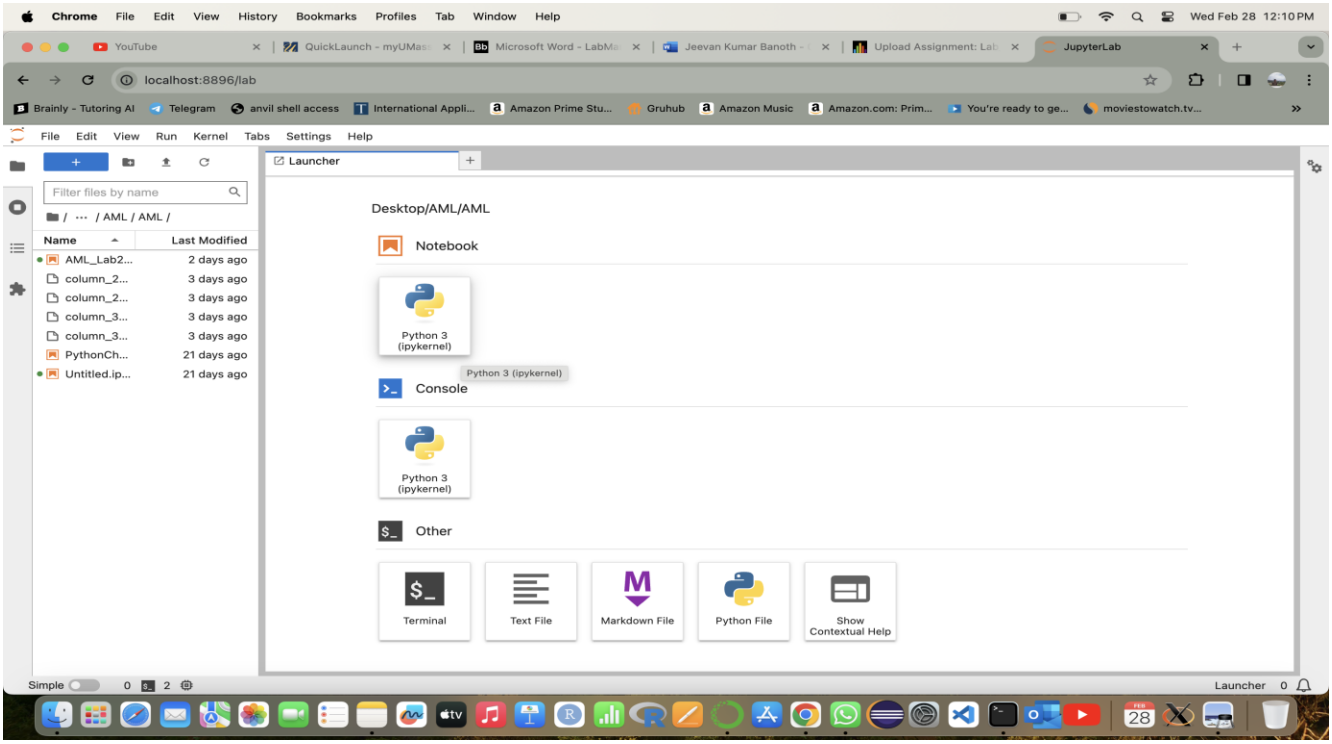
Jeevan Kumar Banoth - 02105145

Advanced machine learning

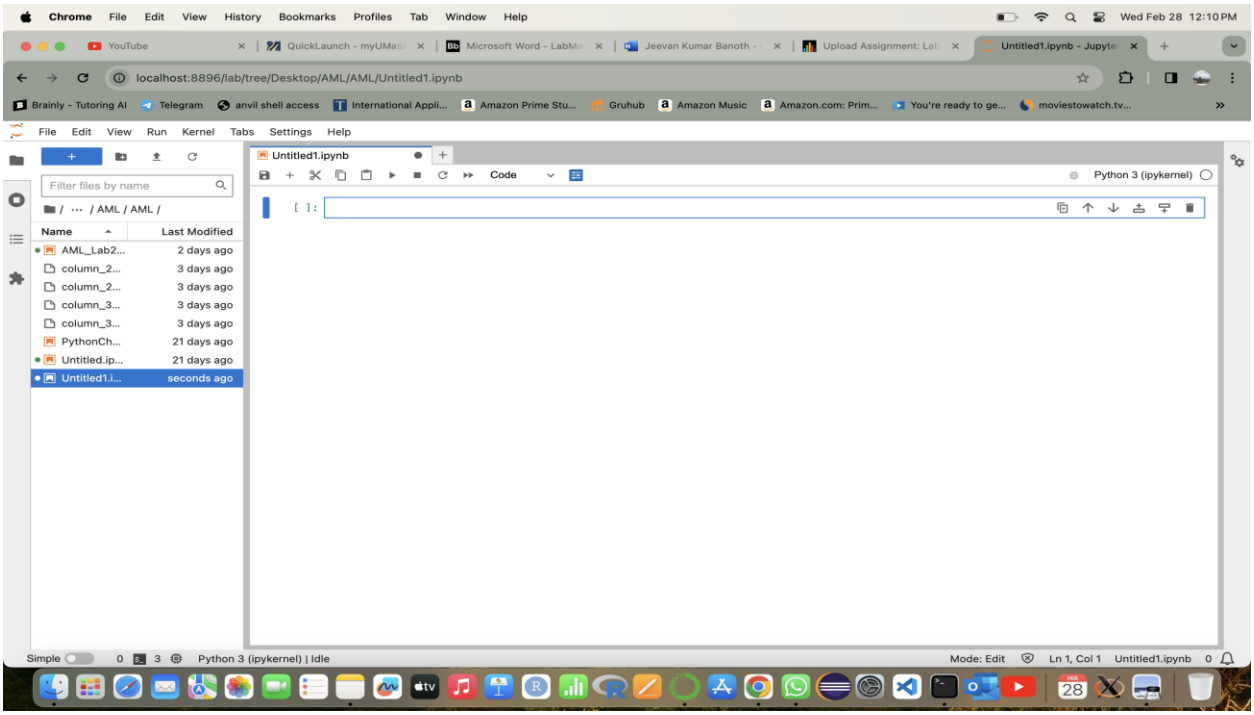
Homework –3:

