

Programowanie zespołowe

Zadanie 5 - raport z testów

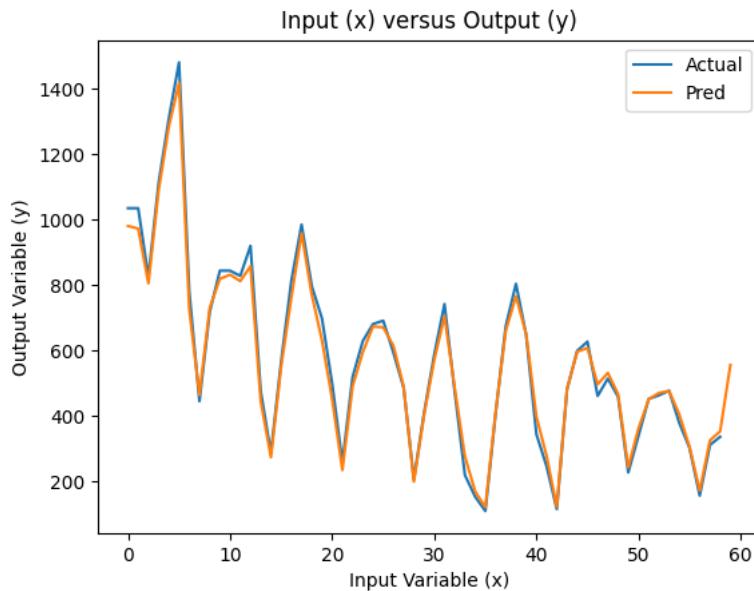
Piotr Popis
Mateusz Wałejko
Adrian Majcher
Jakub Kazimierski

Styczeń 16, 2021

1 Sieć Neuronowa

1.1 Testy jakościowe sieci

Aby przetestować jakość sieci został wybrany losowy zbiór ilości zakażeń dziennych. Takie dane zostały podzielone na dwa zbiory - treningowy i testowy w stosunku 1:2. Próbek niestety było mało, ale maxymalnie je wykorzystaliśmy. Najpierw do sieci podajemy zbiór treningowy, a później na modelu dopasowanym do tego zbioru porównujemy predykcje z wartościami ze zbioru testowego zgodnie z wybraną strategią. Wyniki tego testu wyglądają następująco:



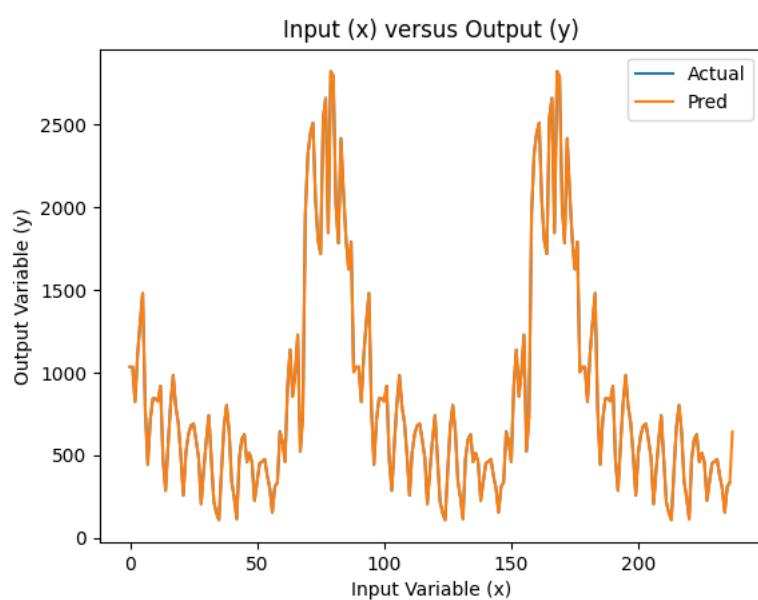
mean – squared – error : 1411.9772

mean – squared – logarithmic – error : 1.4103

mean – absolute – percentage – error : 169.4675

mean – absolute – error : 34.3304

Powyzsza powinna być przetestowana też w wymiarze przyszłości. Mianowicie na ten moment ilość naszych próbek może być niewystarczająca, więc powinniśmy sprawdzić czy wyniki jej predykcji nie są względnie lepsze dla większej ilości danych. Sprawdźmy czy jakość predykcji poprawi się, gdy ilość danych wejściowych się powiększy, a owe dane będą mniej lub bardziej schematyczne". Poniżej zachowanie się sieci dla zbiorów o większej ilości próbek.

