

Suliman

November 15, 2025

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

```
[3]: from IPython.display import Image, display
```

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

```
[5]: import matplotlib.pyplot as pyplot
```

```
[6]: import seaborn as sbn
```

```
[7]: sbn.set(style="whitegrid")
```

```
[8]: df=pd.read_csv("train.csv")
```

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

```
[9]:
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S

```
[10]: print("Section 2 - Explore Data")
```

Section 2 - Explore Data

```
[11]: print("Dataset shape:", df.shape)
```

Dataset shape: (891, 12)

```
[12]: print("\nData info:", df.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 info: None

```
[13]: print(df.isnull().sum())
```

```
PassengerId     0
Survived        0
Pclass          0
Name            0
Sex             0
Age            177
SibSp           0
Parch           0
Ticket          0
Fare            0
Cabin          687
Embarked        2
dtype: int64
```

```
[14]: print(" Fix Missing Values")
```

Fix Missing Values

```
[15]: missing_h = df.isnull().mean() * 100
```

```
[16]: print(missing_h)
```

```
PassengerId    0.000000
Survived        0.000000
Pclass          0.000000
Name            0.000000
Sex             0.000000
Age            19.865320
SibSp           0.000000
Parch           0.000000
Ticket          0.000000
Fare            0.000000
Cabin          77.104377
Embarked        0.224467
dtype: float64
```

```
[17]: cols_to_drop = missing_h[missing_h > 30].index
```

```
[18]: df = df.drop(columns=cols_to_drop)
```

```
[19]: numeric_cols = df.select_dtypes(include=["float64", "int64"]).columns
```

```
[20]: for col in numeric_cols:
      df[col] = df[col].fillna(df[col].median())
```

```
[21]: print(df.isnull().sum())
```

```
PassengerId    0
Survived        0
Pclass          0
Name            0
Sex             0
Age            0
SibSp           0
Parch           0
Ticket          0
Fare            0
Embarked        2
dtype: int64
```

```
[22]: print("Detect Outliers")
```

```
Detect Outliers
```

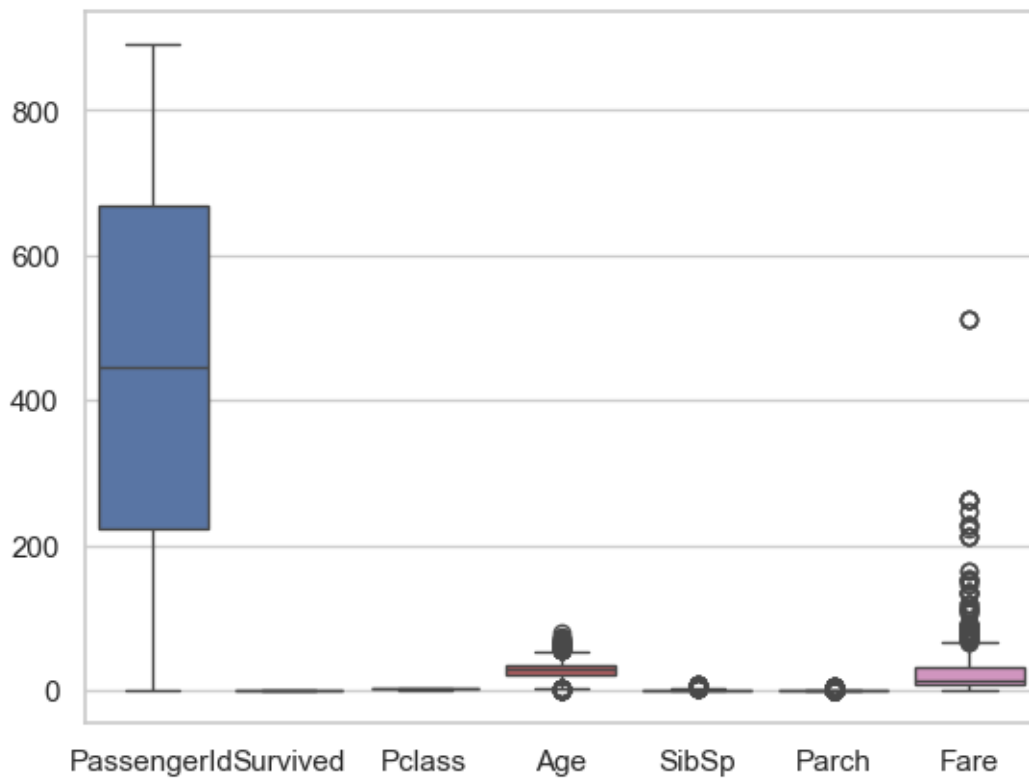
```
[23]: pyplot.figure(figsize=(12,5))
```

```
[23]: <Figure size 1200x500 with 0 Axes>
```

<Figure size 1200x500 with 0 Axes>

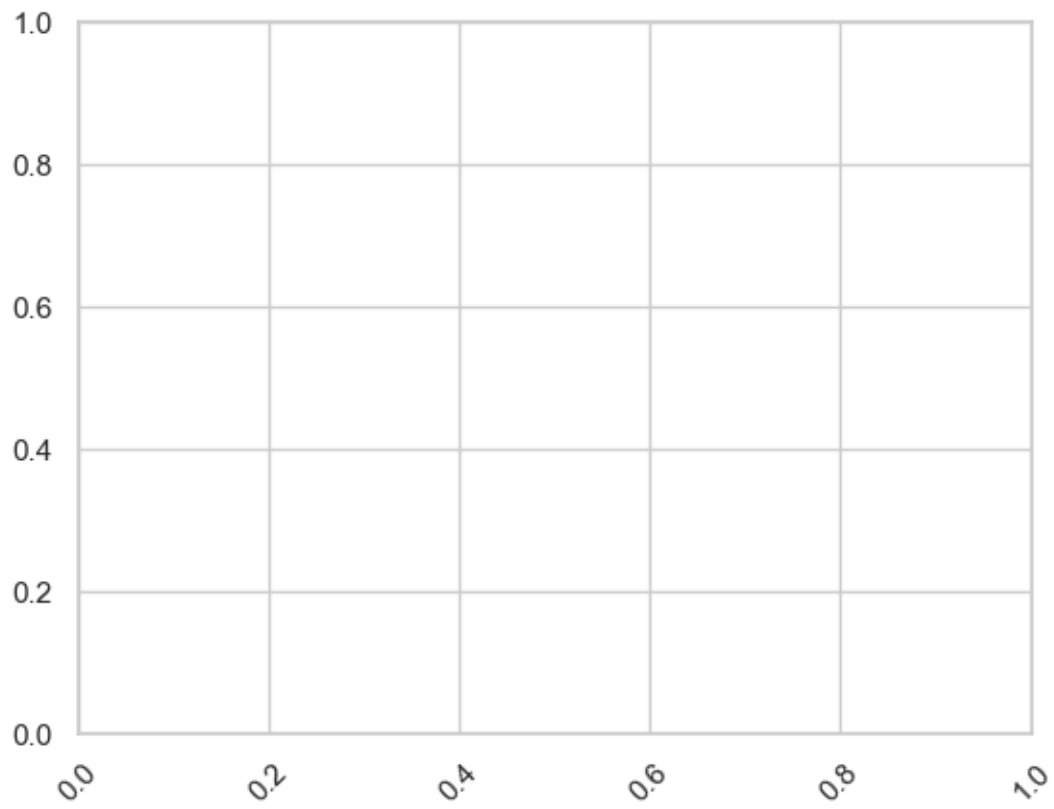
```
[24]: sns.boxplot(data=df[numeric_cols])
```

