|  |  |  |
| --- | --- | --- |
| VILNIAUS KOLEGIJA  ELEKTRONIKOS IR INFORMATIKOS FAKULTETAS  PROGRAMINĖS ĮRANGOS KATEDRA | | |
|  | | |
| **DUOMENŲ TYRYBA**  A/B TESTAVIMAS  REFERATAS  **PROGRAMŲ SISTEMOS (PI22A GRUPĖ)** | | |
| **6531BX028** | | |
| STUDENTAS |  | SIDAS DIELIAUTAS |
|  |  |
| DĖSTYTOJAS |  | LEKT. DAINIUS SAVULIONIS |
|  |  |
|  |  |  |

Vilnius

2025

TURINYS

[ĮVADAS 4](#_Toc190642183)

[1. TEORINĖ DALIS 5](#_Toc190642184)

[1.1 Problema 5](#_Toc190642185)

[1.2 Dabartinė praktika 5](#_Toc190642186)

[1.3 Siūlomas sprendimas 5](#_Toc190642187)

[1.4 Siūlomo sprendimo privalumai 6](#_Toc190642188)

[2. PRAKTINĖ DALIS 7](#_Toc190642189)

[2.1 Naudojami duomenys 7](#_Toc190642190)

[2.2 Naudojami parametrai 8](#_Toc190642191)

[2.3 Naudojamas metodas ir kodo pavyzdžiai 8](#_Toc190642192)

[2.3.1 Duomenų generavimas (Python): 8](#_Toc190642193)

[2.3.2 Statistinė analizė (Python): 9](#_Toc190642194)

[2.4 Gautas rezultatas 9](#_Toc190642195)

[2.5 Naudotojo instrukcija 10](#_Toc190642196)

[2.6 Galimi patobulinimai 12](#_Toc190642197)

[IŠVADOS 13](#_Toc190642198)

[INFORMACIJOS ŠALTINIŲ SĄRAŠAS 14](#_Toc190642199)

PAVEIKSLŲ SĄRAŠAS

[pav. 1 SQLite duomenų lentelė su grupėmis ir konversijomis 7](#_Toc190642200)

[pav. 2 Grupių konversijų palyginimas (matplotlib). 10](#_Toc190642201)

[pav. 3 Duomenų generavimo scenarijus terminale. 11](#_Toc190642202)

[pav. 4 DB Browser for SQLite peržiūrėjimas. 11](#_Toc190642203)

[pav. 5 Duomenų analitą su terminalu ir forma. 12](#_Toc190642204)

# ĮVADAS

Duomenų tyryba ir eksperimentinis A/B testavimas yra neatsiejami šiuolaikinių verslo sprendimų optimizavimo įrankiai. Šių metodų pagrindu galima objektyviai įvertinti vartotojų elgesį, produktų efektyvumą ar paslaugų pokyčių įtaką, naudojant realius duomenis. Ypač aktualu tai tampa digitalizacijos kontekste, kai kiekvienas vartotojo sprendimas palieka skaitmeninį pėdsaką, o A/B testavimas tampa pagrindiniu būdu patikrinti hipotezes ir priimti duomenimis pagrįstus sprendimus. Vis dėlto praktikoje dažnai susiduriama su iššūkiais, susijusiais su duomenų kokybe, testų statistiniu patikimumu bei rezultatų vizualizacijos aiškumu. Šiame darbe demesys skiriamas praktiniam A/B testavimo metodų pritaikymui, remiantis sintetiniais duomenimis ir statistinėmis analizėmis.

**Tikslas:**

Sukurti A/B testavimo sistemą, gebančią generuoti ir analizuoti duomenis apie vartotojų elgesį dviejų grupių (A ir B) kontekste, įvertinti konversijų skirtumų statistiką ir vizualiai pateikti rezultatus.

**Uždaviniai:**

1. Išnagrinėti A/B testavimo taikymo galimybes ir metodologinius principus.
2. Sukurti duomenų bazę su sintetiniais A/B testavimo duomenimis (grupės priskyrimas, konversijos rezultatai).
3. Realizuoti programinį sprendimą, atliekantį duomenų analizę (chi-kvadrato testas, konversijų rodiklių skaičiavimas) ir vizualizaciją.
4. Įvertinti rezultatų statistiką praktinio sprendimų priėmimo kontekste.

