



Insights Analysis



Kristina Kaliagina



Agenda

- **Producing Insights for Clients:**
 - Problem Statement
 - Description of Data
 - Metrics
 - Insights of client
 - **SQL query:**
 - SQL query
 - Steps of SQL query
 - Results of SQL query
-

Producing Insights for Clients

Executive Summary: we have the consumer transaction data across a range of European countries which comprises consumer transaction information at a granular transaction level, on the basis of which it is necessary to conduct an exploratory analysis

Problem statement:

- to produce insights for Lidl;
- to add additional metrics/insights;
- to give examples of any additional fields.

Description of Data

	TxnDateTime	spendout	UserKey	MerchantTag	UserPostalSector
0	2022-01-30	-18.24	207213E18150	asda	G781
1	2022-01-31	-1.00	5C4B04A9F05D	asda	LE36
2	2022-01-31	-68.95	A3610CF3A565	asda	OL14
3	2022-01-31	-45.12	F372B40275E8	asda	NE639
4	2022-01-27	-93.34	EA139F1D3902	asda	EH546
...
2675093	2022-03-26	-29.48	8372EC04277B	lidl	E126
2675094	2022-03-26	-8.13	2BA3F0A4FD5C	lidl	LU11
2675095	2022-03-27	-1.15	7B131BB412FD	lidl	M94
2675096	2022-03-24	-42.46	66E6F56BF544	lidl	E178
2675097	2022-03-23	-23.20	3F5B63C852D6	lidl	RM82

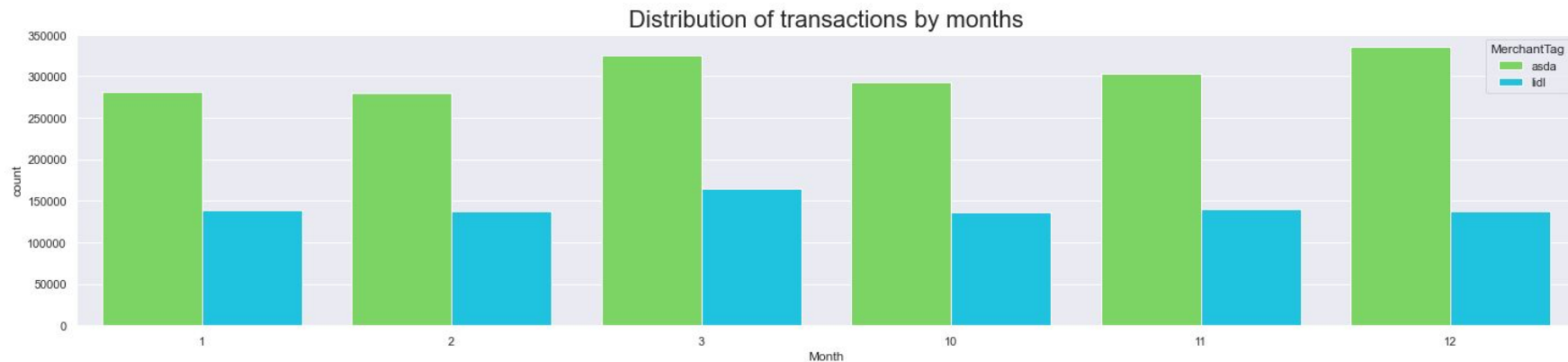
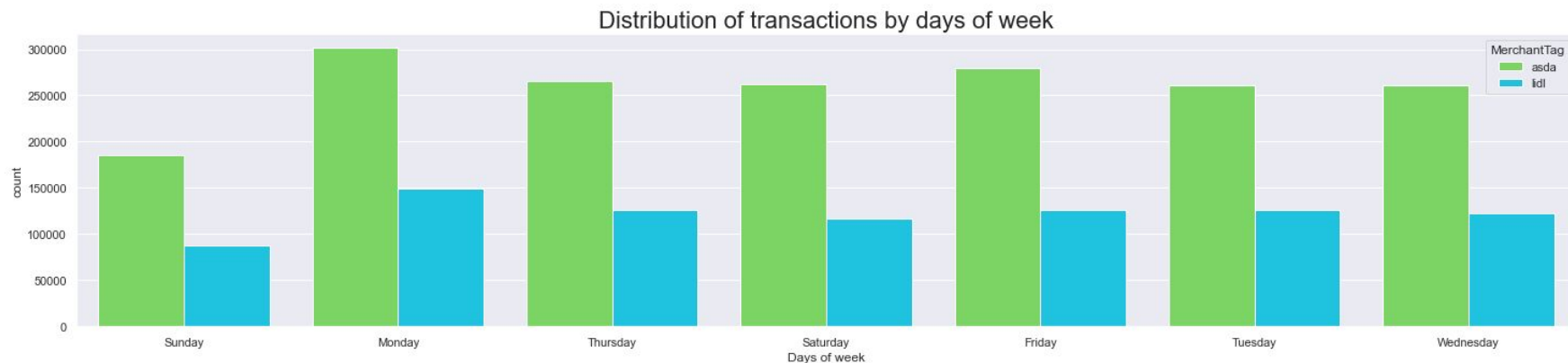
	DataRow
106 MB	Consumer transaction data
object	TxnDateTime
float64	spendout
object	UserKey
object	MerchantTag
object	UserPostalSector

