Politechnika Poznańska
Wydział Informatyki
Instytut Informatyki

Praca dyplomowa inżynierska

SYSTEM WSPIERAJĄCY TESTOWANIE ALGORYTMÓW OPTYMALIZUJĄCYCH USZEREGOWANIE REKLAM TELEWIZYJNYCH

Oskar Kostowski, 127334
Wojciech Obst, 127303
Laura Rakiewicz, 126870
Piotr Terczyński, 127275

Promotor
dr hab. inż. Małgorzata Sterna, prof. nadzw.

Poznań, 2019

Tutaj karta pracy dyplomowej;
oryginał wstawiamy do wersji dla archiwum PP, w pozostałych kopiach wstawiamy ksero.

Spis treści

1 Wstęp **1**

1.1 Cel i zakres pracy . . . 2
1.2 Struktura pracy . . . 2
1.3 Zespół . . . 3
1.4 Metodyka pracy . . . 4
1.5 Narzędzia organizacji pracy . . . 5

2 Opis problemu **7**

2.1 Specyfikacja wymagań . . . 8
2.2 Przyjęte podejście . . . 9

3 Architektura systemu **12**

3.1 Zbiory danych . . . 12
3.2 Przygotowanie zbiorów danych . . . 14
3.2.1 Zmiany dokonane w bazie opisującej program telewizyjny . . . 15
3.2.2 Zmiany dokonane w bazie opisującej oglądalność . . . 15
3.2.3 Nowe schematy zbiorów danych . . . 16
3.2.4 Kodowanie danych . . . 17
3.2.5 Dodatkowe zbiory danych . . . 18
3.3 Przepływ danych . . . 18
3.4 Moduł statystyk . . . 18
3.4.1 Wektor oglądalności . . . 19
3.4.2 Obliczanie oglądalności na podstawie wektora . . . 20
3.4.3 Wpływ importu wektora oglądalności na wydajność . . . 21
3.4.4 Opis instancji . . . 22
3.4.5 Statystyki reklam na żądanie . . . 22
3.5 Moduł generatora instancji . . . 23
3.5.1 Wpływ widoku zmaterializowanego na wydajność generatora instancji . . . 24
3.6 Moduł wizualizacji . . . 25
3.6.1 Podmoduł wizualizacji statystyk instancji . . . 25
3.6.2 Podmoduł wizualizacji statystyk szczegółowych . . . 25

4 Wykorzystane narzędzia **26**

5 Dokumentacja użytkownika **28**

5.1 Interfejs graficzny . . . 28
5.1.1 Widok generatora . . . 28
5.1.2 Widok wizualizacji instancji . . . 31

5.1.3 Widok wizualizacji statystyk 33
5.2 Wizualizacja danych 37
5.2.1 Program telewizyjny 37
5.2.2 Podsumowanie instancji 39
5.2.3 Wykresy statystyk instancji 40
5.2.4 Wykresy statystyk generowanych na żądanie 44
5.2.5 Szczegółowe statystyki opisowe 45
5.2.6 Statystyki niewizualizowane przez system 46
5.2.7 Macierz sąsiadowania 46

6 Podsumowanie **49**

Literatura **50**

A Format pliku instancji testowej **51**

B Instalacja i uruchamianie **53**

Rozdział 1

Wstęp

Reklama telewizyjna to narzędzie marketingowe o wielkim zasięgu, umożliwiające dotarcie do szerokiego grona potencjalnych odbiorców. Jej efektywność w znacznej mierze zależy od tego, kiedy taka reklama jest emitowana, a co za tym idzie, od oglądalności wybranego kanału telewizyjnego, zwłaszcza przez grupę docelowych odbiorców kampanii reklamowej. Jeśli reklama kierowana jest przede wszystkim do konkretnej grupy społecznej, to reklama odniesie dużo większy sukces, jeśli zostanie wyemitowana na kanale telewizyjnym licznie oglądanym w danej chwili przez przedstawicieli tej grupy.

Z tego powodu prowadzone są badania oglądalności, często zlecane zewnętrznym firmom przez reklamodawców, w celu uzyskania informacji umożliwiających zaplanowanie emisji reklam tak, aby dotarły do jak największej liczby odbiorców, zwłaszcza do grup o szczególnych cechach demograficznych, takich jak na przykład wiek czy wykształcenie, czyniących je prawdopodobnymi nabywcami reklamowanych towarów lub usług. Takie badania polegać mogą na utworzeniu grupy kontrolnej o znanych cechach demograficznych i dokonywaniu pomiarów, poprzez rejestrowanie czasów oglądania poszczególnych kanałów telewizyjnych przez uczestników badania. Pomiary te mogą też być rozszerzone o informacje o emitowanych w tym czasie programach i reklamach telewizyjnych, które obejrzał uczestnik badania. Ponadto z każdym widzem biorącym udział w pomiarach oglądalności związana jest waga, wskazująca jaką część społeczeństwa dany widz reprezentuje.

Zebrane w takich badaniach dane historyczne są więc niezwykle przydatne w planowaniu kampanii reklamowych, ich analiza pozwala przewidywać rozkład oglądalności w czasie, z uwzględnieniem poszczególnych grup demograficznych i, co za tym idzie, zaplanować emisję reklam tak, aby osiągnęły jak najlepszy efekt.

Planowanie uszeregowania emisji reklam telewizyjnych może być zamodelowane jako problem szeregowania zadań [1, 2, 3, 4]. Problem taki może być optymalizowany względem różnych kryteriów, jak na przykład maksymalizacja oglądalności reklam wybranego reklamodawcy, a do jego optymalizacji można tworzyć dedykowane mu algorytmy, wykorzystujące wspomniane wcześniej dane historyczne [5].

Takie zbiory danych, w celu pełnego ich wykorzystania, wymagają jednak rozległej analizy, umożliwiającej wykrycie zależności między oglądalnością, a takimi cechami emitowanego programu telewizyjnego, jak na przykład długości reklam, tematyka poprzedzających lub następujących programów (w tym gatunek w przypadku filmów), czy też liczba wystąpień danej reklamy w jednej przerwie reklamowej lub dłuższym okresie czasu, a także typ reklam bezpośrednio sąsiadujących ze sobą lub emitowanych w ramach jednego bloku reklamowego.

Przetwarzanie tak dużej ilości danych w celu przeprowadzenia ich analizy, wspierającej działa-

nie algorytmów optymalizujących uszeregowanie reklam, realizowane w ramach tych algorytmów powodowałoby zbędne przedłużenie czasu ich działania, ponadto wymagałoby dodatkowej pracy od twórcy tych metod. Dużo lepszym rozwiązaniem byłoby więc wcześniejsze przetworzenie danych historycznych (lub dowolnego ich podzbioru, utworzonego na przykład poprzez ograniczenie przedziału czasu, z którego miałyby pochodzić) i stworzenie na ich podstawie instancji testowych, zawierających już wszystkie informacje, potrzebne do działania algorytmów optymalizacyjnych. Podobnie dostarczenie programiście kompleksowych statystyk związanych z danymi historycznymi, a także ich wizualizacji, mogłyby ułatwić mu pracę nad konstrukcją wydajnych metod.

Zespół podjął się więc realizacji tematu, którego celem jest stworzenie systemu zapewniającego wymienione udogodnienia twórcom algorytmów optymalizacji kampanii reklamowych.

1.1 Cel i zakres pracy

Celem pracy było stworzenie systemu wspierającego testowanie algorytmów planujących telewizyjne kampanie reklamowe. Główną funkcjonalnością tego systemu jest generowanie instancji testowych dla aplikacji zewnętrznych w oparciu o dane historyczne. Instancje te będą wykorzystywane przez zewnętrzne algorytmy szeregujące, których celem jest znalezienie odpowiedniego uszeregowania dla reklam.

Aplikacja posiada także rozbudowany system analizy danych, umożliwiający uzyskanie szczegółowego opisu statystycznego danych zawartych w wygenerowanej instancji, a także, przy pomocy dodatkowych filtrów, uzyskanie dodatkowych informacji o interesujących użytkownika obiektach, takich jak na przykład reklamy wybranego produktu. Aplikacja umożliwia również wizualizację danych. Zakres pracy objął zatem:

- analizę dostarczonych danych historycznych i przygotowanie na ich podstawie danych wejściowych dla systemu
- zaprojektowanie i implementację modułu generatora instancji testowych
- opracowanie formatu zapisu instancji testowych
- zaprojektowanie i implementację filtrów demograficznych umożliwiających wyodrębnienie z danych historycznych dotyczących oglądalności informacji o oglądalności w ramach różnych grup docelowych
- zaprojektowanie i implementację modułów generujących statystyki dla:
  - instancji testowej
  - całego zbioru danych historycznych
- zaprojektowanie i implementacja modułu wizualizacji instancji i danych statystycznych

1.2 Struktura pracy

Praca inżynierska składa się z 6 rozdziałów.

W rozdziale 1, poza krótkim wprowadzeniem, przedstawiono również skład zespołu, precyzując zadania przydzielone poszczególnym jego członkom, oraz wspomniano przyjętą metodykę realizacji projektu i wykorzystane narzędzia wspomagające ten proces.

W rozdziale 2 wymieniono wymagania funkcjonalne i pozafunkcjonalne jakie postawiono wobec systemu, zarysowano przyjęte, jak i rozważane sposoby realizacji projektu, a także napotkane

przez zespół projektowy problemy.

W rozdziale 3 opisano architekturę zaprojektowanego systemu, przedstawiono opis poszczególnych modułów, a także zaprezentowano strukturę zbiorów danych historycznych, pełniących w systemie rolę baz danych, przedstawiono proces ich analizy i przetwarzania, w celu usystematyzowania ich i przygotowania do użycia jako danych wejściowych.

Rozdział 4 został poświęcony opisowi technologii i narzędzi informatycznych wykorzystanych przez zespół w procesie tworzenia aplikacji.

W rozdziale 5 zaprezentowano interfejs aplikacji oddanej do dyspozycji użytkownika i opisano zawartą w niej funkcjonalność. Zawiera on też przykładowe wykresy generowane na potrzeby wizualizacji danych.

Rozdział 6 podsumowuje pracę inżynierską, zawiera wnioski sformułowane przez członków zespołu po realizacji projektu, a także opisuje stopień realizacji postawionych przed systemem wymagań oraz przedstawia możliwe kierunki dalszego rozwoju aplikacji.

Jako dodatki do pracy inżynierskiej zamieszczono opis formatu generowanych przez system instancji testowych, a także wskazówki dotyczące procesu instalacji i uruchamiania aplikacji.

1.3 Zespół

Członkowie zespołu projektowego, w ramach pracy inżynierskiej, zrealizowali następujące zadania:

- Oskar Kostowski zajął się tworzeniem modułu generatora instancji testowych wraz z filtrami umożliwiającymi wybór przedziału czasu jak i kanałów telewizyjnych, dla których ma zostać wygenerowana instancja. Przeprowadził także analizę zbioru danych historycznych, na podstawie której przygotował dane wejściowe dla systemu, poprzez usunięcie niepotrzebnych lub błędnych danych, odfiltrowanie i naprawienie nieprawidłowych wpisów, a także zakodowanie danych eksportowanych do instancji testowej.

- Wojciech Obst zajął się przygotowaniem układu graficznego interfejsu użytkownika i jego późniejszym zaimplementowaniem, połączeniem z modułami funkcjonalnymi oraz stworzeniem modułu wizualizacji. Dodatkowo zrealizował funkcjonalność odkodowującą dane na potrzeby wizualizacji.

- Laura Rakiewicz stworzyła metody umożliwiające generowanie statystyk na podstawie danych historycznych dla reklam telewizyjnych. Przygotowała między innymi statystyki wyznaczające liczbę reklam dla danego reklamodawcy, liczbę wystąpień unikalnych reklam w podanym przedziale czasu oraz rozkłady czasowe liczby wystąpień dla poszczególnych typów reklam. Stworzyła także moduł umożliwiający anonimizację danych. Dodatkowo zajmowała się organizacją pracy zespołu.

- Piotr Terczyński stworzył narzędzia do przetwarzania, importu i eksportu uproszczonych danych o oglądalnościach. Przygotował funkcje generujące statystyki dotyczące czasu, oglądalności i współzależności występowania reklam różnych typów i od różnych reklamodawców do pliku instancji. Stworzył też funkcję generującą rozkład emisji reklamy na kanałach. Zajął się filtrowaniem instancji przed przekazaniem do funkcji statystycznych dotyczących reklam i reklamodawców. Podjął się też integracji modułów statystycznych z generatorem i interfejsem użytkownika.

Ponadto, podczas wspólnej dyskusji opracowano format zapisu instancji testowych. Każdy z członków zespołu opisał swoją realizację przydzielonych zadań w odpowiednich rozdziałach pracy inżynierskiej.

1.4 Metodyka pracy

Ze względu na pracę w grupie zdecydowano się na wprowadzenie własnej metodyki pracy, która usprawni współpracę w zespole. Inspirowano się frameworkiem Scrum [6, 7], dostosowano go jednak do potrzeb zespołu. Scrum jest frameworkiem pozwalającym na zwinne podejście do procesu wytwarzania produktu zgodne z Manifestem Agile. Gromadzi on podstawowe wartości, które umożliwiają budowanie oprogramowania w udoskonalony sposób. Głównymi założeniami są przede wszystkim: "przedkładanie ludzi i ich interakcji nad procedury i narzędzia, działające oprogramowanie nad wyczerpującą dokumentację, a także przedkładanie współpracy z klientem nad negocjacje umów oraz reagowanie na zmiany nad realizowanie planu" [8]. Manifest można wykorzystywać także jako kryterium oceny praktyk w zespole i organizacji.

We Frameworku Scrum występują trzy główne role w zespole. Są to: członkowie zespołu czyli w przypadku rozwoju oprogramowania programiści, lider zespołu (ang. Scrum Master) oraz właściciel produktu (ang. Product Owner). Właściciel produktu jest osobą odpowiedzialną za tworzenie list zadań do zrobienia ułożonych w kolejności możliwej do zrealizowania przez zespół. Właściciel ma za zadanie stworzyć zadania tak, aby dostarczyć na rynek produkt jak najlepiej odpowiadający specyfikacji klienta.

Wyróżniamy pięć podstawowych typów wydarzeń we frameworku Scrum, są to: Sprint, Planowanie Sprintu, Codzienny Scrum, Przegląd Sprintu i Retrospektywa Sprintu. Poniżej zostaną opisane typy wydarzeń wraz z odniesieniem do sposobu ich wykorzystania na potrzeby pracy inżynierskiej.

Sprint to okres wykonywania pracy nad rozwojem produktu. Kolejne Sprinty następują po sobie bez przerw. Celem każdego Sprintu jest przyniesienie wartości biznesowej. Każdy Sprint musi mieć jasno określony cel oraz zadania do zrealizowania. Wraz z przebiegiem Sprintu zaplanowane zadania mogą ulegać zmianom, a także mogą powstawać nowe zadania. Maksymalny czas Sprintu ma długość czterech tygodni. Zespół inżynierski zdecydował się na przeprowadzanie Sprintów o długości dwóch tygodni. Krótszy czas trwania Sprintu pozwolił na definiowanie mniej złożonych celi, co ułatwiło proces planowania oraz otrzymanie szybszej informacji zwrotnej, czy prowadzone prace przynoszą oczekiwane efekty.

Planowanie Sprintu (ang. Sprint Planning) to wydarzenie rozpoczynające każdy Sprint. Podczas spotkania planowany jest obszar rozwoju oprogramowania i zadania do wykonania w ramach niego. Uczestniczy w nim cały zespół Scrum. Złożony jest z części, w której właściciel produktu przedstawia zespołowi, co należy zrobić w danym Sprincie. Członkowie zespołu określają ile elementów celu są w stanie zrealizować podczas jednej iteracji Sprintu. Ostatecznie wyznaczany jest cel Sprintu. W części drugiej planowania zespół deweloperski planuje sposób osiągnięcia celu poprzez definicję zadań do wykonania. Zadania należy wyceniać ze względu na czas potrzebny do ich zrealizowania. Polecane jest jednak, aby szacować czas realizacji zadania na większy niż oczekiwany, ze względu na możliwość wystąpienia nieprzewidzianych problemów. Taka praktyka minimalizuje ryzyko braku wykonania zadania.

Codzienny Scrum (ang. Daily Scrum) to krótkie spotkanie, w którym uczestniczy zespół deweloperski, które jak nazwa wskazuje odbywa się codziennie. Podczas spotkania każdy z uczestników ma odpowiedzieć na trzy podstawowe pytania:

1. Co zrobiono dnia poprzedniego, co zbliżyło zespół do osiągnięcia celu?
2. Co dany członek zespołu planuje dzisiaj zrobić?
3. Jakie napotkano przeszkody na drodze do osiągnięcia celu?

Głównymi celami planowania jest zwiększenie wiedzy zespołu deweloperskiego oraz zidentyfikowanie przeszkód, które stoją na drodze do osiągnięcia celu. Podczas pracy inżynierskiej zrezygnowano z Codziennego Scrum'u ze względu na ograniczone możliwości spotkań członków zespołu. Członkowie zespołu mieli nieregulowany czas pracy, dlatego codzienny Scrum często nie wnosiłby korzyści.

Przegląd Sprintu (ang. Sprint Review) to spotkanie, podczas którego zespół deweloperski prezentuje wyniki wykonanej pracy. W spotkaniu uczestniczą zespół deweloperski, właściciel produktu oraz lider zespołu. Celem spotkania jest podsumowanie osiągnięć danego Sprintu i prezentacja produktu.

Retrospekcja Sprintu (ang. Sprint Retrospective) to spotkanie, w którym uczestniczy Zespół Scrum, które ma za zadanie podsumowanie wydarzeń z ukończonego Sprintu. Głównym celem jest wyciągnięcie wniosków z popełnionych błędów i zebranie pomysłów umożliwiających poprawę w przyszłych Sprintach. Sposoby poprawy mają być konkretne i dotyczyć poszczególnych osób i sytuacji. Retrospekcja Sprintu była przeprowadzana w zespole inżynierskim i przyniosła pozytywne rezultaty. Między innymi usprawniono komunikację w zespole i zwiększono częstotliwość udzielania sprzężenia zwrotnego.

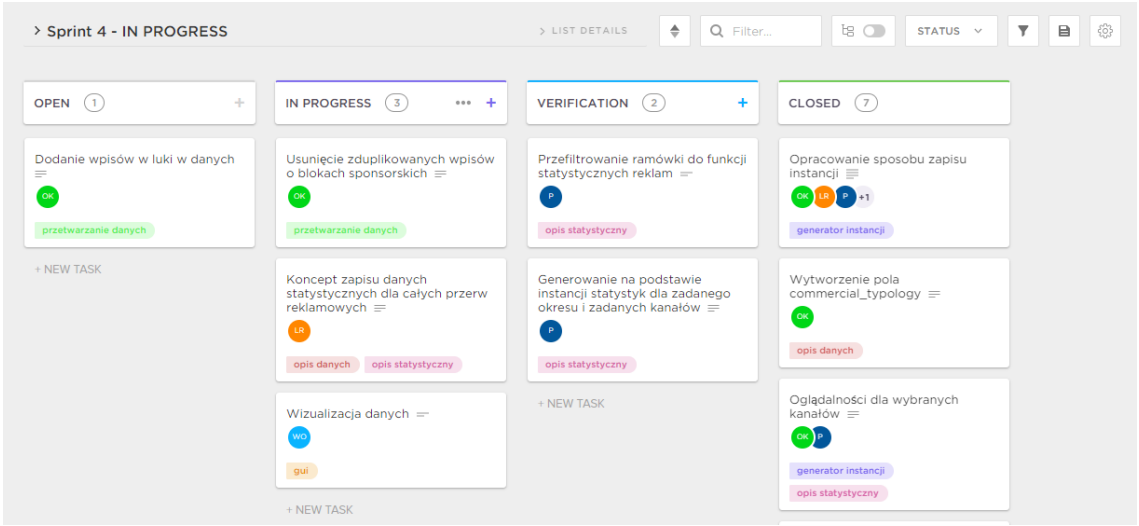
1.5 Narzędzia organizacji pracy

Podczas pracy nad projektem wykorzystano opisane poniżej narzędzia.

Jako platformy służącej do zarządzania projektem użyto ClickUp. Platforma ta ma bardzo rozbudowaną funkcjonalność nawet w darmowej wersji podstawowej. Najważniejszymi z nich jest możliwość tworzenia projektów i zadań. Wyświetlanie zadań możliwe jest w trzech trybach: listy, tablicy i na kalendarzu. Do potrzeb pracy inżynierskiej ustalono następujące użycie platformy. Tworzono folder na zaplanowany z dwutygodniowy okres prac nad projektem. Następnie do folderu dodawano zadania opisujące funkcjonalności do zrealizowania (Rys.1.1). Stworzono etykiety do zadań w celu szybszego wyszukiwania ich na liście wszystkich zadań do wykonania według wymagań w zakresie pracy. Każde stworzone zadanie miało następnie dodawany precyzyjny opis oraz etykietę. Następnie przydzielano zadania konkretnym osobom i ustalano ich priorytety.

Spotkania organizacyjne takie jak Planowanie Sprintu przeprowadzono głównie online. Wykorzystano do tego aplikację Discord, która umożliwia prowadzenie rozmów głosowych.

Ponadto podczas codziennej pracy nad kodem używano systemu kontroli wersji git. Pozwoliło to na śledzenie zmian w kodzie programów, a także umożliwiło pracę zespołową nad projektem. Prywatne repozytorium umieszczono bezpłatnie w serwisie internetowym Bitbucket.



