Open in Colab

In [3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [4]:
```
dataset = pd.read_csv('/content/tested.csv')
```

In [5]:
```
dataset.shape
```

Out[5]: (418, 12)

In [6]:
```
dataset.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 [7]:
```
dataset.isnull().sum()
```

Out[7]:
```
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 [8]:
```
df= dataset.dropna(axis=1, thresh=dataset.shape[0]*0.5)
```

In [9]:
```
df.shape
```

Out[9]: (418, 11)

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
PassengerId     0
Survived        0
Pclass          0
```

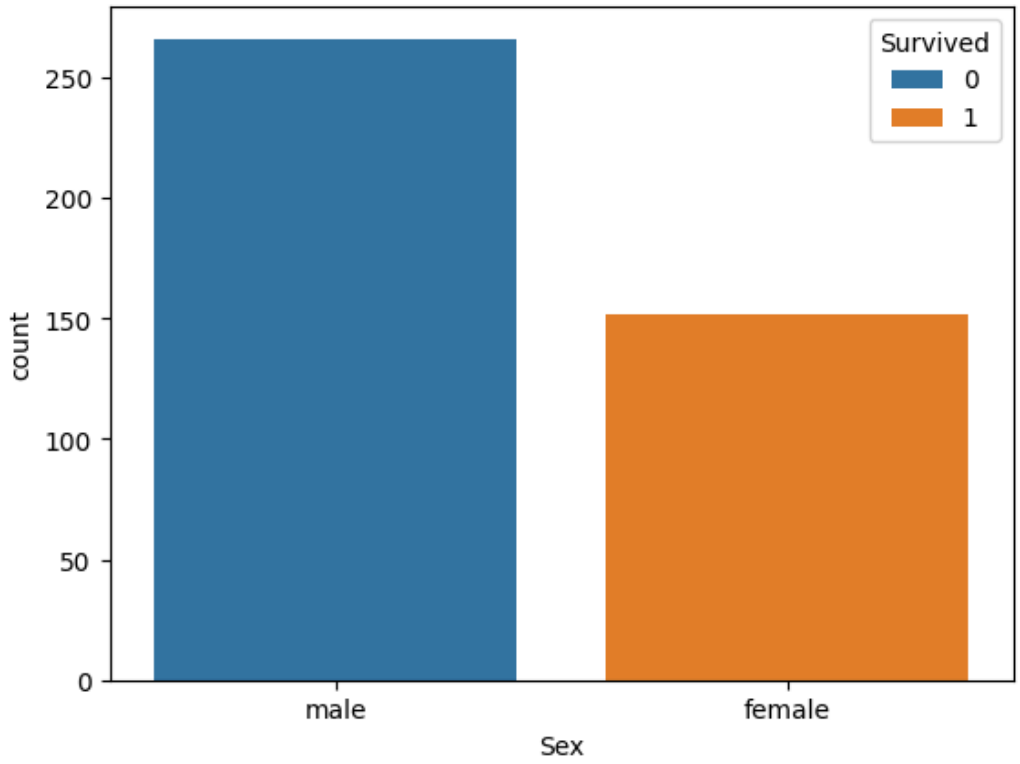
```
Name            0
Sex             0
Age            86
SibSp           0
Parch           0
Ticket          0
Fare            1
Embarked        0
dtype: int64
```

In [11]:
```
dataset.groupby("Survived").Survived.count()
```

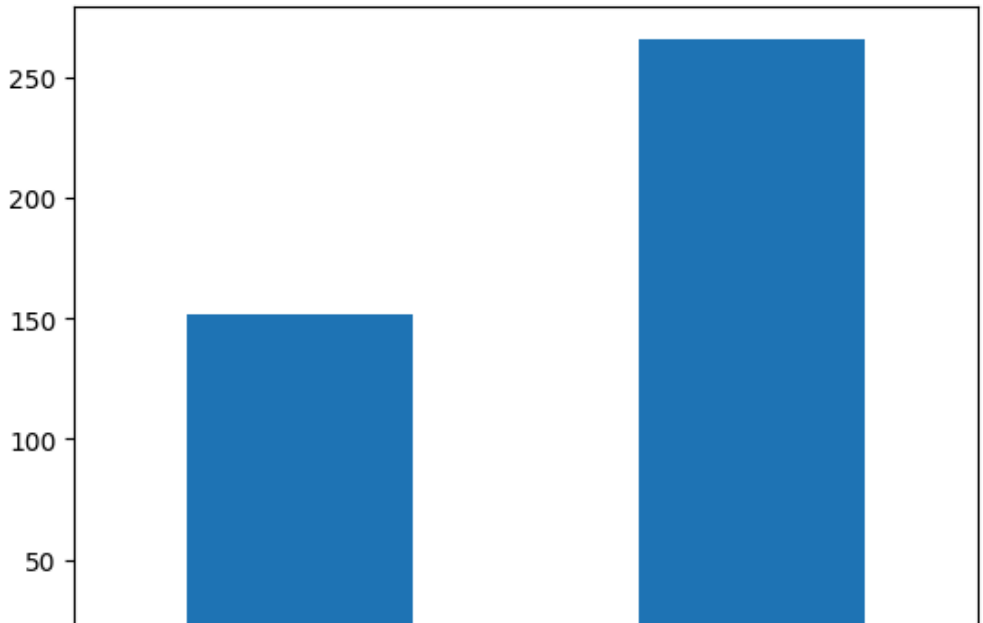
Out[11]:
```
Survived
0    266
1    152
Name: Survived, dtype: int64
```

In [12]:
```
sns.countplot(x="Sex", hue="Survived", data=df)
```

Out[12]:
```
<Axes: xlabel='Sex', ylabel='count'>
```



In [13]:
```
df.groupby("Sex").Sex.count().plot.bar()
plt.show()
print(df.groupby("Sex").Sex.count())
```




