МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

WEB-Аналітика

ЛАБОРАТОРНА РОБОТА №2

Виконав

студент групи ФІ-42мн

Бєш Радомир Андрійович

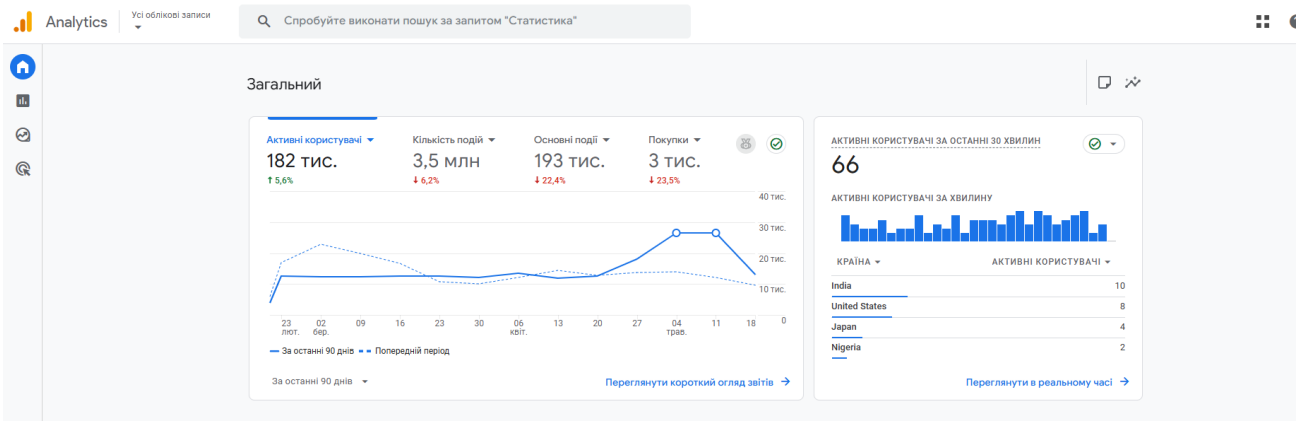
Київ 2025

Завдання:

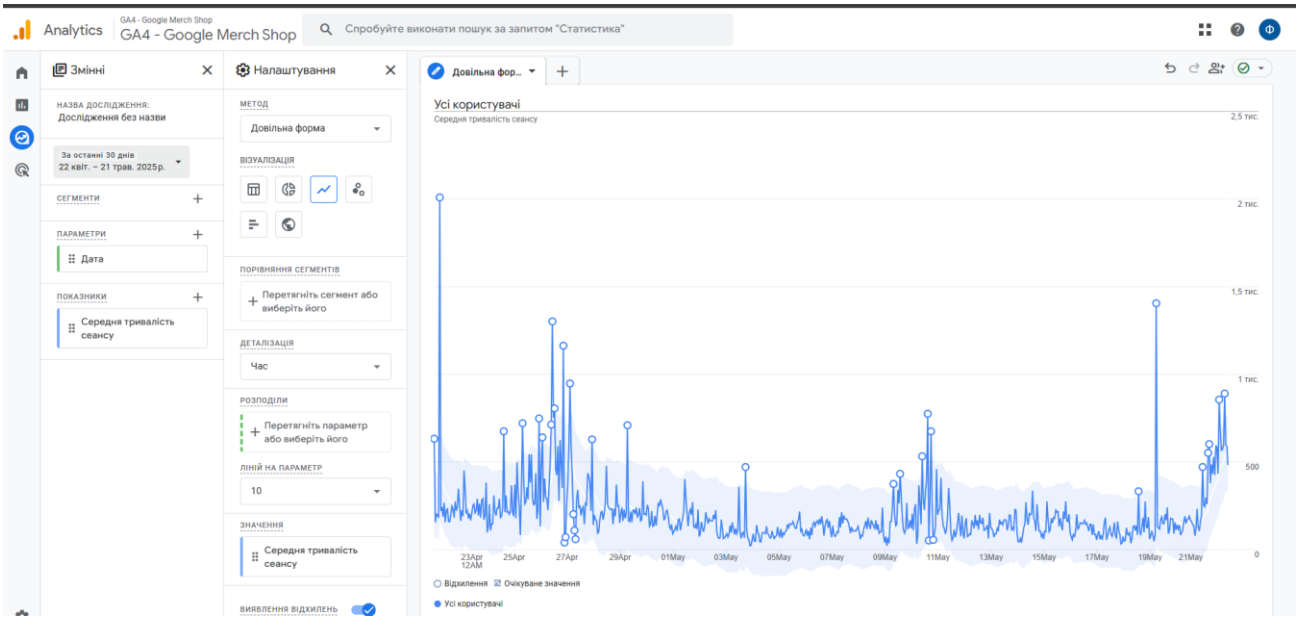
На основі даних з Google Analytics Demo account вибрати ТРИ різні часові ряди і на їх основі шляхом застосування методів визначення аномалій визначити аномалії в поведінці системи.

