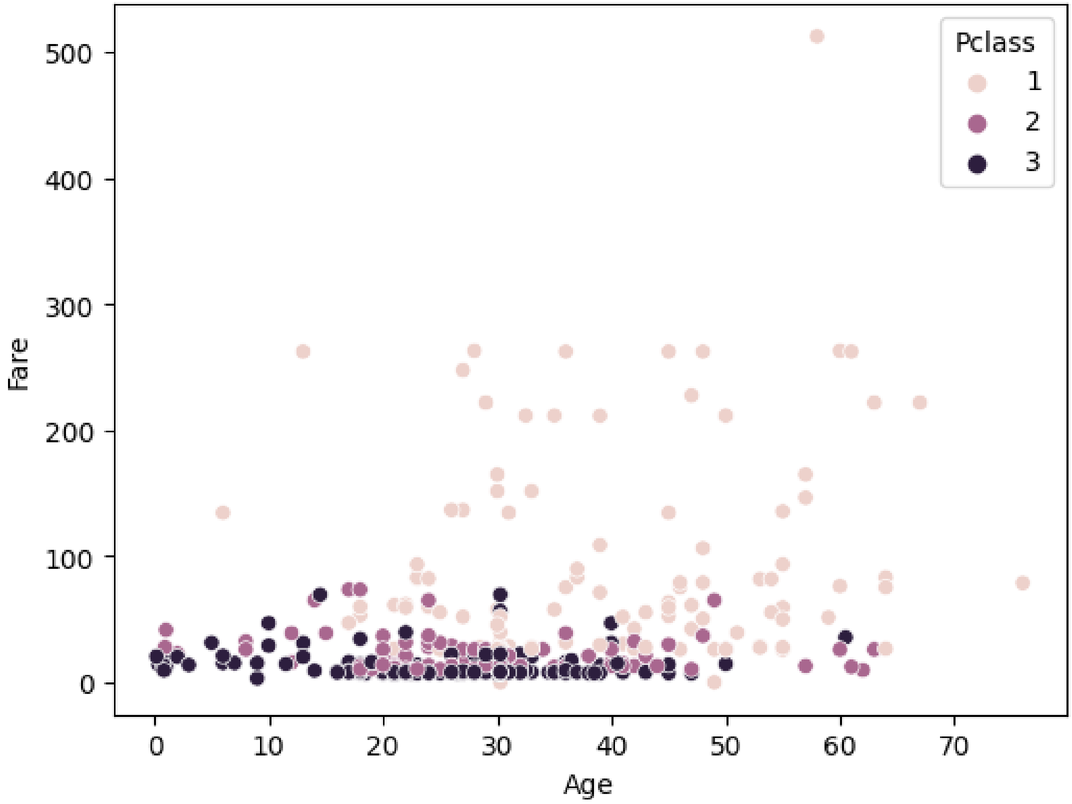
[ ]:

```
plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
sns.boxplot(df1['Age'])

plt.subplot(1,2,2)
sns.boxplot(df1['Fare'])
plt.show()
```



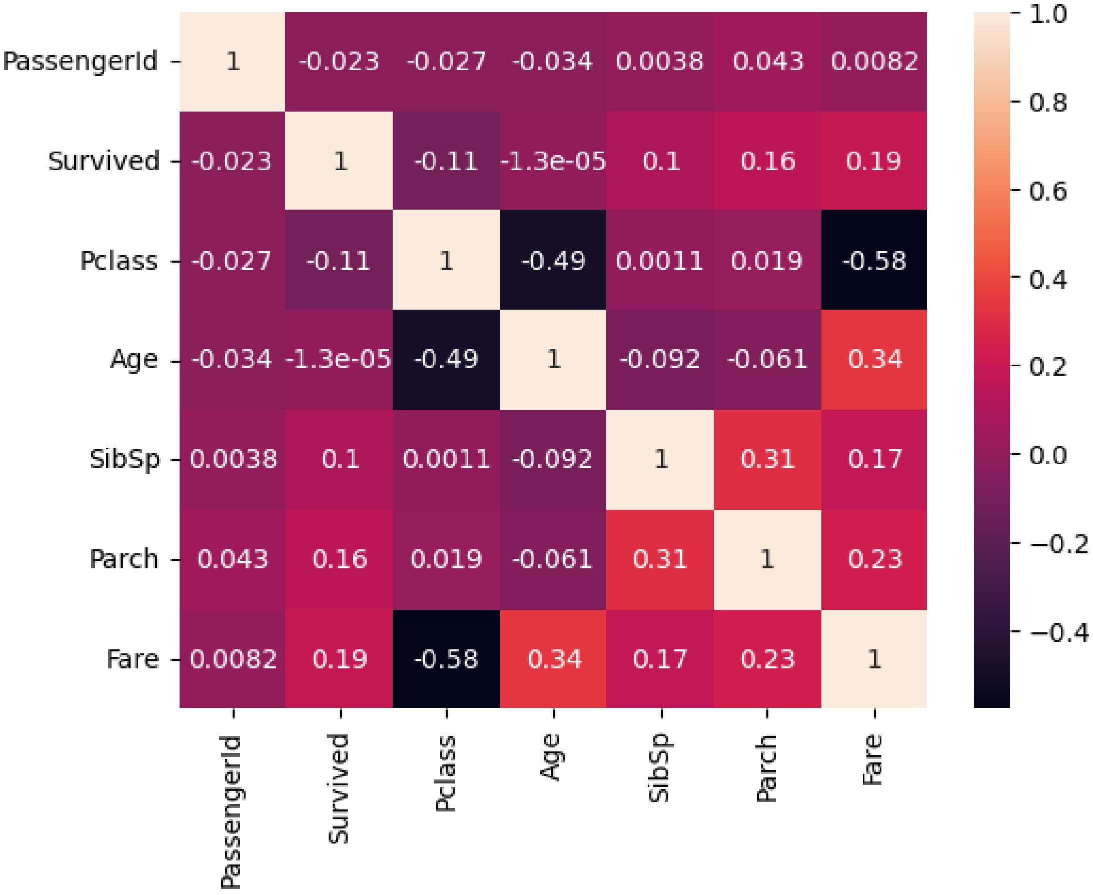
```
[ ]: sns.scatterplot(data=df1,x='Age',y='Fare',hue='Pclass')
```

```
[ ]: <Axes: xlabel='Age', ylabel='Fare'>
```



```
[ ]: sns.heatmap(df.corr(),annot=True)
     plt.show()
```

```
<ipython-input-19-f6412ee67fb3>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df.corr(),annot=True)
```



Encoding using LabelEncoder

```
[ ]: from sklearn.preprocessing import LabelEncoder
     label=LabelEncoder()
     df1['Sex']=label.fit_transform(df1['Sex'])
     df1['Embarked']=label.fit_transform(df1['Embarked'])
     df1
```

```
[ ]:      Survived  Pclass  Sex      Age  SibSp  Parch     Fare  Embarked
     0           0       3    1  34.50000      0      0   7.8292         1
```

```
1            1       3    0  47.00000      1      0    7.0000         2
2            0       2    1  62.00000      0      0    9.6875         1
3            0       3    1  27.00000      0      0    8.6625         2
4            1       3    0  22.00000      1      1   12.2875         2
..         …     …  …        …    …    …        …        …
413          0       3    1  30.27259      0      0    8.0500         2
414          1       1    0  39.00000      0      0  108.9000         0
415          0       3    1  38.50000      0      0    7.2500         2
416          0       3    1  30.27259      0      0    8.0500         2
417          0       3    1  30.27259      1      1   22.3583         0

[418 rows x 8 columns]
```

```
[ ]: df.replace({'Sex':{'male':0,'female':1},'Embarked':{'S':0,'C':1,'Q':
      ↪2}},inplace=True)
```

Input and Output Separation

```
[ ]: x=df1.iloc[:,:-1].values
     x
```

```
[ ]: array([[ 0.    ,  3.    ,  1.    , …,  0.    ,  0.    ,  7.8292],
            [ 1.    ,  3.    ,  0.    , …,  1.    ,  0.    ,  7.    ],
            [ 0.    ,  2.    ,  1.    , …,  0.    ,  0.    ,  9.6875],
            …,
            [ 0.    ,  3.    ,  1.    , …,  0.    ,  0.    ,  7.25  ],
            [ 0.    ,  3.    ,  1.    , …,  0.    ,  0.    ,  8.05  ],
            [ 0.    ,  3.    ,  1.    , …,  1.    ,  1.    , 22.3583]])
```

```
[ ]: y=df1.iloc[:,-1].values
     y
```

```
[ ]: array([1, 2, 1, 2, 2, 2, 1, 2, 0, 2, 2, 2, 2, 2, 2, 0, 1, 0, 2, 0, 0, 2,
            2, 0, 0, 2, 0, 0, 2, 0, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 2, 2,
            2, 2, 0, 1, 0, 2, 2, 0, 2, 2, 0, 1, 2, 2, 2, 0, 2, 2, 2, 1, 0, 2,
            1, 2, 0, 2, 1, 2, 2, 0, 0, 0, 2, 2, 2, 1, 0, 2, 2, 2, 1, 0, 1, 2,
            1, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 1, 2, 0, 2, 1, 1, 2, 2,
            0, 1, 0, 1, 2, 0, 0, 2, 0, 2, 2, 1, 0, 2, 1, 2, 2, 1, 2, 2, 2, 0,
            2, 0, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2,
            2, 2, 2, 2, 2, 2, 1, 0, 2, 2, 2, 2, 0, 2, 0, 2, 2, 0, 2, 0, 2, 2,
            2, 0, 2, 0, 2, 0, 2, 1, 0, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 0, 2,
            2, 2, 1, 2, 0, 2, 2, 0, 1, 2, 0, 2, 2, 2, 2, 2, 2, 2, 1, 2, 0, 2,
            0, 2, 2, 2, 0, 0, 2, 1, 2, 2, 2, 2, 2, 1, 0, 2, 0, 0, 2, 0, 0, 2,
            0, 2, 2, 2, 2, 2, 2, 0, 2, 2, 0, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2,
            0, 2, 2, 2, 2, 2, 0, 1, 0, 1, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 2, 2,
            2, 2, 0, 2, 2, 1, 0, 2, 2, 2, 0, 0, 2, 2, 2, 0, 2, 2, 1, 2, 2, 2,
            2, 2, 2, 0, 2, 1, 0, 1, 0, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 2,
            2, 0, 0, 0, 2, 2, 2, 0, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0, 2,
            2, 2, 2, 2, 2, 2, 1, 2, 2, 0, 2, 2, 0, 2, 0, 2, 0, 0, 2, 0, 2, 2,
            2, 0, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2,
            1, 0, 2, 1, 2, 2, 0, 2, 0, 0, 2, 0, 1, 2, 1, 1, 2, 2, 0, 2, 2, 0])
```

```
[ ]: from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
      ↪30,random_state=42)
```

```
[ ]: x_train
```

```
[ ]: array([[  0.    ,   1.    ,   1.    , ...,   0.    ,   0.    ,  75.2417],
            [  0.    ,   3.    ,   1.    , ...,   0.    ,   0.    ,   7.75  ],
            [  1.    ,   1.    ,   0.    , ...,   1.    ,   0.    , 221.7792],
            ...,
            [  0.    ,   1.    ,   1.    , ...,   0.    ,   0.    ,  75.2417],
            [  0.    ,   2.    ,   1.    , ...,   0.    ,   0.    ,  13.5   ],
            [  0.    ,   3.    ,   1.    , ...,   0.    ,   0.    ,   7.75  ]])
```

```
[ ]: x_test
```

```
[ ]: array([[0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.22920000e+00],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 3.90000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.11337500e+02],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.10000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.75000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.89580000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 3.60000000e+01,
