databricks Food Online Project

(https://databricks.com)

Import Librairies

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import IntegerType, FloatType, DoubleType, StringType, BooleanType, StructType
from pyspark.sql.functions import *
```

DATA PREPARATION

```
data_food = spark.read.load('/FileStore/tables/onlinefoods.csv', sep=',', format="csv", header='True')
```

```
data_food.show(5)
```

```
+---+------+--------------+----------+--------------+--------------------------+-----------+--------+---------+------
--+------+---------+----+
|Age|Gender|Marital Status|Occupation|Monthly Income|Educational Qualifications|Family size|latitude|longitude|Pin co
de|Output| Feedback|_c12|
+---+------+--------------+----------+--------------+--------------------------+-----------+--------+---------+------
--+------+---------+----+
| 20|Female|        Single|   Student|     No Income|             Post Graduate|          4| 12.9766|  77.5993|  5600
01|   Yes| Positive| Yes|
| 24|Female|        Single|   Student|Below Rs.10000|                  Graduate|          3|  12.977|  77.5773|  5600
09|   Yes| Positive| Yes|
| 22|  Male|        Single|   Student|Below Rs.10000|             Post Graduate|          3| 12.9551|  77.6593|  5600
17|   Yes|Negative | Yes|
| 22|Female|        Single|   Student|     No Income|                  Graduate|          6| 12.9473|  77.5616|  5600
19|   Yes| Positive| Yes|
| 22|  Male|        Single|   Student|Below Rs.10000|             Post Graduate|          4|  12.985|  77.5533|  5600
10|   Yes| Positive| Yes|
+---+------+--------------+----------+--------------+--------------------------+-----------+--------+---------+------
--+------+---------+----+
only showing top 5 rows
```

```
data_food.printSchema()
```

```
root
 |-- Age: string (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Marital Status: string (nullable = true)
 |-- Occupation: string (nullable = true)
 |-- Monthly Income: string (nullable = true)
 |-- Educational Qualifications: string (nullable = true)
 |-- Family size: string (nullable = true)
 |-- latitude: string (nullable = true)
 |-- longitude: string (nullable = true)
 |-- Pin code: string (nullable = true)
 |-- Output: string (nullable = true)
 |-- Feedback: string (nullable = true)
 |-- _c12: string (nullable = true)
```

```
data_food.describe().toPandas()
```

	summary	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	
0	count	388	388	388	388	388	388	
1	mean	24.628865979381445	None	None	None	None	None	3.2809
2	stddev	2.975592660672904	None	None	None	None	None	1.3510
3	min	18	Female	Married	Employee	10001 to 25000	Graduate	
4	max	33	Male	Single	Student	No Income	Uneducated	

Valeurs Manquantes

```
data_food.select([count(when(isnan(c), c)).alias(c) for c in data_food.columns]).show()
```

```
+---+------+--------------+----------+--------------+--------------------------+-----------+--------+---------+------
--+------+--------+----+
|Age|Gender|Marital Status|Occupation|Monthly Income|Educational Qualifications|Family size|latitude|longitude|Pin co
de|Output|Feedback|_c12|
+---+------+--------------+----------+--------------+--------------------------+-----------+--------+---------+------
--+------+--------+----+
|  0|     0|             0|         0|             0|                         0|          0|       0|        0|
0|     0|       0|   0|
+---+------+--------------+----------+--------------+--------------------------+-----------+--------+---------+------
--+------+--------+----+
```

Duplication

Nombre de lignes dupliquées

```
duplicates_count = data_food.groupBy(data_food.columns).count()
total_duplicates = duplicates_count.filter(col("count") > 1).agg({"count": "sum"}).collect()[0][0]
print(total_duplicates)
```

```
169
```

Affichage les lignes duppliquées

```
duplicate_rows = duplicates_count.filter(col("count") > 1)
duplicate_data = duplicate_rows.join(data_food, data_food.columns, "inner")
duplicate_data.show()
```