Хід роботи

1) Першим етапом лабораторної роботи було сформувати та завантажити датасет з демо акаунту в Google Analytics, але на превеликий жаль, сервіс в мене не працював.



Я не міг обрати демо акаунт. Навіть коли я створював нове дослідження я просто не міг завантажити дані в форматі .csv, та взагалі не міг завантажити будь-що.



Тому було прийнято рішення знайти вже сформований датасет на Kaggle.

2) Обраний датасет:

Посилання: https://www.kaggle.com/datasets/bobnau/daily-website-visitors

Початок роботи з датасетом, завантаження , читання , первинна обробка.

```
[2] import kagglehub

    # Download latest version
    path = kagglehub.dataset_download("bobnau/daily-website-visitors")

    print("Path to dataset files:", path)
```

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/bobnau/daily-website-visitors?dataset_version_number=1...
100%|██████████| 34.9k/34.9k [00:00<00:00, 19.5MB/s]Extracting files...
Path to dataset files: /root/.cache/kagglehub/datasets/bobnau/daily-website-visitors/versions/1
```

```
csv_path = os.path.join(path, 'daily-website-visitors.csv')
df = pd.read_csv(csv_path)
df.head()
```

	Row	Day	Day.Of.Week	Date	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
0	1	Sunday	1	9/14/2014	2,146	1,582	1,430	152
1	2	Monday	2	9/15/2014	3,621	2,528	2,297	231
2	3	Tuesday	3	9/16/2014	3,698	2,630	2,352	278
3	4	Wednesday	4	9/17/2014	3,667	2,614	2,327	287
4	5	Thursday	5	9/18/2014	3,316	2,366	2,130	236

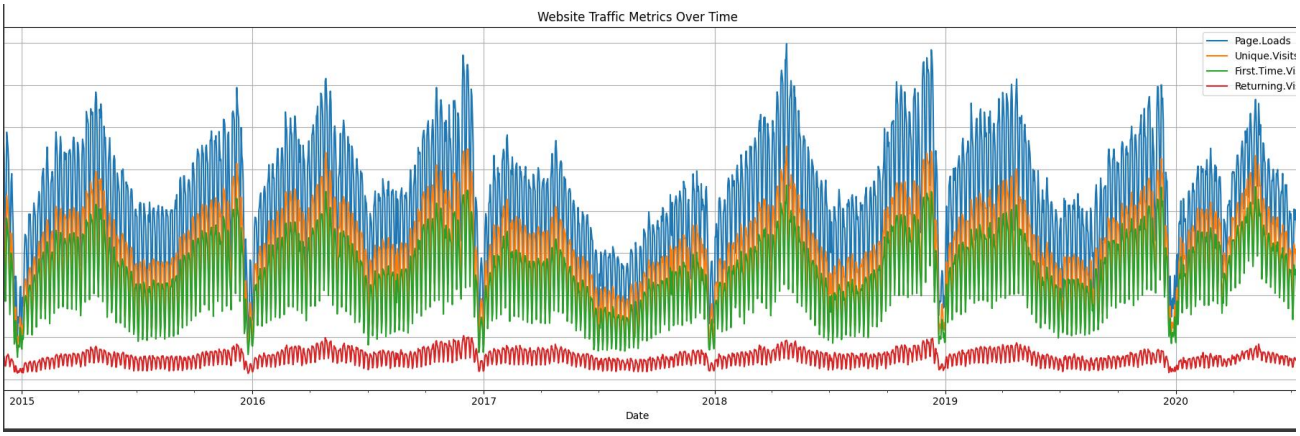
```
[11] df.drop(columns=['Row', 'Day', 'Day.Of.Week'], inplace=True)
```