             0.00000000e+00, 2.00000000e+00, 1.21833000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 5.00000000e+01,
             1.00000000e+00, 0.00000000e+00, 2.60000000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.90000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.92500000e+00],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 4.90000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.60000000e+01],
            [1.00000000e+00, 2.00000000e+00, 0.00000000e+00, 1.90000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.30000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 8.05000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.10000000e+01,
             2.00000000e+00, 0.00000000e+00, 2.41500000e+01],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 5.10000000e+01,
             0.00000000e+00, 1.00000000e+00, 3.94000000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 1.60000000e+01,
             1.00000000e+00, 1.00000000e+00, 8.51670000e+00],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 3.90000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.08900000e+02],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 8.05000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 5.64958000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.80000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.77500000e+00],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 5.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 5.00000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 1.00000000e+01,
             4.00000000e+00, 1.00000000e+00, 2.91250000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.30000000e+01,
             1.00000000e+00, 0.00000000e+00, 1.05000000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 5.70000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.30000000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 4.10000000e+01,
             1.00000000e+00, 0.00000000e+00, 5.18625000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 3.00000000e+00,
             1.00000000e+00, 1.00000000e+00, 1.37750000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 3.00000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.30000000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 2.00000000e+00, 1.52458000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 1.85000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.28330000e+00],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 2.50000000e+01,
             1.00000000e+00, 0.00000000e+00, 5.54417000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 2.65500000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.90000000e+01,
             0.00000000e+00, 2.00000000e+00, 7.22920000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 3.00000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.30000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.20000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.25250000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.20000000e+01,
             0.00000000e+00, 0.00000000e+00, 3.96875000e+01],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 3.30000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.51550000e+02],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 8.05000000e+00],
            [1.00000000e+00, 2.00000000e+00, 0.00000000e+00, 2.20000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.05000000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.05000000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.40000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.75000000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 1.50458000e+01],
            [1.00000000e+00, 2.00000000e+00, 0.00000000e+00, 2.90000000e+01,
             1.00000000e+00, 0.00000000e+00, 2.60000000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 3.25000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.11500000e+02],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.40000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.75000000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 3.02725904e+01,
             1.00000000e+00, 2.00000000e+00, 2.34500000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 4.10000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.50458000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 3.96000000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 2.85000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.77208000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.10000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.85420000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 3.02725904e+01,
             1.00000000e+00, 9.00000000e+00, 6.95500000e+01],