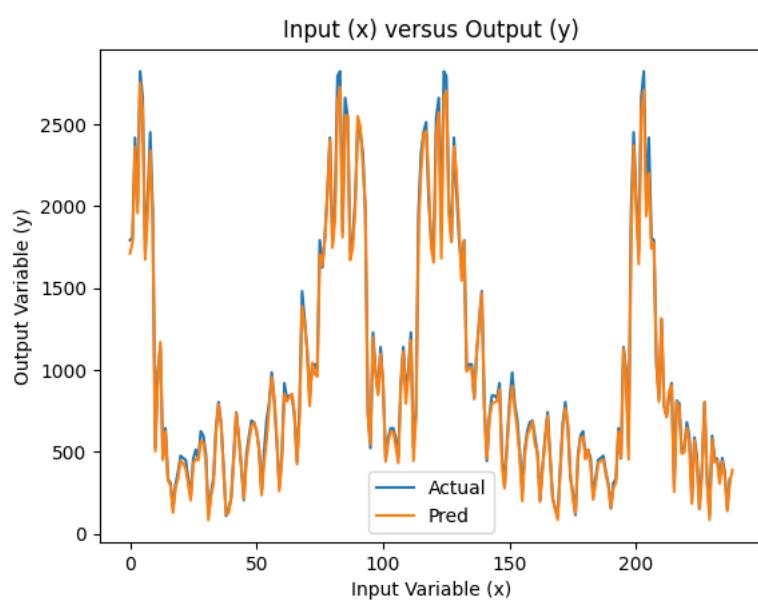


mean - squared - error : 146977.5938

mean - squared - logarithmic - error : 21.0174

mean - absolute - percentage - error : 674.6416

mean - absolute - error : 367.1998



mean - squared - error : 17164.0000

mean - squared - logarithmic - error : 9.8071

mean - absolute - percentage - error : 7224.8936

mean - absolute - error : 3772.8303

Funkcja testująca:

```
def test_neural_network(data, model=None):
    y = asarray(data)
    x = arange(len(data))
    y_train, y_test, x_train, x_test = train_test_split(y, x, test_size=0.33)
    guesser = Predictioner(model_input=model)
    guesser.update_input(train_x=x_train, train_y=y_train)
    history = guesser.fit_model(verbose=0)
    eva = guesser.evaluate(y_test, x_test)
    guesser.predict(x)
    guesser.visualize()
    return eva[0]
```

1.2 Testowanie struktury sieci- metodyka

Do testowania zmian jakości przy zmianie struktury: ilości layerów, aktywacji, ilości neuronów pozwoliliśmy sobie zaimplementować local search o pętli głównej:

```
def local_search(secs, data):
    end_time = get_time() + secs * 1000
    model = get_random_model()
    loss = test_neural_network(data, model)
    best = loss
    best_model = model
    while get_time() < end_time:
        model = get_random_model()
        loss = test_neural_network(data, model)
        if loss < best:
            best = loss
            best_model = copy.deepcopy(model)
            print("Best moved to: {}".format(best))
```

Poniżej tylko część wygenerowanego log'u przejść:

```
mse : 2043.6075 - msle : 10.4585 - mape : 97.7305 - mae : 39.4280
mse : 2051.8403 - msle : 10.5316 - mape : 15137701.0000 - mae : 39.6216
mse : 2297.4768 - msle : 10.7153 - mape : 96.5543 - mae : 41.4534
mse : 2248.6128 - msle : 10.3958 - mape : 96.4404 - mae : 40.6515
mse : 2492.3628 - msle : 11.5391 - mape : 15875577.0000 - mae : 45.4920
mse : 2378.7021 - msle : 10.6031 - mape : 98.0841 - mae : 40.8325
```