Metric 1 - Monthly profit by companies

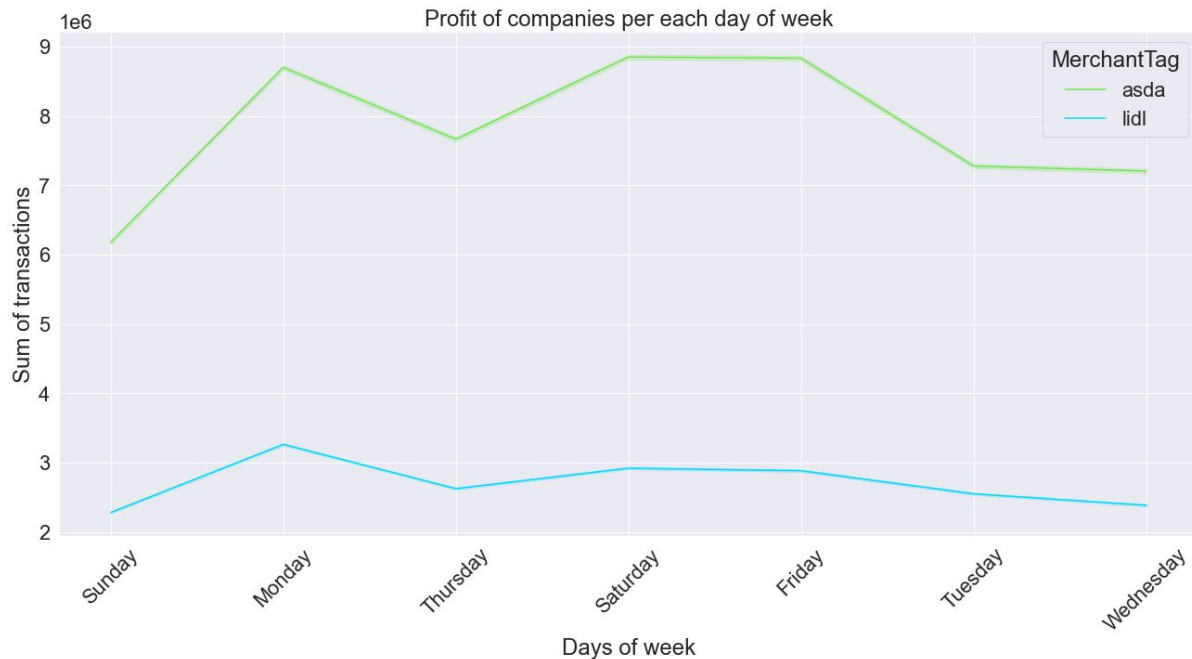


***Monthly profit = Sum of transactions per month**

Metric 2 - The number of transactions per day of the week and per month by company