            [1.00000000e+00, 2.00000000e+00, 0.00000000e+00, 2.40000000e+01,
             1.00000000e+00, 0.00000000e+00, 2.77208000e+01],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 5.50000000e+01,
             2.00000000e+00, 0.00000000e+00, 2.57000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.60000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.25000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.45000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.82920000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 4.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.22500000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 8.05000000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 4.70000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.05000000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.90000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.38583000e+01],
            [1.00000000e+00, 2.00000000e+00, 0.00000000e+00, 2.00000000e+01,
             1.00000000e+00, 0.00000000e+00, 2.60000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 8.05000000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.60000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.30000000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 1.70000000e-01,
             1.00000000e+00, 2.00000000e+00, 2.05750000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.40000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.25000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 5.00000000e+01,
             1.00000000e+00, 0.00000000e+00, 1.45000000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.10000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.15000000e+01],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 2.77208000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 4.00000000e+01,
             1.00000000e+00, 0.00000000e+00, 2.60000000e+01],
            [1.00000000e+00, 2.00000000e+00, 0.00000000e+00, 1.50000000e+01,
             0.00000000e+00, 2.00000000e+00, 3.90000000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.05000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 4.10000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.85000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.40000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.55000000e+00],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 3.90000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.97000000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 4.30000000e+01,
             1.00000000e+00, 0.00000000e+00, 2.77208000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 5.70000000e+01,
             1.00000000e+00, 0.00000000e+00, 1.46520800e+02],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 3.00000000e+01,
             1.00000000e+00, 0.00000000e+00, 2.10000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.20000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.57920000e+00],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 3.16833000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 4.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.97000000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.20000000e+01,
             2.00000000e+00, 0.00000000e+00, 8.66250000e+00],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 3.00000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.60000000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.20000000e+01,
             1.00000000e+00, 0.00000000e+00, 1.39000000e+01],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.60000000e+01,
             1.00000000e+00, 1.00000000e+00, 2.20250000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.65000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.20000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.79580000e+00],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 4.80000000e+01,
             1.00000000e+00, 3.00000000e+00, 2.62375000e+02],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.10000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.22500000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 1.80000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.87920000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 3.20000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.30000000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.40000000e+01,
             2.00000000e+00, 0.00000000e+00, 3.15000000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 1.07083000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 2.40000000e+01,
             1.00000000e+00, 0.00000000e+00, 8.22667000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 1.90000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.05000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 7.05000000e+00],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 2.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.60000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 6.00000000e+00,
