|  |  |
| --- | --- |
| **Obraz zawierający tekst  Opis wygenerowany automatycznie** | |
| **Politechnika Opolska**  **Wydział Elektrotechniki, Automatyki i Informatyki**  **Katedra Informatyki** | |
| **Rok akademicki** | 2020/2021 |
| **Przedmiot** | Ekonometria |
| **Prowadzący zajęcia** | Dr inż. Arkadiusz Gardecki |
| **Nr grupy** | 1 |

|  |
| --- |
| **Projekt wybierający zmienne objaśniające i na ich podstawie ustalający model ekonometryczny** |
|  |

|  |  |
| --- | --- |
| **Nazwisko i imię** | **Nr indeksu** |
|
| **Karol Kliniewski** | **101272** |

Spis treści

[1. Wprowadzenie 3](#_Toc59979568)

[2. Opis interfejsu i pliku danych wejściowych 3](#_Toc59979569)

[3. Realizacja 5](#_Toc59979570)

[4. Kod źródłowy – uruchomienie 7](#_Toc59979571)

[5. Wnioski 7](#_Toc59979572)

# Wprowadzenie

Celem projektu jest stworzenie aplikacji z interfejsem graficznym. Aplikacja ma pozwalać na przyjmowanie danych wejściowych i na ich podstawie dokonania doboru zmiennych objaśniających. Zmienne te posłużą następnie do ustalenia równania modelu. Otrzymany model ostatecznie jest porównywany z pomiarami otrzymanymi w pliku razem z danymi wejściowymi.

Dokładny przebieg programu można opisać przy pomocy poniższych kroków:

1. Obliczyć i umożliwić wyświetlenie współczynnika korelacji R i Ro.

2. Dokonać wyboru zmiennych objaśniających metodą analizy Grafów

3. Obliczyć szacunki parametrów strukturalnych modelu metodą KMNK

4. Obliczyć błędy średnie szacunku parametrów D(a), odchylenie standardowe składnika resztowego Su, współczynnika determinacji R2, współczynnika zmienności losowej v.

# Opis interfejsu i pliku danych wejściowych

Do rozwiązania problemu opisanego we wprowadzeniu zdecydowano się na wykorzystanie języka programowania Python, ponieważ ma wiele dostępnych rozszerzeń ułatwiających operacje na macierzach(jak NumPy) oraz tworzenie wykresów(przy pomocy MatPlotLib).

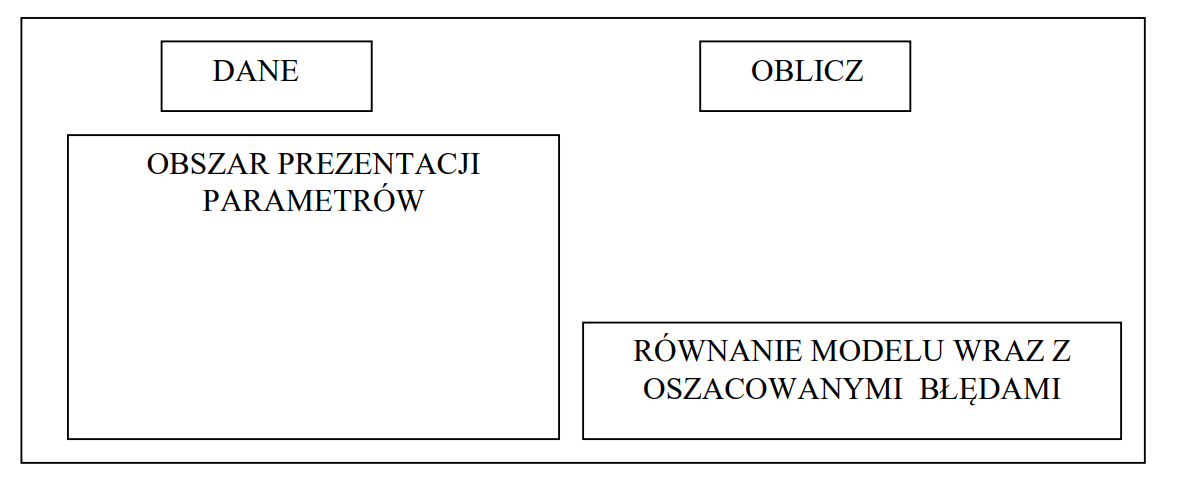
Zgodnie z założeniami aplikacja miała posiadać interfejs graficzny, który składa się z elementów przedstawionych na rysunku 1.:

- Dane – zawiera dane wejściowe, które pochodzą z pliku tekstowego,

- Oblicz – obszar zawierający obliczenia oraz opisujący przeprowadzone operacje na danych,

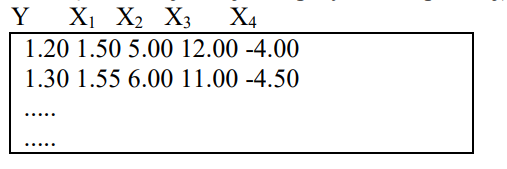
- Równanie modelu wraz z oszacowanymi błędami,

- Obszar prezentacji parametrów – zawiera wykres przedstawiający graficzne rozwiązanie obliczonego modelu i porównuje go do pomiarów.