```
df
```

Date	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
2014-09-14	2,146	1,582	1,430	152
2014-09-15	3,621	2,528	2,297	231
2014-09-16	3,698	2,630	2,352	278
2014-09-17	3,667	2,614	2,327	287
2014-09-18	3,316	2,366	2,130	236
...	...	...	...	...
2020-08-15	2,221	1,696	1,373	323
2020-08-16	2,724	2,037	1,686	351
2020-08-17	3,456	2,638	2,181	457
2020-08-18	3,581	2,683	2,184	499
2020-08-19	2,064	1,564	1,297	267

2167 rows × 4 columns

Датасет містить дані за період з 2015 по 2020 роки.



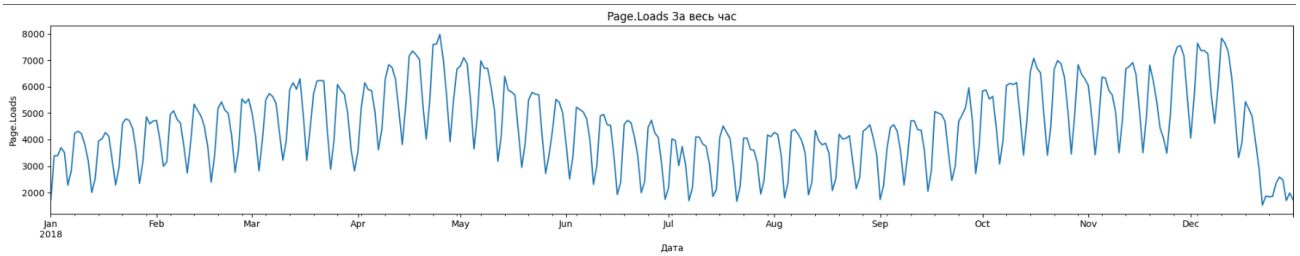
Але для нашого дослідження оберемо часовий проміжок за 2018 рік.

```
start_date = '2018-01-01'
end_date = '2018-12-31'

df_year_2018 = df.loc[start_date:end_date]