Metric 3 - Profit per days of the week by company



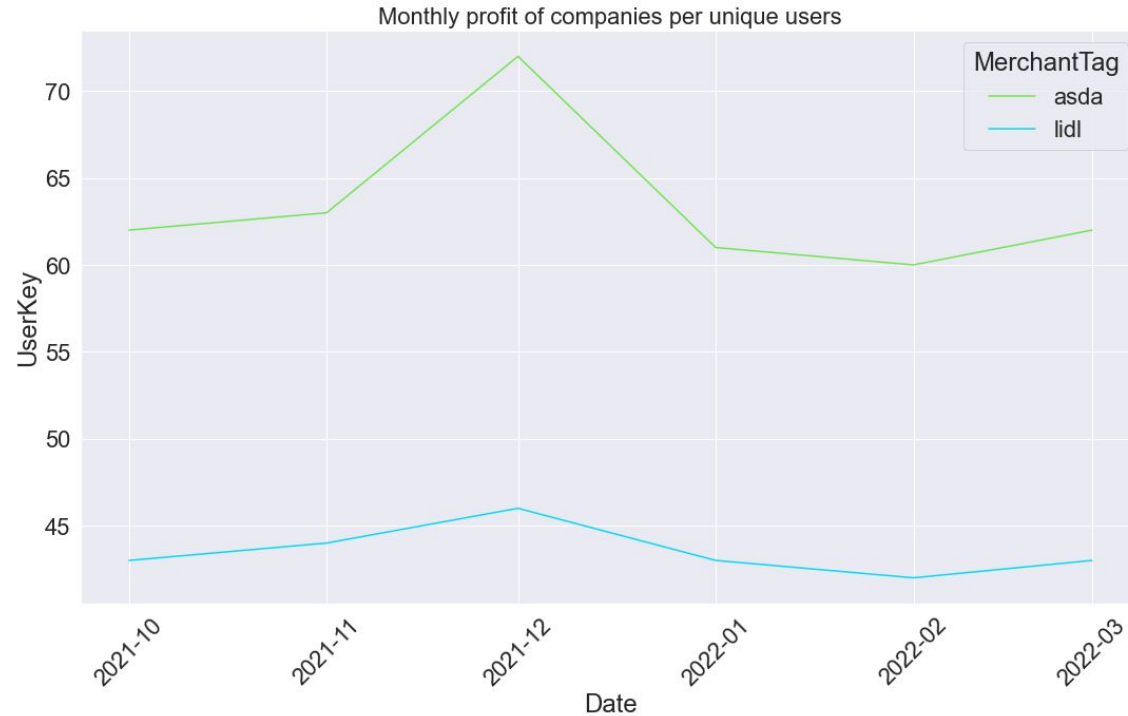
1. Trends in the growth and fall of profits for companies occur on the same days of the week

2. According to the results of 6 months, Asda increased its profit, Lidl remained at the same level.

Metric 4 - Daily and monthly average spendout by company



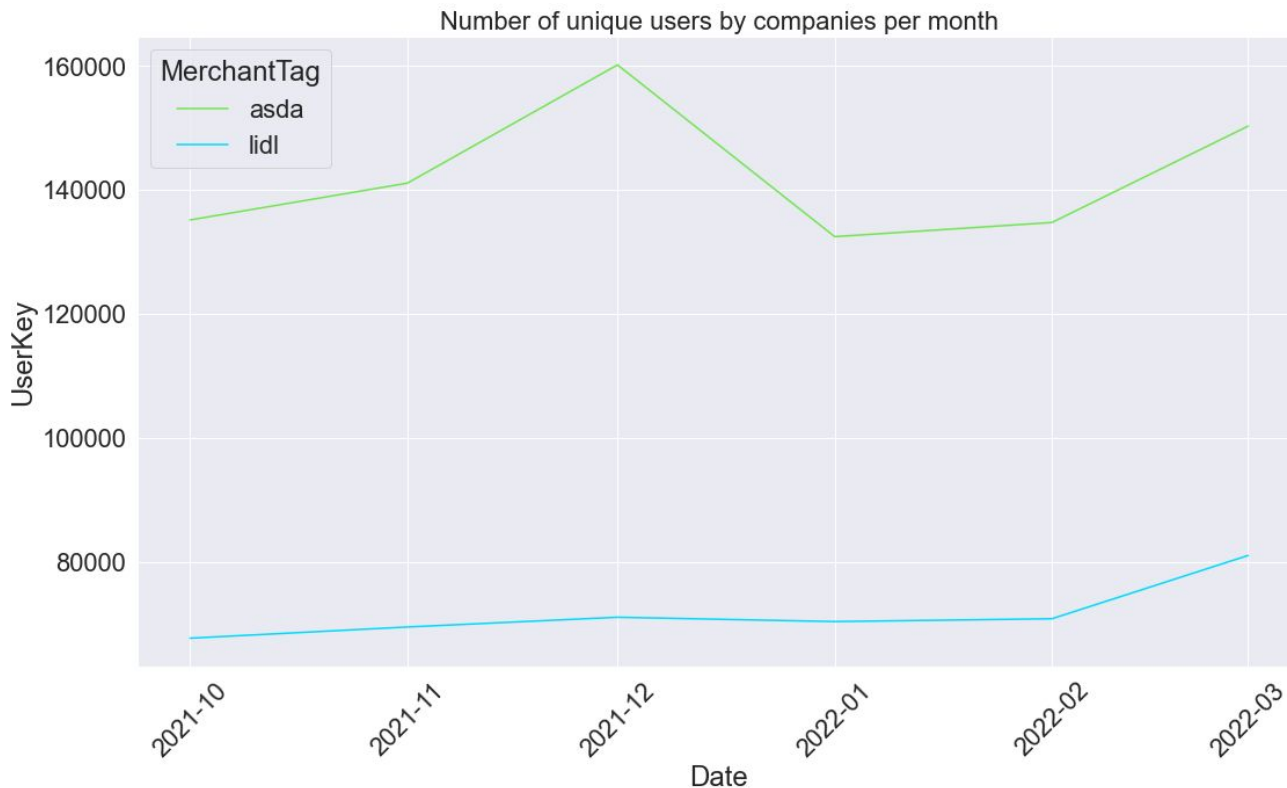
Metric 5 - Monthly profit per unique users by company



