

# Фінальний проект

ДАТА АНАЛІТИКА

Лук'яненко Олександр

[DAN.IT]  
EDUCATION



Ви - ВІ-консультант, якого найняла компанія, яка прагне розширити свій бізнес за допомогою нових магазинів. Вони залучили вас, щоб проаналізувати цікаві закономірності та тенденції в їхніх даних і допомогти їм прийняти обґрунтовані рішення.

Дані: дані про продажі та запаси для фіктивної мережі магазинів іграшок у Мексиці, включаючи інформацію про продукти, магазини, щоденні транзакції продажу та поточний рівень запасів у кожному місці.

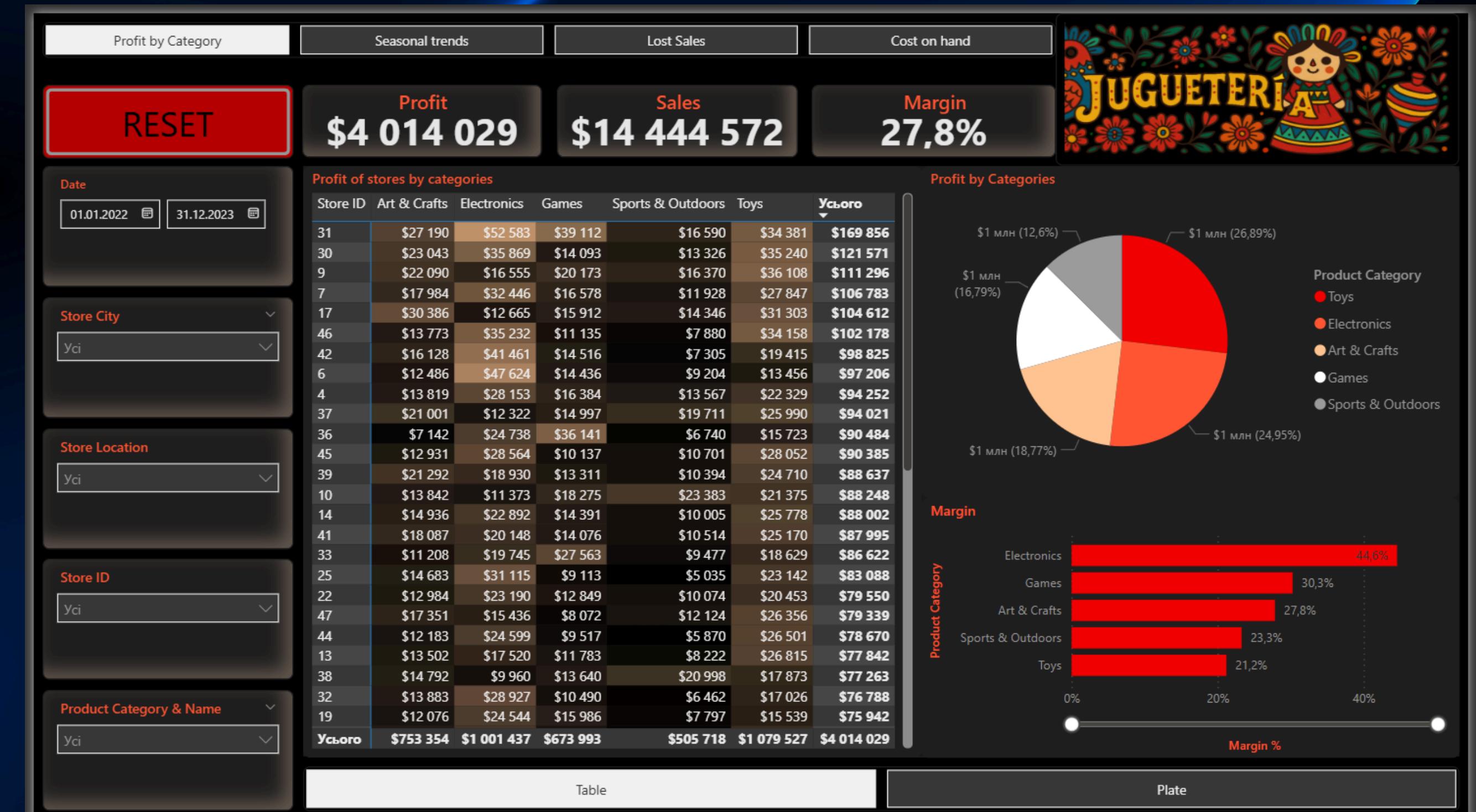
## Завдання 1:

Які категорії продуктів забезпечують найбільший прибуток? Це однаково в усіх магазинах?

Найбільший прибуток дають Toys та Electronics — разом понад 50% доходу.

Але це не однаково для всіх магазинів: у деяких лідирують іграшки, в інших — електроніка.

Тому важливо планувати запаси з урахуванням специфіки кожного магазину.



## Завдання 2:

Чи можете ви знайти якісь сезонні тенденції чи закономірності в даних про продажі?

Years	Season profit %	Season_Comparison	Profit	AverageSeasonProfit
<b>2022</b>			<b>\$2 189 787</b>	
Winter	26,26%		\$575 065	\$191 688
Spring	25,16%		\$550 985	\$183 662
Autumn	24,57%		\$538 069	\$179 356
Summer	24,01%		\$525 668	\$175 223
<b>2023</b>			<b>\$1 824 242</b>	
Spring	36,03% ▲	19,30%	\$657 352	\$219 117
Summer	32,45% ▲	12,63%	\$592 057	\$197 352
Winter	21,62% ▼	-31,42%	\$394 388	\$197 194
Autumn	9,89% ▼	-66,46%	\$180 445	\$180 445
<b>Усього</b>			<b>\$4 014 029</b>	

```

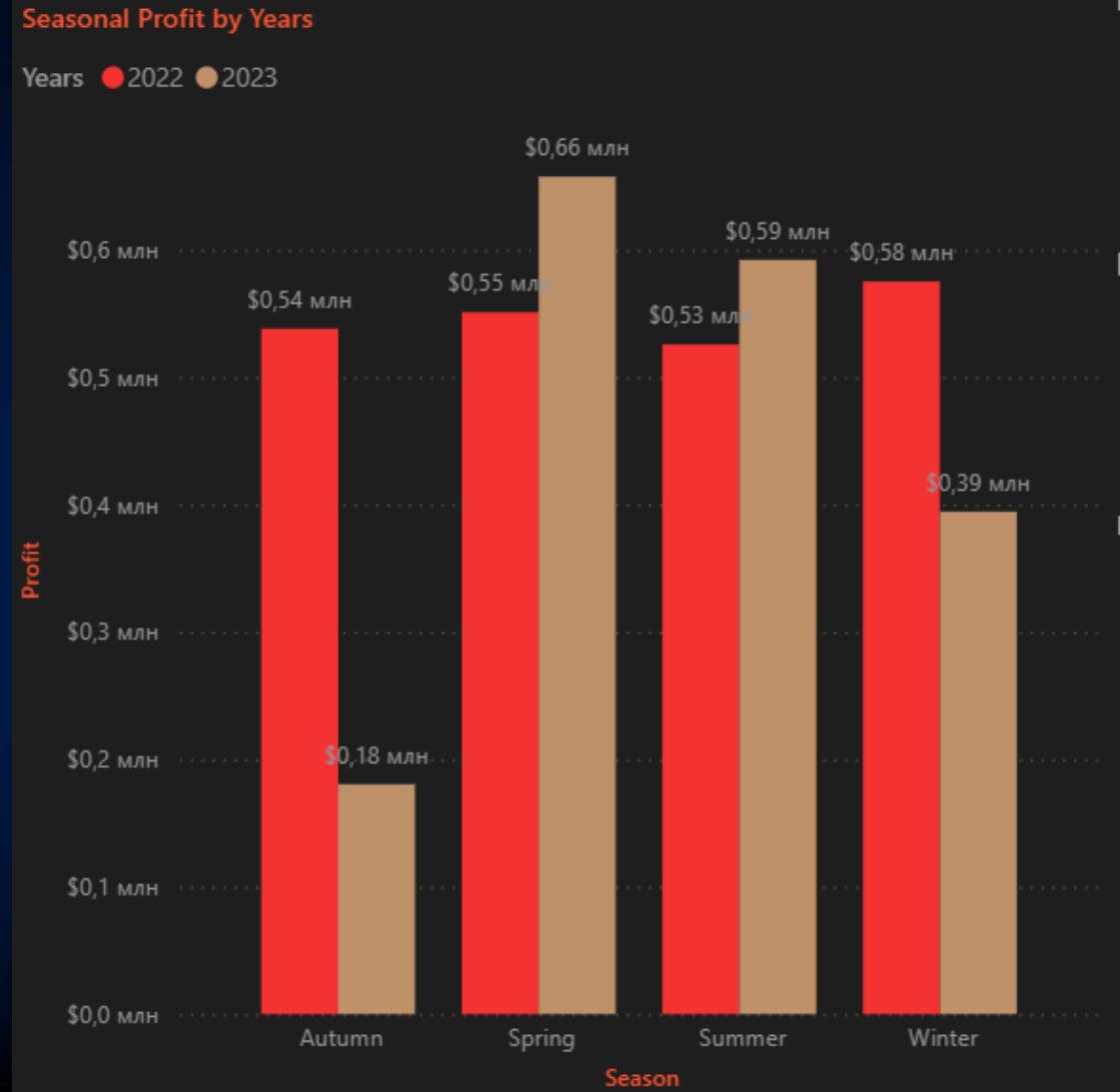
1 season_comparison =
2
3 -- вибираємо рік із контексту матриці
4 var selectedyear = selectedvalue('dimcalendar'[years])
5
6 -- вибираємо сезон
7 var selectedseason = selectedvalue('dimcalendar'[season])
8
9 -- профіт для вибраного сезону поточного року
10 var seasonprofit_currentyear =
11   calculate(
12     [profit],
13     'dimcalendar'[season] = selectedseason,
14     'dimcalendar'[years] = selectedyear -- використовуємо рік із матриці
15   )
16
17 -- профіт для того ж сезону попереднього року
18 var seasonprofit_previousyear =
19   calculate(
20     [profit],
21     'dimcalendar'[season] = selectedseason,
22     'dimcalendar'[years] = selectedyear - 1 -- використовуємо попередній рік
23   )
24
25 -- обчислюємо зміну сезонного профіту між роками
26 return if(
27   hasonevalue('dimcalendar'[years]) && hasonevalue('dimcalendar'[season]),
28   -- переконуємося, що вибраний лише один рік та сезон
29   divide(seasonprofit_currentyear - seasonprofit_previousyear, seasonprofit_previousyear, blank()),
30   blank() -- уникамо помилок, якщо вибрано кілька значень
31 )

