



**WYDZIAŁ  
ELEKTROTECHNIKI  
I INFORMATYKI**  
POLITECHNIKI RZESZOWSKIEJ

**Krzysztof Lang**

Implementacja wybranych algorytmów wypełniania  
brakujących wartości, dla strumieni dużych zbiorów danych

**Praca dyplomowa magisterska**

Opiekun pracy:  
dr Michał Piętał

Rzeszów, 2023



# Spis treści

|  |           |
|--|-----------|
| <b>1. Wstęp</b>  | <b>5</b>  |
| <b>2. Wprowadzenie do wypełniania brakujących wartości</b> | <b>6</b>  |
| 2.1. Rys historyczny                                       | 6         |
| 2.2. Na czym polega wypełnianie brakujących wartości       | 6         |
| 2.3. Korzyści i zagrożenia                                 | 6         |
| 2.4. Perspektywy na przyszłość                             | 6         |
| <b>3. Omówienie narzędzi i danych</b>                      | <b>7</b>  |
| 3.1. Python  | 7         |
| 3.2. Visual Studio Code                                    | 7         |
| 3.3. Biblioteki  | 7         |
| 3.3.1. Pandas  | 7         |
| 3.3.2. NumPy   | 7         |
| 3.3.3. Scikit  | 7         |
| 3.4. Źródła danych   | 7         |
| 3.4.1. Użyte repozytoria danych                            | 7         |
| 3.4.2. Adult Data Set                                      | 8         |
| 3.4.3. Stock Exchange Data                                 | 9         |
| <b>4. Implementacja i testy algorytmu</b>                  | <b>11</b> |
| 4.1. Opis przygotowanego programu                          | 11        |
| 4.1.1. Tworzenie brakujących wartości                      | 11        |
| 4.1.2. Wypełnianie brakujących wartości                    | 13        |
| 4.1.3. Sprawdzenie skuteczności wypełniania                | 13        |
| 4.2. Opis implementacji                                    | 13        |
| 4.2.1. Alg 1   | 13        |
| 4.2.2. Alg 2   | 13        |
| 4.2.3. Alg 3   | 13        |
| 4.3. Napotkane problemy                                    | 13        |
| 4.4. Testy algorytmów na wybranych źródłach danych         | 13        |
| <b>5. Podsumowanie i wnioski końcowe</b>                   | <b>15</b> |
| <b>Załączniki</b>  | <b>16</b> |
| <b>Literatura</b>  | <b>17</b> |



## 1. Wstep

## **2. Wprowadzenie do wypełniania brakujących wartości**

### **2.1. Rys historyczny**

### **2.2. Na czym polega wypełnianie brakujących wartości**

### **2.3. Korzyści i zagrożenia**

### **2.4. Perspektywy na przyszłość**

## **3. Omówienie narzędzi i danych**

### **3.1. Python**

Do przygotowania programu wykorzystanego do przeprowadzenia badań wybrano język Python. Jest to język wysokiego poziomu, charakteryzujący się prostą składnią i wysoką przejrzystością kodu. Programy nie muszą być kompilowane przed uruchomieniem, co znacznie przyspiesza proces prototypowania i debugowania. Oznacza to też że Python jest wolniejszy od wielu innych języków, jednak w przypadku niniejszej pracy nie ma to znaczenia. Dostępna ogromna ilość gotowych bibliotek służących do obróbki i analizy danych znacząco uprościła przygotowanie programu. Podczas pisania kodu trzymano się dobrych praktyk, stosowano wytyczne zawarte w PEP8.

### **3.2. Visual Studio Code**

Jako środowisko programowania wybrano "Microsoft Visual Studio Code". Jest to darmowy edytor kodu obsługujący wiele języków. Ze względu na otwartość kodu, dostępne jest wiele rozszerzeń do programu, które znacznie ułatwiają tworzenie nawet skomplikowanych projektów. W celu umożliwienia pracy nad programem z wielu urządzeń oraz dla zachowania pełnej historii tworzenia programu wykorzystano integrację "Visual Studio Code" z repozytorium GitHub.

