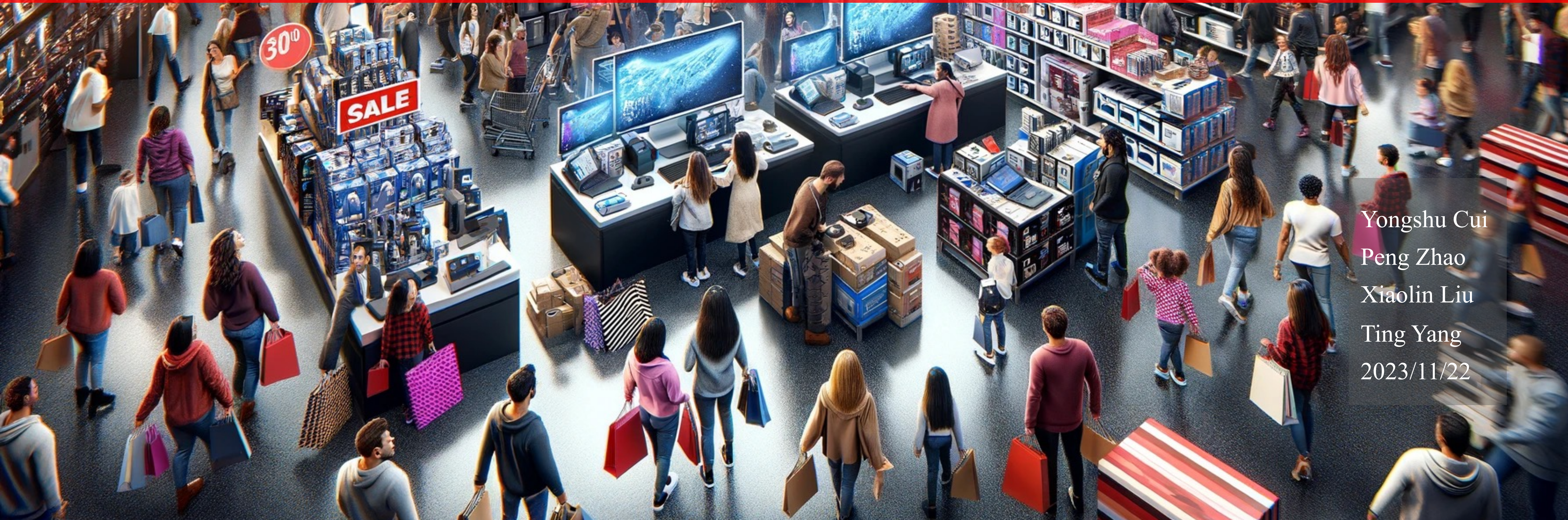
Black Friday Dataset Cleaning Project

Improve data quality and promote accurate analysis



Yongshu Cui
Peng Zhao
Xiaolin Liu
Ting Yang
2023/11/22

Introduction

Introduction

- Black Friday is the beginning of the Christmas shopping season in the United States. Major retailers like Amazon and Costco offer discounts and deals on various products to attract customers. We selected the Black Friday dataset as the data source for our curation project. This dataset provides interesting insights into customer behavior during shopping holidays.

Purpose

- In this project, we focus on a key task: improving the quality of the Black Friday dataset. Through precise cleaning and processing of these data, a more accurate and reliable data foundation is provided. To predict customer spending during Black Friday sales. These predictions can help retailers understand and tailor their products to meet customer preferences.

Dataset Schema

Sample Data

User_ID (int)	Product_ID (string)	Gender (string)	Age (string)	Occupation (short)	City_Category (string)	Stay_In_Current_City_Years (short)	Marital_Status (boolean)	Product_Category_1 (short)	Product_Category_2 (short)	Product_Category_3 (short)	Purchase (short)
1000001	P00069042	F	0-17	10	A	2	false	3			8370
1000001	P00248942	F	0-17	10	A	2	false	1	6	14	15200
1000001	P00087842	F	0-17	10	A	2	false	12			1422
1000001	P00085442	F	0-17	10	A	2	false	12	14		1057