```

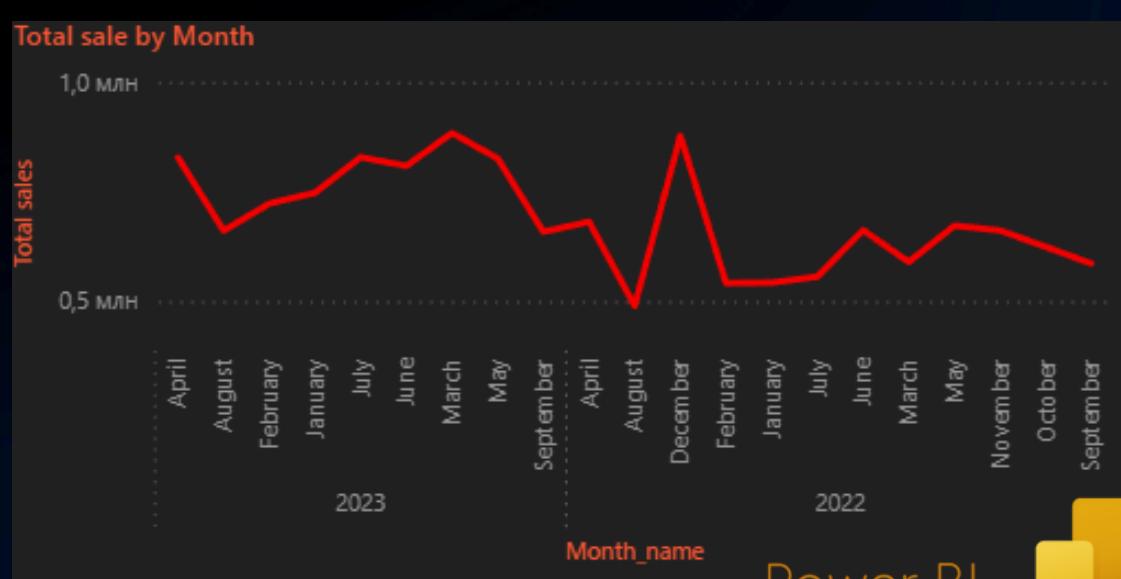
```

1 season_profit % =
2
3 -- визначаємо рік для сезону
4 var selectedyear = selectedvalue('dimcalendar'[years])
5
6 -- визначаємо вибраний сезон
7 var selectedseason = selectedvalue('dimcalendar'[season])
8
9 -- профіт лише для вибраного сезону
10 var seasonprofit =
11   calculate(
12     [profit],
13     'dimcalendar'[season] = selectedseason,
14     'dimcalendar'[years] = selectedyear -- додаємо фільтр року!
15   )
16
17 -- загальний профіт року без сезонного розподілу
18 var yearprofit =
19   calculate(
20     [profit],
21     removefilters('dimcalendar'[season]), -- забираємо сезонний фільтр
22     'dimcalendar'[years] = selectedyear -- фільтруємо тільки вибраний рік
23   )
24
25 -- повертаємо частку сезонного профіту від загального річного
26 return divide(
27   seasonprofit, -- профіт сезону
28   yearprofit, -- загальний профіт року
29   blank() -- запобігає діленню на 0 або blank()
30 )

```



У даних чітко простежуються сезонні тренди. Найбільші продажі — навесні та влітку, найменші — восени. Весна приносить компанії найбільший прибуток, а осінь є періодом найнижчого попиту. Це важливо враховувати при плануванні запасів і маркетингової активності.



### Завдання 3:

## Чи втрачаються продажі через відсутність продуктів у певних місцях?

```

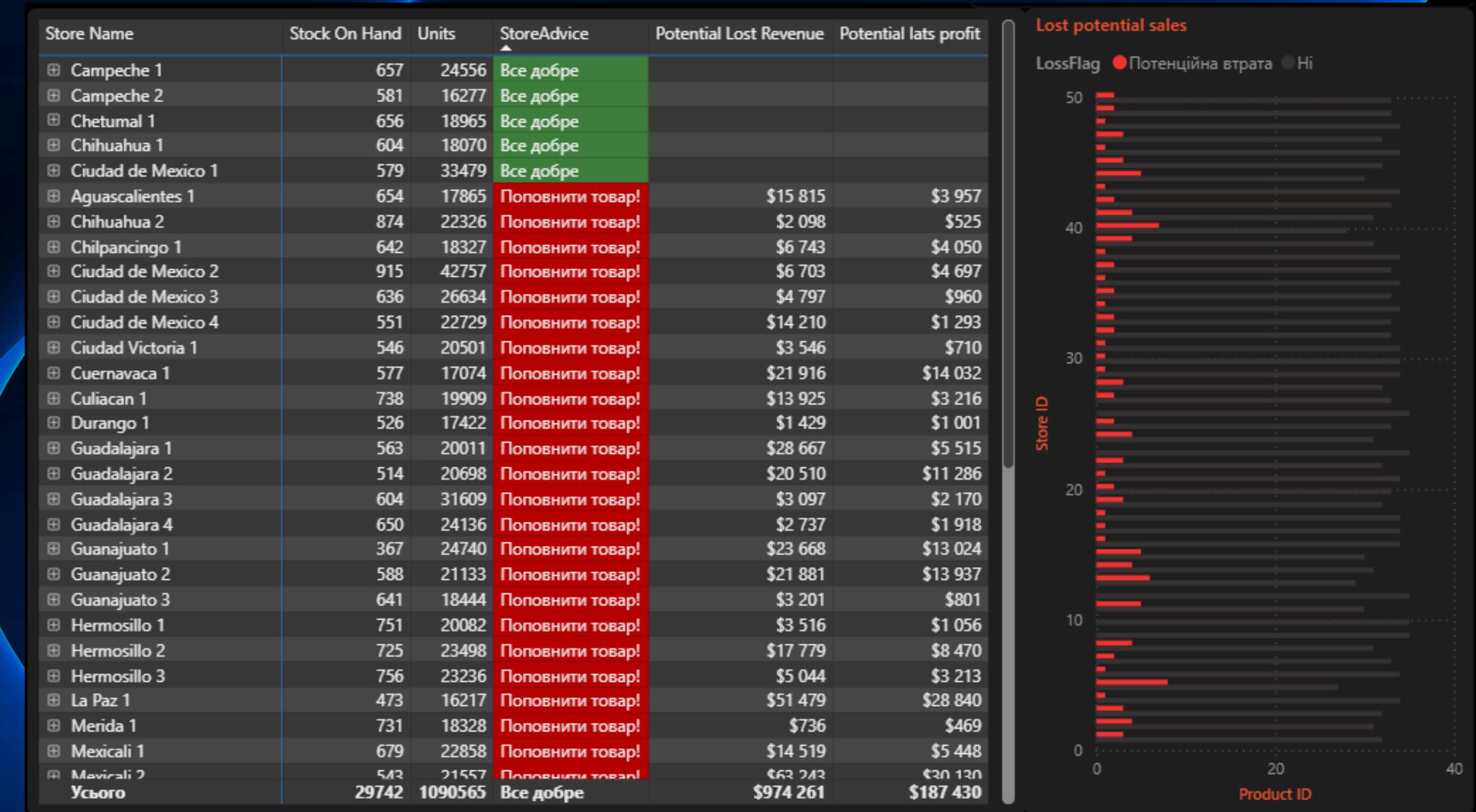
1 storeadvice =
2
3 -- обчислюємо кількість товарів, які потребують поповнення
4 var productsneedingrestock =
5 sumx(
6     values('dim products'[product id]), -- перебираємо всі унікальні товари
7     if(
8         calculate(sum('inventory'[stock on hand])) = 0 && -- якщо запас товару дорівнює нулю
9         calculate(sum('sales'[units])) > 0, -- і були продажі цього товару
10        1, -- цей товар потребує поповнення
11        0 -- інакше – не потребує
12    )
13 )
14
15 -- повертаємо текстову пораду залежно від результату
16 return
17 if(
18     productsneedingrestock > 0,
19     "поповнити товар!", -- якщо хоча б один товар потребує поповнення
20     "все добре" -- якщо всі товари в порядку
21 )

```

```

1 potentiallostrevenue =
2
3 -- обчислюємо кількість втрачених одиниць товару
4 var lostunits =
5 calculate(
6     sum(inventorysalescombined[sold]), // підсумовує кількість проданих одиниць
7     inventorysalescombined[loss] = 1 // тільки для рядків, де втрати = 1
8 )
9
10 -- визначаємо максимальну ціну одиниці серед втрачених продажів
11 var unitprice =
12 calculate(
13     max(inventorysalescombined[price]), // беремо максимальну ціну товару
14     inventorysalescombined[loss] = 1 // знову тільки для втрачених продажів
15 )
16
17 -- повертаємо загальний потенційний втрачений дохід
18 return
19 lostunits * unitprice // множимо кількість втрачених одиниць на ціну

```



Так, компанія втрачає продажі через відсутність товарів. За два роки потенційно втрачено майже \$1 млн продажів та \$187 тис. прибутку.

Найпроблемніші магазини — у Ciudad de Mexico, Guadalajara та Chihuahua. У цих магазинах потрібен більш точний прогноз попиту та швидше поповнення товарів.

## Завдання 4:

Скільки грошей пов'язано із запасами в магазинах іграшок? На скільки днів їх вистачить?

```

1 Avg sales of day =
2 VAR TotalDays = COUNTROWS(ALL('DimCalendar'[Date])) // Підраховує всі дати в календарі, включаючи дні без продажів
3 RETURN
4 DIVIDE([sales quantity], TotalDays, 0) // Ділення загальної кількості продажів на кількість днів, запобігаючи діленню на нуль

1 Days until depletion =
2 DIVIDE(
3   [Stock on hand], // Поточний рівень запасів товару
4   [Avg sales of day], // Середня кількість продажів на день
5   0 // Запобігає діленню на нуль, якщо немає продажів
6 )

