1)Importing neccesssary libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.preprocessing import LabelEncoder,StandardScaler
```

2)Understanding about data.

In [2]:
```
df=pd.read_csv('tested.csv')
copy=pd.read_csv('tested.csv')
```

In [3]:
```
df
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

418 rows × 12 columns

In [4]:
```
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
```

In [5]:
```
df.describe()
```

Out[5]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

3)Data preprocessing

3.1)Dealing with null values.

In [6]:
```
df.isna().sum()
```

Out[6]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [7]:
```
Age_mean=round(df['Age'].mean(),1)#filling null values with mean.
df['Age']=df['Age'].fillna(Age_mean)
```

In [8]:
```
df.isna().sum()
```

Out[8]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [9]:
```
df
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
413	1305	0	3	Spector, Mr. Woolf	male	30.3	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	30.3	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter, Master. Michael J	male	30.3	1	1	2668	22.3583	NaN	C

418 rows × 12 columns

In [10]:
```
df['Fare']=df['Fare'].fillna(round(df['Fare'].mean(),1))
```

In [11]:
```
df.isna().sum()
```

Out[11]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          327
Embarked         0
dtype: int64
```

In [12]:
```
df=df.drop(['Cabin'],axis=1)#Cabin contains lot of null values and is unneccessary for our model prediction.
```

In [13]:
```
df.isna().sum()
```

Out[13]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

3.2)Removing unneccessary columns.

