МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут **КНІТ**
Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 3
З дисципліни: *"Видобування та опрацювання даних"*
На тему: "Етап підготовки даних за CRISP-DM"

Лектор:
асист. каф. ПЗ
Угриновський Б.В.

Виконав:
ст. гр. ПЗ-45
Хруставчук М.Л.

Прийняв:
асист. каф. ПЗ
Симець І.І.

« ____ » _____ 2025 р.

∑= _____ .....

Львів – 2025

Тема роботи: Етап підготовки даних за CRISP-DM.

Мета роботи: Навчитися базовому функціоналу pandas для обробки та підготовки даних до моделювання та використання.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Етап підготовки даних є ключовою фазою методології CRISP-DM та безпосередньо впливає на якість подальшого моделювання. Його основна мета полягає у формуванні коректного, повного та репрезентативного набору даних, придатного для застосування алгоритмів аналізу та машинного навчання. На цьому етапі виконуються операції очищення, інтеграції, трансформації та відбору ознак.

Очищення даних передбачає виявлення та обробку пропущених значень, дублікатів, помилок введення та аномальних спостережень. Наявність пропусків може суттєво спотворювати статистичні оцінки та результати моделей, тому застосовуються стратегії видалення або імпутації значень залежно від їх частки та типу змінної. Аналогічно, дублікати записів можуть викликати перекіс розподілу та повинні бути перевірені на доцільність збереження.

Важливою складовою є трансформація даних – нормалізація, стандартизація, агрегування або створення нових ознак. Це дозволяє привести змінні до узгодженого формату та зменшити вплив масштабів вимірювання. Окрему увагу приділяють виявленню викидів (outliers), які можуть виникати через помилки вимірювання або бути результатом реальних, але рідкісних подій. Для їх аналізу застосовують статистичні методи, такі як Z-Score або міжквартильний розмах (IQR), а також візуальні інструменти.

Практична реалізація підготовки даних у лабораторній роботі здійснюється засобами бібліотеки pandas у середовищі Python. Використовуються операції завантаження даних у DataFrame, аналізу структури (info, describe), видалення колонок, перевірки пропусків, пошуку дублікатів та корекції назв змінних.

Комплексне виконання цих кроків забезпечує підвищення якості даних, зменшення ризику помилкових висновків та підвищення ефективності моделей.

ЗАВДАННЯ

1. Завантажити набір даних у DataFrame за допомогою pandas.
2. Видалити нерелевантні колонки, залишивши ті, що обрані раніше.
3. Визначити відсоток пропущених значень у кожному стовпці.
4. Перевірити дані на дублікати та оцінити їх доцільність.
5. Виправити помилки у назвах колонок.
6. Проаналізувати категоріальні значення та усунути некоректні дублікати.
7. Виявити та обробити викиди (outliers).
8. Опрацювати пропущені або некоректні значення.

ХІД ВИКОНАННЯ

1. Завантажити набір даних у DataFrame за допомогою pandas.

```
import pandas as pd
import numpy as np

from google.colab import drive
drive.mount('/content/drive')

df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/players_fifa23.csv')
df.head()
```

	short_name	player_positions	overall	potential	value_eur	wage_eur	age	height_cm	weight_kg	preferred_foot	...	mentality_composure	defending_marking_awareness	defending_standing_tackle
0	L. Messi	RW	91	91	54000000.0	195000.0	35	169	67	Left	...	96.0	20	35
1	K. Benzema	CF, ST	91	91	64000000.0	450000.0	34	185	81	Right	...	90.0	43	24
2	R. Lewandowski	ST	91	91	84000000.0	420000.0	33	185	81	Right	...	88.0	35	42
3	K. De Bruyne	CM, CAM	91	91	107500000.0	350000.0	31	181	70	Right	...	89.0	68	65
4	K. Mbappé	ST, LW	91	95	190500000.0	230000.0	23	182	73	Right	...	88.0	26	34
5	Cristiano Ronaldo	ST	90	90	41000000.0	220000.0	37	187	83	Right	...	95.0	24	32
6	M. Neuer	GK	90	90	13500000.0	72000.0	36	193	93	Right	...	70.0	17	10
7	T. Courtois	GK	90	91	90000000.0	250000.0	30	199	96	Left	...	66.0	20	18
8	V. van Dijk	CB	90	90	98000000.0	230000.0	30	193	92	Right	...	90.0	92	92
9	M. Salah	RW	90	90	115500000.0	270000.0	30	175	71	Left	...	92.0	38	43