Legenda:

mean absolute percentage error = mape

mean squared error = mse
mean squared logarithmic error=msle
mean absolute error=mae

1.3 Podsumowanie

Teoretycznie z góry możemy założyć, że ilość przypadków względem lat powinna spadać (między innymi przez czynniki typu: rozwój medycyny, szczepienia, badania, większe obycie społeczności z tematyką rozpostrzreniania się wirusa,...). Względem pór roku natomiast możemy pokusić się o stwierdzenie, że w okresie letnim przypadków jest mniej ze względu na promieniowanie słoneczne, które implikuje wyższą temperaturę, może zwiększyć poziom witaminy D w organizmie, a zimą przypadków powinno być więcej.

Drugim problemem jest to, że nasze dane wejściowe to dane bardzo "zozprze-strzenione". Sieć ma problemy z zauważeniem pewnych schematów, zbieżności. Trzecią kwestią jest to, że każde województwo ma zupełnie inne zbiorы danych. Czwartą podział na przypadki wyleczone, zakażenia i zgony. Nasze predykcje zatem nie są perfekcyjne i prawdopodobnie potrzebujemy dużo, dużo więcej danych, aby predykcje takich danych były bardzo dobre. Mimo wszystko Możemy zauważać, że wartość dla dnia $n+1$ jest w zakresie "blędna", a nasze wyniki prezentują się naprawdę dobrze.

2 Testy manualne

2.1 Założenia

Niezwykle istotną częścią projektu, jest interfejs graficzny użytkownika, bowiem to za jego pomocą, osoba korzystająca z aplikacji, otrzymuje możliwość odbioru oraz przeanalizowania danych dotyczących sytuacji związanej z pandemią wirusa Covid-19. Szata graficzna została podzielona na trzy najważniejsze segmenty, są to:

- 1) Responsywna Mapa Polski
Wydzielić w niej można podgląd na osoby:
1.1)zakażone



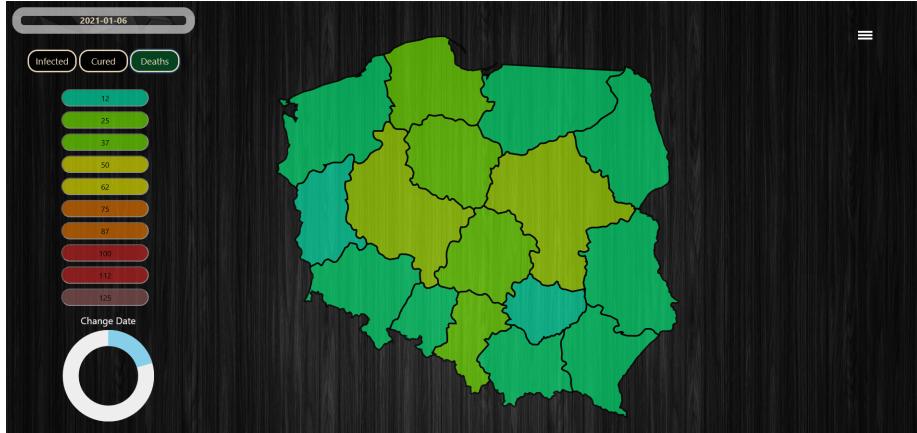
2.1 Responsive map

1.2)Uleczone



2.2 Cured cases

1.3)Zmarłe



2.3 Death cases

Dla zapewnienia komfortu użytkownikowi, województwa charakteryzują się odpowiednią intesywnością oraz barwą koloru, świadczącego o aktualnym, przypadającym na wybrany rejon rozwoju badanej sytuacji. Dodatkowo uwzględniono możliwość wyboru okresu w którym badana sytuacja miała miejsce. Użyty w tym celu został scroll-Knob.