Here is the data schema of the Black Friday dataset

***Data source:** *https://drive.google.com/file/d/1kS25NE46YLJE2GH4yPFaYRP_PxzLngu-/view?pli=1*

Dataset Overview

Columns_name	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
Total_values	550068	550068	550068	550068	550068	550068	550068	550068	550068	550068	550068	550068
Not_null_values	550068	550068	550068	550068	550068	550068	550068	550068	550068	376430	166821	550068
Null_values	0	0	0	0	0	0	0	0	0	173638	383247	0
Unique_values	5891	3631	4	7	21	3	5	2	20	17	15	18105
Maximum	1006040				20				20	18	18	
Minimum	1000001				0				1	2	3	
Mean	1.00E+06				8.08E+00				5.40E+00	9.84E+00	1.27E+01	
Sum	5.52E+11				4.44E+06				2.97E+06	3.70E+06	2.11E+06	
Std_deviation	1727.591586				6.52266				3.936211	5.08659	4.125338	

There are 12 columns and 550068 rows in total , as shown above.

Issues Discovery

Data Inconsistency

```
select gender, count(1) as counts  from black_friday_sales group by gender order by counts desc
```

Console Timing Datasets Charts

temporary_dataset (4 rows)

Views

	gender (string)	counts (long)
0	M	289981
1	Male	124278
2	F	67905
3	Female	67904

Check the distinct values for each column, we find the values in Gender are not consistent, since there are "M" and "Male" for male, "F" and "Female" for female. It might lead to incorrect market targeting, inaccurate user profiling, and decisions based on unreliable gender-related strategies.

Issues Discovery

```
-- check the record counts for gender when grouping by user_id

select  user_id, size(collect_set(gender)) as  gender_set
from black_friday_sales
group by user_id
```

Console Timing Datasets Charts

temporary_dataset (5891 rows)

Views

	user_id (int)	gender_set (int)
0	1000149	2
1	1000190	2
2	1000636	2
3	1001043	2
4	1001129	2
5	1001139	2
6	1001601	2
7	1002431	2
8	1002605	2
9	1003031	2
10	1003373	2
11	1003938	2
12	1004021	2
13	1004552	2
14	1004666	2
15	1004739	2
16	1005158	2
17	1005476	2
18	1005697	2
19	1005853	2

Inconsistent Representations

Group by the userid and count the number of genders, we find that there are 2 genders for one user, which is unreasonable.

Issues Discovery

```
-- details for grouping by user_id having more than 1 gender
select user_id, gender from black_friday_sales group by user_id, gender order by user_id
```

Console | Timing | Datasets | Charts

temporary_dataset (11777 rows)

Views

	user_id (int)	gender (string)
0	1000001	Female
1	1000001	F
2	1000002	Male
3	1000002	M
4	1000003	M
5	1000003	Male
6	1000004	M
7	1000004	Male
8	1000005	Male
9	1000005	M
10	1000006	F
11	1000006	Female
12	1000007	M
13	1000007	Male
14	1000008	M
15	1000008	Male
16	1000009	Male
17	1000009	M
18	1000010	Female
19	1000010	F

Inconsistent Representations

In the beginning, we thought it was a functional dependency issue, but looking into the details, we found the genders are the same, just used different represents. The solution will provide the analysis result.

Data Quality Issue for Incorrect Values

When checking column Stay_In_Current_City_Years, there is no NULL value, but we find there are a lot of values that are mistyped with short, for example result as below table.

```sql
-- check column Stay_In_Current_City_Years has null value or not
select count(*) from black_friday_sales where stay_in_current_city_years is null;
```

Console | Timing | Datasets | Charts

temporary_dataset (1 rows)

	count(1) (long)
0	0

```sql
-- check column Stay_In_Current_City_Years has illegal values or not
select user_id, product_id, gender, age, occupation, city_category, stay_in_current_city_years, marital_status
from black_friday_sales where CAST(stay_in_current_city_years as int) is null;
```

