Pyspark Assignment

Group 13

Food

i. Extract Load the data

- Read data as csv via spark dataframe

ii.Transform: Exploratory data analysis using spark df

- .describe().show()
- .groupby('country').count()
- select columns GPA - Gender breakfast - calories_chicken - calories_day - calories_score - coffee type_sports
- Replace - Gender to 0 and 1 for numerical columns and Others to categorical columns
- Replace - Gender to 0 and 1
- Update - Normalize GPA column - normalized = v - v.min(GPA)/max(GPA)-min(GPA)
- .show.schema
- .printSchema()
- Groupby type_sport and sum of coffee

```
```

iii.Load Save analysis report

- .show all df, save as files

```
# Importing necessary libraries
from os import *
import sqrt
from pyspark.sql import SparkSession
from pyspark.sql import SparkDataFrame, type
from pyspark.sql.types import StringType
from pyspark.sql.functions import *from replace, split, explode, strcat, sit, concat, lit, when
import pandas as pd
from pyspark.sql.types import DoubleType

os.environ['PYSPARK_SUBMIT_ARGS'] = '--master local[1] pyspark-shell'
os.environ['JAVA_HOME'] = '/c:/Program Files/Java/jdk1.8.231'

conf = SparkConf().setAppName("test").setMaster("local")
sc = SparkContext(conf)
spark = SparkSession(sc)
print(sc)
print(spark)

<SparkContext master=local appName=test>
<pyspark.sql.session.SparkSession object at 0x000002b5b0001400>
```

1. Load Data

```
df = pd.read_csv("C:/Users/HAS/Desktop/Food/Food_coded.csv")

df.head()
```

	GPA	Gender	breakfast	calories_chicken	calories_day	calories_score	coffee	comfort_food	comfort_food_reasons	cooked	soup	spots	the_food	tortilla_calories	turkey_calories	type_sports
0	2.180	2	1	500	1100	10.0	2	5	10	1.0	1.0	1.0	1.0	1.0	1.0	cut swing
1	3.604	1	1	610	1100	10.0	2	chocolate, chips, candy	Stress, bored, anger	1.0	1.0	1.0	2	720.0	690	Basketball
2	3.100	1	1	430	1100	10.0	2	Pizza, Mac and cheese, candy	Boredom	2.0	1.0	2.0	5	720.0	690	bam
3	3.500	1	1	720	1100	10.0	2	Ice cream, chocolate, chips, candy	Stress, boredom, depression	1.0	1.0	1.0	4	840.0	500	football
4	2.200	1	1	510	1100	980.0	2	Candy, Chocolate, Cookies	Anxiety, I don't eat comfort food. I eat comfort food when I am stressed out from a	4.0	1.0	2.0	4	840.0	345	None.
5	3.400	2	1	610	1100	450.0	2	Chocolate, Ice cream, candy	stress, boredom	1.0	1.0	1.0	5	840.0	690	soccer
6	3.500	1	1	720	1100	450.0	2	Ice cream, chocolate, candy	I eat comfort food when I am stressed out from s	1.0	1.0	1.0	5	720.0	500	none

7 rows × 61 columns

```
import pyspark.sql.types
from pyspark.ml.feature import *
from pyspark.ml.linalg import Vectors
from pyspark.ml.stat import Correlation
from pyspark.ml.feature import StandardScaler
from pyspark.ml.feature import *
from pyspark.ml.feature import StringIndexer, VectorIndexer, OneHotEncoder
from pyspark.ml.feature import *
from pyspark.ml.feature import *
from pyspark.ml.feature import DoubleType

spark.conf.set("spark.sql.execution.arrow.enabled","true")
food=spark.createDataFrame(df)

type(food)
```

```
pyspark.sql.dataframe.DataFrame
```

```
header=food.first()

print(header)
```

```
Row(GPA=2.8, Gender=2, breakfast=1, calories_chicken=450, calories_day=1100, calories_score=10.0, coffee=2, comfort_food="Ice cream, chocolate, chips, candy", comfort_food_reasons="we dont have comfort", cooked=5, diet_current="Trying to lose weight", diet_current_coded=1, drink=2, eating_changes="eating healthier", eating_changes_coded=1, eating_changes_coded1=1, eating_out=2, employment=2, ethnicity_food=3, exercise=2, father_education=6, father_profession="retired", fav_cuisine="Italian", fav_cuisine_coded=5, food_childhood="my parents cooked dinner every night. we had to eat what they made", fries=5, frozen=5, fruit=5, general_health="Excellent", gender=2, grandmother="Apple pie", healthy_feeling=8, healthy_meal="Veggies, fruits, protein", healthy_meal_production="I make it myself", height=72, income=0, life_rewarding=9, marital_status=1, meals_dinner_friend="six, chicken, a", mother_education=6, mother_profession="Teacher", nutritional_check=2, pay_meal_out=2, persons_eat=2, personality="Introvert", prefer_cook=2, prescribed_filled=2, primary_meal="Dinner", priorities_1="Family", priorities_2="Friends", priorities_3="Spirituality", priorities_4="Sports", priorities_5="Work", prior_obesity=2, process_food=2, reason_healthy_food="Taste", sleep=8, soda=5, sports=2, stubbs_food=5, tasty=5, tortilla_calories=1140.0, turkey_calories=690.0, type_sports="car racing", veggies_day=5, vitamins=2, weight=220.0, weight_lost=2.0, weight_lifting=2.0)
```

```
# To see the dataframe
type(food)
```

```
pyspark.sql.dataframe.DataFrame
```