Rysunek 1 Schemat graficznego interfejsu

Dane wejściowe w stworzonym programie muszą znajdować się w folderze aplikacji w pliku o nazwie „dane.txt” oraz wewnętrzna struktura pliku musi być zgodna z rysunkiem 2.



Rysunek 2 Struktura pliku z danymi

Dodatkowo plik powinien spełniać następujące założenia:  
- nie posiadać nagłówków,

- separatorem kolumn powinna być spacja,

- w przypadku miejsc dziesiętnych, powinny być one rozdzielone kropką,

- kolejność kolumn jak na rysunku 2.

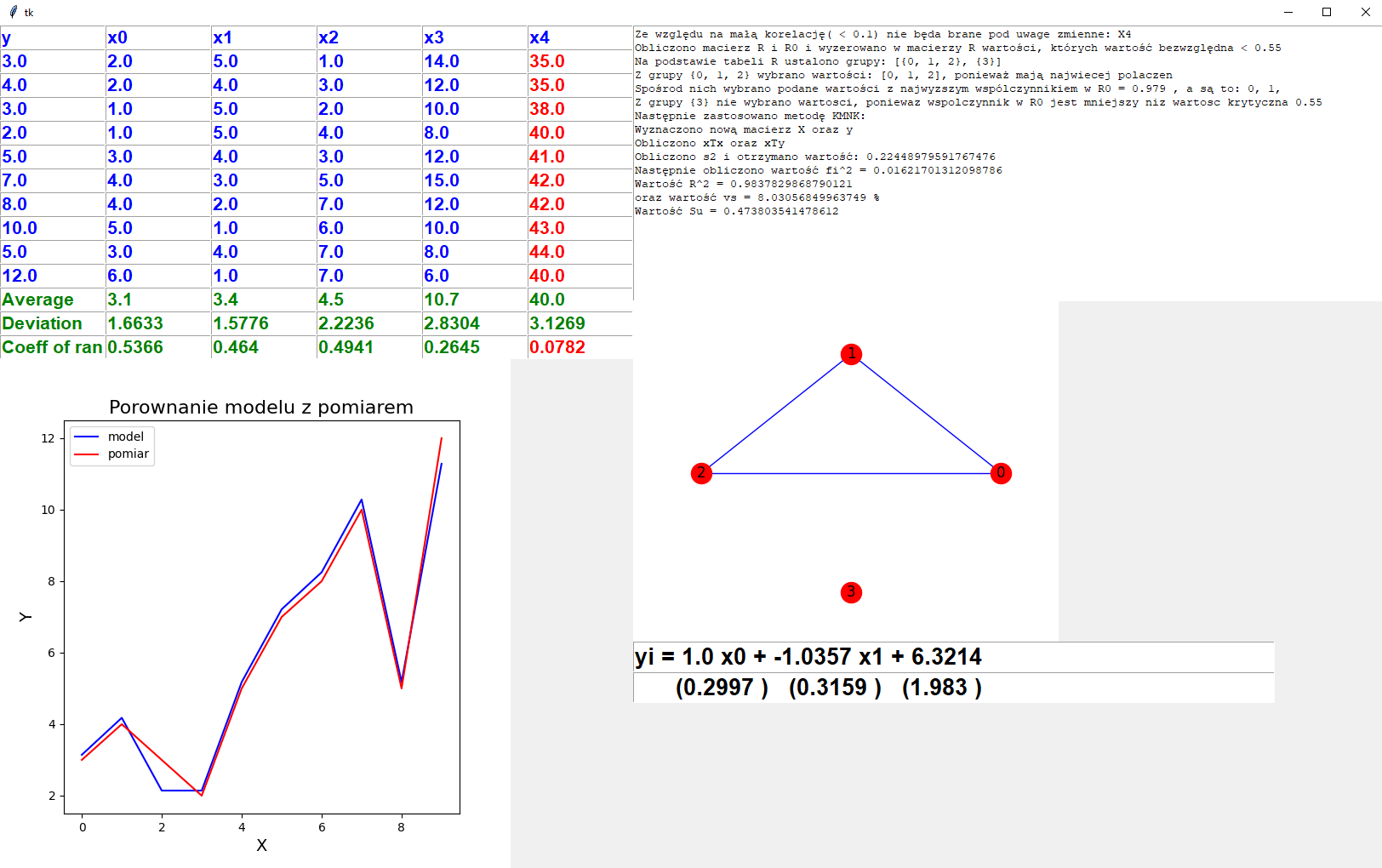
# Realizacja

Program zrealizowano w 4 plikach źródłowych:  
- ***app.py*** – plik uruchomieniowy zawierający instancję graficznego interfejsu,

- ***matrixOperations.py*** – plik odpowiedzialny za wszystkie obliczenia i operacje na danych, w nim jest zapisana cała logika i „logi” operacji, które są później wyświetlane w segmencie „Oblicz” na graficznym interfejsie użytkownika(GUI),

- ***results.py*** – klasa stworzona do bardziej przejrzystego przechowywania obliczonych wartości,

- ***graphicalInterface.py*** – odpowiedzialny za generowanie GUI na podstawie otrzymanych wyników i obliczeń.

