

practical-3-sneha-1

November 21, 2023

```
[1]: import pandas as pd
```

```
[2]: import numpy as np
```

```
[3]: import matplotlib.pyplot as plt
```

```
[ ]: pip install seaborn
```

```
[4]: import seaborn as sns
```

```
[5]: import sklearn
```

```
[6]: import math
```

```
[7]: titanic = pd.read_csv("titanic.csv")  
titanic.sample(10)
```

```
[7]:
```

	PassengerId	Survived	Pclass	\
218	219	1	1	
672	673	0	2	
496	497	1	1	
633	634	0	1	
134	135	0	2	
566	567	0	3	
668	669	0	3	
567	568	0	3	
288	289	1	2	
470	471	0	3	

	Name	Sex	Age	SibSp	Parch	\
218	Bazzani, Miss. Albina	female	32.0	0	0	
672	Mitchell, Mr. Henry Michael	male	70.0	0	0	
496	Eustis, Miss. Elizabeth Mussey	female	54.0	1	0	
633	Parr, Mr. William Henry Marsh	male	NaN	0	0	
134	Sobey, Mr. Samuel James Hayden	male	25.0	0	0	
566	Stoytcheff, Mr. Ilia	male	19.0	0	0	
668	Cook, Mr. Jacob	male	43.0	0	0	

567	Palsson, Mrs. Nils (Alma Cornelia Berglund)	female	29.0	0	4
288	Hosono, Mr. Masabumi	male	42.0	0	0
470	Keefe, Mr. Arthur	male	NaN	0	0

	Ticket	Fare	Cabin	Embarked
218	11813	76.2917	D15	C
672	C.A. 24580	10.5000	NaN	S
496	36947	78.2667	D20	C
633	112052	0.0000	NaN	S
134	C.A. 29178	13.0000	NaN	S
566	349205	7.8958	NaN	S
668	A/5 3536	8.0500	NaN	S
567	349909	21.0750	NaN	S
288	237798	13.0000	NaN	S
470	323592	7.2500	NaN	S

```
[8]: print ("Number Of Passengers:" +str(len(titanic.index)))
```

Number Of Passengers:891

```
[9]: titanic.columns
```

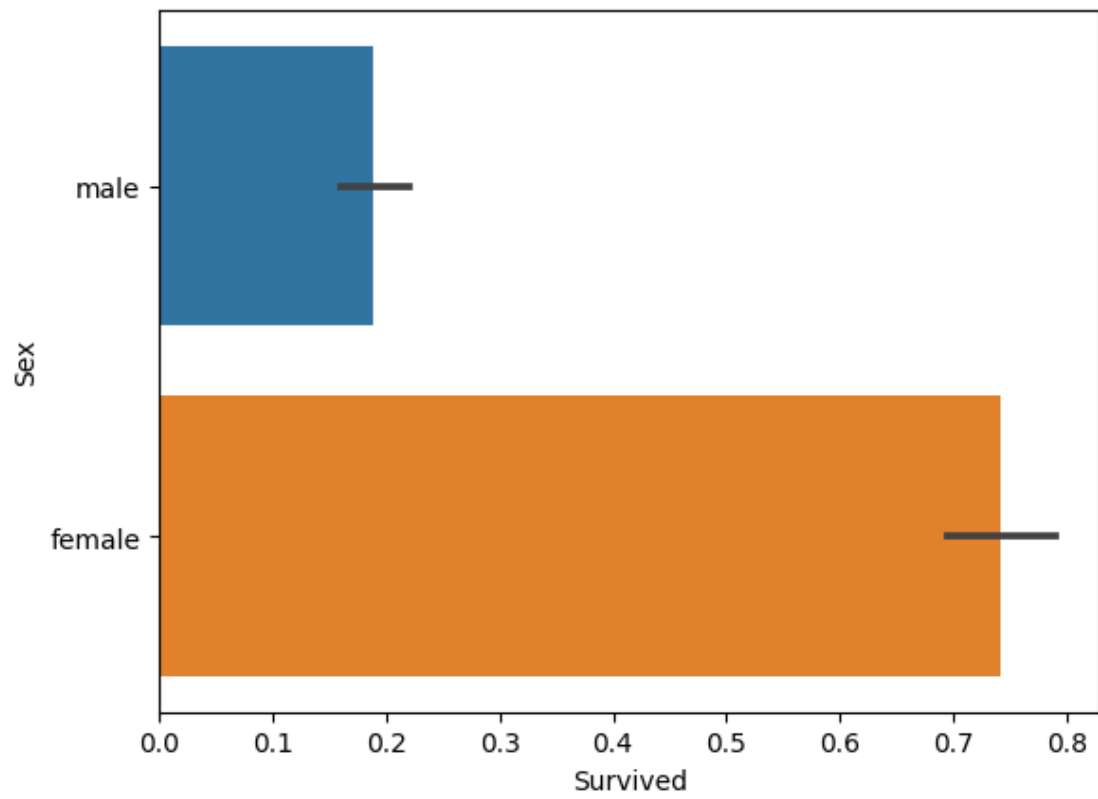
```
[9]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
         'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
         dtype='object')
```

```
[10]: titanic=titanic.drop(columns=['SibSp','Parch','Name'])
```