Console | Timing | Datasets | Charts

temporary_dataset (84726 rows)

	user_id (int)	product_id (string)	gender (string)	age (string)	occupation (short)	city_category (string)	stay_in_current_city_years (string)	marital_status (boolean)
0	1000002	P00285442	Male	55+	16	C	4+	false
1	1000008	P00249542	M	26-35	12	C	4+	true
2	1000008	P00220442	M	26-35	12	C	4+	true
3	1000008	P00156442	M	26-35	12	C	4+	true
4	1000008	P00213742	M	26-35	12	C	4+	true
5	1000008	P00214442	M	26-35	12	C	4+	true
6	1000008	P00303442	M	26-35	12	C	4+	true
7	1000010	P00085942	Female	36-45	1	B	4+	true
8	1000010	P00118742	Female	36-45	1	B	4+	true
9	1000010	P00297942	Female	36-45	1	B	4+	true
10	1000010	P00266842	Female	36-45	1	B	4+	true

Data Quality Issue for NULL Values

```sql
-- check columns with null values based on the overview table
SELECT
    concat(round(AVG(CASE WHEN product_category_2 IS NULL THEN 1 ELSE 0 END)*100, 2),'%') AS Missing_product_category_2_percentage,
    concat(round(AVG(CASE WHEN product_category_3 IS NULL THEN 1 ELSE 0 END)*100, 2),'%') AS Missing_product_category_3_percentage
FROM black_friday_sales;
```

Console Timing Datasets Charts

temporary_dataset (1 rows)

Views

	Missing_product_category_2_percentage (string)	Missing_product_category_3_percentage (string)
0	31.57%	69.67%

Based on the overview table, there are 2 columns with null values: Product_Category_2 and Product_Category_3

Calculating the missing rate for these two columns in the table,

According to the result, we'll use different solutions for these 2 columns in Solutions.

Data Quality Issue for Incorrect Values - Purchase

```sql
-- check column Purchase has null value or not
select count(*) from black_friday_sales where purchase is null;
```

Console | Timing | Datasets | Charts

temporary_dataset (1 rows)

Views

	count(1) (long)
0	0

```sql
-- check column Purchase has illegal values or not
select user_id, product_id, gender, age, occupation, stay_in_current_city_years, marital_status, purchase
from black_friday_sales where CAST(purchase as int) is null;
```

Console | Timing | Datasets | Charts

temporary_dataset (28026 rows)

Views

	user_id (int)	product_id (string)	gender (string)	age (string)	occupation (short)	stay_in_current_city_years (string)	marital_status (boolean)	purchase (string)
0	1000001	P00087842	F	0-17	10	2	false	$1422
1	1000001	P00085442	F	0-17	10	2	false	$1057
2	1000018	P00366542	F	18-25	3	3	false	$1780
3	1000018	P0094142	Female	18-25	3	3	false	$697
4	1000019	P00244842	M	0-17	10	3	false	$1539
5	1000023	P00112342	M	36-45	0	3	true	$584
6	1000026	P00043242	M	26-35	7	2	true	$1848
7	1000028	P00084442	Female	26-35	1	2	true	$758
8	1000033	P00219242	M	46-50	3	1	true	$811
9	1000044	P00115242	M	46-50	17	3	true	$1728
10	1000044	P00124842	M	46-50	17	3	true	$1582

Similar to column Stay_In_Current_City_Years, there is no NULL value in column Purchase, and we also find there are values that are mistyped with short, for example result as figure show.

Data Field Normalization

Based on the overview table, we know that the column Age does not have a NULL value, so it appears that there is nothing wrong with that column that needs to be fixed. But when looking at the detailed data, we see that the raw data uses ranges to represent the user's age, as shown on the right.

```sql
-- check column Age has illegal values or not
select user_id, age
from black_friday_sales where CAST(age as string) is null;
```

Console Timing Datasets Charts

temporary_dataset (0 rows)

	user_id (int)	age (string)

```sql
-- check the distinct values of column Age
select distinct age from black_friday_sales
```

Console Timing Datasets Charts

temporary_dataset (7 rows)

	age (string)
0	18-25
1	26-35
2	0-17
3	46-50
4	51-55
5	36-45
6	55+

Data Field Normalization

Original	Converted
0-17	child
18-25	teenage
26-35	adult
36-45	adult
46-50	adult
51-55	adult
55+	old

When age fields are used for analysis or model training, using descriptive labels rather than discrete numeric ranges can provide more meaningful results.

We plan to use a mapping table to convert the age values.