[24]: <Axes: >



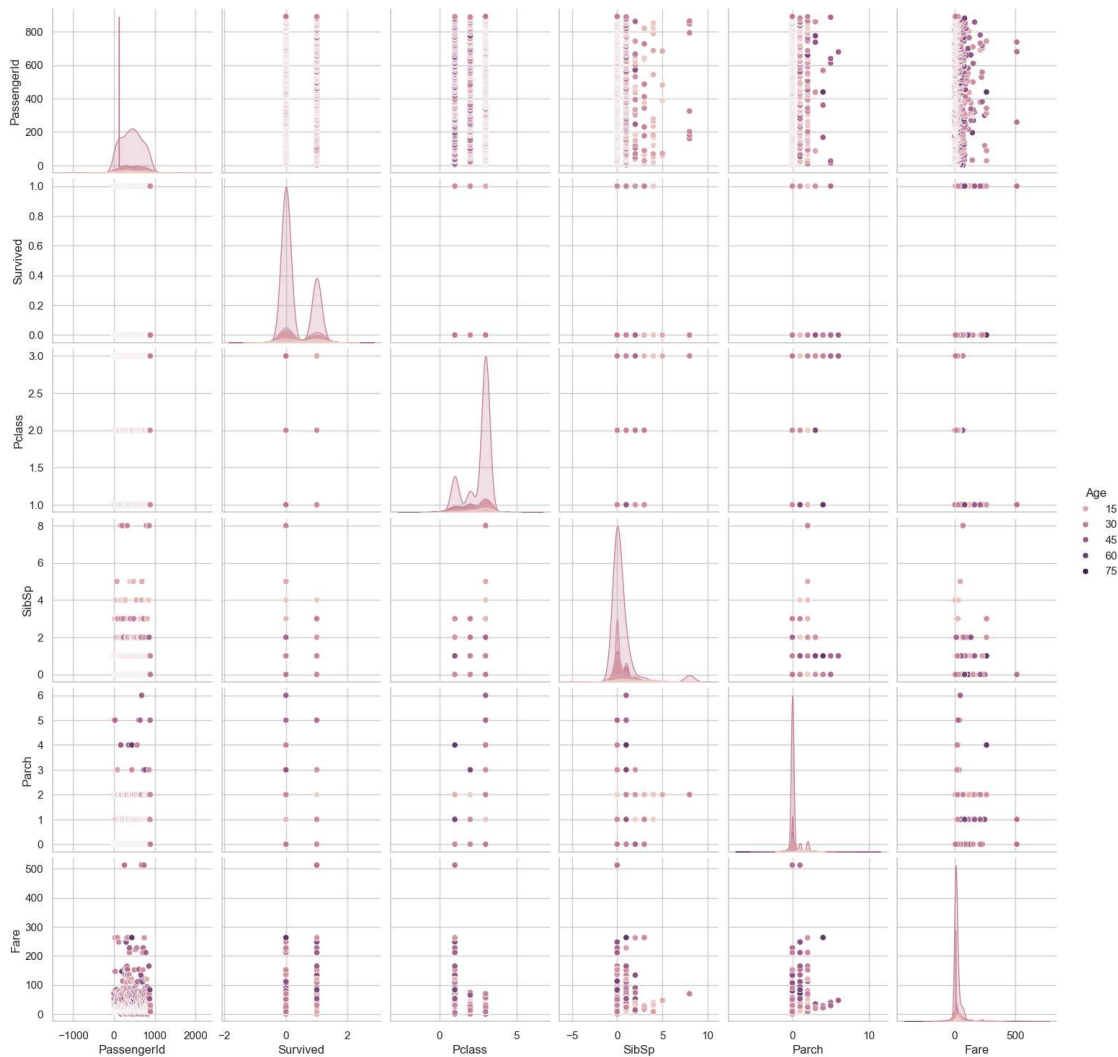
```
[25]: plt.xticks(rotation=45)
```

```
[25]: (array([0. , 0.2, 0.4, 0.6, 0.8, 1. ]),  
      [Text(0.0, 0, '0.0'),  
       Text(0.2, 0, '0.2'),  
       Text(0.4, 0, '0.4'),  
       Text(0.6000000000000001, 0, '0.6'),  
       Text(0.8, 0, '0.8'),  
       Text(1.0, 0, '1.0')])
```



```
[28]: sbn.pairplot(data=df, hue='Age')
```

```
[28]: <seaborn.axisgrid.PairGrid at 0x11c991e50>
```



```
[29]: plt.show()
```

```
[30]: def outliers_iqr(column):
    Q1 = column.quantile(0.25)
    Q3 = column.quantile(0.75)
    IQR = Q3 - Q1
    low = Q1 - 1.5 * IQR
    high = Q3 + 1.5 * IQR
    return (column < low) | (column > high)
```

```
[31]: data_iqr = df.copy()
```

```
[32]: for col in numeric_cols:
    data_iqr[col + "_iqr_outlier"] = outliers_iqr(data_iqr[col])
```

```
[33]: data_iqr[data_iqr.filter(like="_iqr_outlier").any(axis=1)].head()
```

```
[33]:
```

	PassengerId	Survived	Pclass	\
1	2	1	1	
7	8	0	3	
8	9	1	3	
10	11	1	3	
11	12	1	1	

	Name	Sex	Age	SibSp	\
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
7	Palsson, Master. Gosta Leonard	male	2.0	3	
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	
10	Sandstrom, Miss. Marguerite Rut	female	4.0	1	
11	Bonnell, Miss. Elizabeth	female	58.0	0	

	Parch	Ticket	Fare	Embarked	PassengerId_iqr_outlier	\
1	0	PC 17599	71.2833	C	False	
7	1	349909	21.0750	S	False	
8	2	347742	11.1333	S	False	
10	1	PP 9549	16.7000	S	False	
11	0	113783	26.5500	S	False	

	Survived_iqr_outlier	Pclass_iqr_outlier	Age_iqr_outlier	\
1	False	False	False	
7	False	False	True	
8	False	False	False	
10	False	False	False	
11	False	False	True	

	SibSp_iqr_outlier	Parch_iqr_outlier	Fare_iqr_outlier
1	False	False	True
7	True	True	False
8	False	True	False
10	False	True	False
11	False	False	False

```
[34]: def outliers_z(col, threshold=3):
      mean = col.mean()
      std = col.std()
      z = (col - mean) / std
      return abs(z) > threshold
```

```
[35]: data_z = df.copy()
```

```
[36]: for col in numeric_cols:
      data_z[col + "_z_outlier"] = outliers_z(data_z[col])
```

```
[37]: data_z[data_z.filter(like="_z_outlier").any(axis=1)].head()
```

```
[37]:
```

	PassengerId	Survived	Pclass	\
13	14	0	3	
16	17	0	3	
25	26	1	3	
27	28	0	1	
50	51	0	3	

	Name	Sex	Age	SibSp	\
13	Andersson, Mr. Anders Johan	male	39.0	1	
16	Rice, Master. Eugene	male	2.0	4	
25	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...	female	38.0	1	
27	Fortune, Mr. Charles Alexander	male	19.0	3	
50	Panula, Master. Juha Niilo	male	7.0	4	

	Parch	Ticket	Fare	Embarked	PassengerId_z_outlier	\
13	5	347082	31.2750	S	False	
16	1	382652	29.1250	Q	False	
25	5	347077	31.3875	S	False	
27	2	19950	263.0000	S	False	
50	1	3101295	39.6875	S	False	

	Survived_z_outlier	Pclass_z_outlier	Age_z_outlier	SibSp_z_outlier	\
13	False	False	False	False	
16	False	False	False	True	
25	False	False	False	False	
27	False	False	False	False	
50	False	False	False	True	

	Parch_z_outlier	Fare_z_outlier
13	True	False
16	False	False
25	True	False
27	False	True
50	False	False

```
[38]: print("Section 5 - Fix Outliers")
```

Section 5 - Fix Outliers

```
[39]: def replace_outliers(column):
    Q1 = column.quantile(0.25)
    Q3 = column.quantile(0.75)
    IQR = Q3 - Q1
    low = Q1 - 1.5 * IQR
    high = Q3 + 1.5 * IQR
    return column.where((column >= low) & (column <= high), np.nan)
```

```
[40]: clean_data = df.copy()
```

```
[41]: for col in numeric_cols:
      clean_data[col] = replace_outliers(clean_data[col])
```

```
[42]: print(clean_data.isnull().sum())
```

```
PassengerId      0
Survived          0
Pclass           0
Name             0
Sex              0
Age             66
SibSp            46
Parch            213
Ticket           0
Fare            116
Embarked         2
dtype: int64
```

```
[43]: from sklearn.impute import KNNImputer
```

```
[45]: from sklearn.impute import KNNImputer
```

```
[47]: from sklearn.impute import KNNImputer
```

```
[48]: imputer = KNNImputer(n_neighbors=3)
```

```
[49]: data_imputed = pd.DataFrame(
      imputer.fit_transform(clean_data[numeric_cols]),
      columns=numeric_cols
    )
```

```
[50]: data_imputed.head()
```

```
[50]:
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	1.0	0.0	3.0	22.0	1.0	0.0	7.250000
1	2.0	1.0	1.0	38.0	1.0	0.0	27.408333
2	3.0	1.0	3.0	26.0	0.0	0.0	7.925000
3	4.0	1.0	1.0	35.0	1.0	0.0	53.100000
4	5.0	0.0	3.0	35.0	0.0	0.0	8.050000

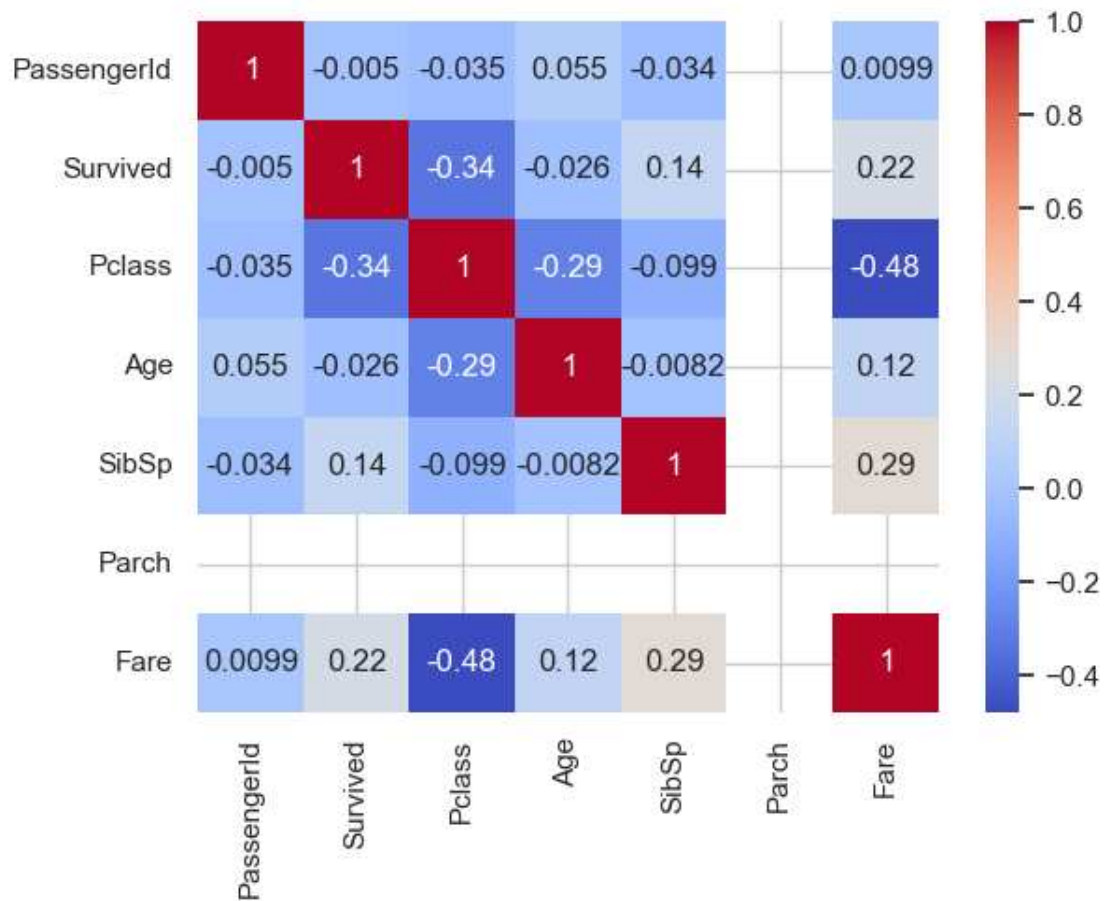
```
[51]: pyplot.figure(figsize=(10,7))
```

