Week 1: Introduction to Machine Learning

Task 1.2: Data Cleaning and Preparation

1. Load and Inspect the Dataset:

Loading the Dataset:

First, we load the Titanic dataset using the seaborn library. The dataset is stored in a pandas DataFrame for further processing.

```
import pandas as pd
import seaborn as sns


data = sns.load_dataset('titanic')
data.head()
```

Inspecting the Dataset:

We inspect the dataset using the head() and describe() methods to get a preliminary understanding of the data.

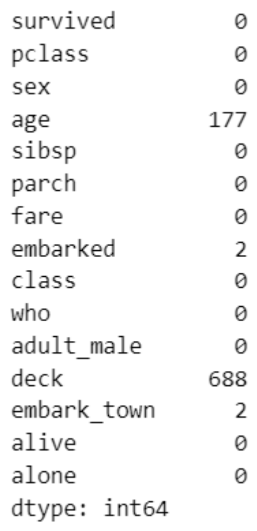
```
data.describe()
```

Checking for Missing Values:

We check for missing values in the dataset. The isnull() function helps identify missing values, and the sum() function provides a count of these values.

```
data.isnull().sum()
```

Output:



```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

From the output, we observe that the 'age', 'embarked', 'embark_town', and 'deck' columns contain missing values.

2. Data Cleaning:

Handling Missing Values:

1. **Drop the 'deck' Column:** The 'deck' column has 688 missing values out of 891, making it better to drop this column.

```
data.drop('deck', axis=1, inplace=True)
```

2. **Impute 'age' Column:** We fill the missing values in the 'age' column with the mean value of the column.

```
data['age'].fillna(data['age'].mean(), inplace=True)
```

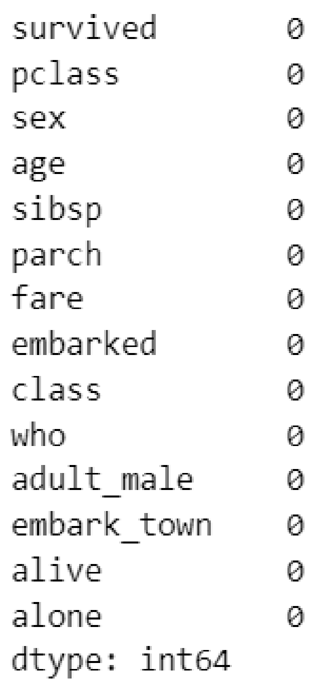
3. **Impute 'embarked' and 'embark_town' Columns:** We fill the missing values in the 'embarked' and 'embark_town' columns with their respective modes (most frequent values).

```
data['embarked'].fillna(data['embarked'].mode()[0],
                        inplace=True)
data['embark_town'].fillna(data['embark_town'].mode()[0],
                        inplace=True)
```

4. **Verify Missing Values:** After imputing, we check again for any remaining missing values.