df_year_2018.head()
```

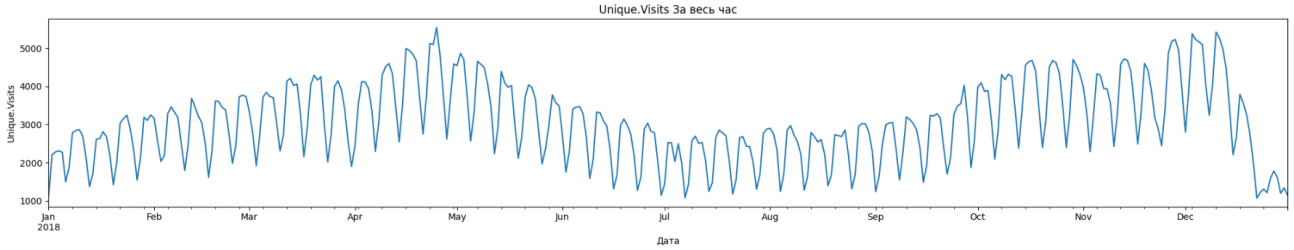
Date	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
2018-01-01	1709	1120	878	242
2018-01-02	3389	2208	1755	453
2018-01-03	3391	2282	1827	455
2018-01-04	3698	2310	1878	432
2018-01-05	3511	2272	1801	471

3) Будуємо графіки трьох часових рядів

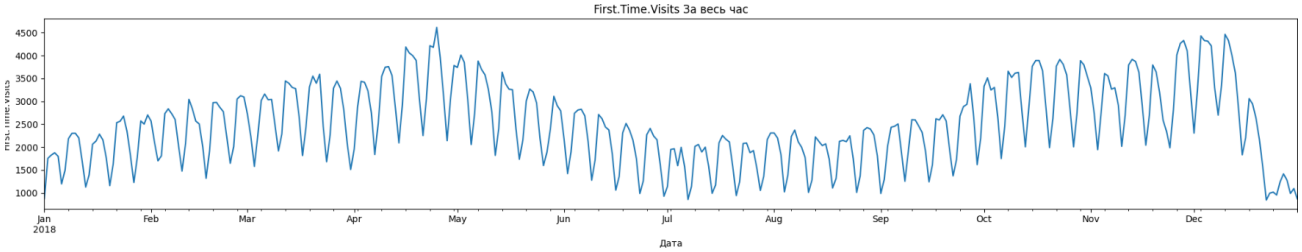
Завантаження сторінки



Унікальні користувачі



Перший візит на сторінку

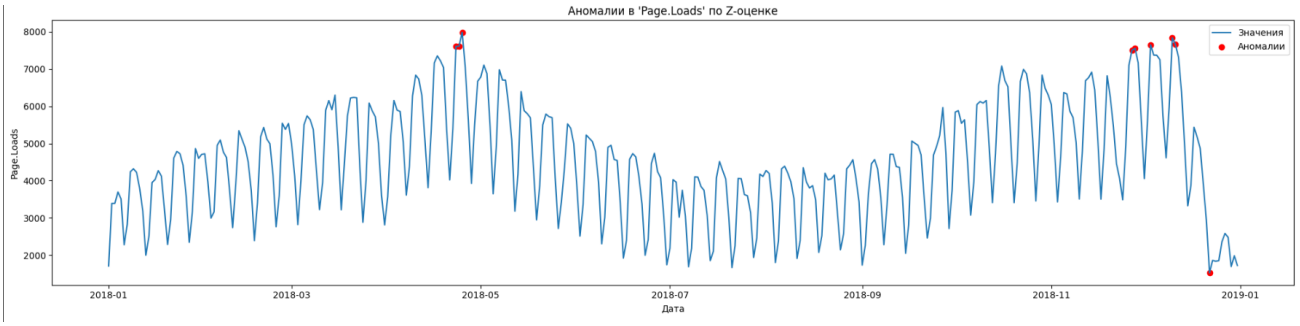


4) Z-Score метод для побудови аномалій

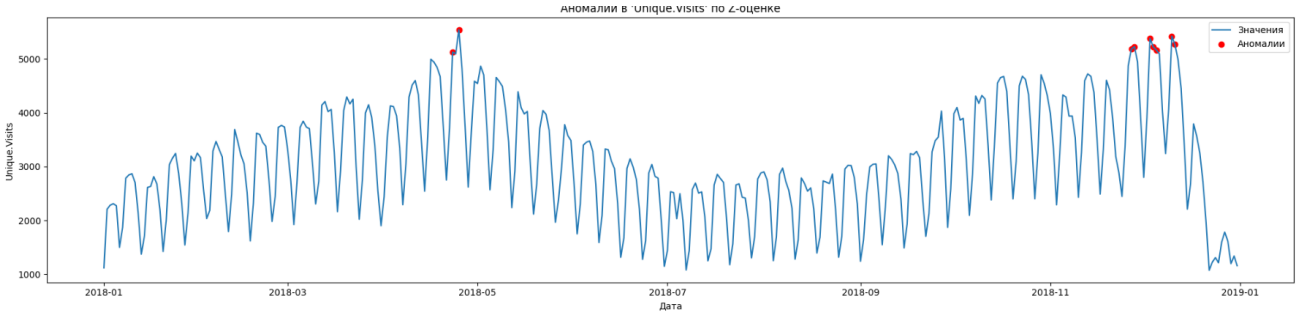
```python
import numpy as np
from scipy.stats import zscore
import matplotlib.pyplot as plt


selected_columns = df_year_2018.columns
threshold = 2

for col in selected_columns:
    z_scores = zscore(df_year_2018[col])