Рис. 1. Завантаження даних та відображення DataFrame

2. Видалити нерелевантні колонки, залишивши ті, що обрані раніше.

```
selected_columns = [
    'short_name',
    'value_eur',
    'player_positions',
    'age',
    'height_cm',
    'weight_kg',
    'overall',
```

```
    'pace',
    'shooting',
    'passing',
    'dribbling',
    'defending',
    'physic'
]

df = df[selected_columns].copy()
df = df[~df['player_positions'].str.contains('GK', na=False)].copy()
df.head()
```

3. Визначити відсоток пропущених значень у кожному стовпці.

```
(df.isnull().sum() / len(df)) * 100
```

short_name	0.000000
value_eur	0.431165
player_positions	0.000000
age	0.000000
height_cm	0.000000
weight_kg	0.000000
overall	0.000000
pace	0.000000
shooting	0.000000
passing	0.000000
dribbling	0.000000
defending	0.000000
physic	0.000000

Рис. 2. Відсоток пропущених значень

4. Перевірити дані на дублікати та оцінити їх доцільність.

```
df.duplicated().sum()
df = df.drop_duplicates()
df.duplicated().sum()
```

У наборі даних повні дублікати відсутні, тому додаткове очищення не виконувалось.

5. Виправити помилки у назвах колонок.

```
df.columns
Index(['short_name', 'value_eur', 'player_positions', 'age', 'height_cm',
       'weight_kg', 'overall', 'pace', 'shooting', 'passing', 'dribbling',
       'defending', 'physic'],
      dtype='object')
```

Назви колонок є коректними та не потребують перейменування.

6. Проаналізувати категоріальні значення та усунути некоректні дублікати.

```
df['player_positions'] = (
    df['player_positions']
    .str.split(', ')
    .apply(lambda x: ', '.join(sorted(x)))
)

df['player_positions'].nunique()
```

7. Виявити та обробити викиди (outliers).

```
# Зберігаємо копію для незалежності
df_iqr = df.copy()

rows_before, cols_before = df_iqr.shape

Q1 = df_iqr['value_eur'].quantile(0.25)
Q3 = df_iqr['value_eur'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df_iqr = df_iqr[
    (df_iqr['value_eur'] >= lower_bound) &
    (df_iqr['value_eur'] <= upper_bound)
]

rows_after, cols_after = df_iqr.shape

print("До очищення:", rows_before, "рядків,", cols_before, "стовпців")
print("Після очищення:", rows_after, "рядків,", cols_after, "стовпців")
print("Видалено рядків:", rows_before - rows_after)

До очищення: 16467 рядків, 13 стовпців
Після очищення: 14473 рядків, 13 стовпців
Видалено рядків: 1994
```

За допомогою методу IQR було визначено межі допустимих значень для змінної value_eur. У результаті видалено 1994 спостереження, які виходили за межі інтервалу. Це дозволило зменшити вплив екстремальних значень на подальший аналіз.

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df_outlier = df.copy()

plt.figure(figsize=(10,5))
sns.boxplot(x=df_outlier['value_eur'])
plt.title('Пошук викидів у value_eur')
plt.show()

top_2 = df_outlier.nlargest(2, 'value_eur')

print("Аномалії до видалення:\n",
      top_2[['short_name', 'overall', 'value_eur']])

df_outlier = df_outlier.drop(top_2.index)

new_top = df_outlier.loc[df_outlier['value_eur'].idxmax()]