Rysunek 1.1: Tablica zadań z folderu

Rozdział 2

Opis problemu

Stworzenie algorytmu optymalizującego uszeregowanie emisji reklam telewizyjnych to zadanie trudne, wymagające wykrycia i uwzględnienia wielu czynników i zależności, które mają wpływ na osiąganą wartość przyjętego kryterium optymalizacji. Należy rozpoznać ograniczenia emisji reklam, na przykład w jakich godzinach i na jakich kanałach telewizyjnych można emitować reklamy danego typu produktu. Trzeba także uwzględnić preferencje widzów, wykorzystać informacje o rzeczywistej oglądalności emitowanego w przeszłości programu telewizyjnego, aby określić tendencje i uwzględniać je w tworzonym uszeregowaniu, zwracając uwagę na czynniki takie jak czas emisji, sąsiedztwo programów określonego typu lub odpowiedniego gatunku filmów, długość całej przerwy reklamowej i to, jaki jest rozkład oglądalności w czasie trwania przerwy reklamowej.

W tym celu niezbędny jest dostęp do danych historycznych dotyczących emitowanego w przeszłości programu telewizyjnego oraz jego oglądalności. Na ich podstawie twórca algorytmu optymalizacji może zarówno wyodrębnić informacje, które powinny być uwzględniane przez algorytm, jak i ocenić jakość uzyskanego uszeregowania emisji reklam telewizyjnych, poprzez porównanie go z rzeczywistym uszeregowaniem pochodzącym z danych historycznych. Istotne jest zatem, aby zbiór owych danych dotyczył możliwie długiego okresu czasu, szerokiego wyboru kanałów telewizyjnych oraz dużej i różnorodnej grupy widzów. Operowanie bezpośrednio na danych historycznych przez algorytm optymalizacji spowodowałoby znaczący spadek jego wydajności.

Dane historyczne otrzymane przez zespół na potrzeby pracy inżynierskiej składały się z ogromnych zbiorów danych, zawierających informacje o programach oraz reklamach telewizyjnych emitowanych w 2014 roku, a także o ich oglądalności oraz cechach demograficznych ich widzów. Zbiór dotyczący telewizyjnej ramówki zawierał ponad 25 milionów wpisów, natomiast zbiór dotyczący oglądalności - ponad 20 milionów.

W przypadku pierwszego zbioru każdy wpis dotyczył emisji pojedynczego programu lub reklamy telewizyjnej i zawierał informacje o ich czasie emisji oraz trwania, w przypadku reklamy były to także dane określające reklamowany produkt, jego producenta oraz sektor rynku, z którym ten produkt jest związany, a także informacje o pozycji reklamy w danej przerwie reklamowej oraz o jej koszcie; w przypadku programu występowała informacja o jego typologii, określająca na przykład gatunek filmu.

Zbiór dotyczący oglądalności składał się natomiast z wpisów dotyczących oglądania przez daną osobę określonego kanału telewizyjnego w określonym czasie. Każdy wpis zawierał informacje o czasie rozpoczęcia, zakończenia i trwania oglądania, identyfikator uczestnika badania, którego dotyczy dany wpis, cech demograficznych danej osoby, takich jak na przykład jej wiek, wykształcenie i zawód; *wadze* związanej z widzem, czyli liczbie szacującej jak dużą część populacji opisują cechy demograficzne tego widza; nazwie oglądanego kanału telewizyjnego.

Na podstawie owych danych historycznych system powinien więc umożliwiać generację instancji testowych, zawierających zarówno dane z oryginalnych zbiorów, przydatne z punktu widzenia algorytmów optymalizacji, jak i dane statystyczne, dotyczące występujących w instancji reklam, programów oraz całych przerw reklamowych.

Podstawową funkcjonalnością omawianego systemu miało być przetwarzanie danych historycznych, przeprowadzenie na nich analizy statystycznej i wytworzenie instancji testowej, która będzie spełniała wymagania projektowanego algorytmu optymalizacji kampanii reklamowych, a także dostarczy mu wszelkich informacji potrzebnych do szeregowania reklam telewizyjnych w celu maksymalizacji (lub minimalizacji) przyjętego kryterium [2].

Dodatkowo system powinien umożliwiać programiście przeprowadzanie szczegółowej analizy pod kątem wykrywania konkretnych zależności w danych oraz ułatwiać ową analizę udostępniając wygodny system wizualizacji.

W szczególności, system powinien udostępniać filtry, umożliwiające uwzględnienie w instancji testowej jedynie oglądalności pochodzącej od osób o określonych cechach (dalej nazywane *filtrami demograficznymi*), a także filtrów umożliwiających wskazanie określonego typu reklam, bądź konkretnej reklamy, w celu uzyskania o nich dodatkowych informacji, pochodzących z całego zbioru danych historycznych, bądź jedynie z zadanego przedziału czasu.

Oznacza to zatem konieczność efektywnego przetwarzania masywnych danych oraz przeprowadzania na nich wielu operacji statystycznych, zapewniając możliwie krótkie czasy przetwarzania, a także minimalizując wykorzystanie pamięci operacyjnej.

2.1 Specyfikacja wymagań

W trakcie dyskusji z potencjalnymi użytkownikami aplikacji ustalono listę wymaganych funkcjonalności systemu, szczególnie zwracając uwagę na dostępne funkcje statystyczne, i sformułowano następujące wymagania funkcjonalne:

- eksport danych opisujących emitowany program oraz jego oglądalności dla jednego (lub kilku) wybranych kanałów telewizyjnych, z wybranego okresu czasu. Szczególnie istotne jest wyodrębnienie informacji dotyczących poszczególnych przerw reklamowych dla zadanego zbioru danych i utworzenie ich opisu obejmującego:
  - czas rozpoczęcia oraz czas trwania przerwy reklamowej
  - rozkład kosztu reklam w poszczególnych jednostkach czasu, wraz z obliczeniem wartości minimalnej, maksymalnej oraz średniej z odchyleniem standardowym
  - rozkład oglądalności w czasie danej przerwy reklamowej, wraz z obliczeniem wartości minimalnej, maksymalnej oraz średniej z odchyleniem standardowym

  Za wartość oglądalności uznaje się sumę *wag* przypisanych widzom uczestniczącym w badaniu (czyli szacowanej liczby widzów które dana osoba reprezentuje w społeczeństwie), oglądających dany kanał w danym czasie.
  W najprostszym przypadku należało zsumować *wagi* przypisane wszystkim widzom oglądającym dany kanał w danym czasie, niezależnie od cech demograficznych tych widzów; jeśli użytkownik wykorzystał *filtry demograficzne* należało najpierw wyznaczyć podzbiór informacji o oglądalności, pochodzącej tylko od osób o wskazanych cechach demograficznych.
- utworzenie opisu statystycznego instancji obejmującego:

  - identyfikację ponownych emisji tej samej reklamy (w oparciu o cechy takie jak nazwa producenta i produktu), reklam dotyczących tego samego typu produktu lub reklam pochodzących od tego samego reklamodawcy i określanie:
    * maksymalnej, minimalnej oraz średniej (wraz z odchyleniem standardowym) liczby powtórzeń danej reklamy podczas jednej przerwy reklamowej
    * rozkładu liczby emisji danej reklamy w ciągu dnia
    * liczby emisji danej reklamy dla zadanego okresu
    * podzbioru kanałów telewizyjnych, na których dana reklama była emitowana
  - rozkład liczby emisji reklam o rozróżnialnych czasach trwania i obliczenie wartości minimalnej, maksymalnej i średniej wraz z odchyleniem standardowym liczby emisji takich reklam
  - liczby reklam określonego typu, które sąsiadowały z reklamami pozostałych typów
  - określenie udziału procentowego łącznego czasu trwania reklam danego typu w ramówce

- generowanie zewnętrznego pliku zawierającego instancję testową, a także opis statystyczny tej instancji
- możliwość wizualizacji danych statystycznych i eksportowanego programu telewizyjnego

Sformułowano także następujące wymagania pozafunkcjonalne:

- wykrycie obecnych w danych typów programów telewizyjnych, typów reklam, a także nazw reklamodawców wraz z przypisaniem im indywidualnych wartości kodujących
- anonimizacja informacji zawartych w zbiorach danych, które miałyby być eksportowane do instancji testowej - instancja testowa powinna zawierać jedynie dane zakodowane
- ustalenie stałego formatu zapisu danych do instancji testowej i opisu statystycznego tej instancji
- prosty i czytelny interfejs użytkownika
- możliwie krótkie czasy przetwarzania danych i eksportu instancji testowych

2.2 Przyjęte podejście

W celu ułatwienia podziału pracy, na podstawie wymaganej funkcjonalności, wyodrębniono następujące moduły systemu:

- moduł generatora - moduł wykonujący podstawowe operacje systemu. Zawiera jedynie filtry umożliwiające wybór przedziału czasu i zbioru kanałów telewizyjnych, które mają zostać uwzględnione w generowanej instancji testowej. Jego działanie sprowadza się do wybrania podzbioru wpisów ze zbioru danych dotyczących programu telewizyjnego, spełniających owe wymagania, uzyskiwania z kolejnego modułu systemu - modułu statystycznego - informacji dotyczących oglądalności danego fragmentu programu telewizyjnego, a także rozróżniania wpisów zawierających informacje o reklamach bądź programach i na tej podstawie eksportu właściwych elementów wpisu do tworzonej instancji testowej.

- moduł statystyczny - moduł uzupełniający działanie generatora instancji. Na podstawie *filtrów demograficznych* ogranicza zbiór oglądalności do takich, które są związane tylko z widzami o zadanych cechach i dopasowuje ów podzbiór do telewizyjnej ramówki emitowanej we wskazanym okresie. Grupuje także wpisy dotyczące reklam, wyodrębniając w ten sposób informacje dotyczące poszczególnych przerw reklamowych i oblicza dla nich wymagane statystyki. Tworzy opis statystyczny instancji testowej, zbierając dane wymienione w rozdziale 2.1. Dostarcza wszystkie te dane do generatora, umożliwiając zapisanie ich w instancji testowej. Ponadto umożliwia także uzyskanie dodatkowych informacji ze zbioru danych historycznych.

- moduł wizualizacji - zapewnia możliwość generowania wykresów obrazujących statystyki wygenerowane przez moduł statystyczny, a także ramówkę telewizyjną eksportowaną przez generator. Generuje wizualizację bazując na pliku instancji testowej.

Zespół doszedł do wniosku, że operacje na tak masywnych danych w zwykłej bazie relacyjnej będą zbyt czasochłonne, dlatego postanowiono wykorzystać technologię PySpark, umożliwiającą przeprowadzanie różnych operacji, w tym wykonywanie zapytań SQL, na dużych zbiorach danych, w sposób wydajny, zarówno pod względem czasu przetwarzania, jak i wykorzystania pamięci operacyjnej.

Postanowiono też możliwie ograniczyć liczbę operacji wykonywanych na całym zbiorze danych, poprzez przygotowywanie osobnych plików z danymi potrzebnymi najczęściej przy podstawowej pracy systemu, które następnie mogłyby być od razu wczytywane przez technologię PySpark, zamiast być każdorazowo odfiltrowywane.

Z kolei do zaimplementowania modułu wizualizacji na początku planowano ograniczyć się jedynie do biblioteki Plotly, jednak z powodu zaistniałych problemów konieczne okazało się dodanie również pakietu PyPlot należącego do biblioteki Matplotlib. Opis zaistniałych problemów został zawarty w dalszej części pracy. Ze względu na dużą różnorodność danych do wizualizowania oraz samą ich liczbę zadbano o uniwersalność metod. W ten sposób wizualizacja sprowadziła się do czterech form:

- opis statystyki tekstem
- wykres słupkowy
- przedstawienie macierzy w postaci tabeli
- wykres Gantt'a

Tekstowy opis jest generowany w ramach tworzenia statystyki i przekazywany jako ciąg znaków do wyświetlenia w dedykowanym elemencie interfejsu. W końcowej wersji projektu, biblioteka Plotly użyta jest jedynie do tworzenia ramówki w formie wykresu Gantt'a. Ze względu na zapis do pliku HTML, jest to rozwiązanie nieoptymalne dla dużych zakresów danych, gdzie wielkość zakresu jest określana przez przedział czasowy oraz liczbę wybranych kanałów. Niestety poszukiwania alternatywnej metody wykonania ramówki zakończyły się niepowodzeniem. Dla tworzenia pozostałych wykresów i macierzy wybrano bibliotekę PyPlot ze względu na dużą elastyczność generowanych figur oraz łatwość implementacji.

W trakcie pracy nad modułem wizualizacji napotkano kilka problemów. Największym, który spowodował także dodanie dodatkowej biblioteki był brak możliwości rozszerzenia obszaru, na którym wykresy były rysowane przy użyciu Plotly. Obydwie biblioteki wykazały się zaskakującym brakiem możliwości dodania pasków przewijania do wykresu. W efekcie dla dużych zakresów

danych wizualizacja traci na czytelności. Dzięki zastosowaniu funkcji zapisu do pliku PDF zawartej w bibliotece Matplotlib, udało się zniwelować brak przewijania w pionie - osiągnięto to poprzez podział wykresów na maksymalnie dwa na stronę. Ten sam typ problemu generuje macierz. Jednak w tym przypadku wzrost rozmiaru następuje zarówno w pionie jak i poziomie, dla dużych instancji tabela nie mieści się na stronie dokumentu.

Rozdział 3

Architektura systemu

Charakter pracy zespołowej wymaga odpowiedniego wykorzystania zasobów ludzkich i organizacji pracy. Podział projektu na moduły wynika z niezależności funkcji w nich zawartych i podziału pracy pomiędzy członków zespołu. Dzięki temu zrównoleglenie pracy nad projektem było łatwiejsze. Pracę nad projektem można podzielić na trzy części związane z przygotowaniem trzech modułów funkcjonalnych: generatora, statystyk i wizualizacji. Poza tym bardzo dużą rolę odegrała wstępna analiza i przetworzenie otrzymanych danych historycznych, pełniących rolę bazy danych dla systemu.

3.1 Zbiory danych

System umożliwia pracę na historycznych danych dotyczących programów i reklam emitowanych przez 135 kanały telewizyjne w 2014 roku, a także ich oglądalności w owym okresie. Dane te zawarto w dwóch zbiorach danych, zapisanych w osobnych plikach csv dla każdego miesiąca:

- *Merged Schedule (MS)* - zawierającym dokładne informacje na temat telewizyjnej ramówki, szczególnie emitowanych reklam
- *Telemetry (TM)* - zbiorze czasów oglądania i danych demograficznych osób biorących udział w badaniu

Ponadto pośród otrzymanych przez zespół danych historycznych znalazł się także zbiór *Users Telemetry (UTM)*, będący wynikiem połączenia zbiorów *MS* oraz *TM*, w taki sposób, aby do każdego wpisu dotyczącego oglądania telewizji przez daną osobę dołączone były informacje o oglądanym programie lub reklamie telewizyjnej, pochodzące ze zbioru *MS*. Zbiór *UTM* nie zawierał jednak wielu kluczowych informacji, niezbędnych do dokładnego opisu reklam, dlatego zespół zdecydował się nie korzystać z tego zbioru.

Zbiór *TM*, otrzymany przez zespół na początku pracy, składał się z ponad 22 milionów wpisów. W każdym wpisie zawarto informacje takie jak:

- rok
- miesiąc
- dzień
- czas rozpoczęcia oglądania
- czas zakończenia oglądania

- czas oglądania
- zakodowane dane demograficzne i cechy oglądającego:
  - płeć
  - wiek
  - liczba członków rodziny
  - liczba i wiek posiadanych dzieci
  - posiadane wykształcenie
  - przychody
  - miejsce zamieszkania
  - wykonywany zawód
  - ile czasu miesięcznie spędza na oglądaniu telewizji
  - jakie kanały telewizyjne ogląda
- unikalny identyfikator oglądającego
- jaką część populacji reprezentuje dana osoba
- nazwa oglądanego kanału

Natomiast zbiór *MS* zawierał początkowo ponad 26 milionów wpisów, w których znajdowały się informacje takie jak:

- rok
- miesiąc
- dzień
- nazwa kanału
- czas rozpoczęcia emisji programu/reklamy
- czas zakończenia emisji programu/reklamy
- czas trwania emisji programu/reklamy
- opis reklamy
- marka produktu reklamowanego
- podmarka produktu reklamowanego
- sektor rynku produktu reklamowanego
- kategoria produktu reklamowanego
- klasa produktu reklamowanego
- koncern produktu reklamowanego
- numer przerwy reklamowej
- liczba reklam w danej przerwie reklamowej

- pozycja reklamy w przerwie reklamowej
- producent reklamy
- opis produktu reklamowanego
- koszt nominalny reklamy
- koszt rzeczywisty reklamy
- opis kategorii czasu reklamowego
- typ bloku reklamowego
- opis programu (dla wpisu o reklamie dotyczy najbliższego programu)
- typologia programu (kategorie i rodzaje programów)

3.2 Przygotowanie zbiorów danych

Zbiory danych zostały przeanalizowane w celu wykrycia ewentualnych nieprawidłowości, a także odrzucenia zbędnych informacji, niepotrzebnych z punktu widzenia algorytmów optymalizacji, korzystających z opracowanego systemu. Okazało się, że baza *Merged Schedule* zawierała liczne wpisy, które mogłyby utrudnić lub nawet uniemożliwić prawidłowe działanie tych algorytmów.

Pierwszym wykrytym problemem były nakładające się na siebie wpisy oznaczające bloki reklam sponsorskich, występujących przed rozpoczęciem lub po zakończeniu programu. Poza wpisami dotyczącymi poszczególnych reklam zawierających się w tym bloku dodawane były także pojedyncze wpisy, których czasy rozpoczęcia i zakończenia pokrywały się z rozpoczęciem i zakończeniem całego bloku sponsorskiego. Taki wpis pod względem zawartych w nim danych był nieodróżnialny od wpisu oznaczającego program telewizyjny. W celu rozpoznania go generator instancji musiałby analizować wpisy z nim sąsiadujące w poszukiwaniu reklam oznaczonych jako spoty sponsorskie, co oznaczałoby znaczący wzrost czasu przetwarzania. Przy dalszej analizie danych okazało się, że identyczny problem występuje w przypadku reklam oznaczonych jako ogłoszenia.

Następną wykrytą nieprawidłowością w bazie *MS* były luki w ramówce, długości od kilku sekund do minuty. Takie braki występowały bardzo często w trakcie bloków reklamowych, a także między reklamami a programami. Ustalono, że brakujące wpisy to autopromocja kanału telewizyjnego - zapowiedzi programów telewizyjnych, wyświetlanie loga stacji telewizyjnej.

Zauważono także przypadki nakładających się na siebie w czasie wpisów niezwiązanych z blokami sponsorskimi i ogłoszeniami, występujące najczęściej w programach całonocnych. Dla takich programów występował pojedynczy wpis, którego czas zakończenia pokrywał się z czasem rozpoczęcia następnego programu, nie uwzględniając bloków reklamowych występujących w czasie trwania tego programu.

Co więcej, takie bloki reklamowe miały przypisane nieprawidłowe wartości numeru bloku reklamowego - numeracja tych bloków rozpoczynała się od 1, niezależnie od liczby bloków reklamowych wyemitowanych już w danym dniu.

Ponadto, przeprowadzona analiza bazy *Merged Schedule*, pozwoliła ustalić, które pola tej bazy zawierają istotne dla algorytmu optymalizacji informacje.

W bazie *TM* nie znaleziono żadnych nieprawidłowości. Wpisy w tym zbiorze danych składają się z czasu rozpoczęcia i zakończenia oglądania przez widza uczestniczącego w badaniu telemetrycznym, nazwę oglądanego kanału telewizyjnego, pozwalających przypisać oglądalność do poszczególnych wpisów w bazie *MS*, a także informacji o uczestniku badania, którego ten wpis

dotyczy. Szczególnie istotne są pola wskazujące jak dużą część społeczeństwa reprezentuje dany oglądający, a także zawierające ciąg opisujący cechy tego widza, w tym przedział wiekowy, do którego należy oglądający, posiadane przezeń wykształcenie i inne dane demograficzne.

3.2.1 Zmiany dokonane w bazie opisującej program telewizyjny

W związku z obserwacjami opisanymi powyżej, podjęto działania mające na celu eliminację nieprawidłowości w zbiorze danych.

Dodatkowe wpisy przy blokach sponsorskich i ogłoszeniach, jako że nie zawierały żadnych istotnych informacji, a ich filtracja na etapie generowania instancji, na potrzeby zewnętrznych algorytmów optymalizacji, skutkowałaby zauważalnym spowolnieniem działania systemu, zostały usunięte.

Brakujące informacje dotyczące autopromocji stacji telewizyjnych zostały uzupełnione wpisami o stałej strukturze - posiadającymi domyślne wartości w polach związanych z cechami reklam, ale nieprzypisanych do żadnego bloku reklamowego. Te wpisy są rozpoznawane przez generator instancji i są w niej oznaczane literą A, umożliwiając twórcy zewnętrznego algorytmu, korzystającego z tej instancji, ich osobne przetwarzanie.

Wpisy dotyczące programów, które nie uwzględniały przerw reklamowych, zostały podzielone na większą liczbę wpisów, w taki sposób aby ich czasy trwania nie nakładały się na czasy trwania owych przerw.

Aby zapewnić łatwiejsze identyfikowanie bloków reklamowych, niewymagające porównywania dat, wygenerowano nową numerację bloków reklamowych, która nigdy nie jest zerowana w ramach danego kanału telewizyjnego. Pozwoliło to uzyskać niepowtarzalny numer identyfikacyjny dla każdego bloku na danym kanale, a także rozwiązało to problem nieprawidłowej numeracji, związany z powyżej opisanym problemem nakładających się wpisów.

Usunięto także pola rok, miesiąc i dzień, uznane za zbędne, natomiast pola opisujące sektor rynku, kategorię i klasę reklamowanego produktu zostały złączone w pole opisujące dokładnie typ reklamy, na potrzeby funkcji statystycznych systemu wiążących się z reklamami produktów określonego typu, podobnie na podstawie połączenia pól zawierających informacje o koncernie, marce i podmarce utworzono identyfikator reklamy pozwalający odnaleźć wszystkie wystąpienia tej samej reklamy.

