|  | Politechnika Bydgoska im. J.J. Śniadeckich Wydział Telekomunikacji, Informatyki i Elektrotechniki | | |  |
| --- | --- | --- | --- | --- |
| **Przedmiot** | Algorytmy i eksploracja danych | | | |
| **Prowadzący** | dr inż. Michał Kruczkowski | | | |
| **Temat** | drzewa decyzyjne - eksploracja zbiorów danych | | | |
| **Studenci** | Adam Szreiber, Cezary Naskręt | | | |
| **Nr ćw.** | 1 | **Data wykonania** | 10.10.2023 | |

# 

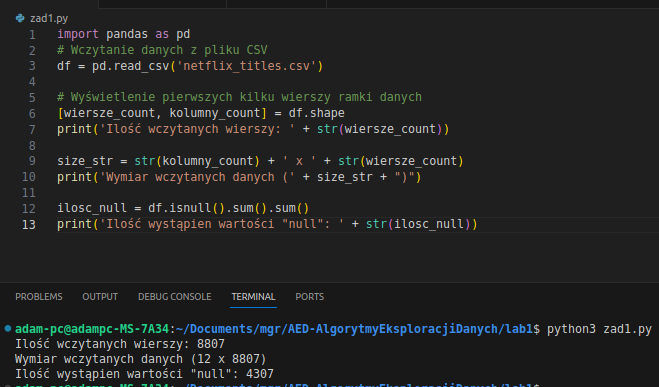
# Zadanie 1

Należało wczytać dane do obiektu “DataFrame”, co uczyniono metodą pd.read\_csv()

Następnie odwołano się do pola shape które reprezentuje wielowymiarowość.

Zliczanie ilości wystąpień wartości null, odbywa się za pomocą metody isnull()

w ten sposób otrzymano DataFrame zawierający wartości true/false w zależności od wartości w tym polu w pierwotnej tablicy.

Następnie sum() zlicza ilość wystąpień true , i zwraca wektor ilością wystąpień null w danym wierszu.  
Kolejne wywołanie sum() zlicza wszystkie wartości w wektorze i zwraca ilość wszystkich wystąpień wartości `null` w DataFrame.

Ryc 1. Kod realizujący założenia.

# 

# Zadanie 2

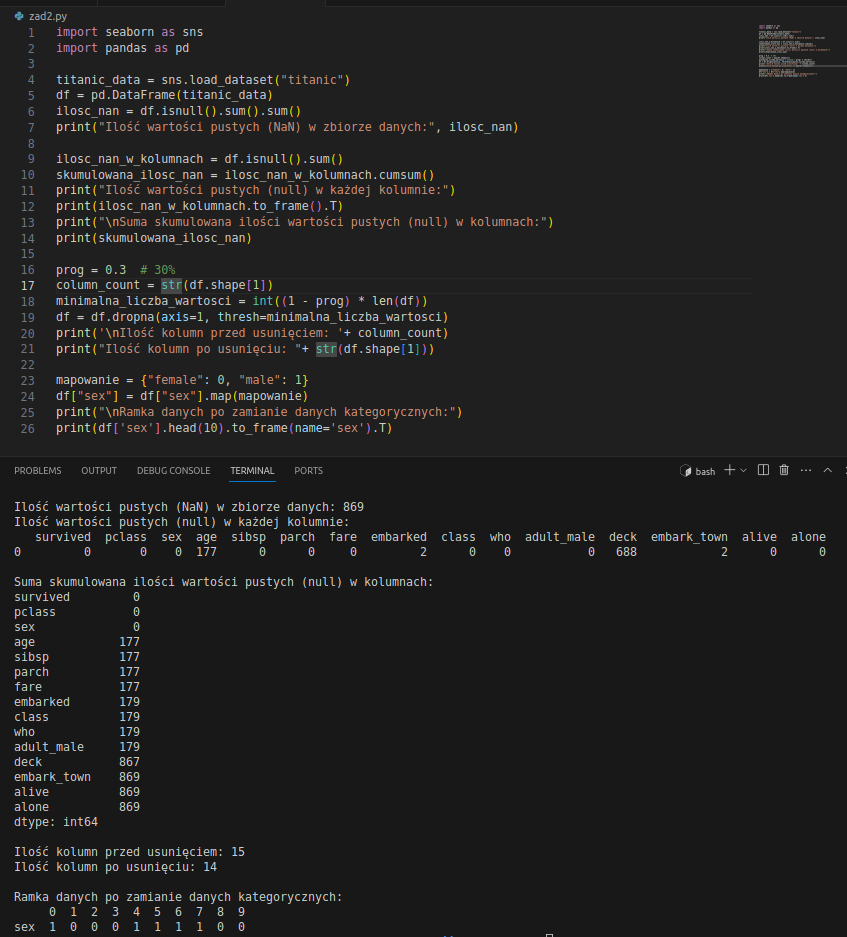
Należało wczytać zbiór danych “titanic”.