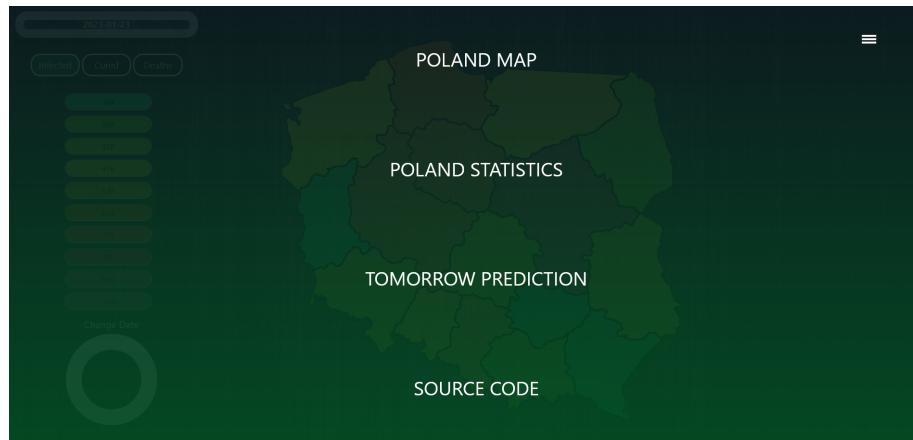
Przekazywane na ekran wyniki, są tworzone na podstawie, danych z Bazy, które za pomocą prostych metod statystycznych, są uśredniane, tak, aby powodować jak najmniejsze odchylenia danych na przestrzeni czasu.

Dla wnikliwych użytkowników, przygotowany został panel boczny, zawierający dokładne liczbowe odwzorowanie sytuacji prezentowanej na mapie za pomocą kolorów. Pojawia się on po wyborze przez osobę korzystającą z aplikacji, danego województwa.



2.4 Side panel

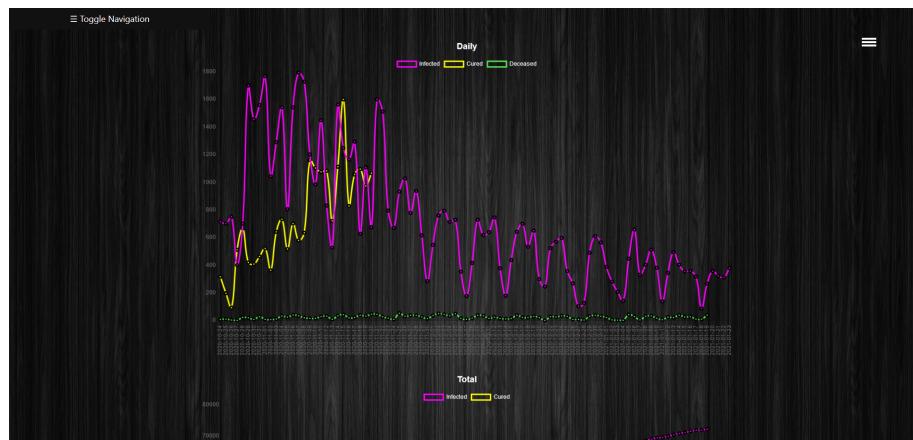
Wspomniane zostały również inne podstrony graficznego interfejsu, użytkownik, może uzyskać do nich dostęp za pomocą menu, którego przywołanie odbywa się poprzez interakcję z przyciskiem w prawym górnym rogu.



2.5 Menu

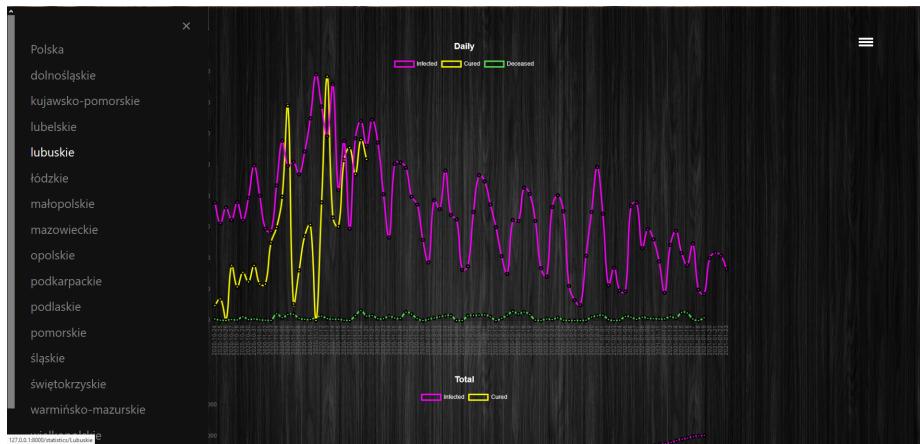
Zawiera on przekierowania na stronę dotychczasowych statystyk prezentowanych na interaktywnych wykresach, a także przewidywanych za pomocą starynie dobranej sieci neuronowej, przypadków Covid-19 na najbliższy czas.