```

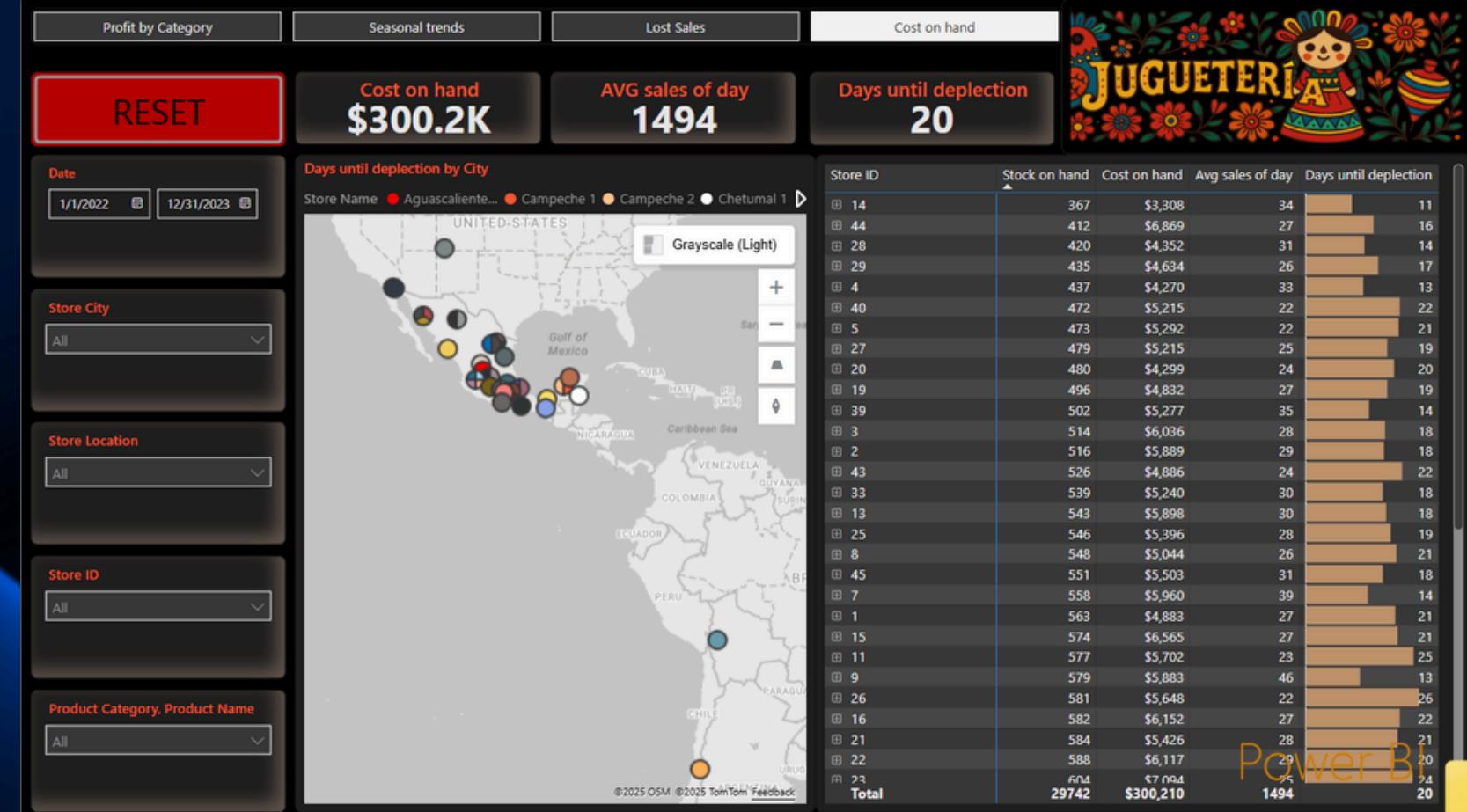
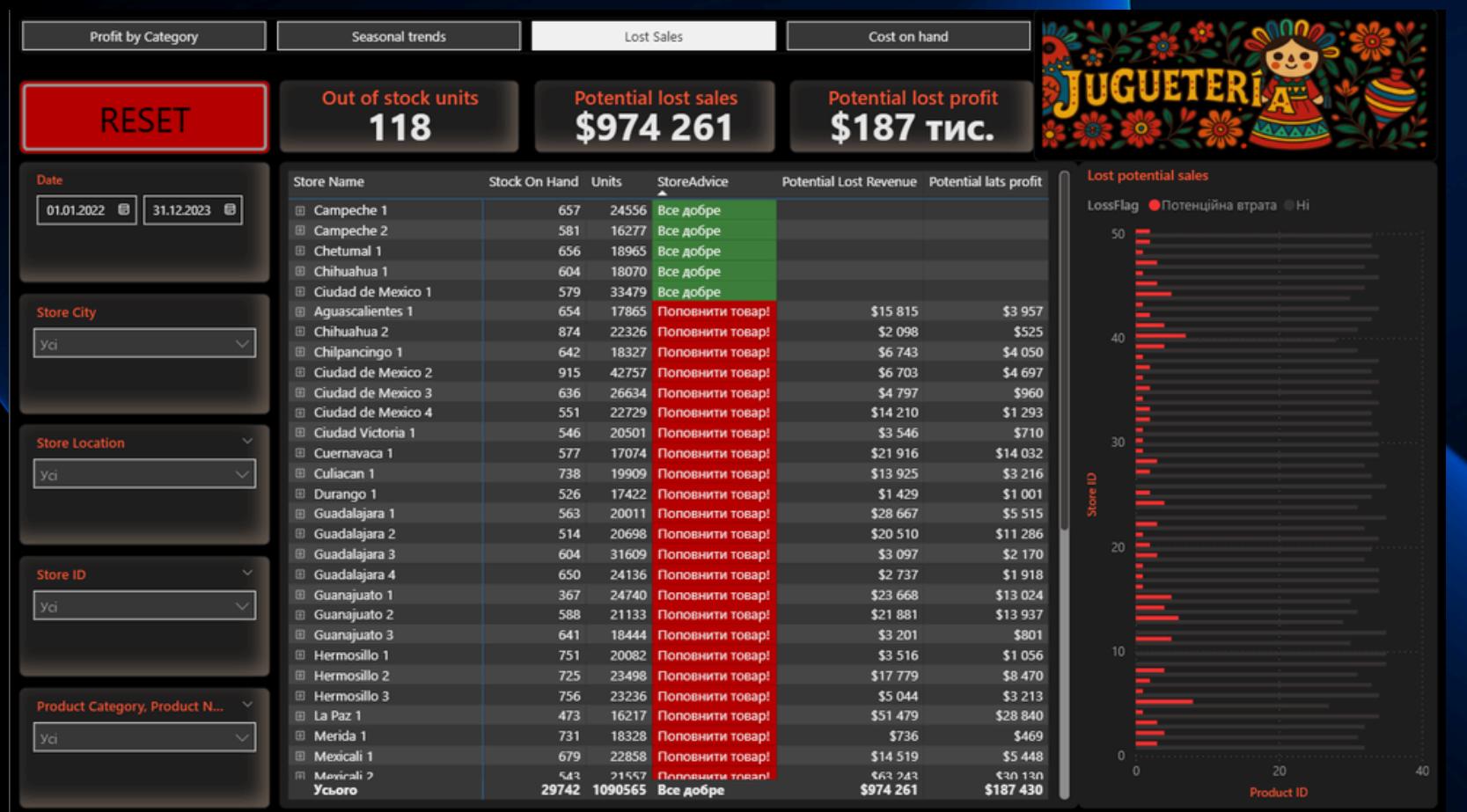
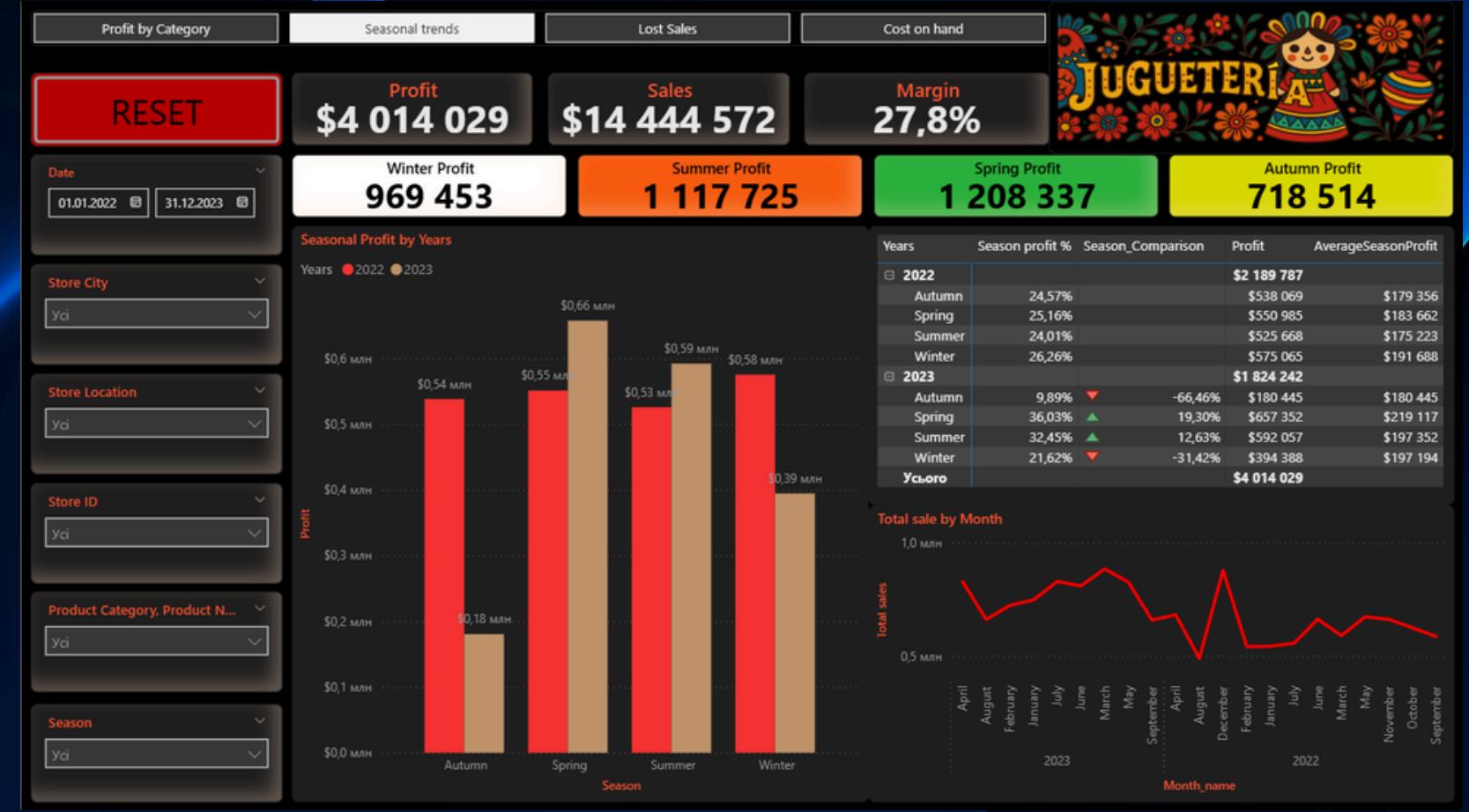
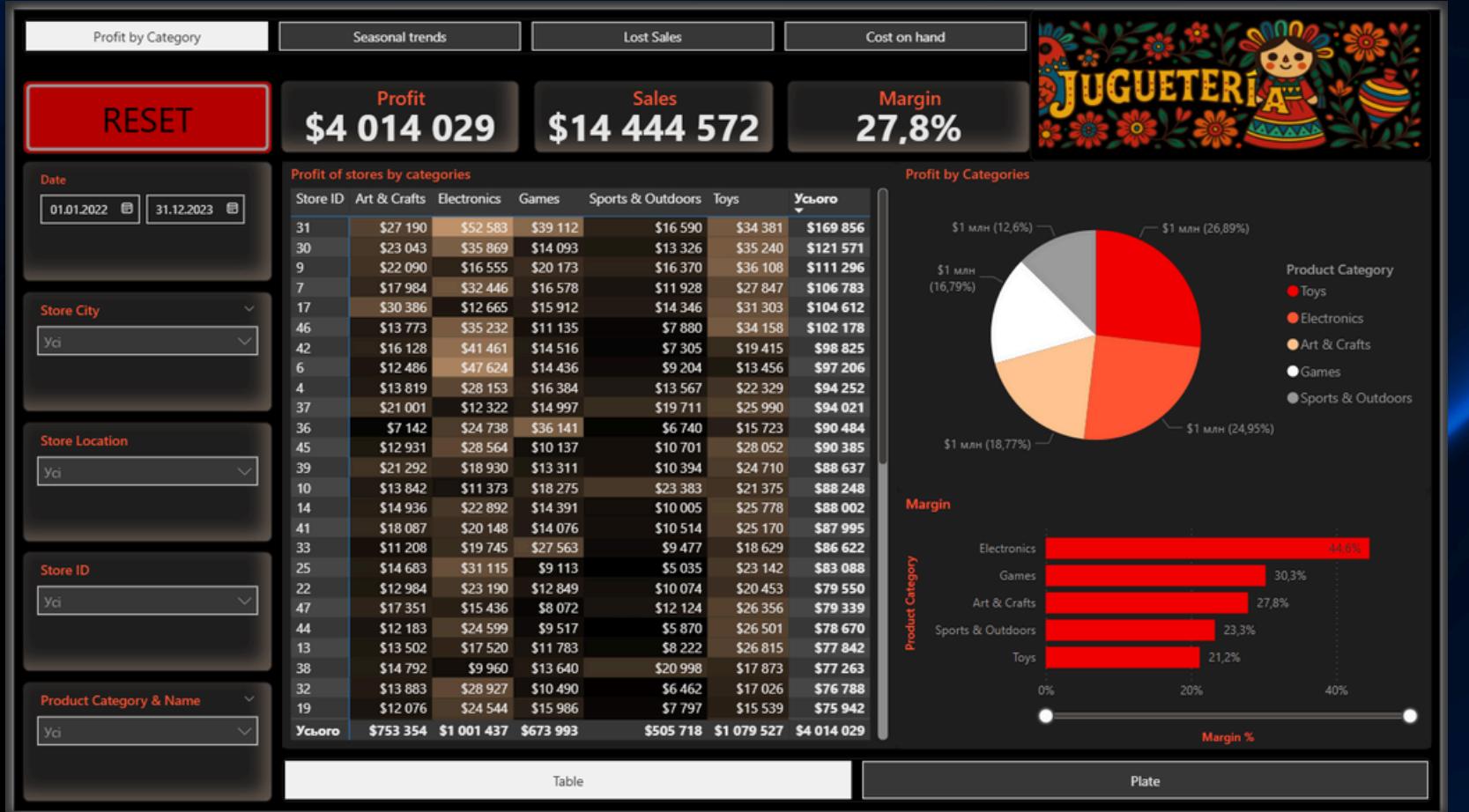
У компанії зараз запасів на суму близько \$300 тисяч.

