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 = pd.read_csv('/content/avgfriends.csv')
data
```

	a	b	age	friends
0	0	Will	33	385
1	1	Jean-Luc	26	2
2	2	Hugh	55	221
3	3	Deanna	40	465
4	4	Quark	68	21
...	...	...	...	...
495	495	Data	46	155
496	496	Gowron	39	275
497	497	Lwaxana	34	423
498	498	Jadzia	62	36
499	499	Leeta	62	12

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

data
```

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

	id	name	age	friends
0	0	Will	33	385
1	1	Jean-Luc	26	2
2	2	Hugh	55	221
3	3	Deanna	40	465
4	4	Quark	68	21
...	...	...	...	...
495	495	Data	46	155
496	496	Gowron	39	275
497	497	Lwaxana	34	423
498	498	Jadzia	62	36
499	499	Leeta	62	12

500 rows × 4 columns

average_friends_by_age = data.groupby('age')['friends'].mean()

print(average_friends_by_age)

```
average_friends_by_age = data.groupby('age')['friends'].mean()
print(average_friends_by_age)
```

```
age
18    343.375000
19    213.272727
20    165.000000
21    350.875000
22    206.428571
23    246.300000
24    233.800000
25    197.454545
26    242.058824
27    228.125000
28    209.100000
29    215.916667
30    235.818182
31    267.250000
32    207.909091
33    325.333333
34    245.500000
35    211.625000
36    246.600000
37    249.333333
38    193.533333
39    169.285714
40    250.823529
41    268.555556
```

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

```
data2 = pd.read_csv('/content/temp.csv')
data2
```

	ITE00100554	18000101	TMAX	-75	Unnamed: 4	Unnamed: 5	E	Unnamed: 7
0	ITE00100554	18000101	TMIN	-148	NaN	NaN	E	NaN
1	GM000010962	18000101	PRCP	0	NaN	NaN	E	NaN
2	EZE00100082	18000101	TMAX	-86	NaN	NaN	E	NaN
3	EZE00100082	18000101	TMIN	-135	NaN	NaN	E	NaN
4	ITE00100554	18000102	TMAX	-60	NaN	I	E	NaN

```
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
```

	Weather stationID	Date	Temp Type	Temp Value
0	ITE00100554	18000101	TMIN	-148
1	GM000010962	18000101	PRCP	0
2	EZE00100082	18000101	TMAX	-86
3	EZE00100082	18000101	TMIN	-135
4	ITE00100554	18000102	TMAX	-60
...	...	...	...	...
1819	ITE00100554	18001231	TMAX	50
1820	ITE00100554	18001231	TMIN	25
1821	GM000010962	18001231	PRCP	16

```
temp_by_location = data2.groupby(['Weather stationID', 'Date', 'Temp Type'])['Temp Value'].min()

# Display the result

print(temp_by_location)
```

```
Weather stationID  Date      Temp Type
EZE00100082        18000101  TMAX         -86
                             TMIN        -135
                   18000102  TMAX         -44
                             TMIN        -130
                   18000103  TMAX         -10
                                          ...
ITE00100554        18001229  TMIN          16
                   18001230  TMAX          50
                             TMIN          31
                   18001231  TMAX          50
                             TMIN          25
Name: Temp Value, Length: 1824, dtype: int64
```

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
```

	44	8602	37.19
0	35	5368	65.89
1	2	3391	40.64
2	47	6694	14.98
3	29	680	13.08
4	91	8900	24.59
...	...	...	...
9994	61	229	86.69
9995	50	4331	92.79
9996	2	9155	56.05
9997	23	2477	14.89
9998	97	5707	54.90

```
# 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
```

	Weather stationID	Date	Temp Type
0	35	5368	65.89
1	2	3391	40.64
2	47	6694	14.98
3	29	680	13.08
4	91	8900	24.59
...	...	...	...
9994	61	229	86.69
9995	50	4331	92.79
9996	2	9155	56.05
9997	23	2477	14.89
9998	97	5707	54.90

9999 rows × 3 columns

```
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}")
```

```
Customer 68: Total Spent = 6375.450000000001
Customer 73: Total Spent = 6206.200000000001
Customer 39: Total Spent = 6193.110000000001
Customer 54: Total Spent = 6065.390000000001
Customer 71: Total Spent = 5995.660000000002
Customer 2: Total Spent = 5994.59
Customer 97: Total Spent = 5977.1900000000005
Customer 46: Total Spent = 5963.109999999999
Customer 42: Total Spent = 5696.840000000002
Customer 59: Total Spent = 5642.889999999999
Customer 41: Total Spent = 5637.620000000001
Customer 0: Total Spent = 5524.949999999999
Customer 8: Total Spent = 5517.24
Customer 85: Total Spent = 5503.43
Customer 61: Total Spent = 5497.4800000000005
Customer 32: Total Spent = 5496.05
Customer 58: Total Spent = 5437.73
Customer 63: Total Spent = 5415.1500000000015
Customer 15: Total Spent = 5413.51
Customer 6: Total Spent = 5397.879999999999
Customer 92: Total Spent = 5379.279999999999
Customer 43: Total Spent = 5368.83
Customer 70: Total Spent = 5368.249999999999
Customer 72: Total Spent = 5337.439999999999
Customer 34: Total Spent = 5330.8
Customer 9: Total Spent = 5322.65
Customer 55: Total Spent = 5298.089999999999
Customer 90: Total Spent = 5290.41
Customer 64: Total Spent = 5288.689999999999
Customer 93: Total Spent = 5265.75
Customer 24: Total Spent = 5259.92
```

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}")
```

```
data: 16
and: 7
big: 6
to: 6
of: 5
the: 5
that: 4
with: 4
a: 4
in: 3
or: 3
analysis: 3
refers: 2
large: 2
higher: 2
used: 2
is: 2
only: 2
challenges: 2
sampling: 2
thus: 2
be: 2
information: 2
volume: 2
variety: 2
for: 2
```