

Acquiring the dataset:

You can use various methods to acquire a dataset, such as downloading it from a public repository or using a dataset that comes with a library (e.g., scikit-learn datasets).

Cleaning the data by identifying and handling missing values:

Use pandas to load the dataset into a DataFrame.

Identify missing values using `isnull()` and handle them using methods like `dropna()`, `fillna()`, or imputation.

Transforming the data:

Perform any necessary transformations, such as feature engineering, creating new features, or handling outliers.

Reducing the data:

If needed, you can reduce dimensionality using techniques like Principal Component Analysis (PCA) or feature selection.

Assessing data quality:

Use descriptive statistics and visualization techniques to understand the distribution of your data.

Identify potential issues such as outliers or skewed distributions.

Encoding categorical data:

Convert categorical variables into numerical representations. You can use methods like one-hot encoding or label encoding.

Splitting the dataset:

Split your dataset into training and testing sets using tools like `train_test_split` from `scikit-learn`.

Scaling the features:

Standardize or normalize your numerical features to ensure they are on a similar scale. You can use tools like `StandardScaler` or `MinMaxScaler` from `scikit-learn`.

###Code :

```
[ ]: # Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer

df = pd.read_csv('Data.csv')

imputer = SimpleImputer(strategy='mean')
df['3'] = imputer.fit_transform(df[['3']])

df['4'] = df['3'].apply(lambda x: np.log(x + 1))

df_encoded = pd.get_dummies(df, columns=['3'])

X_train, X_test, y_train, y_test = train_test_split(df_encoded.drop('4', axis=1),
                                                    df_encoded['t4'],
                                                    test_size=0.2,
                                                    random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```