Lab_3_Manual_Encoding Categorical Data

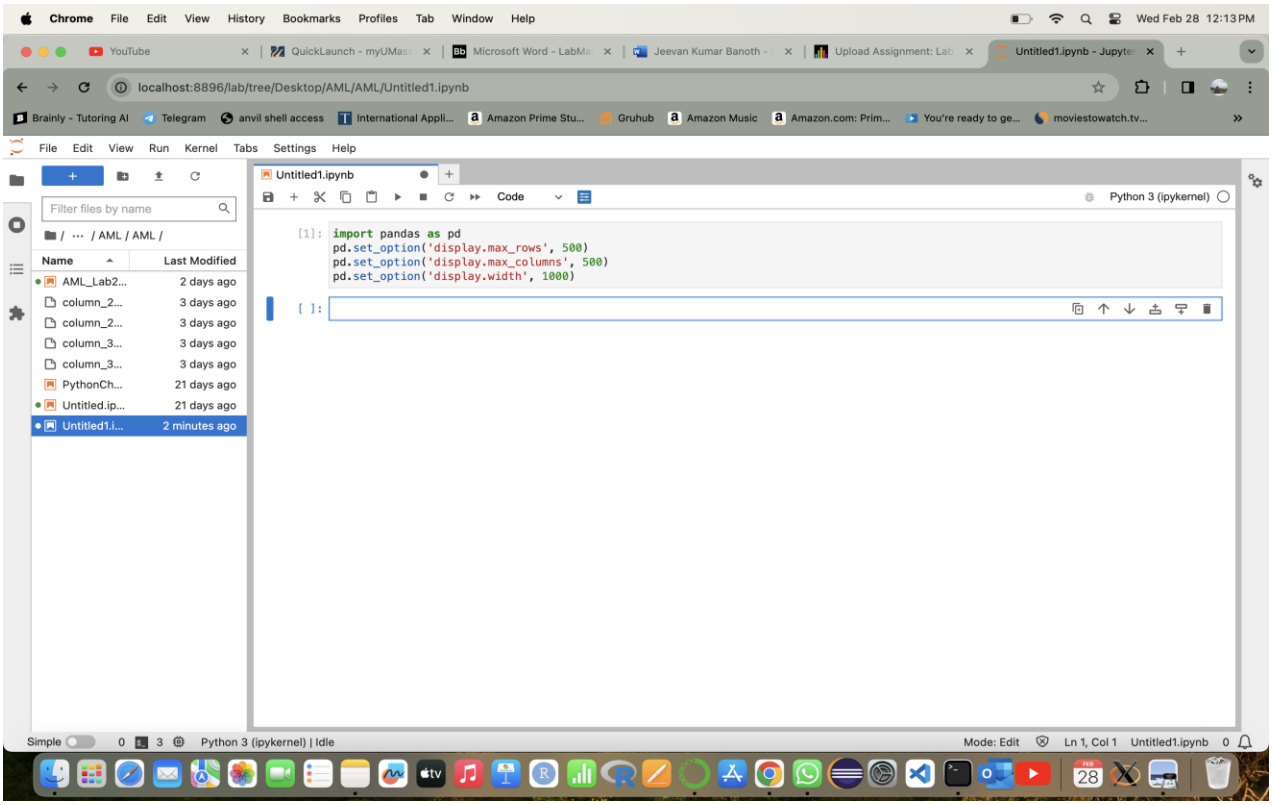
- First, we need to open the anaconda application and then open Jupyter lab, It looks like below.



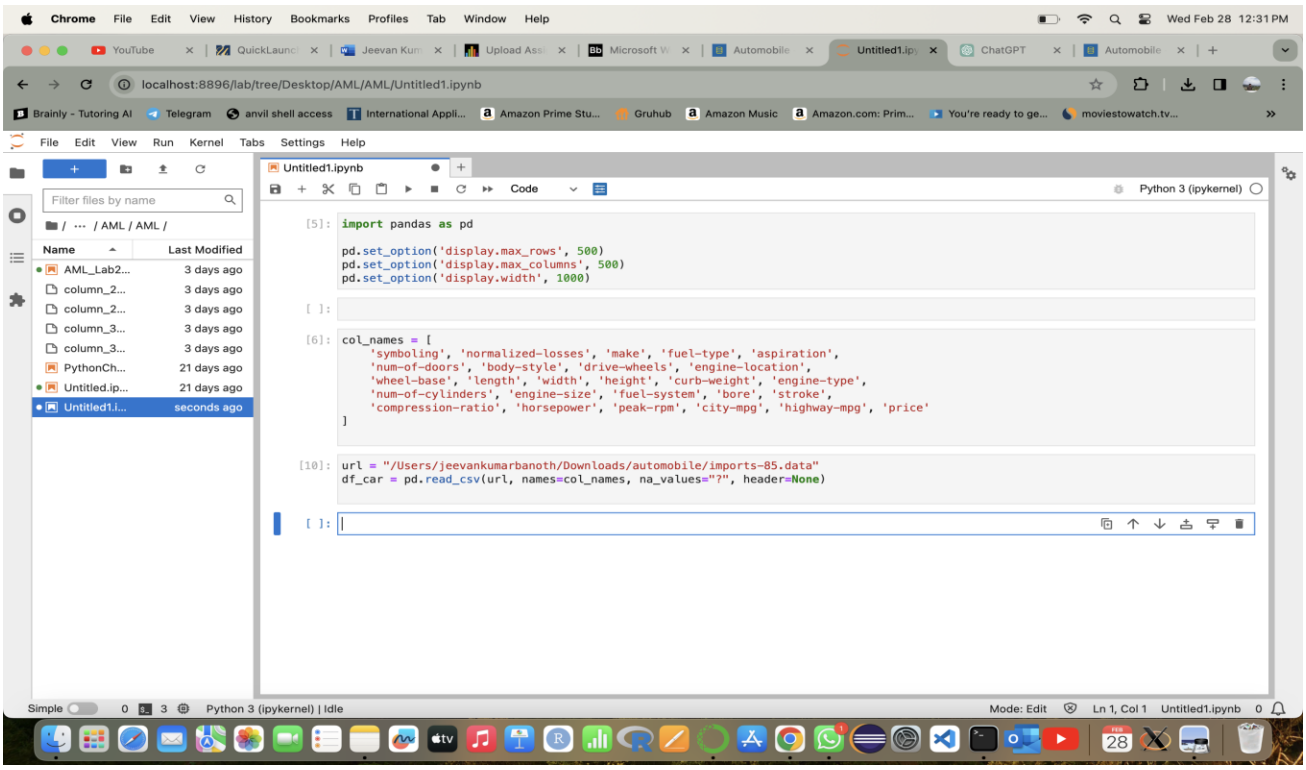
- Then click on the python 3 ipykernel in notebook, so it will redirect to chrome and will open a notebook as follows.