Przejdzmy więc do podstrony statystyk.

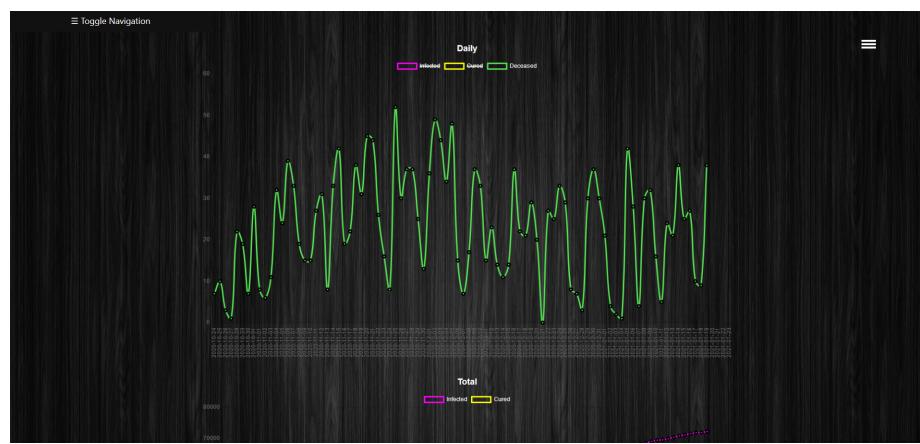


2.6 Charts

Wykresy, pozwalają użytkownikowi zarówno na obserwację wszystkich danych jak i tylko tych przezeń wybranych.



2.7 Charts



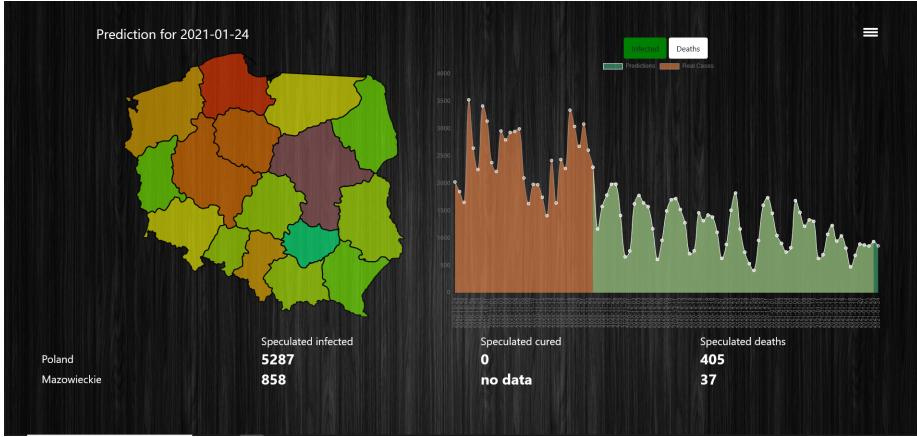
2.8 Charts

Wykresy wizualizują faktyczny stan najbardziej istotnych zagadnień jak np. stosunek osób, którym udało się wyzdrowieć do osób, które wciąż są zakażone, czy też dostępność respiratorów.