print(f"Новий найдорожчий: {new_top['short_name']} | "
      f"Overall: {new_top['overall']} | "
      f"Value: {new_top['value_eur']}")
```

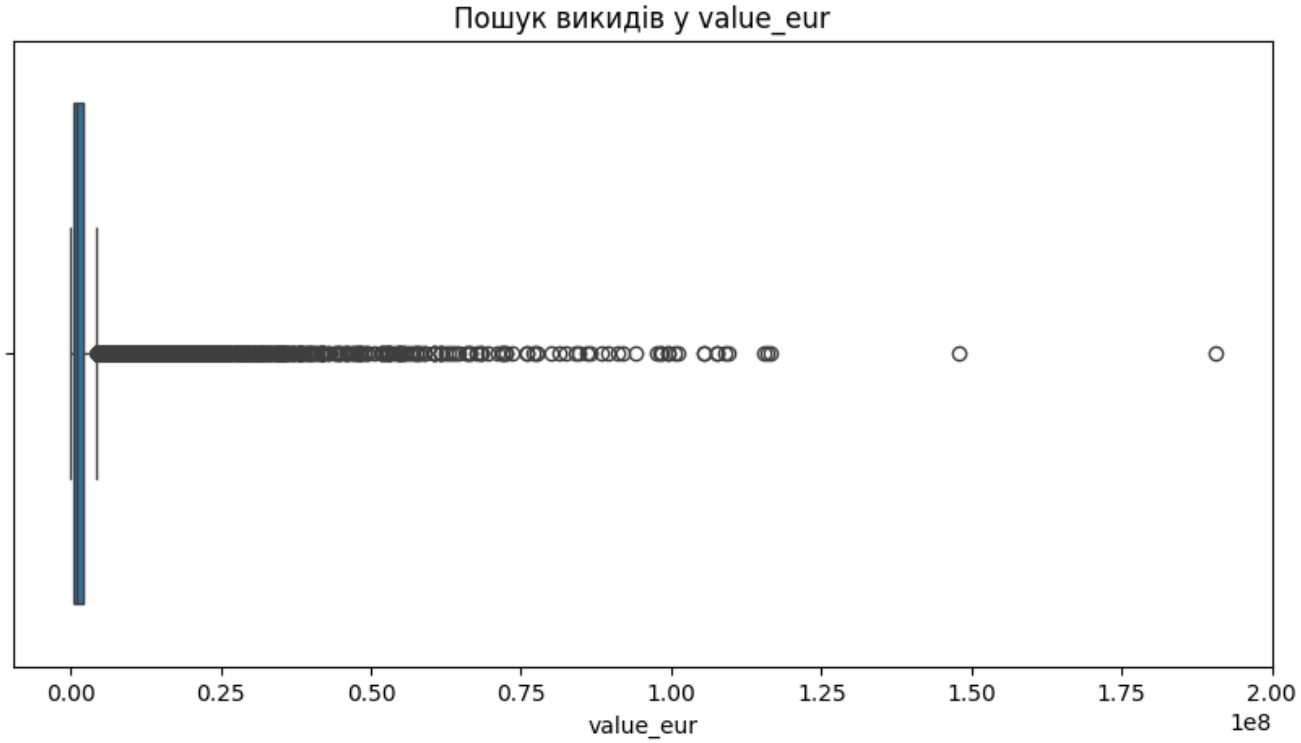

Рис. 3. Виявлення викидів (value_eur)

```
Аномалії до видалення:
     short_name  overall    value_eur
4    K. Mbappé       91  190500000.0
28  E. Haaland       88  148000000.0
Новий найдорожчий: F. de Jong | Overall: 87 | Value: 116500000.0
```

Найбільші значення вартості значно перевищують основну масу спостережень, тому вони були розглянуті як аномалії та видалені для стабілізації розподілу.

8. Опрацювати пропущені або некоректні значення.

```
df.isnull().sum()
df = df.dropna()
df.isnull().sum()
```

	0
short_name	0
value_eur	0
player_positions	0
age	0
height_cm	0
weight_kg	0
overall	0
pace	0
shooting	0
passing	0
dribbling	0
defending	0
physic	0

Рис. 4. Перевірка даних після очищення

Після видалення пропущених значень у наборі даних відсутні null-значення. Дані є повністю очищеними та придатними для подальшого аналізу.

ВИСНОВКИ

У ході лабораторної роботи було виконано очищення та підготовку набору даних відповідно до етапу Data Preparation методології CRISP-DM. Проведено аналіз пропущених значень, перевірку на дублікати, нормалізацію категоріальних ознак та обробку викидів. Отриманий набір даних є структурованим, узгодженим та придатним для подальшого моделювання.