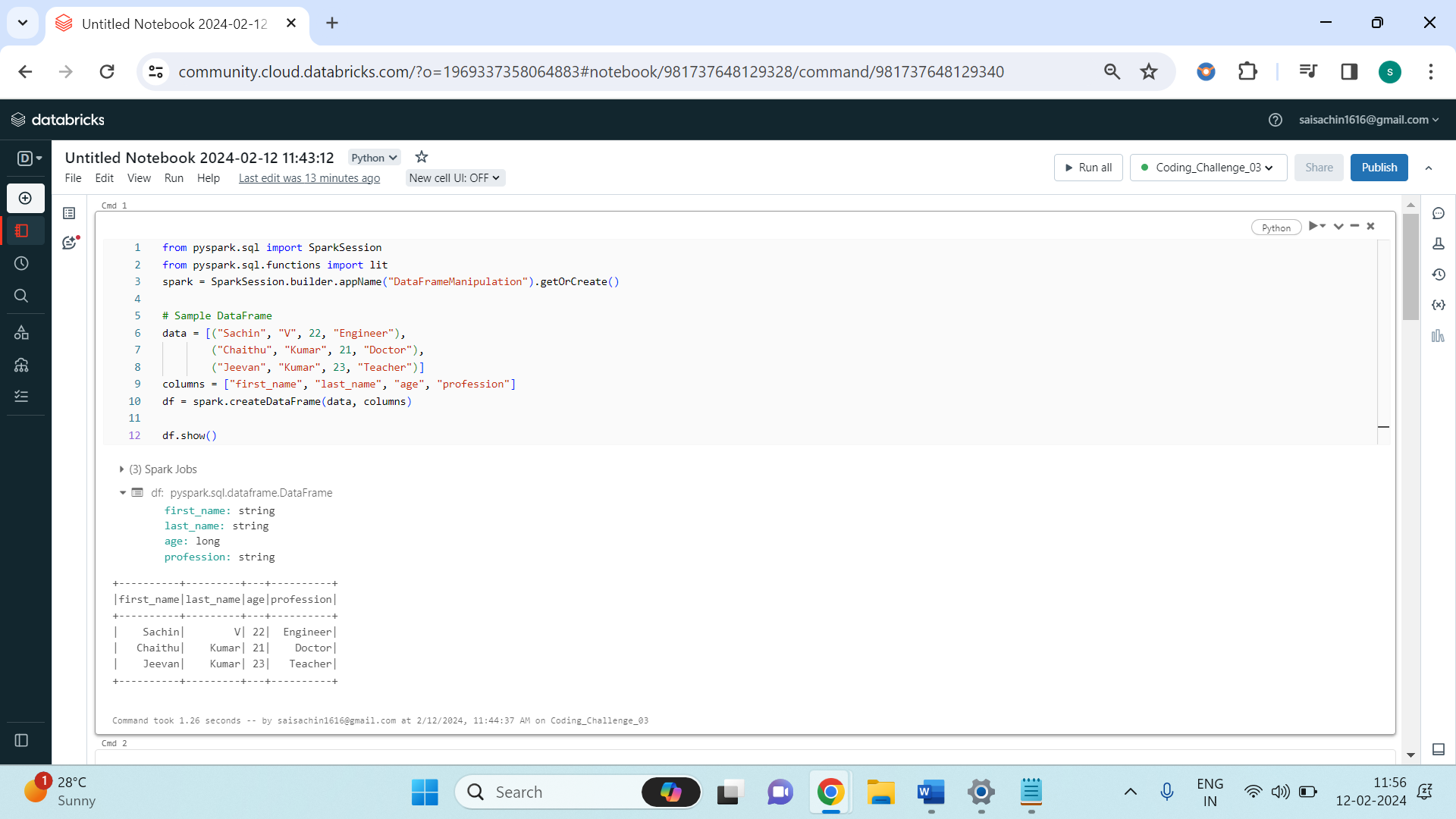
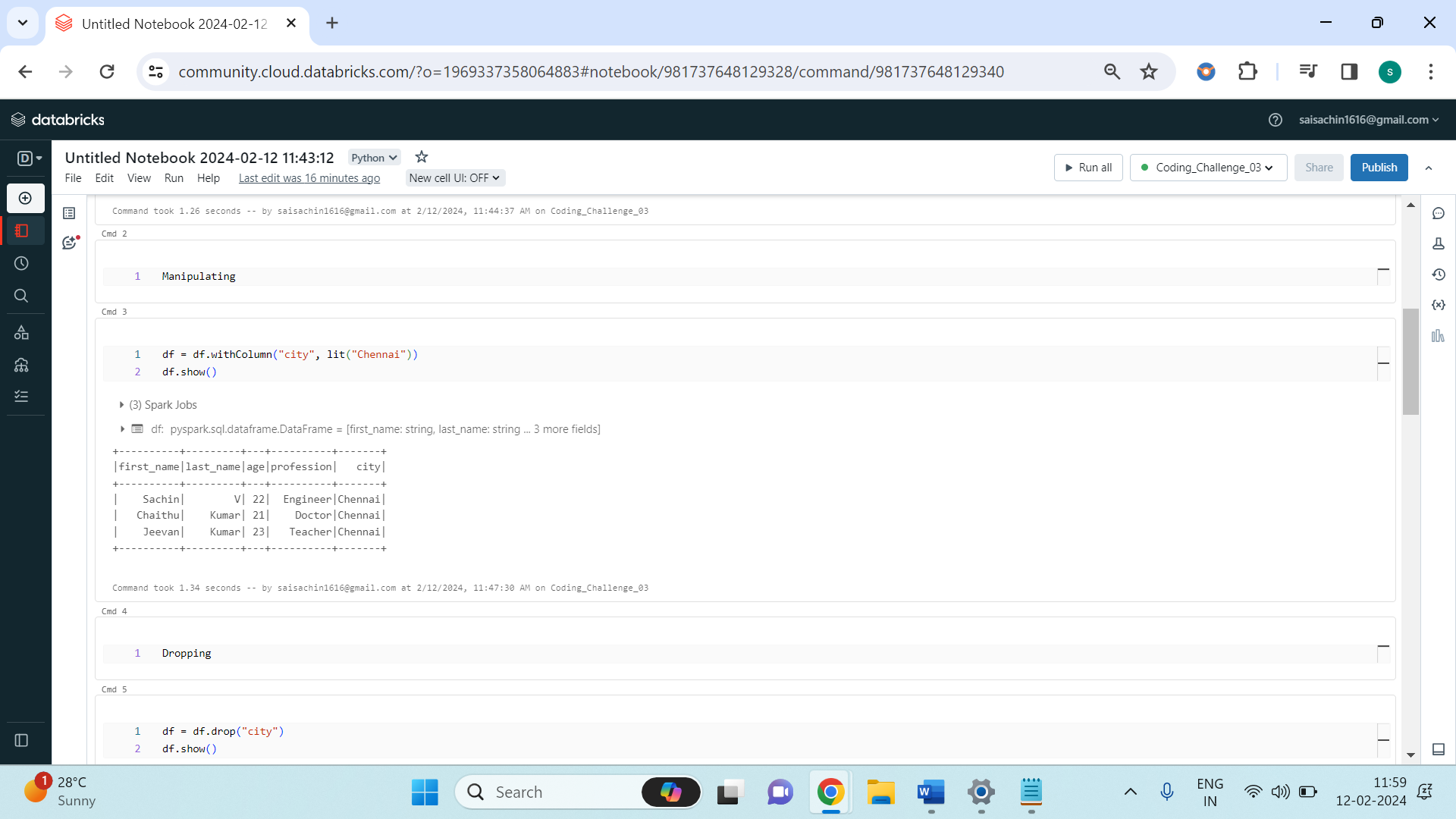
The above questions states me that I need to take a sample data set and perform manipulating, Droping, Sorting, Aggregations, Joining, and GrouoBy Dataframes:

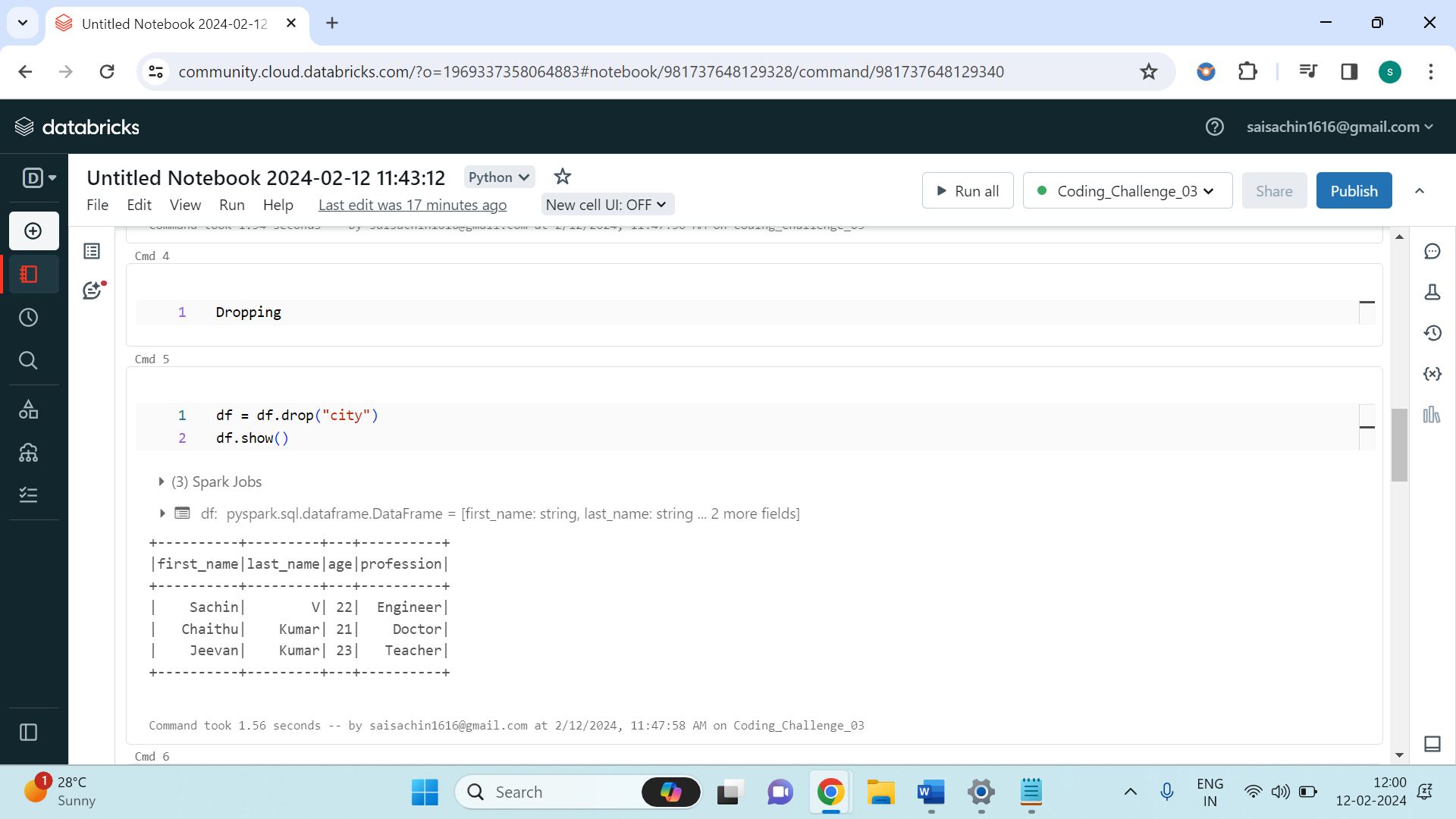


From the above image, you can see that the code begins by importing the necessary modules from the PySpark library. It imports the **SparkSession** class, which serves as the entry point to programming with Spark, and the **lit** function from **pyspark.sql.functions**, used to create a column with a constant value. Following the imports, a SparkSession named **spark** is initialized with the application name "**DataFrameManipulation**". This session serves as the interface to interact with Spark functionalities.

