Task 1: Data Preprocessing

Problem Statement

Prepare a dataset for analysis by cleaning and preprocessing it.

Steps Completed

1. **Dataset Selection**: The Titanic dataset (`titanic.csv`) was selected for this task. This dataset is commonly used for classification problems and requires significant preprocessing.
2. **Data Loading**: The dataset was loaded using Pandas.
3. **Handling Missing Values**:
    - Missing values in the `Age` column were filled with the median age.
    - The `Cabin` column was dropped due to a high number of missing values and its limited relevance for this specific task.
4. **Feature Engineering/Transformation**: The `Sex` column (categorical) was converted into a numerical format using one-hot encoding (`Sex_male`).
5. **Data Normalization**: The numerical features `Age` and `Fare` were normalized using `MinMaxScaler` to scale them to a range between 0 and 1. This is crucial for many machine learning algorithms.
6. **Saving Cleaned Dataset**: The preprocessed data was saved to a new CSV file named `cleaned_titanic.csv` .

Code Implementation

The preprocessing steps are implemented in the `preprocess_titanic.py` script.

```
 import pandas as pd
from sklearn.preprocessing import MinMaxScaler

def preprocess_data(input_filepath, output_filepath):
    df = pd.read_csv(input_filepath)

    df["Age"].fillna(df["Age"].median(), inplace=True)
    if "Cabin" in df.columns:
        df.drop("Cabin", axis=1, inplace=True)

    # The original dataset from Stanford did not have an
```

```
'Embarked' column, so it was removed from the script.
    # If using a different Titanic dataset with 'Embarked',
uncomment the line below:
    # df["Embarked"].fillna(df["Embarked"].mode()[0],
inplace=True)

    df = pd.get_dummies(df, columns=["Sex"], drop_first=True)

    scaler = MinMaxScaler()
    df[["Age", "Fare"]] = scaler.fit_transform(df[["Age",
"Fare"]])

    df.to_csv(output_filepath, index=False)
    print(f"Cleaned data saved to {output_filepath}")

if __name__ == "__main__":
    input_file = "../titanic.csv"
    output_file = "cleaned_titanic.csv"
    preprocess_data(input_file, output_file)
```

Output

Upon execution, the script generates a `cleaned_titanic.csv` file in the `task1_data_preprocessing` directory. A confirmation message is printed to the console:

```
Cleaned data saved to cleaned_titanic.csv
```

This cleaned dataset is now ready for further analysis or model building.