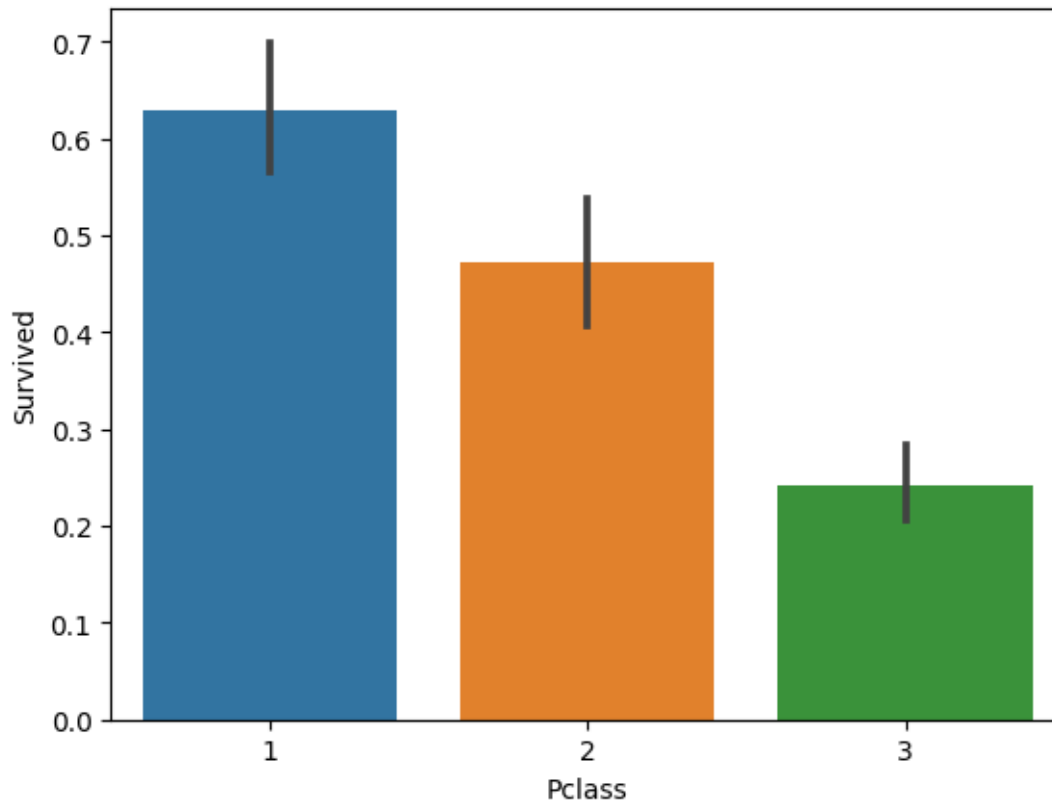
```
[11]: titanic.columns
```

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

```
[12]: # Sex based survival of people
sns.barplot(x=titanic.Survived, y=titanic.Sex);
```



```
[13]: # Passenger class based survival of people
sns.barplot(y=titanic.Survived, x=titanic.Pclass);
```