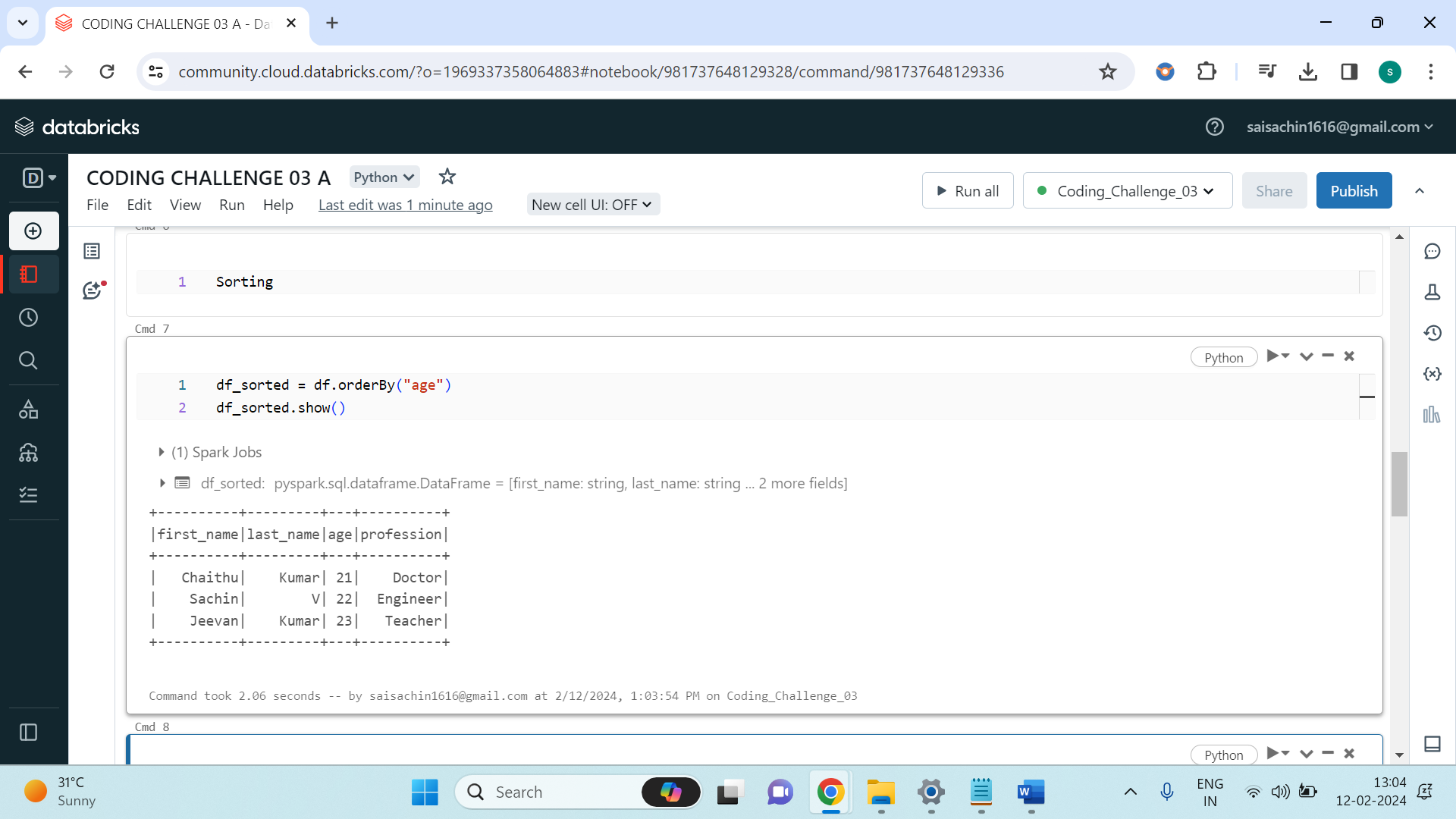
Next, a sample DataFrame named **df** is created using predefined data and column names. This DataFrame contains information about individuals, such as their names, ages, and professions.