### **3.3. Biblioteki**

#### **3.3.1. Pandas**

#### **3.3.2. NumPy**

#### **3.3.3. Scikit**

### **3.4. Źródła danych**

#### **3.4.1. Użyte repozytoria danych**

Aby wyniki badań niosły ze sobą odpowiednią wartość merytoryczną, potrzebne są odpowiednie zbiory danych na których zostaną przeprowadzone testy. W celu znalezienia odpowiednich zbiorów danych, przyjęto następujące założenia:

- zbiór danych musi być wystarczająco duży,
- zbiór danych musi zawierać odpowiednią ilość atrybutów aby modele decyzyjne miały do dyspozycji wystarczającą ilość danych uczących,

- atrybuty powinny zawierać różnorodne typy danych w celu przetestowania wypełniania zarówno danych liczbowych (całkowitych i zmiennoprzecinkowych) jak i kategorycznych,
- zbiór danych nie może mieć pustych wartości.

Do wyszukania odpowiednich zbiorów danych wykorzystano narzędzie "Google Dataset Search". Z jego pomocą wybrano 2 zbiory danych z różnych dziedzin. Po uprzedniej ich obróbce zostały wykorzystane do przeprowadzenia testów algorytmów wypełniania.

### 3.4.2. Adult Data Set

Zbiór danych "Adult Data Set" zawiera dane ze spisu ludności przeprowadzonego w roku 1994 w Stanach Zjednoczonych. Jest szeroko wykorzystywany do testowania uczenia maszynowego. Zawiera ponad 30000 rekordów i 15 atrybutów. [1] Opis atrybutów:

- age: wiek spisanej osoby, liczba całkowita,
- workclass: rodzaj zatrudnienia, dane kategoryczne, 8 możliwych wartości,
- fnlwgt: jaka proporcja populacji ma identyczny zestaw pozostałych wartości, liczba całkowita,
- education: osiągnięty poziom edukacji, dane kategoryczne, 16 możliwych wartości,
- education-num: osiągnięty poziom edukacji zakodowany jako liczba całkowita,
- marital-status: status matrymonialny, dane kategoryczne, 7 możliwych wartości,
- occupation: zawód, dane kategoryczne, 14 możliwych wartości,
- relationship: rola w związku, dane kategoryczne, 6 możliwych wartości,
- race: klasyfikacja rasowa, dane kategoryczne, 5 możliwych wartości,
- sex: płeć, dane kategoryczne, 2 możliwe wartości,
- capital-gain: zysk kapitału w związku z inwestycjami, liczba całkowita,
- capital-loss: strata kapitału w związku z inwestycjami, liczba całkowita,



- hours-per-week: ilość godzin pracujących w tygodniu, liczba całkowita,
- native-country: kraj pochodzenia, dane kategoryczne, 41 możliwych wartości,
- attribute: czy osoba zarabia powyżej czy poniżej 50000\$ rocznie.

Ten zbiór danych został wybrany ze względu na występowanie zarówno atrybutów liczbowych jak i kategorycznych, zadowalającą ilość rekordów oraz atrybutów. Ma na celu przetestowanie skuteczności działania algorytmów do wypełniania brakujących miejsc w zbiorach danych z brakami w danych o różnych typach. Nie wymaga dodatkowej obróbki przed rozpoczęciem testów.

### 3.4.3. Stock Exchange Data

Zbiór danych "Stock Exchange Data" zawiera informacje o cenach akcji na giełdach w różnych krajach w latach 1965-2021. Dane zostały zebrane z "Yahoo Finance", posiadającego dane o giełdzie z wielu lat w wielu krajach. Posiada ponad 100000 rekordów i 9 atrybutów. [2] Opis atrybutów:

- Index: symbol wskazujący z jakiej giełdy pochodzą dane, dane kategoryczne, 5 możliwych wartości,
- Date: data obserwacji, dane kategoryczne,
- Open: cena akcji podczas otwarcia, liczba wymierna,
- High: najwyższa cena w ciągu dnia, liczba wymierna,
- Low: najniższa cena w ciągu dnia, liczba wymierna,
- Close: cena akcji w momencie zamknięcia, liczba wymierna,
- Adj Close: cena akcji w momencie zamknięcia skorygowana o podziały jak i dywidendy, liczba wymierna,
- Volume: liczba akcji będących przedmiotem obrotu w ciągu dnia sesyjnego, liczba całkowita,
- CloseUSD: cena akcji w momencie zamknięcia wyrażona w dolarach amerykańskich