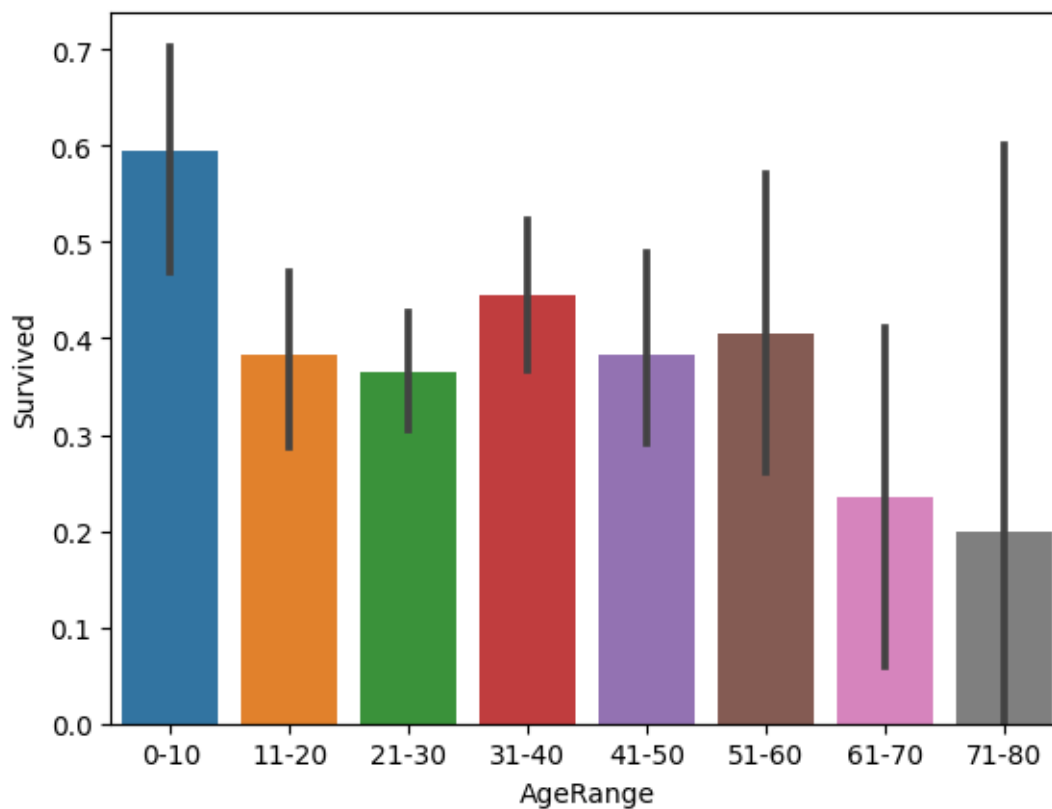


```
[14]: # Assuming you have a DataFrame named 'titanic' with 'Age' and 'Survived'
      ↪ columns

      # Define the age ranges
      age_ranges = ["0-10", "11-20", "21-30", "31-40", "41-50", "51-60", "61-70",
      ↪ "71-80"]

[15]: titanic['AgeRange'] = pd.cut(titanic['Age'], bins=[0, 10, 20, 30, 40, 50, 60,
      ↪ 70, 80], labels=age_ranges)

[16]: sns.barplot(x='AgeRange', y='Survived', data=titanic, order=age_ranges);
```



```
[17]: titanic.isnull().sum()
```

```
[17]: PassengerId      0
      Survived        0
      Pclass          0
      Sex             0
      Age            177
      Ticket          0
      Fare           0
      Cabin          687
      Embarked        2
      AgeRange       177
      dtype: int64
```

```
[18]: titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  -