Solution for Data Inconsistency

```sql
-- issue 1 redundant value in column gender
-- solution
select
User_id,
Product_ID,
case when Gender='Female' then 'F'
     when Gender='Male' then 'M'
     else Gender  end as Gender
,
Age,
occupation,
City_Category,
Stay_IN_Current_City_Years,
Marital_Status,
Product_Category_1,
Product_Category_2,
Product_Category_3,
Purchase
from
black_friday_sales
```

Output Dataset solve_issue1

- This issue involve 192182 lines, about 34.93% of the dataset
- To solve it, we decide to use replacement to union the values in column gender, details refer to SQL Female to F Male to M

Solution for Inconsistent Representations

- Total 84726 incorrect values found, about 15.4% of the dataset

- Solution: to clear the different gender value for one user, we should try to use the same word to display gender. And when we fixed issue 1 above by replacing "Female" with "F" and "Male" with "M", we find this issue also be resolved. Check the result dataset of issue 1, we get the below result, which one user only have one gender representation.

```sql
select  user_id, size(collect_set(gender)) as  gender_set
from solve_issue1
group by user_id
```

Console | Timing | Datasets | Charts

temporary_dataset (5891 rows)

Views

	user_id (int)	gender_set (int)
0	1000149	1
1	1000190	1
2	1000636	1
3	1001043	1
4	1001129	1
5	1001139	1
6	1001601	1
7	1002431	1
8	1002605	1
9	1003031	1
10	1003373	1

Solution for Incorrect values

```
-- check the percentage for each city_category base on all invalid values

select city_category,
concat(round((count(*) / 84726) * 100, 4), '%') as percentage
from solve_issue1
where CAST(stay_in_current_city_years as int) is null
group by city_category
```

Console Timing Datasets Charts

temporary_dataset (3 rows)

Views

	city_category (string)	percentage (string)
0	B	40.8493%
1	C	32.8081%
2	A	26.3426%

Stay_In_Current_City_Years

- This issue involves 84726 lines, about 15.4% of the dataset.
- By checking the percentage for each City_Category based on the invalid values in Stay_In_City_Years, we find when City_Category is ‘B’, it might have a bigger probabiltiy to stay in currenty city for longer time.

Solution for Incorrect values

The solution we decide to base on below table, setting the value in final_adding_year to the Stay_In_Current_City_Years of condition in age_range and city_category.

age_range	city_category	base_year	year_buffer_added	final_adding_year
0-17	A	4	1	5
0-17	B	4	5	9
0-17	C	4	10	14
18-55	A	14	1	15
18-55	B	14	10	24
18-55	C	14	5	19
55+	A	30	5	35
55+	B	30	3	33
55+	C	30	10	40

```
-- issue 3 data quality issue for incorrect values in Stay_In_Current_City_Years
-- solution: details can be found in report, this solution is based on the dataset only, not considering possible expand NULL values

select
User_id,
Product_ID,
Gender,
Age,
occupation,
City_Category,
case when age = '0-17' and city_category = 'A' then 5
    when age = '0-17' and city_category = 'B' then 9
    when age = '0-17' and city_category = 'C' then 14
    when age in ('18-25', '26-35', '36-45', '46-50', '51-55') and city_category = 'A' then 15
    when age in ('18-25', '26-35', '36-45', '46-50', '51-55') and city_category = 'B' then 24
    when age in ('18-25', '26-35', '36-45', '46-50', '51-55') and city_category = 'C' then 19
    when age = '55+' and city_category = 'A' then 35
    when age = '55+' and city_category = 'B' then 33
    when age = '55+' and city_category = 'C' then 40
end as Stay_IN_Current_City_Years,
Marital_Status,
Product_Category_1,
Product_Category_2,
Product_Category_3,
Purchase
from
solve_issue1
```

Console Timing Datasets Charts

solve_issue2 (**550068** rows)

Views

	User_id (int)	Product_ID (string)	Gender (string)	Age (string)	occupation (short)	City_Category (string)	Stay_IN_Current_City_Years (int)	Marital_Status (boolean)
0	1000001	P00069042	F	0-17	10	A	5	false
1	1000001	P00248942	F	0-17	10	A	5	false
2	1000001	P00087842	F	0-17	10	A	5	false
3	1000001	P00085442	F	0-17	10	A	5	false
4	1000002	P00285442	M	55+	16	C	40	false
5	1000003	P00193542	M	26-35	15	A	15	false
6	1000004	P00184942	M	46-50	7	B	24	true
7	1000004	P00346142	M	46-50	7	B	24	true

Solution for Null Values

This issue involves 2 columns:

- Product_Category_2: 173,638 NULL values total, about 31.57% of the dataset.
- Product_Category_3: 383,247 NULL values total, about 69.67% of the dateset.