Ten zbiór danych został wybrany ze względu na bardzo popularną kategorię danych, to jest dane finansowe. Ma na celu przetestowanie skuteczności działania algorytmów w przypadku danych numerycznych, w szczególności liczb wymiernych. W celu lepszego przygotowania do testów zakodowano kolumnę "Data" z wykorzystaniem "label encoding", to jest zamiany danych na postać numeryczną. Usunięto też rekordy posiadające wartość "0" w kolumnie "Volume". Ich duża ilość (ponad 30%) mogła by negatywnie wpłynąć na uczenie modeli decyzyjnych. W wyniku tego zmniejszono liczbę rekordów do ponad 62000, co wciąż jest ilością spełniającą założenia dla zbiorów danych.

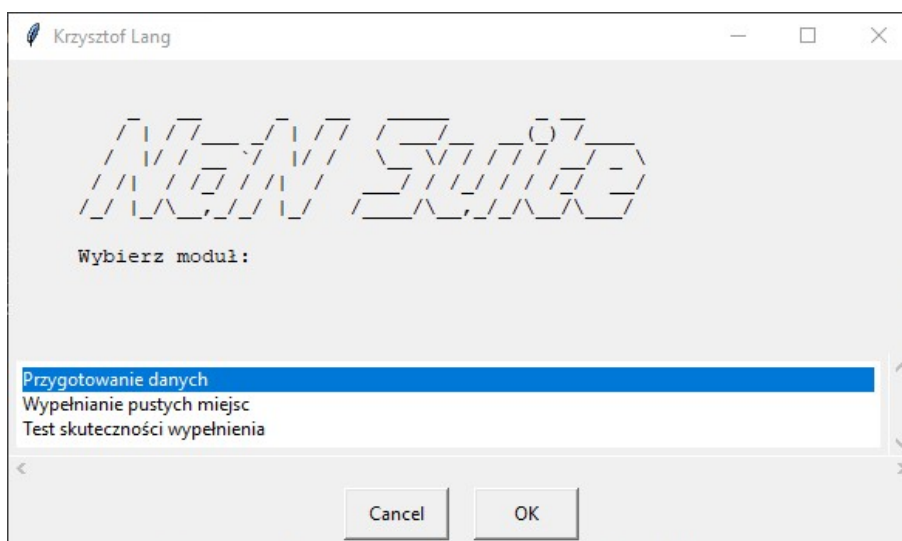
## 4. Implementacja i testy algorytmu

### 4.1. Opis przygotowanego programu

Program został napisany w języku Python, z implementacją prostego interfejsu graficznego. Został nazwany "NaN Suite". Program miał spełniać 3 role:

- 1) Przygotować dane do wypełniania poprzez sztuczne utworzenie brakujących wartości.
- 2) Wypełnić brakujące wartości z wykorzystaniem wybranych algorytmów.
- 3) Ocenić skuteczność wypełniania w celu porównania algorytmów.