```

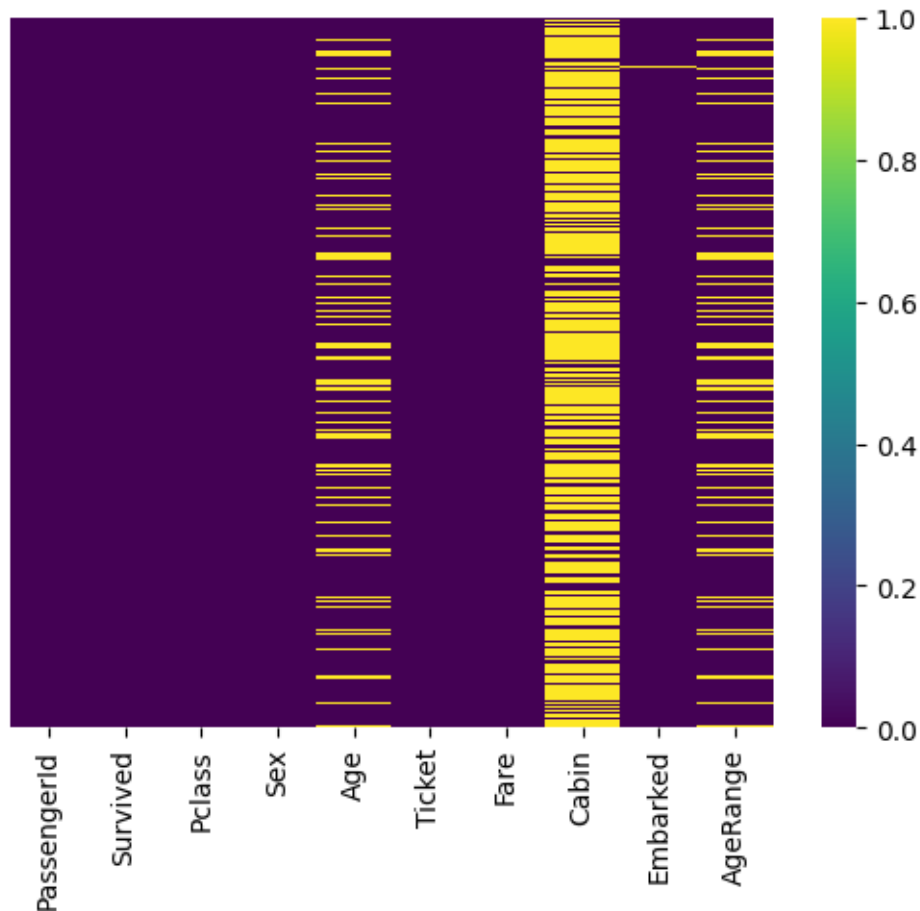
```

0  PassengerId  891 non-null    int64
1  Survived    891 non-null    int64
2  Pclass      891 non-null    int64
3  Sex         891 non-null    object
4  Age         714 non-null    float64
5  Ticket      891 non-null    object
6  Fare        891 non-null    float64
7  Cabin       204 non-null    object
8  Embarked    889 non-null    object
9  AgeRange    714 non-null    category
dtypes: category(1), float64(2), int64(3), object(4)
memory usage: 64.0+ KB

```

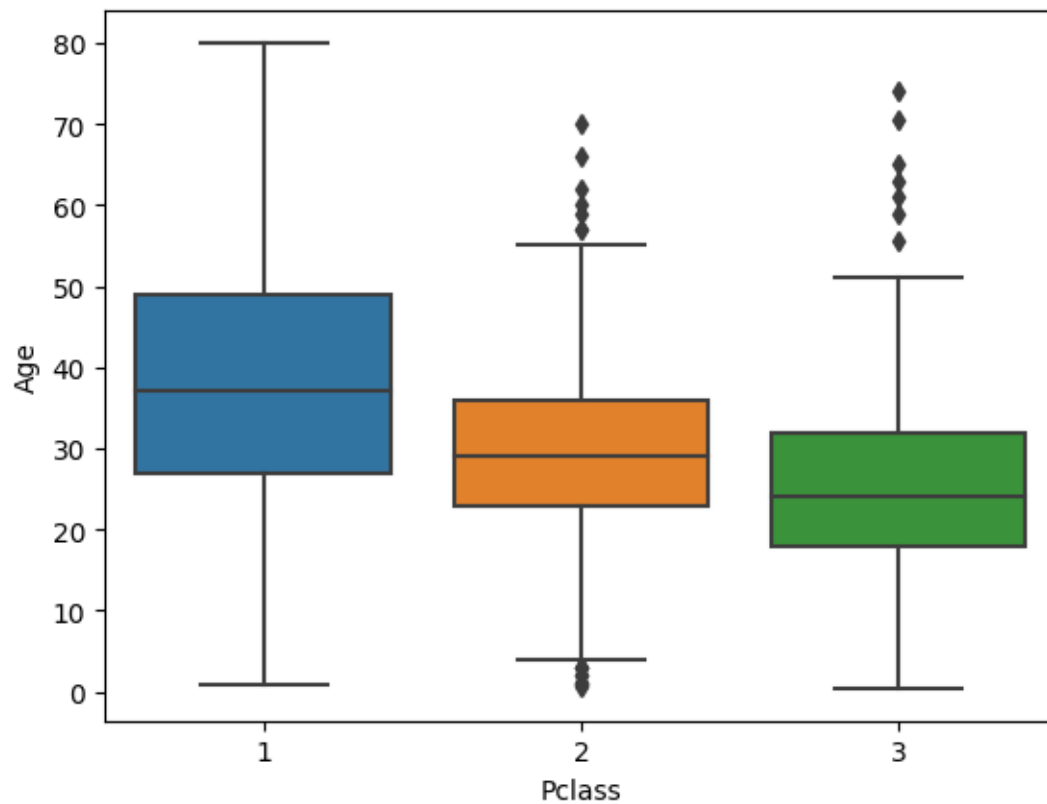
```
[19]: sns.heatmap(titanic.isnull(), yticklabels=False, cmap="viridis")
```

```
[19]: <Axes: >
```