Solution :
For Product_Category_2: using the nearest value of the mean of the product category 2 group by product_category_1 to fill the null value of product_category_2.
For Product_Category_3 : there are almost 70% percent missing, and no good dependency function can be found, we decide to drop this column.

```
--  issue 3 null value in Product_Category_2
-- solution: using the nearest value of the mean of the product category 2 group by product_category_1 to fill the nul
select
product_category_1,
ceiling(avg(product_category_2)) as avg_2
from
solve_issue2
group by product_category_1
order by product_category_1
```

Output Dataset Output Dataset (optional)

Show Code Examples Change Command Dismiss Submit

Console Timing Datasets Charts

temporary_dataset (20 rows)

Views

	product_category_1 (short)	avg_2 (long)
0	1	8
1	2	7
2	3	5
3	4	6
4	5	11
5	6	11
6	7	13
7	8	15
8	9	15
9	10	15
10	11	16

```
--  issue 3 null value in Product_Category_2
-- solution: using the nearest value of the mean of the product category 2 group by product_category_1 to fill the nul
select
User_id,
Product_ID,
Gender,
Age,
occupation,
City_Category,
Stay_IN_Current_City_Years,
Marital_Status,
Product_Category_1,
case when product_category_1 = 1 and product_category_2 is null then 8
     when product_category_1 = 2 and product_category_2 is null then 7
     when product_category_1 = 3 and product_category_2 is null then 5
     when product_category_1 = 4 and product_category_2 is null then 6
     when product_category_1 = 5 and product_category_2 is null then 11
     when product_category_1 = 6 and product_category_2 is null then 11
     when product_category_1 = 7 and product_category_2 is null then 13
     when product_category_1 = 8 and product_category_2 is null then 15
     when product_category_1 = 9 and product_category_2 is null then 15
     when product_category_1 = 10 and product_category_2 is null then 15
     when product_category_1 = 11 and product_category_2 is null then 15
     when product_category_1 = 12 and product_category_2 is null then 15
     when product_category_1 = 13 and product_category_2 is null then 16
     when product_category_1 = 14 and product_category_2 is null then 17
     when product_category_1 = 15 and product_category_2 is null then 17
     else 5 end as Product_Category_2,
Product_Category_3,
Purchase
from
solve_issue2
```

Solution for Incorrect Values - Purchase

```
-- issue 6 wrong value in column purchase
-- solution: remove the $ in value and cast all the value to short
select
User_id,
Product_ID,
Gender,
Age,
occupation,
City_Category,
Stay_IN_Current_City_Years,
Marital_Status,
Product_Category_1,
Product_Category_2,
Product_Category_3,
cast(if(cast(purchase as string) rlike '$',replace(cast(purchase as string),"$",''),cast(purchase as string)) as int) as pu
from
solve_issue3
```

- ✓ Some value with $, affected 28,026 lines, 5.9% of the dataset.
- ✓ Solution: we think the number for the incorrect lines is correct, we just need to remove the "$" symbol and cast the values to short.

Solution for data field normalization

```
--  issue 5 data type or value in Age needs to be changed
--  solution: mapping
--      0-17 to child
---     18-25 to teenage
--      26-55 to adult
--      55+ to old
select
User_id,
Product_ID,
Gender,
case when Age='0-17' then 'child'
     when Age='18-25' then 'teenager'
     when Age='55+' then 'old'
     else 'adult' end as Age
,
occupation,
City_Category,
Stay_IN_Current_City_Years,
Marital_Status,
Product_Category_1,
Product_Category_2,
-- Product_Category_3,
cast(if(cast(purchase as string) rlike '$',replace(cast(purchase as string),"$",''),cast(purchase as string)) as int) a
from
solve_issue3
```

Solution: use the mapping table we introduced in last section, we update the dataset

Validation

Check the overview of the updated dataset, the NULL values not exist any more, and Product_Category_3 column has been removed. Let's check the result of other 5 issues.