In [14]:
```
df
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	S
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	0	3	Spector, Mr. Woolf	male	30.3	0	0	A.5. 3236	8.0500	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	S
416	1308	0	3	Ware, Mr. Frederick	male	30.3	0	0	359309	8.0500	S
417	1309	0	3	Peter, Master. Michael J	male	30.3	1	1	2668	22.3583	C

418 rows × 11 columns

In [15]:
```
df=df.drop(['Embarked','Name','Ticket','Fare'],axis=1)
```

In [16]:
```
df
```

Out[16]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch
0	892	0	3	male	34.5	0	0
1	893	1	3	female	47.0	1	0
2	894	0	2	male	62.0	0	0
3	895	0	3	male	27.0	0	0
4	896	1	3	female	22.0	1	1
...	...	...	...	...	...	...	...
413	1305	0	3	male	30.3	0	0
414	1306	1	1	female	39.0	0	0
415	1307	0	3	male	38.5	0	0
416	1308	0	3	male	30.3	0	0
417	1309	0	3	male	30.3	1	1

418 rows × 7 columns

4)Feature Engineering.

In [17]:
```
df
```

Out[17]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch
0	892	0	3	male	34.5	0	0
1	893	1	3	female	47.0	1	0
2	894	0	2	male	62.0	0	0
3	895	0	3	male	27.0	0	0
4	896	1	3	female	22.0	1	1
...	...	...	...	...	...	...	...
413	1305	0	3	male	30.3	0	0
414	1306	1	1	female	39.0	0	0
415	1307	0	3	male	38.5	0	0
416	1308	0	3	male	30.3	0	0
417	1309	0	3	male	30.3	1	1

418 rows × 7 columns

In [18]:
```
family=np.array(df['SibSp'])+np.array(df['Parch'])#
```

In [19]:
```
df.insert(loc=5,column='Family',value=family)
```

In [20]:
```
df=df.drop(['SibSp','Parch'],axis=1)#We have used SibSp and Parch and added a new col family.
```

In [21]:
```
df
```

Out[21]:

	PassengerId	Survived	Pclass	Sex	Age	Family
0	892	0	3	male	34.5	0
1	893	1	3	female	47.0	1
2	894	0	2	male	62.0	0
3	895	0	3	male	27.0	0
4	896	1	3	female	22.0	2
...	...	...	...	...	...	...
413	1305	0	3	male	30.3	0
414	1306	1	1	female	39.0	0
415	1307	0	3	male	38.5	0
416	1308	0	3	male	30.3	0
417	1309	0	3	male	30.3	2

418 rows × 6 columns

In [22]:
```
df
```

Out[22]:

	PassengerId	Survived	Pclass	Sex	Age	Family
0	892	0	3	male	34.5	0
1	893	1	3	female	47.0	1
2	894	0	2	male	62.0	0
3	895	0	3	male	27.0	0
4	896	1	3	female	22.0	2
...	...	...	...	...	...	...
413	1305	0	3	male	30.3	0
414	1306	1	1	female	39.0	0
415	1307	0	3	male	38.5	0
416	1308	0	3	male	30.3	0
417	1309	0	3	male	30.3	2

418 rows × 6 columns

In [23]:
```
bins=[np.NINF,18,35,52,np.inf]
labels=['Children','Youth','Middle-Aged','Senior']
df['Age_Label']=pd.cut(df['Age'],bins=bins,labels=labels)
```

In [24]:
```
df
```

Out[24]:

	PassengerId	Survived	Pclass	Sex	Age	Family	Age_Label
0	892	0	3	male	34.5	0	Youth
1	893	1	3	female	47.0	1	Middle-Aged
2	894	0	2	male	62.0	0	Senior
3	895	0	3	male	27.0	0	Youth
4	896	1	3	female	22.0	2	Youth
...	...	...	...	...	...	...	...
413	1305	0	3	male	30.3	0	Youth
414	1306	1	1	female	39.0	0	Middle-Aged
415	1307	0	3	male	38.5	0	Middle-Aged
416	1308	0	3	male	30.3	0	Youth
417	1309	0	3	male	30.3	2	Youth

418 rows × 7 columns

In [25]:
```
df=df.drop(['Age','PassengerId'],axis=1)
```

Encoding our data.

In [26]:
```
lab_enc=LabelEncoder()
df['Sex']=lab_enc.fit_transform(df['Sex'])
```

In [27]:
```
df
```

Out[27]:

	Survived	Pclass	Sex	Family	Age_Label
0	0	3	1	0	Youth
1	1	3	0	1	Middle-Aged
2	0	2	1	0	Senior
3	0	3	1	0	Youth
4	1	3	0	2	Youth
...	...	...	...	...	...
413	0	3	1	0	Youth
414	1	1	0	0	Middle-Aged
415	0	3	1	0	Middle-Aged
416	0	3	1	0	Youth
417	0	3	1	2	Youth

418 rows × 5 columns

In [28]:
```
df['Age_Label']=lab_enc.fit_transform(df['Age_Label'])#Encoding Age_label col
```

In [29]:
```
df
```

Out[29]:

	Survived	Pclass	Sex	Family	Age_Label
0	0	3	1	0	3
1	1	3	0	1	1
2	0	2	1	0	2
3	0	3	1	0	3
4	1	3	0	2	3
...	...	...	...	...	...
413	0	3	1	0	3
414	1	1	0	0	1
415	0	3	1	0	1
416	0	3	1	0	3
417	0	3	1	2	3

418 rows × 5 columns

In [30]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Survived   418 non-null    int64
 1   Pclass     418 non-null    int64
 2   Sex        418 non-null    int64
 3   Family     418 non-null    int64
 4   Age_Label  418 non-null    int64
dtypes: int64(5)
memory usage: 16.5 KB
```

In [31]:
```
df.describe()
```

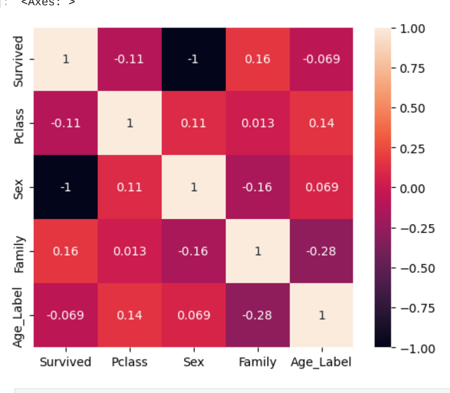
Out[31]:

	Survived	Pclass	Sex	Family	Age_Label
count	418.000000	418.000000	418.000000	418.000000	418.000000
mean	0.363636	2.265550	0.636364	0.839713	2.181818
std	0.481622	0.841838	0.481622	1.519072	1.133935
min	0.000000	1.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000	0.000000	1.000000
50%	0.000000	3.000000	1.000000	0.000000	3.000000
75%	1.000000	3.000000	1.000000	1.000000	3.000000
max	1.000000	3.000000	1.000000	10.000000	3.000000

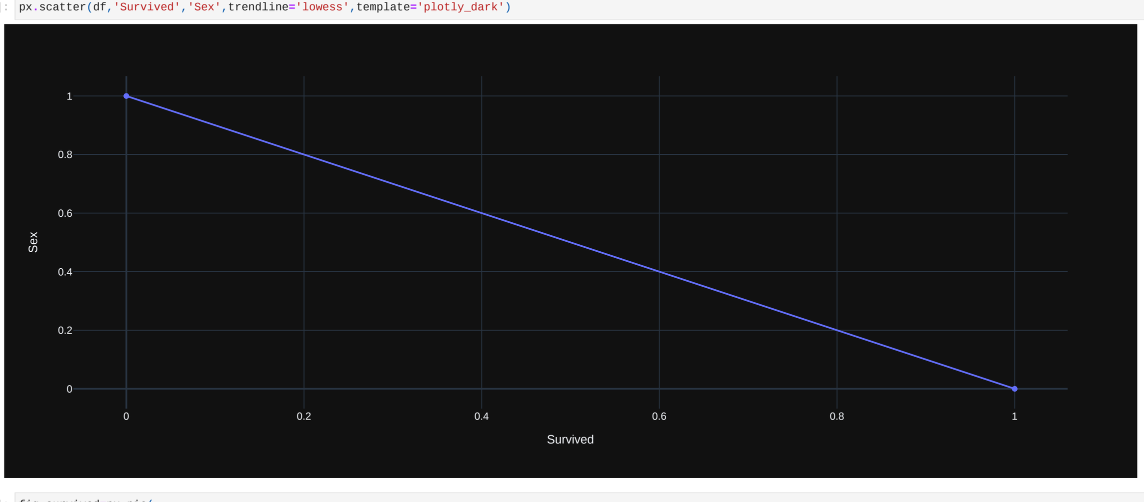
5)Explorartory Data Analysis

In [32]:
```
corr=df.corr()
sns.heatmap(corr,annot=True,square=True,robust=True)
```