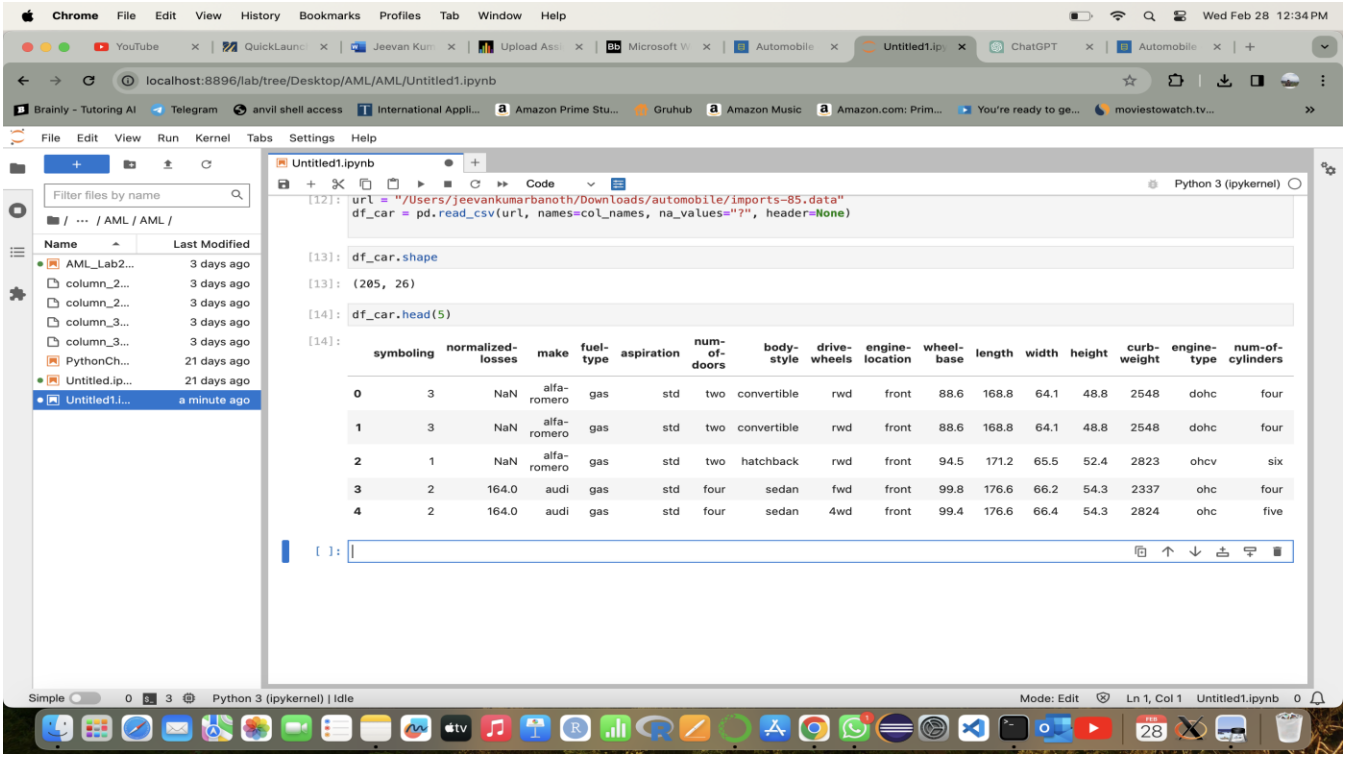
Step 1: Importing and exploring the data:



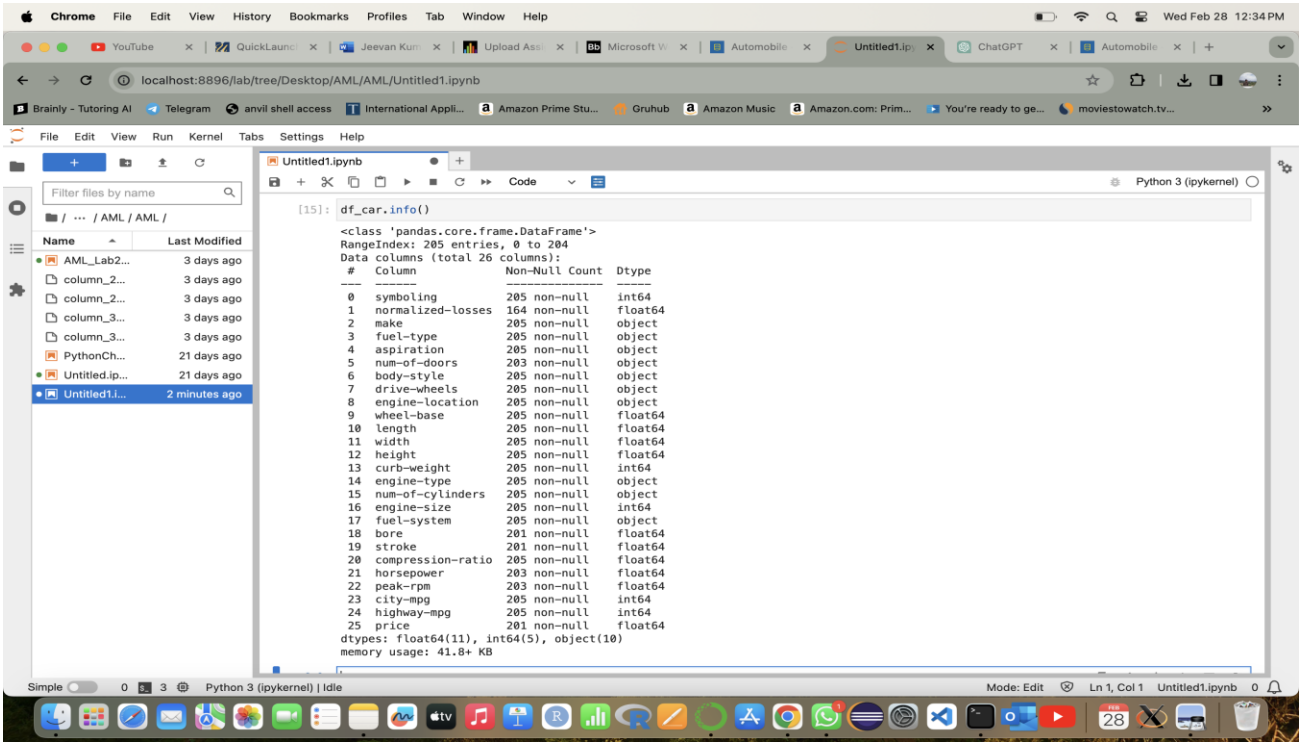
- By using the keyboard, we have written the code for importing the panda's package and setting up some default package options for displaying.
- Now we are loading the dataset into the Panda’s Data frame by assigning the dataset containing file location to it in the place of url.
- And we are defining the col_names that exist in the dataset by looking at attributes listed in the dataset description.



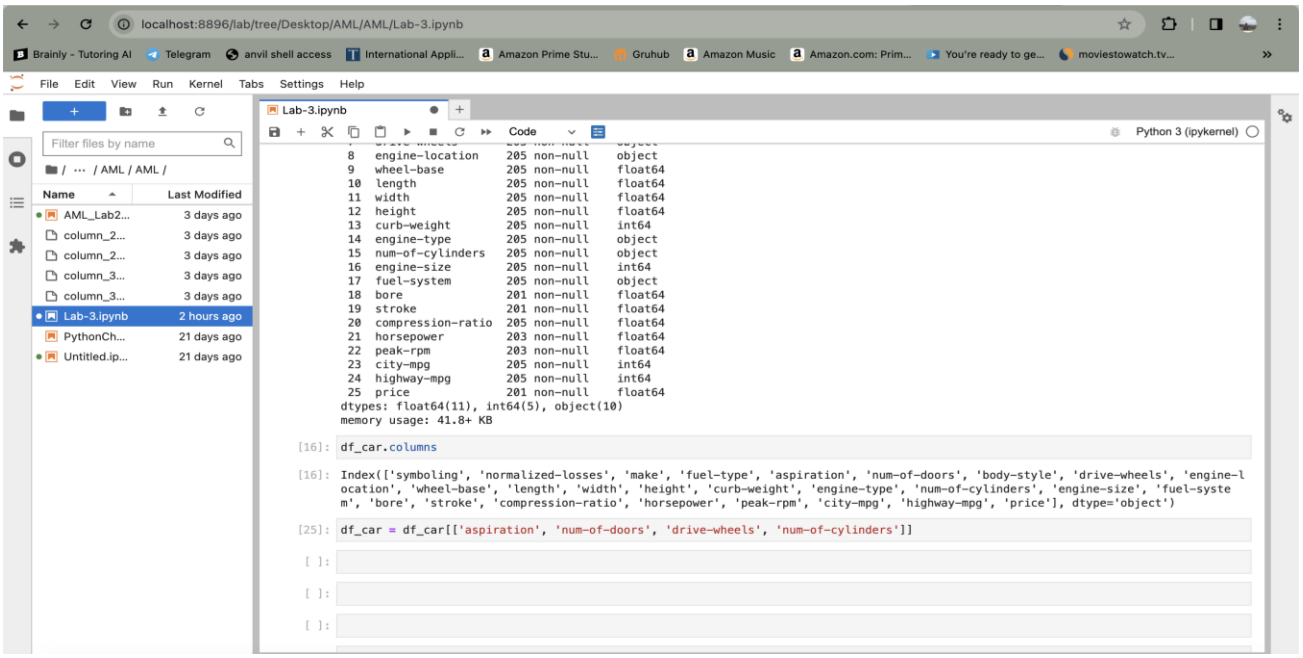
- Now, we will see the number of rows (instances) and columns (features) in the dataset by using the shape () command.
- Next, we are using the head command to examine the data.
- There are 5 columns (0-5), some of them contain numerical data, and some of them are text, Nan values too.
- Below is the image of using shape and head commands.