3.2.2 Zmiany dokonane w bazie opisującej oglądalność

W przypadku bazy *Telemetry*, za najistotniejszą uznano sprawną ekstrakcję danych demograficznych opisujących widzów, w celu uproszczenia implementacji funkcjonalności związanej z filtrowaniem oglądalności względem tych danych. W związku z tym wyznaczono niezbędne informacje zawarte w polu opisującym widza, odrzucając pozostałe, które zostały uznane za nieistotne dla użytkownika. W wyniku tego dane zawartych w oryginalnych danych historycznych zostało zastąpione informacjami o:

- przedziale wiekowym
- liczbie członków rodziny
- posiadaniu dzieci
- posiadaniu zwierząt domowych
- posiadanym wykształceniu

- wykonywanym zawodzie
- przychodach
- wielkości zamieszkiwanej miejscowości

Zarówno informacje o wieku, przychodach, wielkości miejscowości zamieszkanej przez daną osobę jak i liczbie członków jej rodziny zostały podzielone na dwa pola, ponieważ te dane były zapisywane w oryginalnej bazie jako przedziały. Pozostałe dane zapisane są w pojedynczych polach.

Dodatkowo usunięto wpisy dotyczące kanałów telewizyjnych, które nie występowały w zbiorze *MS*. Usunięto także pola rok, miesiąc i dzień, ponieważ informacje te łatwo uzyskać z pól dotyczących początku i końca oglądania, zapisanych jako daty z dokładnością do sekundy.

3.2.3 Nowe schematy zbiorów danych

W efekcie zmian opisanych powyżej, ukształtowały się nowe schematy baz opisujących oglądalność (*TM*) oraz program telewizyjny (*MS*), usunięto także część wpisów występujących w tych zbiorach, w przypadku *MS* dodano także wiele nowych. W efekcie *TM* zawiera niecałe 20 milionów wpisów, a *MS* - ponad 31 milionów.

Baza *TM* po wprowadzonych zmianach zawiera pola opisujące:

- czas rozpoczęcia oglądania
- czas zakończenia oglądania
- czas oglądania
- informacje o oglądającym takie jak:
  - płeć
  - wiek
  - liczba członków rodziny
  - czy ma dzieci w przedziale 4-14 lat
  - czy ma dzieci w przedziale 0-3 lat
  - czy ma psa
  - czy ma kota
  - wykonywany zawód
  - posiadane wykształcenie
  - przychody
  - wielkość zamieszkiwanej miejscowości
  - czy zajmuje się zakupami
- unikalny identyfikator oglądającego
- jaką część populacji reprezentuje dana osoba
- nazwa oglądanego kanału

Natomiast baza *MS* zawiera obecnie pola opisujące:

- nazwa kanału

- czas rozpoczęcia emisji programu/reklamy
- czas zakończenia emisji programu/reklamy
- czas trwania emisji programu/reklamy
- opis reklamy
- typologia reklamy
- koncern produktu reklamowanego
- marka produktu reklamowanego
- podmarka produktu reklamowanego
- identyfikator reklamy
- numer przerwy reklamowej
- liczba reklam w danej przerwie reklamowej
- pozycja reklamy w przerwie reklamowej
- producent reklamy
- opis produktu reklamowanego
- koszt nominalny reklamy
- koszt rzeczywisty reklamy
- opis kategorii czasu reklamowego
- typ bloku reklamowego
- opis programu (dla wpisu o reklamie dotyczy najbliższego programu)
- typologia programu (kategorie i rodzaje programów)

3.2.4 Kodowanie danych

Ze względu na poufność danych, a także w celu zmniejszenia rozmiaru pliku z wygenerowaną instancją, dane w bazie *Merged Schedule* zostały zakodowane. Dla każdego pola tej bazy, którego zawartość jest umieszczana w instancji, wygenerowano listę wszystkich jego rozróżnialnych wartości występujących w bazie, a następnie zastąpiono jego zawartość odpowiadającym jej numerem linii w tak utworzonej liście.

Wyjątkiem są pola takie jak czas trwania reklamy/programu lub numer bloku reklamowego, które już zawierają wartości liczbowe, a także pole opisujące typologię programu, które już w początkowej wersji bazy *MS* było zanonimizowane.

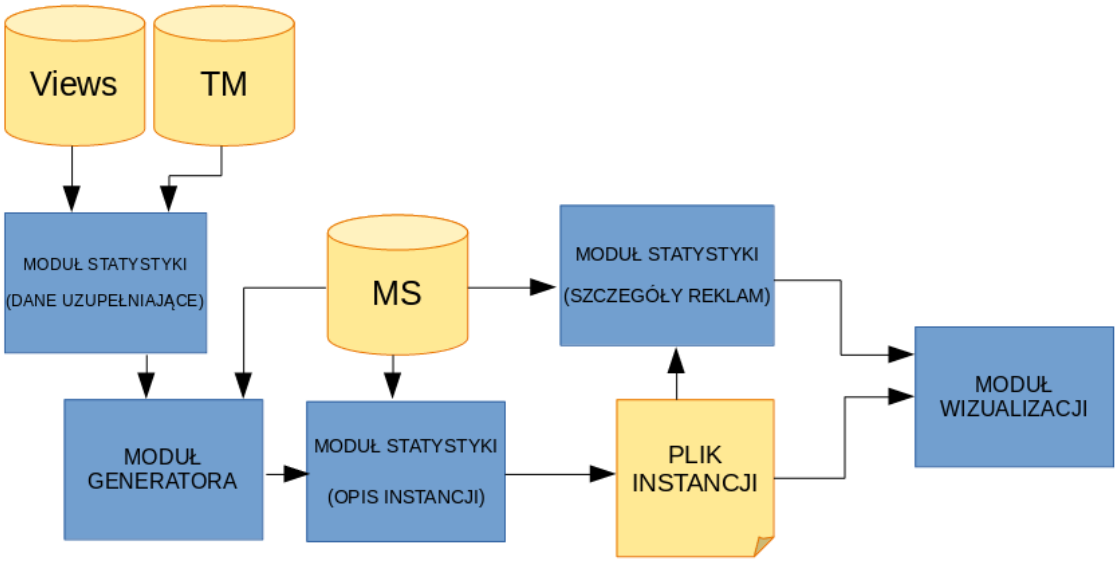
3.2.5 Dodatkowe zbiory danych

Dane zawarte w bazie *TM* podczas generowania instancji testowej muszą zostać przypisane do odpowiednich wpisów bazy *MS*. Instancja testowa zawiera między innymi informację o oglądalności danej reklamy, jak i całej przerwy reklamowej. W celu ułatwienia przypisania oglądalności do danej jednostki czasu na danym kanale, został utworzony plik csv *Views*, zawierający sumę wag wszystkich osób oglądających dany kanał podczas danego punktu pomiarowego. Punkty pomiarowe występują co minutę, dlatego w pliku *Views* znajdują się wpisy dla każdej kolejnej minuty w 2014 roku, zawierające wartości reprezentujące oglądalność 135 kanałów telewizyjnych w danym momencie. Plik ten jest wykorzystywany do szybszego przypisania oglądalności do konkretnych reklam i przerw reklamowych, gdy użytkownik systemu nie wymaga uwzględnienia oglądalności ograniczonej grupy osób, poprzez użycie filtrów demograficznych, a więc dla ustawień domyślnych.

Wygenerowane zostały także pliki zawierające listy rozróżnialnych wartości pól zbiorów danych *MS* i *TM*, które zostały wykorzystane do przydzielenia numeracji tym wartościom w ramach kodowania, a także do opisów dostępnych wartości filtrów w interfejsie użytkownika.

3.3 Przepływ danych

Wszystkie dane wyjściowe pochodzą z baz *Telemetry TM* i *Merged Schedule MS*. Baza *Views* zawiera przekształcone dane z bazy *TM*. Baza *TM* i *Views* są źródłem danych wejściowych tylko dla podmodułu danych uzupełniających modułu statystyk i służą jedynie generowaniu oglądalności do pliku instancji. Moduł generatora zapisuje instancję, którą uzupełnia podmoduł opisu instancji modułu statystycznego i finalizuje generowanie pliku instancji. Z pliku instancji korzysta moduł wizualizacji. Poza tym z pliku instancji i bazy *MS* korzysta podmoduł szczegółów reklam generując dane które wizualizuje moduł wizualizacji.



Rysunek 3.1: Przepływ danych między modułami

3.4 Moduł statystyk

Moduł statystyk można podzielić na trzy podmoduły funkcjonalne:

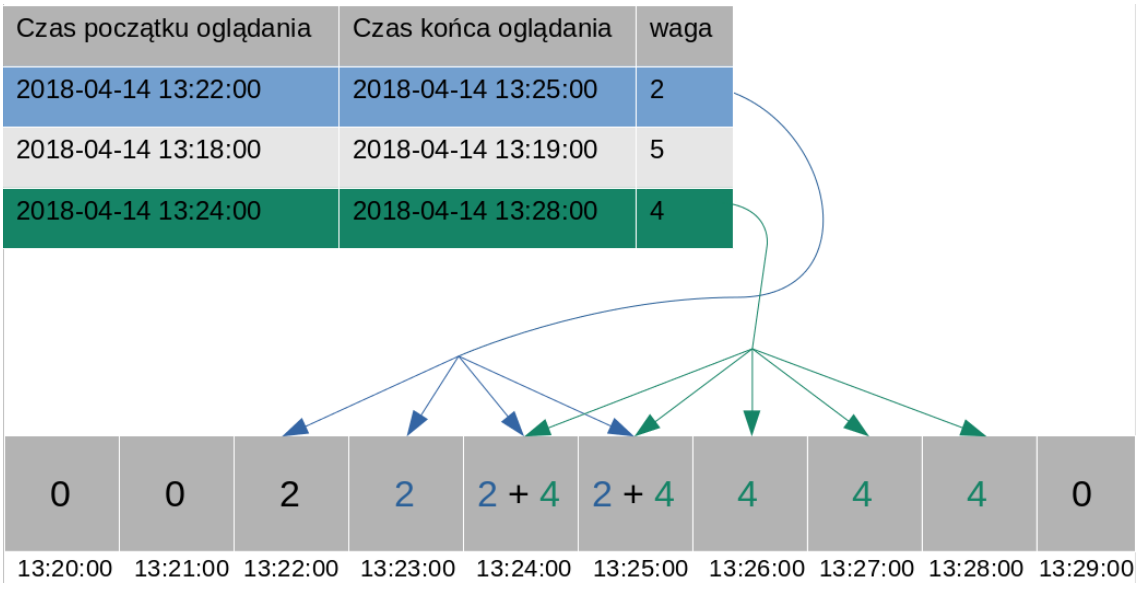
- podmoduł danych uzupełniających
- podmoduł opisu instancji
- podmoduł szczegółów reklam

Podmoduł danych uzupełniających generuje informacje o oglądalnościach w czasie i oblicza na podstawie tej informacji statystyki oglądalności poszczególnych pozycji ramówki. Dla wyliczenia podstawowych danych o oglądalności takich jak minimalna, maksymalna, średnia i odchylenie standardowe oglądalności dla zadanego okresu czasu wykorzystano bazę *TM* i wywodzącą się z niej *Views*.

3.4.1 Wektor oglądalności

Wektor oglądalności określa się dla zadanego kanału i przedziału czasu. Kolejne elementy tego wektora określają oglądalności pojedynczego kanału w kolejnych pełnych minutach. Pojedynczy element wektora jest określony przez sumę wag widzów którzy w minucie odpowiadającej temu elementowi oglądali dany kanał. Widzowie brani pod uwagę przy wyznaczaniu wektora mogą być wybrani poprzez zastosowanie filtrów nałożonych na cechy demograficzne widzów występujące w bazie *TM*.

Na rysunku 3.2 przedstawiona jest graficznie idea generowania wektora oglądalności. Przedstawiona tabelka reprezentuje uproszczoną bazę *TM*. Poniżej znajduje się przykładowy wektor oglądalności dla zakresu od 14.04.2018 13:20:00 do 14.04.2018 13:29:00. Każdej minucie odpowiada jeden element wektora obliczony poprzez zsumowanie wag widzów którzy w danej minucie oglądali analizowany kanał. Dodatkowo brane są pod uwagę cechy demograficzne widzów których dla uproszczenia nie zaznaczono na rysunku.



Rysunek 3.2: Przykład konstrukcji wektora oglądalności

Omawiany wektor jest tworzony dla całego analizowanego w instancji przedziału czasu. Umożliwia on wielokrotne i szybkie generowanie statystyk oglądalności dla mniejszych przedziałów czasu np. pojedynczych programów lub przerw reklamowych zawierających się w zadanej instancji. Jeden wektor dotyczy konkretnej grupy demograficznej i konkretnego kanału. Czas potrzebny na wygenerowanie pojedynczego wektora może zostać skrócony przez wygenerowanie wektorów dla

wszystkich kanałów z całego dostępnego czasu i wszystkich grup demograficznych i wyeksportowanie ich do osobnej bazy *Views* w celu późniejszego wczytania potrzebnych zakresów wektorów. Dla każdego kanału w instancji tworzony jest jeden wektor dla nieodfiltrowanych grup demograficznych i opcjonalnie jeden dla odfiltrowanych grup demograficznych. Ten pierwszy wektor może być wczytany z pliku *Views*, natomiast drugi musi zostać wygenerowany kiedy użytkownik wybierze jakiekolwiek filtry demograficzne.

3.4.2 Obliczanie oglądalności na podstawie wektora

Czas startu i końca oglądania danego kanału przez danego widza zapisanych w bazie *TM* jest zaokrąglony do pełnych minut. W praktyce oznacza to, że oglądalności są przybliżone. Analizowany przedział czasu dla którego wyznaczane są statystyki dotyczy pojedynczych programów lub przerw reklamowych i określony jest z dokładnością sekundową. Z tego powodu stosuje się przybliżoną metodę obliczania statystyk. Każdy jeden element wektora oglądalności, w takim przypadku przybliża oglądalność w swoim otoczeniu w dziedzinie czasu. Czas odpowiadający elementowi wektora oglądalności określa się mianem punktu pomiarowego. Dla danego punktu pomiarowego wartość a określa mniejszą z odległości punktu pomiarowego od początku i końca analizowanego przedziału czasu, gdy punkt znajduje się poza analizowanym przedziałem czasu. Analogicznie wartość b określa mniejszą z odległości punktu pomiarowego od początku i końca analizowanego przedziału czasu, gdy punkt zawiera się w analizowanym przedziale czasu. Odległości te są określone w sekundach. Na rysunku 3.3 przedstawiono punkty pomiarowe, powyższe odległości i okno czasowe odpowiadające wyznaczonemu przedziałowi.

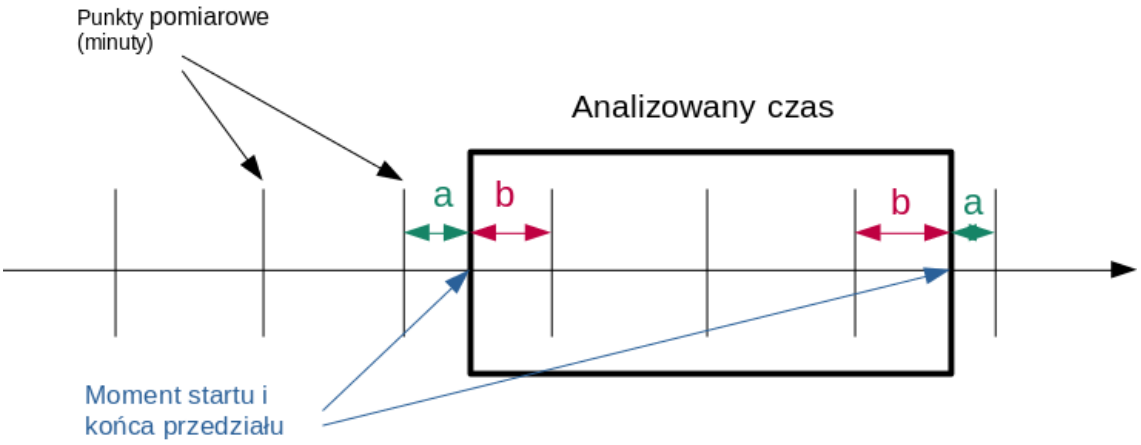
Początkowo przy obliczaniu wartości średniej i odchylenia standardowego dla pewnego przedziału czasu określano wagi wartości kolejnych punktów pomiarowych znajdujących się w badanym przedziale oraz sąsiadujących z nim według reguły:

1. jeżeli punkt pomiarowy zawiera się w analizowanym przedziale czasu i $b > 60$ to waga wartości wektora oglądalności w tym punkcie wynosi 120
2. jeżeli punkt pomiarowy zawiera się w analizowanym przedziale czasu i $b \leq 60$ to waga wartości wektora oglądalności w tym punkcie wynosi min(szerokość_analizowanego_przedziału,60+b)
3. jeżeli punkt pomiarowy jest poza analizowanym przedziałem czasu i $a \leq 60$ to waga wartości wektora oglądalności w tym punkcie wynosi $60 - a$
4. jeżeli punkt pomiarowy jest poza analizowanym przedziałem czasu i $a > 60$ to waga wartości wektora oglądalności w tym punkcie wynosi 0

Jest to podejście w którym waga punktu rośnie im większa część przedziału który reprezentuje dany punkt (w tym przypadku przedział to 120 sekund) znajduje się w badanym przedziale.

Dane podejście jest ryzykowne, ponieważ łatwo przy jego analizie i implementacji popełnić błąd, który będzie później trudny do wykrycia, ze względu na rozmiar danych i mnogość przypadków.

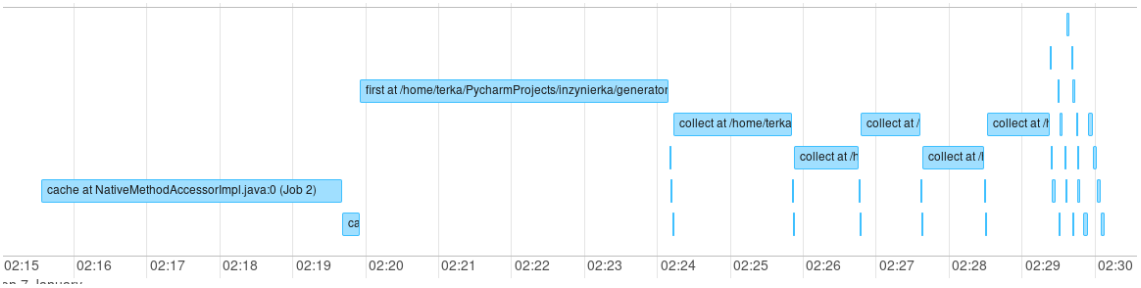
Z tego względu zastosowano podejście prostsze. Do obliczeń brane są pod uwagę wszystkie punkty pomiarowe które zawierają się w zadanym przedziale czasu, jak i te dla których $a \leq 60$. W ten sposób każdy przedział korzysta z przynajmniej dwóch wartości wektora oglądalności w punktach pomiarowych. Wartości wektora oglądalności w tych punktach przyjmują tę samą wagę i służą wyliczeniu wartości minimalnej, maksymalnej, średniej i odchylenia standardowego oglądalności w analizowanym przedziale czasu.