All profit per the month / number of unique users

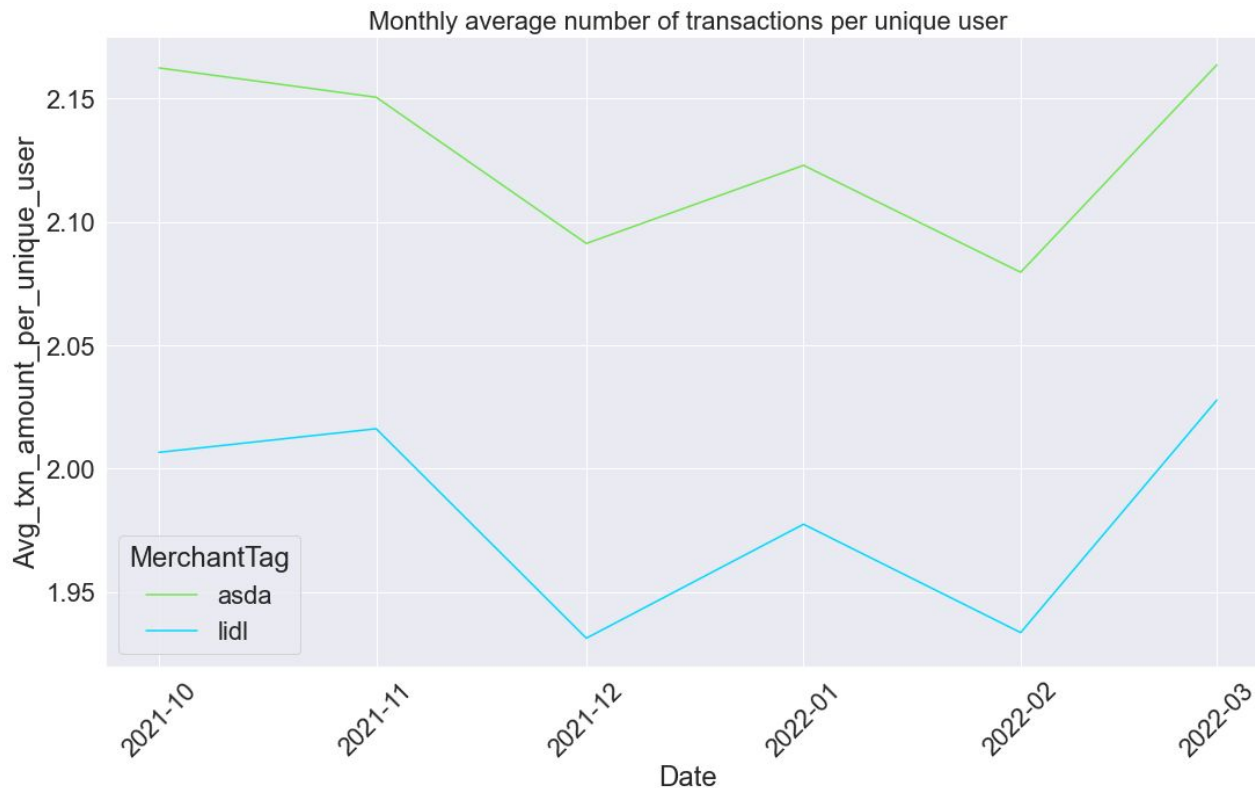
Metric 6 - The number of unique users per month by company

Asda has almost 2x more unique users



Metric 7 - Number of transactions per month \ number of unique users

Users return to a competitor more often than to Lidl



Insights of Lidl



1. The peak of profit falls on the winter holidays
2. Lidl is most in demand on Monday and Friday, and least on Sunday
3. The largest number of transactions was made in March 2022
4. Despite the growth of unique users, profit remained at the same level in 6 months
5. Lidl's average check in December is at its all-time high.
6. Competitor's wallets are higher than Lidl's.
7. lidl is aimed at people with lower earnings
8. Lidl has lower customer loyalty than its rival Asda.

