```
In [1]: import pandas as pd

        # Create a sample DataFrame
        data = {'numeric_feature': [1, 2, None, 4, 5, 6, 7, None, 9, 10],
                'categorical_feature': ['A', 'B', 'A', 'C', 'B', 'A', 'C', 'B', 'A', 'C'
                'another_numeric': [100, 200, 150, 300, 250, 100, 350, 200, 150, 300],
                'target': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1]}

        sample_df = pd.DataFrame(data)

        # Save the DataFrame to a CSV file in the Colab environment
        sample_df.to_csv('sample_data.csv', index=False)

        print("Sample data created and saved as 'sample_data.csv'")
        print("\nSample DataFrame:")
        print(sample_df.head())
```

```
Sample data created and saved as 'sample_data.csv'

Sample DataFrame:
   numeric_feature categorical_feature  another_numeric  target
0              1.0                   A              100       0
1              2.0                   B              200       1
2              NaN                   A              150       0
3              4.0                   C              300       1
4              5.0                   B              250       0
```

```
In [2]: df = pd.read_csv('sample_data.csv')
```

```
In [4]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler, OneHotEncoder
        from sklearn.compose import ColumnTransformer
        from sklearn.pipeline import Pipeline
        from sklearn.impute import SimpleImputer

        # 1. Load the Data
        # Replace 'your_data.csv' with the actual path to your data file
        try:
            df = pd.read_csv('your_data.csv')
        except FileNotFoundError:
            print("Please upload your data file 'your_data.csv'.")
            # Create a dummy DataFrame for demonstration if the file is not found
            data = {'numeric_feature': [1, 2, None, 4, 5],
                    'categorical_feature': ['A', 'B', 'A', 'C', 'B'],
                    'target': [0, 1, 0, 1, 0]}
            df = pd.DataFrame(data)
            print("Using a dummy dataset for demonstration.")


        # Define features and target (adjust based on your dataset)
        features = ['numeric_feature', 'categorical_feature']
        target = 'target'

        X = df[features]
        y = df[target]

        # Identify numerical and categorical features
```

```python
numerical_features = X.select_dtypes(include=['int64', 'float64']).columns
categorical_features = X.select_dtypes(include=['object', 'category']).columns

# 2. and 3. Preprocessing and Transformation Pipeline
# Create transformers for numerical and categorical features
numerical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')), # Handle missing numerical da
    ('scaler', StandardScaler()) # Scale numerical features
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')), # Handle missing categ
    ('onehot', OneHotEncoder(handle_unknown='ignore')) # One-hot encode categori
])

# Combine transformers using ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numerical_transformer, numerical_features),
        ('cat', categorical_transformer, categorical_features)
    ])

# Create the full pipeline
pipeline = Pipeline(steps=[('preprocessor', preprocessor)])

# Fit and transform the data
X_processed = pipeline.fit_transform(X)

# Display the processed data (optional)
print("\nProcessed Data (first 5 rows):")
print(pd.DataFrame(X_processed).head())

# 4. Loading (Example: Prepare for further use or save)
# You can now use X_processed and y for machine learning or analysis.
# For example, split into training and testing sets:
X_train, X_test, y_train, y_test = train_test_split(X_processed, y, test_size=0.

print("\nData split into training and testing sets:")
print(f"Training features shape: {X_train.shape}")
print(f"Testing features shape: {X_test.shape}")
print(f"Training target shape: {y_train.shape}")
print(f"Testing target shape: {y_test.shape}")
```

```
Please upload your data file 'your_data.csv'.
Using a dummy dataset for demonstration.

Processed Data (first 5 rows):
          0    1    2    3
0 -1.414214  1.0  0.0  0.0
1 -0.707107  0.0  1.0  0.0
2  0.000000  1.0  0.0  0.0
3  0.707107  0.0  0.0  1.0
4  1.414214  0.0  1.0  0.0

Data split into training and testing sets:
Training features shape: (4, 4)
Testing features shape: (1, 4)
Training target shape: (4,)
Testing target shape: (1,)
```