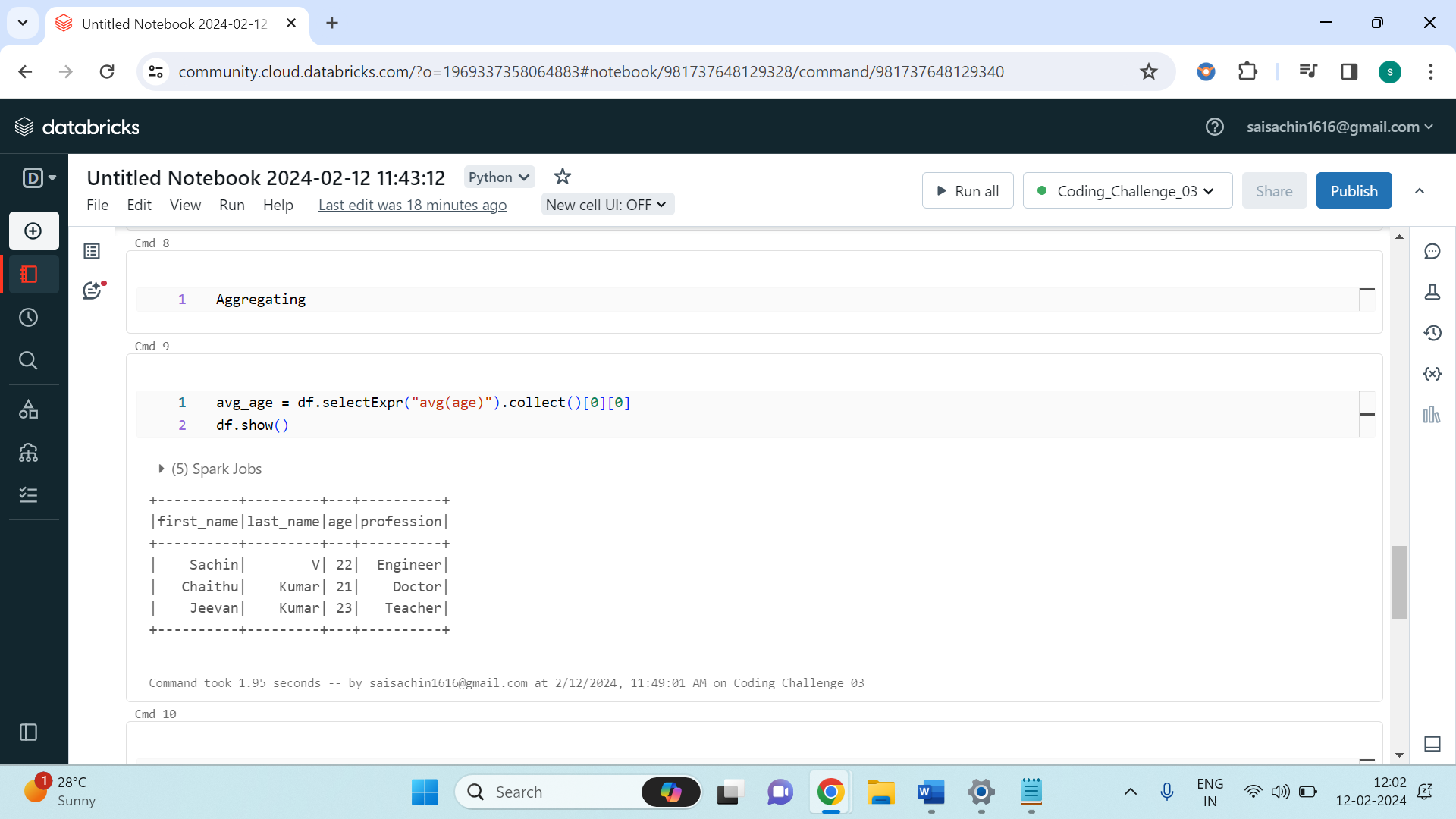
Subsequently, a new column named "city" with the constant value "Chennai" is added to **df**, and then the column is dropped.