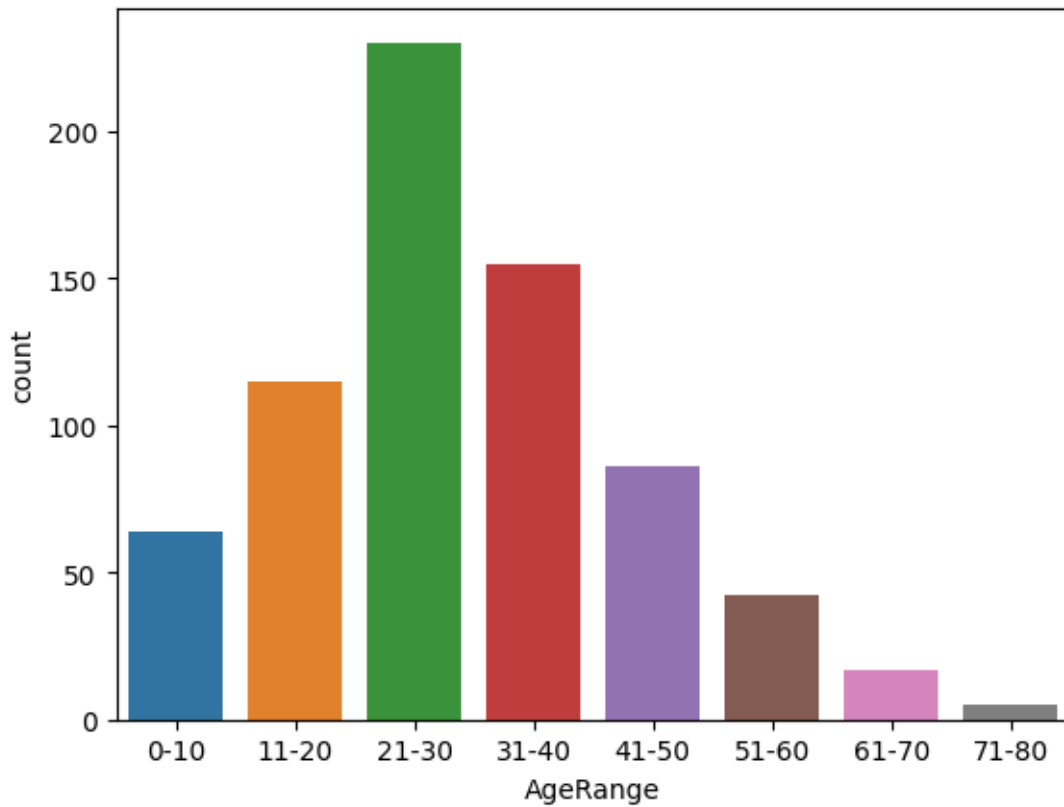
```
[20]: sns.boxplot(x="Pclass", y="Age", data=titanic)
```

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



```
[21]: sns.countplot(x='AgeRange', data=titanic)
      # Show the plot
```

```
[21]: <Axes: xlabel='AgeRange', ylabel='count'>
```