```
Sex
female    152
male      266
Name: Sex, dtype: int64
```

In [14]:
```
numeric_columns = df.select_dtypes(include=['number']).columns.to_list()
numeric_columns
```

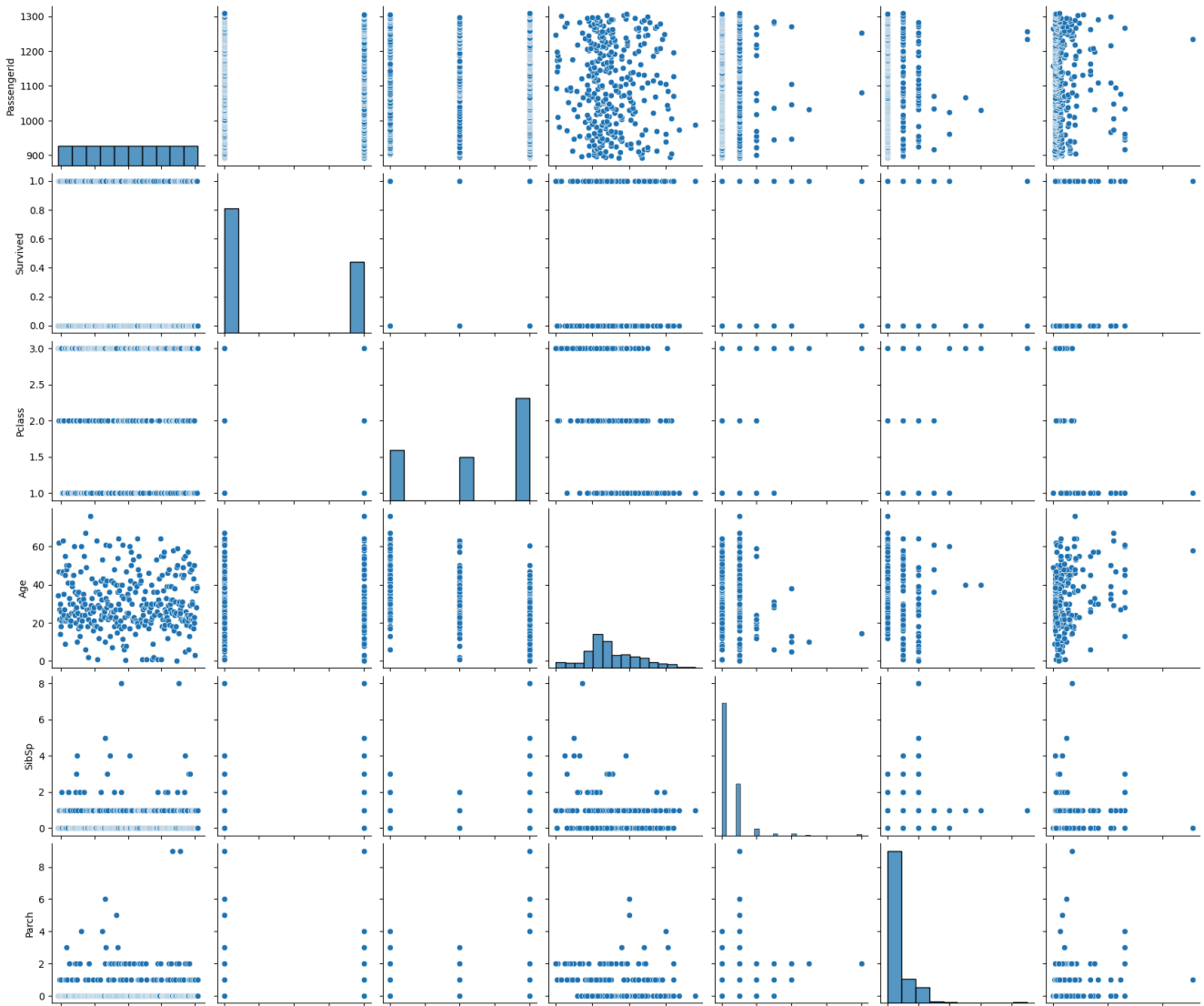
Out[14]:
```
['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
```

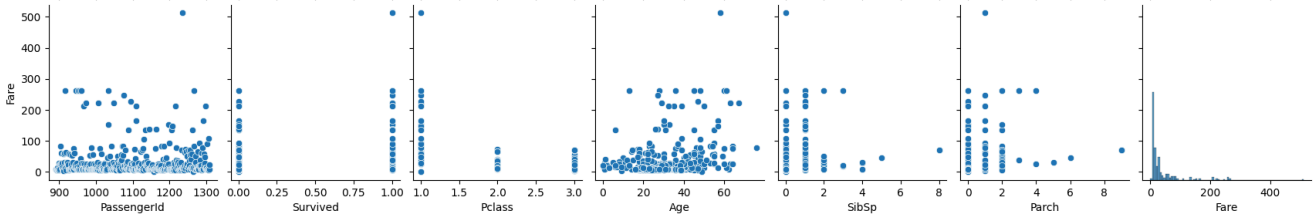
In [15]:
```
ndf= df[numeric_columns]
ndf.head()
```

Out[15]:

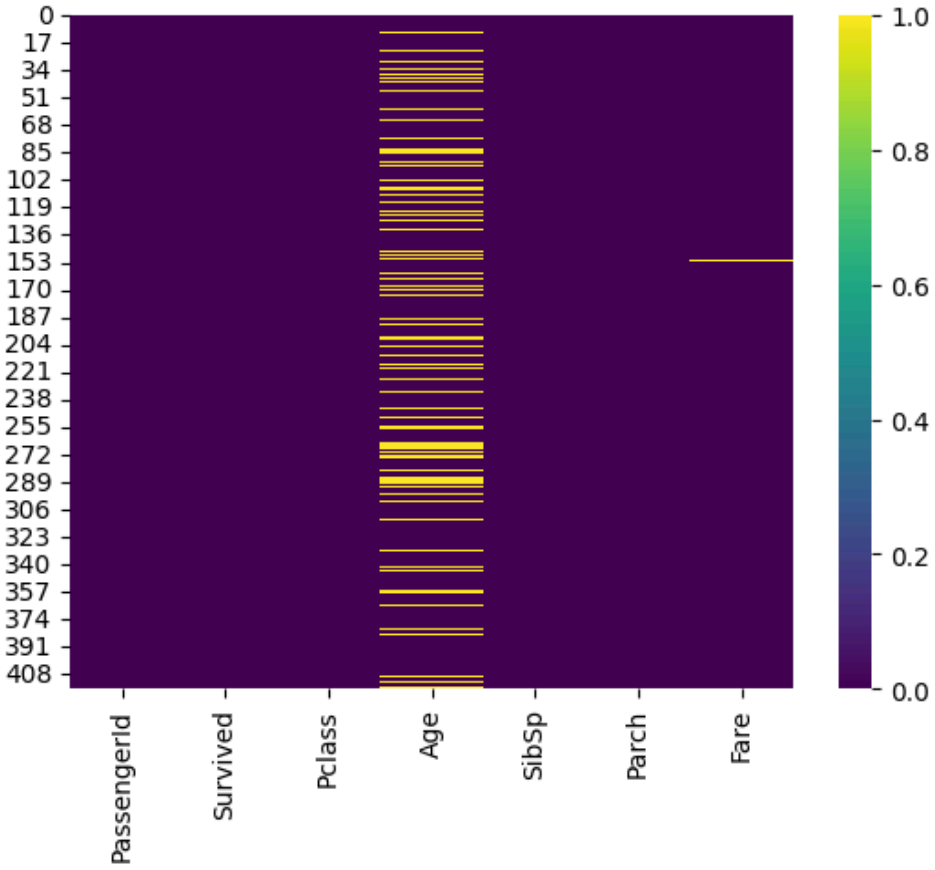
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	892	0	3	34.5	0	0	7.8292
1	893	1	3	47.0	1	0	7.0000
2	894	0	2	62.0	0	0	9.6875
3	895	0	3	27.0	0	0	8.6625
4	896	1	3	22.0	1	1	12.2875

In [16]:
```
sns.pairplot(ndf)
plt.show()
```





In [17]:
```
sns.heatmap(ndf.isnull(), cmap='viridis')
plt.show()
```



In [18]:
```
feature_columns = ndf.columns[ndf.notnull().all()].to_list()
target_columns = ndf.columns[ndf.isnull().any()].to_list()
