[Raport zadanie1]

[Jacek Gross]

[Numer albumu 155285]

[Wykład specjalizujący]

[27.10.2023]

# [Czyszczenie danych]

# Z powodu błędnego wczytywania usuwam pierwszy wiersz, który jest pusty i mógłby źle wpłynąć na program  
data.drop(index=0, inplace=True)

# Przykładowe czyszczenie danych - usunięcie duplikatów  
data.drop\_duplicates()  
  
# Zmiana formatu kolumny "Fax" na string  
data['Fax'] = data['Fax'].apply(lambda x: str(int(x)) if x.is\_integer() else str(x))  
  
  
# Zmiana formatu kolumny "Telefon" na string  
data['Telefon'] = data['Telefon'].apply(lambda x: str(int(x)) if x.is\_integer() else str(x))

[Wybrany przeze mnie zestaw danych zawiera spis placówek edukacyjnych w województwie zachodniopomorskim. Według mnie należy przede wszystkim usunąć z tego zbioru danych placówki, które mogą się powtórzyć oraz zapewnić odpowiedni format danych np. dla kolumny ‘telefon’. Kolumna ta składa się z samych cyfr, ale nie jest liczbą tylko napisem.]

# [Weryfikacja poprawności danych]

# Sprawdzenie czy kolumna 'Telefon' ma długość 9 i składa się z cyfr  
invalid\_phone\_length = data[~(data['Telefon'].str.len() == 9)]  
invalid\_phone\_format = data[~data['Telefon'].str.isnumeric()]

[Do pokazania przykładu weryfikacji poprawności danych znowu posłużyłem się kolumną ‘telefon’. Do zmiennej przypisuje wartości z kolumny, które **nie mają** długości 9 i do kolejnej zmiennej przypisuję wartości, których ciąg znaków nie składa się z cyfr. Następnie wypisuję te wartości.]

if not invalid\_phone\_length.empty:  
 print("Napotkano nieprawidłowe długości numerów telefonów:")  
 print(invalid\_phone\_length)  
  
if not invalid\_phone\_format.empty:  
 print("Napotkano numery telefonów, które nie składają się z samych cyfr:")  
 print(invalid\_phone\_format)

# [Reformatowanie danych]

# Reformatowanie danych  
  
def reformat\_fax(data):  
 # Sprawdzenie warunków: wartość niepusta i długość 9  
 if len(data) == 9 and data.isdigit():  
 # Formatowanie wartości kolumny Fax co 3 cyfry  
 formatted\_fax = '-'.join(data[i:i+3] for i in range(0, len(data), 3))  
 return formatted\_fax  
 else:  
 return data  
  
  
# Zastosowanie funkcji reformatującej na kolumnie 'Fax'  
data['Fax'] = data['Fax'].apply(reformat\_fax)  
print(data['Fax'])

[Reformatowanie danych wykonuję przy użyciu funkcji, która sprawdza czy wartość kolumny nie jest pusta i czy ma długość 9. Następnie każdą wartość formatuję do postaci ‘xxx-xxx-xxx’ – co 3 znaki dodaj ‘-’. Następnie wykonuję te funkcję dla kolumny ‘Fax’]

# [Koniec programu]

# Zapisanie oczyszczonych danych do nowego pliku  
with pd.ExcelWriter("cleaned\_zachodniopomorskie.xlsx") as writer:  
 data.to\_excel(writer)

[Program kończy swoje działanie zapisując przerobione dane do pliku w celu sprawdzenia wykonania kodu. Niektóre z tych metod można również użyć dla innych kolumn np. metody dla kolumny ‘Fax’ i ‘Telefon’ mogą zostać użyte zamiennie. Inną kolumną, którą mógłbym uwzględnić jest ‘REGON’, który w zbiorze danych też nie powinien być liczbą tylko napisem. Podczas wykonywania ćwiczenia starałem się dobierać kolumny, dla których łatwo jest pokazać i wykonać metody **data wrangling’u.**]