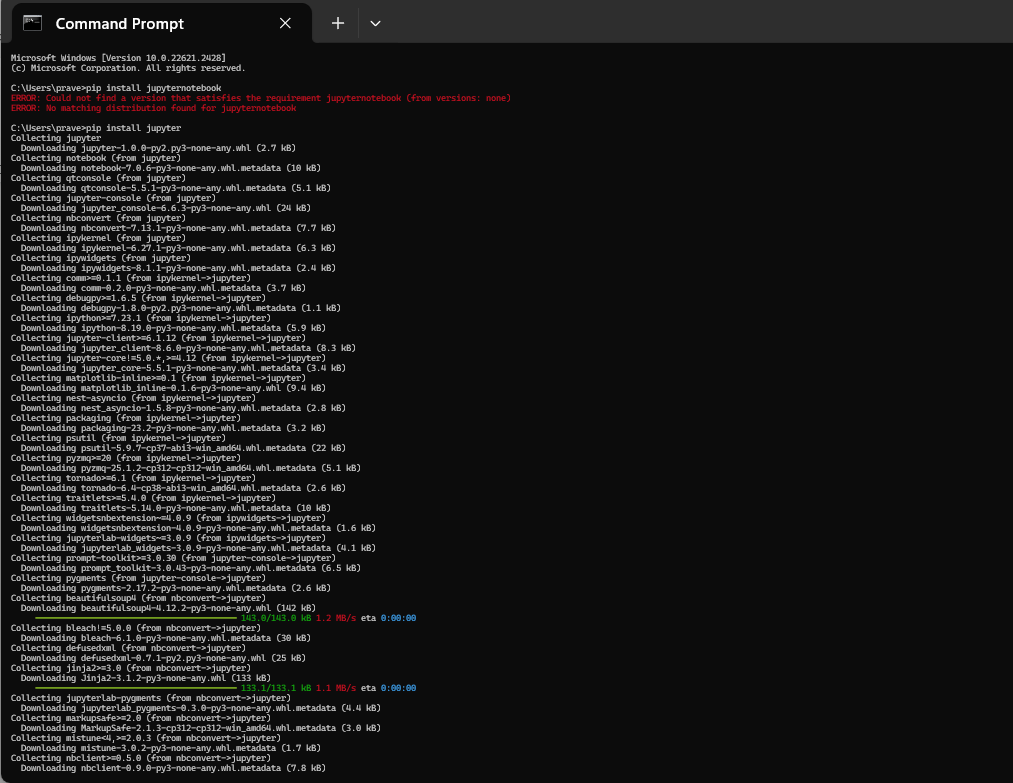
K. Praveen kumar

Pyspark assignment-4

**Executing a Sample program in Jupyter Notebook:**

To execute a program, we need to install the jupyter notebook in our local system by giving the below command in the command prompt:

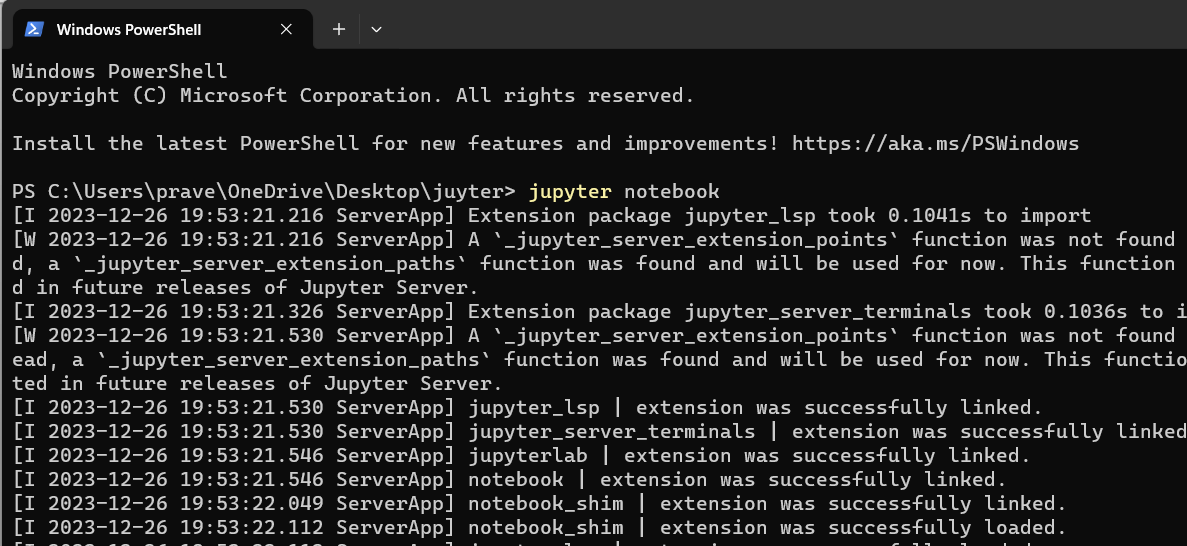
pip install jupyter



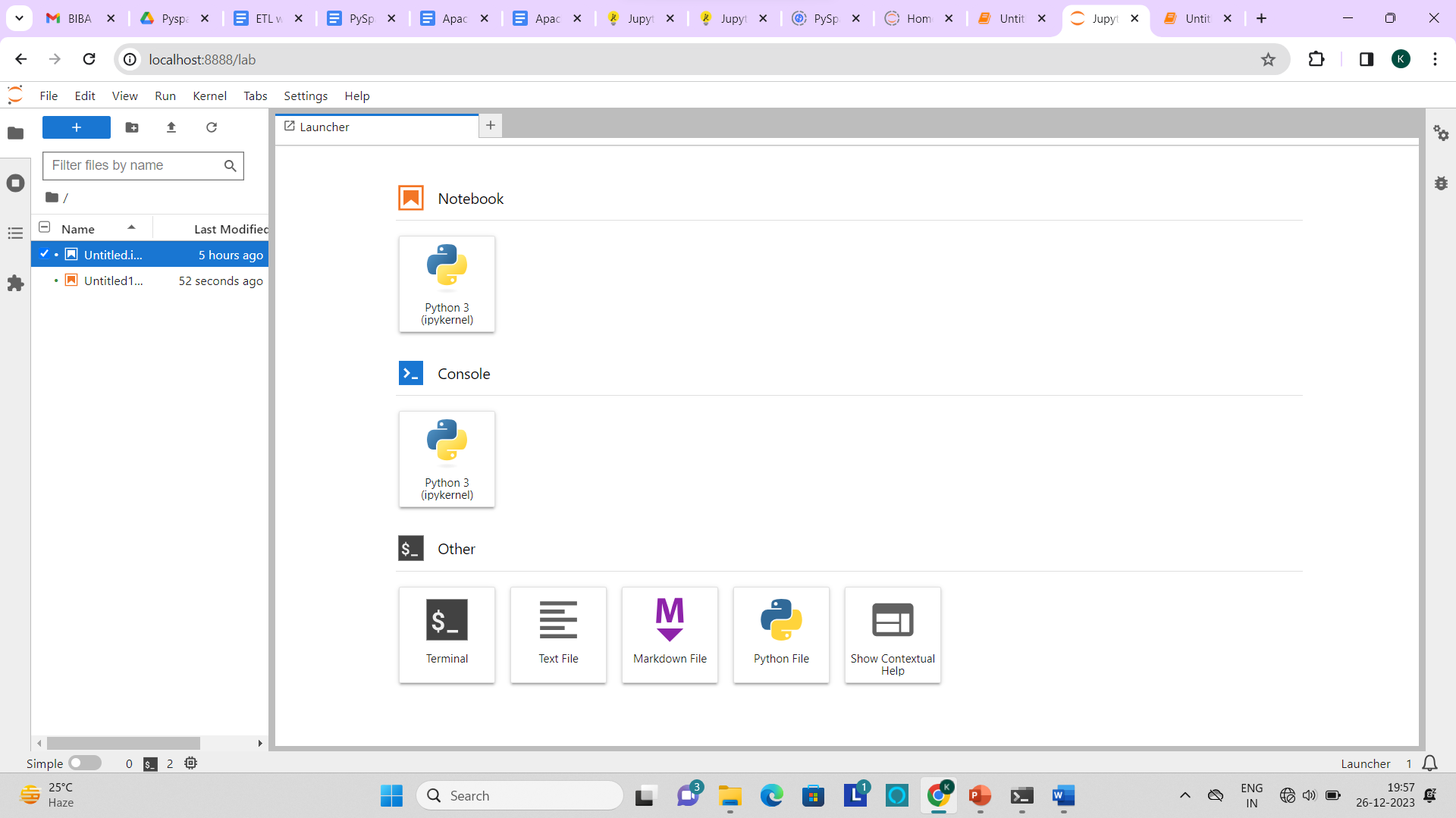
It will take up to few minutes and after the installation is done. Go to desktop and create a folder.

Open the particular folder and right click and click on open in terminal.

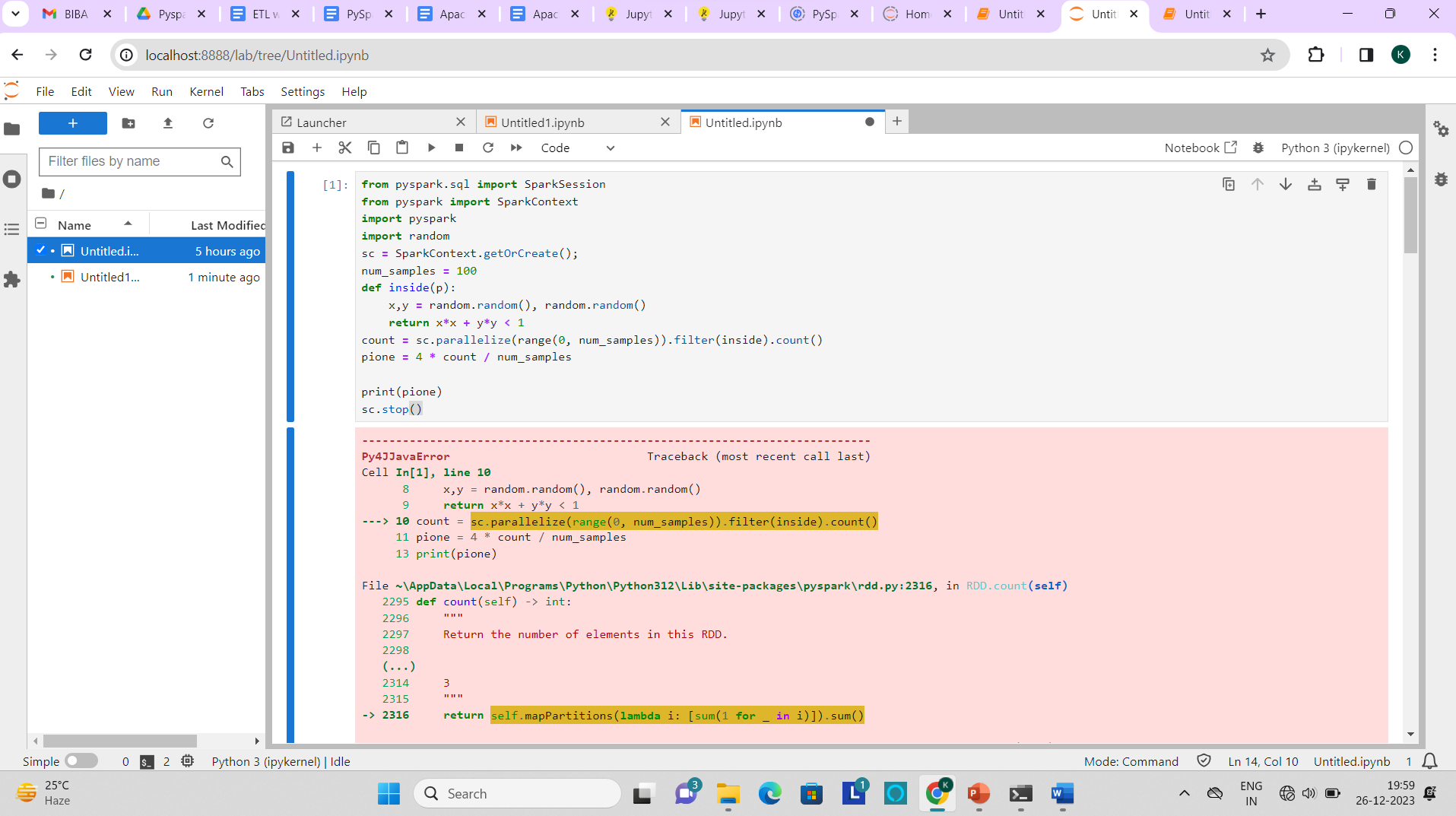
Now the Windows PowerShell will be opened and in that give the command as “Jupyter notebook”.



Now it will be redirected to the web and opens the jupyter notebook with an local host id of 8888 as seen in the below image:



In that we can create a new file and write the sample program in it.



So in this way we can run the pysaprk examples in the jupyter notebook.

**PySpark ETL Work Flow:**

**1.Extract:** Retrieve data from various sources like databases, files or API’s

**2.Transform:** Clean, Aggregate and manipulate data to fit your analysis needs.

**3.Load:** Store the transformed data into a database or data warehouse for analysis.

Example:

from pySpark.sql import SparkSession

from pyspark.sql.functions import col, concat, lit, floor, rand

**#initializing a spark session**

Spark= SparkSession.builder.appName(“complexETL”).getOrCreate()

**#define the external sources and target paths**

Source\_path= ‘actual source\_path’

Target\_path= ‘actual output\_path’

**#Extract: Read data from an external CSV file**

df= Spark.read.csv(source\_path, header=True, schema= ‘cust\_id int, first\_name string, last\_name string, cust\_order int, cust\_status string’)

**#Transformation 1: concatenate first and last Names**

df= df.withColumn(“full\_name”, concat(col(“first\_name”),lit(“ ”), col(“last\_name”)))

**#Transformation2: Calculate Net salary**

df= df.withColumn(“net\_salary”, floor(lit(10000)+rand()\*lit(50)))

**#adding age column**

df=df.withColumn(“age”,floor(lit(20)+rand()\*lit(31)))

**#Load: Save the transferred data to an external CSV file**

df.write.csv(target\_path, mode=”overwrite”, header=True)