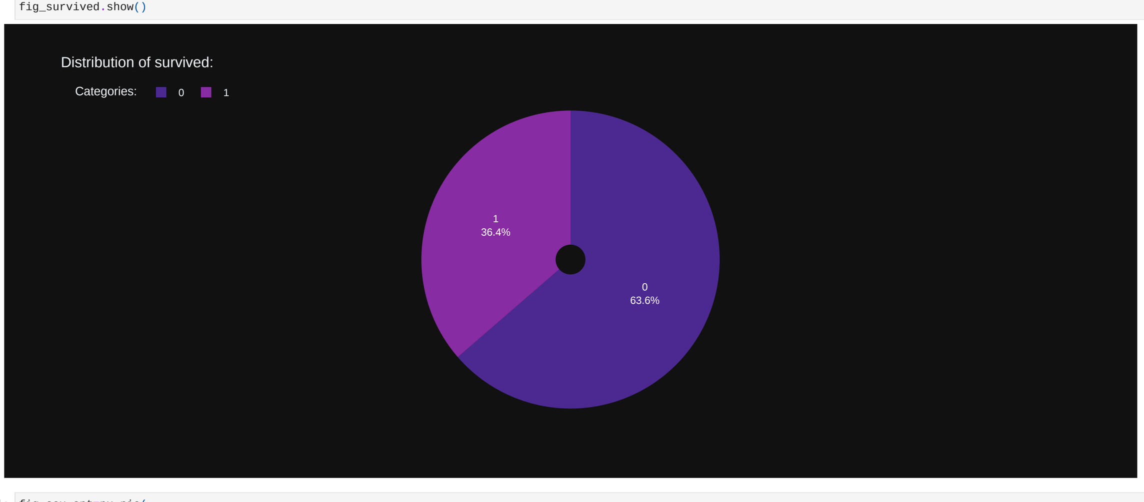
Out[32]: <Axes: >



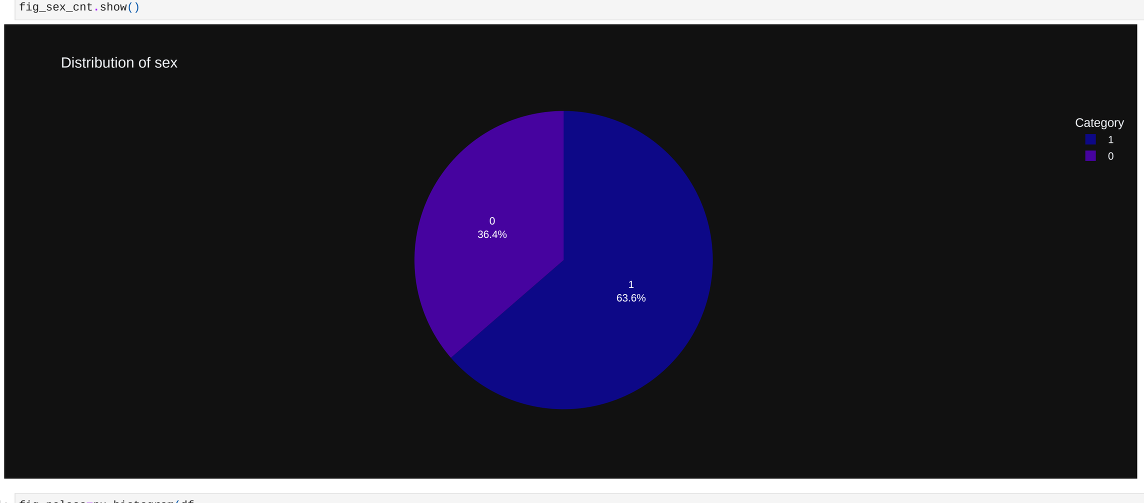
In [33]:
```
px.scatter(df,'Survived','Sex',trendline='lowess',template='plotly_dark')
```



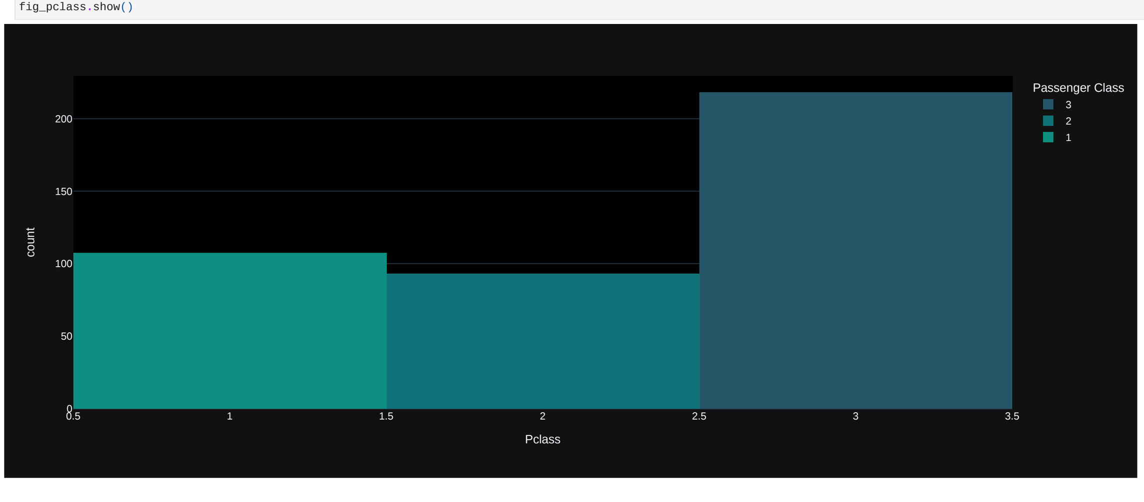
In [34]:
```
fig_survived=px.pie(
    df,
    names=df['Survived'].value_counts().index,
    values=df['Survived'].value_counts().values,
    template='plotly_dark',
    title='Distribution of survived:',
    hole=0.1,
    color_discrete_sequence=px.colors.sequential.Agsunset)
fig_survived.update_traces(textinfo='percent+label')
fig_survived.update_layout(
    legend_title_text='Categories:',
    legend=dict(orientation='h',yanchor='bottom', y=1.02))
fig_survived.show()
```