```
--------+------+---------+----+-----+
| 23|  Male|        Single|       Student| Below Rs.10000|            Post Graduate|          2|  12.977|  77.5773|
560009|   Yes|Negative | Yes|    2|
| 23|  Male|        Single|       Student| Below Rs.10000|            Post Graduate|          2|  12.977|  77.5773|
560009|   Yes|Negative | Yes|    2|
| 25|  Male|        Single|       Student|      No Income|            Post Graduate|          1| 12.9343|  77.6044|
560029|   Yes| Positive| Yes|    2|
| 25|  Male|        Single|       Student|      No Income|            Post Graduate|          1| 12.9343|  77.6044|
560029|   Yes| Positive| Yes|    2|
| 24|Female|        Single|      Employee| 10001 to 25000|            Post Graduate|          4| 12.9337|    77.59|
560011|   Yes| Positive| Yes|    2|
| 24|Female|        Single|      Employee| 10001 to 25000|            Post Graduate|          4| 12.9337|    77.59|
560011|   Yes| Positive| Yes|    2|
| 23|Female|        Single|       Student|      No Income|            Post Graduate|          3| 12.9369|  77.6407|
560095|    No| Positive|  No|    3|
| 23|Female|        Single|       Student|      No Income|            Post Graduate|          3| 12.9369|  77.6407|
560095|    No| Positive|  No|    3|
| 23|Female|        Single|       Student|      No Income|            Post Graduate|          3| 12.9369|  77.6407|
560095|    No| Positive|  No|    3|
| 32|  Male|       Married|      Employee|More than 50000|            Post Graduate|          6| 12.9369|  77.6407|
560095|   Yes| Positive| Yes|    2|
```

```
data_food= data_food.dropDuplicates()
```

```
duplicates_count = data_food.groupBy(data_food.columns).count()
total_duplicates = duplicates_count.filter(col("count") > 1).agg({"count": "sum"}).collect()[0][0]
print(total_duplicates)
```

```
None
```

Valeurs uniques de chaque colonne

```
[(col_name, data_food.agg(approx_count_distinct(col(col_name))).collect()[0][0]) for col_name in data_food.columns]
```

```
Out[28]: [('Age', 16),
 ('Gender', 2),
 ('Marital Status', 3),
 ('Occupation', 4),
 ('Monthly Income', 5),
 ('Educational Qualifications', 5),
 ('Family size', 6),
 ('latitude', 78),
 ('longitude', 75),
 ('Pin code', 80),
 ('Output', 2),
 ('Feedback', 2),
 ('_c12', 2)]
```

Les valeurs unique de chaque category

```
colonnes = ["Gender", "Marital Status", "Occupation", "Monthly Income", "Educational Qualifications", "Family size",

for colonne in colonnes:
    print(f"Valeurs uniques pour la colonne '{colonne}':")
    unique_values = data_food.select(col(colonne)).distinct().rdd.map(lambda row: row[0]).collect()
    for valeur in unique_values:
        print(valeur)
```

```
School
Uneducated
Post Graduate
Graduate
Ph.D
Valeurs uniques pour la colonne 'Family size':
3
5
6
1
4
2
Valeurs uniques pour la colonne 'Output':
No
Yes
Valeurs uniques pour la colonne 'Feedback':
Negative
Positive
Valeurs uniques pour la colonne '_C12':
No
Yes
```

Convert Types

```
data_food = data_food.withColumn("Age", col("Age").cast("integer")) \
    .withColumn("Family size", col("Family size").cast("integer")) \
    .withColumn("Pin code", col("Pin code").cast("integer"))
```

```
data_food.printSchema()
```

```
root
 |-- Age: integer (nullable = true)
```

```
 |-- Gender: string (nullable = true)
 |-- Marital Status: string (nullable = true)
 |-- Occupation: string (nullable = true)
 |-- Monthly Income: string (nullable = true)
 |-- Educational Qualifications: string (nullable = true)
 |-- Family size: integer (nullable = true)
 |-- latitude: string (nullable = true)
 |-- longitude: string (nullable = true)
 |-- Pin code: integer (nullable = true)
 |-- Output: string (nullable = true)
 |-- Feedback: string (nullable = true)
 |-- _c12: string (nullable = true)
```