У середньому цього вистачає на 20 днів продажів.

При цьому магазини сильно відрізняються: деякі мають запаси тільки на 11 днів, інші — на 25 і більше.

Це означає, що компанії важливо оптимізувати розподіл товарів між магазинами, щоб уникнути як дефіциту, так і надлишкових запасів.

Store ID	Stock on hand	Cost on hand	Avg sales of day	Days until depletion
# 14	367	\$3 308	34	11
# 44	412	\$6 869	27	16
# 28	420	\$4 352	31	14
# 29	435	\$4 634	26	17
# 4	437	\$4 270	33	13
# 40	472	\$5 215	22	22
# 5	473	\$5 292	22	21
# 27	479	\$5 215	25	19
# 20	480	\$4 299	24	20
# 19	496	\$4 832	27	19
# 39	502	\$5 277	35	14
# 3	514	\$6 036	28	18
# 2	516	\$5 889	29	18
# 43	526	\$4 886	24	22
# 33	539	\$5 240	30	18
# 13	543	\$5 898	30	18
# 25	546	\$5 396	28	19
# 8	548	\$5 044	26	21
# 45	551	\$5 503	31	18
# 7	558	\$5 960	39	14
# 1	563	\$4 883	27	21
# 15	574	\$6 565	27	21
# 11	577	\$5 702	23	25
# 9	579	\$5 883	46	13
# 26	581	\$5 648	22	26
# 16	582	\$6 152	27	22
# 21	584	\$5 426	28	21
# 22	588	\$6 117	29	20
# 23	604	\$7 001	25	24
<b>Усього</b>	<b>29742</b>	<b>\$300 210</b>	<b>1494</b>	<b>20</b>



# Запити SQL

1. Покажіть середню зарплату співробітників за кожен рік, до 2005 року.

```
3 • SELECT
4     YEAR(from_date) AS report_year,
5     ROUND(AVG(salary), 2) AS avg_salary
6  FROM salaries
7 GROUP BY report_year
8 HAVING report_year BETWEEN MIN(report_year) AND 2005
9 ORDER BY report_year;
```

report_year	avg_salary
1985	53182.36
1986	54084.78
1987	54959.63
1988	55862.45
1989	56840.67
1990	57839.46
1991	58803.87

2. Покажіть середню зарплату співробітників по кожному відділу.  
Примітка: потрібно розрахувати по поточній зарплаті, та поточному відділу співробітників

```
16 • SELECT
17     d.dept_name,
18     ROUND(AVG(s.salary), 2) AS avg_salary
19  FROM departments d
20 JOIN dept_emp de ON d.dept_no = de.dept_no
21 JOIN salaries s ON de.emp_no = s.emp_no
22 WHERE CURRENT_DATE BETWEEN de.from_date AND de.to_date
23     AND CURRENT_DATE BETWEEN s.from_date AND s.to_date
24 GROUP BY d.dept_name
25 ORDER BY avg_salary DESC;
```

dept_name	avg_salary
Sales	88852.97
Marketing	80058.85
Finance	78559.94
Research	67913.37
Production	67843.30
Development	67657.92
Customer Service	67285.23
Quality Management	65441.99

3. Покажіть середню зарплату співробітників по кожному відділу за кожний рік

```
29 • SELECT
30     d.dept_name,
31     YEAR(s.from_date) AS salary_year,
32     ROUND(AVG(s.salary), 2) AS avg_salary
33  FROM departments d
34 JOIN dept_emp de ON d.dept_no = de.dept_no
35 JOIN salaries s ON de.emp_no = s.emp_no
36 GROUP BY d.dept_name, salary_year
37 ORDER BY salary_year, avg_salary DESC;
```

dept_name	salary_year	avg_salary
Sales	1985	69975.04
Marketing	1985	61614.57
Finance	1985	60285.39
Research	1985	49098.05
Production	1985	48924.27
Development	1985	48706.24
Customer Service	1985	48195.14
Quality Management	1985	46644.21
Human Resources	1985	44956.80



# Запити SQL

4. Покажіть відділи в яких зараз працює більше 15000 співробітників.

```
43 • SELECT
44     d.dept_name,
45     COUNT(de.emp_no) AS active_employees
46     FROM departments d
47     JOIN dept_emp de ON d.dept_no = de.dept_no
48     WHERE CURRENT_DATE BETWEEN de.from_date AND de.to_date
49     GROUP BY d.dept_name
50     HAVING COUNT(de.emp_no) > 15000
51     ORDER BY active_employees DESC;
```

	dept_name	active_employees
▶	Development	61386
	Production	53304
	Sales	37701
	Customer Service	17569
	Research	15441

5. Для менеджера який працює найдовше покажіть його номер, відділ, дату прийому на роботу, прізвище

```
56 • SELECT
57     e.emp_no,
58     e.last_name,
59     e.hire_date,
60     d.dept_name
61     FROM employees e
62     JOIN dept_manager dm ON e.emp_no = dm.emp_no
63     JOIN departments d ON dm.dept_no = d.dept_no
64     WHERE CURRENT_DATE BETWEEN dm.from_date AND dm.to_date
65     ORDER BY e.hire_date ASC
66     LIMIT 1;
```

	emp_no	last_name	hire_date	dept_name
▶	110114	Legleitner	1985-01-14	Finance



# Запити SQL

6. Покажіть топ-10 діючих співробітників компанії з найбільшою різницею між їх зарплатою і середньою зарплатою в їх відділі.

```

71 • WITH current_assignments AS (
72     SELECT emp_no, dept_no
73     FROM dept_emp
74     WHERE CURRENT_DATE BETWEEN from_date AND to_date
75 ),
76     current_salaries AS (
77     SELECT emp_no, salary
78     FROM salaries
79     WHERE CURRENT_DATE BETWEEN from_date AND to_date
80 ),
81     avg_salary_per_dept AS (
82     SELECT ca.dept_no, ROUND(AVG(cs.salary), 2) AS avg_dept_salary
83     FROM current_assignments ca
84     JOIN current_salaries cs ON ca.emp_no = cs.emp_no
85     GROUP BY ca.dept_no
86 ),
87     employee_salary_diff AS (
88     SELECT
89         ca.emp_no,
90         d.dept_name,
91         ROUND(cs.salary - a.avg_dept_salary, 2) AS salary_diff
92     FROM current_assignments ca
93     JOIN current_salaries cs ON ca.emp_no = cs.emp_no
94     JOIN avg_salary_per_dept a ON ca.dept_no = a.dept_no
95     JOIN departments d ON ca.dept_no = d.dept_no
96 )
97     SELECT emp_no, dept_name, salary_diff
98     FROM employee_salary_diff
99     ORDER BY salary_diff DESC
100    LIMIT 10;

```

	emp_no	dept_name	salary_diff
▶	421835	Human Resources	78031.10
	18006	Customer Service	77580.77
	13386	Development	76776.08
	96957	Customer Service	76664.77
	432583	Customer Service	76651.77
	98169	Customer Service	76546.77
	485205	Customer Service	74269.77
	419748	Development	73126.08
	28337	Customer Service	71502.77
	235645	Customer Service	71351.77

7. Для кожного відділу покажіть другого по порядку менеджера. Необхідно вивести відділ, прізвище ім'я менеджера, дату прийому на роботу менеджера і дату коли він став менеджером відділу

```

105 • WITH ranked_managers AS (
106     SELECT
107         dm.dept_no,
108         dm.emp_no,
109         dm.from_date AS manager_start_date,
110         ROW_NUMBER() OVER (PARTITION BY dm.dept_no ORDER BY dm.from_date) AS rn
111     FROM dept_manager dm
112 )
113     SELECT
114         d.dept_name,
115         e.last_name,
116         e.first_name,
117         e.hire_date,
118         rm.manager_start_date
119     FROM ranked_managers rm
120     JOIN employees e ON rm.emp_no = e.emp_no
121     JOIN departments d ON rm.dept_no = d.dept_no
122     WHERE rm.rn = 2
123     ORDER BY d.dept_name;
124
125 • SELECT
126         d.dept_name,
127         e.first_name,
128         e.last_name,
129         e.hire_date,
130         fm.from_date AS manager_start_date
131     FROM first_two_managers fm
132     JOIN employees e ON fm.emp_no = e.emp_no
133     JOIN departments d ON fm.dept_no = d.dept_no;

```

	dept_name	last_name	first_name	hire_date	manager_start_date
▶	Customer Service	Giarratana	Marjo	1988-02-12	1988-10-17
	Development	DasSarma	Leon	1986-10-21	1992-04-25
	Finance	Legleitner	Isamu	1985-01-14	1989-12-17
	Human Resources	Sigstam	Karsten	1985-08-04	1992-03-21
	Marketing	Minakawa	Vishwani	1986-04-12	1991-10-01
	Production	Cools	Rosine	1985-11-22	1988-09-09
	Quality Management	Hofmeyr	Rutger	1989-01-07	1989-05-06
	Research	Kambil	Hilary	1988-01-31	1991-04-08
	Sales	Zhang	Hauke	1986-12-30	1991-03-07