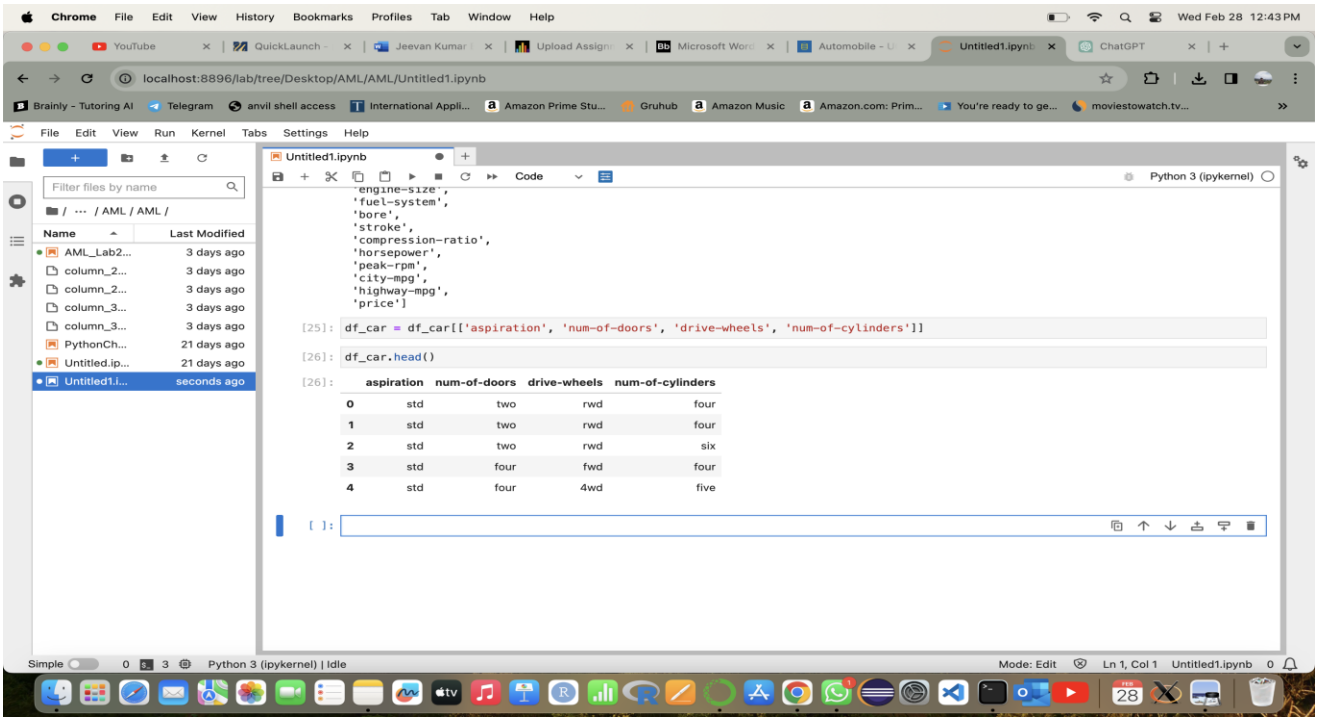
- To display the information about columns we are using info () command here.
- This command gives the information about column names, Dtypes, etc



- So, now we are dropping the columns that we won't use for coding to make the view of dataset simpler.

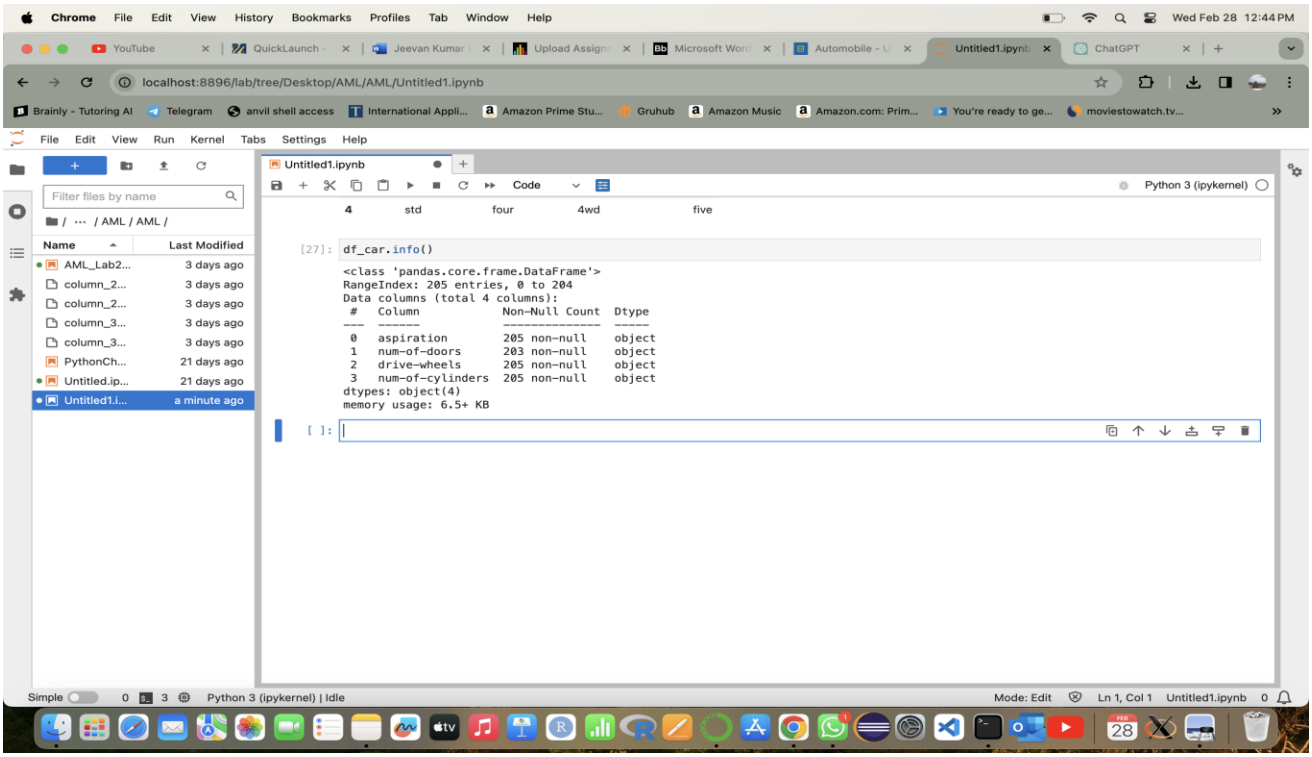


- Now we have four columns that all contain text values, we are using head () command to view those values.

