Titanic logestic modal

In [1]:

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

In [2]:

```
data=pd.read_csv("/home/placement/Downloads/Titanic Dataset.csv")
data
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

```
In [3]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

data.isna().sum()

In [4]:
```
data.head(10)
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

In [5]:
```
data["Sex"]=data["Sex"].map({"male":1,"female":2})
```

In [6]:
```
data["Pclass"].unique()
```

Out[6]:
```
array([3, 1, 2])
```

In [7]:
```
data["Survived"].unique()
```

Out[7]:
```
array([0, 1])
```

In [8]:
```
data["SibSp"].unique()
```

Out[8]:
```
array([1, 0, 3, 4, 2, 5, 8])
```

```
In [9]: data["Age"].unique()
```

```
Out[9]: array([22.  , 38.  , 26.  , 35.  ,   nan, 54.  ,  2.  , 27.  , 14.  ,
               4.  , 58.  , 20.  , 39.  , 55.  , 31.  , 34.  , 15.  , 28.  ,
               8.  , 19.  , 40.  , 66.  , 42.  , 21.  , 18.  ,  3.  ,  7.  ,
              49.  , 29.  , 65.  , 28.5 ,  5.  , 11.  , 45.  , 17.  , 32.  ,
              16.  , 25.  ,  0.83, 30.  , 33.  , 23.  , 24.  , 46.  , 59.  ,
              71.  , 37.  , 47.  , 14.5 , 70.5 , 32.5 , 12.  ,  9.  , 36.5 ,
              51.  , 55.5 , 40.5 , 44.  ,  1.  , 61.  , 56.  , 50.  , 36.  ,
              45.5 , 20.5 , 62.  , 41.  , 52.  , 63.  , 23.5 ,  0.92, 43.  ,
              60.  , 10.  , 64.  , 13.  , 48.  ,  0.75, 53.  , 57.  , 80.  ,
              70.  , 24.5 ,  6.  ,  0.67, 30.5 ,  0.42, 34.5 , 74.  ])
```

```
In [10]: data=data.drop("PassengerId",axis=1)
         data
```

Out[10]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	NaN	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	2	38.0	1	0	PC 17599	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	2	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	2	35.0	1	0	113803	53.1000	C123	S
4	0	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	Montvila, Rev. Juozas	1	27.0	0	0	211536	13.0000	NaN	S
887	1	1	Graham, Miss. Margaret Edith	2	19.0	0	0	112053	30.0000	B42	S
888	0	3	Johnston, Miss. Catherine Helen "Carrie"	2	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	1	1	Behr, Mr. Karl Howell	1	26.0	0	0	111369	30.0000	C148	C
890	0	3	Dooley, Mr. Patrick	1	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 11 columns

In [11]:

```
list(data)
```

Out[11]:

```
['Survived',
 'Pclass',
 'Name',
 'Sex',
 'Age',
 'SibSp',
 'Parch',
 'Ticket',
 'Fare',
 'Cabin',
 'Embarked']
```

In [12]:

```
data=data.drop(["Name",'Parch',"SibSp",'Ticket',"Cabin"],axis=1)
data
```

Out[12]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	35.0	53.1000	S
4	0	3	1	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	S
887	1	1	2	19.0	30.0000	S
888	0	3	2	NaN	23.4500	S
889	1	1	1	26.0	30.0000	C
890	0	3	1	32.0	7.7500	Q

891 rows × 6 columns

```
In [13]: list(data)
```

```
Out[13]: ['Survived', 'Pclass', 'Sex', 'Age', 'Fare', 'Embarked']
```

```
In [14]: data.isna().sum()
```

```
Out[14]: Survived      0
         Pclass        0
         Sex           0
         Age         177
         Fare          0
         Embarked      2
         dtype: int64
```

```
In [ ]:
```

```
In [15]: 
         data['Age'].fillna(35,inplace = True)
```

```
In [ ]:
```

```
In [ ]:
```

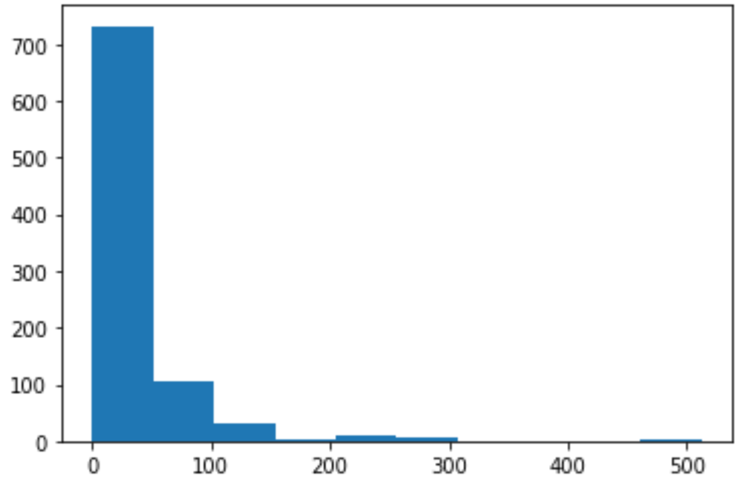
```
In [16]: data.isna().sum()
```