             3.00000000e+00, 1.00000000e+00, 2.10750000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 4.60000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.92000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 1.40000000e+01,
             0.00000000e+00, 0.00000000e+00, 9.22500000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.92500000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 1.00000000e+01,
             5.00000000e+00, 2.00000000e+00, 4.69000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.10000000e+01,
             3.00000000e+00, 0.00000000e+00, 1.80000000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 1.70000000e+01,
             0.00000000e+00, 0.00000000e+00, 4.71000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 7.75000000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 3.50000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.23500000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.40000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.77500000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 7.75000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.30000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.05000000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 2.70000000e+01,
             0.00000000e+00, 0.00000000e+00, 8.66250000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.70000000e+01,
             1.00000000e+00, 0.00000000e+00, 7.92500000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 4.00000000e+00, 2.54667000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 1.70000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.89580000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.30000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.05000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 7.22500000e+00],
            [1.00000000e+00, 3.00000000e+00, 0.00000000e+00, 2.20000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.72500000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 7.89580000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 6.10000000e+01,
             0.00000000e+00, 0.00000000e+00, 1.23500000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 1.15000000e+01,
             1.00000000e+00, 1.00000000e+00, 1.45000000e+01],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 6.00000000e+00,
             0.00000000e+00, 2.00000000e+00, 1.34500000e+02],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 6.43750000e+00],
            [1.00000000e+00, 1.00000000e+00, 0.00000000e+00, 2.20000000e+01,
             0.00000000e+00, 1.00000000e+00, 6.19792000e+01],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.60000000e+01,
             1.00000000e+00, 1.00000000e+00, 2.90000000e+01],
            [1.00000000e+00, 2.00000000e+00, 0.00000000e+00, 9.20000000e-01,
             1.00000000e+00, 2.00000000e+00, 2.77500000e+01],
            [1.00000000e+00, 2.00000000e+00, 0.00000000e+00, 2.20000000e+01,
             0.00000000e+00, 0.00000000e+00, 2.10000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             0.00000000e+00, 0.00000000e+00, 8.71250000e+00],
            [0.00000000e+00, 2.00000000e+00, 1.00000000e+00, 2.70000000e+01,
             1.00000000e+00, 0.00000000e+00, 2.60000000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 1.80000000e+01,
             0.00000000e+00, 0.00000000e+00, 8.66250000e+00],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.02725904e+01,
             2.00000000e+00, 0.00000000e+00, 2.16792000e+01],
            [0.00000000e+00, 3.00000000e+00, 1.00000000e+00, 3.30000000e+01,
             0.00000000e+00, 0.00000000e+00, 7.85420000e+00],
            [0.00000000e+00, 1.00000000e+00, 1.00000000e+00, 2.30000000e+01,
             0.00000000e+00, 0.00000000e+00, 9.35000000e+01]])
```

```
[ ]: y_train
```

```
[ ]: array([0, 1, 2, 0, 0, 2, 2, 0, 1, 0, 2, 1, 2, 2, 0, 0, 2, 2, 0, 2, 0, 0,
            2, 2, 0, 1, 0, 2, 0, 2, 2, 2, 2, 2, 2, 0, 0, 1, 2, 2, 1, 2, 2, 2,
            0, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 1,
            2, 2, 2, 0, 2, 2, 0, 0, 2, 2, 0, 2, 0, 2, 2, 2, 2, 0, 2, 0, 0, 2,
            2, 2, 1, 2, 2, 2, 2, 0, 2, 1, 2, 2, 0, 0, 2, 2, 2, 2, 2, 0, 0, 0,
            0, 2, 0, 0, 2, 2, 1, 0, 2, 1, 2, 0, 2, 2, 0, 2, 2, 2, 1, 1, 2, 0,
            2, 0, 2, 2, 1, 2, 1, 2, 0, 2, 2, 2, 2, 0, 0, 2, 2, 2, 0, 2, 2, 0,
            2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 1, 1, 2, 0, 2, 1, 2, 0, 2, 2, 2,
            0, 2, 0, 2, 2, 1, 2, 2, 1, 0, 0, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 0,
            0, 2, 2, 2, 0, 2, 2, 1, 2, 0, 0, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 0,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 0, 2, 2, 0, 2, 2, 1, 1,