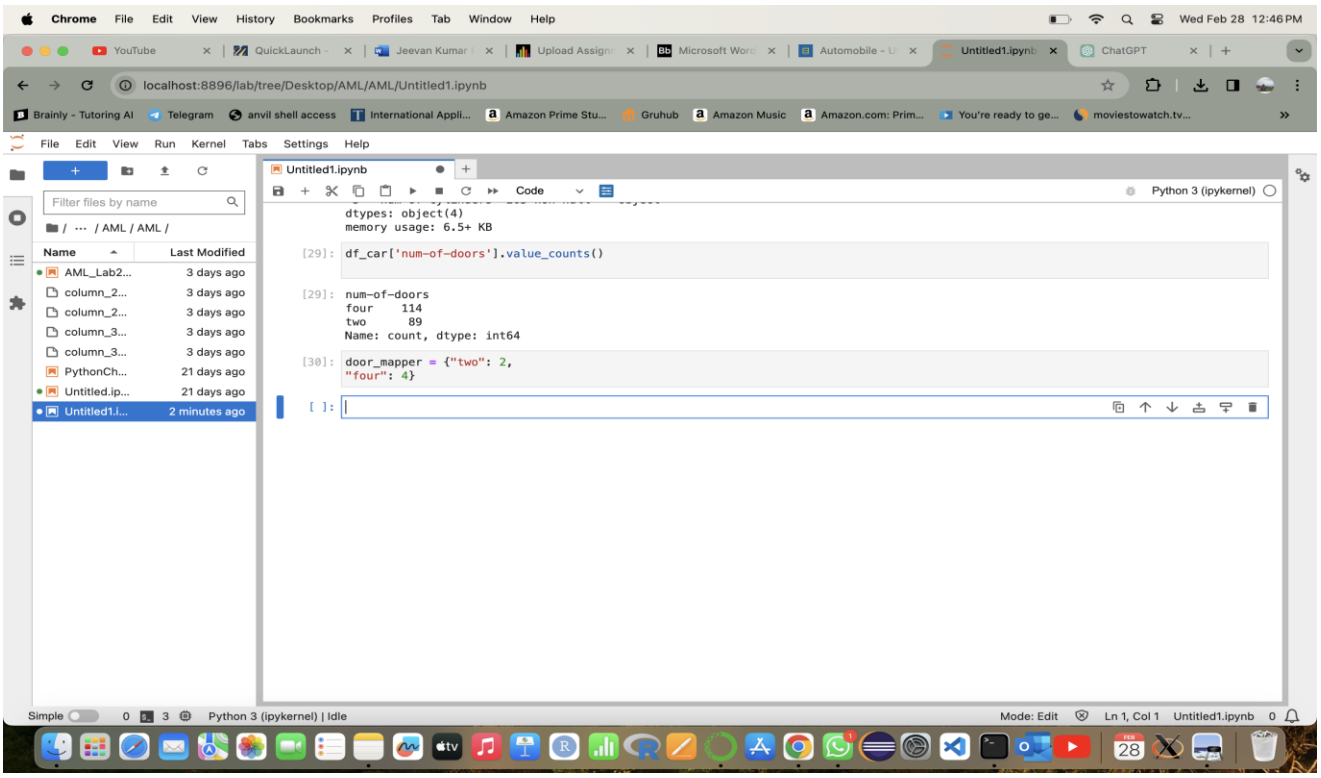


Step 2: Encoding ordinal features:

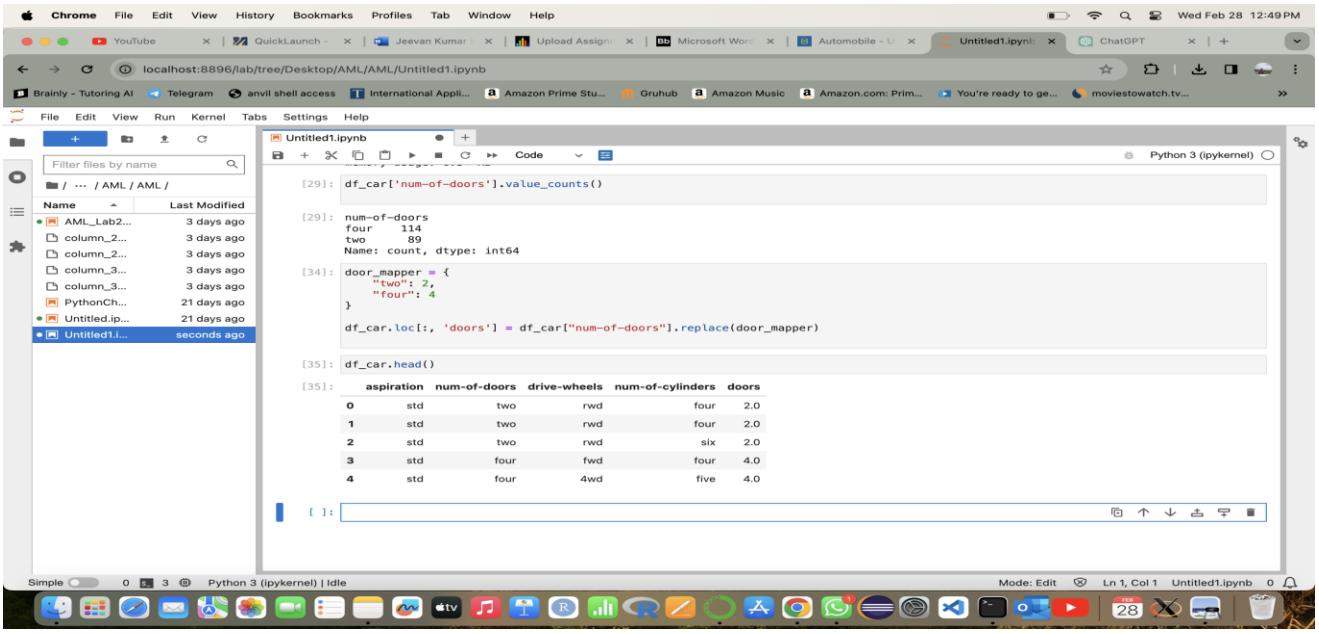
- Most of the ML algorithms requires the values which are numerical. But here, num-of-cylinders and num-of-doors features have an ordinal value so we can convert these features into numerical counterparts.
- Aspiration and drive-wheels don't have an ordinal value, so we need to convert them differently.
- Now, in this step we use mapper function to convert ordinal functions into ordered numerical values.



- First, we start by what values the ordinal function contain, starting with the num-of-doors feature, we can use **value_counts** to discover the values.
- The ordinal values of value_counts feature is shown below.

