**Analiza danych doświadczalnych**

*znaleźć dane z jakiegoś eksperymentu w internecie (fizyka, biologia, ...) i*

*wykonać analizę danych z użyciem bibliotek do Pythona, wyciągnąć wnioski*

*Tran Ngoc Khanh Nam*

**Harmonogram:**

|  |  |
| --- | --- |
| 21.03 (lab 2) | Stworzyć harmonogram |
| 04.04 (lab 3) | Używać funkcji do przetwarzania danych **(pandas, numpy)**:  - Read file (csv, json, excel,...), Dataframe  - Filtrować, usunąć, dodać, zmieniać, zastępować,... dane |
| 18.04 (lab 4) | Data cleaning, outliers, duplicates |
| 16.05 (lab 5) | Rysować wykresy: pie, column,... **(matplotlib, seaborn)**  Stworzyć program (interfejs + widżet) **(ipywidgets)** (1/2) |
| 30.05 (lab 6) | Stworzyć program (interfejs + widżet) **(ipywidgets)** (2/2)  Zastosować do różnorodnych danych, prezentacja |
| (lab 7) | Oddanie projektu, podsumowanie |

**Raport:**

W tym projekcie przeprowadziliśmy analizę danych przy użyciu Pythona i środowiska Jupyter Notebook. Głównym celem było analizowanie i uzyskiwanie wglądu w zbiór danych przy użyciu różnych technik manipulacji danych i wizualizacji. Przejdźmy teraz przez różne aspekty projektu.

**1. Wykorzystane biblioteki:**

Wykorzystaliśmy następujące biblioteki do naszej analizy danych:

* pandas: Używane do manipulacji i analizy danych.
* ipywidgets: Używane do tworzenia interaktywnych widżetów dla użytkownika.
* matplotlib: Używane do wizualizacji danych.
* IPython.display: Używane do wyświetlania wyników w Jupyter Notebook.

**2. Wprowadzenie danych:**

Wczytaliśmy dane z plików CSV przy użyciu biblioteki pandas. Użytkownik mógł określić wiele plików, które zostały połączone w jedną ramkę danych. Ścieżka do plików wejściowych oraz nazwy plików zostały wprowadzone przez użytkownika przy użyciu ipywidgets.

**3. Czyszczenie i przygotowanie danych:**

Przeprowadziliśmy kilka kroków czyszczenia i przygotowania danych na wczytanych danych, w tym:

* Usuwanie wierszy zawierających brakujące wartości.
* Usuwanie niepotrzebnych kolumn.
* Zmiana typów danych kolumn na podstawie wyboru użytkownika.

Użytkownik mógł wybrać typy danych dla każdej kolumny za pomocą rozwijanych menu. Zmiany typów danych odzwierciedlały się w ramce danych, a użytkownik otrzymywał informacje o poprzednich i zaktualizowanych typach danych dla każdej kolumny.

**4. Ogólne informacje o zbiorze danych:**

Wyświetliliśmy ogólne informacje o zbiorze danych, takie jak liczba wierszy, unikalne wartości, pierwsze i ostatnie wartości, typy danych, średnia, odchylenie standardowe i zużycie pamięci dla każdej kolumny. Informacje te zostały przedstawione w formacie tabelarycznym za pomocą ipywidgets.

**5. Modyfikacja danych:**

Zapewniliśmy opcje modyfikowania ramki danych, w tym dodawanie, modyfikowanie i usuwanie kolumn. Jednak konkretne kody dotyczące modyfikacji zostały zakomentowane w notatniku Jupyter.

6. Wizualizacja danych:

Zaimplementowaliśmy różne rodzaje wizualizacji danych na podstawie wyboru użytkownika przy użyciu biblioteki matplotlib. Dostępne typy wykresów obejmowały:

* Wykres słupkowy
* Wykres liniowy
* Histogram
* Wykres kołowy
* Wykres punktowy
* Wykres liniowy dla wielu serii danych
* Wykres słupkowy-liniowy dla wielu serii danych

Użytkownik mógł wybrać typ wykresu oraz odpowiadające kolumny dla osi x i y z zbioru danych. Mogli również dostosować etykiety i tytuł wykresu. Wybrany wykres był wyświetlany przy użyciu matplotlib.

**7. Podsumowanie:**

W tym projekcie pomyślnie przeprowadziliśmy analizę podanego zbioru danych przy użyciu Pythona i Jupyter Notebook. Przeprowadziliśmy czyszczenie, przygotowanie, manipulację i wizualizację danych, aby uzyskać wgląd w dane. Interaktywność ipywidgets umożliwiła użytkownikom dostosowanie analizy do ich potrzeb. Projekt ten pokazał potęgę Pythona i jego bibliotek do zadań związanych z analizą danych.

Należy zauważyć, że konkretne kody dotyczące modyfikacji danych (dodawanie, zmiana, usuwanie) zostały zakomentowane w dostarczonym notatniku Jupyter. Możesz odkomentować i zmodyfikować kod według potrzeb.

Podsumowując, ten projekt zapewnił praktyczne doświadczenie w analizie danych za pomocą Pythona i zaprezentował wykorzystanie różnych bibliotek i technik do eksploracji i wizualizacji danych.