

Hands-On Lab: Generative AI for Data Preparation

Estimated time needed: 30 minutes

Overview

In this lab, you will learn how to use generative AI to prepare data using the text, chatbox.

Objectives

After completing this lab, you will be able to:

- 1. Sign up on <https://www.chatterbox.ai>
- 2. Upload a dataset
- 3. Handle missing values
- 4. Perform the data standardization
- 5. Perform the data normalization

Prerequisites:

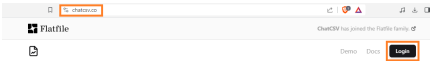
- A Chatbox account
- A basic understanding of EDA

Dataset

The dataset is a filtered and modified version of the [Large Price Prediction open specification dataset](#), available under the Database Contents License (DCL) v1.0 on the Kaggle website. While loading down the CSV or Comma-separated, click [here](#) to download the data set.

Task 1: Sign in on Chatterbox

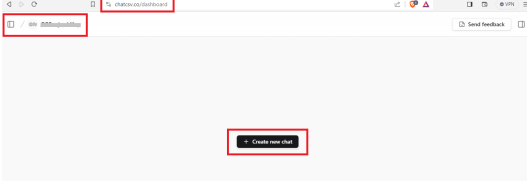
Step 1: If you do not have an account, click <https://www.chatterbox.ai> and then click **Sign** at the top right corner.



Step 2: You can login using gmail, Github, or your email ID. Click any one option and follow the steps to create your account on Chatterbox.



Step 3: After you create your account, login with your credentials and the chatterbox dashboard will be displayed.



Task 2: Upload Dataset

Step 1: On the dashboard screen, click **Create new Chat** to start preparing data.

Step 2: Click **Attach file** and attach the dataset `laptop_price_dataset_model.csv` from the location where you have downloaded the dataset from the link provided earlier

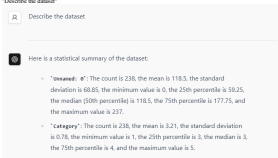


Step 3: Scroll down the dataset to view the details presented by the GPT.

The dataframe "lpt" has 238 entries and 13 columns. Here is a quick rundown of the columns:

- 1. "screen": "The column seems to be an index column with integer values."
- 2. "manufacturer": "This column contains object (string) values, likely the names of the laptop manufacturers."
- 3. "category": "This column contains integer values, likely representing different categories of laptops."
- 4. "screen": "This column contains object (string) values, likely representing the type of screen the laptop has."
- 5. "lpt": "This column contains integer values, likely representing different types of GPUs."

Step 4: Write a prompt to get the statistical description of the dataset.



Task 3: Handle missing values

Step 1: Write a prompt "Identify the attributes with missing data" and press Enter. The response will display the attributes with missing values in "Screen", "Size", and "Weight".

You need to replace the missing values with appropriate values. The following are the rules for this:

- Missing values in columns containing categorical values need to be replaced with the most frequent entries.
- Missing entries in columns with continuous data need to be replaced with the mean value of the columns. If a value is missing in the target column, you may need to drop that row. The prompt response will be something as shown below.



The attributes with missing data in the dataframe are:

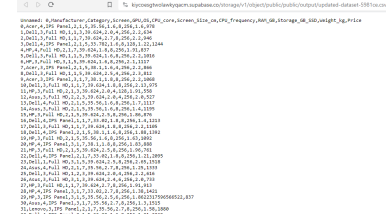
- 1. "Screen_Size_cm": This attribute has 4 missing values.
- 2. "Weight_kg": This attribute has 5 missing values.

Step 2: Write a prompt: "Replace the missing values with the mean value in the respective column and save the updated dataset."



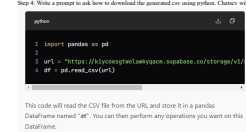
updated dataset: laptop.csv

If you open the generated link for the updated dataset, it will be shown as below:



Step 3: You can open the link and perform the following on the dataset:

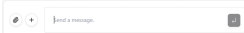
- Copy all the data and paste it into an Excel sheet.
- Click **Data**, then click **Text to Columns**, select **Comma** as the delimiter.
- Save the new dataset as a new file.



Task 4: Perform the data standardization

Note that the parameter "Screen_Size_cm" represents the screen size in centimeters instead of the standard unit, which is inches. Also, the parameter "Weight_kg" needs to be expressed in pounds.

Step 1: First, you need to update the newly updated dataset. Click the "+" icon and attach the new dataset you saved in the previous step.



Step 2: Write the following prompts in the chat box to change the attributes into standard units:

- 1. Units under "Screen_Size_cm" is assumed to be in centimeters. Convert this data into inches. Modify the name of the attribute to "Screen_Size_inch".
- 2. Units under "Weight_kg" is assumed to be in kilograms. Convert this data into pounds. Modify the name of the attribute to "Weight_pounds".

1. Data under "Screen_Size_cm" is assumed to be in centimeters. Convert this data into inches. Modify the name of the attribute to "Screen_Size_inch".

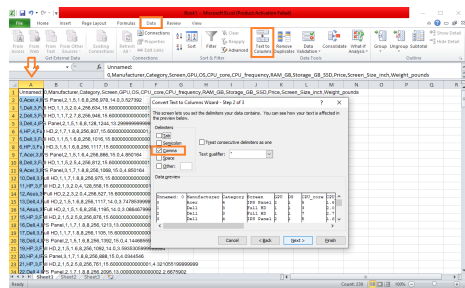
2. Data under "Weight_kg" is assumed to be in kilograms. Convert this data into pounds. Modify the name of the attribute to "Weight_pounds".

I have converted the "Screen_Size_cm" column to inches and renamed it to "Screen_Size_inch". Similarly, I have converted the "Weight_kg" column to pounds and renamed it to "Weight_pounds". The modified database has been saved as a CSV file.

You can download the CSV file from [here](#).

Step 3: You can open the link and perform the following on the dataset:

- Copy all the data and paste it into an Excel sheet.
- Click Data, then click Text to Columns, select Columns as the delimiter.
- Save the new dataset as a new file.



Task 5: Perform the data normalization

You may require normalizing the data under some attributes. Since there are many normalization forms, you need to mention the exact needs and tasks. Also, you can save the normalized data as a new attribute or change the original attribute. You need to provide clear details in the prompt. For example, assume that the data under "CPU_frequency" needs to be normalized as per the maximum value under the attribute. You want the changes to reflect directly under the attribute instead of creating a new attribute.

Step 1: Upload the new updated dataset from the previous step. Click the "+" icon and attach the new dataset.

Step 2: Write a prompt: "Normalize the content under the attribute 'CPU_frequency' concerning its maximum value, and do not create a new attribute."

1. Normalize the content under the attribute "CPU_frequency" concerning its maximum value and do not create a new attribute.

I have normalized the "CPU_frequency" column by dividing each value by the maximum value in the column. The updated values are now stored in the same column.

Let's save this updated database as a CSV file. The updated database with the normalized "CPU_frequency" column has been saved as a CSV file. You can access it using the following link:

[normalized_cpu_frequency](#)

Practice problems

- Create a prompt to generate a Python code that converts the values under Price from USD to Euros.
- Modify the normalization prompt to perform min-max normalization on the CPU_frequency parameter.

Conclusion

In this lab, you have learned to handle missing values in your dataset, and performed data standardization and data normalization.

Author(s)

Dr. Pooya

© IBM Corporation. All rights reserved.