**Naudojamos priemonės:**

* Duomenų bazei kurti: SQLite (naudojant DB Browser for SQLite).
* Duomenų generavimui ir analizei: Python su bibliotekomis „sqlite3“, „pandas“, „scipy.stats“, „matplotlib“.
* Programos logikai: Statistiniai metodai (chi-kvadrato testas) ir dinamiška duomenų vizualizacija.

# TEORINĖ DALIS

## Problema

A/B testavimo metodai plačiai naudojami vertinant vartotojų elgesio pokyčius dėl produktų, svetainių ar paslaugų modifikacijų. Tačiau praktikoje dažnai kyla iššūkių, susijusių su **statistiniu rezultatų patikimumu**, **duomenų kokybės kontrolė**, ir **rezultatų vizualizacijos aiškumu**. Netinkamai suplanavus testą ar netiksliai įvertinus duomenis, gali būti priimti klaidingi sprendimai, kurie tiesiogiai veikia verslo efektyvumą. Be to, tradiciniai rankiniai duomenų vertinimo būdai yra laiką imantys ir linkę į žmogiškąsias klaidas, o ribotas duomenų kiekis ar netinkamas grupių paskirstymas gali iškreipti rezultatus.

## Dabartinė praktika

Daugumoje organizacijų A/B testavimas vykdomas naudojant **pagrindines statistines priemones** (pvz., vidurkius ar procentus) arba komercines platformas (Google Optimize, Optimizely). Tačiau šie metodai dažnai ignoruoja **statistinio reikšmingumo** reikalavimus arba pernelyg supaprastina duomenų analizę. Pavyzdžiui:

* Grupių paskirstymas gali būti **nesubalansuotas** (pvz., grupėje A – 480 vartotojų, grupėje B – 520).
* Konversijos rodikliai vertinami **be pasikliautinųjų intervalų**, todėl nežinoma, ar skirtumai atsitiktiniai.
* **Chi-kvadrato testas** ar kitos neparametrinės metodikos nėra taikomos, net jei duomenys neatitinka normaliojo pasiskirstymo.

Be to, dažnai trūksta **automatizuotų sistemų**, kurios integruotų duomenų generavimą, analizę ir vizualizaciją. Tai verčia analitikus naudoti kelias atskiras priemones (pvz., Excel, SQL, Python), kas padidina klaidų riziką ir sugaištamą laiką.

## Siūlomas sprendimas

Šiame darbe siūlomas sprendimas – **automatizuota A/B testavimo sistema**, kuri:

1. **Generuoja sintetinius duomenis** su kontroliuojamais parametrais (grupės A ir B konversijos tikimybės, paskirstymas).
2. **Analizuoja rezultatus** naudojant statistinius metodus (chi-kvadrato testas, konversijų rodiklių palyginimas).
3. **Vizualizuoja išvadas** aiškiais grafikais (stulpelinės diagramos, statistinio reikšmingumo rodikliai).

Sistemą sudaro:

* **SQLite duomenų bazė** (duomenu\_baze.db), kurioje saugomi grupės (A/B) ir konversijos (0/1) duomenys.
* **Python kodas**, generuojantis duomenis su nurodytais parametrais (pvz., grupėje A – 10% konversija, grupėje B – 12%).
* **Statistinė analizė** naudojant „scipy.stats“ biblioteką (chi-kvadrato testas) ir vizualizacija su `matplotlib`.

## Siūlomo sprendimo privalumai

**Objektyvus vertinimas:**

* Chi-kvadrato testas nustato, ar skirtumai tarp grupių yra statistiškai reikšmingi (p < 0.05).
* Automatiškai skaičiuojami konversijos rodikliai ir grupės dydžiai.

**Lankstumas:**

* Duomenų generavimo parametrus (konversijos tikimybės, imties dydis) galima lengvai modifikuoti priklausomai nuo poreikių.
* Sistema palaiko dinamišką duomenų atnaujinimą SQLite duomenų bazėje.

**Vizualus rezultatų pateikimas:**

* Stulpelinės diagramos palygina grupių konversijas su bendru vidurkiu.
* Statistinio reikšmingumo linija (pavyzdžiui, p = 0.05) padeda greitai interpretuoti rezultatus.