Następnie po raz kolejny wykorzystano składnie isnull() oraz podwójne sum() aby uzyskać ilość pustych wartości w zbiorze.

Otrzymanie skumulowanej sumy ilości wartości null w poszczególnych kolumnach wykonano poprzez uruchomienie metody .cumsum() na wektorze ilości null w poszczególnych kolumnach.

W kolejnym kroku należało usunąć kolumny które posiadają więcej niż 30% danych pustych, posłużyła w tym metoda dropna() z parametrem thresh=minimalna ilość wartości non-Null

W ostatnim kroku zaimplementowano mapowanie pola sex z wartościami “female” | “male” na wartości odpowiadające 0 | 1.



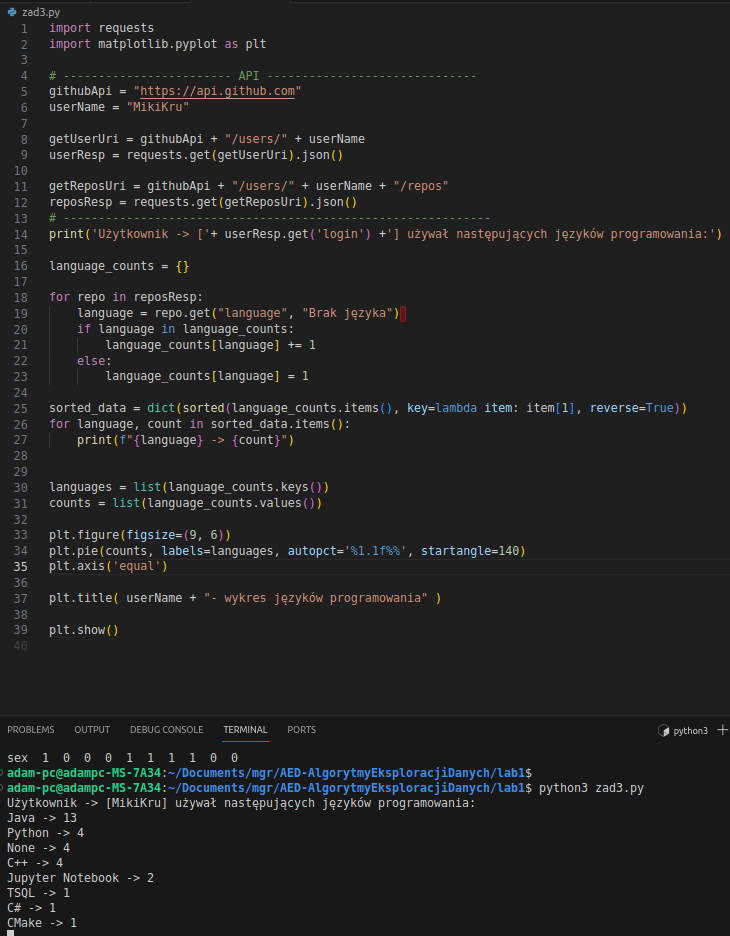
Ryc 2. Ukazanie wyników i kodu realizującego zadanie 2

# Zadanie 3

Wymagało pobrania użytkownika oraz repozytoriów użytkownika z github, użyto do tego biblioteki requests.

Następnie odbywało się liczenie ilości wystąpień konkretnego języka programowania.