    anomalies = np.abs(z_scores) > threshold

    plt.figure(figsize=(20, 5))
    plt.plot(df_year_2018.index, df_year_2018[col], label='Значення')
    plt.scatter(df_year_2018.index[anomalies], df_year_2018[col][anomalies], color='red', label='Аномалії')
    plt.title(f"Аномалії в '{col}' за Z-оцінкою")
    plt.xlabel("Дата")
    plt.ylabel(col)
    plt.legend()
    plt.tight_layout()
    plt.show()
```

Результати:

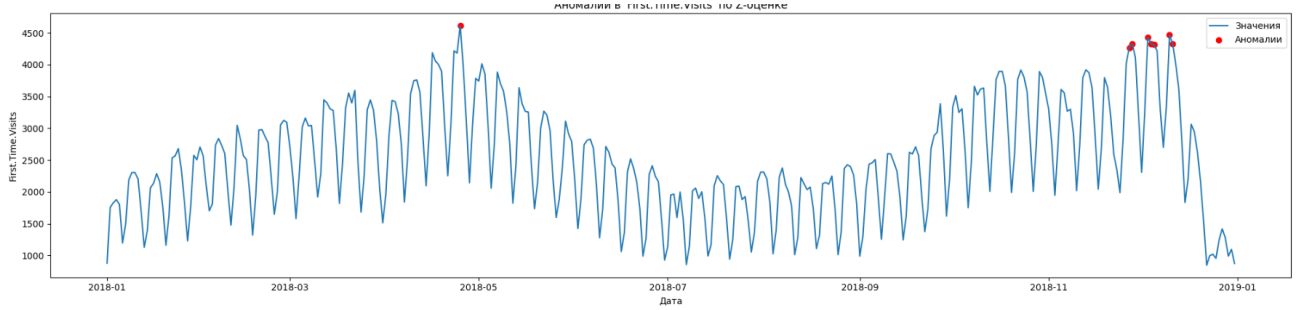
Завантаження сторінки



Унікальні користувачі



Перший візит на сторінку



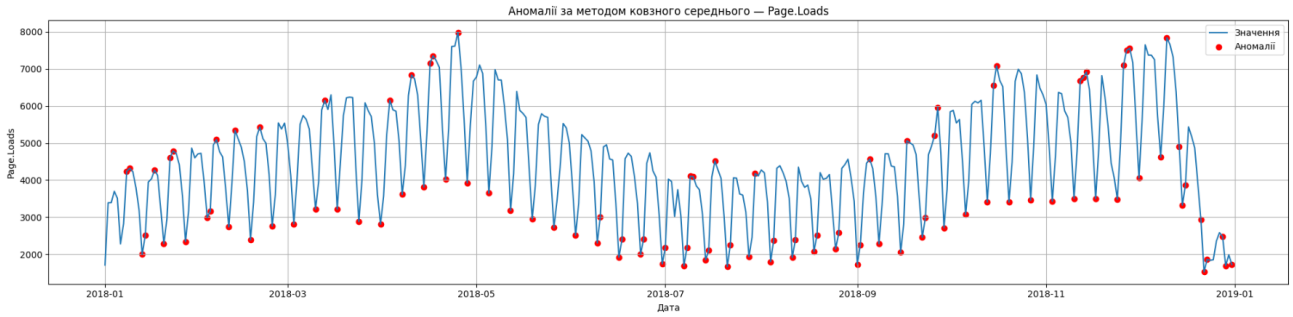
5) Метод Moving Avarage або Ковзне середнє

Обрали вікно в тиждень

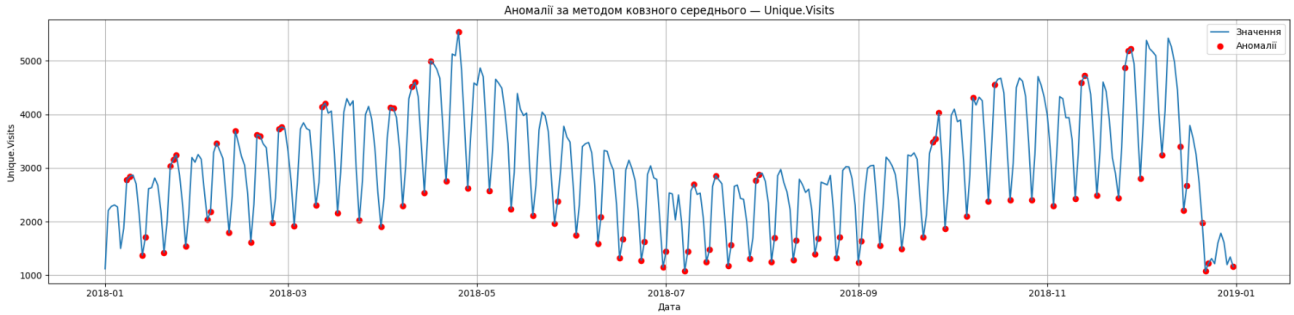
```python
window_size = 7  # 7-дневное скользящее среднее
threshold = 1