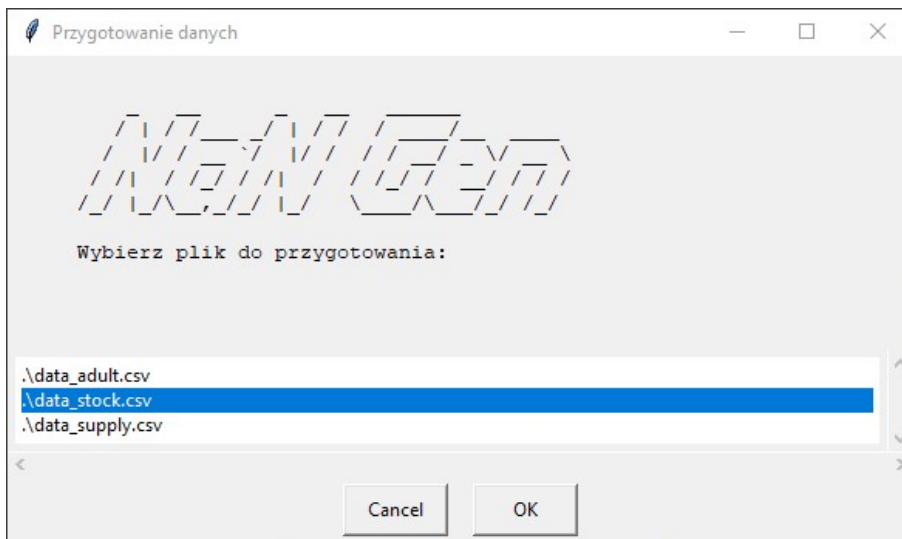
Poszczególne role zrealizowano jako osobne moduły. W kolejnych podrozdziałach zaprezentowane zostanie działanie programu na przykładowym pliku. Po uruchomieniu programu pokazuje się okno służące do wyboru modułu, pokazane na rysunku 4.1.



Rysunek 4.1: Główne okno programu, pozwalające na wybór modułu do uruchomienia

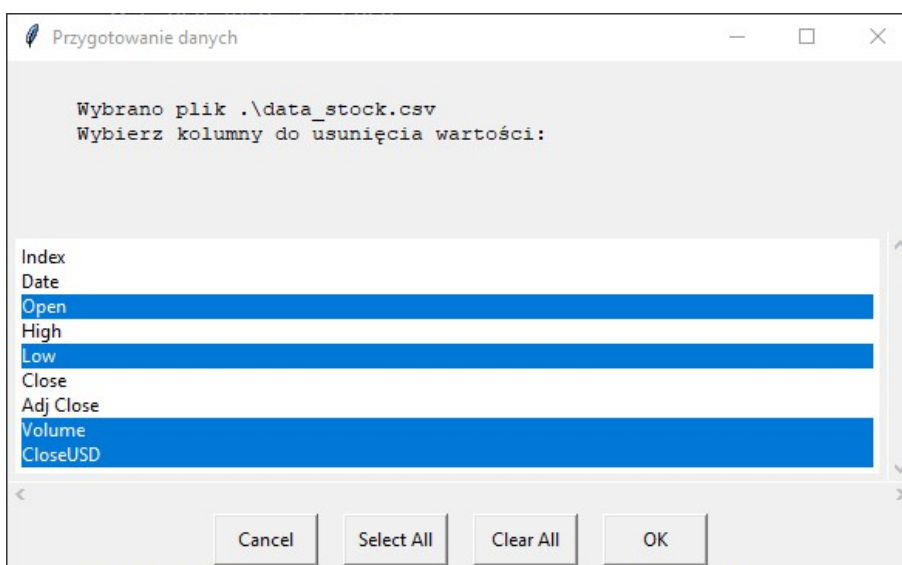
#### 4.1.1. Tworzenie brakujących wartości

Pierwszy moduł odpowiada z przygotowanie danych do wypełniania. Pierwszym krokiem jest wybranie pliku który ma zostać przygotowany z użyciem okna pokazanego na rysunku 4.2.