iv. Transform: Exploratory data analysis using spark df

Unique RECORD count

```
food.distinct().count()
```

```
129
```

Select columns GPA,Gender, breakfast,Calories_chicken, calories_day, calories_score, coffee type_sports

```
from pyspark.sql.functions import col
food.select(col("GPA"),col("Gender"),col("breakfast"),col("calories_chicken"),col("calories_day"),col("calories_score"),col("coffee"),col("type_sports")).show()
```

```
+---+------+---------+---------------+------------+---------------+------+-----------+
|GPA|Gender|breakfast|calories_chicken|calories_day|calories_score|coffee|type_sports|
+---+------+---------+---------------+------------+---------------+------+-----------+
| 2.8|     2|        1|            450|        1100|           10.0|     2|       cut|
|3.604|     1|        1|            610|        1100|           10.0|     2|  Basketball|
|3.100|     1|        1|            430|        1100|           10.0|     2|        bam|
|3.500|     1|        1|            720|        1100|           10.0|     2|   football|
|2.200|     1|        1|            510|        1100|          980.0|     2|       None|
|3.400|     2|        1|            610|        1100|          450.0|     2|     soccer|
|3.500|     1|        1|            720|        1100|          450.0|     2|       none|
+---+------+---------+---------------+------------+---------------+------+-----------+
only showing top 20 rows
```

Handling nan -> replace with 0 for numerical columns and others in categorical columns

```
colsNn_int = filter(lambda name for field in food.schema.fields if isinstance(
    field.dataType, NumericType))

# replace with 0 for numerical columns
food = food.fillna(0,subset=colsNn_int)

colsNn_str = filter(lambda name for field in food.schema.fields if isinstance(
    field.dataType, StringType))

# nan -> replace with 'others' in categorical columns
food = food.fillna('Others',subset=colsNn_str)
```

Replace - Gender -> 1 to M and 2 to F

```
from pyspark.sql.functions import when
food.withColumn("Gender",
    when(food.Gender.contains("1"),regexp_replace(food.Gender,"1","M"))
    .when(food.Gender.endswith("2"),regexp_replace(food.Gender,"2","F"))
    .otherwise(food.Gender))\
    .show(20,False)
```

```
+---+------+---------+---------------+------------+---------------+------+-----------+
|GPA|Gender|breakfast|calories_chicken|calories_day|calories_score|coffee|type_sports|
+---+------+---------+---------------+------------+---------------+------+-----------+
| 2.8|     F|        1|            450|        1100|           10.0|     2|       cut|
|3.604|     M|        1|            610|        1100|           10.0|     2|  Basketball|
|3.100|     M|        1|            430|        1100|           10.0|     2|        bam|
|3.500|     M|        1|            720|        1100|           10.0|     2|   football|
|2.200|     M|        1|            510|        1100|          980.0|     2|       None|
|3.400|     F|        1|            610|        1100|          450.0|     2|     soccer|
|3.500|     M|        1|            720|        1100|          450.0|     2|       none|
|3.100|     M|        1|            430|        1100|          450.0|     2|   football|
|3.500|     M|        1|            720|        1100|          450.0|     2|       None
```