for col in df_year_2018.columns:
    series = df_year_2018[[col]].copy()
    series['rolling_mean'] = series[col].rolling(window=window_size).mean()
    series['rolling_std'] = series[col].rolling(window=window_size).std()

    anomalies = series[
        (series[col] > series['rolling_mean'] + threshold * series['rolling_std']) |
        (series[col] < series['rolling_mean'] - threshold * series['rolling_std'])
    ]

    plt.figure(figsize=(20, 5))
    plt.plot(series.index, series[col], label='Значення')
    plt.scatter(anomalies.index, anomalies[col], color='red', label='Аномалії')
    plt.title(f"Аномалії за методом ковзного середнього — {col}")
    plt.xlabel("Дата")
    plt.ylabel(col)
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

Результати:

Завантаження сторінки



Унікальні користувачі



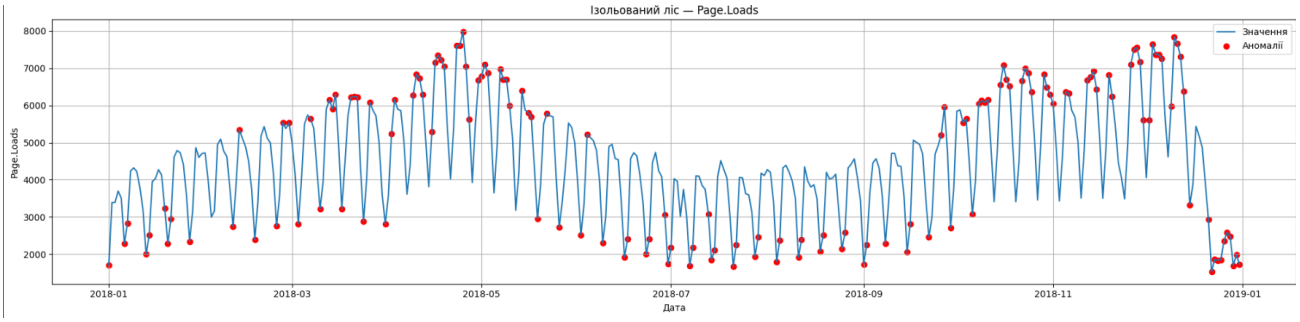
Перший візит на сторінку

6) Метод ізольованого лісу

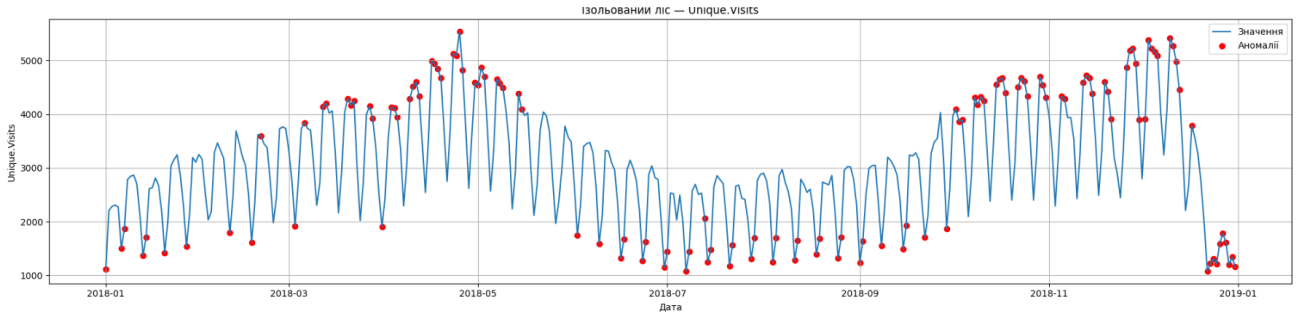
```python
from sklearn.ensemble import IsolationForest

# Функція для побудови графіка з аномаліями
def draw_anomaly(data, anomalies, title):
    prop = data.columns[0]
    plt.figure(figsize=(20, 5))
    plt.plot(data.index, data[prop], label='Значення')
    plt.scatter(anomalies.index, anomalies[prop], color='red', label='Аномалії')
    plt.title(f"{title}")
    plt.xlabel("Дата")
    plt.ylabel(prop)
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()

# Функція Isolation Forest
def isolation_forest(data):
    prop = data.columns[0]
    model = IsolationForest(n_estimators=100, contamination='auto', random_state=42)