```
Out[16]: Survived    0
         Pclass      0
         Sex         0
         Age         0
         Fare        0
         Embarked    2
         dtype: int64
```

```
In [17]: import matplotlib.pyplot as mb
         import seaborn as sb
```

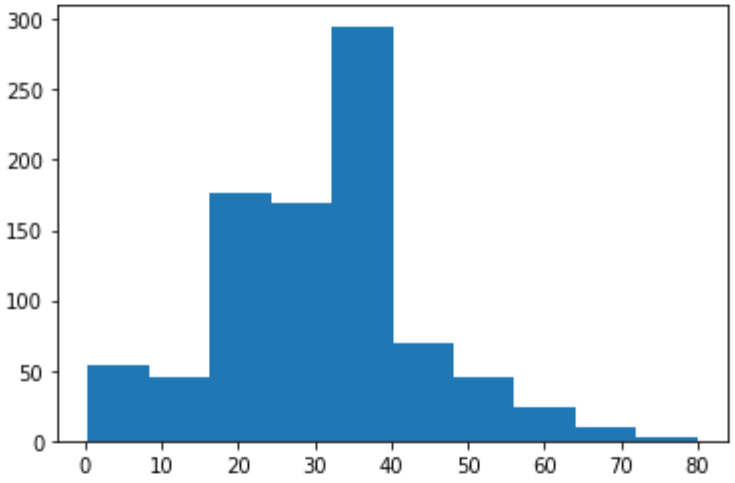
```
In [18]: mb.hist(data["Fare"])
```

```
Out[18]: (array([732., 106.,  31.,   2.,  11.,   6.,   0.,   0.,   0.,   3.]),
          array([  0.     ,  51.23292, 102.46584, 153.69876, 204.93168, 256.1646 ,
                 307.39752, 358.63044, 409.86336, 461.09628, 512.3292 ]),
          <BarContainer object of 10 artists>)
```



```
In [19]: mb.hist(data["Age"])
```

```
Out[19]: (array([ 54.,  46., 177., 169., 295.,  70.,  45.,  24.,   9.,   2.]),
          array([ 0.42 ,  8.378, 16.336, 24.294, 32.252, 40.21 , 48.168, 56.126,
                 64.084, 72.042, 80.   ]),
          <BarContainer object of 10 artists>)
```



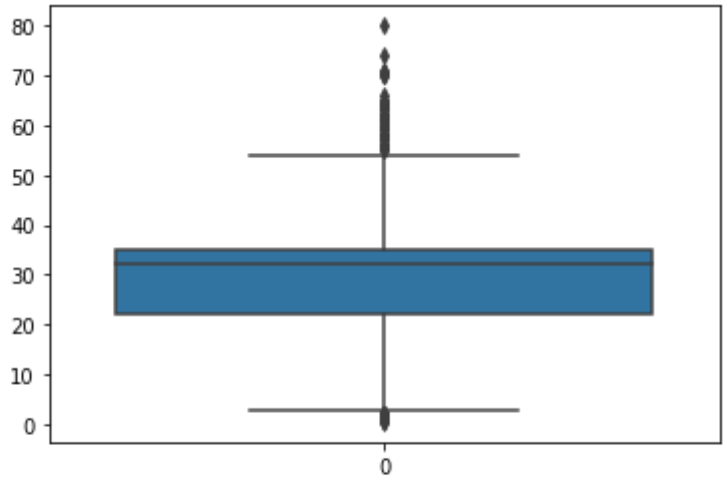
```
In [20]: data["Age"]
```

```
Out[20]: 0      22.0
         1      38.0
         2      26.0
         3      35.0
         4      35.0
                ...
         886    27.0
         887    19.0
         888    35.0
         889    26.0
         890    32.0
         Name: Age, Length: 891, dtype: float64
```

Boxplot