```

In [19]:
```
feature_columns, target_columns
```

Out[19]: (['PassengerId', 'Survived', 'Pclass', 'SibSp', 'Parch'], ['Age', 'Fare'])

In [20]:
```
def predict_missing_values(new_df, target_column):
    df_combined = pd.concat([new_df, ndf[target_column]], axis=1)
    X_train = df_combined.dropna()[feature_columns]
    y_train = df_combined.dropna()[target_column]
    X_to_predict = df_combined[df_combined.isnull().any(axis=1)][feature_columns]
    model = LinearRegression()
    model.fit(X_train, y_train)
    df_combined.loc[df_combined[target_column].isnull(), target_column] = model.predict(X_to_predic
    return df_combined
```

In [21]:
```
ndf[pd.isnull(ndf["Age"])].head()
```

Out[21]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
10	902	0	3	NaN	0	0	7.8958
22	914	1	1	NaN	0	0	31.6833
29	921	0	3	NaN	2	0	21.6792
33	925	1	3	NaN	1	2	23.4500

36	928	1	3	NaN	0	0	8.0500

In [22]:
```
ndf[pd.isnull(ndf["Fare"])]
```

Out[22]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
152	1044	0	3	60.5	0	0	NaN

In [23]:
```
new_df = ndf[feature_columns]
for target_column in target_columns:
    new_df = predict_missing_values(new_df, target_column)
```

In [24]:
```
Predicted_Age_of_10th_row = new_df.loc[10, 'Age']
Predicted_Age_of_10th_row
```

Out[24]: 25.965440616916858

In [25]:
```
Predicted_fare = new_df['Fare'][152]
Predicted_fare
```

Out[25]: -3.0227562479530974

In [26]:
```
new_df[pd.isnull(new_df["Age"])].head()
```

Out[26]:

	PassengerId	Survived	Pclass	SibSp	Parch	Age	Fare

In [27]:
```
new_df[pd.isnull(new_df["Fare"])].head()
```

Out[27]:

	PassengerId	Survived	Pclass	SibSp	Parch	Age	Fare

In [28]:
```
sns.heatmap(new_df.isnull(), cmap='viridis')
plt.show()
```