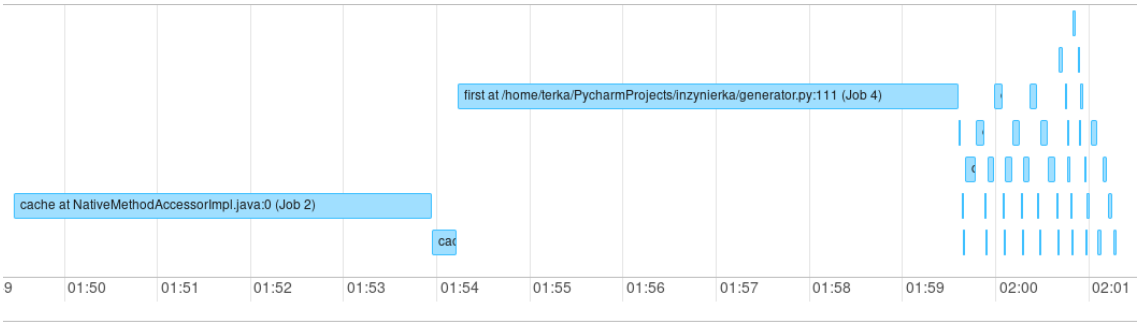
Rysunek 3.3: Wykorzystanie danych z wektora oglądalności

3.4.3 Wpływ importu wektora oglądalności na wydajność

W związku z dużą ilością danych na których pracuje program, wydajność jest bardzo istotna z punktu widzenia użytkownika. Używane narzędzie do zarządzania bazami danych Apache Spark udostępnia interfejs do podglądania przebiegu pracy na zbiorze danych i identyfikacji wąskich gardeł w zapytaniach. Dzięki temu można bez problemu przeanalizować wydajność poszczególnych operacji. Narzędzie pozwala na generowanie wykresów Gantt'a, prezentujących rozkład operacji bazodanowych w czasie. Wykorzystano tę możliwość do porównania wydajności głównej funkcjonalności aplikacji, czyli generowania instancji. Generowanie wektora oglądalności wiąże się z przeszukaniem całej bazy *TM* w poszukiwaniu wpisów z zadanego okresu czasu i zadanych kanałów. Wektor oglądalności wygenerowany bez filtrów demograficznych został wyeksportowany do bazy *Views*. Umożliwia to szybsze wczytanie z pliku gotowego wektora bez przeglądania całej bazy *TM*. Dla każdego kanału wektor wczytywany/generowany jest osobno. Dla porównania wydajności ładowania wektora z bazy *Views* i generowania z bazy *TM*, wygenerowano instancję dla pięciu kanałów, dziesięciu dni i oglądalnościach kolejno ładowanych z bazy *Views* i generowanych z bazy *TM*. Warto zwrócić uwagę na rysunku 3.4 na pięć ostatnich dłuższych bloków. Te bloki dotyczą przeglądania bazy *TM* dla pięciu kolejnych kanałów. Dla porównania na rysunku 3.5 operacje te są zastąpione operacjami wczytującymi gotowy wektor oglądalności z osobnej bazy. Różnica pomiędzy czasami wynosi 2min 35sec. Z tej różnicy można w przybliżeniu wnioskować, że wczytanie gotowych wektorów oglądalności skraca czas generowania jednego kanału do instancji o $\frac{2minuty35sekund}{5} = 43$ sekundy



Rysunek 3.4: Wykres Gantt'a (14 minut 25 sekund - wygenerowany wektor oglądalności)



Rysunek 3.5: Wykres Gantt'a (11 minut 50 sekund - zaimportowany wektor oglądalności)

3.4.4 Opis instancji

Podmoduł opisu instancji analizuje plik generowany w module generatora zawierający fragment ramówki telewizyjnej. Korzystając z danych z generatora dopisuje na końcu pliku instancji następujące statystyki:

- liczbę reklam i programów
- sumaryczną długość reklam i programów
- procentowy udział reklam i programów w czasie antenowym
- minimalną, maksymalną, średnią i medianę długości reklam i programów (w całej instancji), a także ich odchylenie standardowe
- unikalne długości reklamy w postaci listy możliwych długości
- procentowy udział typów reklam w czasie antenowym dla danych z instancji

Do pliku instancji razem z innymi statystykami zapisywane są macierze sąsiadowania. Są to struktury opisujące wzajemne położenie reklam w ramówce wyświetlane w formie macierzy. Generowane są cztery takie struktury w dwóch typach i względem dwóch cech. Cechami są typ reklamy i reklamodawca. Pierwszy typ takiej struktury opisuje dla każdej pary cech reklam liczbę ich wystąpień bezpośrednio po sobie w analizowanym czasie antenowym. Drugi typ generuje dla każdej pary cech reklam liczbę emisji par reklam z tymi cechami w tej samej przerwie reklamowej. Dane cztery struktury to:

- macierz typu pierwszego według cechy - typ reklamy
- macierz typu pierwszego według cechy - reklamodawca
- macierz typu drugiego według cechy - typ reklamy
- macierz typu drugiego według cechy - reklamodawca

3.4.5 Statystyki reklam na żądanie

Generowaniem statystyk reklam na żądanie zajmuje się podmoduł szczegółów reklam umożliwiający analizę konkretnych reklam, reklamodawców i typów reklam. Użytkownik wybiera plik instancji na podstawie którego chce dokonać analizy. W module statystyk reklam na podstawie zakresu czasu i kanałów zapisanych w pliku instancji dane pobierane są z bazy *MS* i analogicznie

jak przy generowaniu instancji zapisywane w widoku *mstemp*. Ramówka wczytana do widoku *mstemp* jest filtrowana, w zależności od tego czego ma dotyczyć analiza. Filtrowanie odbywa się odpowiednio po cechach reklam takich jak:

- identyfikator unikalnej reklamy
- nazwa reklamodawcy
- typ reklamy

W zależności od wybranej cechy dostępne są różne funkcje analizujące. Przefiltrowana ramówka trafia do odpowiednio wybranej w interfejsie graficznym funkcji analizującej. Wyjątkiem dla którego funkcja analizująca nie korzysta z przefiltrowanego widoku *mstemp* jest rozkład emisji danej reklamy na wszystkich kanałach w całym przedziale dostępnych danych historycznych i w funkcji odfiltrowującej wartości konkretnych wierszy z macierzy sąsiadowania, która korzysta z danych zapisanych w pliku instancji. Dla filtrowania po identyfikatorze unikalnej reklamy dostępne są funkcje generujące informacje o:

- maksymalnej liczbie emisji danej reklamy w przeciągu jednej przerwy
- rozkładzie emisji danej reklamy zarówno w przedziale instancji jak i w wszystkich danych historycznych
- czasie pierwszej i ostatniej emisji oraz sumarycznej liczbie emisji danej reklamy

Jeżeli użytkownik wybierze filtrowanie według reklamodawcy to może otrzymać informacje o:

- liczbie reklam z różnymi podmarkami danego reklamodawcy
- czasie pierwszej i ostatniej emisji oraz sumarycznej liczbie emisji reklam danego reklamodawcy

W przypadku kiedy użytkownik zaznaczy filtrowanie według typu reklamy to może użyć funkcji generujących:

- rozkład godzinowy emisji danego typu reklamy
- odfiltrowaną macierz sąsiadowania

3.5 Moduł generatora instancji

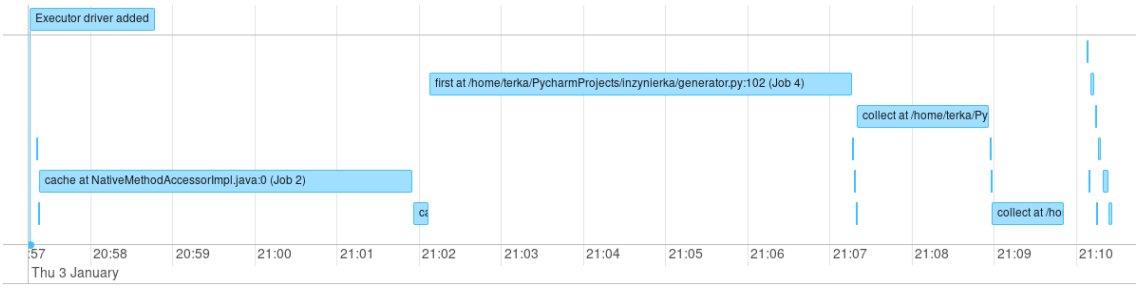
Użytkownik chcąc wygenerować instancję podaje w graficznym interfejsie zakres czasu, listę kanałów i filtry demograficzne. Pierwsze dwa, czyli zakres czasu i lista kanałów, trafiają do zapytania tworzącego bazodanowy widok tymczasowy o nazwie *mstemp* z wybranym fragmentem ramówki. Dla każdego kanału wczytywane lub generowane są wektory oglądalności poprzez moduł statystyk. Generator instancji formatuje wybrany zakres ramówki. Dla każdego wpisu dotyczącego reklamy zapisuje jej czas trwania, typ programu następującego po reklamie, numer bloku reklamowego, typ reklamy, koszt, umiejscowienie spotu w ramówce, identyfikator reklamy i identyfikator reklamodawcy. Dla programu zapisuje informacje o czasie trwania i typie programu. Z danych otrzymanych z modułu statystyk dodaje informacje o średniej, minimalnej, maksymalnej i odchyleniu standardowym oglądalności dla całego zakresu widzów, jak i dla widzów ograniczonych filtrami demograficznymi. Wpisy dotyczące autopromocji zawierają jedynie czas trwania.

Po opisaniu ramówki jednostkami programów, reklam i autopromocji generowane są wpisy dotyczące przerw reklamowych które zawierają informacje o czasie trwania przerwy, numerze bloku reklamowego, średniej, minimalnej, maksymalnej i odchyleniu standardowym długości pojedynczych reklam w bloku, liczbie reklam w bloku, średnim, minimalnym, maksymalnym i odchyleniu standardowym kosztu pojedynczych reklam w bloku i o średniej, minimalnej, maksymalnej i odchyleniu standardowym oglądalności dla całego zakresu widzów jak i dla widzów ograniczonych filtrami demograficznymi. Kolejny krok to analiza i dołączenie do tak wygenerowanego pliku dodatkowych statystyk przez podmoduł opisu instancji modułu statystyk.

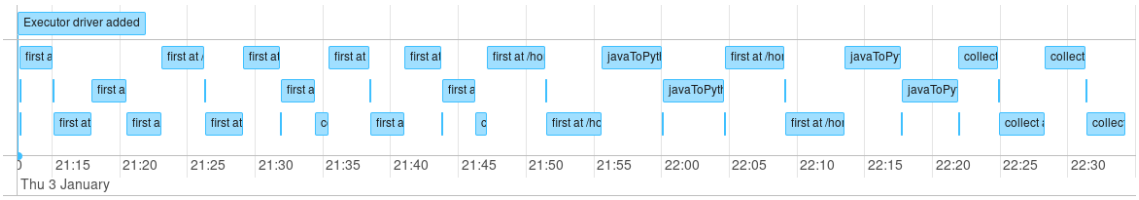
3.5.1 Wpływ widoku zmaterializowanego na wydajność generatora instancji

Widok zmaterializowany to wynik zapytania bazodanowego zapisany w pamięci komputera, na którym można wykonywać kolejne zapytania bazodanowe. Do wygenerowania większości statystyk zawartych w pliku instancji wystarcza ramówka telewizyjna (*MS*) ograniczona czasem początku i końca przedziału instancji, oraz zbiorem badanych kanałów. Domyślnie widok *mstemp* nie był zmaterializowany, co oznacza, że przy każdym odwołaniu do niego następowało odwołanie do bazy *MS*. W tym przypadku zwielokrotniony był ten sam krok filtrowania. Wykorzystując możliwości Apache Spark, zapisano wynik pośredni wielu zapytań w postaci widoku zmaterializowanego pod tą samą nazwą *mstemp* w pamięci operacyjnej komputera do ponownego wykorzystania przez kolejne funkcje analizujące. Pozwoliło to znacznie skrócić czas generowania pliku instancji.

Na poniższym porównaniu generowana jest instancja dla dwóch kanałów, czterech dni i oglądalnościach liczonych dla zbioru widzów ograniczonego do kobiet. Pierwszy wykres na rysunku 3.6 pokazuje 4 dłuższe zadania, pierwsze dwa z nich dotyczą przeszukiwania bazy *MS* i zapisywania jej do pamięci podręcznej, a dwa kolejne odnoszą się do bazy *TM* w celu wygenerowania oglądalności. Większa część pozostałych bloków dotyczy zapytań odnoszących się do widoku zmaterializowanego w pamięci operacyjnej. Na drugim rysunku 3.7 przedstawiono wykres Gantt'a dla generowania tej samej instancji, bez użycia widoku zmaterializowanego. Wyraźnie widać, że jest więcej zadań trwających około 5 minut. Wynika to z tego, że z każdym zapytaniem odnoszącym się do bazy *MS* przeszukiwana jest cała baza dla odfiltrowania zakresu instancji. Rozwiązanie korzystające z widoku zmaterializowanego jest szybsze ponad 6 razy.



Rysunek 3.6: Wykres Gantt'a (13 minut 11 sekund - używa widoku zmaterializowanego)



Rysunek 3.7: Wykres Gantt'a (1 godzina 21 minut 47 sekund - nie używa widoku zmaterializowanego)

3.6 Moduł wizualizacji

Moduł wizualizacji został podzielony na dwa podmoduły, każdy połączony z jednym, odpowiadającym mu widokiem. Taka forma struktury została wybrana w głównej mierze dla wygody i pogrupowania - funkcje potrzebne do wizualizacji różniły się między sobą w zależności od widoku. Różnice były zarówno w samej strukturze funkcji jak i przyjmowanych danych.

3.6.1 Podmoduł wizualizacji statystyk instancji

Można wyróżnić dwie części podmodułu wizualizacji statystyk instancji, część odpowiedzialną za wygląd i działanie interfejsu graficznego, oraz podłączoną do niej poprzez wywołania część funkcjonalną wizualizacji. Odpowiada ona za pobranie, wizualizowanie i wydobywanie potrzebnych danych. Źródłem danych jest plik instancji, który zostaje wczytany i przetworzony. Ten etap jest sztywno dostosowany do formatu pliku instancji, a co za tym idzie wszelkie zmiany formatu będą powodowały konieczność zmiany kodu. Każdy kanał telewizyjny zawarty w instancji posiada określoną i identyczną liczbę przypisanych wykresów. Wykresy są generowane kolejno dla każdego kanału i zapisywane do pliku PDF. Zależnie od wyborów w interfejsie moduł ten może dodatkowo wygenerować plik HTML z wykresem Gantt'a prezentującym program telewizyjny oraz dodać macierze sąsiadowania. Wygląd interfejsu i znaczenie jego elementów oraz efektów działania wizualizacji opisano w dalszych rozdziałach.

3.6.2 Podmoduł wizualizacji statystyk szczegółowych

Podmoduł wizualizacji statystyk szczegółowych również może zostać podzielony na część odpowiadającą za interfejs i funkcjonalność. Wymaga również pliku instancji, jednak wyczytuje z niego jedynie ogólne dane - wybrane kanały oraz przedział czasowy. Na podstawie tych danych i wyborów w interfejsie korzysta z funkcji zawartych w module generatora w celu utworzenia ramy danych *(ang. dataframe)* i pobrania z niej statystyk. Następnie wydobyte statystyki są wizualizowane w odpowiadającej im formie. Każda statystyka występująca w tym module i obrazowana wykresem jest generowana do tymczasowego pliku PDF. Wygenerowanie kolejnej nadpisuje poprzednią, w celu zachowania konkretnego wykresu należy użyć opcji *"Zapisz jako..."* programu, w którym plik został otwarty. Znaczenie elementów interfejsu oraz efekty wizualizacji zostały opisane w dalszych rozdziałach.

Rozdział 4

Wykorzystane narzędzia

Jedną z najważniejszych decyzji architektonicznych był dobór narzędzi programistycznych. Głównym czynnikiem, na który zwrócono uwagę była konieczność działania na dużych zbiorach danych. Potrzebne było więc zastosowanie narzędzia, odpowiedniego do tego typu problemów.

Zdecydowano się, na wybór **Apache Spark**, ze względu na to, że umożliwia wykonywanie obliczeń rozproszonych, a także udostępnia szereg narzędzi do analizy i przetwarzania danych. Apache Spark to silnik umożliwiający przetwarzanie danych w pamięci. Silnik ten udostępnia zaawansowane API w językach programowania Python, R, Java i Scala. Używając tego silnika możliwe jest przyspieszenie wykonania operacji na danych. Dodatkową zaletą jest redukcja zapisów i odczytów danych z dysku.

DataFrame w Apache Spark to rozproszona kolekcja danych zorganizowana w kolumny o określonych nazwach i typach. Ma ona format tabelaryczny z dodatkowymi metadanymi, które umożliwiają optymalizację zapytań [9]. DataFrame to struktura zoptymalizowana do wykonywania operacji równoległych. Kolekcję danych można stworzyć z różnych typów danych: bazy danych, plików tekstowych lub istniejących RDD.

Ta funkcjonalność umożliwiła wczytanie dostarczonych do sytemu danych zapisanych w plikach tekstowych typu csv. Danym zorganizowanym w kolumny przypisano odpowiednie typy.

RDD (Resilient Distributed Datasets) to fundamentalna struktura danych w Apache Spark. Jest to trwała rozproszona kolekcja obiektów. Zbiór rekordów podzielonych na partycje. RDD jest używane głównie w przypadkach, gdy dane, na których wykonuje się operacje nie są ustrukturyzowane, lub gdy nie ma znaczenia ich typ.

Jako, że w przypadku pracy inżynierskiej dane są podzielone na kolumny, którym można nadać typ, RDD nie zostało wykorzystane, lecz początkowo rozważano jego użycie.

Spark SQL to narzędzie przeznaczone do wykonywania działań na ustrukturyzowanym zbiorze danych. Spark SQL umożliwia transformację zapytań SQL w zoptymalizowany plan logiczny. Następnie plan jest transformowany na plan wykonań fizycznych. W czasie egzekucji wybierany jest najlepszy plan fizyczny spośród całego planu. W Apache Spark istnieje możliwość zapisu w pamięci operacyjnej tabel z użyciem formatu kolumnowego. Dzięki temu, Spark SQL może skanować tylko wymagane kolumny i dostosować kompresję w celu zminimalizowania zużycia pamięci [10].

Platforma Spark umożliwia używanie składni języka SQL, dzięki czemu możliwe jest wykonywanie operacji na bazach danych oraz DataFrame. Podczas tworzenia oprogramowania wykorzystano szereg zapytań generujących funkcje statystyczne. Każdorazowo zwracano uwagę na optymalizację zapytań i możliwe do wykonania usprawnienia zapytań.

Podjęto decyzję o tworzeniu oprogramowania w języku **Python** z wykorzystaniem interfejsu PySpark dla języka Python. Python to wysokopoziomowy język programowania o typowaniu dy-

namicznym. Ten rodzaj typowania, oznacza, że typy są przepisywane zmiennym podczas działania programu. Głównym czynnikiem przemawiającym za tym wyborem był bardzo rozbudowany interfejs Spark'a dla tego języka. Dodatkowym powodem takiej decyzji był fakt posiadania rozbudowanego pakietu bibliotek, co umożliwiło szeroki rozwój oprogramowania. Do stworzenia projektu użyto między innymi takich bibliotek jak: Matplotlib, Plotly, TkInter, NumPy.

Matplotlib jest biblioteką wydaną na licencji Open Source, najczęściej używaną do wizualizacji w projekcie. Umożliwia wizualizację danych 2D, między innymi generowanie wykresów i histogramów. Korzystając z tej biblioteki posiadamy pełną kontrolę nad dostosowywaniem wyglądu wykresów do swoich potrzeb. Istnieje możliwość edycji cech wykresów takich jak styl linii, właściwość czcionek i osi [11]. Jej kolekcja komend *matplotlib.pyplot* umożliwia wygodne tworzenie wykresów. Zostały w niej stworzone wszystkie wykresy pracy inżynierskiej z wyjątkiem wykresów Gantt'a.

Plotly jest biblioteką języka Python, która umożliwia tworzenie interaktywnych wykresów online [12]. Pozwala jednak na pominięcie sieciowego aspektu poprzez moduł *offline*, który umożliwia zapisanie wykresów do pliku HTML. Biblioteka ta użyta jest w celu stworzenia ramówki w formie wykresu Gantt'a.

Jednym z wymagań postawionych przed projektem było przygotowanie interfejsu do obsługi funkcjonalności. Do zrealizowania tego celu wykorzystano bibliotekę o nazwie **TkInter** służącą do tworzenia graficznego interfejsu użytkownika. Jest to zorientowana obiektowo technologia dostępna na platformie Windows i Unix [13].

W trakcie realizacji projektu użyto także biblioteki **Numpy**, która jest podstawowym pakietem wykorzystywanym do obliczeń z wykorzystaniem Pythona [14]. Zawiera w sobie funkcje zoptymalizowane do obliczeń statystycznych. Dodatkowo, jest to biblioteka silnie nastawiona na wykonywanie obliczeń wektorowych i macierzowych, co jest pożądaną cechą w przypadku obróbki masywnych danych i generacji wykresów.

Do przygotowania implementacji systemu użyto środowiska PyCharm firmy JetBrains. Daje wiele możliwości w procesie rozwoju oprogramowania. Dla zespołu najważniejsze było umożliwienie tworzenia projektów desktopowych, edycję i formatowanie kodu. Ponadto posiada graficzny debugger oraz podświetlanie składni.

Rozdział 5

Dokumentacja użytkownika

W ramach dokumentacji użytkownika przedstawiono części aplikacji, istotne z punktu widzenia osób korzystających z systemu. W szczególności w niniejszym rozdziale omówiono elementy tj.:

- interfejs graficzny
- pliki PDF zawierające wykresy
- pliki PDF zawierające macierze
- pliki HTML zawierające program telewizyjny

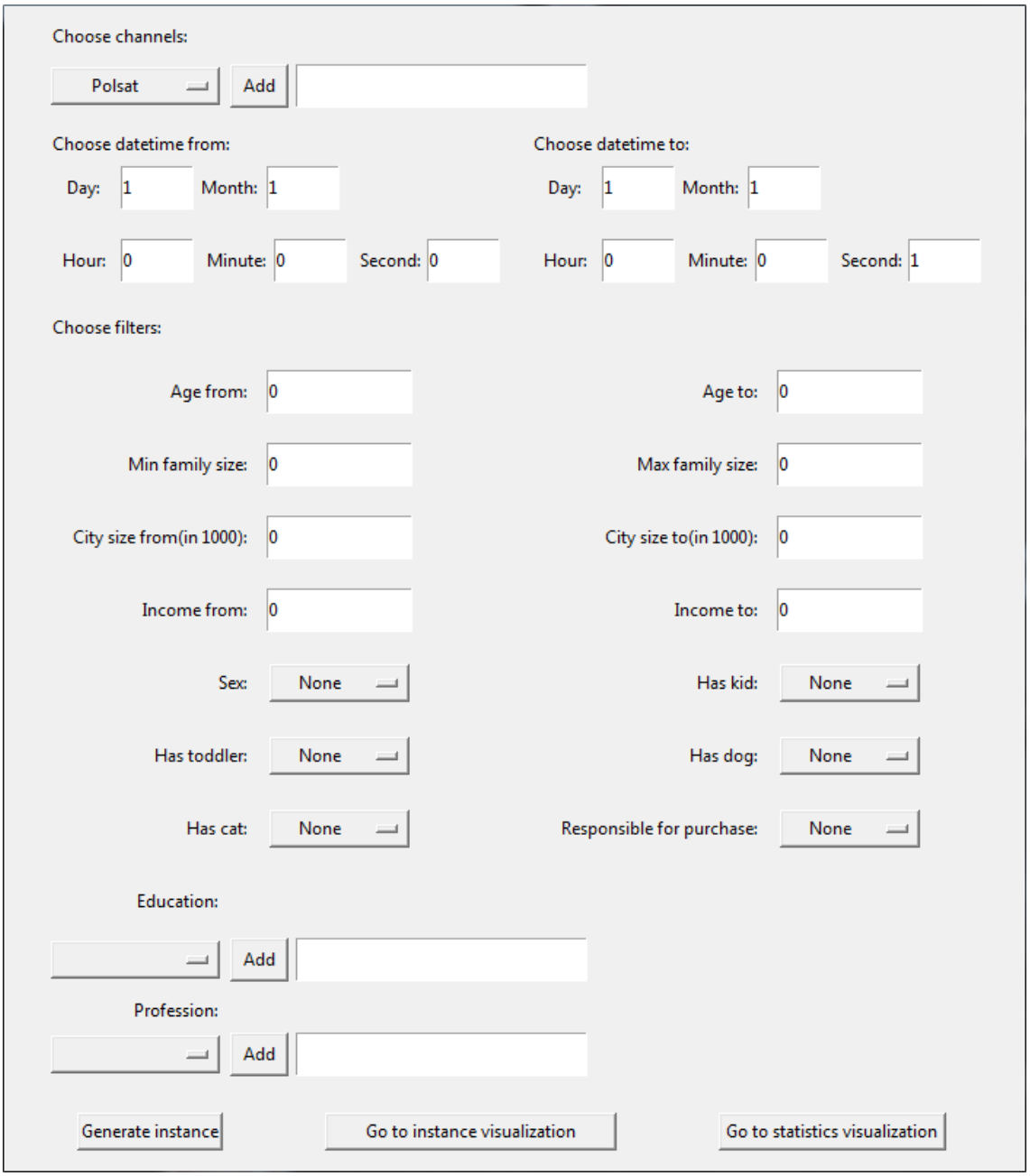
Każdy możliwy rodzaj efektu wizualizacji, jak i cały dostępny interfejs zostaną szczegółowo opisane w poniższych podrozdziałach.