```
In [21]: sb.boxplot(data["Age"])
```

```
Out[21]: <Axes: >
```



```
In [22]: data.describe()
```

```
Out[22]:
```

	Survived	Pclass	Sex	Age	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	1.352413	30.752155	32.204208
std	0.486592	0.836071	0.477990	13.173100	49.693429
min	0.000000	1.000000	1.000000	0.420000	0.000000
25%	0.000000	2.000000	1.000000	22.000000	7.910400
50%	0.000000	3.000000	1.000000	32.000000	14.454200
75%	1.000000	3.000000	2.000000	35.000000	31.000000
max	1.000000	3.000000	2.000000	80.000000	512.329200

In [23]:
```
data["Age"].unique()
```

Out[23]:
```
array([22.  , 38.  , 26.  , 35.  , 54.  ,  2.  , 27.  , 14.  ,  4.  ,
       58.  , 20.  , 39.  , 55.  , 31.  , 34.  , 15.  , 28.  ,  8.  ,
       19.  , 40.  , 66.  , 42.  , 21.  , 18.  ,  3.  ,  7.  , 49.  ,
       29.  , 65.  , 28.5 ,  5.  , 11.  , 45.  , 17.  , 32.  , 16.  ,
       25.  ,  0.83, 30.  , 33.  , 23.  , 24.  , 46.  , 59.  , 71.  ,
       37.  , 47.  , 14.5 , 70.5 , 32.5 , 12.  ,  9.  , 36.5 , 51.  ,
       55.5 , 40.5 , 44.  ,  1.  , 61.  , 56.  , 50.  , 36.  , 45.5 ,
       20.5 , 62.  , 41.  , 52.  , 63.  , 23.5 ,  0.92, 43.  , 60.  ,
       10.  , 64.  , 13.  , 48.  ,  0.75, 53.  , 57.  , 80.  , 70.  ,
       24.5 ,  6.  ,  0.67, 30.5 ,  0.42, 34.5 , 74.  ])
```

In [24]:
```
data.groupby(["Age"]).count()
```

Out[24]:

	Survived	**Pclass**	**Sex**	**Fare**	**Embarked**
Age					
0.42	1	1	1	1	1
0.67	1	1	1	1	1
0.75	2	2	2	2	2
0.83	2	2	2	2	2
0.92	1	1	1	1	1
...	...	...	...	...	...
70.00	2	2	2	2	2
70.50	1	1	1	1	1
71.00	2	2	2	2	2
74.00	1	1	1	1	1
80.00	1	1	1	1	1

88 rows × 5 columns

In [25]:
```
#k["Pclass"]=k["Pclass"].map({1:"F",2:"S",3:"T"})
data
```

Out[25]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	35.0	53.1000	S
4	0	3	1	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	S
887	1	1	2	19.0	30.0000	S
888	0	3	2	35.0	23.4500	S
889	1	1	1	26.0	30.0000	C
890	0	3	1	32.0	7.7500	Q

891 rows × 6 columns

In [26]:

```
data=pd.get_dummies(data,dtype=int)
data
```

Out[26]:

	Survived	Pclass	Sex	Age	Fare	Embarked_C	Embarked_Q	Embarked_S
0	0	3	1	22.0	7.2500	0	0	1
1	1	1	2	38.0	71.2833	1	0	0
2	1	3	2	26.0	7.9250	0	0	1
3	1	1	2	35.0	53.1000	0	0	1
4	0	3	1	35.0	8.0500	0	0	1
...	...	...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	0	0	1
887	1	1	2	19.0	30.0000	0	0	1
888	0	3	2	35.0	23.4500	0	0	1
889	1	1	1	26.0	30.0000	1	0	0
890	0	3	1	32.0	7.7500	0	1	0

891 rows × 8 columns

Correlation