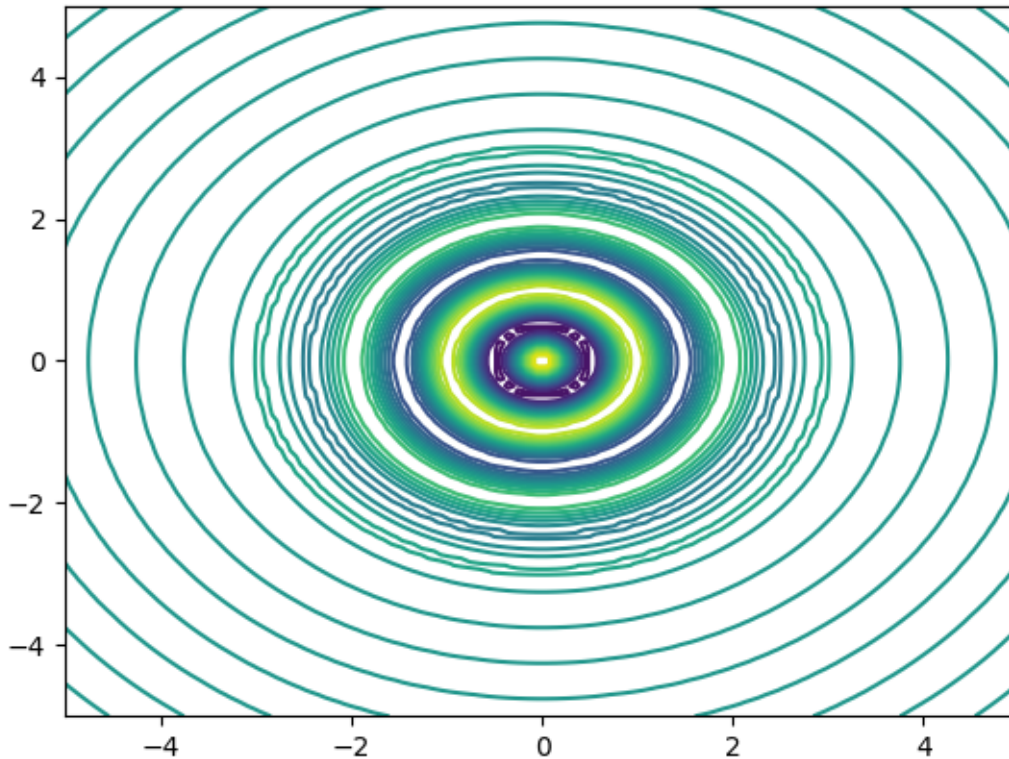


```
[22]: x = np.linspace(-5, 5, 100)
      y = np.linspace(-5, 5, 100)
      X, Y = np.meshgrid(x, y)
```

```
[23]: # Define a function to plot (example: a 2D Gaussian)
      Z = np.exp(-0.5 * (X**2 + Y**2) / 2) * np.cos(2 * np.pi * np.sqrt(X**2 + Y**2))
```

```
[24]: import numpy as np
```

```
[25]: # Create a contour plot using matplotlib.pyplot.contour
      contour_levels = 20
      contour_plot = pltV.contour(X, Y, Z, levels=contour_levels, cmap='viridis')
```

```
[26]: sex= pd.get_dummies(titanic['Sex'],drop_first=True)
sex.head(5)
```

```
[26]:      male
0    True
1   False
2   False
3   False
4    True
```

```
[27]: from sklearn.preprocessing import StandardScaler
```

```
[29]: scaler = StandardScaler()
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 10 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   Sex         891 non-null   object
4   Age         714 non-null   float64
5   Ticket      891 non-null   object
6   Fare        891 non-null   float64
7   Cabin       204 non-null   object
8   Embarked    889 non-null   object
9   AgeRange    714 non-null   category
dtypes: category(1), float64(2), int64(3), object(4)
memory usage: 64.0+ KB

```

```
[30]: titanic.drop(['AgeRange', 'Ticket', 'Cabin'], axis = 1, inplace = True)
titanic.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 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   Sex             891 non-null   object
4   Age             714 non-null   float64
5   Fare            891 non-null   float64
6   Embarked        889 non-null   object
dtypes: float64(2), int64(3), object(2)
memory usage: 48.9+ KB