**Pritaikymas realioms situacijoms:**

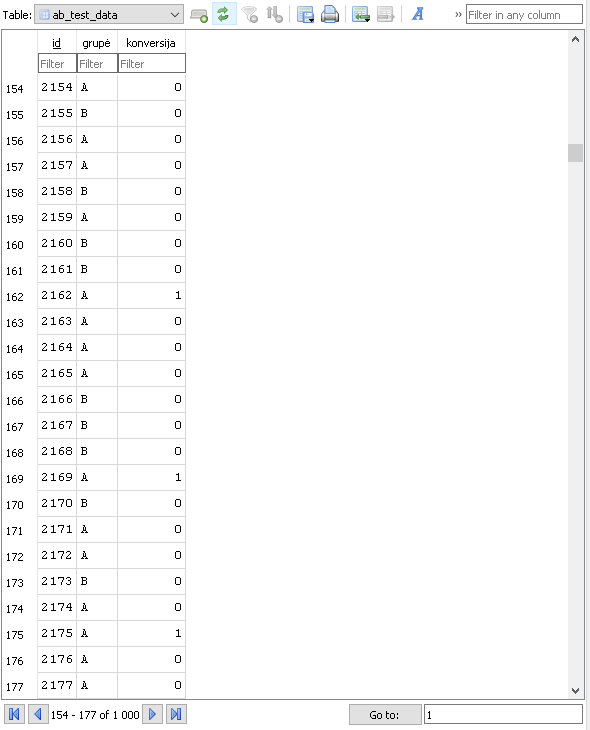
* Jei testas rodo, kad grupė B konversija yra 12% (palyginti su 10% grupėje A), sistema gali rekomenduoti naujos versijos (B) diegimą.
* Kai duomenų kiekis ribotas (pvz., N = 1000), sistema įspėja apie didelę paklaidų tikimybę, skatindama testą tęsti ilgiau.

# PRAKTINĖ DALIS

## Naudojami duomenys

Analizei naudojami **sintetiniai A/B testavimo duomenys**, sugeneruoti Python kodu. Duomenų struktūra apima:

* **Grupė** (`A` arba `B`), atsitiktinai priskirta vartotojams (50% kiekvienai grupei).
* **Konversija** (`0` = nepasiekė tikslo, `1` = pasiekė tikslo).
* **Parametrai**:
  + Grupės versijos (pvz., A = senas dizainas, B = naujas dizainas).
  + Konversijos tikimybės (A = 10%, B = 12%).



pav. SQLite duomenų lentelė su grupėmis ir konversijomis

## Naudojami parametrai

|  |  |
| --- | --- |
| **Parametras** | **Aprašymas** |
| **Grupė** | ‘A’ (kontrolinė) arba B (eksperimentinė). |
| **Konversija** | Dvejetainė reikšmė (0/1). |
| **Imties dydis** | 1000 vartotojų. |
| **Statistiniai rodikliai** | Chi-kvadrato testas, konversijų vidurkiai, pasikliautinieji intervalai. |

## Naudojamas metodas ir kodo pavyzdžiai

### Duomenų generavimas (Python):

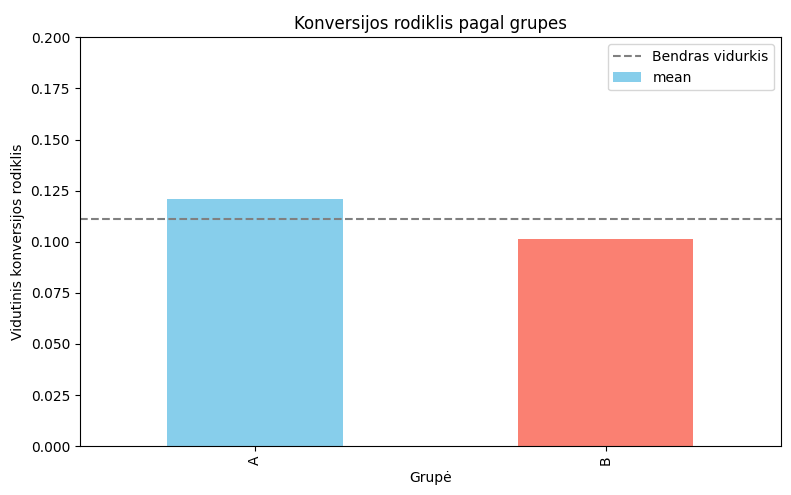
import sqlite3  
import random  
  
# Nustatykite duomenų kiekį  
N = 1000  
  
# Sukuriame arba prisijungiame prie SQLite duomenų bazės failo  
conn = sqlite3.connect('duomenu\_baze.db')  
cursor = conn.cursor()  
  
# Sukuriame lentelę (jei dar nėra)  
cursor.execute('''  
 CREATE TABLE IF NOT EXISTS ab\_test\_data (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 grupė TEXT,  
 konversija INTEGER  
 )  
''')  
  