```
In [27]: cor=data.corr()
         cor
```

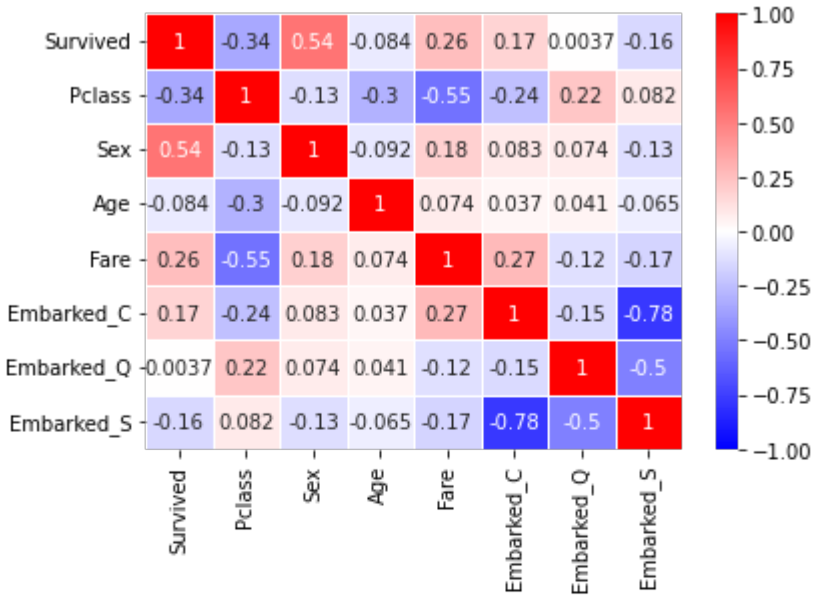
Out[27]:

	Survived	Pclass	Sex	Age	Fare	Embarked_C	Embarked_Q	Embarked_S
Survived	1.000000	-0.338481	0.543351	-0.083713	0.257307	0.168240	0.003650	-0.155660
Pclass	-0.338481	1.000000	-0.131900	-0.299255	-0.549500	-0.243292	0.221009	0.081720
Sex	0.543351	-0.131900	1.000000	-0.091930	0.182333	0.082853	0.074115	-0.125722
Age	-0.083713	-0.299255	-0.091930	1.000000	0.074199	0.036953	0.040528	-0.065062
Fare	0.257307	-0.549500	0.182333	0.074199	1.000000	0.269335	-0.117216	-0.166603
Embarked_C	0.168240	-0.243292	0.082853	0.036953	0.269335	1.000000	-0.148258	-0.778359
Embarked_Q	0.003650	0.221009	0.074115	0.040528	-0.117216	-0.148258	1.000000	-0.496624
Embarked_S	-0.155660	0.081720	-0.125722	-0.065062	-0.166603	-0.778359	-0.496624	1.000000

Heat map

```
In [28]: import seaborn as s
         s.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidths=.5,cmap='bwr')

Out[28]: <Axes: >
```



```
In [29]: data.groupby("Survived").count()

Out[29]:
```

	Pclass	Sex	Age	Fare	Embarked_C	Embarked_Q	Embarked_S
Survived							
0	549	549	549	549	549	549	549
1	342	342	342	342	342	342	342

In [30]:

```
y=data['Survived']
x=data.drop("Survived",axis=1)
y
```

Out[30]:

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

In [31]:

```
y
```

Out[31]:

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

x_train,y_train

In [32]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [33]:
```
x.isna().sum()
```

Out[33]:
```
Pclass        0
Sex           0
Age           0
Fare          0
Embarked_C    0
Embarked_Q    0
Embarked_S    0
dtype: int64
```

In [34]:
```
x_train
```

Out[34]:

	Pclass	Sex	Age	Fare	Embarked_C	Embarked_Q	Embarked_S
6	1	1	54.0	51.8625	0	0	1
718	3	1	35.0	15.5000	0	1	0
685	2	1	25.0	41.5792	1	0	0
73	3	1	26.0	14.4542	1	0	0
882	3	2	22.0	10.5167	0	0	1
...	...	...	...	...	...	...	...
106	3	2	21.0	7.6500	0	0	1
270	1	1	35.0	31.0000	0	0	1
860	3	1	41.0	14.1083	0	0	1
435	1	2	14.0	120.0000	0	0	1
102	1	1	21.0	77.2875	0	0	1

596 rows × 7 columns

In [35]:

```
y_train
```

Out[35]:

```
6      0
718    0
685    0
73     0
882    0
      ..
106    1
270    0
860    0
435    1
102    0
Name: Survived, Length: 596, dtype: int64
```

LogisticRegression

In [36]:

```
import warnings
warnings.filterwarnings("ignore")
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[36]:

```
▾ LogisticRegression

LogisticRegression()
```

```
In [37]: ypred=classifier.predict(x_test)
         ypred
```

```
Out[37]: array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
                1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
                1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1,
                0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1,
                0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
                1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0,
                0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1,
                0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
                0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0,
                1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0,
                0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
                0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
                0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
                1, 0, 0, 0, 0, 0, 1, 1, 0])
```

confusion_matrix

```
In [38]: from sklearn.metrics import confusion_matrix
         confusion_matrix(y_test,ypred)
```

```
Out[38]: array([[151,  24],
                [ 31,  89]])
```

Accuracy_score

```
In [39]: from sklearn.metrics import accuracy_score
         accuracy_score(y_test,ypred)
```

```
Out[39]: 0.8135593220338984
```

```
In [ ]:
```

In [ ]:

In [ ]:

In [ ]: