**PRACTICAL-7**

**AIM:** Perform the following apache spark program in DATABRICKS.

**1**. Find the average number of friends by age. (avgfriends.csv)

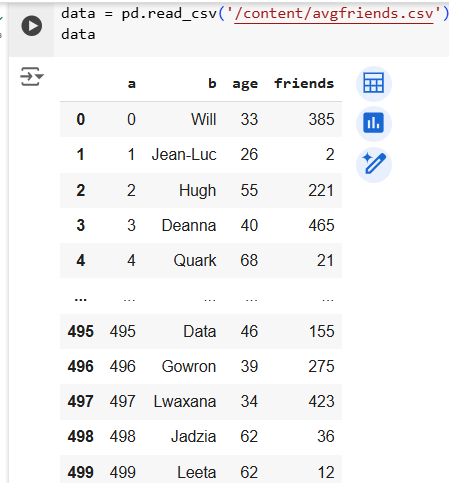
**Code:-**

import pandas as pd

import numpy as np

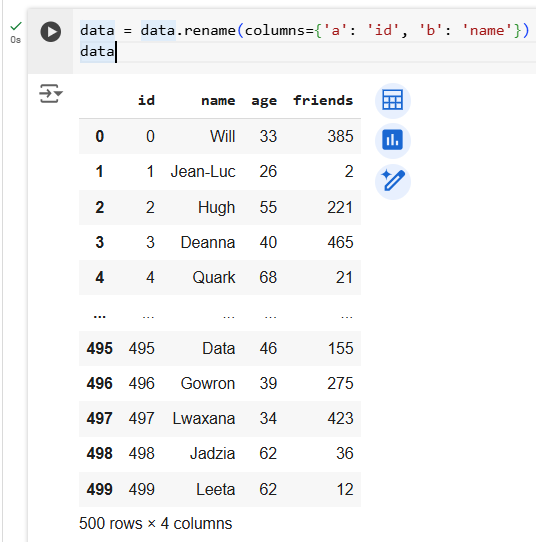
data = pd.read\_csv('/content/avgfriends.csv')

data

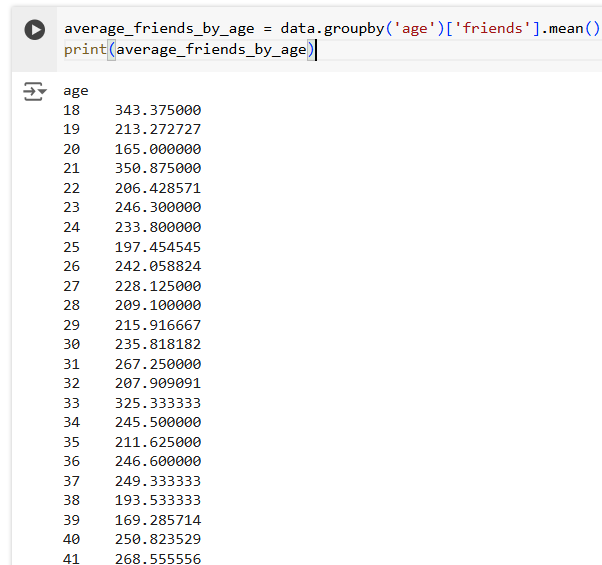


data = data.rename(columns={'a': 'id', 'b': 'name'})

data



average\_friends\_by\_age = data.groupby('age')['friends'].mean()

print(average\_friends\_by\_age)

**2.** Use the dataset given and write the code to find the minimum temperature by the location (each whether station) and understand it and modify it to find maximum temperature by the location. (temp.csv)

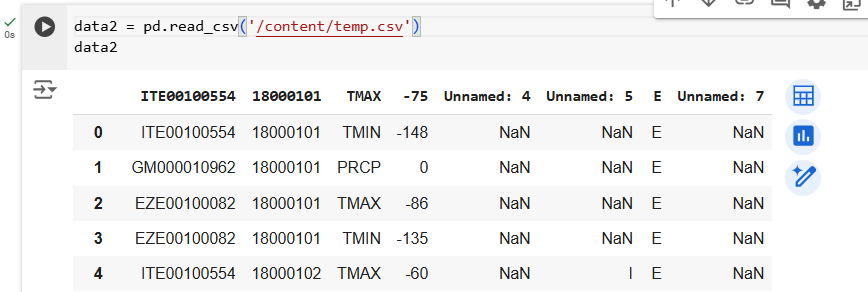
**Code:-**

import numpy as np

import pandas as pd

data2 = pd.read\_csv('/content/temp.csv')

data2



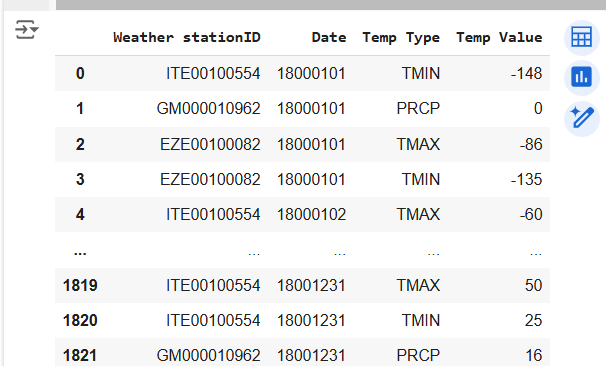
new\_column\_names = ["Weather stationID", "Date", "Temp Type", "Temp Value"]