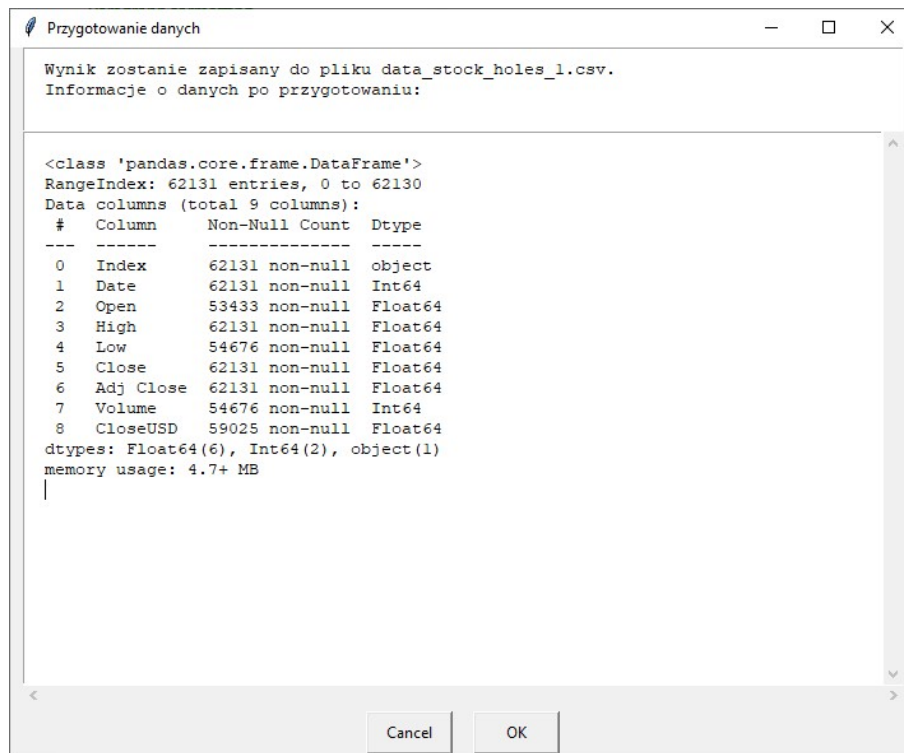
Rysunek 4.2: Okno wyboru pliku do przygotowania

Następnie wybrane zostają kolumny w których mają zostać usunięte dane. Wybrać można dowolną ilość, lecz zalecane jest poniżej 50%. Wyboru dokonuje się z użyciem okna pokazanego na rysunku 4.3.



Rysunek 4.3: Okno wyboru kolumn

W wybranych kolumnach zostaje usunięte od 5% do 15% wartości - dla każdej kolumny ilość jest losowana. Dodatkowo stworzony zostaje plik przechowujący informacje które wartości zostały usunięte. Po zakończeniu usuwania wartości wyświetlone zostaje okno z podsumowaniem jak na rysunku 4.4.



Rysunek 4.4: Okno z podsumowaniem

#### 4.1.2. Wypełnianie brakujących wartości

Drugi moduł służy do wypełniania brakujących wartości w zbiorze danych z użyciem wybranego algorytmu. Najpierw należy wskazać plik który ma zostać wypełniony, tak jak na rysunku 4.5

Następnie z użyciem okna jak na rysunku 4.6 wybrany zostaje algorytm który ma zostać wykorzystany do wypełniania brakujących wartości.

Puste miejsca zostają wypełnione, po czym wyświetlone zostaje okno z podsumowaniem, jak na rysunku 4.7.

