Data Cleaning and preprocessing of Titanic Dataset using Python

titanic_preprocessing.py > ...

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler

# Load dataset
df = pd.read_csv("titanic.csv")  # Make sure titanic.csv is in the same folder

# Show basic info
print(df.head())
print(df.info())

# Handle missing values
df['Age'].fillna(df['Age'].median(), inplace=True)
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
df.drop('Cabin', axis=1, inplace=True)

# Encode categorical variables
df = pd.get_dummies(df, columns=['Sex', 'Embarked'], drop_first=True)

# Normalize numerical columns
scaler = StandardScaler()
df[['Age', 'Fare']] = scaler.fit_transform(df[['Age', 'Fare']])

# Boxplot for outliers
sns.boxplot(x=df['Fare'])
plt.title("Fare Outliers")
plt.show()

# Remove Fare outliers using IQR
Q1 = df['Fare'].quantile(0.25)
Q3 = df['Fare'].quantile(0.75)
IQR = Q3 - Q1
df = df[~((df['Fare'] < (Q1 - 1.5 * IQR)) | (df['Fare'] > (Q3 + 1.5 * IQR)))]

# Save cleaned data
df.to_csv("titanic_cleaned.csv", index=False)
print("Data preprocessing complete.")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\titanic_preprocessing _task> python -u "d:\titanic_preprocessing _task\titanic_preprocessing.py"
   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
<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
None
d:\titanic_preprocessing _task\titanic_preprocessing.py:15: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inpl
ace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operat
ion inplace on the original object.


  df['Age'].fillna(df['Age'].median(), inplace=True)
d:\titanic_preprocessing _task\titanic_preprocessing.py:16: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inpl
ace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operat
```


```
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
None
d:\titanic_preprocessing _task\titanic_preprocessing.py:15: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inpl
ace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operat
ion inplace on the original object.


  df['Age'].fillna(df['Age'].median(), inplace=True)
d:\titanic_preprocessing _task\titanic_preprocessing.py:16: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inpl
ace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operat
ion inplace on the original object.


  df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
Data preprocessing complete.
PS D:\titanic_preprocessing _task>
```