5.1 Interfejs graficzny

Interfejs graficzny dostępny dla użytkownika składa się z trzech różnych widoków, każdy przypisany do odrębnej funkcjonalności. Wszystkie zawierają się w jednym oknie. Na potrzeby monitorów niższej rozdzielczości został zaimplementowany pasek przewijania do pionowej nawigacji. Dla monitorów o większej rozdzielczości istnieje możliwość rozciągnięcia okna tak, aby wszystkie elementy były widoczne.

5.1.1 Widok generatora

Przedstawiony na Rys. 5.1 interfejs generatora pozwala na konfigurację pod kątem danych zapisywanych do instancji.

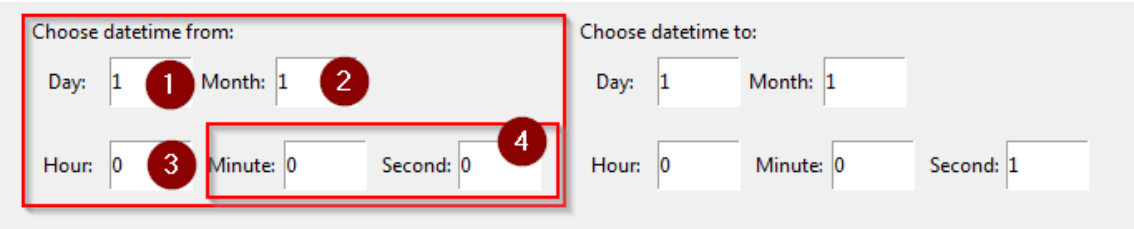


Rysunek 5.1: Pełen widok interfejsu generatora

Wybór kanałów, które ma zawierać instancja odbywa się poprzez połączenie trzech elementów interfejsu zaprezentowanych na Rys. 5.2. Menu rozwijane *(1)* pozwala zobaczyć pełną listę dostępnych kanałów oraz wybór konkretnego. Przycisk "Add" *(2)* doda aktualny kanał z listy do wybranych. Pole edycji tekstu *(3)* pozwala na zobaczenie aktualnie wybranych kanałów, które są reprezentowane poprzez ciąg nazw rozdzielonych średnikami. Pozwala to także na usunięcie uprzednio wybranego kanału lub dopisanie ręcznie, bez konieczności użycia listy.

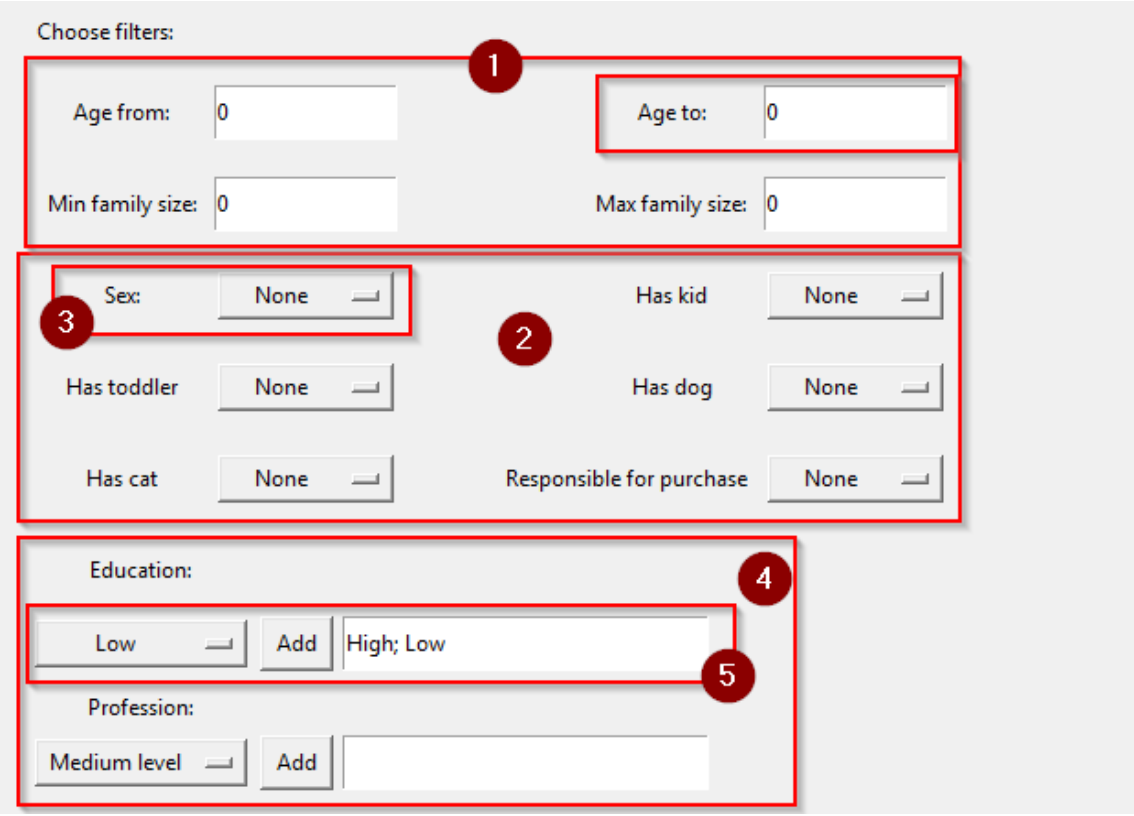
Rysunek 5.2: Wybór kanałów

Wybór daty początku, analogicznie końca, zakresu odbywa się poprzez pięć pól tekstowych widocznych na Rys. 5.3. Każde pole posiada wbudowane funkcje sprawdzające wprowadzane dane. Pole dnia *(1)* pozwala jedynie na wprowadzenie liczb z zakresu od 1 do maksymalnej liczby dni miesiąca. Pole miesiąca *(2)* pozwala na wprowadzenie liczb od 1 do 12 oraz definiuję maksymalną wartość dla pola dni. Pole godzin *(3)* zezwala jedynie na zakres od 0 do 23, a pola minut i sekund *(4)* od 0 do 59.



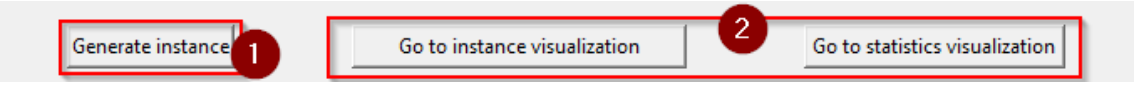
Rysunek 5.3: Wybór zakresu dat

Wybór filtrów ograniczających zbiór reklam poddawanych analizie w ramach bloku(Rys. 5.4) jest opcjonalny i może zostać podzielony na trzy sekcje. Każda sekcja różni się metodą wprowadzania i rodzajem przekazywanej wartości. Sekcja pierwsza *(1)* umożliwia wybór zakresów: wieku oraz rozmiaru rodziny. Wybranie górnej granicy zakresu jako 0 sprawia, że dany filtr nie zostanie dodany. Sekcja druga *(2)* odpowiada za filtry mogące przyjąć jedynie dwie wartości. Są to rozwijane listy dostępnych opcji, gdzie "None" odpowiada za niewybranie danego filtra. Filtr płci *(3)* różni się od pozostałych dostępnymi wartościami, gdyż zamiast "True"/"False"/"None" posiada "Male"/"Female"/"None". Trzecia sekcja *(4)* zawiera filtry wielowartościowe. Ich działanie opiera się na tych samych zasadach co dodawanie kanałów. Wybrane wartości mogą być modyfikowane ręcznie lub poprzez rozwijaną listę i przycisk *(5)*



Rysunek 5.4: Wybór filtrów

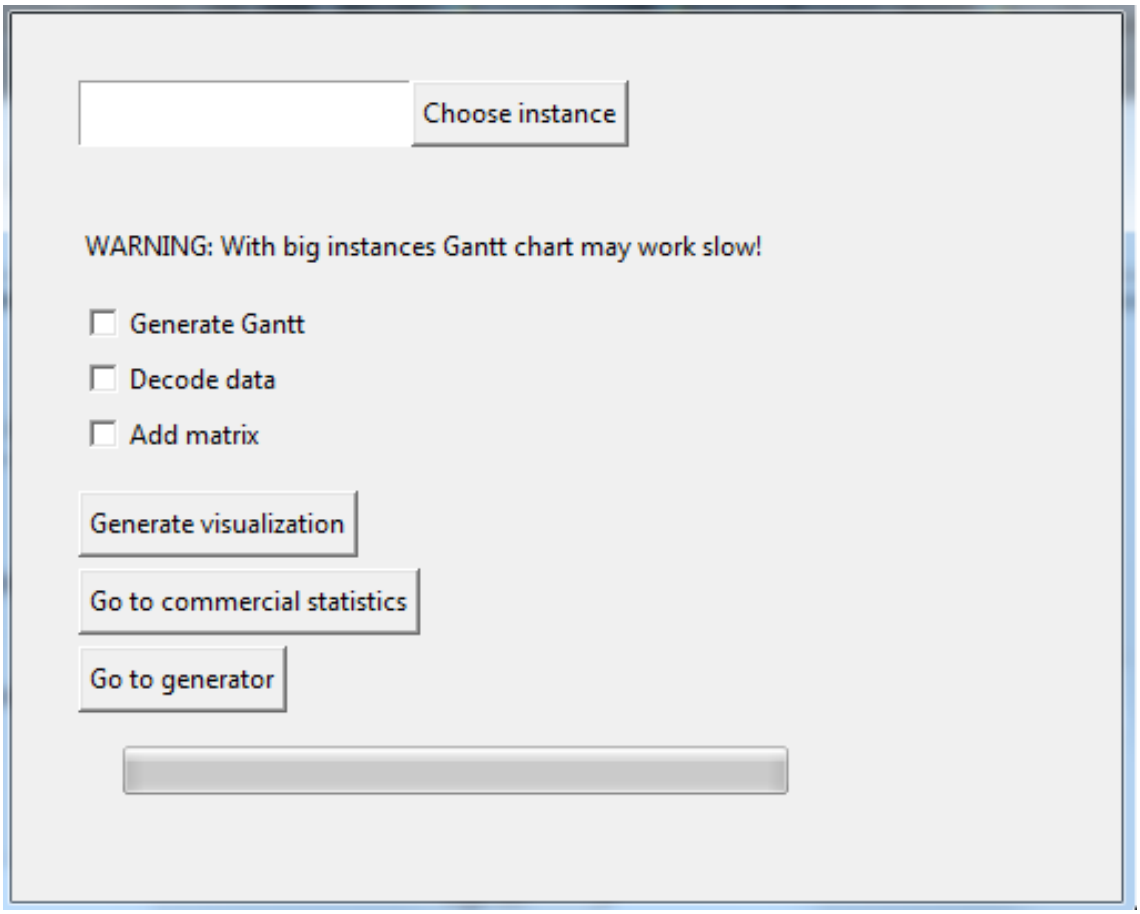
Ostatnią częścią widoku są przyciski służące do nawigacji widoczne na Rys. 5.5. Przycisk "Generate instance" *(1)* rozpocznie proces generacji zadanej instancji do pliku. Pozostałe przyciski *(2)* pozwalają na przejście do pozostałych dwóch widoków aplikacji.



Rysunek 5.5: Przyciski nawigacji

5.1.2 Widok wizualizacji instancji

Widok interfejsu przedstawiony na Rys. 5.6 pozwala na wizualizację danych instancji z uprzednio wygenerowanego pliku.



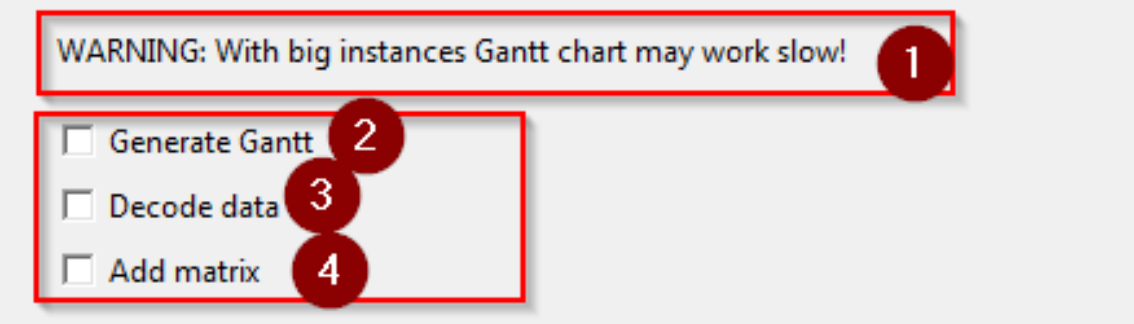
Rysunek 5.6: Pełen widok interfejsu wizualizacji instancji

Segment widoczny na Rys. 5.7, odpowiadający za wybór pliku instancji, składa się z dwóch części: pola tekstowego zawierającego ścieżkę do pliku *(1)* oraz przycisku *(2)*, którego użycie otwiera okno przeglądania plików. Wyboru można dokonać wpisując ścieżkę ręcznie w pole, lub używając wywołanego eksploratora.

Rysunek 5.7: Wybór pliku

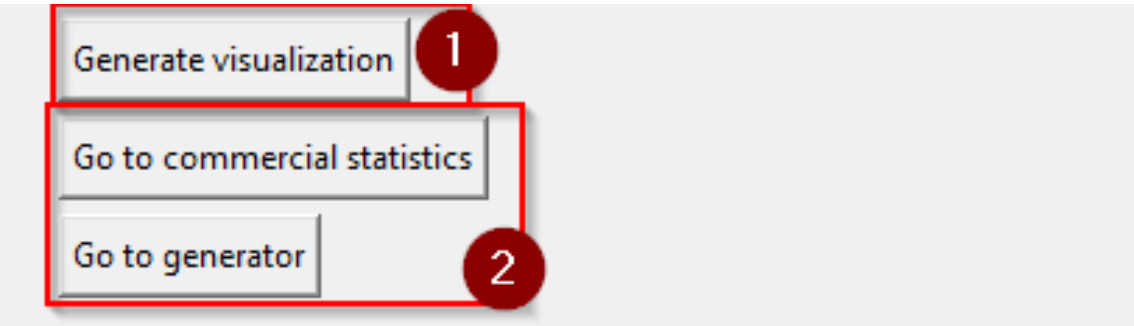
Istnieją trzy opcje wizualizacji instancji, za wybór których odpowiada element widoku widoczny na Rys. 5.8. Wykresy danych, takich jak koszt, oglądalność i czas trwania, dla kanałów i bloków reklamowych zawartych w instancji generowane są zawsze do pliku PDF. Poza nimi możliwe jest wygenerowanie ramówki w postaci wykresu Gantt'a *(2)*. Wykres ten jest generowany do odrębnego pliku HTML i automatycznie otwierany. Ze względów na liczbę danych nawet w małej instancji i kwestie techniczne, wygenerowana wizualizacja ramówki może wymagać większej ilości czasu na przetworzenie poleceń nawigacji (np. przybliżenie) lub przeglądarka nie będzie w stanie jej

otworzyć w przypadkach jeszcze większych instancji. Ostrzeżenie jest wyświetlane jako tekst nad dostępnymi opcjami *(1)* Kolejną opcją jest dekodowanie danych *(3)*. Dane w pliku są zakodowane w celu anonimizacji, wybranie tej opcji sprawi, że na wykresach elementy takie jak nazwa kanału wypisane będą jako prawdziwe nazwy. Ostatnią opcją jest wygenerowanie macierzy sąsiadowania dla sektorów reklam zawartych w instancji *(4)*. Owa macierz zostanie dodana jako ostatnia strona wygenerowanego pliku PDF.



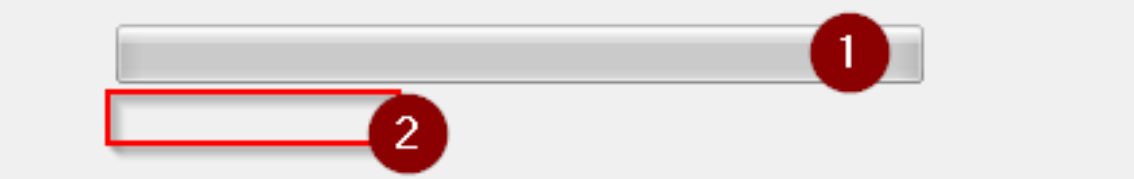
Rysunek 5.8: Opcje wizualizacji

Pierwszy z przycisków *(1)*, widocznych na Rys. 5.9, rozpoczyna proces wizualizacji. Postęp można monitorować wykorzystując następną część tego widoku. Pozostałe przyciski *(2)* pozwalają na nawigację do kolejnych widoków aplikacji.



Rysunek 5.9: Przyciski nawigacji

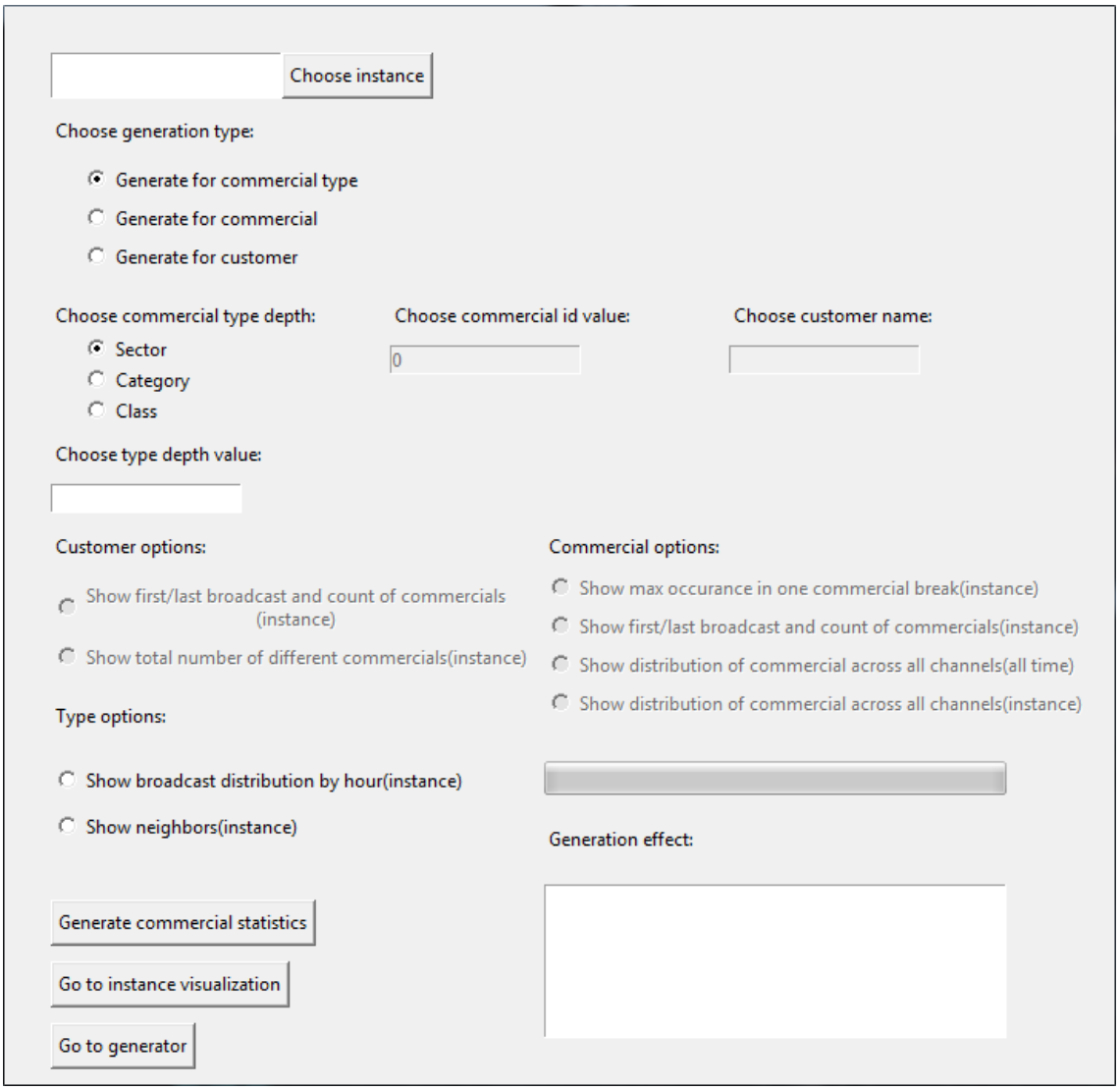
Część widoku z Rys. 5.10 odpowiada za wyświetlanie postępu procesu wizualizacji. Składa się z paska postępu *(1)* oraz tekstu opisującego aktualny etap generacji *(2)*.