# Select the desired columns and rename them

data2 = data2[[data2.columns[0], data2.columns[1], data2.columns[2], data2.columns[3]]]  # Select the first 4 columns

data2.columns = new\_column\_names  # Rename the selected columns

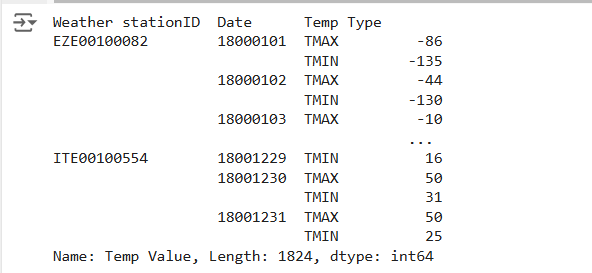
data2



temp\_by\_location = data2.groupby(['Weather stationID', 'Date', 'Temp Type'])['Temp Value'].min()

# Display the result

print(temp\_by\_location)



**3.** Use a given dataset of customers and their spending; find how much amount is spent by the individual customer in total, creating proper RDD in the databricks python notebook and sort out result based on the total spent. (customerorders.csv)

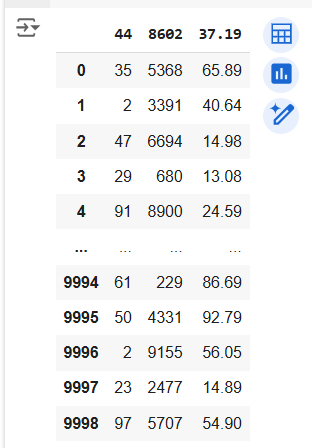
**Code:-**

import numpy as np

import pandas as pd

data3 = pd.read\_csv('/content/customerorders.csv')

data3

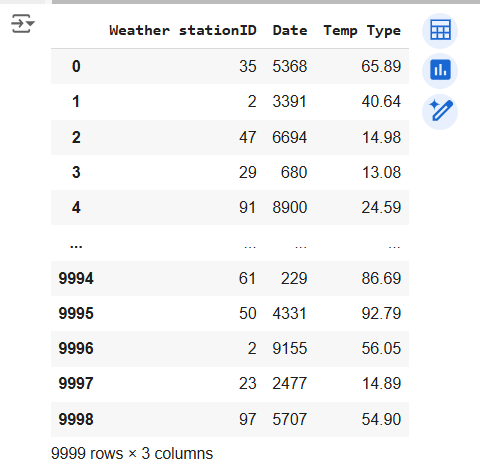


# Select the desired columns and rename them

data3 = data3[[data3.columns[0], data3.columns[1], data3.columns[2]]]

data3.columns = new\_column\_names[:3]

data3



from pyspark import SparkContext, SparkConf

# Create a Spark context

conf = SparkConf().setAppName("CustomerSpending")

sc = SparkContext(conf=conf)

# Load the data into an RDD

customer\_orders = sc.textFile("/content/customerorders.csv")

# Split each line into customer ID and order amount

customer\_amounts = customer\_orders.map(lambda line: line.split(",")).map(lambda fields: (int(fields[0]), float(fields[2])))

# Calculate total spending for each customer

total\_spending = customer\_amounts.reduceByKey(lambda a, b: a + b)

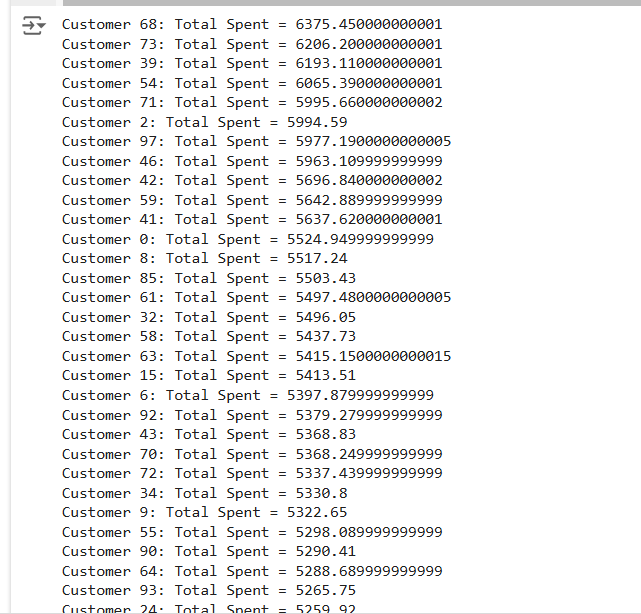
# Sort the results by total spending in descending order

sorted\_spending = total\_spending.sortBy(lambda x: x[1], ascending=False)

# Collect and print the results

for customer\_id, total\_amount in sorted\_spending.collect():

    print(f"Customer {customer\_id}: Total Spent = {total\_amount}")



**4.** Use a text-file given as dataset and count the number of words occur in it. Also, use regular expressions to clean and count the number of words and sort out your output.(wordcount data.txt)

**Code:-**

import re

from pyspark import SparkContext, SparkConf

conf = SparkConf().setAppName("WordCount")

sc = SparkContext(conf=conf)

# Load the text file into an RDD

text\_file = sc.textFile("/content/wordcount data.txt")

# Clean and split the text into words using regular expressions

words = text\_file.flatMap(lambda line: re.findall(r'\b\w+\b', line.lower()))

# Count the occurrences of each word

word\_counts = words.map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)

# Sort the word counts in descending order

sorted\_word\_counts = word\_counts.sortBy(lambda x: x[1], ascending=False)

# Collect and print the results

for word, count in sorted\_word\_counts.collect():

    print(f"{word}: {count}")