# Дизайн бази даних

1. Створіть базу даних для управління курсами.

```
1  DROP DATABASE IF EXISTS stepproject;
2 • CREATE DATABASE stepproject;
3 • USE stepproject;
4
5 • CREATE TABLE teachers (
6     teacher_no INT PRIMARY KEY AUTO_INCREMENT,
7     teacher_name VARCHAR(100) NOT NULL,
8     phone_no VARCHAR(20)
9 );
10
11 • CREATE TABLE courses (
12     course_no INT PRIMARY KEY AUTO_INCREMENT,
13     course_name VARCHAR(100) NOT NULL,
14     start_date DATE,
15     end_date DATE
16 );
17
18 • CREATE TABLE students (
19     student_no INT PRIMARY KEY AUTO_INCREMENT,
20     teacher_no INT NOT NULL,
21     course_no INT NOT NULL,
22     student_name VARCHAR(100) NOT NULL,
23     email VARCHAR(100),
24     birth_date DATE,
25     FOREIGN KEY (teacher_no) REFERENCES teachers(teacher_no),
26     FOREIGN KEY (course_no) REFERENCES courses(course_no)
27 );
```

2. Додайте будь-які данні (7-10 рядків) в кожну таблицю.

```
31      -- Вставка викладачів
32 • INSERT INTO teachers (teacher_name, phone_no) VALUES
33     ('Олена Ковальчук', '0671234567'),
34     ('Ігор Сидоренко', '0509876543'),
35     ('Марія Гончар', '0631122334'),
36     ('Андрій Литвин', '0669988776'),
37     ('Наталія Шевченко', '0685566778'),
38     ('Василь Бондар', '0953344556'),
39     ('Тетяна Романенко', '0734455667');
40
41      -- Вставка курсів
42 • INSERT INTO courses (course_name, start_date, end_date) VALUES
43     ('Основи програмування', '2024-09-01', '2024-12-15'),
44     ('Бази даних', '2024-09-01', '2024-12-15'),
45     ('Веб-розробка', '2024-10-01', '2025-01-20'),
46     ('Математика для IT', '2024-09-15', '2024-12-30'),
47     ('Машинне навчання', '2025-01-10', '2025-04-25'),
48     ('Кібербезпека', '2024-11-01', '2025-02-28'),
49     ('Аналіз даних', '2025-02-01', '2025-05-15');
```

```
51      -- Вставка студентів (включно з дублікатами)
52 • INSERT INTO students (teacher_no, course_no, student_name, email, birth_date) VALUES
53     (1, 1, 'Анна Мельник', 'anna.m@example.com', '2003-05-12'),
54     (2, 2, 'Олександр Іванов', 'olex.iv@example.com', '2002-11-23'),
55     (3, 3, 'Ірина Кравець', 'irina.k@example.com', '2004-02-08'),
56     (4, 4, 'Дмитро Савченко', 'd.sav@example.com', '2001-07-19'),
57     (5, 5, 'Катерина Лисенко', 'katya.l@example.com', '2003-09-30'),
58     (6, 6, 'Богдан Черненко', 'bogdan.c@example.com', '2002-03-15'),
59     (7, 7, 'Ольга Ткаченко', 'olga.t@example.com', '2004-12-01'),
60     -- дублікати
61     (1, 1, 'Анна Мельник', 'anna.m@example.com', '2003-05-12'),
62     (1, 1, 'Анна Мельник', 'anna.m@example.com', '2003-05-12'),
63     (1, 1, 'Анна Мельник', 'anna.m@example.com', '2003-05-12');
```

4. Спеціально зробіть 3 дубляжі в таблиці students  
(додайте ще 3 однакові рядки)



# Дизайн бази даних

3. По кожному викладачу покажіть кількість студентів з якими він працював

```
65      -- Рахуємо кількість студентів по кожному викладачу
66 •  SELECT
67      t.teacher_name,
68      COUNT(s.student_no) AS student_count
69  FROM teachers t
70  LEFT JOIN students s ON t.teacher_no = s.teacher_no
71  GROUP BY t.teacher_name
72  ORDER BY student_count DESC;
```

	teacher_name	student_count
▶	Олена Ковальчук	4
	Ігор Сидоренко	1
	Марія Гончар	1
	Андрій Литвин	1
	Наталія Шевченко	1
	Василь Бондар	1
	Тетяна Романенко	1

5. Напишіть запит який виведе дублюючі рядки в таблиці students

```
75      -- Виводимо дублюючі рядки
76 •  SELECT *
77  FROM students
78  WHERE (teacher_no, course_no, student_name, email, birth_date) IN (
79      SELECT teacher_no, course_no, student_name, email, birth_date
80      FROM students
81      GROUP BY teacher_no, course_no, student_name, email, birth_date
82      HAVING COUNT(*) > 1
83  )
84  ORDER BY teacher_no, course_no, student_name;
```

	student_no	teacher_no	course_no	student_name	email	birth_date
▶	1	1	1	Анна Мельник	anna.m@example.com	2003-05-12
	8	1	1	Анна Мельник	anna.m@example.com	2003-05-12
	9	1	1	Анна Мельник	anna.m@example.com	2003-05-12
	10	1	1	Анна Мельник	anna.m@example.com	2003-05-12



# Python

1. Завантажте набір даних IKEA
2. Виконайте дослідницький аналіз набору даних, включаючи описову статистику та візуалізації(за бажанням). Опишіть результати.
3. На основі EDA та вашого здорового глузду виберіть дві гіпотези, які ви хочете перевірити/ проаналізувати. Дляожної гіпотези перерахуйте нульову гіпотезу та інші можливі альтернативні гіпотези, розробіть тести для їх розрізnenня та виконайте їх. Опишіть результати.
4. Навчіть модель передбачати ціну меблів. Зазначте, які стовпці не слід включати до моделі і чому. Створіть конвеєр перехресної перевірки для навчання та оцінки моделі, включаючи (за необхідності) такі кроки, як заповнення пропущених значень та нормалізація. Запропонуйте методи покращення продуктивності моделі. Опишіть результати.



# Імпорт бібліотек та завантаження даних

```
1 # 1. Імпорт необхідних бібліотек
2 import pandas as pd
3 import numpy as np
4 import seaborn as sns
5 import matplotlib.pyplot as plt
6 from scipy.stats import ttest_ind, chi2_contingency, mannwhitneyu
7 from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV, KFold
8 from sklearn.preprocessing import OneHotEncoder, StandardScaler
9 from sklearn.compose import ColumnTransformer
10 from sklearn.pipeline import Pipeline
11 from sklearn.impute import SimpleImputer
12 from sklearn.ensemble import RandomForestRegressor
13 from sklearn.metrics import r2_score
14 import warnings
15 warnings.filterwarnings('ignore')
16 sns.set_theme(style="whitegrid")
```

Імпортую всі бібліотеки, які використовуються для аналізу, статистики, побудови моделі та оцінювання її якості.

```
19 # 2. Завантаження даних
20 #
21 url = "https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-11-03/ikea.csv"
22 df = pd.read_csv(url)
23 print("Дані успішно завантажено!")
```

Дані завантажено напряму з GitHub. Датасет містить інформацію про товари IKEA: назви, ціни, розміри, дизайнерів, категорії.



# Очищення даних

```
29 # Видаляємо непотрібні стовпці та дублікати
30 df.drop(columns=['link'], inplace=True, errors='ignore')
31 df.drop_duplicates(subset='item_id', inplace=True)
32
33 # Очищаємо old_price від тексту та приводимо до числового типу
34 def clean_old_price(val): 1 usage
35     if pd.isna(val): return np.nan
36     if str(val).strip() == "No old price": return np.nan
37     try:
38         return float(str(val).replace("SR", "").replace(",", ".").strip())
39     except:
40         return np.nan
41
42 df['old_price'] = df['old_price'].apply(clean_old_price)
```

Видаляю зайві колонки та дублікати item\_id:  
link не несе корисної інформації для аналізу.  
item\_id має бути унікальним — тому дублікати видалені.

Привожу стару ціну до числового формату. Видаляю текст "SR", коми, пропуски.

```
44 # Очищення імен дизайнерів
45 def clean_designers(value, removeIKEA=False, emptyValue=np.nan): 1 usage
46     if not isinstance(value, str): return emptyValue
47     if len(value) > 0 and value[0].isdigit(): return emptyValue
48     parts = [p.strip() for p in value.replace("&", "/").split("/") if p.strip() != ""]
49     if removeIKEA:
50         parts = [p for p in parts if p != "IKEA of Sweden"]
51     return "/".join(sorted(parts)) if parts else emptyValue
52
53 df['designer_clean'] = df['designer'].apply(clean_designers)
```

Очищення дизайнерів (бо були некоректні значення: цифри, порожні рядки)