```
data.isnull().sum()
```

Output:



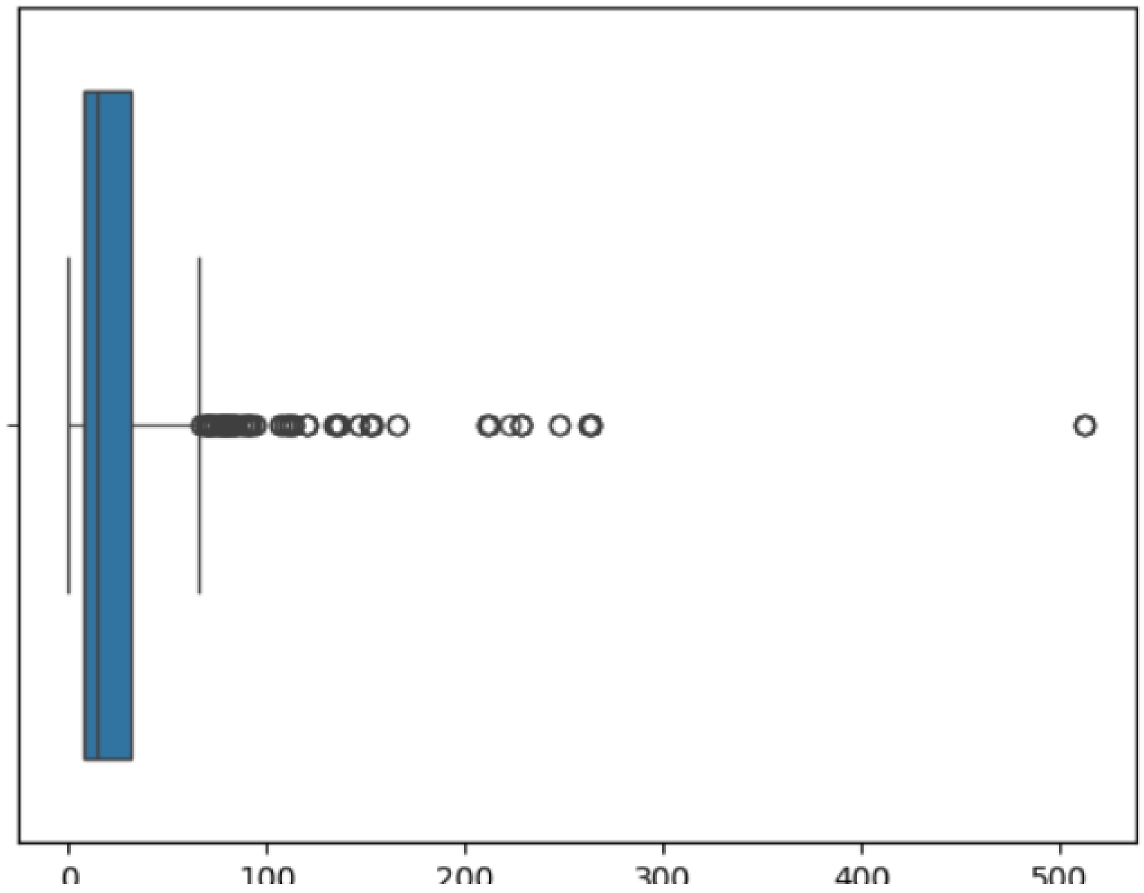
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       0
class          0
who            0
adult_male     0
embark_town    0
alive          0
alone          0
dtype: int64
```

All missing values have been handled successfully.

Handling Outliers:

We identify and handle outliers in the 'fare' column using the Interquartile Range (IQR) method. We also visualize the outliers by help of box plot.

```
sns.boxplot(data, x='fare')
```



As we can see there are outliers present. To fix that :

1. **Calculate IQR:**

```
import numpy as np
Q1 = data['fare'].quantile(0.25)
Q3 = data['fare'].quantile(0.75)
IQR = Q3 - Q1
ub = Q3 + (1.5 * IQR)
lb = Q1 - (1.5 * IQR)
```

2. **Cap the Outliers:** We cap the values below the lower bound to the lower bound and values above the upper bound to the upper bound.

```
data['fare'] = np.where(data['fare'] < lb, lb, data['fare'])
data['fare'] = np.where(data['fare'] > ub, ub, data['fare'])
```

3. Data Transformation:

Converting Categorical Data:

We convert categorical variables into numeric format using One-Hot Encoding.

```
from sklearn.preprocessing import OneHotEncoder, LabelEncoder,
StandardScaler


titanic = pd.get_dummies(data, columns=['sex', 'embarked',
'class', 'who', 'adult_male', 'embark_town', 'alive', 'alone'],
drop_first=True)
```

Standardizing Numerical Values:

We standardize the numerical values for 'age' and 'fare' using the StandardScaler.

```
scaler = StandardScaler()

titanic[['age', 'fare']] = scaler.fit_transform(titanic[['age',
'fare']])
```

Inspect the Cleaned and Transformed Dataset:

Finally, we inspect the first few rows of the cleaned and transformed dataset to verify the changes.

```
print(titanic.head())
```

Saving the Cleaned Dataset:

We save the cleaned and transformed dataset to a new CSV file.

```
titanic.to_excel('cleaned_titanic.xlsx', index=False)
```

Outcome: The dataset is now cleaned and transformed, ready for further analysis or modeling. The detailed steps and justifications ensure transparency and reproducibility of the preprocessing phase.