            1, 0, 2, 1, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 0,
            2, 1, 1, 2, 2, 2, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 1, 0, 0,
            2, 2, 1, 0, 2, 1])
```

```
[ ]: y_test
```

```
[ ]: array([0, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 1, 2, 2, 2,
       2, 2, 0, 1, 0, 2, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 2, 0, 1, 2, 0, 2,
       0, 2, 2, 0, 2, 2, 1, 0, 2, 2, 0, 2, 2, 2, 2, 1, 2, 2, 0, 2, 2, 2,
       2, 2, 0, 0, 0, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 0, 0, 1, 2, 2, 1, 2,
       2, 2, 0, 2, 0, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 0, 1,
       2, 1, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 2, 2])
```

Scaling/Normalization

```
[ ]: from sklearn.preprocessing import StandardScaler
     scaler=StandardScaler()
     scaler.fit(x_train)
     x_train=scaler.transform(x_train)
     x_test=scaler.transform(x_test)
```

```
[ ]: x_train
```

```
[ ]: array([[-0.78310898, -1.47596812,  0.78310898, …, -0.50269793,
         -0.43509701,  0.60029624],
        [-0.78310898,  0.8727114 ,  0.78310898, …, -0.50269793,
         -0.43509701, -0.51006323],
        [ 1.27696148, -1.47596812, -1.27696148, …,  0.5766241 ,
         -0.43509701,  3.01110083],
        …,
        [-0.78310898, -1.47596812,  0.78310898, …, -0.50269793,
         -0.43509701,  0.60029624],
        [-0.78310898, -0.30162836,  0.78310898, …, -0.50269793,
         -0.43509701, -0.41546542],
        [-0.78310898,  0.8727114 ,  0.78310898, …, -0.50269793,
         -0.43509701, -0.51006323]])
```

```
[ ]: x_test
```

```
[ ]: array([[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
         -4.46969232e-01, -5.02697935e-01, -4.35097005e-01,
         -5.18631327e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
  6.18125338e-01, -5.02697935e-01, -4.35097005e-01,
  2.83931615e+00],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -7.51281967e-01, -5.02697935e-01, -4.35097005e-01,
 -5.10063233e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  3.13812604e-01, -5.02697935e-01, -4.35097005e-01,
 -5.07664561e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
  3.89890787e-01, -5.02697935e-01,  1.66487532e+00,
 -4.37127500e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
  1.45498536e+00,  5.76624102e-01, -4.35097005e-01,
 -2.09818013e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -1.42656498e-01, -5.02697935e-01, -4.35097005e-01,
 -5.07184169e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  1.37890717e+00, -5.02697935e-01, -4.35097005e-01,
 -2.09818013e-01],
[ 1.27696148e+00, -3.01628363e-01, -1.27696148e+00,
 -9.03438334e-01, -5.02697935e-01, -4.35097005e-01,
 -4.23691320e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.05127695e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -7.51281967e-01,  1.65594614e+00, -4.35097005e-01,
 -2.40253830e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
  1.53106354e+00, -5.02697935e-01,  6.14889156e-01,
  1.06360115e-02],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -1.13167288e+00,  5.76624102e-01,  6.14889156e-01,
 -4.97449643e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
  6.18125338e-01, -5.02697935e-01, -4.35097005e-01,
  1.15403562e+00],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.05127695e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
  2.91892572e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -2.18734682e-01, -5.02697935e-01, -4.35097005e-01,
 -5.09651938e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  1.83537628e+00, -5.02697935e-01, -4.35097005e-01,
  1.85025016e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -1.58814199e+00,  3.81459021e+00,  6.14889156e-01,
 -1.58406160e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -5.99125599e-01,  5.76624102e-01, -4.35097005e-01,
 -4.64820802e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
  1.98753264e+00, -5.02697935e-01, -4.35097005e-01,
 -4.23691320e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  7.70281705e-01,  5.76624102e-01, -4.35097005e-01,
  2.15666480e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -2.12068927e+00,  5.76624102e-01,  6.14889156e-01,
 -4.10941181e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -6.65783144e-02, -5.02697935e-01, -4.35097005e-01,
 -4.23691320e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -4.58401348e-02, -5.02697935e-01,  1.66487532e+00,
 -3.86743884e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -9.41477425e-01, -5.02697935e-01, -4.35097005e-01,
 -5.17741285e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
 -4.46969232e-01,  5.76624102e-01, -4.35097005e-01,
  2.74550737e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -2.00769527e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  6.18125338e-01, -5.02697935e-01,  1.66487532e+00,
 -5.18631327e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -6.65783144e-02, -5.02697935e-01, -4.35097005e-01,