Column_name	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_IN_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Purchase
Total_values	550068	550068	550068	550068	550068	550068	550068	550068	550068	550068	550068
Not_null_values	550068	550068	550068	550068	550068	550068	550068	550068	550068	550068	550068
Null_values	0	0	0	0	0	0	0	0	0	0	0
Unique_values	5891	3631	2	4	21	3	9	2	20	9	18105
Maximum	1006040				20		40		20	17	23961
Minimum	1000001				0		5		1	5	12
Mean	1.00E+06				8.08E+00		20.46977		5.40E+00	7.26E+00	9.26E+03
Sum	5.52E+11				4.44E+06		11259760		2.97E+06	4.00E+06	5.10E+09
Std_deviation	1727.592				6.52266		5.285947		3.936211	3.806973	5023.065

Validation

```sql
-- validation for issue 1, inconsistent value of gender
select distinct gender from final_dataset
```

Console Timing Datasets Charts

temporary_dataset (2 rows)

Views

	gender (string)
0	F
1	M

Data Inconsistency

We solved the data inconsistency in the Gender column by using the mapping relationship as Female -> F, Male -> M.

Validation

Inconsistent representations

The result shows that there is one gender for one user.

```
-- validation for issue 2, inconsistent representation
select  user_id, size(collect_set(gender)) as  gender_set
from final_dataset
group by user_id
```

Console Timing Datasets Charts

temporary_dataset (5891 rows)

Views

	user_id (int)	gender_set (int)
0	1000149	1
1	1000190	1
2	1000636	1
3	1001043	1
4	1001129	1
5	1001139	1
6	1001601	1
7	1002431	1
8	1002605	1
9	1003031	1
10	1003373	1
11	1003938	1
12	1004021	1
13	1004552	1
14	1004666	1
15	1004739	1
16	1005158	1
17	1005476	1
18	1005697	1
19	1005853	1

Validation

```
-- validation for issue 3, data quality issue for incorrect values in Stay_In_Current_City_Years
select user_id, product_id, gender, age, occupation, city_category, stay_in_current_city_years, marital_status
from final_dataset where CAST(stay_in_current_city_years as int) is null;
```

Console | Timing | Datasets | Charts

temporary_dataset (0 rows)

Views

user_id (int)	product_id (string)	gender (string)	age (string)	occupation (short)	city_category (string)	stay_in_current_city_years (int)	marital_status (boolean)

Data quality issue for incorrect values - Stay_In_Current_City_Years

No more incorrect values with '+' as string in Stay_In_Current_City_Years

Validation

```sql
-- validation for issue 5, data quality issue for incorrect vaules in Purchase
select user_id, product_id, gender, age, occupation, stay_in_current_city_years, marital_status, purchase
from final_dataset where CAST(purchase as int) is null;
```

Console Timing Datasets Charts

temporary_dataset (**0** rows)

Views

	user_id (int)	product_id (string)	gender (string)	age (string)	occupation (short)	stay_in_current_city_years (int)	marital_status (boolean)	purchase (int)

Data quality issue for incorrect values – Purchase

No more incorrect values with '$' as string in Purchase

Validation

```sql
-- validation for issue 6, data field normalization in age
select distinct age from final_dataset
```

Console | Timing | Datasets | Charts

temporary_dataset (4 rows)

Views

	age (string)
0	old
1	adult
2	teenager
3	child

Data field normalization

The values in age column are no more age range, now use descriptive labels

Summary

Initial Issues: Data inconsistencies, erroneous entries, and missing values.

Curation Techniques:

- Mapping for uniform data representation.
- Replacing incorrect values.
- Filling in missing data for completeness.

Result: A cleansed dataset ready for machine learning modeling.

Impact: Provides a solid foundation for accurate analysis and predictive modeling of Black Friday sales trends.

Thank you very much for watching !