```

```
[39]: from sklearn.preprocessing import LabelEncoder
LE = LabelEncoder()
titanic['Sex'] = LE.fit_transform(titanic['Sex'])
titanic['Embarked'] = LE.fit_transform(titanic['Embarked'])
```

```
[40]: titanic.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 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   Sex             891 non-null   int64
4   Age             714 non-null   float64
5   Fare            891 non-null   float64
6   Embarked        891 non-null   int64

```

```
dtypes: float64(2), int64(5)
memory usage: 48.9 KB
```

```
[41]: X = titanic.drop("Survived", axis=1)
      y = titanic["Survived"]
```

```
[42]: X.sample(10)
```

```
[42]:
```

	PassengerId	Pclass	Sex	Age	Fare	Embarked
22	23	3	0	15.0	8.0292	1
265	266	2	1	36.0	10.5000	2
177	178	1	0	50.0	28.7125	0
839	840	1	1	NaN	29.7000	0
398	399	2	1	23.0	10.5000	2
308	309	2	1	30.0	24.0000	0
153	154	3	1	40.5	14.5000	2
809	810	1	0	33.0	53.1000	2
300	301	3	0	NaN	7.7500	1
626	627	2	1	57.0	12.3500	1

```
[43]: y.sample(10)
```

```
[43]: 711    0
      364    0
      559    1
      147    0
      519    0
      378    0
      325    1
      856    1
      61     1
      468    0
      Name: Survived, dtype: int64
```

```
[44]: X_scaled = X.columns.astype(str, int)
```

```
[45]: from sklearn.model_selection import train_test_split
```

```
[65]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
      ↪random_state=1)
      X.columns = X.columns.astype(str)
```

```
[47]: from sklearn.linear_model import LogisticRegression
```

```
[48]: model=LogisticRegression()
```

```
[49]: X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     891 non-null   int64
1   Pclass          891 non-null   int64
2   Sex             891 non-null   int64
3   Age            714 non-null   float64
4   Fare           891 non-null   float64
5   Embarked       891 non-null   int64
dtypes: float64(2), int64(4)
memory usage: 41.9 KB
```

```
[50]: Pcl= pd.get_dummies(titanic['Pclass'],drop_first=True)
      Pcl.head(5)
```

```
[50]:      2      3
0  False  True
1  False False
2  False  True
3  False False
4  False  True
```

```
[51]: X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     891 non-null   int64
1   Pclass          891 non-null   int64
2   Sex             891 non-null   int64
3   Age            714 non-null   float64
4   Fare           891 non-null   float64
5   Embarked       891 non-null   int64
dtypes: float64(2), int64(4)
memory usage: 41.9 KB
```

```
[66]: X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 623 entries, 114 to 37
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     623 non-null   int64
```

```

1   Pclass      623 non-null   int64
2   Sex         623 non-null   int64
3   Age         496 non-null   float64
4   Fare        623 non-null   float64
5   Embarked    623 non-null   int64
dtypes: float64(2), int64(4)
memory usage: 34.1 KB

```

```
[73]: median_age = X_train['Age'].median()
      X_train['Age'] = X_train['Age'].fillna(median_age)
```

```
[74]: X_train.isnull().sum()
```

```
[74]: PassengerId    0
      Pclass        0
      Sex           0
      Age           0
      Fare          0
      Embarked      0
      dtype: int64
```

```
[75]: model.fit(X_train, y_train)
```

```

C:\Users\lenovo\AppData\Roaming\Python\Python311\site-
packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max_iter) or scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(

```
[75]: LogisticRegression()
```

```
[77]: model.fit(X_train, y_train)
```

```

C:\Users\lenovo\AppData\Roaming\Python\Python311\site-
packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max_iter) or scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
regression
    n_iter_i = _check_optimize_result(
```

```
[77]: LogisticRegression()
```

```
[79]: predictions = model.predict(X_train)
```

```
[80]: from sklearn.metrics import accuracy_score
```

```
[82]: accuracy = accuracy_score(y_train, predictions)
      print(f'Accuracy: {accuracy:.2f}')
```

Accuracy: 0.81

```
[84]: from sklearn.metrics import classification_report
```

```
[86]: print(classification_report(y_train, predictions))
```

	precision	recall	f1-score	support
0	0.84	0.87	0.85	396
1	0.75	0.71	0.73	227
accuracy			0.81	623
macro avg	0.80	0.79	0.79	623
weighted avg	0.81	0.81	0.81	623

```
[ ]:
```