 -4.23691320e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  8.55780528e-02, -5.02697935e-01, -4.35097005e-01,
 -2.66987993e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -6.75203783e-01, -5.02697935e-01, -4.35097005e-01,
  1.53659019e-02],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
  1.61656236e-01, -5.02697935e-01, -4.35097005e-01,
  1.85570458e+00],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.05127695e-01],
[ 1.27696148e+00, -3.01628363e-01, -1.27696148e+00,
 -6.75203783e-01, -5.02697935e-01, -4.35097005e-01,
 -4.64820802e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -4.46969232e-01, -5.02697935e-01, -4.35097005e-01,
 -4.64820802e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -5.23047416e-01, -5.02697935e-01, -4.35097005e-01,
 -5.10063233e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -3.90034242e-01],
[ 1.27696148e+00, -3.01628363e-01, -1.27696148e+00,
 -1.42656498e-01,  5.76624102e-01, -4.35097005e-01,
 -2.09818013e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  1.23617145e-01, -5.02697935e-01, -4.35097005e-01,
  2.84198956e+00],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -5.23047416e-01, -5.02697935e-01, -4.35097005e-01,
 -5.10063233e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -4.58401348e-02,  5.76624102e-01,  1.66487532e+00,
 -2.51770085e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
  7.70281705e-01, -5.02697935e-01, -4.35097005e-01,
 -3.90034242e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
  1.39263701e-02],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -1.80695590e-01, -5.02697935e-01, -4.35097005e-01,
 -1.81507768e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -7.51281967e-01, -5.02697935e-01, -4.35097005e-01,
 -5.08348956e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -4.58401348e-02,  5.76624102e-01,  9.01477845e+00,
  5.06657567e-01],
[ 1.27696148e+00, -3.01628363e-01, -1.27696148e+00,
 -5.23047416e-01,  5.76624102e-01, -4.35097005e-01,
 -1.81507768e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
  1.83537628e+00,  1.65594614e+00, -4.35097005e-01,
 -2.14753551e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  3.89890787e-01, -5.02697935e-01, -4.35097005e-01,
 -5.18289129e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  2.75773512e-01, -5.02697935e-01, -4.35097005e-01,
 -5.08760251e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
  1.07459444e+00, -5.02697935e-01, -4.35097005e-01,
 -5.18700424e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.05127695e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
  1.22675081e+00, -5.02697935e-01, -4.35097005e-01,
 -4.64820802e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -1.42656498e-01, -5.02697935e-01, -4.35097005e-01,
 -4.09570746e-01],
[ 1.27696148e+00, -3.01628363e-01, -1.27696148e+00,
 -8.27360150e-01,  5.76624102e-01, -4.35097005e-01,
 -2.09818013e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.05127695e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -3.70891049e-01, -5.02697935e-01, -4.35097005e-01,
 -4.23691320e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -2.33599053e+00,  5.76624102e-01,  1.66487532e+00,
 -2.99068989e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -5.23047416e-01, -5.02697935e-01, -4.35097005e-01,
 -5.18289129e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  1.45498536e+00,  5.76624102e-01, -4.35097005e-01,
 -3.99013631e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -7.51281967e-01, -5.02697935e-01, -4.35097005e-01,
 -4.48369010e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -1.81507768e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
  6.94203521e-01,  5.76624102e-01, -4.35097005e-01,
 -2.09818013e-01],
[ 1.27696148e+00, -3.01628363e-01, -1.27696148e+00,
 -1.20775107e+00, -5.02697935e-01,  1.66487532e+00,
  4.05529435e-03],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -4.46969232e-01, -5.02697935e-01, -4.35097005e-01,
 -4.64820802e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  7.70281705e-01, -5.02697935e-01, -4.35097005e-01,
 -5.08418054e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -5.23047416e-01, -5.02697935e-01, -4.35097005e-01,
 -5.13353591e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  6.18125338e-01, -5.02697935e-01, -4.35097005e-01,
 -1.48946379e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  9.22438072e-01,  5.76624102e-01, -4.35097005e-01,
 -1.81507768e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  1.98753264e+00,  5.76624102e-01, -4.35097005e-01,
  1.77296523e+00],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -6.65783144e-02,  5.76624102e-01, -4.35097005e-01,
 -2.92076977e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  8.55780528e-02, -5.02697935e-01, -4.35097005e-01,
 -5.12873199e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -1.16317539e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  1.07459444e+00, -5.02697935e-01, -4.35097005e-01,
