Step-by-Step Preprocessing Pipeline for Machine Learning

1. Load the Data

- Use pandas to load CSV or Excel files.

  Example:

```
  df = pd.read_csv("your_dataset.csv")
```

2. Understand the Data

- Check dimensions, column types, head/tail.

```
  df.info()
  df.describe()
  df.head()
  df.tail()
```

- Check for nulls:

```
  df.isnull().sum()
```

3. Handle Missing Values

- Numerical: Impute with mean/median

```
  df['col'] = df['col'].fillna(df['col'].mean())
```

- Categorical: Impute with mode or 'Unknown'

```
  df['col'] = df['col'].fillna(df['col'].mode()[0])
```

- Drop columns/rows if necessary:

```
  df = df.dropna(axis=1)  # drop columns
  df = df.dropna(axis=0)  # drop rows
```

4. Fix Incorrect/Outlier Values

- Use visualization like seaborn's boxplot to find outliers.

```
  sns.boxplot(df['col'])
```

- Replace or drop outliers.

```
  df.loc[df['col'] > threshold, 'col'] = df['col'].median()
```

5. Convert Categorical to Numerical

- Label Encoding:

```
  from sklearn.preprocessing import LabelEncoder
  le = LabelEncoder()
  df['col'] = le.fit_transform(df['col'])
```

- One-Hot Encoding:

```
  df = pd.get_dummies(df, columns=['col'], drop_first=True)
```

6. Feature Engineering (Optional)

- Create new features.

```
  df['new_col'] = df['col1'] * df['col2']
```

- Extract time components.

```
  df['date'] = pd.to_datetime(df['date'])
  df['year'] = df['date'].dt.year
```

7. Feature Scaling

- StandardScaler:

```
  from sklearn.preprocessing import StandardScaler
  scaler = StandardScaler()
  df[['col1', 'col2']] = scaler.fit_transform(df[['col1', 'col2']])
```

- MinMaxScaler:

```
  from sklearn.preprocessing import MinMaxScaler
  scaler = MinMaxScaler()

  df[['col1', 'col2']] = scaler.fit_transform(df[['col1', 'col2']])
```

8. Remove Unnecessary Features

- Drop ID or unrelated fields.

```
  df = df.drop(['id', 'name', 'unrelated_col'], axis=1)
```

9. Check for Class Imbalance

- df['target'].value_counts()

- Apply techniques like SMOTE or class weight adjustment.

10. Split Dataset

- from sklearn.model_selection import train_test_split

```
  X = df.drop('target', axis=1)

  y = df['target']

  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Now You're Ready to Train!