```
[51]: <Figure size 1000x700 with 0 Axes>
```

```
<Figure size 1000x700 with 0 Axes>
```

```
[52]: sbn.heatmap(data_imputed.corr(), annot=True, cmap="coolwarm")
```

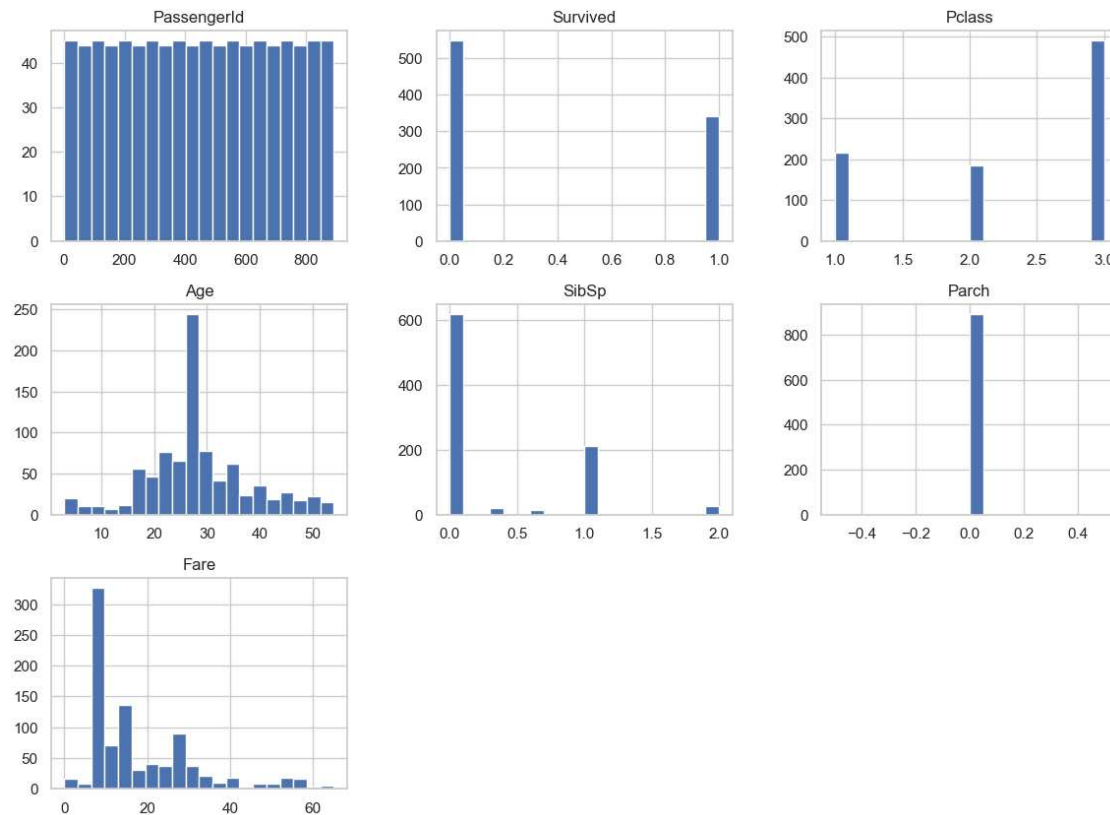
[52]: <Axes: >



[53]: `pyplot.show()`

[54]: `data_imputed.hist(figsize=(14,10), bins=20)`

[54]: `array([[<Axes: title={'center': 'PassengerId'}>,
<Axes: title={'center': 'Survived'}>,
<Axes: title={'center': 'Pclass'}>],
[<Axes: title={'center': 'Age'}>,
<Axes: title={'center': 'SibSp'}>,
<Axes: title={'center': 'Parch'}>],
[<Axes: title={'center': 'Fare'}>, <Axes: >, <Axes: >]],
dtype=object)`



```
[55]: plt.tight_layout()
```

<Figure size 640x480 with 0 Axes>

```
[56]: plt.show()
```

```
[57]: clean_numeric = data_imputed.copy()
```

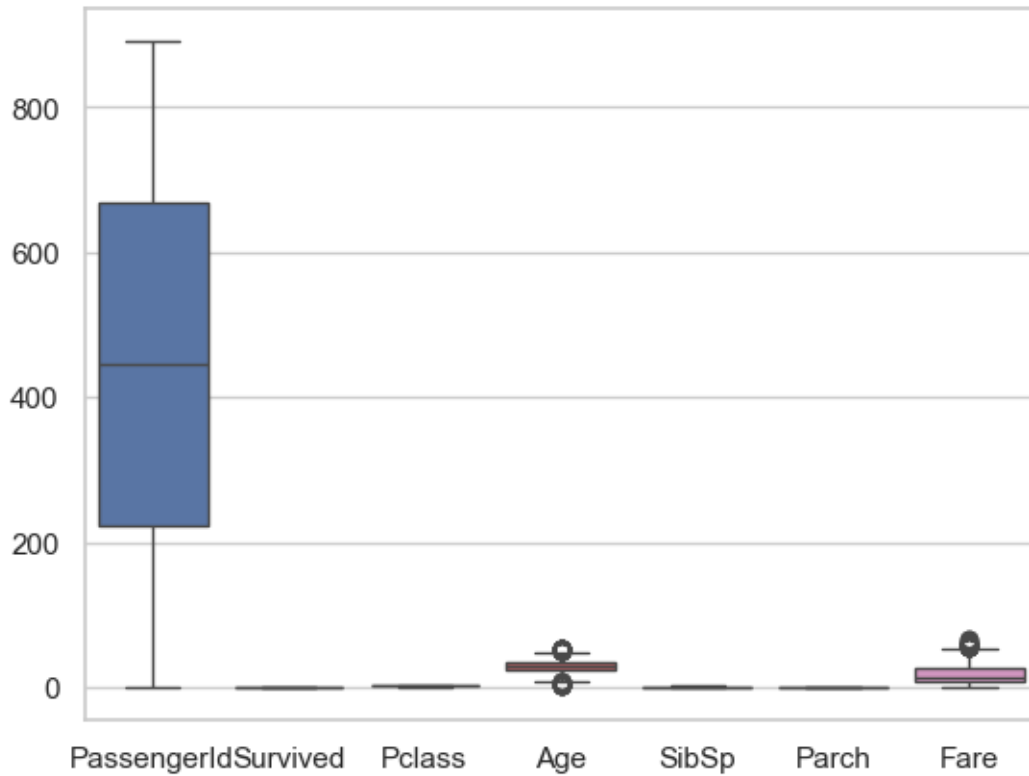
```
[58]: plt.figure(figsize=(12, 5))
```

