Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

# Лабораторна робота №3
## **Робота з пропущеними даними**

Виконала

студентка групи ІТ-91: Перевірив:

Луцай Катерина Новотарський М. А.

Київ 2023

**Мета:** навчитися працювати з наборами даних, які містять відсутні або помилкові дані.

Варіант: 15 – dataset “Credit Risk Datasett” (https://www.kaggle.com/laotse/credit-risk-dataset)

**Хід виконання роботи:**

```
[1] import pandas as pd
    import numpy as np
```

```
[2] # connecting to gdrive
    from google.colab import drive
    drive.mount('/content/gdrive', force_remount=True)
    gdrive_path = f"/content/gdrive/MyDrive/ds/"
```

```
    Mounted at /content/gdrive
```

```
[16] # reading dataset to pandas dataframe
     data = pd.read_csv("/content/gdrive/MyDrive/ds/credit_risk_dataset.csv")
     # show first 15 rows of the dataframe
     data.head(15)
```

| | person_age | person_income | person_home_ownership | person_emp_length | loan_intent | loan_grade | l |
|---|---|---|---|---|---|---|---|
| 0 | 22 | 59000 | RENT | 123.0 | PERSONAL | D | |
| 1 | 21 | 9600 | OWN | 5.0 | EDUCATION | B | |
| 2 | 25 | 9600 | MORTGAGE | 1.0 | MEDICAL | C | |
| 3 | 23 | 65500 | RENT | 4.0 | MEDICAL | C | |
| 4 | 24 | 54400 | RENT | 8.0 | MEDICAL | C | |
| 5 | 21 | 9900 | OWN | 2.0 | VENTURE | A | |
| 6 | 26 | 77100 | RENT | 8.0 | EDUCATION | B | |
| 7 | 24 | 78956 | RENT | 5.0 | MEDICAL | B | |
| 8 | 24 | 83000 | RENT | 8.0 | PERSONAL | A | |
| 9 | 21 | 10000 | OWN | 6.0 | VENTURE | D | |
| 10 | 22 | 85000 | RENT | 6.0 | VENTURE | B | |
| 11 | 21 | 10000 | OWN | 2.0 | HOMEIMPROVEMENT | A | |
| 12 | 23 | 95000 | RENT | 2.0 | VENTURE | A | |
| 13 | 26 | 108160 | RENT | 4.0 | EDUCATION | E | |
| 14 | 23 | 115000 | RENT | 2.0 | EDUCATION | A | |

```
[6]  # counting total number of cells in the dataframe
     cells = np.product(data.shape)
     # percentage of missing values in total
     print(f"Missing {nulls.sum()/cells * 100}% of data")
```

```
Missing 1.025904668364998% of data
```

```
[7]  # selecting subset of the dataframe reduced in rows containing null values
     dropped_data = data.dropna()
     # show general info of the selected dataframe
     dropped_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 28638 entries, 0 to 32580
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   person_age                  28638 non-null  int64
 1   person_income               28638 non-null  int64
 2   person_home_ownership       28638 non-null  object
 3   person_emp_length           28638 non-null  float64
 4   loan_intent                 28638 non-null  object
 5   loan_grade                  28638 non-null  object
 6   loan_amnt                   28638 non-null  int64
 7   loan_int_rate               28638 non-null  float64
 8   loan_status                 28638 non-null  int64
 9   loan_percent_income         28638 non-null  float64
 10  cb_person_default_on_file   28638 non-null  object
 11  cb_person_cred_hist_length  28638 non-null  int64
dtypes: float64(3), int64(5), object(4)
memory usage: 2.8+ MB
```

```
[8]  print(f"Rows total: {data.shape[0]}")
     print(f"Rows dropped: {data.shape[0] - dropped_data.shape[0]}")
     print(f"Rows left: {dropped_data.shape[0]}")
```

```
Rows total: 32581
Rows dropped: 3943
Rows left: 28638
```

```
[9]  # selecting subset of the dataframe reduced in columns containing null values
     dropped_data = data.dropna(axis=1)
     # show general info of the selected dataframe
     dropped_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32581 entries, 0 to 32580
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   person_age                  32581 non-null  int64
 1   person_income               32581 non-null  int64
 2   person_home_ownership       32581 non-null  object
 3   loan_intent                 32581 non-null  object
 4   loan_grade                  32581 non-null  object
 5   loan_amnt                   32581 non-null  int64
 6   loan_status                 32581 non-null  int64
 7   loan_percent_income         32581 non-null  float64
 8   cb_person_default_on_file   32581 non-null  object
 9   cb_person_cred_hist_length  32581 non-null  int64
dtypes: float64(1), int64(5), object(4)
memory usage: 2.5+ MB
```

```
[10] print(f"Columns total: {data.shape[1]}")
     print(f"Columns dropped: {data.shape[1] - dropped_data.shape[1]}")
     print(f"Columns left: {dropped_data.shape[1]}")
```

```
Columns total: 12
Columns dropped: 2
Columns left: 10
```

```
[11] # show general info of the dataframe where gaps are filled with zeros
     data.fillna(0).info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32581 entries, 0 to 32580
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   person_age                  32581 non-null  int64
 1   person_income               32581 non-null  int64
 2   person_home_ownership       32581 non-null  object
 3   person_emp_length           32581 non-null  float64
 4   loan_intent                 32581 non-null  object
 5   loan_grade                  32581 non-null  object
 6   loan_amnt                   32581 non-null  int64
 7   loan_int_rate               32581 non-null  float64
 8   loan_status                 32581 non-null  int64
 9   loan_percent_income         32581 non-null  float64
 10  cb_person_default_on_file   32581 non-null  object
 11  cb_person_cred_hist_length  32581 non-null  int64
dtypes: float64(3), int64(5), object(4)
memory usage: 3.0+ MB
```

```
[12] # selecting subset of dataframe rows containing null values in any column
     subset_data = data[data.isnull().any(axis=1)]
     # show general info of the selected dataframe subset
     subset_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3943 entries, 39 to 32570
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   person_age                  3943 non-null   int64
 1   person_income               3943 non-null   int64
 2   person_home_ownership       3943 non-null   object
 3   person_emp_length           3048 non-null   float64
 4   loan_intent                 3943 non-null   object
 5   loan_grade                  3943 non-null   object
 6   loan_amnt                   3943 non-null   int64
 7   loan_int_rate               827 non-null    float64
 8   loan_status                 3943 non-null   int64
 9   loan_percent_income         3943 non-null   float64
 10  cb_person_default_on_file   3943 non-null   object
 11  cb_person_cred_hist_length  3943 non-null   int64
dtypes: float64(3), int64(5), object(4)
memory usage: 400.5+ KB
```

```
[13] # show subset where gaps are filled with numerical value 100
     # in column range from 4th to 9th contained null values
     subset_data.fillna(100).iloc[:, 3:8]
```

| | person_emp_length | loan_intent | loan_grade | loan_amnt | loan_int_rate |
|---|---|---|---|---|---|
| **39** | 3.0 | DEBTCONSOLIDATION | D | 30000 | 100.0 |
| **50** | 4.0 | DEBTCONSOLIDATION | D | 30000 | 100.0 |
| **57** | 3.0 | PERSONAL | A | 35000 | 100.0 |
| **59** | 2.0 | VENTURE | E | 1750 | 100.0 |
| **62** | 0.0 | EDUCATION | B | 10000 | 100.0 |
| **...** | ... | ... | ... | ... | ... |
| **32547** | 0.0 | VENTURE | C | 1400 | 100.0 |
| **32552** | 2.0 | EDUCATION | C | 10000 | 100.0 |
| **32553** | 2.0 | MEDICAL | C | 5000 | 100.0 |
| **32569** | 1.0 | PERSONAL | A | 7500 | 100.0 |
| **32570** | 5.0 | HOMEIMPROVEMENT | B | 4500 | 100.0 |

3943 rows × 5 columns

```
[14] # using next row valid observations to fill gaps in the original dataframe
     data = data.fillna(method='bfill', axis=0).fillna(0)
     # selecting subset in column range from 4th to 9th contained null values
     # to show rows with filled gaps and rows providing valid values to fill
     data.iloc[list(subset_data.index) + list(subset_data.index + 1), 3:8].sort_index()
```

| | person_emp_length | loan_intent | loan_grade | loan_amnt | loan_int_rate |
|---|---|---|---|---|---|
| **39** | 3.0 | DEBTCONSOLIDATION | D | 30000 | 17.99 |
| **40** | 6.0 | MEDICAL | E | 30000 | 17.99 |
| **50** | 4.0 | DEBTCONSOLIDATION | D | 30000 | 18.62 |
| **51** | 7.0 | DEBTCONSOLIDATION | F | 30000 | 18.62 |
| **57** | 3.0 | PERSONAL | A | 35000 | 7.29 |
| **...** | ... | ... | ... | ... | ... |
| **32554** | 1.0 | HOMEIMPROVEMENT | D | 15000 | 16.29 |
| **32569** | 1.0 | PERSONAL | A | 7500 | 10.00 |
| **32570** | 5.0 | HOMEIMPROVEMENT | B | 4500 | 10.00 |
| **32570** | 5.0 | HOMEIMPROVEMENT | B | 4500 | 10.00 |
| **32571** | 1.0 | VENTURE | B | 20000 | 10.00 |

7886 rows × 5 columns

Вихідний код у jupyter notebook:

https://colab.research.google.com/drive/1GNN3rmYNiS7iZw78HvWywvQMZgOtQ-57?usp=sharing

**Висновки:** було розглянуто основні методи мови Python для опрацювання наборів даних, які містять відсутні або помилкові дані, з використанням структур даних та інструментів бібліотеки Pandas.