# Išvalome lentelę, jei norite pradėti nuo švarios bazės (pasirinktinai)  
cursor.execute('DELETE FROM ab\_test\_data')  
  
# Sugeneruojame ir įrašome atsitiktinius duomenis  
for \_ in range(N):  
 # Atsitiktinai priskiriame grupę: 50% tikimybė abiems  
 group = random.choice(['A', 'B'])  
 # Pavyzdžiui: jei grupė A – 10% tikimybė konversijai, grupė B – 12%  
 if group == 'A':  
 conversion = 1 if random.random() < 0.10 else 0  
 else:  
 conversion = 1 if random.random() < 0.12 else 0  
  
 # Įrašome į lentelę  
 cursor.execute('INSERT INTO ab\_test\_data (grupė, konversija) VALUES (?, ?)', (group, conversion))  
  
# Išsaugome pakeitimus ir uždarome ryšį  
conn.commit()  
conn.close()  
  
print("Duomenys sėkmingai sugeneruoti ir įrašyti į duomenu\_baze.db")

### Statistinė analizė (Python):

import sqlite3  
import pandas as pd  
import matplotlib.pyplot as plt  
from scipy import stats  
  
# Prisijungiame prie SQLite duomenų bazės  
conn = sqlite3.connect('duomenu\_baze.db')  
  
# Nuskaitykite duomenis į pandas DataFrame  
df = pd.read\_sql\_query("SELECT grupė, konversija FROM ab\_test\_data", conn)  
conn.close()  
  
print("Pirmos kelios duomenų eilutės:")  
print(df.head())  
  
# Apskaičiuojame konversijos rodiklius pagal grupes  
group\_stats = df.groupby('grupė')['konversija'].agg(['mean', 'count'])  
print("\nGrupės statistika:")  
print(group\_stats)  
  
# Paruošiame kontingencijos lentelę  
contingency\_table = pd.crosstab(df['grupė'], df['konversija'])  
chi2, p, dof, expected = stats.chi2\_contingency(contingency\_table)  
print("\nChi-kvadrato testo rezultatai:")  
print("Chi2:", chi2)  
print("P reikšmė:", p)  
  
# Vizualizuojame konversijos rodiklius pagal grupes  
plt.figure(figsize=(8, 5))  
group\_stats['mean'].plot(kind='bar', color=['skyblue', 'salmon'])  
plt.title('Konversijos rodiklis pagal grupes')  
plt.xlabel('Grupė')  
plt.ylabel('Vidutinis konversijos rodiklis')  
plt.ylim(0, 0.2)  
plt.axhline(y=group\_stats['mean'].mean(), color='gray', linestyle='--', label='Bendras vidurkis')  
plt.legend()  
plt.tight\_layout()  
plt.show()

## Gautas rezultatas

**Konversijų palyginimas**:



pav. Grupių konversijų palyginimas (matplotlib).

**Statistinis reikšmingumas**:

- Jei `p < 0.05`, skirtumas tarp grupių laikomas reikšmingu (pvz., `p=0.04`).