```
[58]: <Figure size 1200x500 with 0 Axes>
```

<Figure size 1200x500 with 0 Axes>

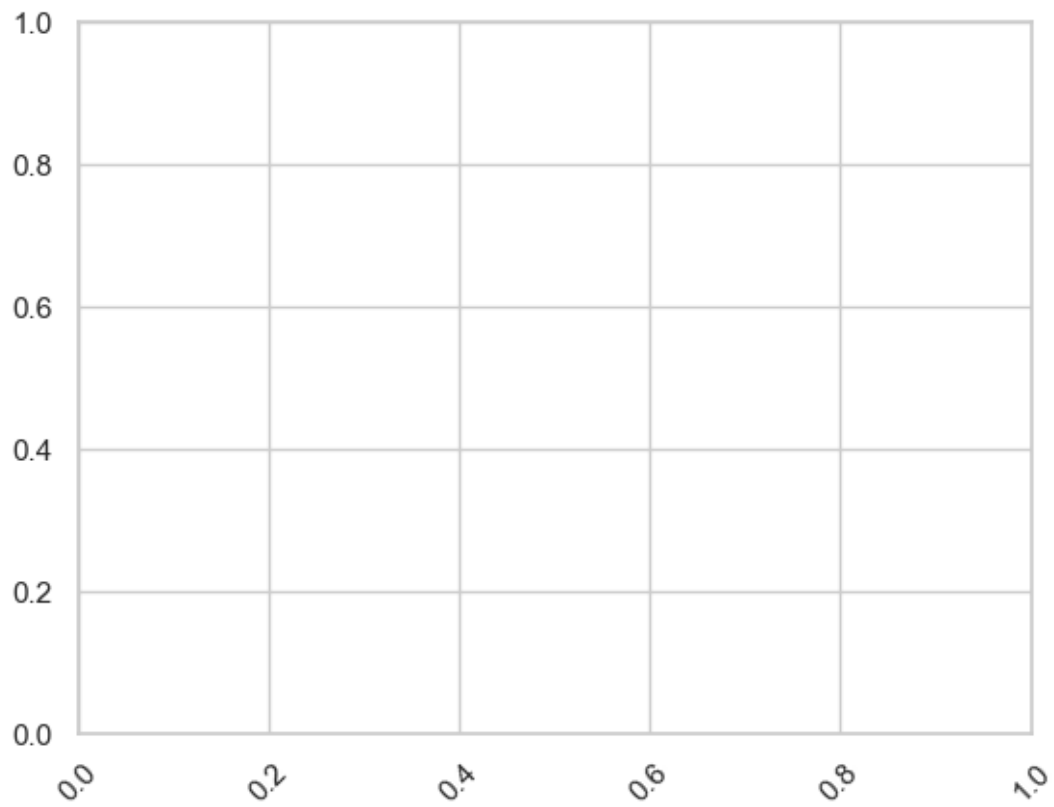
```
[59]: sns.boxplot(data=clean_numeric)
```

```
[59]: <Axes: >
```



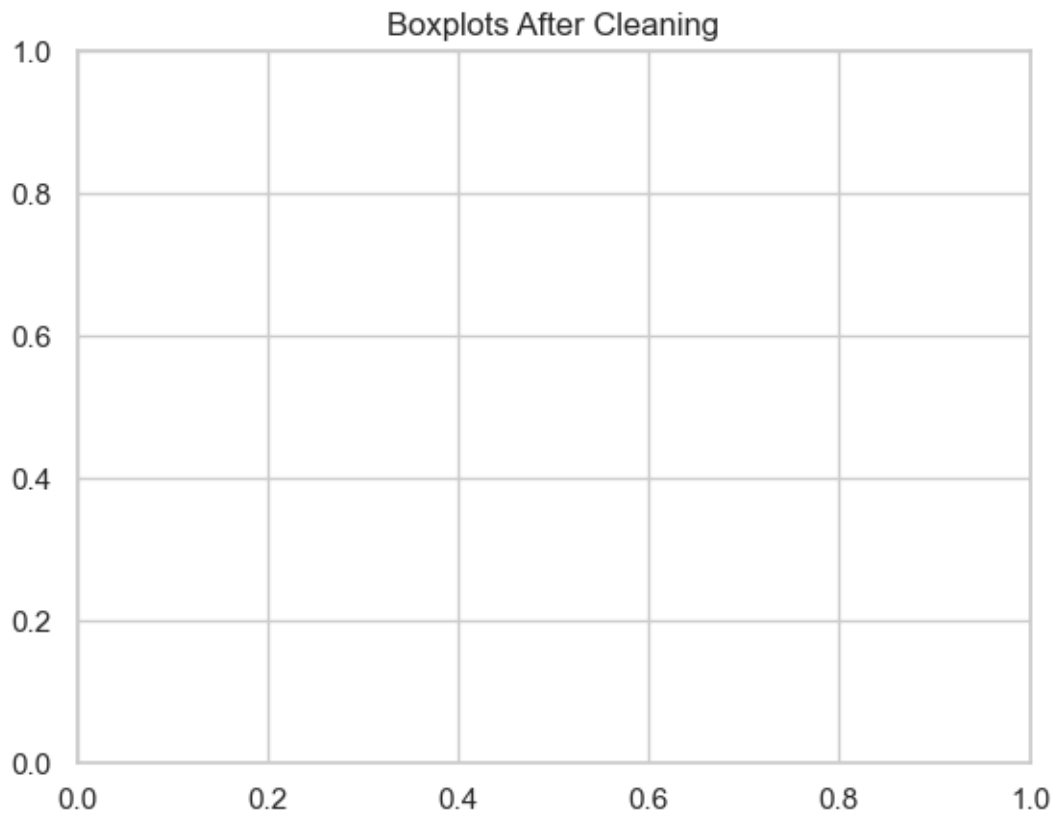
```
[60]: pyplot.xticks(rotation=45)
```

```
[60]: (array([0. , 0.2, 0.4, 0.6, 0.8, 1. ]),
      [Text(0.0, 0, '0.0'),
       Text(0.2, 0, '0.2'),
       Text(0.4, 0, '0.4'),
       Text(0.60000000000000001, 0, '0.6'),
       Text(0.8, 0, '0.8'),
       Text(1.0, 0, '1.0')])
```



```
[61]: pyplot.title("Boxplots After Cleaning")
```

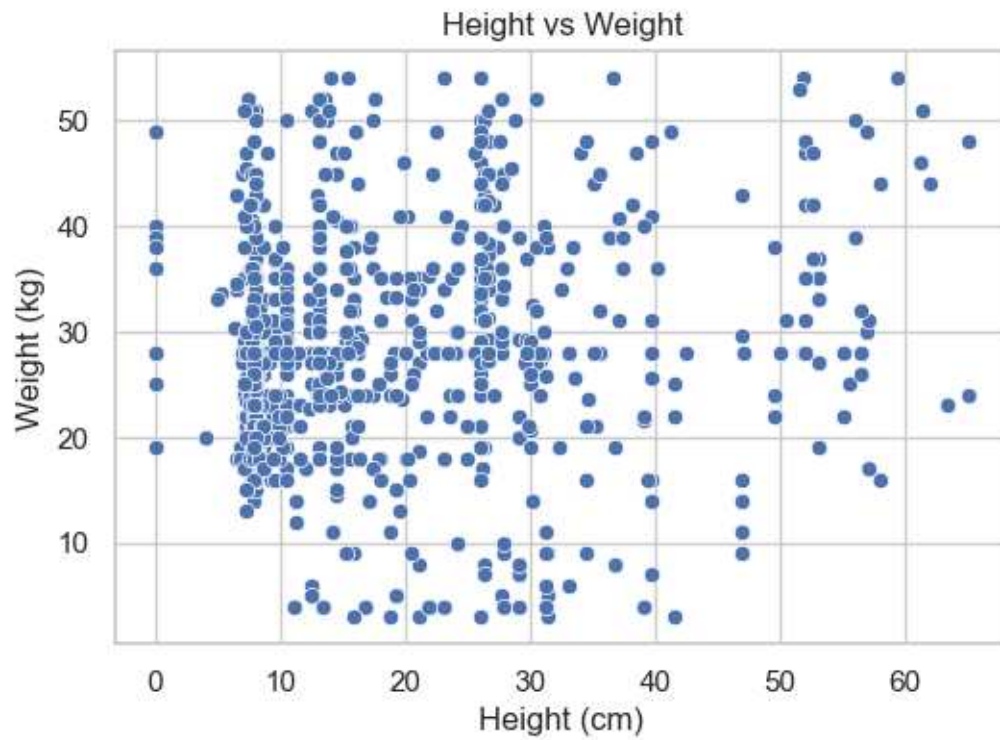
```
[61]: Text(0.5, 1.0, 'Boxplots After Cleaning')
```



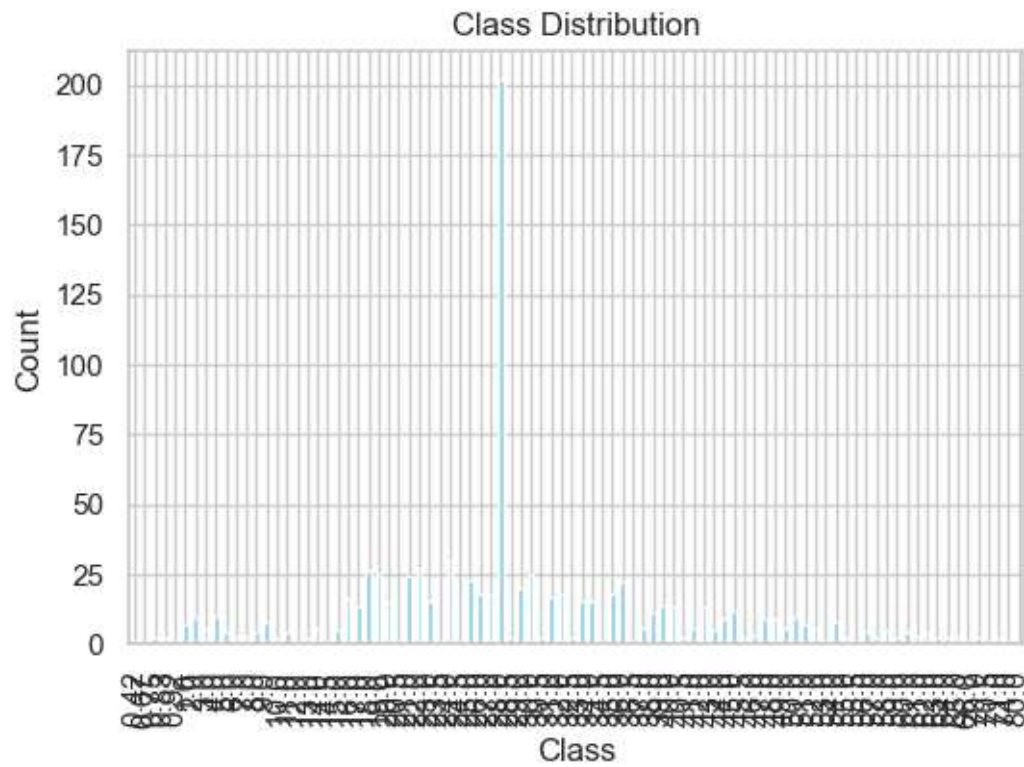
```
[62]: plt.show()
```