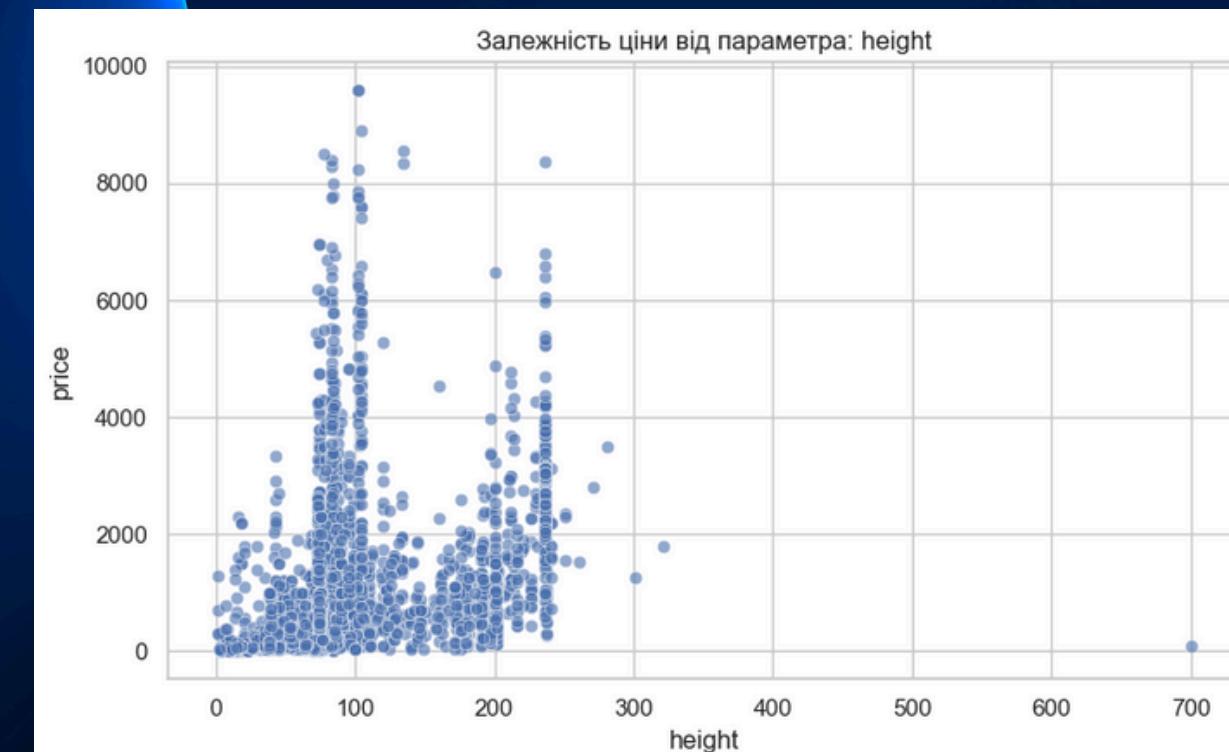
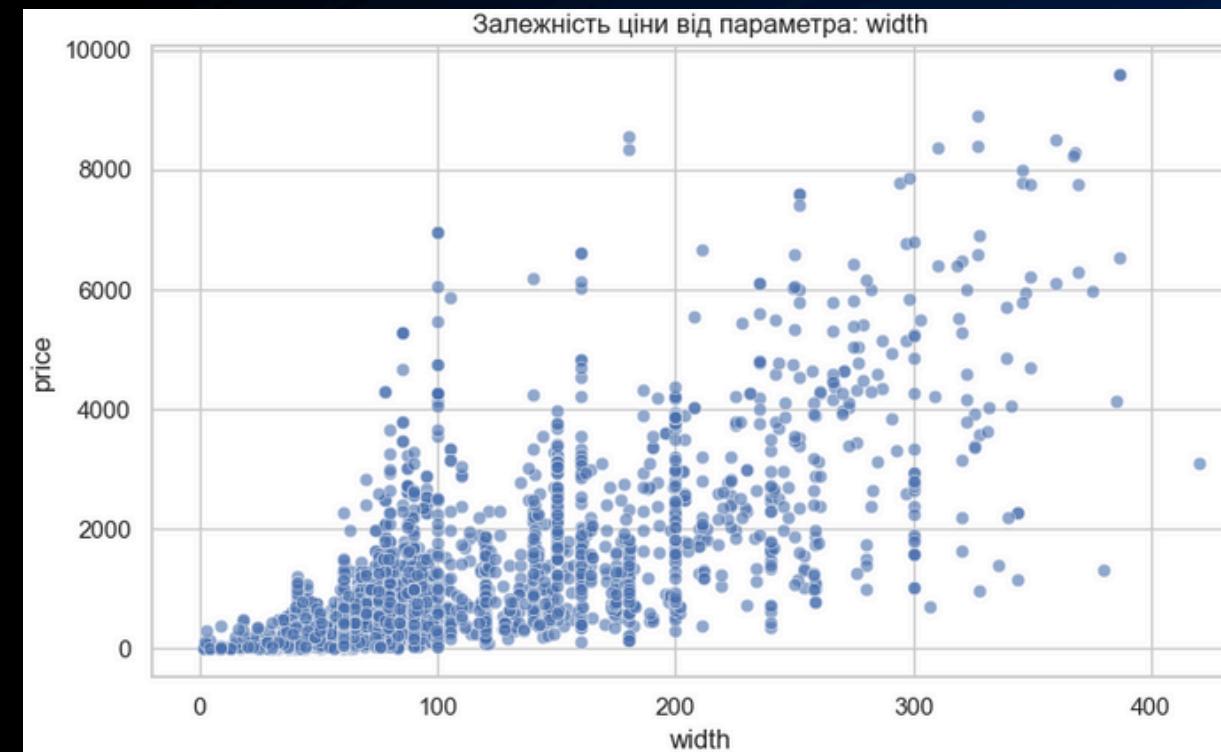
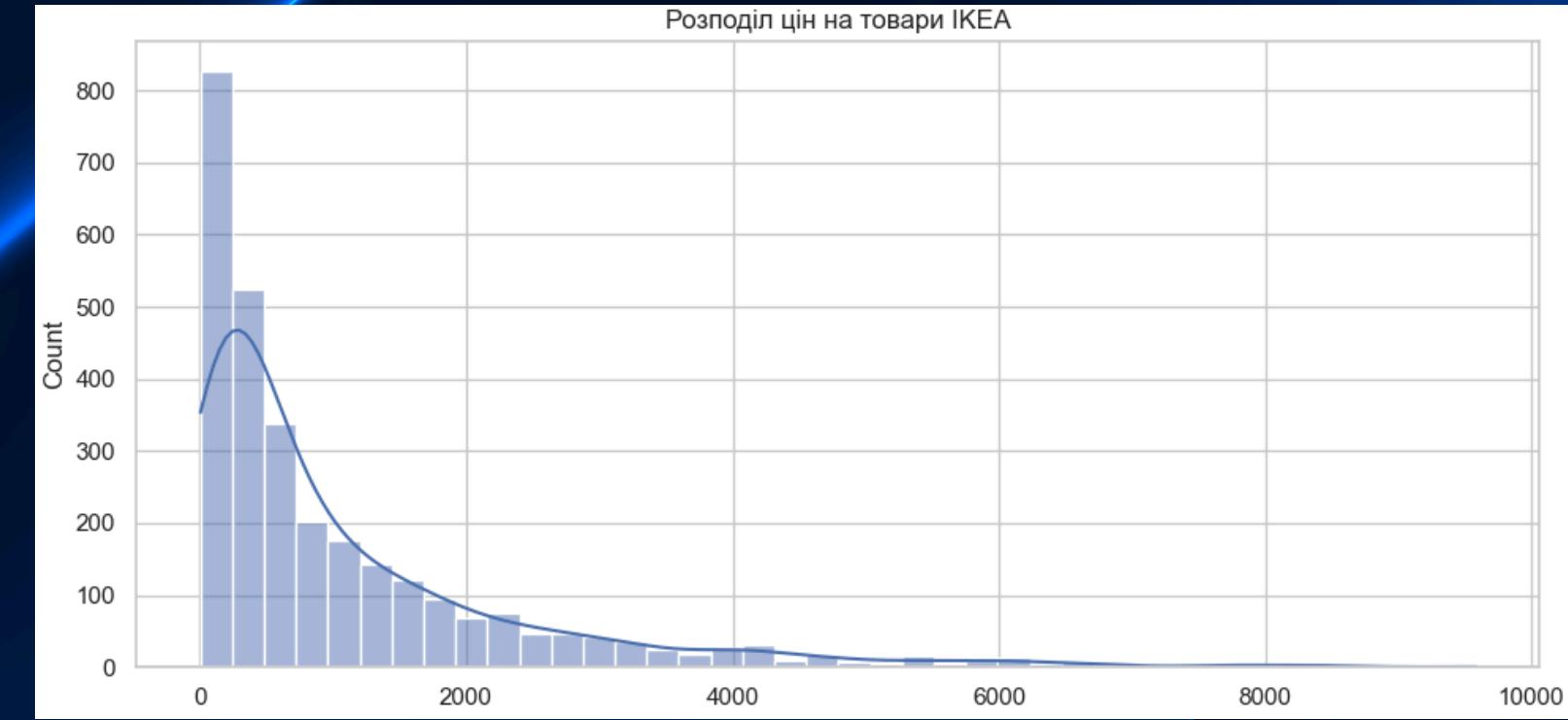
# Створення нових ознак

```
59 df['price'] = pd.to_numeric(df['price'], errors='coerce')
60 df['name_length'] = df['name'].astype(str).apply(len)
61 df['desc_length'] = df['short_description'].astype(str).apply(len)
62 df['other_colors'] = df['other_colors'].fillna('No')
63 df['sellable_online'] = df['sellable_online'].astype(str).str.lower().map({'true': True, 'false': False})
64
65 # Перетворюємо розміри на числові
66 df['depth'] = pd.to_numeric(df['depth'], errors='coerce')
67 df['height'] = pd.to_numeric(df['height'], errors='coerce')
68 df['width'] = pd.to_numeric(df['width'], errors='coerce')
69 df['volume'] = df['depth'] * df['height'] * df['width']
70
71 # Додаємо медіанну ціну для дизайнерів і категорій
72 designer_median = df.groupby('designer_clean')['price'].median()
73 category_median = df.groupby('category')['price'].median()
74 df['designer_price'] = df['designer_clean'].map(designer_median)
75 df['category_price'] = df['category'].map(category_median)
```



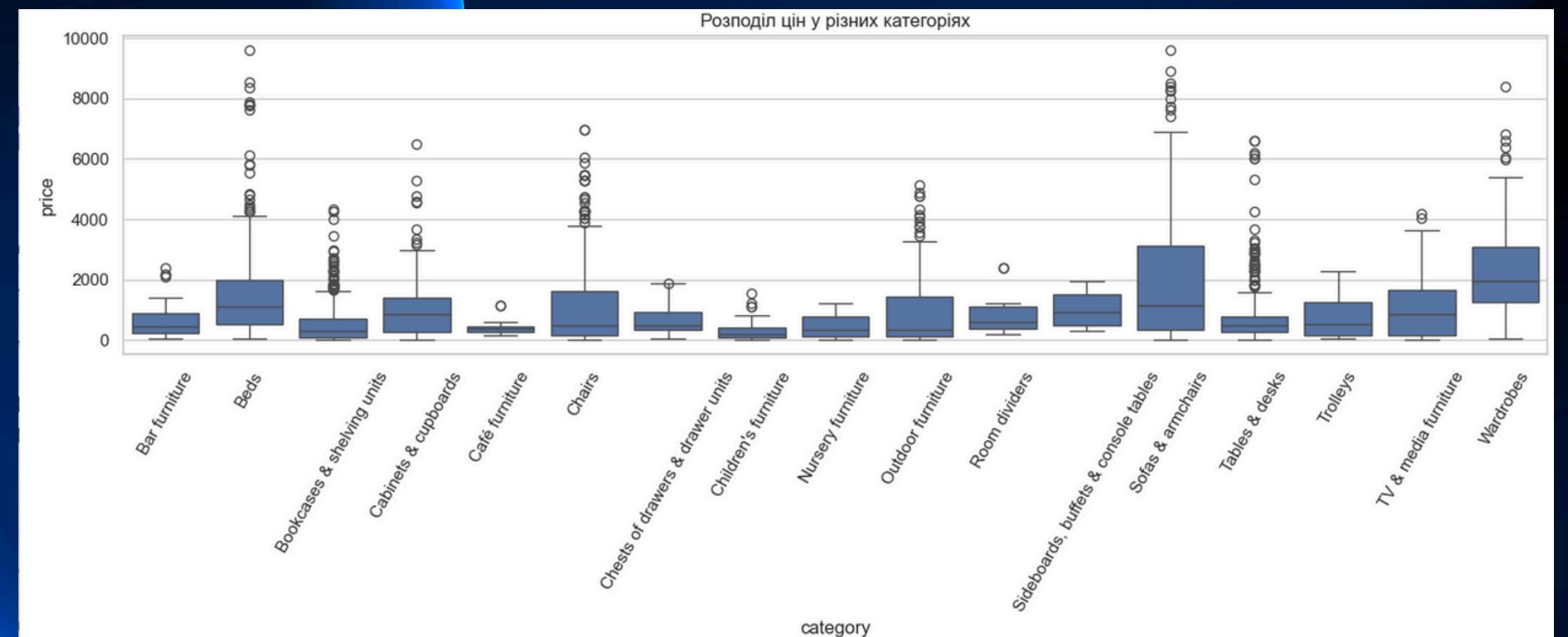
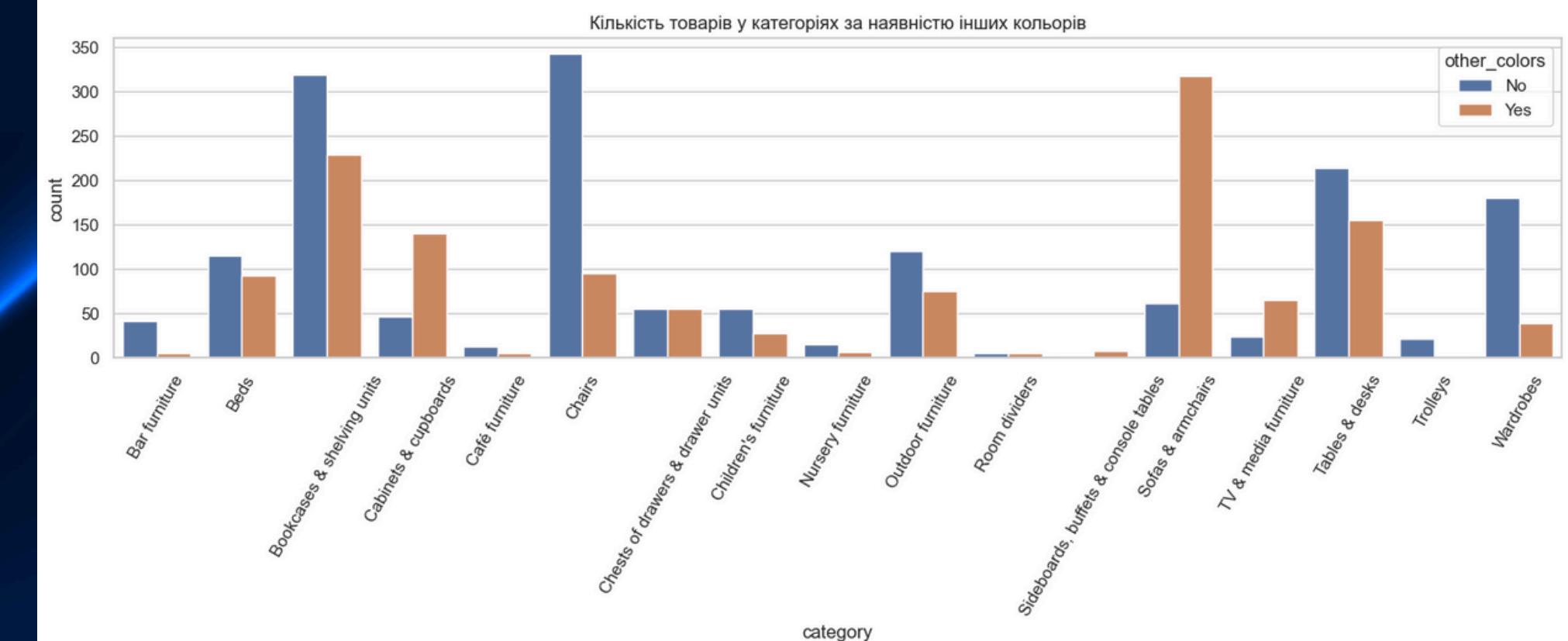
# Візуалізація

```
81 # Розподіл цін  
82 plt.figure(figsize=(10, 5))  
83 sns.histplot(df['price'].dropna(), bins=40, kde=True)  
84 plt.title("Розподіл цін на товари IKEA")  
85 plt.tight_layout()  
86 plt.show()  
87  
88 # Залежність ціни від розмірів  
89 for dim in ['width', 'height', 'depth']:   
90     plt.figure(figsize=(8, 5))  
91     sns.scatterplot(data=df, x=dim, y='price', alpha=0.6)  
92     plt.title(f"Залежність ціни від параметра: {dim}")  
93     plt.tight_layout()  
94     plt.show()
```



