Data cleaning is an essential step in preparing data for machine learning models. It involves identifying and handling missing data, removing duplicates, dealing with outliers, and transforming the data to a suitable format for analysis. Here's a step-by-step guide on how to perform data cleaning for an ML model using Python:

1. Import the necessary libraries:

```
import pandas as pd
import numpy as np
```

2. Load the dataset into a pandas DataFrame:

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

3. Handle missing data:

```
  **Identify missing values:**
```

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

4 Decide how to handle missing values based on the specific dataset and problem. Options include:

- Removing rows or columns with missing values:

```
df.dropna(axis=0)  # Remove rows with any missing value
df.dropna(axis=1)  # Remove columns with any missing value
```

5 Imputing missing values with mean, median, or mode:

```
Imputing missing values with mean, median, or mode:
```

6 Remove duplicates:

```
df.drop_duplicates(inplace=True)
```

7 Deal with outliers:

- Identify outliers using visualization techniques or statistical methods.
- Decide on an appropriate strategy based on the specific dataset and problem. Options include:
- Removing outliers:

```
# Assuming 'column_name' is the column containing outliers
df = df[(np.abs(df['column_name'] - df['column_name'].mean()) / df['column_name'].std()) < 3]
```

-Capping or flooring outliers to a specific threshold:

```
# Assuming 'column_name' is the column containing outliers
df['column_name'] = np.where(df['column_name'] > upper_threshold, upper_threshold, df['column_name'])
df['column_name'] = np.where(df['column_name'] < lower_threshold, lower_threshold, df['column_name'])
```

8 Transform the data:

- Convert categorical variables to numerical format using one-hot encoding or label encoding:

```
# One-hot encoding
df_encoded = pd.get_dummies(df, columns=['categorical_column'])

# Label encoding
```

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

-Standardize or normalize numerical features:

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Standardization
scaler = StandardScaler()
df['numerical_column'] = scaler.fit_transform(df['numerical_column'])

# Normalization
scaler = MinMaxScaler()
df['numerical_column'] = scaler.fit_transform(df['numerical_column'])
```

9 Save the cleaned dataset:

```
df.to_csv('cleaned_dataset.csv', index=False)
```