```
data_food = data_food.withColumn("latitude", col("latitude").cast("float")) \
    .withColumn("longitude", col("longitude").cast("float")) \
    .withColumn("_c12", col("_c12").cast("integer"))
```

```
data_food.printSchema()
```

```
root
 |-- Age: integer (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Marital Status: string (nullable = true)
 |-- Occupation: string (nullable = true)
 |-- Monthly Income: string (nullable = true)
 |-- Educational Qualifications: string (nullable = true)
 |-- Family size: integer (nullable = true)
 |-- latitude: float (nullable = true)
 |-- longitude: float (nullable = true)
 |-- Pin code: integer (nullable = true)
 |-- Output: string (nullable = true)
 |-- Feedback: string (nullable = true)
 |-- _c12: integer (nullable = true)
```

Data Analysis

```
data_food.createOrReplaceTempView("food")
```

```
%sql Select * FROM food
```

Table

	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	Family s
1	25	Male	Married	Employee	More than 50000	Post Graduate	6
2	20	Male	Single	Student	No Income	Graduate	2
3	28	Male	Married	Self Employeed	10001 to 25000	Graduate	2
4	23	Female	Single	Student	No Income	Post Graduate	2
5	22	Male	Single	Student	Below Rs.10000	Post Graduate	4
6	23	Male	Single	Student	No Income	Graduate	4

285 rows

Les occupations les plus courantes

```
%sql SELECT Occupation, COUNT(*) AS count FROM food
GROUP BY Occupation
ORDER BY count DESC
```

Table

	Occupation	count

1	Student	144
2	Employee	94
3	Self Employeed	38
4	House wife	9

4 rows

Nombre de female et male

```
%sql SELECT Gender, COUNT(*) AS count FROM food
GROUP BY Gender
ORDER BY count DESC
```

Table

	Gender	count
1	Male	164
2	Female	121

2 rows

Untitled

```
%sql  SELECT Gender, `Monthly Income`,COUNT(*) AS count FROM food
GROUP BY Gender,`Monthly Income`
ORDER BY Gender,`Monthly Income`
```

Table

	Gender	Monthly Income	count
1	Female	10001 to 25000	16
2	Female	25001 to 50000	28
3	Female	Below Rs.10000	7
4	Female	More than 50000	11
5	Female	No Income	59
6	Male	10001 to 25000	20
7	Male	25001 to 50000	24
8	Male	Below Rs.10000	12
9	Male	More than 50000	36
10	Male	No Income	72

10 rows

```
%sql  SELECT Gender, `Marital Status`,COUNT(*) AS count FROM food
GROUP BY Gender,`Marital Status`
ORDER BY Gender,`Marital Status`
```

Table

	Gender	Marital Status	count
1	Female	Married	42
2	Female	Prefer not to say	5
3	Female	Single	74
4	Male	Married	45
5	Male	Prefer not to say	4
6	Male	Single	115

6 rows

```
%sql  SELECT Gender, `Occupation`,COUNT(*) AS count FROM food
GROUP BY Gender,`Occupation`
ORDER BY Gender,`Occupation`
```

Table

	Gender	Occupation	count
1	Female	Employee	41
2	Female	House wife	9
3	Female	Self Employeed	12
4	Female	Student	59
5	Male	Employee	53
6	Male	Self Employeed	26
7	Male	Student	85

7 rows

```
%sql  SELECT Gender, `Educational Qualifications`,COUNT(*) AS count FROM food
GROUP BY Gender,`Educational Qualifications`
ORDER BY Gender,`Educational Qualifications`
```

Table

	Gender	Educational Qualifications	count
1	Female	Graduate	52
2	Female	Ph.D	9
3	Female	Post Graduate	54
4	Female	School	5
5	Female	Uneducated	1
6	Male	Graduate	74
7	Male	Ph.D	12
8	Male	Post Graduate	71
9	Male	School	6
10	Male	Uneducated	1

10 rows