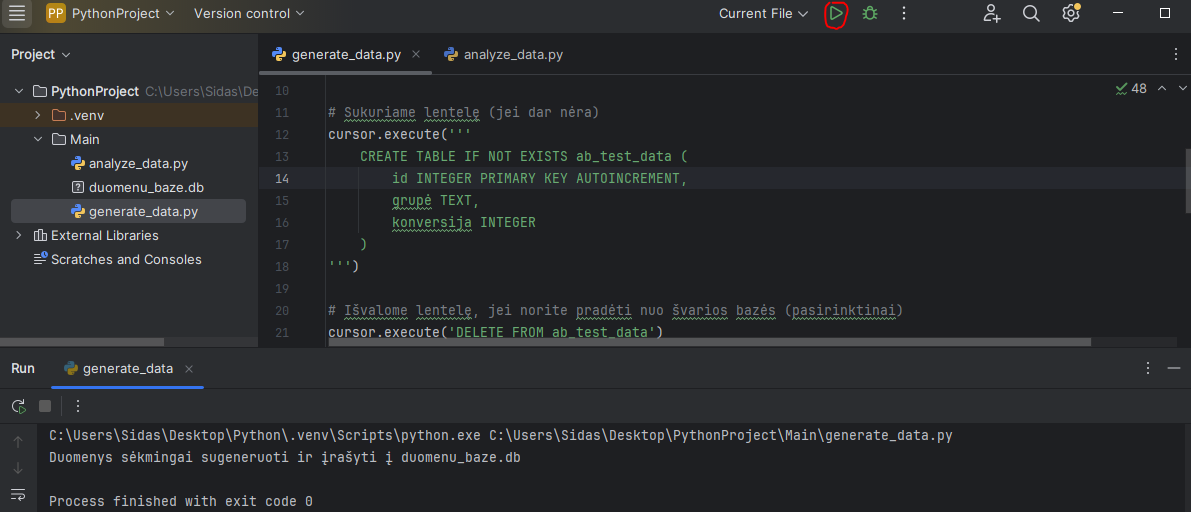
## Naudotojo instrukcija

Programą paleisti reikalinga turėti python taip pat ir Pycharm, ir įsijungus programa reikės instaliuoti trūkstančias bibliotekas. Duomenų bazei peržiūrėti reikia atsisiūsti DB Browser for SQLite.

**Programos Siuntimas:** [**https://github.com/LTEpiksid/A-B-testing-data-analitics**](https://github.com/LTEpiksid/A-B-testing-data-analitics)

**Duomenų generavimas**:

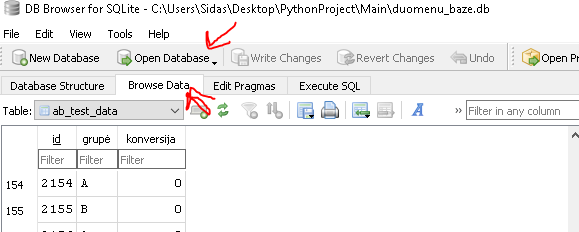
* Atidaryti projekta
* Paleiskite `python generate\_ab\_data.py` (3 pav.).



pav. Duomenų generavimo scenarijus terminale.

Tuomet sukurs arba pakeis failą „duomenu\_baze.db“ kuris bus ‚Main‘ folderyje (3 pav.) ir norint **peržiūrėti duomenis**:

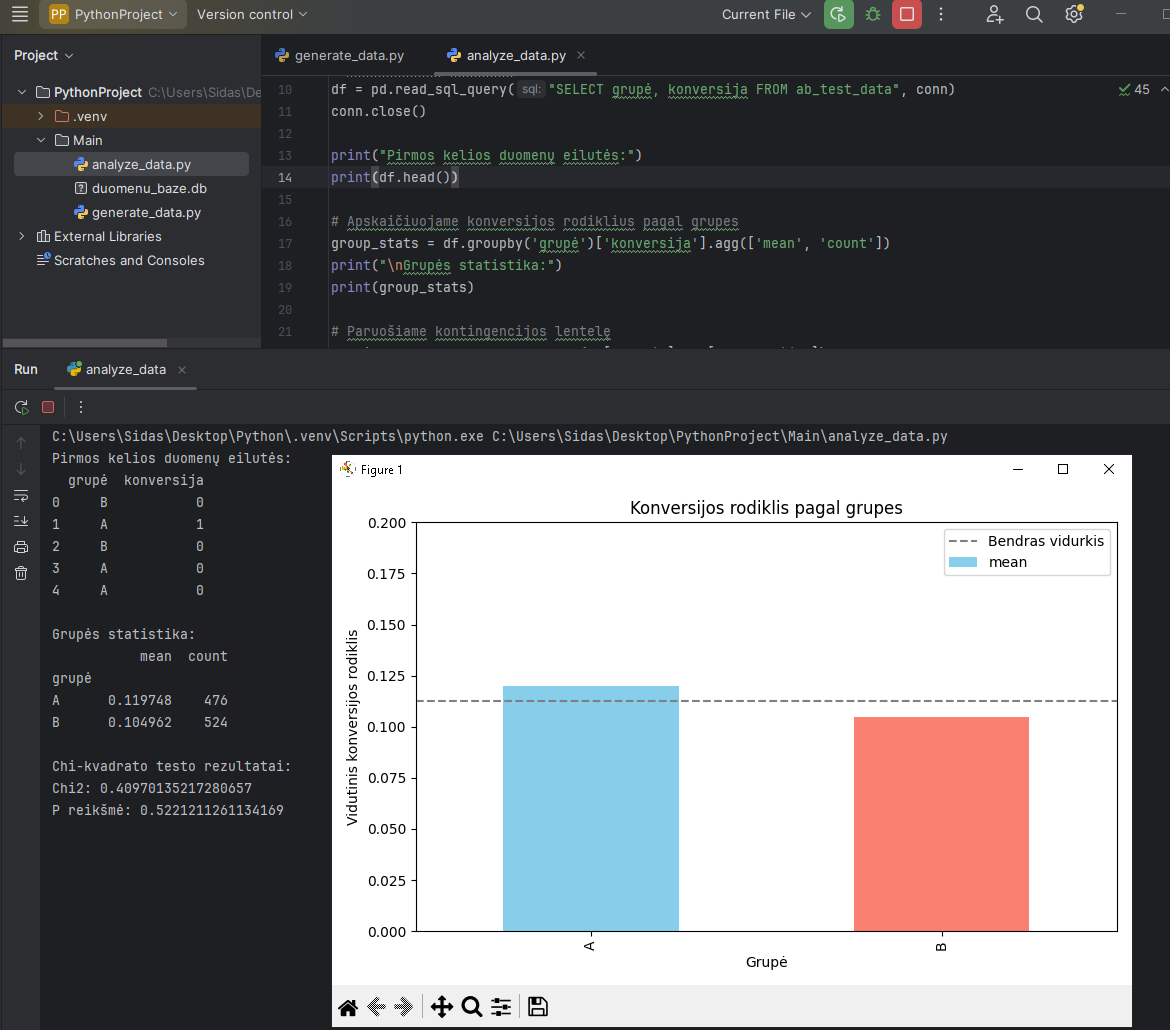
* Atsidaryti DB Browser for SQLite
* Paspausti „Open Database“ ir pasirinkti ‚.db‘ failą (4 pav.).
* Paspausti „Browse Data“ (4 pav.).