# Візуалізація

```
96 # Розподіл категорій і кольорів
97 plt.figure(figsize=(14, 6))
98 grp = df.groupby(['category', 'other_colors']).size().reset_index(name='count')
99 sns.barplot(data=grp, x='category', y='count', hue='other_colors')
100 plt.xticks(rotation=60)
101 plt.title("Кількість товарів у категоріях за наявністю інших кольорів")
102 plt.tight_layout()
103 plt.show()
104
105 # Boxplot цін по категоріях
106 plt.figure(figsize=(14, 6))
107 sns.boxplot(data=df, x='category', y='price')
108 plt.xticks(rotation=60)
109 plt.title("Розподіл цін у різних категоріях")
110 plt.tight_layout()
111 plt.show()
```



# Перевірка гіпотез

Чи відрізняється ціна товарів, які продаються онлайн, від тих, що тільки в магазинах?

```
118 online = df[df['sellable_online'] == True]['price'].dropna()
119 offline = df[df['sellable_online'] == False]['price'].dropna()
120
121 if len(online) >= 5 and len(offline) >= 5:
122     t_stat, p_t = ttest_ind(online, offline, nan_policy='omit')
123     m_stat, p_m = mannwhitneyu(online, offline, alternative='two-sided')
124     print(f"Онлайн vs Оффлайн: Т-тест p = {p_t:.4f} | Манна-Вітні p = {p_m:.4f}")
125 else:
126     print("Недостатньо даних для тесту онлайн/офлайн")
```

Чи пов'язані категорії товарів з наявністю інших категорій товарів з наявністю інших кольорів?

```
129 cont = pd.crosstab(df['category'], df['other_colors'])
130 chi2, p, dof, ex = chi2_contingency(cont)
131 n = cont.sum().sum()
132 cram_v = np.sqrt(chi2 / (n * (min(cont.shape) - 1)))
133 print(f"Chi2 тест p = {p:.4f} | Коефіцієнт Крамера V = {cram_v:.3f}")
```

Дані успішно завантажено!

Онлайн vs Оффлайн: Т-тест p = 0.0406 | Манна-Вітні p = 0.0064  
Chi<sup>2</sup> тест p = 0.0000 | Коефіцієнт Крамера V = 0.430

ціна статистично відрізняється

між категорією і кольорами є сильний статистичний зв'язок.



# Підготовка даних для моделі

```
138     num_features = ['depth', 'height', 'width', 'name_length', 'desc_length',
139                         'designer_price', 'category_price']
140     cat_features = ['other_colors']
141
142     # Залишаємо тільки потрібні колонки (X та y)
143     df_model = df[num_features + cat_features + ['price']].dropna()
144     X = df_model[num_features + cat_features]
145     y = df_model['price']
146
147     print("Розмір датафрейму для моделі:", df_model.shape)
148     print("Колонки моделі:", df_model.columns.tolist())
```

Розмір датафрейму для моделі: (1561, 9)

Колонки моделі: ['depth', 'height', 'width', 'name\_length', 'desc\_length', 'designer\_price', 'category\_price', 'other\_colors', 'price']



# Побудова та оцінка моделі

Пайплайн для числових та категоріальних даних

```
155 num_pipe = Pipeline([
156     ('imputer', SimpleImputer(strategy='median')),
157     ('scaler', StandardScaler())
158 ])
159
160 cat_pipe = Pipeline([
161     ('imputer', SimpleImputer(strategy='constant', fill_value='Unknown')),
162     ('onehot', OneHotEncoder(handle_unknown='ignore', sparse_output=False))
163 ])
164
165 preprocessor = ColumnTransformer([
166     ('num', num_pipe, num_features),
167     ('cat', cat_pipe, cat_features)
168 ])
```

Результат

```
Базова модель RandomForest: R2 = 0.9117
Найкращі параметри GridSearchCV: {'model__max_depth': 15, 'model__max_features': 'sqrt', 'model__n_estimators': 100}
R2 під час крос-валідації (train): 0.8532
Використовуємо підібрану модель? False
Фінальна модель: R2 на тесті = 0.9117
```

Базова модель RandomForest

```
171 X_train, X_test, y_train, y_test = train_test_split(*arrays: X, y, test_size=0.2, random_state=42)
172 base_model = Pipeline([
173     ('pre', preprocessor),
174     ('model', RandomForestRegressor(random_state=42))
175 ])
176
177 base_model.fit(X_train, y_train)
178 y_pred = base_model.predict(X_test)
179 base_r2 = r2_score(y_test, y_pred)
180 print(f"Базова модель RandomForest: R2 = {base_r2:.4f}")
```

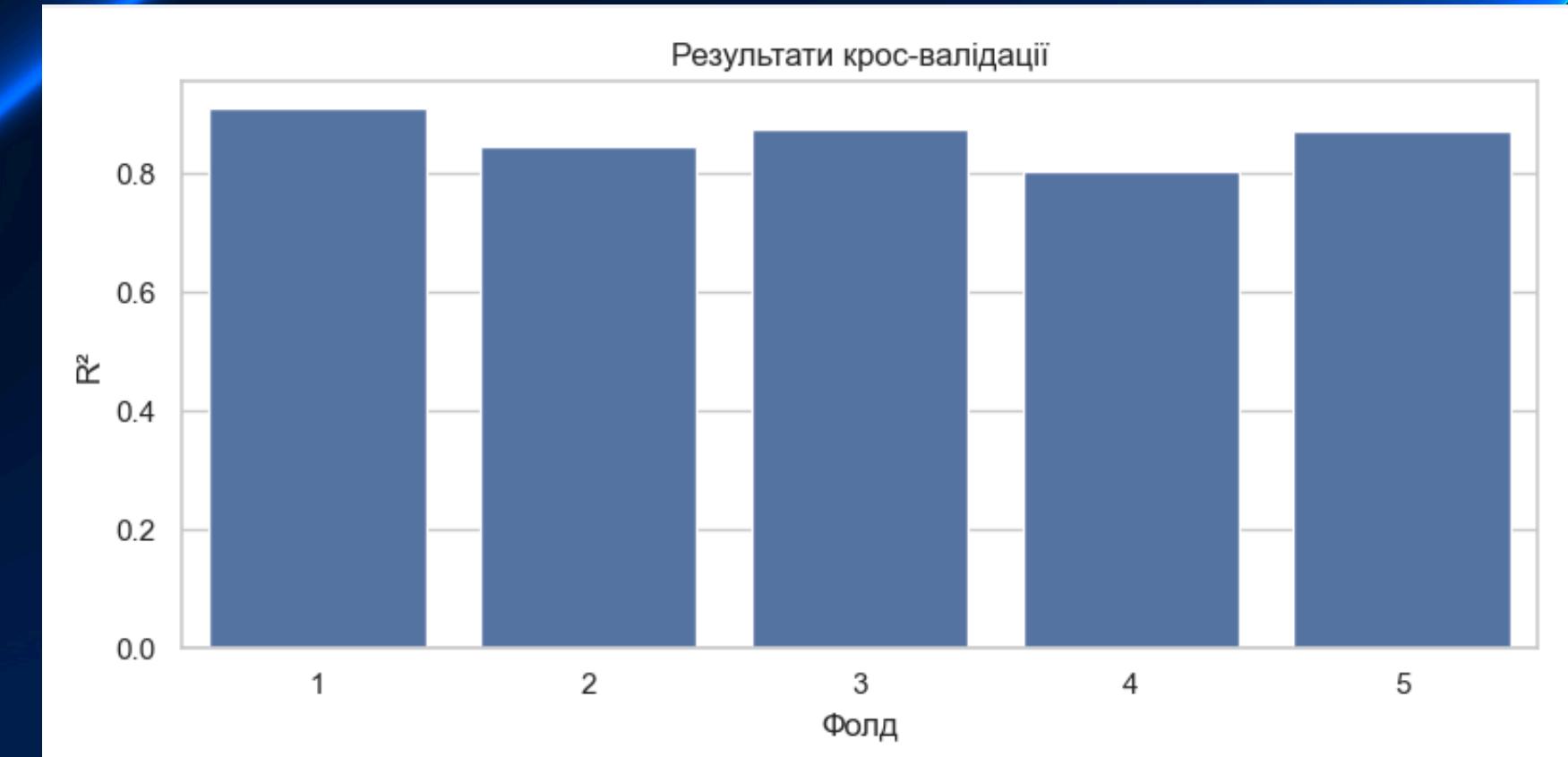
Підбір найкращих гіперпараметрів через GridSearchCV

```
189 grid = GridSearchCV(
190     Pipeline([('pre', preprocessor), ('model', RandomForestRegressor(random_state=42))]),
191     param_grid, cv=5, scoring='r2', n_jobs=-1
192 )
193 grid.fit(X_train, y_train)
194
195 print("Найкращі параметри GridSearchCV:", grid.best_params_)
196 print(f"R2 під час крос-валідації (train): {grid.best_score_:.4f}")
197
198 use_best = grid.best_score_ > base_r2
199 print("Використовуємо підібрану модель?", use_best)
200
201 final_model = grid.best_estimator_ if use_best else base_model
```