```
%sql  SELECT Occupation, `Monthly Income`,COUNT(*) AS count FROM food
GROUP BY Occupation,`Monthly Income`
ORDER BY Occupation,`Monthly Income`
```

Table

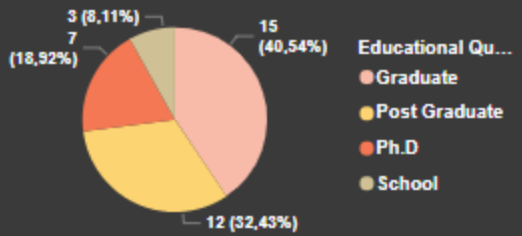
	Occupation	Monthly Income	count
1	Employee	10001 to 25000	21
2	Employee	25001 to 50000	38
3	Employee	Below Rs.10000	6
4	Employee	More than 50000	29
5	House wife	No Income	9
6	Self Employeed	10001 to 25000	10
7	Self Employeed	25001 to 50000	11
8	Self Employeed	More than 50000	17
9	Student	10001 to 25000	5
10	Student	25001 to 50000	3
11	Student	Below Rs.10000	13
12	Student	More than 50000	1
13	Student	No Income	122

13 rows

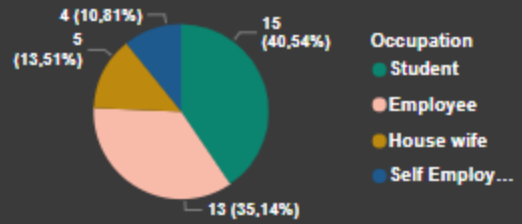
Save modifications in new file

Customer Overview : Order Analysis

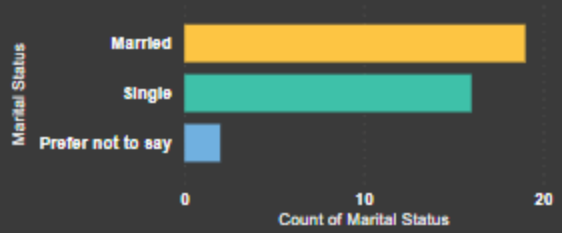
Educational Qualification



Occupation



Marital Status



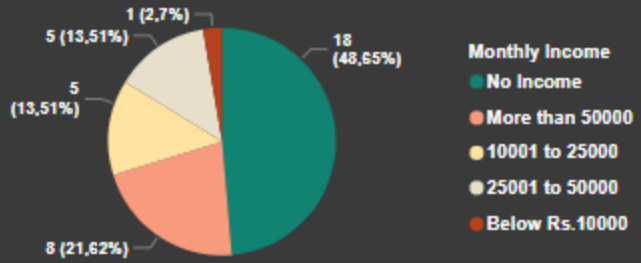
Interval Age

18 33

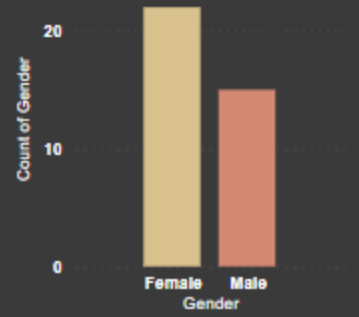
Family size

1	4
2	5
3	6

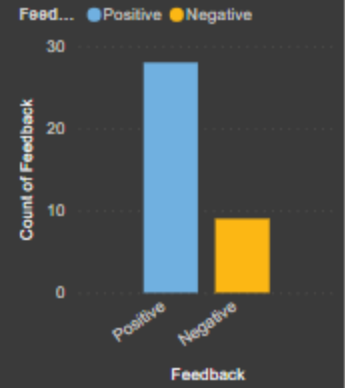
Monthly Income



Gender



Feedbacks



Filters