 -1.48946379e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -6.75203783e-01,  1.65594614e+00, -4.35097005e-01,
 -4.95050972e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -6.65783144e-02, -5.02697935e-01, -4.35097005e-01,
 -2.09818013e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -6.75203783e-01,  5.76624102e-01, -4.35097005e-01,
 -4.08884707e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -3.70891049e-01,  5.76624102e-01,  6.14889156e-01,
 -2.75213890e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.46969232e-01, -5.02697935e-01, -4.35097005e-01,
 -5.11708412e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -6.75203783e-01, -5.02697935e-01, -4.35097005e-01,
 -5.09309741e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
  1.30282899e+00,  5.76624102e-01,  2.71486148e+00,
  3.67897453e+00],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -7.51281967e-01, -5.02697935e-01, -4.35097005e-01,
 -5.18700424e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -9.79516517e-01, -5.02697935e-01, -4.35097005e-01,
 -5.07937661e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
  8.55780528e-02, -5.02697935e-01, -4.35097005e-01,
 -4.23691320e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -5.23047416e-01,  1.65594614e+00, -4.35097005e-01,
 -1.19333152e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -4.61393894e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -5.23047416e-01,  5.76624102e-01, -4.35097005e-01,
  7.15870081e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -9.03438334e-01, -5.02697935e-01, -4.35097005e-01,
 -4.64820802e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.21579488e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -4.46969232e-01, -5.02697935e-01, -4.35097005e-01,
 -2.09818013e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -1.89245472e+00,  2.73526817e+00,  6.14889156e-01,
 -2.90843093e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
  1.15067262e+00, -5.02697935e-01, -4.35097005e-01,
  6.65417368e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -1.28382925e+00, -5.02697935e-01, -4.35097005e-01,
 -4.85796838e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.46969232e-01, -5.02697935e-01, -4.35097005e-01,
 -5.07184169e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -1.58814199e+00,  4.89391225e+00,  1.66487532e+00,
  1.34024458e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  9.49986922e-03,  2.73526817e+00, -4.35097005e-01,
 -3.41432356e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -1.05559470e+00, -5.02697935e-01, -4.35097005e-01,
  1.37314817e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.10063233e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
  3.13812604e-01, -5.02697935e-01, -4.35097005e-01,
 -4.34384986e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -5.23047416e-01, -5.02697935e-01, -4.35097005e-01,
 -5.09651938e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.10063233e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -5.99125599e-01, -5.02697935e-01, -4.35097005e-01,
 -5.21579488e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -2.94812865e-01, -5.02697935e-01, -4.35097005e-01,
 -4.95050972e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -2.94812865e-01,  5.76624102e-01, -4.35097005e-01,
 -5.07184169e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -4.58401348e-02, -5.02697935e-01,  3.76484764e+00,
 -2.18591754e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -1.05559470e+00, -5.02697935e-01, -4.35097005e-01,
 -5.07664561e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -5.99125599e-01, -5.02697935e-01, -4.35097005e-01,
 -4.64820802e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.18700424e-01],
[ 1.27696148e+00,  8.72711396e-01, -1.27696148e+00,
 -6.75203783e-01, -5.02697935e-01, -4.35097005e-01,
 -5.10474528e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.07664561e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
  2.29184538e+00, -5.02697935e-01, -4.35097005e-01,
 -4.34384986e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -1.47402471e+00,  5.76624102e-01,  6.14889156e-01,
 -3.99013631e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -6.37564628e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -1.89245472e+00, -5.02697935e-01,  1.66487532e+00,
  1.57520151e+00],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -5.31656211e-01],
[ 1.27696148e+00, -1.47596812e+00, -1.27696148e+00,
 -6.75203783e-01, -5.02697935e-01,  6.14889156e-01,
  3.82104333e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -3.70891049e-01,  5.76624102e-01,  6.14889156e-01,
 -1.60462634e-01],
[ 1.27696148e+00, -3.01628363e-01, -1.27696148e+00,
 -2.27893189e+00,  5.76624102e-01,  1.66487532e+00,
 -1.81027375e-01],
[ 1.27696148e+00, -3.01628363e-01, -1.27696148e+00,
 -6.75203783e-01, -5.02697935e-01, -4.35097005e-01,