Rysunek 5.10: Postęp generacji

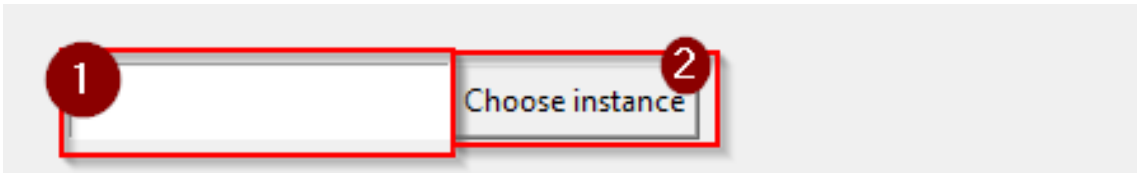
5.1.3 Widok wizualizacji statystyk

Ostatnim widokiem aplikacji, widocznym na Rys. 5.11, jest widok wizualizacji statystyk szczegółowych. Pozwala on na wyświetlenie konkretnych, szczegółowych statystyk dla odfiltrowanych z instancji danych.



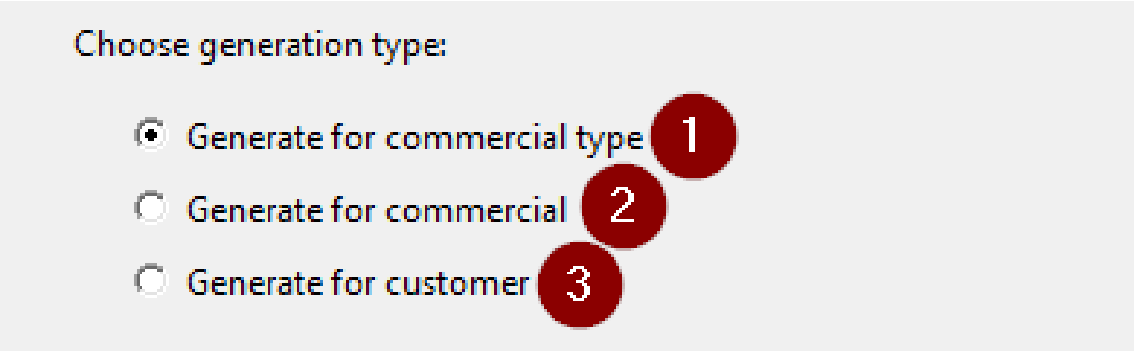
Rysunek 5.11: Pełen widok interfejsu wizualizacji statystyk

Wybór instancji odbywa się na tej samej zasadzie co w poprzednim widoku, element za to odpowiedzialny przedstawia Rys. 5.12. Nazwa pliku znajduje się w polu tekstowym *(1)*, które można wypełnić ręcznie lub poprzez eksplorator wywołany użyciem przycisku *(2)*.



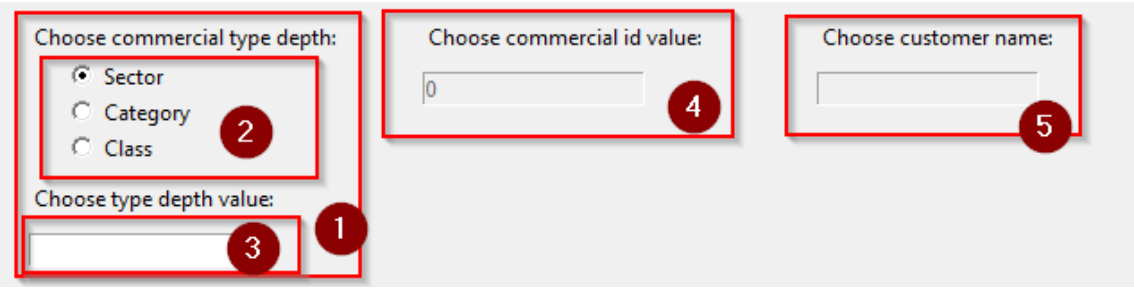
Rysunek 5.12: Wybór pliku

Dostępne są trzy kategorie wizualizacji, których wybór odbywa się poprzez elementy widoczne na Rys. 5.13, każda przyjmuje inne wartości i udostępnia inne opcje, które zostaną opisane w dalszej części pracy. Wizualizacja może wystąpić dla statystyk wynikających z wyboru konkretnego typu reklamy *(1)*, konkretnej reklamy *(2)* lub reklamodawcy *(3)*.



Rysunek 5.13: Wybór kategorii

W zależności od wyboru w poprzedniej części, należy podać różne wartości niezbędne do określenia wizualizowanego obiektu. Widok tego etapu przedstawia Rys. 5.14. Dla typu reklamy *(1)* dostępne są trzy podkategorie opisujące poziom dokładności typu *(2)*. W związku z tym, zależnie od wyboru w polu wartości *(3)* należy wpisać pierwsze dwie, cztery lub wszystkie sześć cyfr kodu typu. Dla konkretnej reklamy, do pola tekstowego *(4)* należy wpisać jej unikalny numer ID. Z kolei dla reklamodawcy, w pole tekstowe *(5)* należy wpisać jego pełną nazwę, wielkimi literami. W trakcie generowania wizualizacji zostanie ona automatycznie zakodowana.



Rysunek 5.14: Wybór wartości filtra

Każda kategoria udostępnia inne opcje wizualizacji. Wyboru konkretnej opcji dokonuje się w sekcji widoku, którą prezentuje Rys. 5.15. Dla reklamodawcy są to:

- pierwsza/ostatnia transmisja oraz liczba reklam dla danego reklamodawcy
- liczba reklam danego reklamodawcy

Dla konkretnej reklamy:

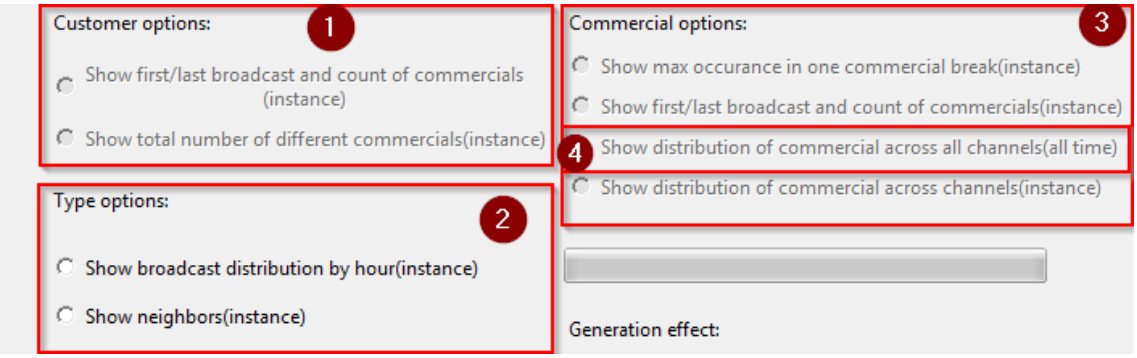
- maksymalna liczba wystąpień reklamy w bloku reklamowym
- pierwsza/ostatnia transmisja reklamy i liczba jej wystąpień
- liczba wystąpień reklamy na konkretnych kanałach telewizyjnych zawartych w instancji
- liczba wystąpień reklamy na wszystkich kanałach *(4)*

Opcja oznaczona cyfrą *(4)* jest wyróżniona, gdyż ta statystyka nie wymaga pliku instancji i jest generowana dla całości danych historycznych. Ostatnim elementem tego fragmentu widoku są opcje dla typu reklamy:

- Godzinowy rozkład reklam danego typu

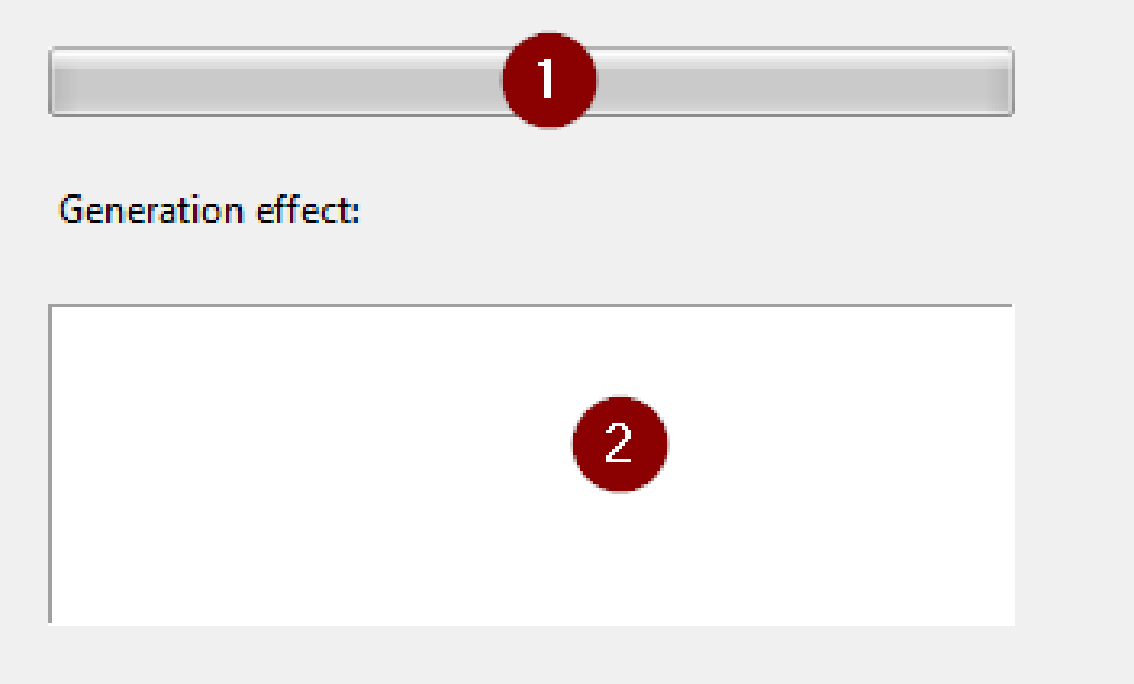
- Macierz sąsiadowania reklam w ramówce dla zadanego typu

Warto też zauważyć, że na końcu każdej opcji znajduje się informacja, czy wymaga ona podania pliku instancji. Gdy jest on wymagany w nawiasach pojawia się słowo *instance*, gdy plik nie jest potrzebny jest to wyrażenie *all time*.



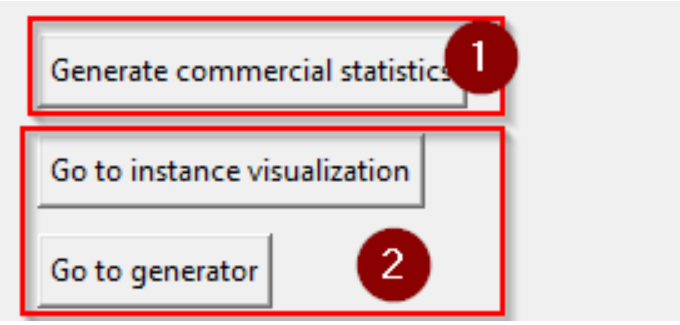
Rysunek 5.15: Wybór opcji

Kolejna część widoku, znajdująca się na Rys. 5.16, obrazuje postęp procesu. Zarówno poprzez pasek postępu *(1)*, jak i wypisywanie krótkich komunikatów o wystąpieniu kolejnych etapów. Wspomniane komunikaty pojawiają się w oznaczonym obszarze tekstowym *(2)*. Jeśli na etapie generacji wystąpi błąd, również zostanie zapisany w owym obszarze. Dla opcji, które nie zwracają wykresów, po udanej generacji, wyciągnięta statystyka zastąpi listę postępu.



Rysunek 5.16: Postęp i efekt wizualizacji

Ostatnią częścią widoku są widoczne na Rys. 5.17 przyciski nawigacji do pozostałych ekranów aplikacji *(2)* oraz przycisk rozpoczynający proces z wybranymi opcjami *(1)*.



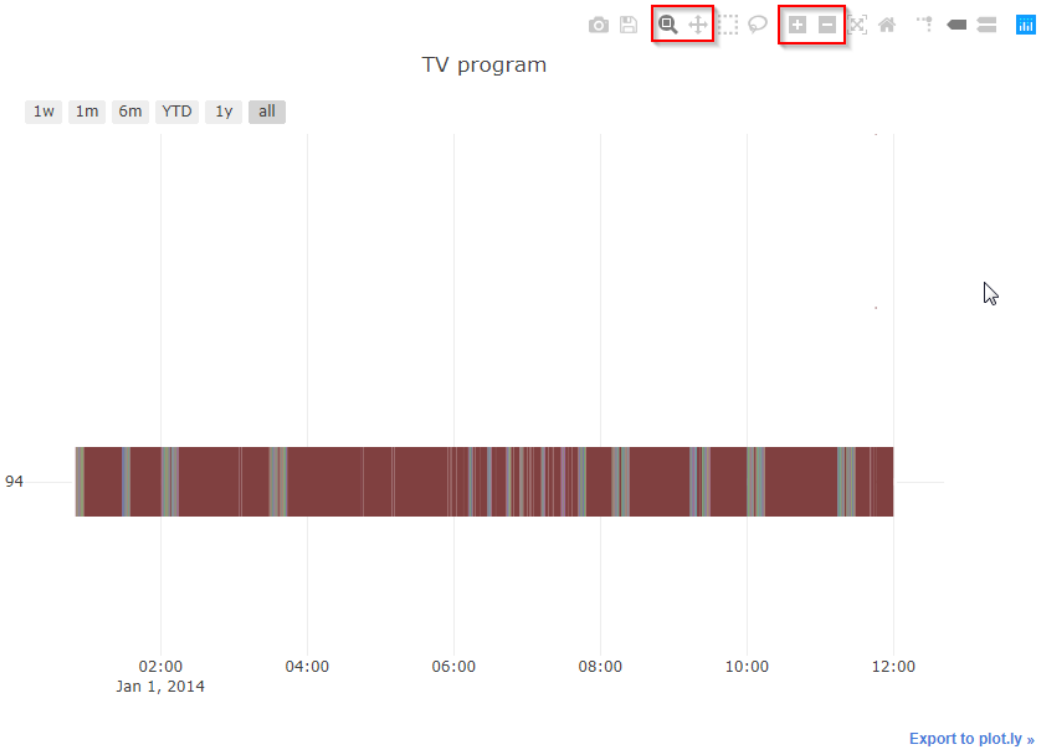
Rysunek 5.17: Przyciski nawigacji

5.2 Wizualizacja danych

Wizualizacja pliku instancji jest zapisywana do oddzielnego pliku. Pojedyncze statystyki z widoku wizualizacji statystyk pojawiają się jako element interfejsu, z kolei wykresy i macierz generowane z tego widoku zapisywane są do pliku tymczasowego i automatycznie otwierane.

5.2.1 Program telewizyjny

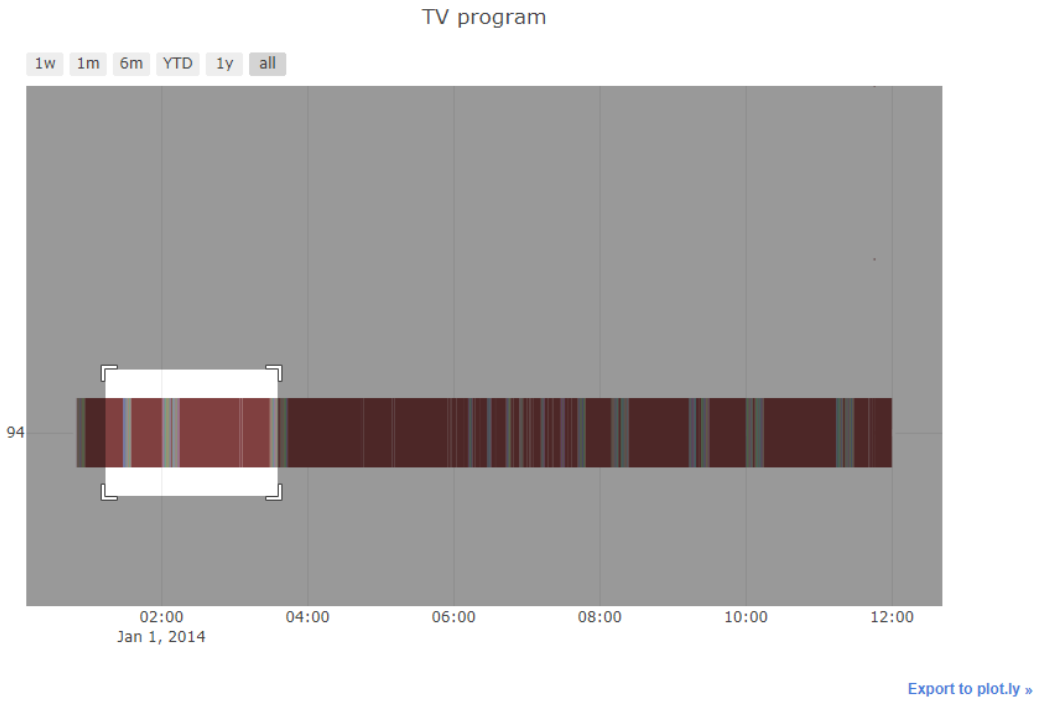
Wizualizacja programu telewizyjnego generowana jest do pliku HTML i otwierana w przeglądarce. Na Rys. 5.18 widać pełen wygenerowany widok.



Rysunek 5.18: Pełen widok

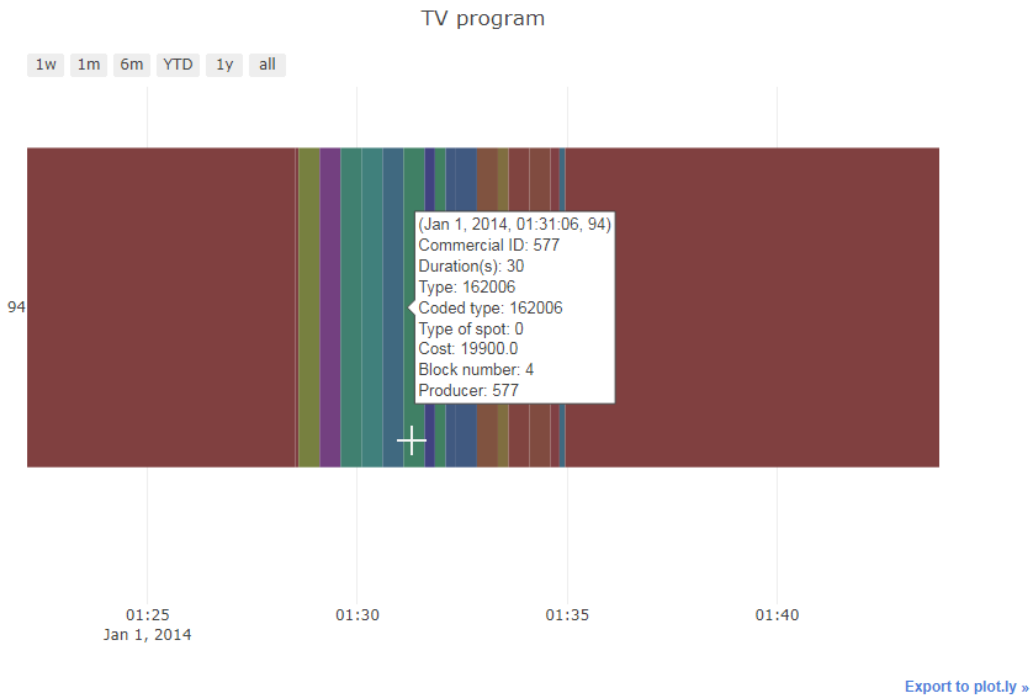
Jak można zauważyć, przy dłuższym przedziale czasowym instancji początkowy wygląd jest mało czytelny. Istnieje jednak możliwość nawigacji po wykresie - przybliżanie, wybór fragmentu. Aby wybrać fragment wykresu do przybliżenia, należy zaznaczyć obszar przytrzymując lewy przycisk myszy, tak jak zaprezentowano na Rys. 5.19. Dodatkowo, w prawym górnym rogu znajdują

się przyciski służące do nawigacji i obsługi wykresu. Są one tworzone automatycznie przez bibliotekę odpowiedzialną za generację wykresu. Na Rys. 5.18 czerwoną ramką zostały oznaczone te przyciski, które są ważne lub użyteczne dla wizualizacji. Pierwsze dwa odpowiadają za tryb kursora, kolejno: wybór obszaru do przybliżenia, przesuwanie wykresu. Kolejne dwa to przybliżenie/oddalenie. Pozostałe nie mają większego zastosowania w omawianej aplikacji.



Rysunek 5.19: Sposób wyboru fragmentu

W ramówce każda reklama jest oznaczona innym kolorem. Każdy program natomiast posiada jeden, ten sam kolor. Po najechaniu kursorem w pobliże początku bloku danego koloru wyświetli jego dane, tak jak na Rys. 5.20.



Rysunek 5.20: Przybliżenie fragmentu programu

5.2.2 Podsumowanie instancji

Pierwszą stronę pliku PDF zawierającego statystyki wybranej instancji zajmuje tabela zawierająca dane statystyczne opisujące całą instancję. Pozwala ona na przeanalizowanie cech kluczowych dla danej instancji.