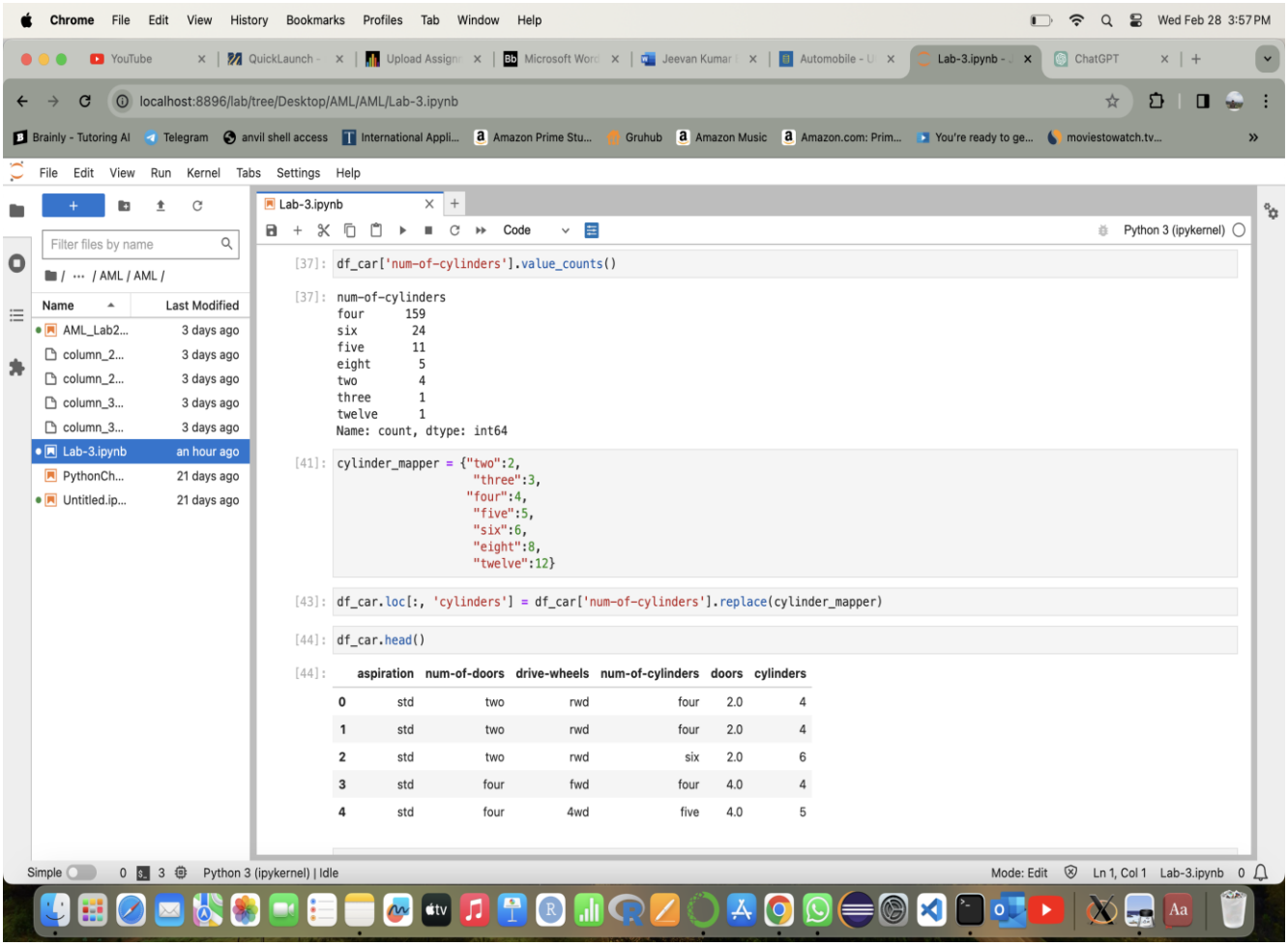


- It has two values, four and two, for this, we need to create a dictionary and map the variable names to their numerical values as above.
- We can use the replace method from pandas to generate new numerical column based on num-of-doors column.



- As shown in the above image, we are using the head function for generating detailed information about it.

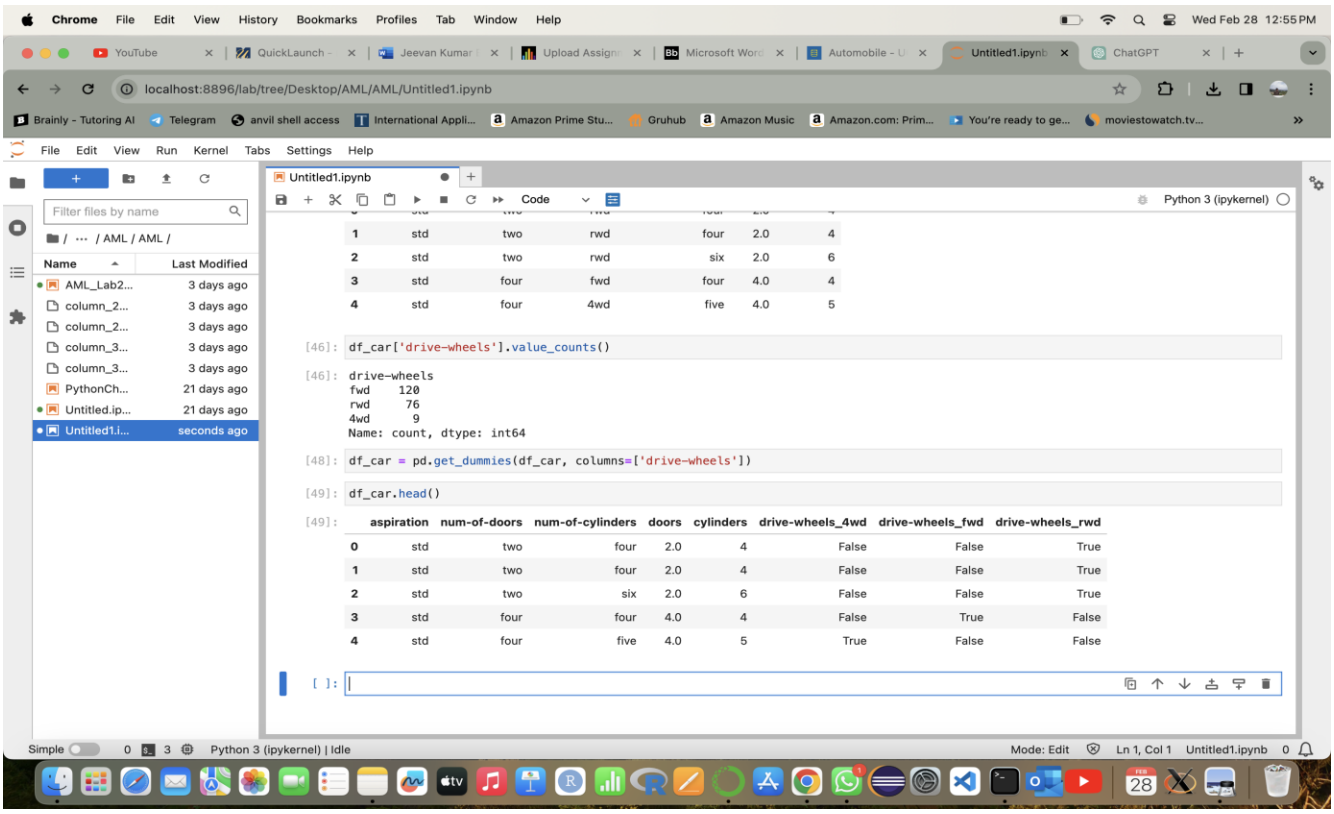
- Now we are replacing the above num-of-doors with num-of-cylinders as below.