# Побудова та оцінка моделі

```
203 # з Оцінка на тестовій вибірці та крос-валідації
204 y_pred_final = final_model.predict(X_test)
205 print(f"Фінальна модель: R2 на тесті = {r2_score(y_test, y_pred_final):.4f}")
206
207 cv = KFold(n_splits=5, shuffle=True, random_state=42)
208 cv_scores = cross_val_score(final_model, X, y, cv=cv, scoring='r2', n_jobs=-1)
209 print("R2 для кожного фолду:", np.round(cv_scores, decimals=4))
210 print(f"Середній R2: {cv_scores.mean():.4f}")
211
212 # Візуалізація результатів крос-валідації
213 plt.figure(figsize=(8, 4))
214 sns.barplot(x=np.arange(1, len(cv_scores)+1), y=cv_scores)
215 plt.xlabel("Фолд")
216 plt.ylabel("R2")
217 plt.title("Результати крос-валідації")
218 plt.tight_layout()
219 plt.show()
```



R<sup>2</sup> для кожного фолду: [0.9084 0.8464 0.8746 0.8022 0.8724]

Середній R<sup>2</sup>: 0.8608



# Аналіз важливості ознак

```
224 final_model.fit(X, y)
225 rf = final_model.named_steps['model']
226 cat_cols = final_model.named_steps['pre'].named_transformers_['cat'].named_steps['onehot'].get_feature_names_out(cat_features)
227 all_features = np.concatenate([num_features, cat_cols])
228
229 imp_df = pd.DataFrame({
230     'Ознака': all_features,
231     'Важливість': rf.feature_importances_
232 }).sort_values(by='Важливість', ascending=False)
233
234 print("\nТоп важливих ознак:\n", imp_df.head(10))
235
236 # Групування ознак (особливо other_colors)
237 groups = {k: 0 for k in num_features + ['other_colors']}
238 for _, row in imp_df.iterrows():
239     for key in groups:
240         if row['Ознака'].startswith(key):
241             groups[key] += row['Важливість']
242             break
243
244 group_df = pd.DataFrame.from_dict(groups, orient='index', columns=['Важливість'])
245 group_df['Важливість (%)'] = 100 * group_df['Важливість'] / group_df['Важливість'].sum()
246 print("\nЗгрупована важливість ознак:\n", group_df.sort_values(by='Важливість', ascending=False))
```

Топ важливих ознак:		
	Ознака	Важливість
2	width	0.623749
0	depth	0.185606
5	designer_price	0.064339
4	desc_length	0.044590
1	height	0.042373
6	category_price	0.016883
3	name_length	0.012422
7	other_colors_No	0.005359
8	other_colors_Yes	0.004679

Згрупована важливість ознак:		
	Важливість	Важливість (%)
width	0.623749	62.374927
depth	0.185606	18.560575
designer_price	0.064339	6.433896
desc_length	0.044590	4.459007
height	0.042373	4.237334
category_price	0.016883	1.688313
name_length	0.012422	1.242165
other_colors	0.010038	1.003783



# Завдання по EXCEL

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
№	Прізвище	Ім'я	По батькові	Дата народження	Стать	Вік на момент прийому на роботу	Пенсійний вік	Дата прийому на роботу	Дата звільнення	Стаж	Причина звільнення	Відділи		
99	ВЕЛИЧКО	САМІЙЛО	ВАСИЛЬОВИЧ	17.07.1964	чоловіча	42	ні	20.03.2007		15р.9м.12д.		Бухгалтерія		
48	ГОНЧARENKO	НІНА	ІВАНІВНА	18.01.1979	жіноча	34	ні	08.05.2013		9р.7м.24д.		Бухгалтерія		
13	ІВАНОВА	АНТОНІНА	МИКОЛАЇВНА	22.04.1960	жіноча	54	так	04.12.2014		8р.0м.28д.		Бухгалтерія		
9	КОВАЛЬ	ОЛЕКСАНДР	ІВАНОВИЧ	14.07.1966	чоловіча	46	ні	09.06.2013		9р.6м.23д.		Бухгалтерія		
62	КОВТУН	ІЛЛЯ	ЮРІЙОВИЧ	17.09.1946	чоловіча	63	так	19.08.2010		12р.4м.13д.		Бухгалтерія		
46	ЛІТВІНЕНКО	МАРІЯ	ІВАНІВНА	04.07.1965	жіноча	44	ні	05.11.2009		13р.1м.27д.		Бухгалтерія		
77	МАКАРЕНКО	АНТОН	СЕМЕНОВИЧ	17.05.1991	чоловіча	21	ні	26.09.2012	31.07.2018	5р.10м.5д.	за власним бажанням	Бухгалтерія		
84	МАРЧУК	ЄВГЕН	КИРИЛОВИЧ	14.06.1979	чоловіча	32	ні	09.10.2011	28.07.2016	4р.9м.19д.	за власним бажанням	Бухгалтерія		

У стовпці G вкажіть гендерну принадлежність працівників

```
=IF(RIGHT(E18;3)="вич";"чоловіча";"жіноча")
```

У стовпці H вкажіть вік працівників на дату прийому на роботу (кількість повних років)

```
=DATEDIF(F18;J18;"Y")
```

У стовпці I вкажіть, чи досяг працівник пенсійного віку на момент прийому на роботу, якщо пенсійний вік становить 60 років.

```
=IF(DATEDIF(F18;$F$14;"Y")>=60;"так";"ні")
```

Заповніть стовпець L; якщо в стовпці K дата звільнення відсутня, то розрахуйте стаж на дату звіту (клітинка F14).

```
=DATEDIF(J18;IF(K18="";$F$14;K18);"Y") & "р." &  
DATEDIF(J18;IF(K18="";$F$14;K18);"  
YM") & "м." &  
DATEDIF(J18;IF(K18="";$F$14;K18);"  
MD") & "д."
```

7. Додайте стовпець "Відділ", в який додайте відповідні значення для кожного співробітника з аркуша "Відділи".

```
=XLOOKUP(TRIM(C18 & " " & D18 & " " & E18);  
Відділи!B:B; Відділи!C:C; "")
```



# Завдання по EXCEL

№	Показник	Значення
1	Кількість працюючих на дату звіту	39
2	Кількість звільнених на дату звіту	61
3	Кількість звільнених на 01.01.2009	3
4	Середній вік звільнених	45
5	Працівників якої статі більше звільнили	Чоловіча
6	Кількість співробітників із прізвищем, яке починається на "Р"	4
7	Кількість причин звільнення	61
8	Найчастіша причина звільнення	за власним бажанням

1. =COUNT(J18:J117)-COUNT(K18:K117)
2. =COUNT(K18:K117)
3. =COUNTIF(K18:K117; "<="&DATE(2009;1;1))
4. =ROUND(AVERAGE(IF(K18:K117<>""; DATEDIF(F18:F117; K18:K117; "Y") + DATEDIF(F18:F117; K18:K117; "YM")/12)); 0)
5. =IF(COUNTIFS(K18:K117; "<>"; G18:G117; "жіноча") > COUNTIFS(K18:K117; "<>"; G18:G117; "чоловіча"); "Жіноча"; "Чоловіча")
6. =COUNTIF(C18:C117; "Р\*")
7. =COUNTA(M18:M117)
8. =INDEX(SORTBY(UNIQUE(FILTER(M18:M117; M18:M117<>""))); -COUNTIF(M18:M117; UNIQUE(FILTER(M18:M117; M18:M117<>"")))); 1)

B	C	D	E	F	G	H	I	J	K	L	M	N
№	Прізвище	Ім'я	По батькові	Дата народження	Стать	Вік на момент прийому на роботу	Пенсійний вік	Дата прийому на роботу	Дата звільнення	Стаж	Причина звільнення	Відділи
17	ВЕЛИЧКО	САМІЙЛО	ВАСИЛЬОВИЧ	17.07.1964	чоловіча	42	ні	20.03.2007		15р.9м.12д.		Бухгалтерія
18												





**Country General Manager Email**

Country	General Manager	Email
Argentina	Facundo Gonzalez	<a href="mailto:f.gonzalez@mcdonalds.com">f.gonzalez@mcdonalds.com</a>
Colombia	Radamel Lopez	<a href="mailto:r.lopez@mcdonalds.com">r.lopez@mcdonalds.com</a>
Brazil	Joao Silva	<a href="mailto:j.silva@mcdonalds.com">j.silva@mcdonalds.com</a>
Ecuador	Jaime Lomo	<a href="mailto:j.lomo@mcdonalds.com">j.lomo@mcdonalds.com</a>
Peru	Samuel Armando	<a href="mailto:s.armando@mcdonalds.com">s.armando@mcdonalds.com</a>
Chile	Alvaro Sanchez	<a href="mailto:a.sanchez@mcdonalds.com">a.sanchez@mcdonalds.com</a>
Bolivia	Angel Garcia	<a href="mailto:a.garcia@mcdonalds.com">a.garcia@mcdonalds.com</a>

**KPIs**

	Sales (M)	Amount	Profit	Amount	Customers	Amount
Actual	\$ 2 544		Actual	\$ 890	Actual	87,0
Target	\$ 3 000		Target	\$ 1 000	Target	100,0
% Complete	85%		% Complete	89%	% Complete	87%
Remainder	15%		Remainder	11%	Remainder	13%

**Sales**

Figures in \$M	2021	2022	Sales by country	Figures in \$M	Customer Satisfaction	
Jan	201,9	215,3	Argentina	953,3	Speed (54%)	54%
Feb	204,2	217,6	Colombia	432,4	Quality (86%)	86%
Mar	198,6	220,1	Brazil	553,2	Hygiene (93%)	93%
Apr	199,2	206,4	Ecuador	445,1	Service (53%)	53%
May	206,4	204,3	Peru	425,1	Availability (95%)	95%
Jun	195,3	203	Chile	253,6		
Jul	192,4	201,5	Bolivia	387,5		
Aug	186,3	200,6				
Sep	194,2	210,6				
Oct	199	216,4				
Nov	205,2	222,3				
Dec	204,3	225,8				