```
[64]: clean_numeric = data_imputed.copy()

plt.figure(figsize=(6, 4))
sns.scatterplot(x=clean_numeric["Fare"], y=clean_numeric["Age"])
plt.title("Height vs Weight")
plt.xlabel("Height (cm)")
plt.ylabel("Weight (kg)")
plt.show()
```



```
[66]: df["Age"].value_counts().sort_index().plot(
      kind="bar", figsize=(6, 4), color="skyblue"
    )
    pyplot.title("Class Distribution")
    pyplot.xlabel("Class")
    pyplot.ylabel("Count")
    pyplot.show()
```



[]:

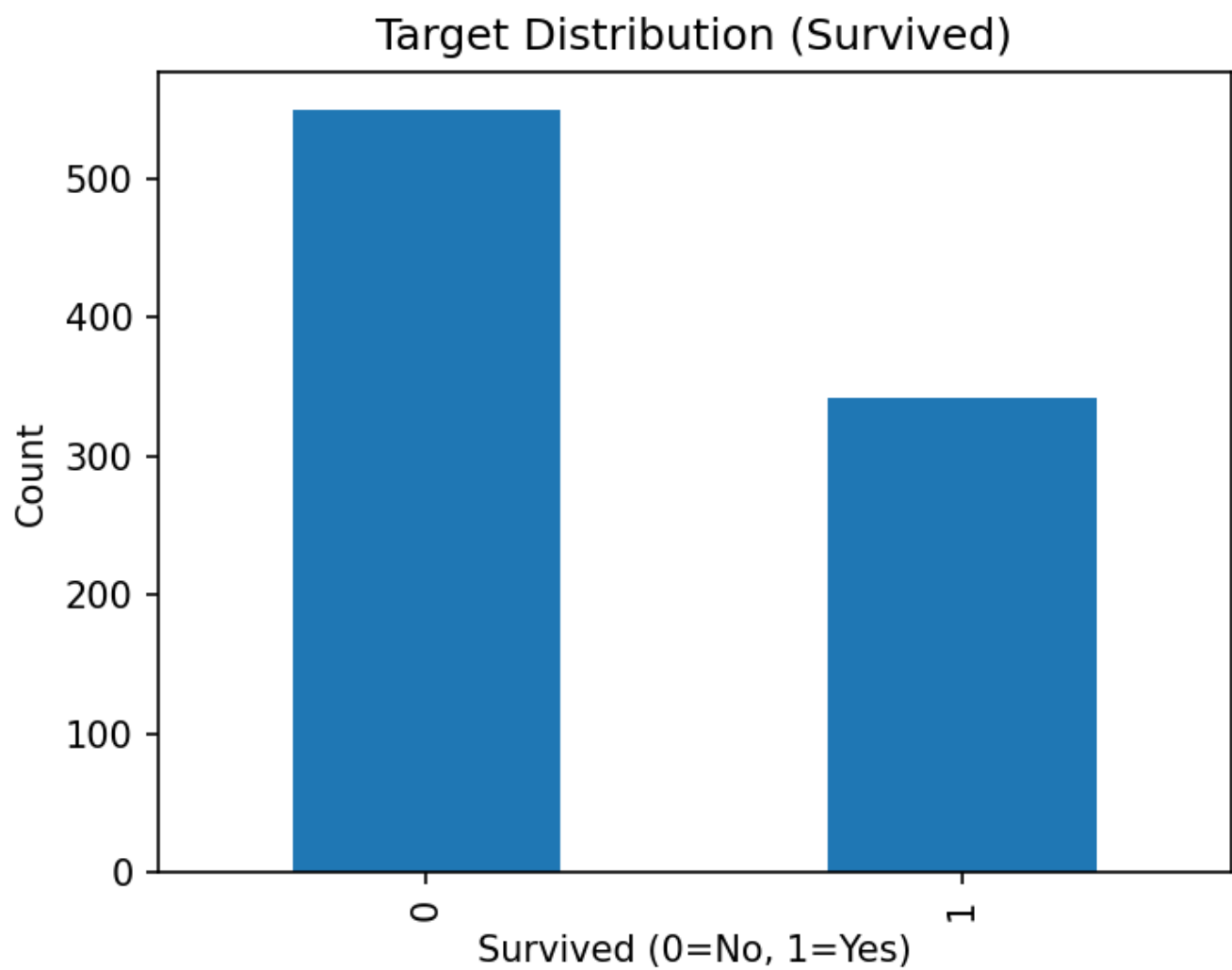
Lecture 10 - Titanic (train.csv) Report

Dataset: train.csv | Target: Survived

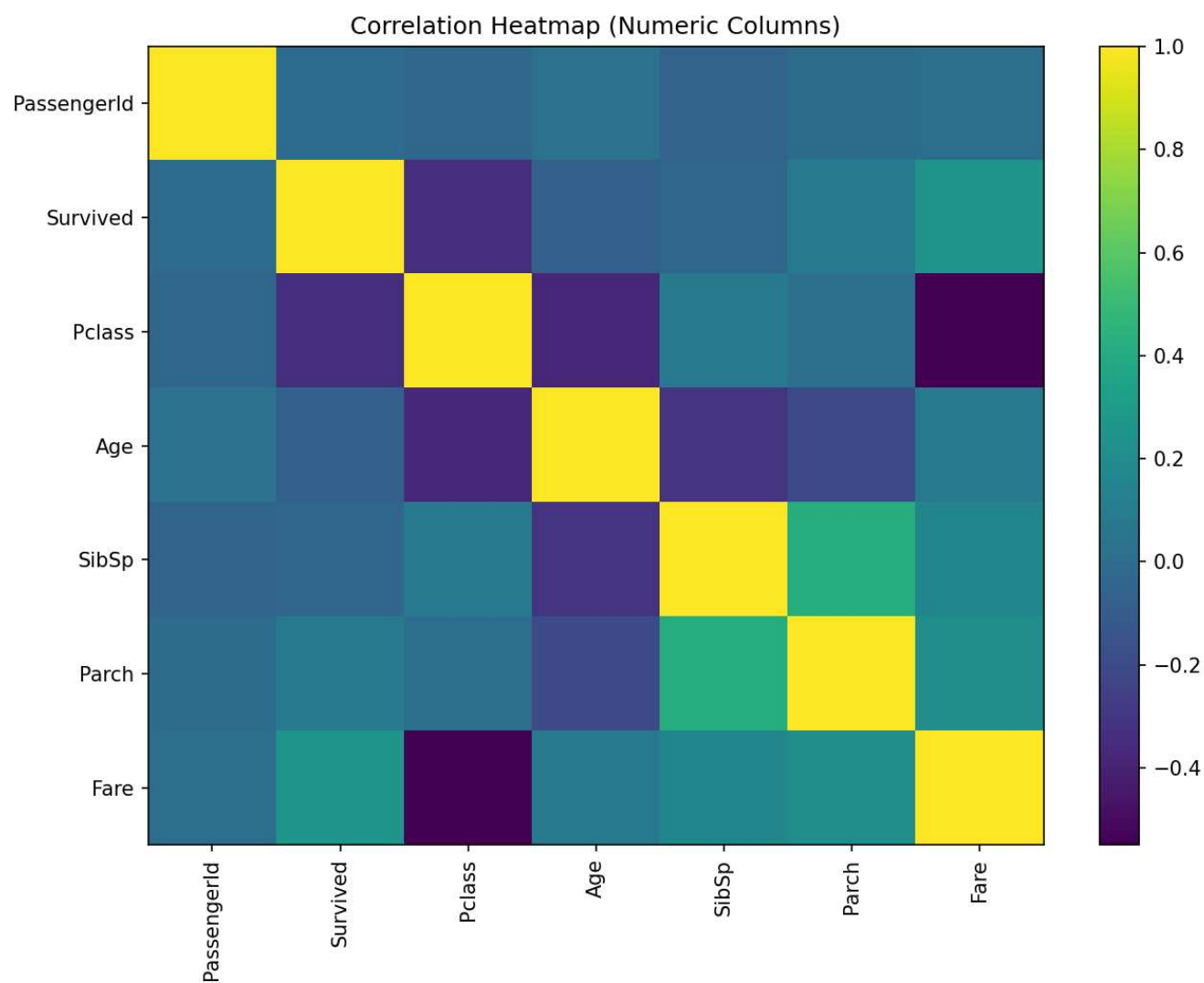
