

```
import pandas as pd, numpy as np, matplotlib.pyplot
as plt, seaborn as sns
from sklearn.preprocessing import StandardScaler,
LabelEncoder

# Load dataset
df = pd.read_csv("your_dataset.csv") # Replace with
your dataset file

# Basic info
print(df.head(), "\n", df.info(), "\n", df.describe(),
"\nMissing values:\n", df.isnull().sum())

# Handle missing values
for col in df.columns:
    if df[col].dtype == 'object':
        df[col].fillna(df[col].mode()[0], inplace=True)
    else:
        df[col].fillna(df[col].mean(), inplace=True)

# Encode categorical features
le = LabelEncoder()
for col in df.select_dtypes(include=['object']).columns:
    df[col] = le.fit_transform(df[col])

# Standardize numerical features
scaler = StandardScaler()
num_cols =
df.select_dtypes(include=[np.number]).columns
df[num_cols] = scaler.fit_transform(df[num_cols])
```

```
df[num_cols].describe()

# Visualize & remove outliers
plt.figure(figsize=(10,6));
sns.boxplot(data=df[num_cols]); plt.title("Boxplot of Numerical Features"); plt.show()
Q1, Q3 = df[num_cols].quantile(0.25),
df[num_cols].quantile(0.75)
IQR = Q3 - Q1
df = df[~((df[num_cols] < (Q1 - 1.5*IQR)) |
(df[num_cols] > (Q3 + 1.5*IQR))).any(axis=1)]

print("Final cleaned dataset shape:", df.shape)
```

you want, I can also make a **single run**