Ostatnim krokiem było utworzenie wykresu kołowego z otrzymanymi danymi - w tym celu użyto bibliotekę pyplot



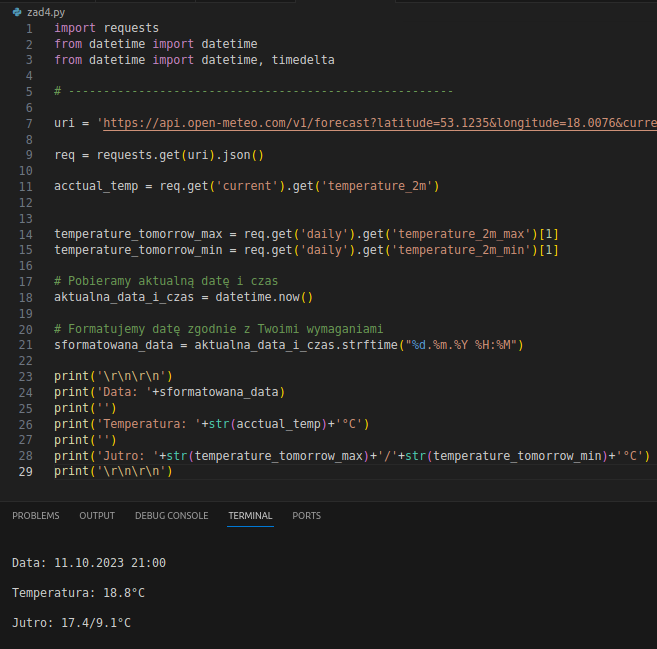
Ryc 3. Prezentacja kodu i wyników

# Zadanie 4

Należało pobrać dane odnośnie pogody - aktualna temperatura oraz najwyższa i najniższa temperatura na kolejny dzień.   
W query umieszczono lokalizację która nas interesuje ( Bydgoszcz, latitude=53.1235 longitude=18.0076 ), następnie parametr current - aktualna temperatura, daily z parametrami temperature\_2m\_min oraz temperature\_m2\_max - zwraca najniższą i najwyższą dzienną temperature, forecast\_days - specyfikuje ilość dni.

Precyzyjnie skonstruowane zapytanie pozwoliło otrzymać wszystkie wymagane informacje, w kolejnych krokach pozostało tylko wyciąganie i prezentowanie danych.

<https://api.open-meteo.com/v1/forecast?latitude=53.1235&longitude=18.0076&current=temperature_2m&daily=temperature_2m_max,temperature_2m_min&timezone=Europe%2FBerlin&forecast_days=2>



Ryc 4. Prezentacja kodu i wyników w terminalu.

# Wnioski

Podczas tego laboratorium, eksplorowaliśmy publiczne API GitHub, co pozwoliło na zdobycie danych związanych z repozytoriami użytkownika. Przeprowadziliśmy analizę, w której zliczaliśmy ilość repozytoriów napisanych w konkretnym języku programowania. Do komunikacji z API wykorzystaliśmy bibliotekę Requests, która umożliwia pobieranie danych z zewnętrznych źródeł. Następnie wykorzystaliśmy bibliotekę Pandas do przetwarzania i analizy tych danych, co pozwoliło na wyodrębnienie istotnych informacji oraz przygotowanie ich do dalszej analizy.

Ponadto, w trakcie tego laboratorium nauczyliśmy się tworzyć wykresy kołowe przy pomocy biblioteki Matplotlib pyplot. Dzięki temu stworzyliśmy wykres, który wizualizował udział procentowy repozytoriów w różnych językach programowania, co pozwoliło na lepsze zrozumienie preferencji użytkownika w kontekście programowania.

Biblioteki Pythona, takie jak Pandas, Matplotlib i Requests, stanowią potężne narzędzia do przeprowadzania analizy danych i komunikacji z zewnętrznymi źródłami danych.

Eksploracja danych jest kluczowym etapem w analizie danych, pozwalającym zrozumieć i przygotować dane przed przystąpieniem do modelowania. Pozwala na zrozumienie charakterystyki zbioru danych i identyfikowanie problemów z brakującymi danymi lub innymi anomaliami.

Repozytorium dostępne pod adresem <https://github.com/AdamSzr/AED-AlgorytmyEksploracjiDanych/tree/main/lab1>