#### 4.1.3. Sprawdzenie skuteczności wypełniania

### 4.2. Opis implementacji

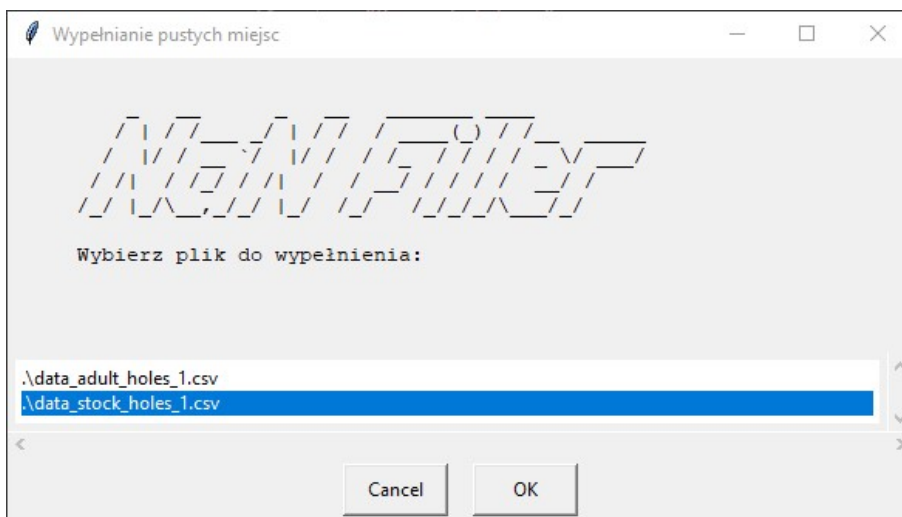
#### 4.2.1. Alg 1

#### 4.2.2. Alg 2

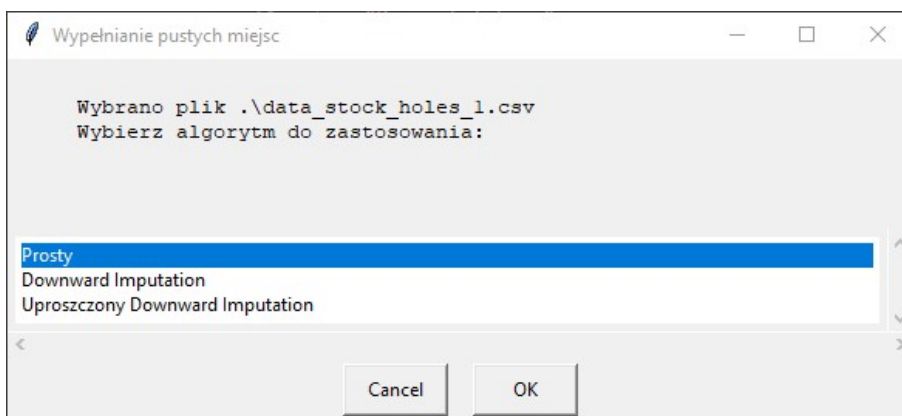
#### 4.2.3. Alg 3

### 4.3. Napotkane problemy

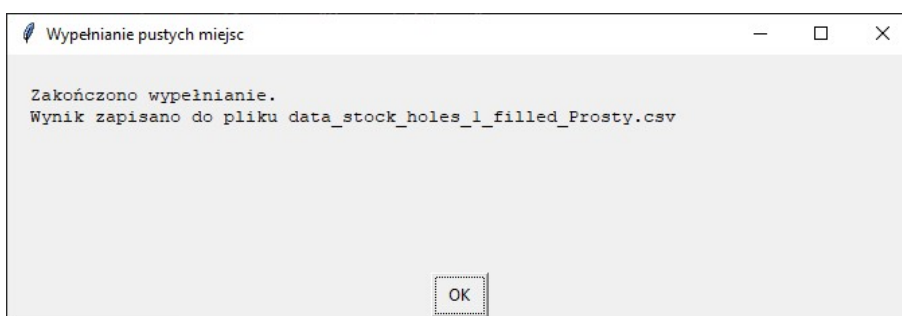
### 4.4. Testy algorytmów na wybranych źródłach danych



Rysunek 4.5: Okno wyboru pliku do wypełnienia



Rysunek 4.6: Okno wyboru algorytmu



Rysunek 4.7: Okno z podsumowaniem wypełniania

## 5. Podsumowanie i wnioski końcowe

## Załączniki



## Literatura

- [1] [archive.ics.uci.edu/ml/datasets/adult](https://archive.ics.uci.edu/ml/datasets/adult). Dostęp 26.02.2023.
- [2] [www.kaggle.com/datasets/mattiuzc/stock-exchange-data](https://www.kaggle.com/datasets/mattiuzc/stock-exchange-data). Dostęp 26.02.2023.

**STRESZCZENIE PRACY DYPLOMOWEJ MAGISTERSKIEJ**  
**IMPLEMENTACJA WYBRANYCH ALGORYTMÓW**  
**WYPEŁNIANIA BRAKUJĄCYCH WARTOŚCI, DLA STRUMIENI**  
**DUŻYCH ZBIORÓW DANYCH**

Autor: Krzysztof Lang, nr albumu: EF-148853

Opiekun: dr Michał Piętał

Słowa kluczowe: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po polsku

**MSC THESIS ABSTRACT**  
**IMPLEMENTATION OF SELECTED MISSING VALUE FILLING**  
**ALGORITHMS FOR LARGE DATA SETS**

Author: Krzysztof Lang, nr albumu: EF-148853

Supervisor: Michał Piętał, PhD

Key words: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po angielsku