    pred = model.fit_predict(data[[prop]])
    return data[pred == -1]


for col in df_year_2018.columns:
    series = df_year_2018[[col]].copy()
    anomalies = isolation_forest(series)
    draw_anomaly(series, anomalies, f"Ізольований ліс – {col}")
```

Результати:

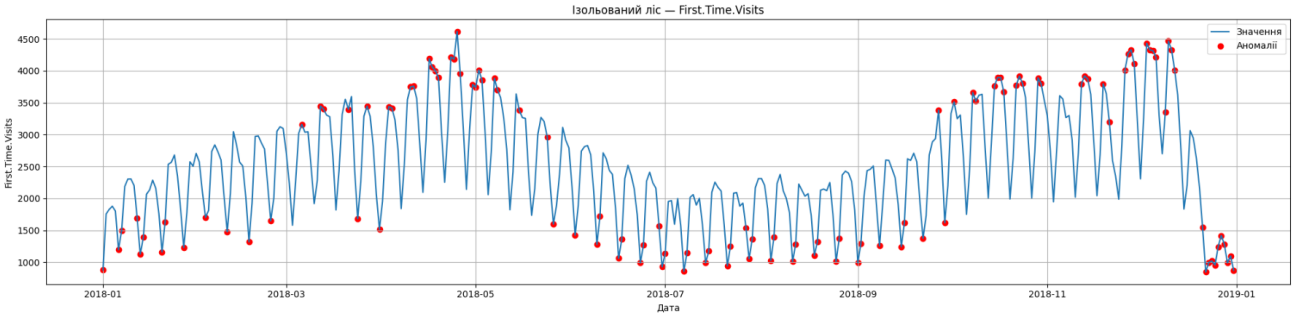
Завантаження сторінки



Унікальні користувачі



Перший візит на сторінку



Висновки: у ході лабораторної роботи було проведено аналіз часових рядів на основі даних із Google Analytics. Було обрано три часові ряди за 2018 рік: завантаження сторінки, унікальні користувачі та перші візити. Для виявлення аномалій застосовувалися три різні методи: Z-Score, ковзне середнє (Moving Average) та ізольований ліс. Метод Z-Score показав себе як чутливий до різких піків і провалів у даних, дозволяючи швидко виявити одиничні аномальні дні. Ковзне середнє із вікном у сім днів забезпечило згладжування рядів і дало можливість виявляти відхилення від локальних трендів. Метод ізольованого лісу, який базується на машинному навчанні, виявив нестандартні структури та зміни в поведінці користувачів, зокрема тривалі спади чи підйоми активності.

Загалом, усі три методи підтвердили наявність аномалій у поведінці користувачів на вебсайті протягом 2018 року. Різні підходи дозволили виявити різні типи аномалій, тому їх комбінація є найбільш ефективною для повного розуміння поведінкових змін. Отримані результати можуть бути корисними для глибшого аналізу користувацької активності, виявлення технічних проблем, впливу маркетингових кампаній або нетипової поведінки, пов’язаної з ботами чи зовнішніми факторами.