W tabeli przedstawiono wartości liczbowe dla miar takich jak:

- **Total number of commercials:** całkowita liczba reklam występujących w instancji
- **Total number of programs:** całkowita liczba programów występujących w instancji
- **Sum of commercial durations[s]:** suma długości wszystkich reklam mierzona w sekundach
- **Sum of program durations[s]:** suma długości wszystkich programów mierzona w sekundach
- **Percent of commercial time in instance:** stosunek sumarycznego czasu trwania reklam do całości czasu trwania instancji wyrażony w procentach
- **Percent of program time in instance:** stosunek sumarycznego czasu trwania programów do całości czasu trwania instancji wyrażony w procentach
- **Minimum commercial duration[s]:** minimalny czas trwania reklamy w sekundach
- **Minimum program duration[s]:** minimalny czas trwania programu w sekundach
- **Maximum commercial duration[s]:** maksymalny czas trwania reklamy w sekundach
- **Maximum program duration[s]:** maksymalny czas trwania programu w sekundach

- **Average commercial duration[s]:** średni czas trwania reklamy w sekundach
- **Average program duration[s]:** średni czas trwania programu w sekundach
- **Standard deviation of commercial durations[s]:** odchylenie standardowe długości trwania reklamy w sekundach
- **Standard deviation of program durations[s]:** odchylenie standardowe długości trwania programu w sekundach
- **Median commercial duration[s]:** mediana długości trwania reklamy w sekundach
- **Median program duration[s]:** mediana długości trwania programu w sekundach

Poniższy rysunek przedstawia przykładową tabelę zawierającą dane dla wybranej instancji (Rys. 5.21).

	Value
Total number of commercials	249
Total number of programs	57
Sum of commercial durations[s]	6521
Sum of program durations[s]	32497
Percent of commercial time in instance	16.71
Percent of program time in instance	83.29
Minimum commercial duration[s]	8
Minimum program duration[s]	5
Maximum commercial duration[s]	45
Maximum program duration[s]	3594
Average commercial duration[s]	26.19
Average program duration[s]	570.12
Standard deviation of commercial durations[s]	7.04
Standard deviation of program durations[s]	924.67
Median commercial duration[s]	30.0
Median program duration[s]	130.0

Rysunek 5.21: Tabela podsumowująca instancję

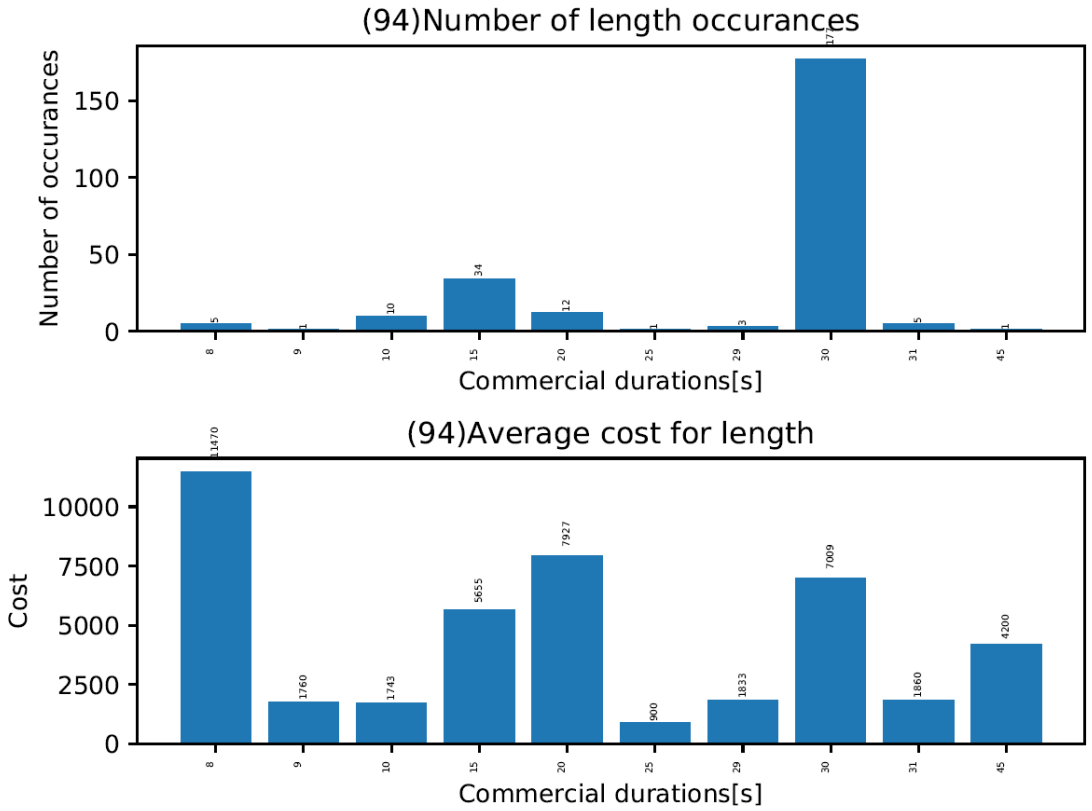
5.2.3 Wykresy statystyk instancji

Każde żądanie wizualizacji wiąże się z wygenerowaniem wykresów i zapisaniem ich do nowego pliku PDF. Dla wizualizacji instancji na jedną stronę przypadają dwa wykresy, które są również pogrupowane według kanałów. Nazwa kanału jest zapisana w nawiasach, w tytule wykresu. Dla wykresów statystyk instancji, dla każdego kanału, generowanych jest dziewięć wykresów. Są to:

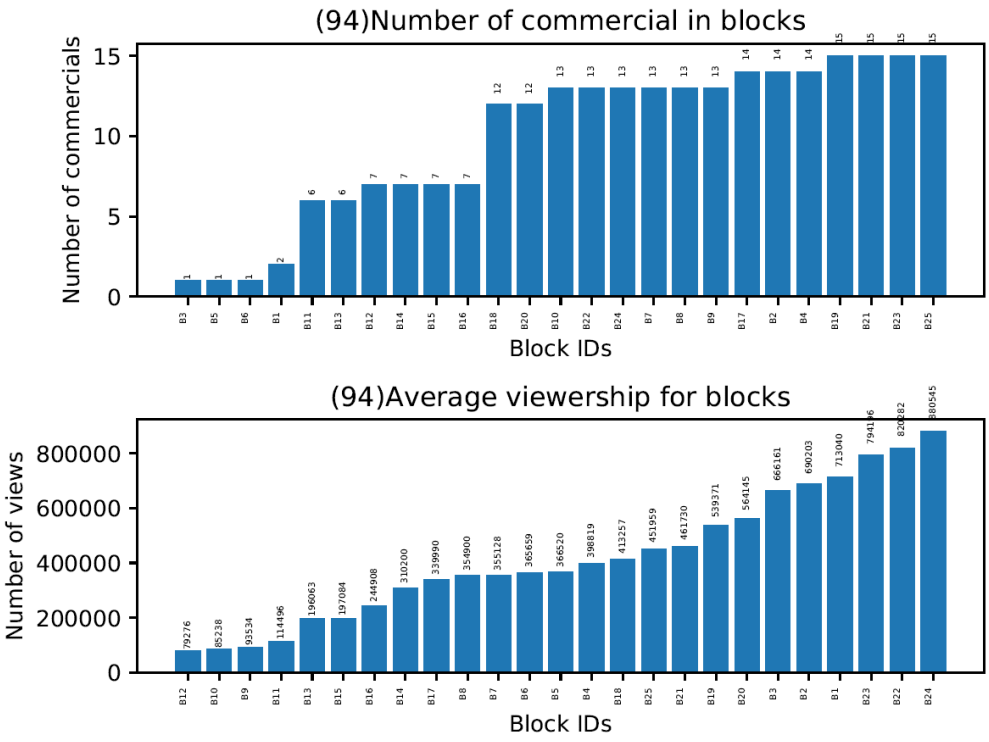
- **Number of length occurances** - liczba wystąpień reklam o konkretnej długości (Rys. 5.22)
- **Average cost for length** - średni koszt reklam w zależności od długości (Rys. 5.22)

- **Number of commercial in blocks** - liczba reklam dla bloków na danym kanale (Rys. 5.23)
- **Average viewership for blocks** - średnia oglądalność dla bloków (Rys. 5.23)
- **Min/Max viewership for blocks** - minimalna i maksymalna oglądalność dla bloków (Rys. 5.24)
- **Average commercial time for blocks** [s] - średni czas trwania reklamy dla bloków (Rys. 5.24)
- **Min/Max time of commercial for blocks** [s] - minimalny i maksymalny czas trwania reklamy dla bloków (Rys. 5.25)
- **Average cost for blocks** - średni koszt dla bloków (Rys. 5.25)
- **Min/Max cost of commercial for blocks** - minimalny i maksymalny koszt dla bloków (Rys. 5.26)

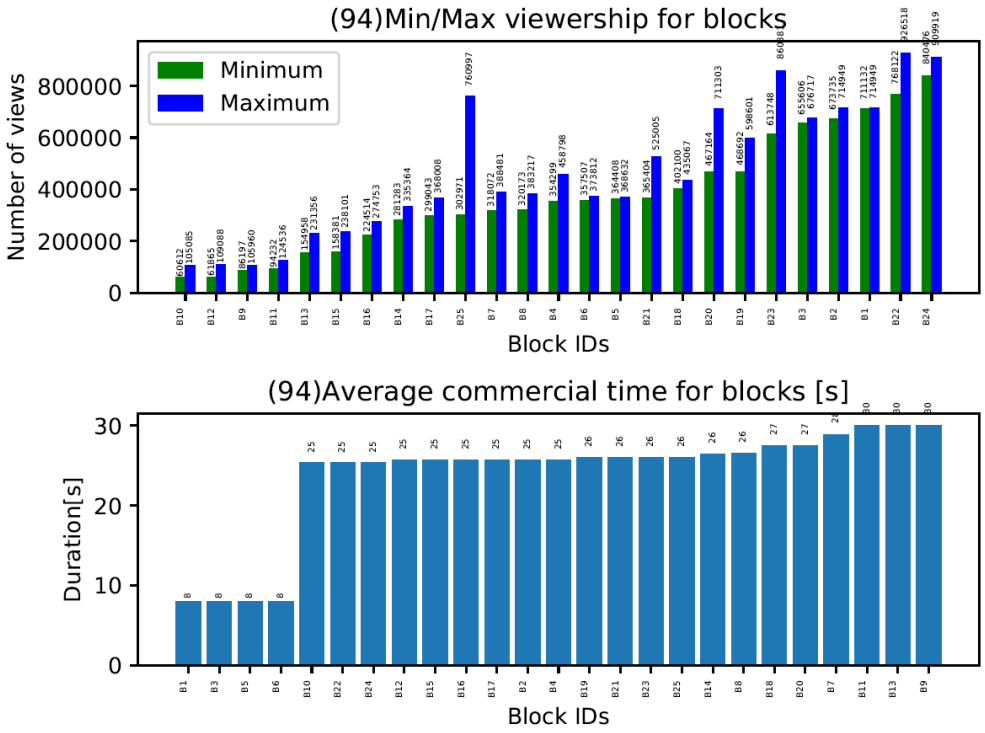
Rysunki od 5.22 do 5.26 są przykładami powyżej wymienionych wykresów. Wszystkie odnoszą się do okresu zawartego w wybranej do wizualizacji instancji oraz kanału *(94)*. Przykładowe wykresy zawierają liczbę 94 w nazwie. Liczba ta oznacza nazwę kanału, dla której wygenerowano statystyki powstałą w trakcie anonimizacji danych. W przypadku, gdy instancja została wygenerowana dla wielu kanałów, zestaw wykresów opisanych powyżej generowany jest dla każdego z kanałów z osobna.



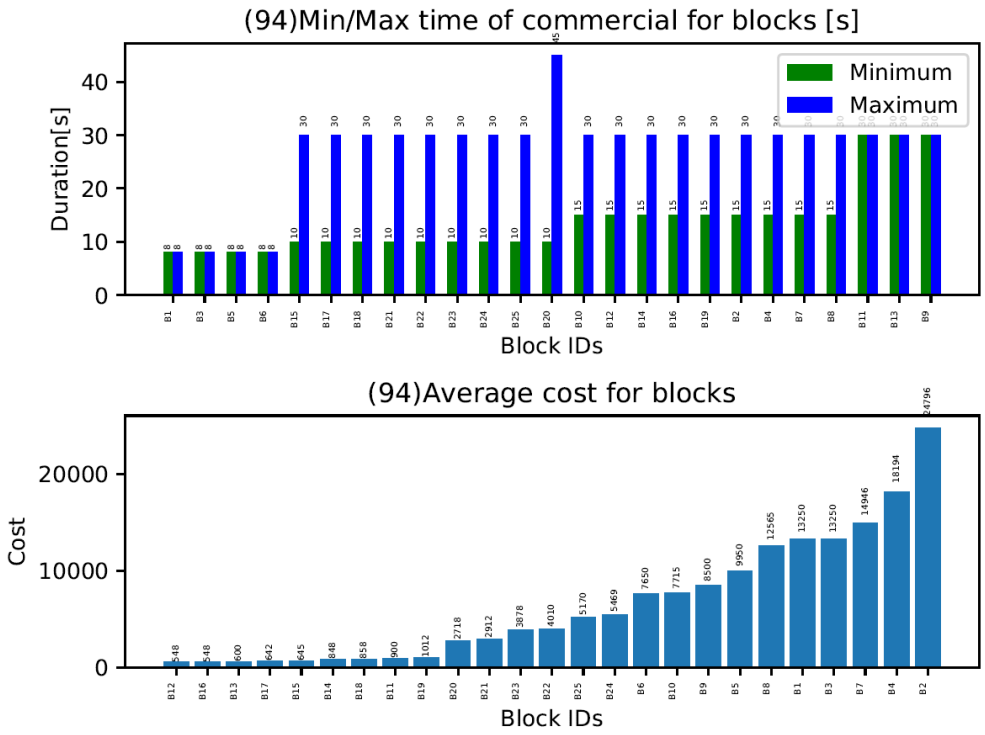
Rysunek 5.22: Przykładowe wykresy wystąpień i średnich kosztów dla długości reklamy



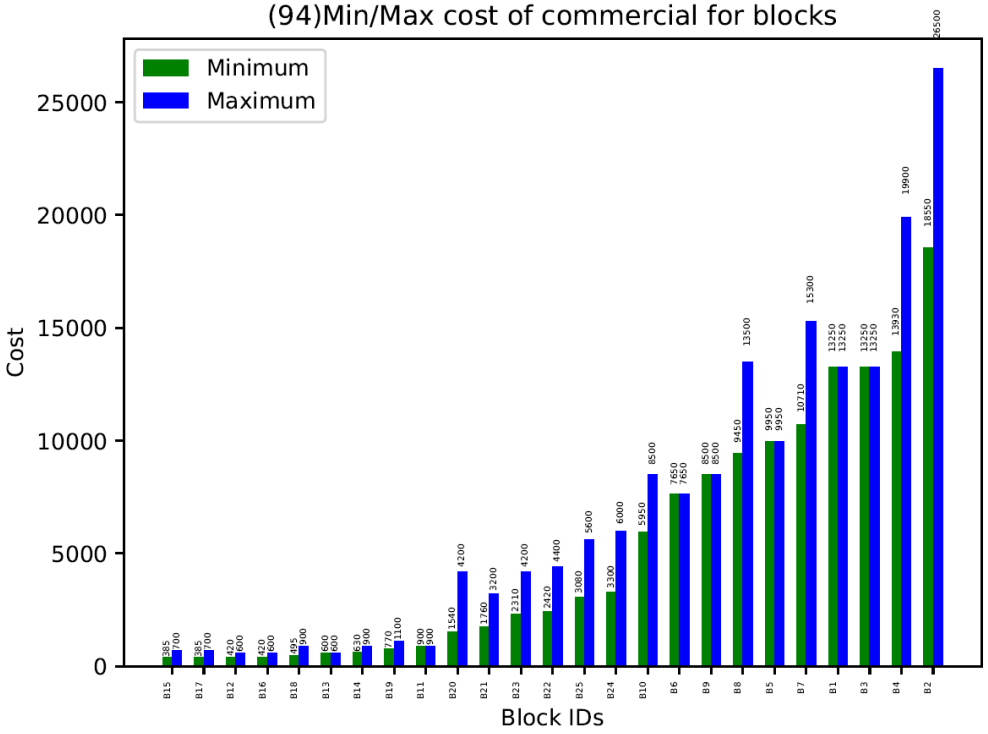
Rysunek 5.23: Przykładowe wykresy liczby reklam i średniej oglądalności dla bloków



Rysunek 5.24: Przykładowe wykresy minimalnych i maksymalnych oglądalności i średnich czasów trwania reklam dla bloków reklamowych



Rysunek 5.25: Przykładowe wykresy minimalnych i maksymalnych czasów trwania reklam i średnich kosztów dla bloków reklamowych



Rysunek 5.26: Przykładowy wykres minimalnych i maksymalnych kosztów dla bloków reklamowych

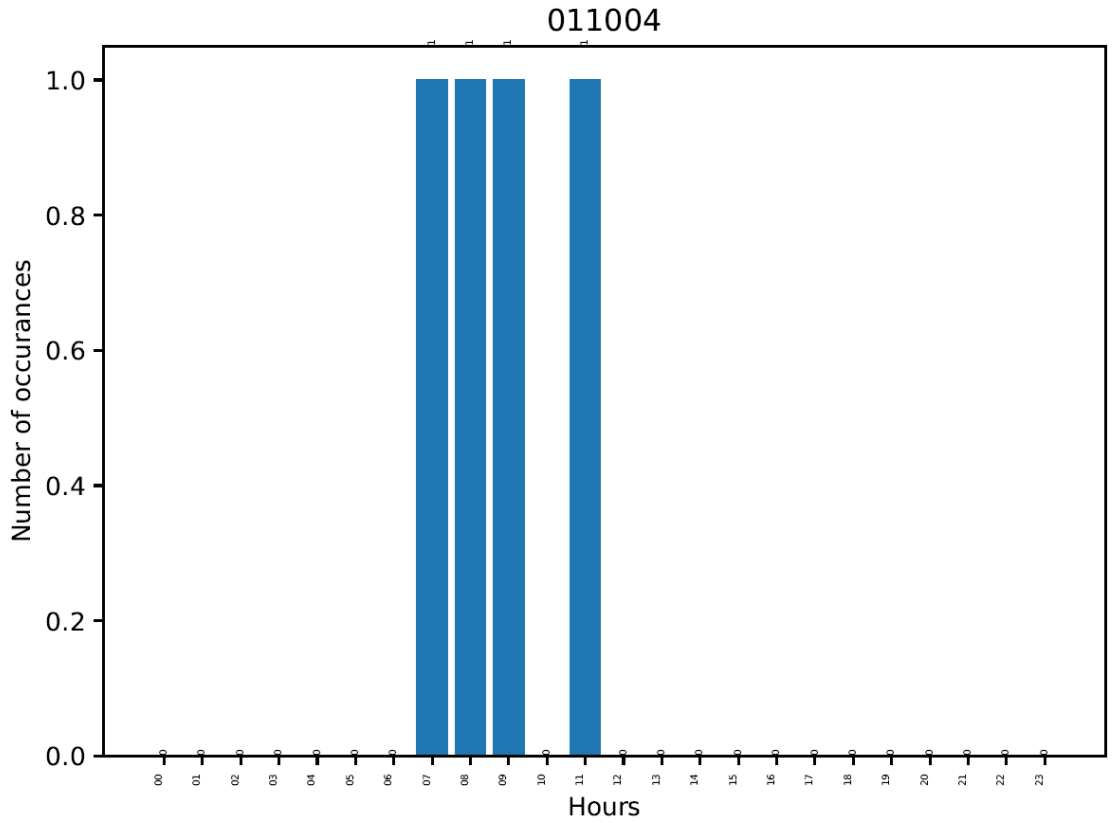
5.2.4 Wykresy statystyk generowanych na żądanie

Generowaniem statystyk na żądanie zajmuje się podmoduł modułu statystyk odpowiedzialny za generację szczegółowego opisu reklam. Wśród wygenerowanych przez moduł statystyk, dwie z nich są przedstawiane w postaci wykresów. Obie statystyki dotyczą przedziału czasu zadanego w wygenerowanej instancji testowej.