Other metrics and fields of the full depth of data

Metrics/Insights:

- The share of transactions per month by location (online and in-store)
- The average daily *spendout* online and in-store
- Which locations customers tend to buy online or in-store
- The share of *SpendIn* in *spendout*
- The profit in each location
- The average *spendout* by consumer/business
- The average *spendout* by different age/group

Additional fields:

- *Txntype* - how client paid for purchase (cash or card) - it can help to estimate the need for staff and self-service cash desks
- *UserMaritalStatus* - single, married, married and have children - helps to customize and give special offers
- *SpendInReason* - why we refunded our client (bad quality, system error and etc) - to understand which area of business we can develop.

SQL query

SQL query

```
DECLARE @row int = 1
DECLARE @year int = null
DECLARE @quarter int = null

SELECT
    ROW_NUMBER() OVER( ORDER BY YEAR(date), DATEPART(qq, date)) AS Row,
    YEAR(date) AS Year,
    DATEPART(qq, date) AS Quarter
INTO
    #yearQuarterTable
FROM
    MainTable
WHERE
    YEAR(date) >= '2019'
GROUP BY
    YEAR(date), DATEPART(qq, date)

SELECT
    @year = Year,
    @quarter = Quarter
FROM
    #yearQuarterTable
WHERE
    [Row] = @row

CREATE TABLE #merch_comp (
    merchant_parent VARCHAR(150),
    Year INT NOT NULL,
    Quarter INT NOT NULL,
    Spend FLOAT ,
    Txns INT NOT NULL
)
```

```
WHILE @year IS NOT NULL
BEGIN
    CREATE TABLE #merch_users (
        customerid BINARY(32)
    )

    INSERT INTO
        #merch_users
    SELECT
        distinct a.customerid
    from
        MainTable a
    WHERE
        YEAR(txndatetime) = @year
        AND DATEPART(qq, date) = @quarter
        AND merchant_parent = 'b365'
        AND spend < 0
        AND country IN ('UK')

    INSERT INTO #merch_comp
    SELECT
        merchant_parent as Tags,
        YEAR(date) AS Year,
        DATEPART(qq,date) AS Quarter,
        ABS(SUM(spend)) AS Spend,
        COUNT(*) AS Txns
    from
        MainTable a
    WHERE
        YEAR(date) = @year
        AND DATEPART(qq,date) = @quarter
        AND spend < 0
        AND merchant_parent IN ('the national lottery', 'sky bet', 'paddypower', 'b365', 'william hill')
        AND country IN ('UK')
        AND customerid in (SELECT customerid
                           FROM #merch_users)
    GROUP BY YEAR(date), DATEPART(qq,date), merchant_parent

    DROP TABLE #merch_users

    --Fetch next year-quarter pair
    SET @year = NULL
    SET @quarter = NULL
    SET @row = @row + 1

    SELECT @year = Year, @quarter = Quarter
    FROM #yearQuarterTable
    WHERE [Row] = @row
END
```


Steps of SQL query

- Create integer variable and set initial values:

row=1

- Create Quarter, Year, Row number table from 2019:

Row	Year	Quarter
1	2019	1
2	2019	2
3	2019	3

- Set value of year and quarter variables:

year=2019

quarter=1

- Create an empty temporary table:

Merchant_ parent	Year	Quarter	Spend	Txns
---------------------	------	---------	-------	------

Steps of SQL query

- The WHILE loop
- Create a temporary table #merch_users:

customerid
1
2
...

- Populate the #merch_comp

Merchant_parent	Year	Quarter	Spend	Txns
b365	2019	1	1000	10
the national lottery	2019	1	2000	20
sky bet	2019	1	1500	15
paddypower	2019	1	1200	12
william hill	2019	1	1800	18

- Delete the #merch_users table
- Update @year and @quarter to the next row in the #yearQuarterTable.
- The loop will continue until all rows in #yearQuarterTable have been processed.

Results of SQL query

The result table of this query would be the #merch_comp table, which contains aggregated spend and transaction data for each merchant and quarter.

The SQL query is aiming to identify which of the five listed merchants have the highest spend and/or transaction volume in each quarter, for transactions made

Thank you!