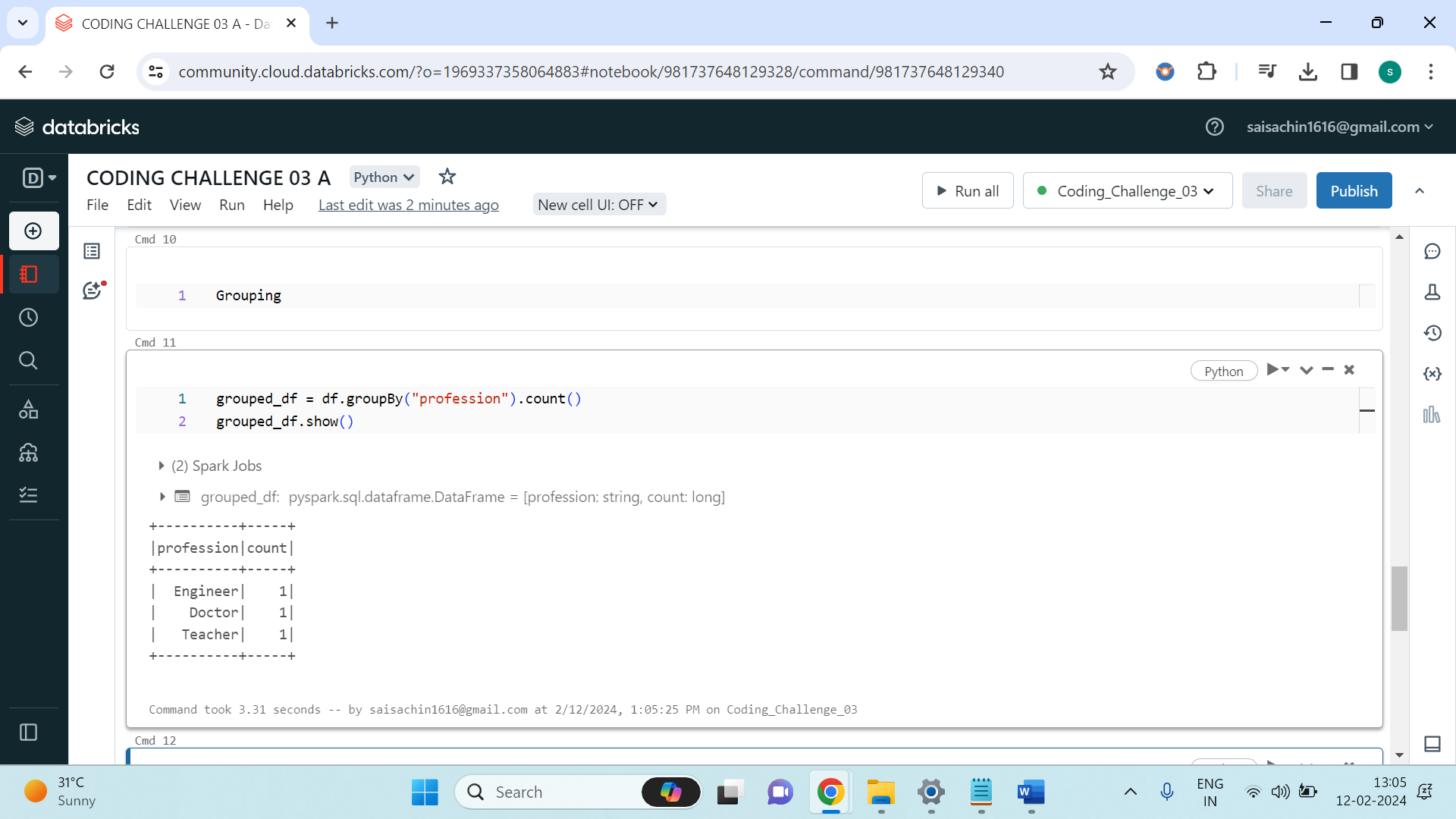
The DataFrame is then sorted by age.