In [35]:
```
fig_sex_cnt=px.pie(
    df,
    names=df['Sex'].value_counts().index,
    values=df['Sex'].value_counts().values,
    color_discrete_sequence=px.colors.sequential.Plasma,
    title='Distribution of sex'
    )
fig_sex_cnt.update_traces(textinfo='percent+label')
fig_sex_cnt.update_layout(legend_title_text='Category',template='plotly_dark')
fig_sex_cnt.show()
```



In [36]:
```
fig_pclass=px.histogram(df,
                        x=str('Pclass'),
                        color=str('Pclass'),
                        color_discrete_sequence=px.colors.sequential.Aggrnyl)
fig_pclass.update_legends(title_text='Passenger Class')
fig_pclass.update_layout(plot_bgcolor='black',template='plotly_dark')
fig_pclass.show()
```



6)Model Preparation.

In [37]:
```
y=df['Survived']
```

In [38]:
```
df=df.drop(['Survived'],axis=1)
```

In [39]:
```
df
```

Out[39]:

	Pclass	Sex	Family	Age_Label
0	3	1	0	3
1	3	0	1	1
2	2	1	0	2
3	3	1	0	3
4	3	0	2	3
...	...	...	...	...
413	3	1	0	3
414	1	0	0	1
415	3	1	0	1
416	3	1	0	3
417	3	1	2	3

418 rows × 4 columns

Scaling data using standard scalar.

In [40]:
```
std_sclr=StandardScaler()
cols=df.columns
prcd_data=pd.DataFrame(std_sclr.fit_transform(df),columns=[cols])
```

In [41]:
```
prcd_data
```

Out[41]:

	Pclass	Sex	Family	Age_Label
0	0.873482	0.755929	-0.553443	0.722407
1	0.873482	-1.322876	0.105643	-1.043477
2	-0.315819	0.755929	-0.553443	-0.160535
3	0.873482	0.755929	-0.553443	0.722407
4	0.873482	-1.322876	0.764728	0.722407
...	...	...	...	...
413	0.873482	0.755929	-0.553443	0.722407
414	-1.505120	-1.322876	-0.553443	-1.043477
415	0.873482	0.755929	-0.553443	-1.043477
416	0.873482	0.755929	-0.553443	0.722407
417	0.873482	0.755929	0.764728	0.722407

418 rows × 4 columns

In [42]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn.model_selection import train_test_split
```

In [43]:
```
X_train,X_test,Y_train,Y_test=train_test_split(prcd_data,y,test_size=0.2,random_state=42)
```

In [44]:
```
log_reg=LogisticRegression()
log_reg.fit(X_train,Y_train)
```

Out[44]:
```
▾ LogisticRegression
LogisticRegression()
```

7)Testing our model.

In [45]:
```
y_pred=log_reg.predict(X_test)
```

In [46]:
```
accuracy_score(Y_test,y_pred)
```

Out[46]: 1.0

In [47]:
```
confusion_matrix(Y_test,y_pred)
```

Out[47]:
```
array([[50,  0],
       [ 0, 34]])
```

In [48]:
```
print(classification_report(Y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        50
           1       1.00      1.00      1.00        34

    accuracy                           1.00        84
   macro avg       1.00      1.00      1.00        84
```

```
weighted avg       1.00      1.00      1.00        84
```

```
In [ ]:
```