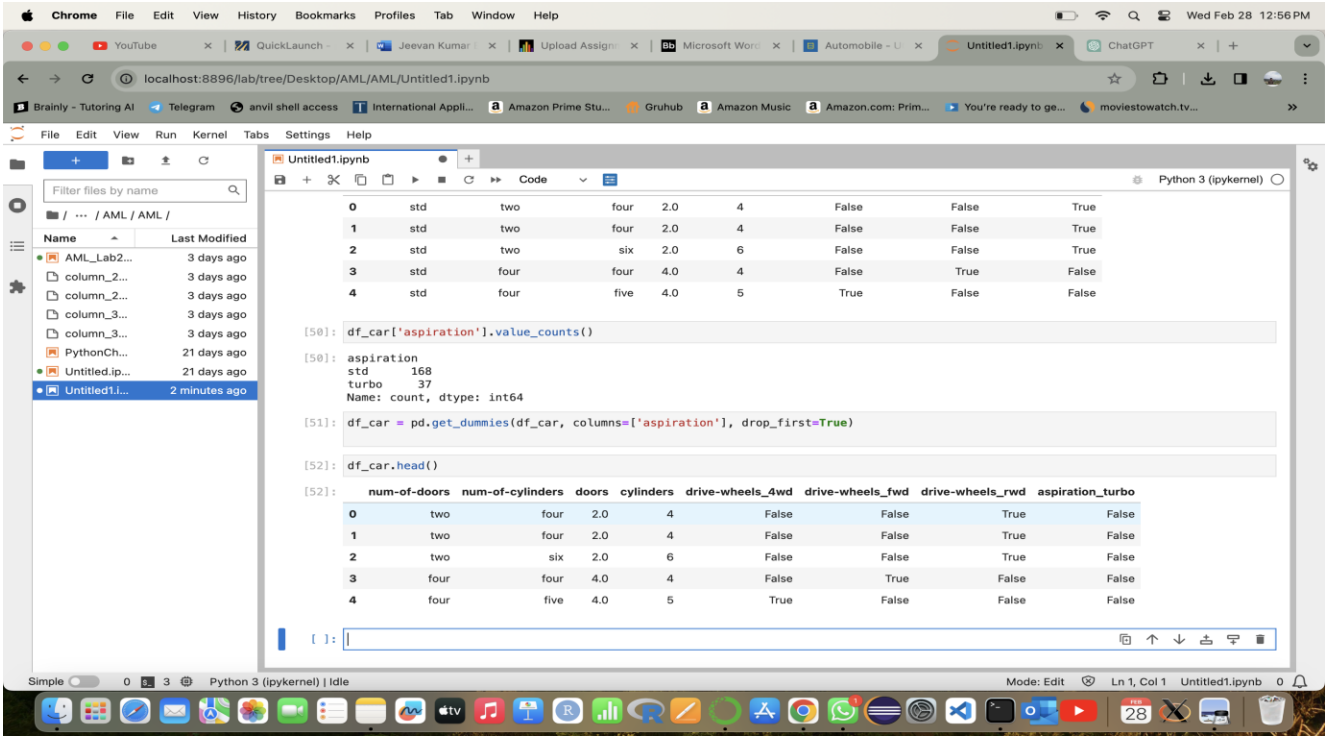
- Similarly for the Cylinders column we can create a mapper dictionary and replace the values in the dataset with the respective values as above.

Step 3: Encoding non-ordinal categorical data:

- In this step, we are using a library from pandas called “get_dummies”, we will encode non-ordinal data by using it.
- Here for the non-ordinal data, we can't create a mapper dictionary and replace it.
- The ‘get_dummies’ method adds the separate columns for separate values and then converts the data into Boolean Data types.



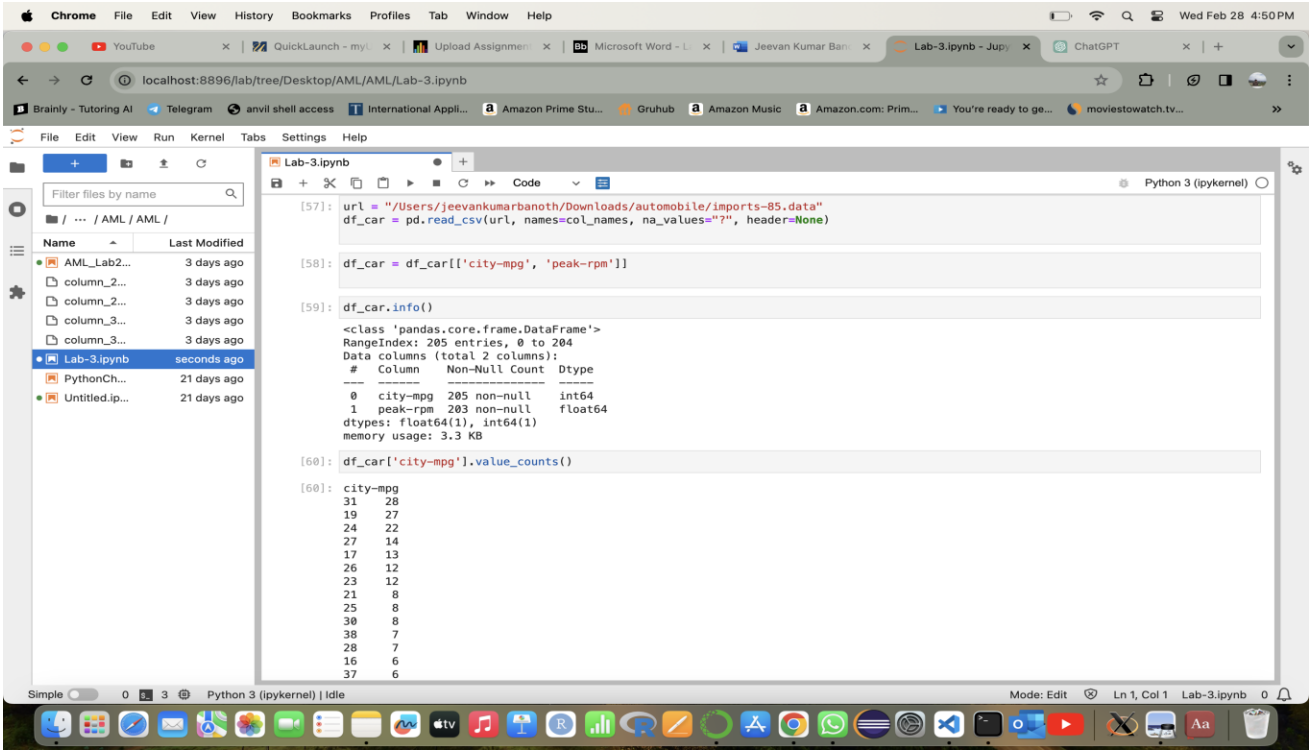
- As we examine the dataset, we can see there are three new columns in the dataset.
- Here, the binary variables enable us to express the information in the numerical way without any order to it.
- Now we are examining the final column which is “aspiration”, it has only two values std and turbo.
- We will use here, get_dummies method but will specify drop_first as true.