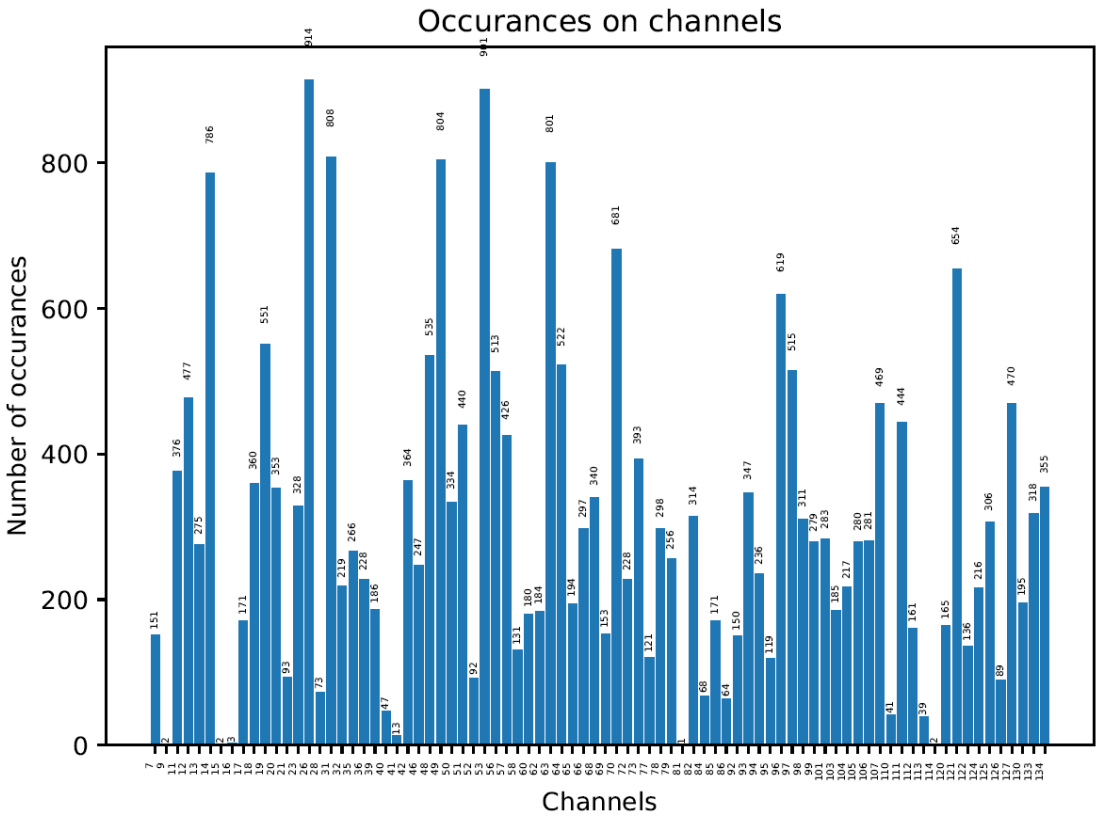
Statystykami generowanymi na żądanie są:

- liczba wystąpień reklamy w ciągu doby dla przedziału czasu zadanego w instancji (Rys. 5.27)
- lista kanałów na której wyświetlana była zadana reklama (Rys. 5.28)

Lista kanałów, na których była wyświetlana reklama może dotyczyć także całości danych historycznych. Wygląd wykresu jest wtedy identyczny jak dla przedziału instancji, lecz zawiera więcej informacji.



Rysunek 5.27: Liczba wystąpień reklamy w ciągu doby dla przedziału czasu zadanego w instancji



Rysunek 5.28: Lista kanałów na których wyświetlana była zadana reklama dla całości danych historycznych

5.2.5 Szczegółowe statystyki opisowe

Wśród statystyk generowanych na żądanie użytkownika, generowane są także statystyki, których wyniki można przedstawić w formie liczby lub opisu. Zdanie opisujące daną statystykę pojawia się w dedykowanym elemencie interfejsu jak na (Rys. 5.29). Wszystkie opisy generowane są w języku angielskim i dotyczą przedziału dla czasu zadanego w danej instancji. Zaprezentowana statystyka przykładowa przedstawia czas pierwszej i ostatniej emisji reklamy oraz całkowitą liczbę jej emisji.

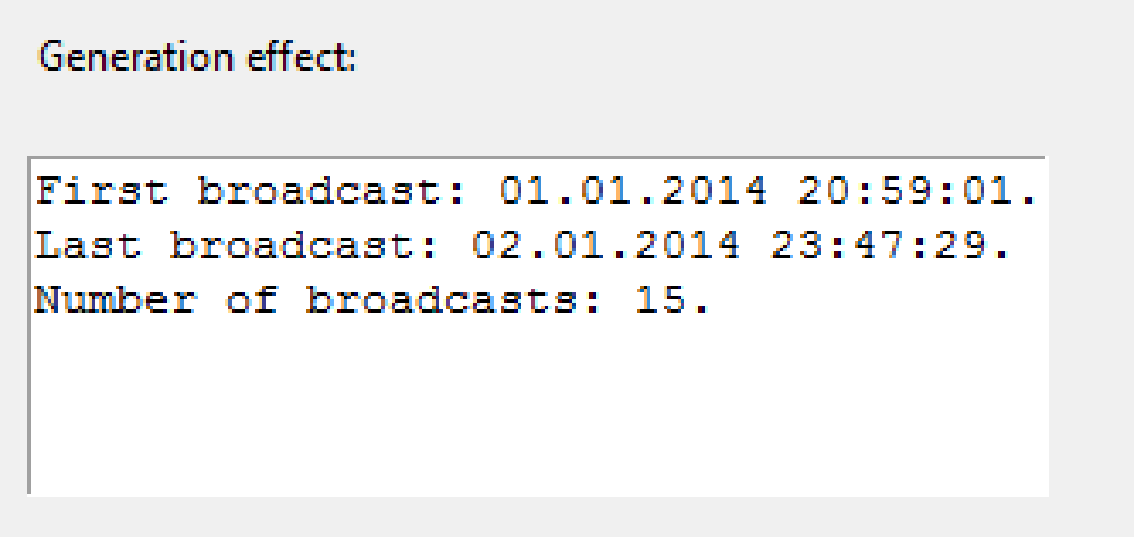
Pełna lista komunikatów generowanych przez moduł na żądanie użytkownika obejmuje następujące pozycje:

- "Commercial has appeared a maximum of *(liczba emisji)* times in a single commercial break."
  Komunikat ten określa maksymalną liczbę emisji reklamy w przerwie reklamowej.

- "First broadcast: *(data pierwszego wyświetlenia reklamy w formacie dzień.miesiąc.rok godzina:minuta:sekundy)*
  Last broadcast: *(data ostatniego wyświetlenia w formacie dzień.miesiąc.rok godzina:minuta:sekundy)*
  Number of broadcasts: *(liczba emisji)*"
  Określa czas pierwszej i ostatniej emisji unikalnej reklamy oraz całkowitą liczbę ich emisji w danej instancji lub czas pierwszej i ostatniej emisji reklam dla zadanego reklamodawcy oraz całkowitą liczbę ich emisji w danej instancji w zależności od wybranej opcji.

- "Number of different adds from given producer: *(liczbę unikalnych reklam dla danego reklamodawcy)*"

Komunikat określa liczbę unikalnych reklam dla wybranego przez użytkownika reklamodawcy.



Rysunek 5.29: Przykład statystyki w formie opisowej

5.2.6 Statystyki niewizualizowane przez system

Wśród statystyk generowanych przez aplikację na podstawie danych historycznych znajdują się także takie, których system nie wizualizuje. Istnieje jednak możliwość odczytania tych statystyk z pliku wygenerowanej instancji. Są to między innymi:

- odchylenie standardowe i mediana długości reklamy w bloku reklamowym
- procentowy udział reklam i programów w przedziale czasu zadanym w instancji
- lista unikalnych długości reklam w zadanej instancji

5.2.7 Macierz sąsiadowania

W systemie przygotowano cztery typy macierzy sąsiadowania, są to między innymi:

- bezpośrednie sąsiedztwo reklam zadanego typu
- występowanie reklam zadanego typu w tym samym bloku reklamowym
- bezpośrednie sąsiedztwo reklam zadanego reklamodawcy
- występowanie reklam zadanego reklamodawcy w tym samym bloku reklamowym

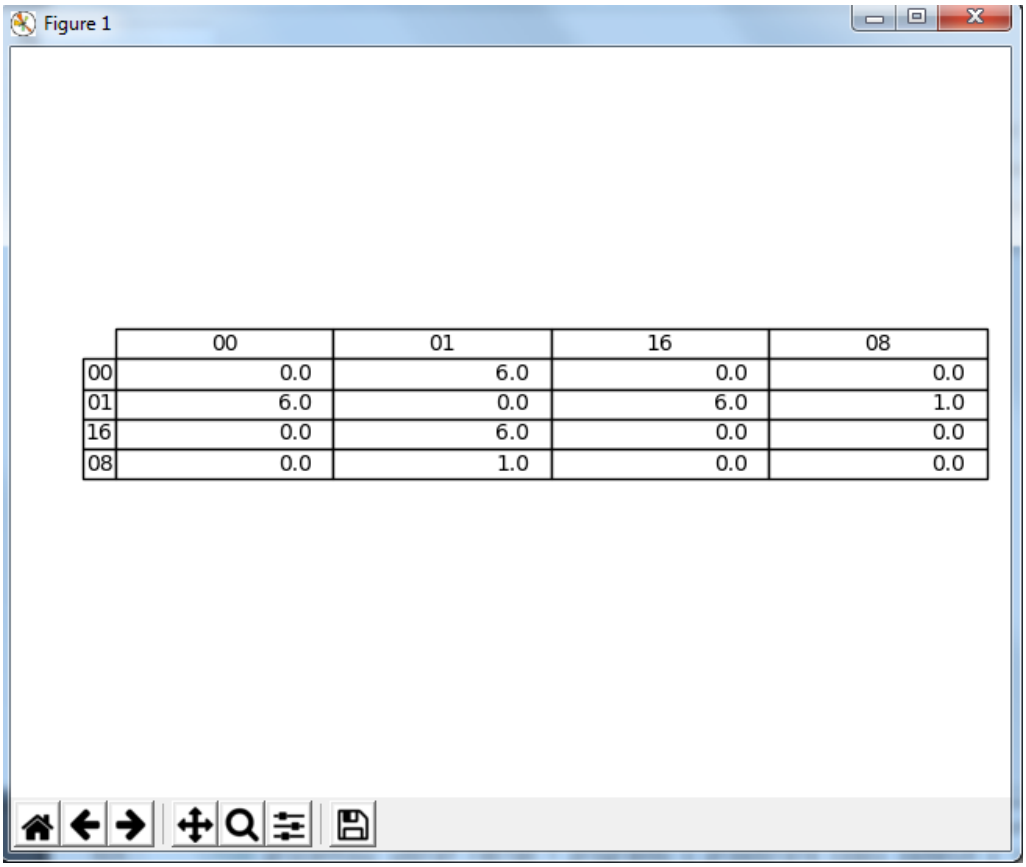
Macierze, podobnie jak wcześniej omówione wykresy, są również generowane do pliku PDF. W przypadku, gdy jest to część procesu wizualizacji instancji generowane są wszystkie cztery wymienione powyżej macierze - występują w takiej samej kolejności jak przedstawiono i zajmują ostatnie strony pliku oraz są grupowane po sektorze/producencie w celu poprawienia czytelności. Natomiast w przypadku gdy użytkownik systemu wybierze opcję generacji statystyk na żądanie, tworzona jest pojedyncza macierz, która następnie zostaje wyświetlona w oknie generowanym przez bibliotekę Matplotlib. Przykłady takich macierzy zostały przedstawione kolejno na Rys. 5.30, który zawiera

macierz umieszczoną w pliku PDF zawierającym statystyki stworzone dla analizowanej instancji oraz na Rys. 5.31, który przedstawia okno aplikacji zawierające statystkę przygotowaną na żądanie użytkownika systemu. Liczby będące nazwami wierszy i kolumn oznaczają zakodowane nazwy sektorów. Wartość w każdej komórce, w tym przypadku, to liczba reklam z danych sektorów rynku, które bezpośrednio ze sobą sąsiadują.

Przyciski widoczne na Rys. 5.31 w lewym dolnym rogu są automatycznie generowane przez bibliotekę Matplotlib. W kontekście tego projektu, pierwsze pięć z nich nie posiada funkcjonalności. Kolejne dwa są wykorzystywane w systemie: drugi od prawej otwiera okno ustawień pozycji tabeli oraz ostatni umożliwia zapisanie macierzy do pliku.

	18	00	05	16	01	09	19	10	11	08	04	17	20	14	03	06
18	8.0	182.0	16.0	19.0	14.0	4.0	15.0	5.0	3.0	4.0	3.0	1.0	3.0	1.0	1.0	2.0
00	182.0	308.0	69.0	93.0	41.0	52.0	36.0	20.0	19.0	9.0	26.0	2.0	5.0	6.0	0.0	4.0
05	16.0	69.0	6.0	19.0	7.0	7.0	7.0	2.0	13.0	6.0	4.0	2.0	4.0	0.0	0.0	0.0
16	19.0	93.0	19.0	40.0	21.0	7.0	24.0	5.0	18.0	11.0	5.0	4.0	0.0	1.0	3.0	0.0
01	14.0	41.0	7.0	21.0	12.0	3.0	4.0	6.0	5.0	4.0	2.0	2.0	0.0	0.0	0.0	0.0
09	4.0	52.0	7.0	7.0	3.0	0.0	2.0	2.0	5.0	1.0	4.0	0.0	1.0	0.0	0.0	0.0
19	15.0	36.0	7.0	24.0	4.0	2.0	20.0	1.0	5.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0
10	5.0	20.0	2.0	5.0	6.0	2.0	1.0	0.0	3.0	2.0	0.0	0.0	1.0	0.0	0.0	0.0
11	3.0	19.0	13.0	18.0	5.0	5.0	5.0	3.0	4.0	3.0	2.0	0.0	1.0	0.0	0.0	0.0
08	4.0	9.0	6.0	11.0	4.0	1.0	0.0	2.0	3.0	2.0	0.0	0.0	0.0	0.0	0.0	2.0
04	3.0	26.0	4.0	5.0	2.0	4.0	2.0	0.0	2.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0
17	1.0	2.0	2.0	4.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	1.0
20	3.0	5.0	4.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
14	1.0	6.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
03	1.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
06	2.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	1.0	1.0	0.0	0.0	0.0

Rysunek 5.30: Przykładowa macierz dla statystyk instancji



Rysunek 5.31: Przykładowa macierz dla statystyk na żądanie

Rozdział 6

Podsumowanie

W ramach niniejszej pracy inżynierskiej powstał system umożliwiający generowanie instancji testowych dla algorytmów optymalizujących ustawienie reklam w przerwie reklamowej na podstawie danych historycznych, a także generowanie statystyk i wizualizację tych danych. Aplikacja spełnia wszystkie postawione wymagania, istnieje jednak wiele możliwości jej rozwoju.

Aktualnie pobieranie danych z bazy opisującej program telewizyjny dla przedziału dat odbywa się poprzez porównywanie pojedynczych rekordów do daty początku i końca przedziału. Baza ta jest posortowana według daty więc istnieje możliwość usprawnienia tego procesu. Jednym ze sposobów poprawy może być wykorzystanie połowienia binarnego do znajdowania początku i końca przedziału.

Kolejną możliwością jest dostarczenie większej ilości danych historycznych. Pomiary z kilku lat pozwoliłyby na zbadanie jak zmieniają się preferencje odbiorców z biegiem czasu. Algorytmy mogłyby wykorzystywać te informacje w celu planowania przyszłych kampanii reklamowych.

Innym sposobem na dostarczenie nowych wartościowych danych może być rozszerzenie aplikacji o nowe funkcjonalności. Jedną z wielu możliwości jest podzielenie odbiorców na grupy, a następnie wyspecyfikowanie dla nich najczęściej oglądanych kanałów i typów programów.

W obecnych czasach coraz więcej danych jest zbieranych i przechowywanych na różnych nośnikach. Dzięki rozwojowi technologicznemu coraz większą wagę przykłada się do rosnącej ilości informacji. Należy więc mieć świadomość, że bez użycia zaawansowanej technologii człowiek nie jest w stanie wykorzystać zgromadzonych danych i odpowiednio ich przeanalizować.

Literatura

[1] E.Pesch G. Schmidt M. Sterna J. Węglarz J. Błażewicz, K. Ecker. *Handbook on Scheduling From Theory to Practice*. Springer, Heidelberg, 2019.

[2] M. Sterna. Scheduling model for television advertisement broadcasting problem. Technical Report RA-2/2018, Institute of Computing Science, Poznan University of Technology, 2018.

[3] M. J. Brusco. Scheduling advertising slots for television. *The Journal of the Operational Research Society*, 59(10):1363–1372, 2008.

[4] F. Díaz-Núñez, N. Halman, O.C Vasquez. The tv advertisements scheduling problem. *Optimization Letters*, 2018. doi: 10.1007/s11590-018-1251-0.

[5] R. H. Huang S. P. Chuang M. S. Wuang, C.L. Yang. Scheduling of television commercials. *IEEE International Conference on Industrial Engineering and Engineering Management*, pages 803 – 807, 2010.

[6] M. Beedle K. Schwaber. *Agile software development with Scrum*. Upper Saddle River, NJ : Prentice Hall, 2002.

[7] Krystian Kaczor. *Scrum i nie tylko. Teoria i praktyka w metodach Agile*. PWN, 2014.

[8] *Manifesto for Agile Software Development*. [on-line] `https://agilemanifesto.org/`, data dostępu: 14.12.2018.

[9] Ch. Lian M. Armbrust, R. S. Xin. Spark sql: Relational data processing in spark. *Interest Group on Management of Data*, pages 1383–1394, June 2015.

[10] *Apache Spark SQL*. [on-line] `https://spark.apache.org/docs/latest/sql-performance-tuning.html`, data dostępu: 02.01.2019.

[11] *Matplotlib*. [on-line] `https://matplotlib.org/`, data dostępu: 28.12.2018.

[12] *Plotly*. [on-line] `https://plot.ly/python/`, data dostępu: 03.01.2019.

[13] *TkInter*. [on-line] `https://wiki.python.org/moin/TkInter`, data dostępu: 03.01.2019.

[14] *Numpy*. [on-line] `http://www.numpy.org/`, data dostępu: 04.01.2019.

Dodatek A

Format pliku instancji testowej

```
kanał kanał kanał ...
data_początku(format - '%d.%m.%Y %H:%M:%S')
data_końca(format - '%d.%m.%Y %H:%M:%S')
%
Liczba_analizowanych_kanałów
Sumaryczna_liczba_programów
Sumaryczna_liczba_reklam_i_autopromocji
%
Kanał_1
czas_rozpoczęcia(format - '%Y-%m-%d %H:%M:%S')
R(wpis dotyczący reklamy) czas_trwania(sekundy) typ_programu
    numer_bloku_reklamowego typ_reklamy koszt umiejscowienie_spotu
    identyfikator_reklamy identyfikator_reklamodawcy oglądalność_ś
    rednia oglądalność_średnia_filtrowana
P(wpis dotyczący programu) czas_trwania(sekundy) typ_programu (średnia
     min max odchylenie)_oglądalności (średnia min max odchylenie)_oglą
    dalności_filtrowanej
A(wpis dotyczący autopromocji) czas_trwania(sekundy)
...
%
Kanał_2
czas_rozpoczęcia(format - '%Y-%m-%d %H:%M:%S')
... ramówki dla innych kanałów (programy, reklamy i autopromocje
    pojedynczo)
%
sumaryczna_liczba_bloków_reklamowych
%
Kanał_1
B(wpis dotyczący bloku reklamowego) czas_trwania(sekundy)
    numer_bloku_reklamowego max_długość_reklamy min_długość_reklamy
    liczba_reklam_w_bloku średnia_długość_reklamy odchylenie_długoś
    ci_reklam min_koszt_reklamy max_koszt_reklamy średni_koszt_reklamy
    odchylenie_kosztu_reklamy (średnia min max odchylenie)_oglądalności
     (średnia min max odchylenie)_oglądalności_filtrowanej
```

```
...
%
Kanał 2
... bloki dla innych kanałów
%
%STATISTICS%
liczba_reklam
liczba_programów
sumaryczna_długość_reklam_[s]
sumaryczna_długość_programów_[s]
procentowy_udział_reklam_w_czasie_antenowym
procentowy_udział_programów_w_czasie_antenowym
minimalna_długość_reklamy_[s]
minimalna_długość_programu_[s]
maksymalna_długość_reklamy_[s]
maksymalna_długość_programu_[s]
średnia_długość_reklamy_[s]
średnia_długość_programu_[s]
odchylenie_standardowe_długości_reklamy_[s]
odchylenie_standardowe_długości_programu_[s]
mediana_długości_reklamy_[s]
mediana_długości_programu_[s]
liczba_unikalnych_długości_reklamy
dur1 dur2 dur3 dur4 ... durn
liczba_typów_reklam
typ procentowy_udział_w_czasie_antenowym_reklam
...
%
liczba_znalezionych_różnych_sąsiednich_typów_reklam
typ1 typ2 liczba_sąsiadujących_reklam_danych_typów
...
%
liczba_znalezionych_różnych_sąsiednich_reklamodawców
reklamodawca1 reklamodawca2 liczba_sąsiadują
    cych_reklam_danych_reklamodawców
...
%
liczba_znalezionych_różnych_typów_reklam_w_przerwach
typ1 typ2 liczba_występujących_w_przerwach_reklam_danych_typów
...
%
liczba_znalezionych_różnych_reklamodawców_w_przerwach
reklamodawca1 reklamodawca2 liczba_występują
    cych_w_przerwach_reklam_danych_reklamodawców
...
%
```

Dodatek B

Instalacja i uruchamianie

Do uruchomienia i poprawnego działania aplikacja wymaga zainstalowanego środowiska Python w wersji 3.6 lub nowszej, a także Java JRE w wersji 8 lub nowszej. Wymagane są także następujące biblioteki języka Python:

- PySpark w wersji 2.3.2
- tk
- plotly
- numpy
- matplotlib

Zaleca się instalację wymienionych pakietów za pomocą narzędzia *pip*.

```
$pip install pyspark==2.3.2
$pip install plotly numpy matplotlib
```

Ważne jest aby główny węzeł PySpark pracował na tej samej wersji interpretera Python co węzły robocze wywoływane przez aplikację. W tym celu należy ustawić zmienną środowiskową PYSPARK_PYTHON na ścieżkę do prawidłowej wersji interpretera języka Python.
Aby uruchomić aplikację należy wywołać w interpreterze języka Python plik *run.py*.

```
$python run.py
```

Instytut Informatyki, Wydział Informatyki
Politechnika Poznańska

Skład przy użyciu systemu LaTeX.

BibTeX:

```
@mastersthesis{ key,
    author = " Oskar Kostowski \and Wojciech Obst \and Laura Rakiewicz \and Piotr Terczyński ",
    title = "{System wspierający testowanie algorytmów optymalizujących uszeregowanie reklam
telewizyjnych}",
    school = "Poznan University of Technology",
    address = "Pozna{\'n}, Poland",
    year = "2019",
}
```