 -2.92076977e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02, -5.02697935e-01, -4.35097005e-01,
 -4.94228382e-01],
[-7.83108976e-01, -3.01628363e-01,  7.83108976e-01,
 -2.94812865e-01,  5.76624102e-01, -4.35097005e-01,
 -2.09818013e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -9.79516517e-01, -5.02697935e-01, -4.35097005e-01,
 -4.95050972e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
 -4.58401348e-02,  1.65594614e+00, -4.35097005e-01,
 -2.80902920e-01],
[-7.83108976e-01,  8.72711396e-01,  7.83108976e-01,
  1.61656236e-01, -5.02697935e-01, -4.35097005e-01,
 -5.08348956e-01],
[-7.83108976e-01, -1.47596812e+00,  7.83108976e-01,
 -5.99125599e-01, -5.02697935e-01, -4.35097005e-01,
  9.00678006e-01]])
```

Model Creation

1.**Logistic Regression**

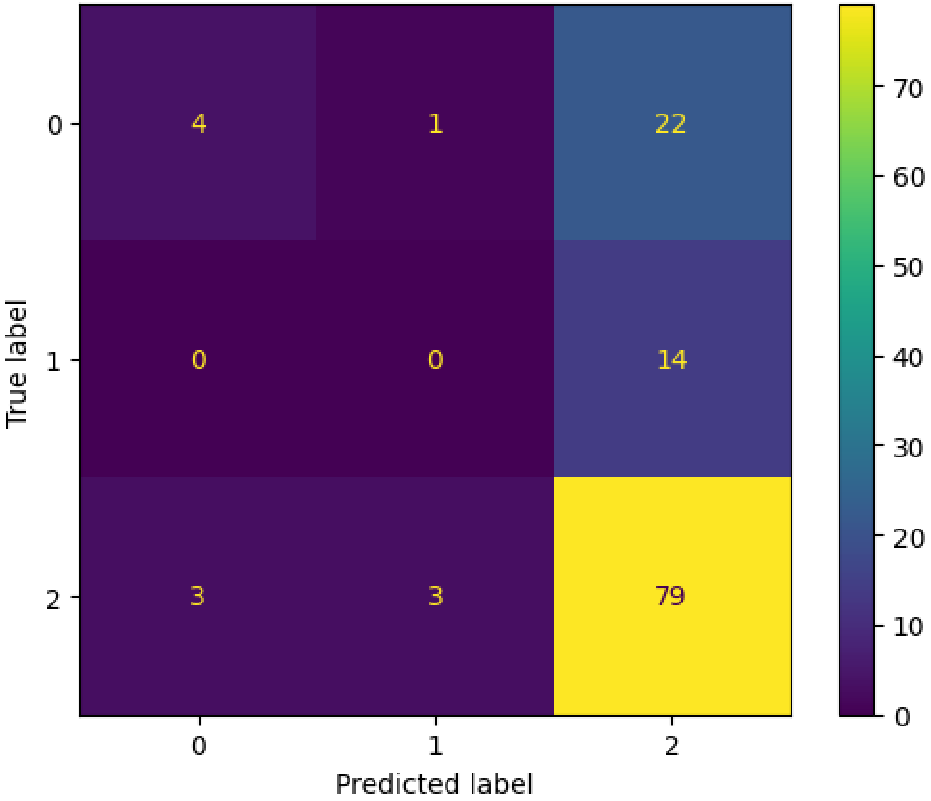
```
[ ]: from sklearn.linear_model import LogisticRegression
     lr=LogisticRegression()
     lr.fit(x_train,y_train)
     lr_pred = lr.predict(x_test)
     lr_pred
```

```
[ ]: array([2, 0, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 0, 2, 2, 1, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0])
```

```
[ ]: from sklearn.metrics import␣
      ↪confusion_matrix,accuracy_score,classification_report,ConfusionMatrixDisplay
     result=confusion_matrix(y_test,lr_pred)
     print(result)
     print(accuracy_score(y_test,lr_pred))
     print(classification_report(y_test,lr_pred))
     labels=[0,1,2]
     cmd=ConfusionMatrixDisplay(result,display_labels=labels)
     cmd.plot()
```

```
[[ 4  1 22]
 [ 0  0 14]
 [ 3  3 79]]
0.6587301587301587
              precision    recall  f1-score   support

           0       0.57      0.15      0.24        27
           1       0.00      0.00      0.00        14
           2       0.69      0.93      0.79        85

    accuracy                           0.66       126
   macro avg       0.42      0.36      0.34       126
weighted avg       0.59      0.66      0.58       126
```

```
[ ]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
     0x7e31e53949a0>
```



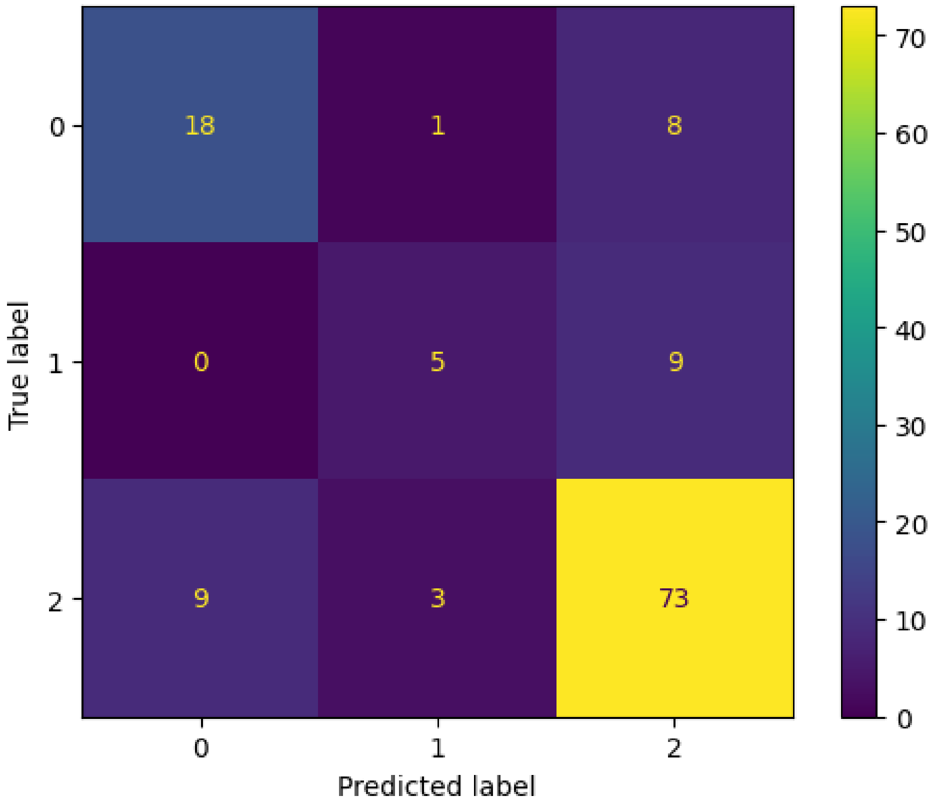
2.**Decision Tree Classifier**

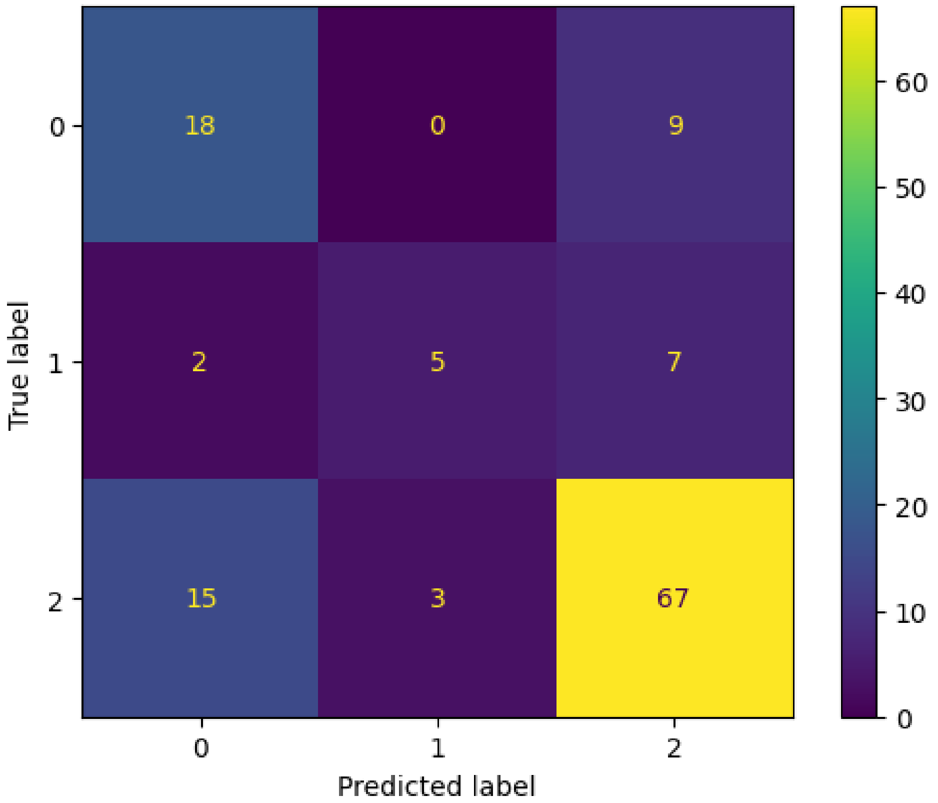
3.**Random Forest Classifier**

```
[ ]: from sklearn.tree import DecisionTreeClassifier
     from sklearn.ensemble import RandomForestClassifier
     dec=DecisionTreeClassifier(criterion='entropy')
     rf=RandomForestClassifier(n_estimators=10,criterion='entropy')
     lst_model=[dec,rf]
```

```
[ ]: from sklearn.metrics import␣
      ↪confusion_matrix,accuracy_score,classification_report,ConfusionMatrixDisplay
     for i in lst_model:
       print(i)
       i.fit(x_train,y_train)
       y_pred=i.predict(x_test)
       y_pred
       print(accuracy_score(y_test,y_pred))
       print("***************")
       print(classification_report(y_test,y_pred))
       print("***************")
       print(confusion_matrix(y_test,y_pred))
       print("***************")
       result=confusion_matrix(y_test,y_pred)
       labels=[0,1,2]
       cmd=ConfusionMatrixDisplay(result,display_labels=labels)
       cmd.plot()
```

```
DecisionTreeClassifier(criterion='entropy')
0.7619047619047619
***************
              precision    recall  f1-score   support

           0       0.67      0.67      0.67        27
           1       0.56      0.36      0.43        14
           2       0.81      0.86      0.83        85

    accuracy                           0.76       126
   macro avg       0.68      0.63      0.65       126
weighted avg       0.75      0.76      0.75       126

***************
[[18  1  8]
 [ 0  5  9]
 [ 9  3 73]]
***************
RandomForestClassifier(criterion='entropy', n_estimators=10)
0.7142857142857143
***************
              precision    recall  f1-score   support

           0       0.51      0.67      0.58        27
           1       0.62      0.36      0.45        14
           2       0.81      0.79      0.80        85

    accuracy                           0.71       126
   macro avg       0.65      0.60      0.61       126
weighted avg       0.72      0.71      0.71       126

***************
[[18  0  9]
 [ 2  5  7]
 [15  3 67]]
***************
```


18
1
8
0
5
9
9
3
73
True label
0
1
2
Predicted label
0
1
2
70
60
50
40
30
20
10
0



Model Prediction

```
[ ]: data={"pclass":1,"Sex":1,"Age":31.0,"SibSp":2,"Parch":1,"Fare":8.
      ↪2051,"Embarked":0}
     new_df2=pd.DataFrame(data,index=[0])
     new_df2
```

```
[ ]:    pclass  Sex   Age  SibSp  Parch    Fare  Embarked
     0       1    1  31.0      2      1  8.2051         0
```

```
[ ]: predi=dec.predict(new_df2)
     if predi==1:
       print("This Person is Survived")
     else:
       print("This Person is not Survived")
```

```
This Person is not Survived
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:432: UserWarning: X has
feature names, but DecisionTreeClassifier was fitted without feature names
  warnings.warn(
```