W celu uruchomienia aplikacji należy otworzyć folder Program, upewnić się, że plik „dane.txt” znajduje się w tym folderze i posiada poprawny format(W przypadku, gdy brakuje pliku „dane.txt” lub jest niepoprawny, to powinno zostać wyświetlone okienko informujące o tym błędzie), po czym uruchomić plik Projekt\_ekonometria.exe. Jeśli wszystko przejdzie pomyślnie zostanie wyświetlony graficzny interfejs podobny jak na rysunku 3.   


Rysunek 3 Graficzny interfejs aplikacji

Dla każdej kolumny ze zmiennymi x0, x1,.. xn zostanie obliczona średnia wartość, odchylenie standardowe i współczynnik zmienności. Jeśli ten współczynnik jest mniejszy od wartości 0.1 to dana zmienna zostanie dodana do „blacklist” i nie będzie uwzględniana w kolejnych obliczeniach(na rysunku zmienna x4, została dodatkowo oznaczona czerwonym kolorem).

Następnie dane bez wykreślonej(wykreślonych) wartości trafiają do funkcji obliczającej R0 oraz R i są przekazywane do analizującej grafy(Uwaga: indeksowanie po „wykreśleniu” nie zmienia się, tzn. w przypadku, gdyby wykreślono x0, to x0 w grafie będzie oznaczać pierwszą „nieskreśloną zmienną” z danych, x1 drugą „nieskreśloną” itd.). Funkcja ta szuka połączeń między zmiennymi, gdy wartość bezwzględna z korelacji między nimi w R jest większa od wartości krytycznej, ustalonej jako 0.55 w kodzie programu.

Po przeanalizowaniu połączeń są tworzone grupy zmiennych i z każdej z grup wybierane są zmienne, które mają najwięcej połączeń z innymi zmiennymi i mają najwyższy współczynnik korelacji w R0(w przypadku, gdy istnieje parę takich zmiennych, to wszystkie są brane pod uwagę, a w przypadku, gdy jest jedna zmienna w grupie, to jest brana pod uwagę tylko jeśli współczynnik korelacji w R0 jest większy niż wartość krytyczna.).

W kolejnym kroku ze zmiennych wybranych przy pomocy metody analizy grafów powstaje nowa macierz X zawierająca te zmienne oraz kolumnę składającą się z samych jedynek(wyraz wolny).

Ta macierz przekształcona pozwala obliczyć współczynniki modelu:

a = (XTX)-1 XTy

oraz średnie błędy szacunku parametrów D(ai) =

gdzie:

D2(ai) =

Otrzymane równanie modelu w połączeniu z wartościami zmiennych objaśniających jest przedstawiane na wykresie jako „model”. Natomiast wartości „y” są przedstawiane jako „pomiar”.

# Kod źródłowy – uruchomienie

Możliwe jest także uruchomienie programu z poziomu kodu źródłowego. Aby było to możliwe jest konieczne posiadanie zainstalowanego programu **Python** w wersji 3.7+ (można sprawdzić wpisując w konsolę „*python –version*”) oraz programu **pip** (podobna komenda – „*pip –version*”). Jeśli mamy oba te programy to należy zainstalować wymagane rozszerzenia(komenda „*pip install -r requirements.txt*”). Gdy wszystko przebiegło pomyślnie wystarczy wpisać(będąc w konsoli w folderze source\_code) „*python app.py*” i aplikacja powinna zostać uruchomiona z poziomu kodu źródłowego.

# Wnioski

Dla załączonych danych testowych obliczony model niemalże pokrywa się z pomiarami. Oznacza, to że przyjęta metoda jest dla tych danych skuteczna. Najtrudniejszym i najciekawszym fragmentem programu jest funkcja analizująca grafy, gdyż metoda ta jest stosunkowo prosta do zrealizowania przez człowieka, jednak algorytm ją realizujący jest stosunkowo skomplikowany. Aplikacja powinna sobie radzić ze skalowaniem(zwiększaniem liczby zmiennych oraz sytuacją, gdy z grafów nie zostanie wybrana żadna zmienna(może się to zdarzyć, gdy są same „pojedyncze grupy” i wszystkie współczynniki korelacji w R0 są mniejsze od wartości krytycznej).

Aplikacja zawiera opisy przeprowadzanych działań na danych i sposobu doboru zmiennych objaśniających(z grafu) w obszarze „Oblicz”. Macierze X, R0, R nie zostały wyświetlone na interfejsie graficznym, żeby nie wprowadzać chaosu i nie pozwolić na przejrzystą analizę danych, jednak w przypadku uruchomienia z kodu źródłowego, te dane są dostępne w konsoli.