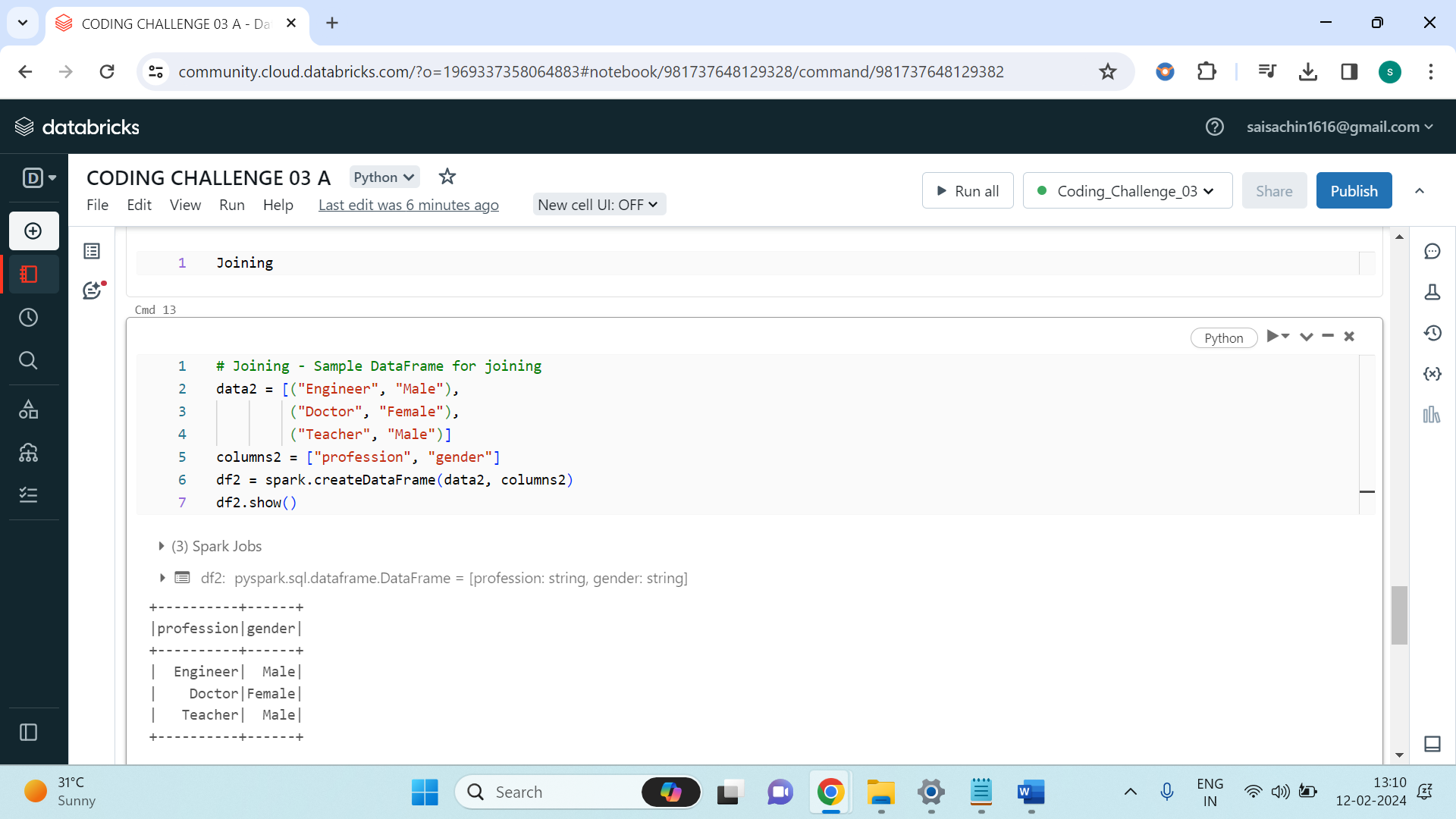
The average age is calculated as shown below.