Rows: 891 | Columns: 12

Used features: Pclass, Sex, Age, SibSp, Parch, Fare, Embarked

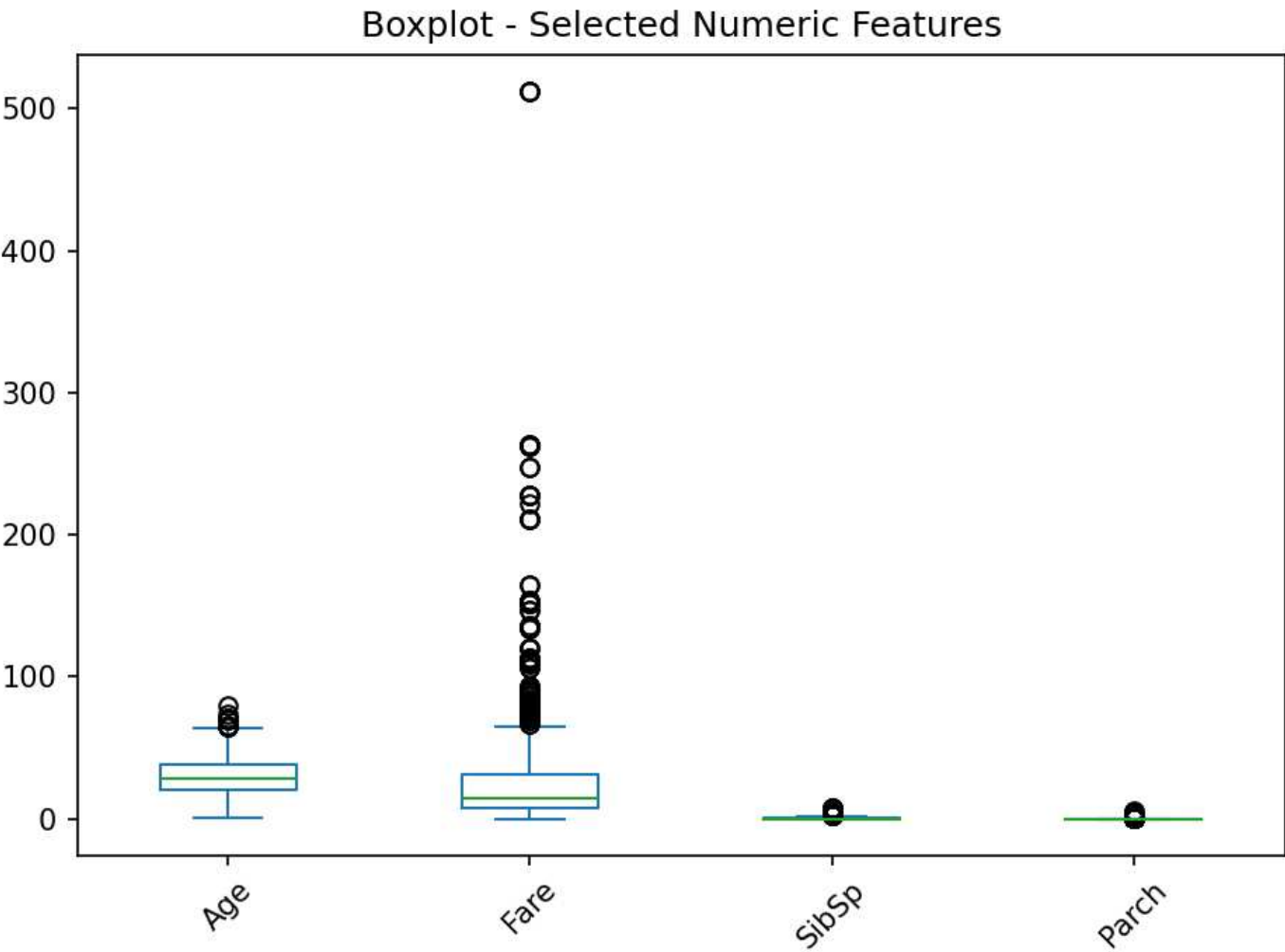
Target Distribution



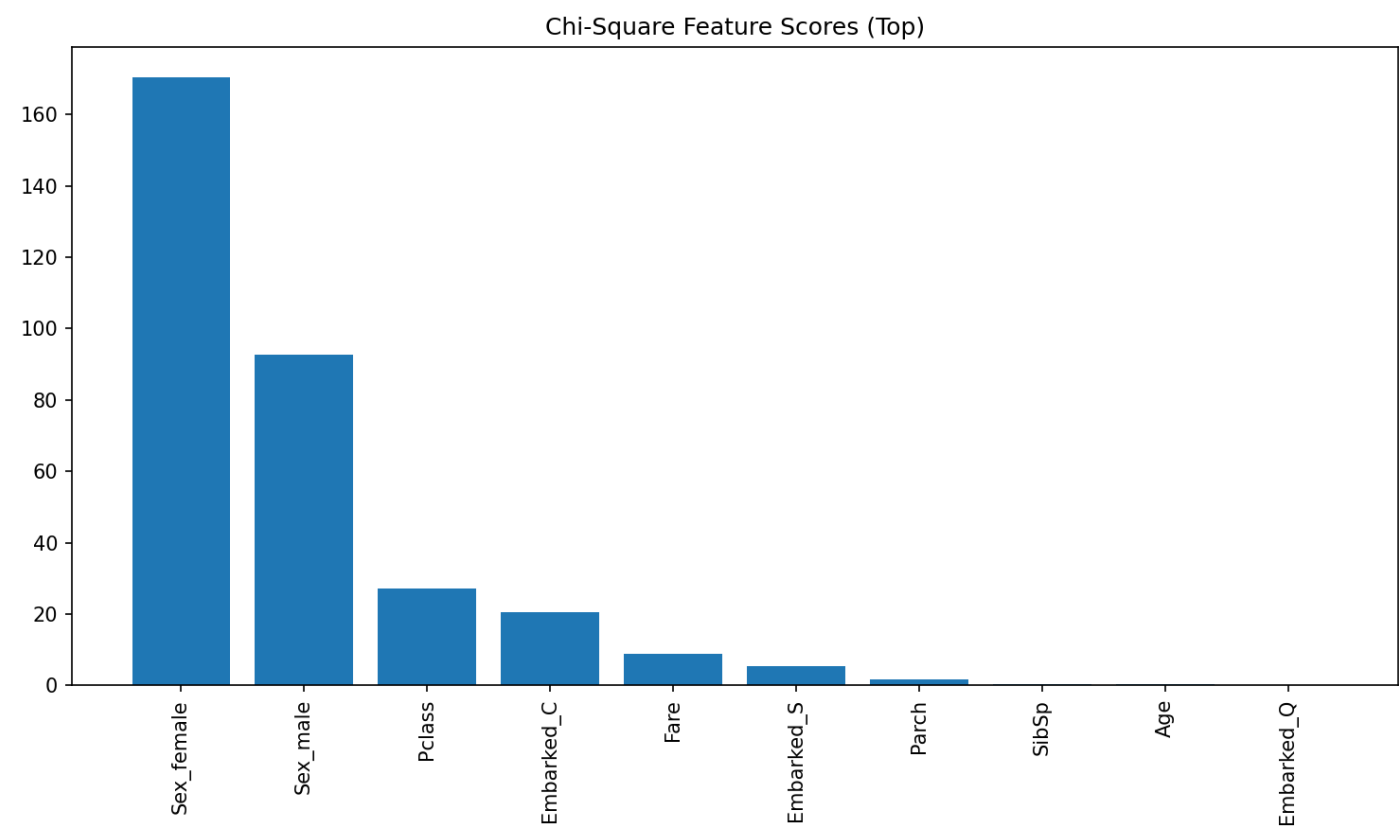
Correlation Heatmap (Numeric)



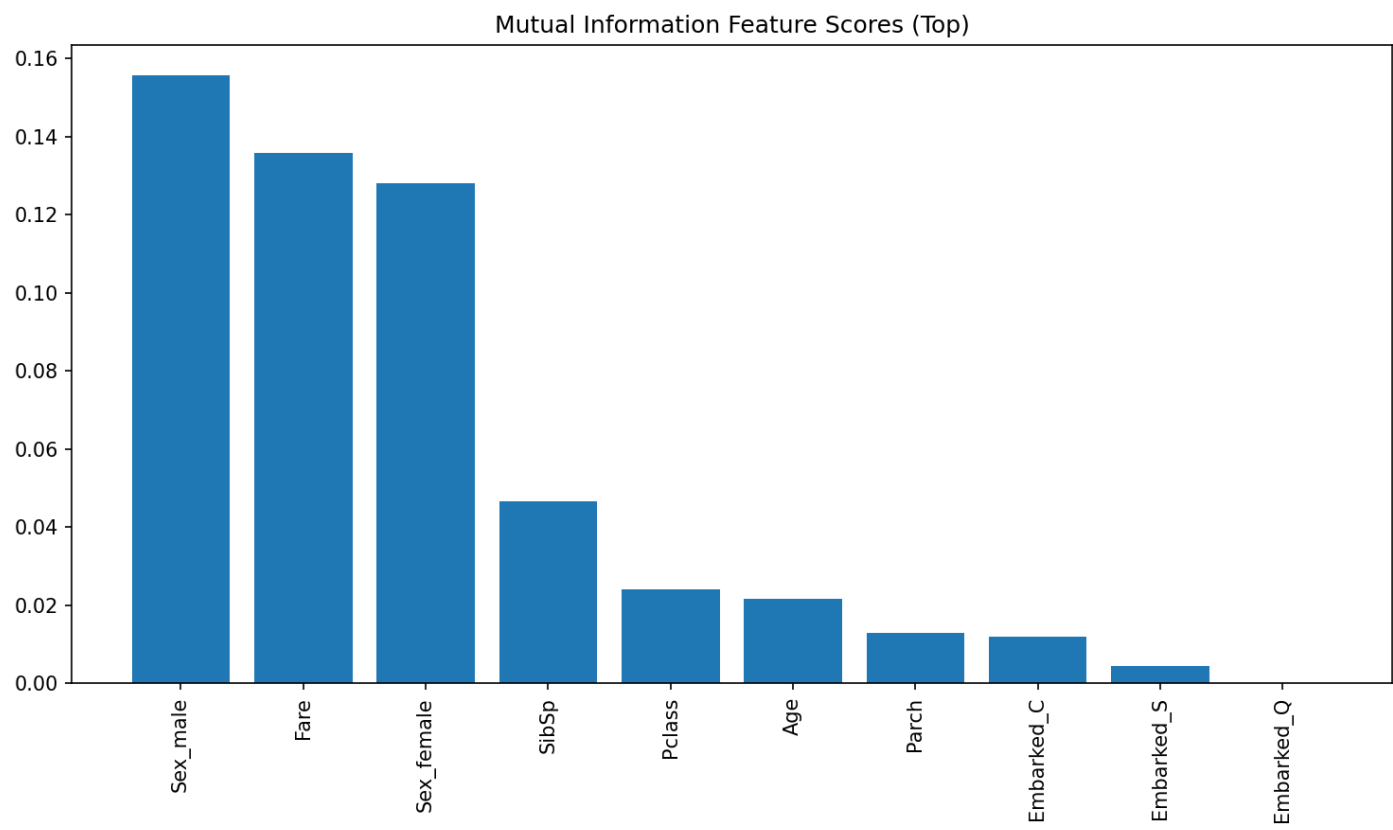
Boxplot - Numeric Features



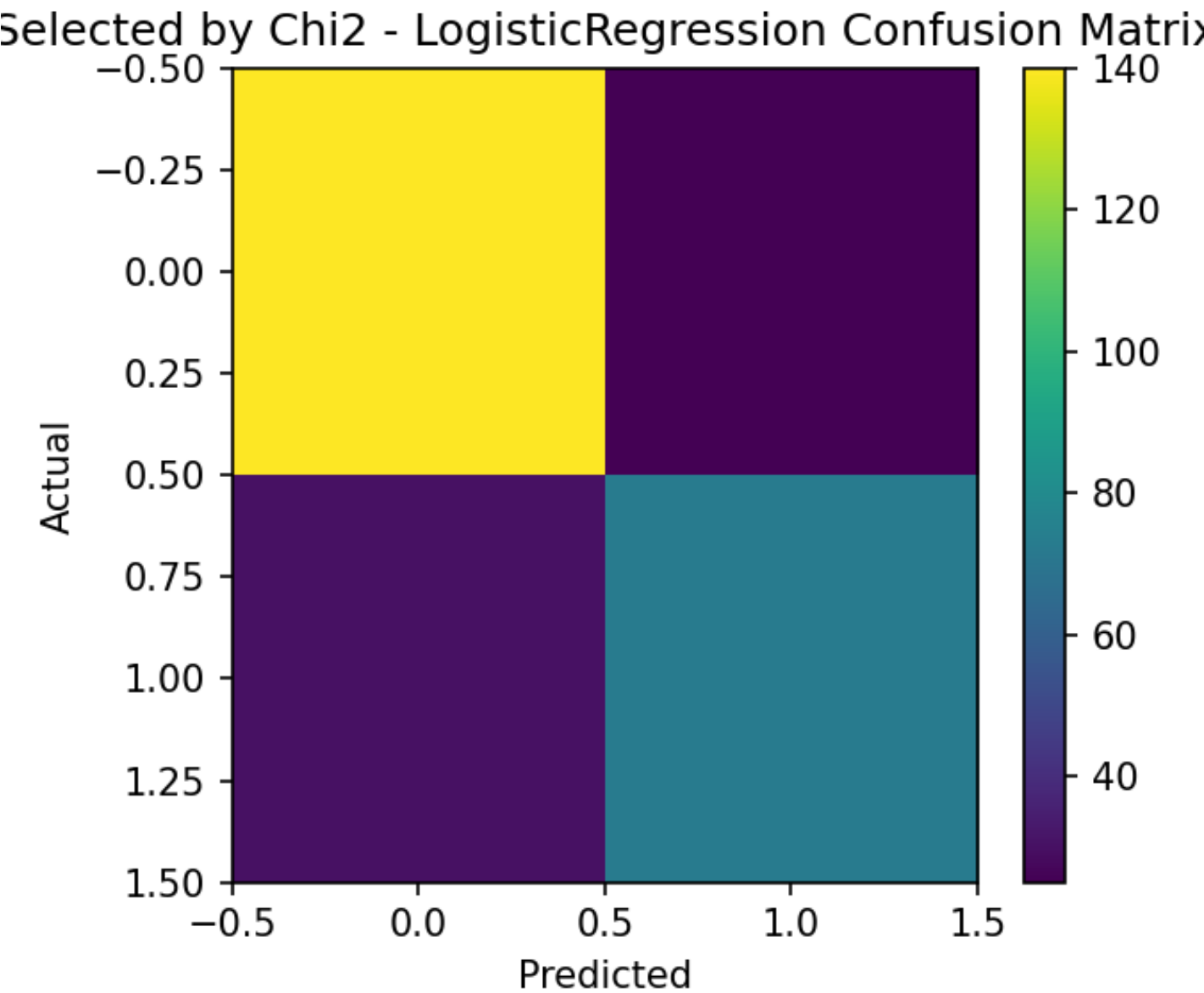
Chi-Square Feature Scores



Mutual Information Feature Scores

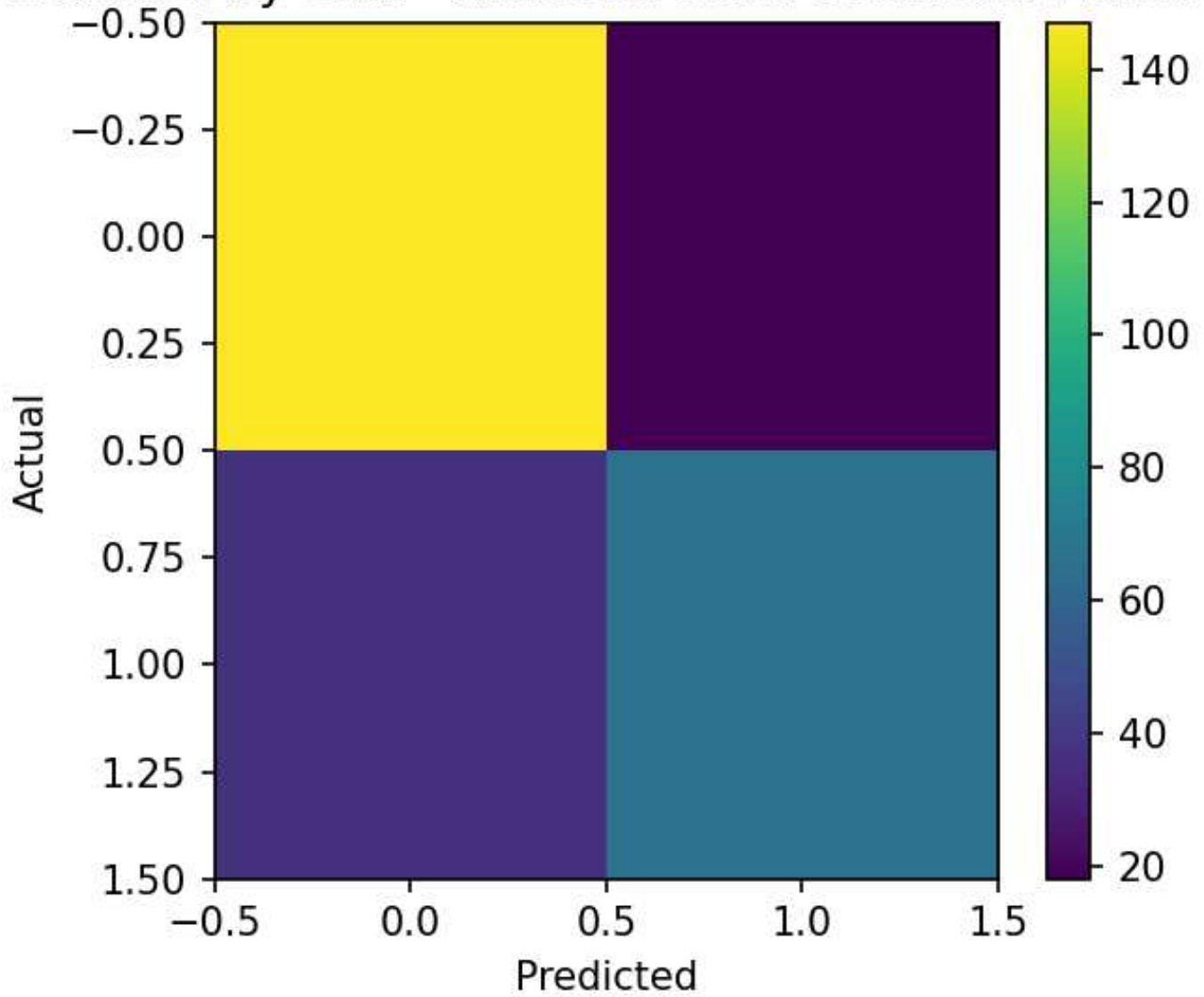


Selected by Chi2 - LogisticRegression Confusion Matrix

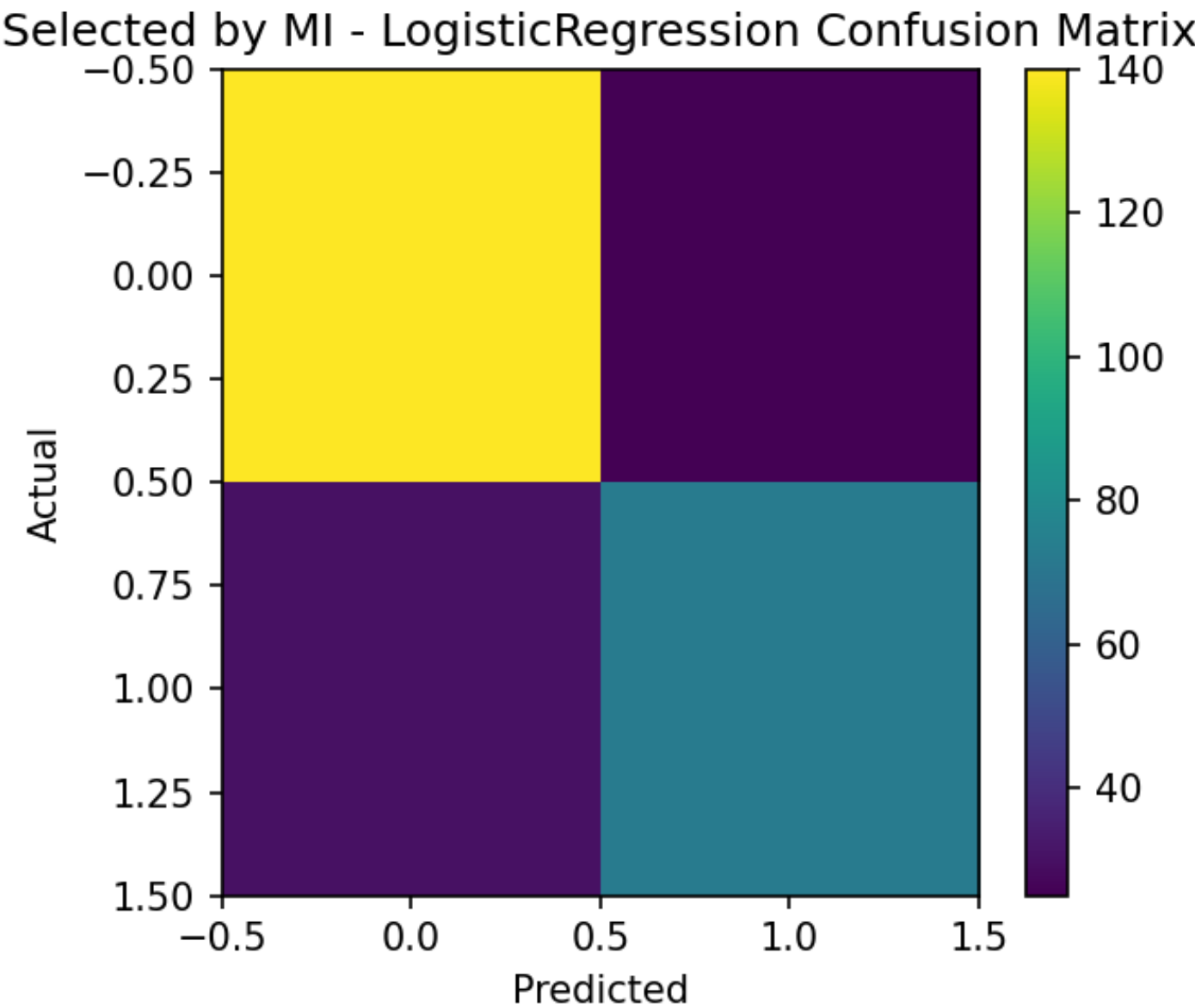


Selected by Chi2 - RandomForest Confusion Matrix

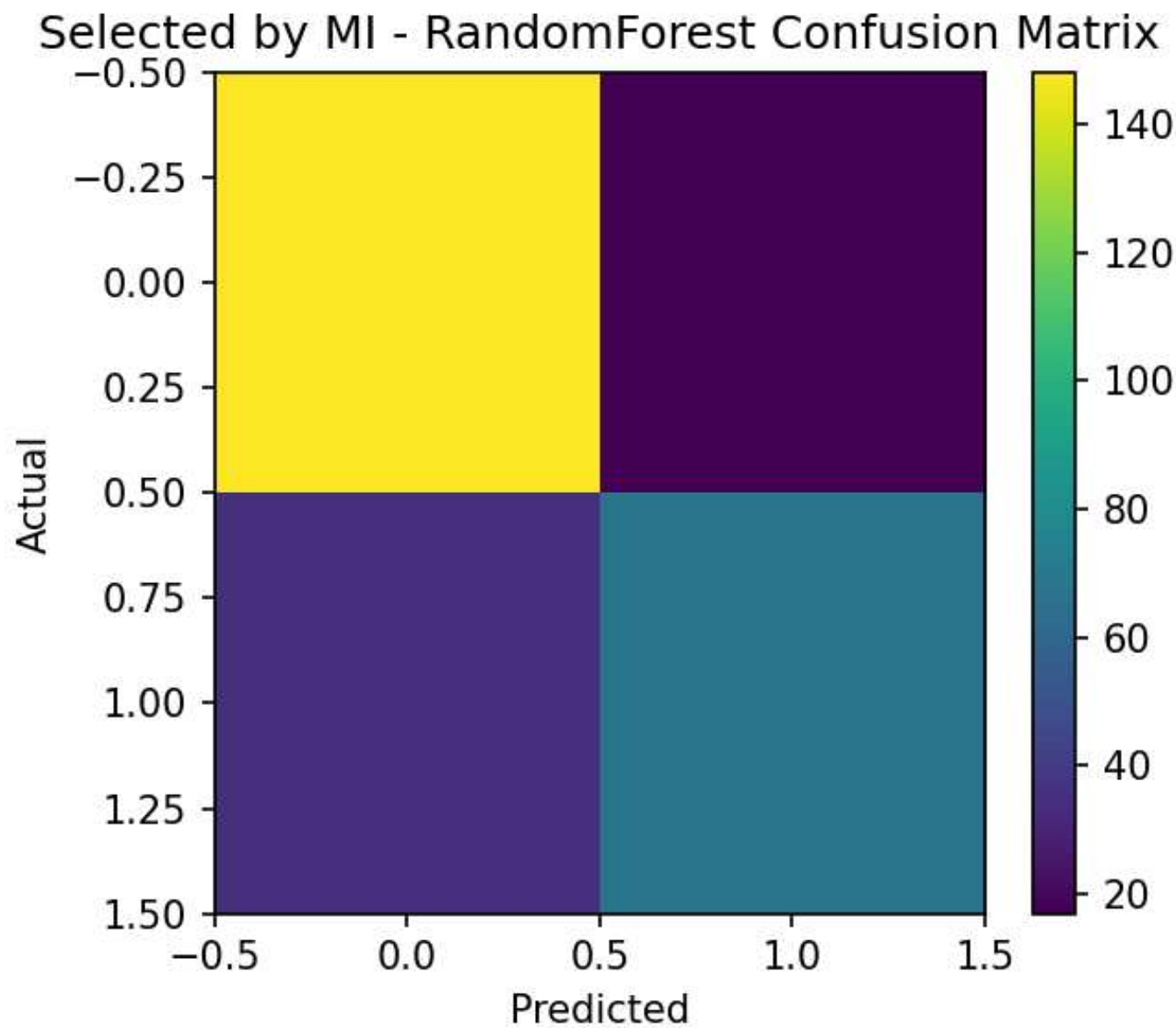
