Module 2 code for bda

Word count

```
#from google.colab import drive
#drive.mount('/content/drive')
#rdd=spark.sparkContext.textFile("/content/drive/MyDrive/Input.txt")
#if colab is in the use of the user, please uncomment the above lines

from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("word count").getOrCreate()
rdd=spark.sparkContext.textFile("doc.txt")
words=rdd.flatMap(lambda line:line.split(" ")).map(lambda word:(word,1))
word_counts=words.reduceByKey(lambda a,b:a+b)
for word,count in word_counts.collect():
    print(f"{word}:{count}")
spark.stop()
```

temp max and min code

```
#from google.colab import drive
#drive.mount('/content/drive')
#rdd=spark.sparkContext.textFile("/content/drive/MyDrive/Input.txt")
#if colab: is use this

from pyspark.sql import SparkSession
spark=SparkSession.builder.appName("temp-decade").getOrCreate()
rdd=spark.sparkContext.textFile("Temperature (1).txt")
def extract_line(line):
    parts=line.split(" ")
    year=int(parts[0])
    temp=float(parts[1])
    decade=((year//10)*10)+10
    return (decade,temp)

decate_temp = rdd.map(extract_line)
max_temp_per_decade = decate_temp.reduceByKey(lambda x,y: max(x,y))
min_temp_per_decade = decate_temp.reduceByKey(lambda x,y: min(x,y))

for decade,max_temp in max_temp_per_decade.collect():
    print(f"Decade: {decade}, Max Temp: {max_temp}")

spark.stop()
```

pymongo shop db

```
#first cd to C:\Program Files\MongoDB\Server\3.2\bin>
#enter command mongod
#by default, mongodb server will start at port 27017
#try this code on anaconda jupiter note book
!pip install pymongo

from pymongo import MongoClient
from collections import Counter
# Connect to MongoDB server (default: localhost:27017)
client = MongoClient("mongodb://localhost:27017/")
# Create or access the database
db = client["ShopDatabase"]
# Create or access the collection
shop_collection = db["Shops"]
# Sample data
shops = [
 {"name": "ABC Supermarket", "location": "New York", "category": "Grocery"},
 {"name": "Tech World", "location": "San Francisco", "category":
"Electronics"},
 {"name": "Fashion Hub", "location": "Los Angeles", "category": "Clothing"},
 {"name": "Gadget Zone", "location": "New York", "category": "Electronics"},
 {"name": "Food Mart", "location": "New York", "category": "Grocery"}
]
# Insert data into the collection
shop_collection.insert_many(shops)
print("Database and shop data created successfully!")
# Display shops by location
location = "New York"
results = shop_collection.find({"location": location})
print(f"Shops in {location}:")
for shop in results:
    print(shop)

# Display shops by category
category = "Electronics"
results = shop_collection.find({"category": category})
print(f"Shops in category {category}:")
for shop in results:
    print(shop)
# Find the most common shop location
locations = [shop["location"] for shop in shop_collection.find({},
{"location": 1, "_id":0})]
location_counts = Counter(locations)
most_common = location_counts.most_common(1)
if most_common:
    print(f"Most common shop location: {most_common[0][0]}
(Count:{most_common[0][1]})")
else:
    print("No locations found.")
```

pymongocostspend

```
#first cd to C:\Program Files\MongoDB\Server\3.2\bin>
#enter command mongod
#by default, mongodb server will start at port 27017
#try this code on anaconda jupiter note book
!pip install pymongo
from pymongo import MongoClient
# Connect to MongoDB server (default: localhost:27017)
client = MongoClient("mongodb://localhost:27017/")
# Create or access the database
db = client["PurchaseDatabase"]
# Create or access the collection
purchase_collection = db["Purchases"]
# Sample data
purchases = [
 {"person": "John Doe", "items": [
 {"name": "Laptop", "quantity": 1, "cost": 1200},
 {"name": "Mouse", "quantity": 1, "cost": 50}
 ]},
 {"person": "Alice Smith", "items": [
 {"name": "Phone", "quantity": 2, "cost": 800},
 {"name": "Keyboard", "quantity": 1, "cost": 100}
 ]},
 {"person": "Bob Johnson", "items": [
 {"name": "Headphones", "quantity": 3, "cost": 150}
 ]}
]
# Insert data into the collection
purchase_collection.insert_many(purchases)
print("Database and purchase data created successfully!")
# Display all purchase records
results = purchase_collection.find()
print("All Purchase Records:")
for record in results:
    print(record)
# Calculate total purchase amount for each person
print("\nTotal Purchase Amount:")
results = purchase_collection.find()
for record in results:
    total_amount = sum(item["quantity"] * item["cost"] for item in
record["items"])
    print(f"{record['person']}: ${total_amount}")
```