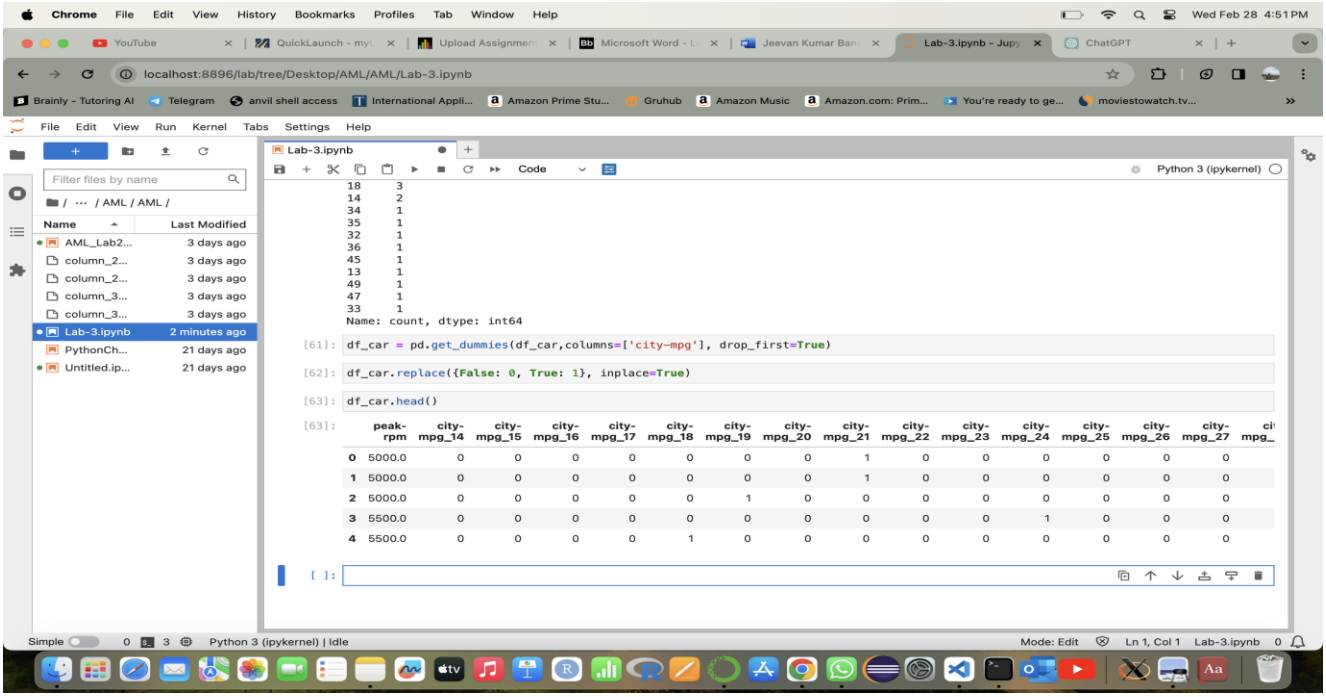
Challenge task:

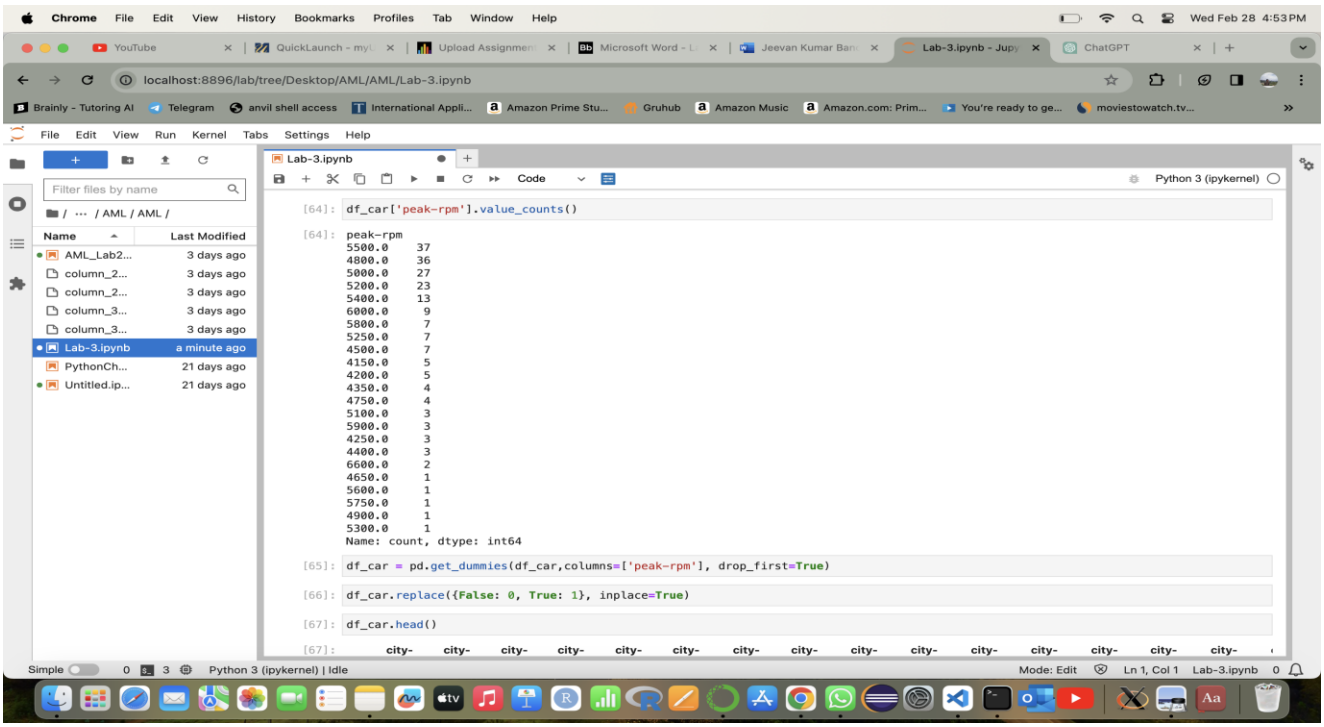
Go back to the beginning of this lab and add other columns to the dataset. Add at least two columns. How would you encode the values of for that column? Update the code to include some of the other features. Do it for at least one more column.

- So, here I have used the city-mpg and peak-rpm from the dataset.
- To do this, we must load the dataset again as follows.



- So, we simply must use the pandas 'get_dummies' method to create new columns and with the value names and convert them into binary values.
- The same method is used for the engine location column too.





Conclusion:

- We started by loading the dataset to get an idea of its structure, size, and variables.
- We converted the raw data from a CSV file into a Data Frame to make it easier to work with.
- We identified which columns held object data types, which usually represent categories.
- We took care to handle categorical data properly, classifying them as ordinal or non-ordinal.
- For ordinal features, we created a mapper dictionary to convert categorical values into numerical equivalents while keeping the order of the categories.
- We replaced the original ordinal values in the dataset with their numerical equivalents.

To summarize, our data preprocessing steps lay the groundwork for accurate and efficient training of machine learning models. We ensure our dataset is in a usable format for advanced data analysis and modeling techniques. This involves handling data import, examining the data, segregating it, and encoding it. Data preprocessing is an essential and foundational stage in our data analysis and modeling pipeline.