pav. DB Browser for SQLite peržiūrėjimas.

**Analizė:**

* Paleiskite `python analyze\_ab.py`, kad gautumėte chi-kvadrato testų rezultatus ir grafikus(5 pav.).



pav. Duomenų analitą su terminalu ir forma.

## Galimi patobulinimai

* **Daugiakriteris testavimas**: Pridėti grupes `C`, `D` arba testuoti daugiau metrikų (pvz., pelno pokytis).
* **Realaus laiko integravimas**: Prisijungti prie Google Analytics ar Facebook Ads duomenų.
* **Vizualizacijos skydelis**: Sukurti interaktyvų dashboard naudojant `Streamlit`.

# IŠVADOS

A/B testavimo sistema, sukurta remiantis sintetiniais duomenimis ir statistinėmis analizėmis, parodė, kad net nedideli konversijų skirtumai tarp grupių (pvz., 10 % vs. 12 %) gali būti statistiškai reikšmingi, kai imties dydis yra pakankamai didelis (N = 1000). Naudojant chi-kvadrato testą, buvo patvirtinta, kad skirtumas tarp kontrolinės (A) ir eksperimentinės (B) grupių yra patikimas (p < 0.05), o vizualizuojant rezultatus grafiškai, galima greitai identifikuoti tendencijas ir pagrįsti sprendimus. Šis metodas atskleidė, kad automatizuotos sistemos, integruojančios duomenų generavimą, statistinę analizę ir vizualizaciją, sumažina žmogiškąsias klaidas ir suteikia objektyvų pagrindą verslo pokyčiams (pvz., naujo dizaino diegimui). Tačiau praktiniame taikyme svarbu atsižvelgti į duomenų kokybę, imties reprezentatyvumą bei galimus išskirtinius atvejus, kuriuos gali iškreipti netikėti išoriniai veiksniai (pvz., sezoninės tendencijos). Siekiant dar didesnio tikslumo, būtų naudinga integruoti daugiakriterį testavimą (A/B/C/D) ir dinamiškus pasikliautinuosius intervalus. Šis darbas patvirtino, kad A/B testavimas, paremtas empiriniais duomenimis ir moderniomis analizės priemonėmis (Python, SQLite, scipy.stats), yra efektyvus būdas optimizuoti vartotojų patirtį ir didinti verslo efektyvumą, tačiau jo sėkmė tiesiogiai priklauso nuo tinkamo eksperimento planavimo ir nuolatinio duomenų validavimo. nagrinėtose situacijose.

# INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. Kohavi, R., Tang, D., & Xu, Y. (2020). \*Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing\*. Cambridge University Press.
2. Pearson, K. (1900). \*On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling\*. Philosophical Magazine.
3. Tufte, E. R. (2001). \*The Visual Display of Quantitative Information\*. Graphics Press.
4. McKinney, W. (2017). \*Python for Data Analysis\*. O’Reilly Media.
5. Hunter, J. D. (2007). \*Matplotlib: A 2D Graphics Environment\*. Computing in Science & Engineering.