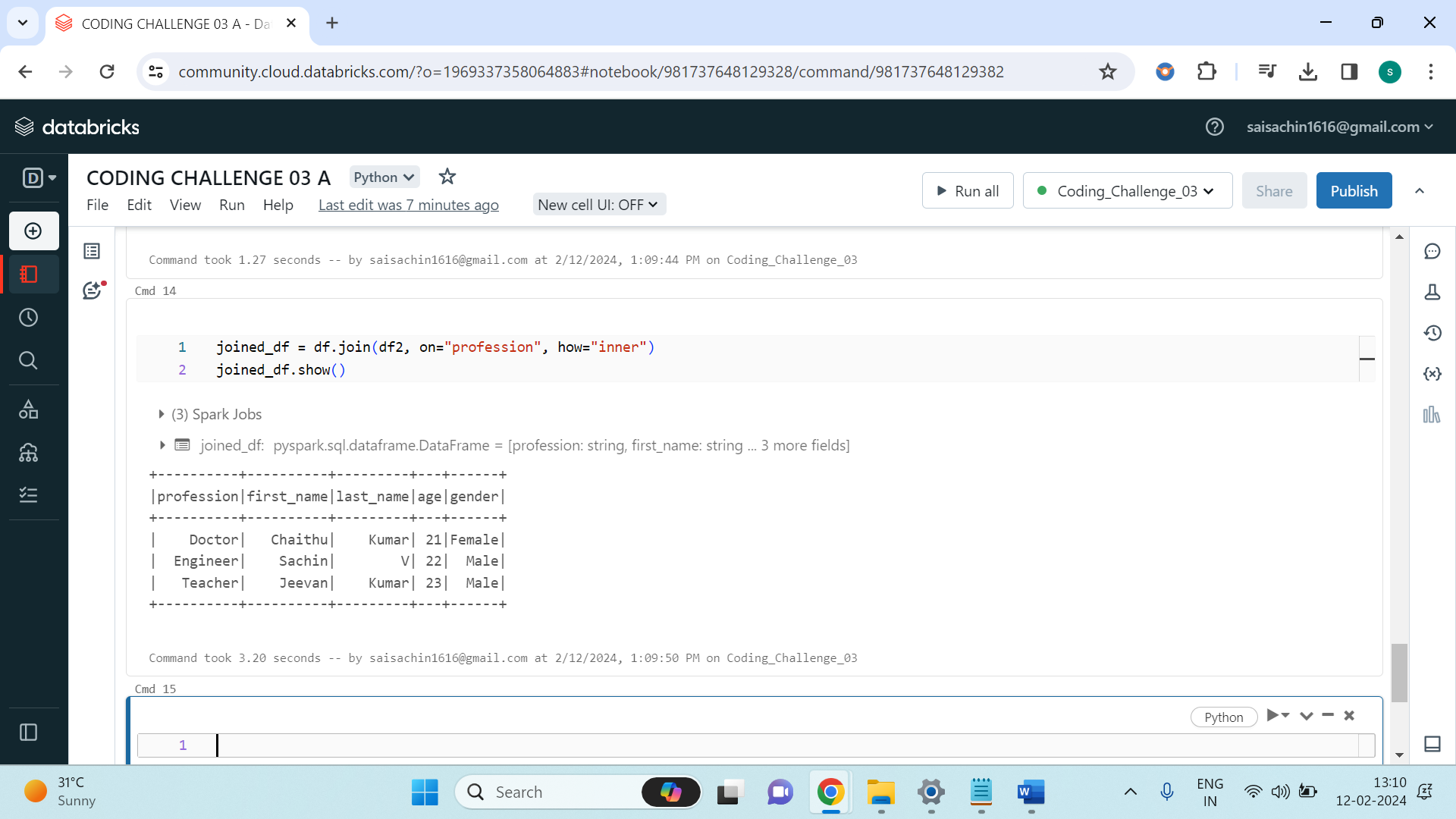


Additionally, the DataFrame is grouped by profession to count the occurrences of each profession.



Furthermore, another sample DataFrame **df2** is created, containing information about professions and genders. The two DataFrames **df** and **df2** are then joined based on the "profession" column using an inner join operation (**join()**) to produce a new DataFrame **inner\_join\_df**.





Overall, the code showcases basic DataFrame manipulation operations such as column addition, dropping, sorting, aggregating, grouping, and joining in PySpark, demonstrating some fundamental functionalities for data processing and analysis within a Spark environment.