Selected by Chi2 - RandomForest Confusion Matrix



Selected by MI - LogisticRegression Confusion Matrix



Selected by MI - RandomForest Confusion Matrix



Model Results (Accuracy + Classification Report)

Selected by Chi-Square | Top features: Sex_female, Sex_male, Pclass, Embarked_C, Fare, Embarked_S, Parch, SibSp, Age, Embarked_Q

LogisticRegression | Accuracy: 0.7948

	precision	recall	f1-score	support	0	0.82	0.85	
0.84	165	1	0.74	0.71	0.73	103	accuracy	
0.79	268	macro avg	0.78	0.78	0.78	268	weighted avg	0.79
0.79	0.79	268						

RandomForest | Accuracy: 0.7948

	precision	recall	f1-score	support	0	0.80	0.89	
0.84	165	1	0.79	0.64	0.71	103	accuracy	
0.79	268	macro avg	0.79	0.77	0.77	268	weighted avg	0.79
0.79	0.79	268						

Selected by Mutual Info | Top features: Sex_male, Fare, Sex_female, SibSp, Pclass, Age, Parch, Embarked_C, Embarked_S, Embarked_Q

LogisticRegression | Accuracy: 0.7948

	precision	recall	f1-score	support	0	0.82	0.85	
0.84	165	1	0.74	0.71	0.73	103	accuracy	
0.79	268	macro avg	0.78	0.78	0.78	268	weighted avg	0.79
0.79	0.79	268						

RandomForest | Accuracy: 0.8060

	precision	recall	f1-score	support	0	0.81	0.90	
0.85	165	1	0.80	0.66	0.72	103	accuracy	
0.81	268	macro avg	0.80	0.78	0.79	268	weighted avg	0.81
0.81	0.80	268						