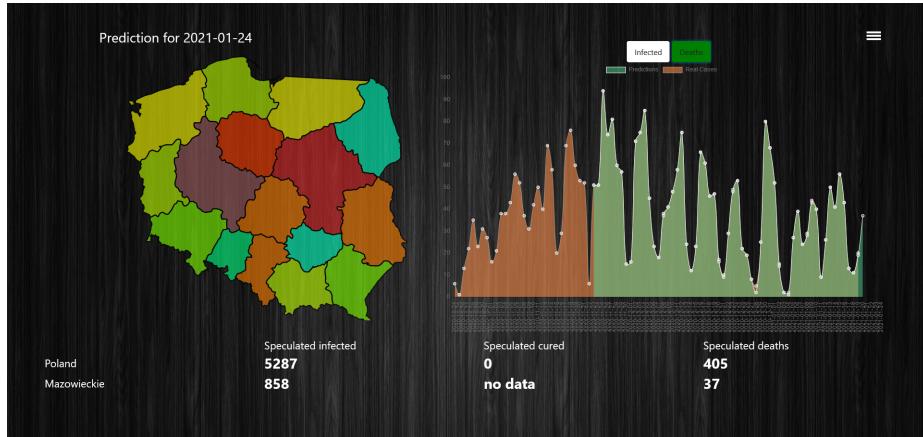


2.9 Charts

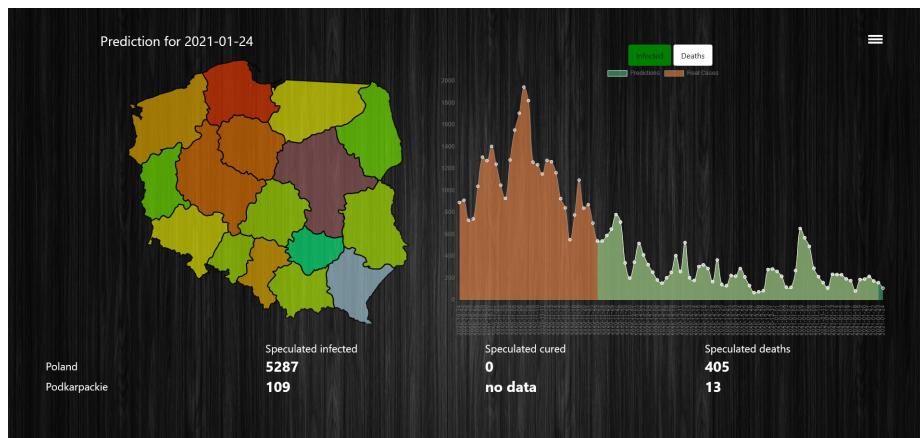
Ostatnią omawianą postroną, jest ta zawierająca przewidywany stan pandemii w najbliższym czasie. Poza dokładnym odwzorowaniem stanu faktycznego i przewidywanego, na interaktywnym wykresie, zawiera również pomniejszoną wersję, interaktywnej mapy polski, pozwalającej użytkownikowi, skorzystać z wizualizacji dla danego województwa.



2.10 Predictions

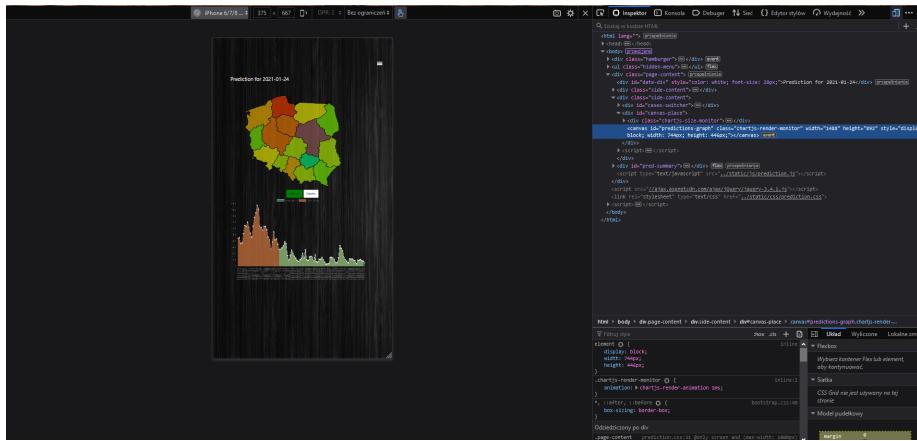


2.11 Predictions



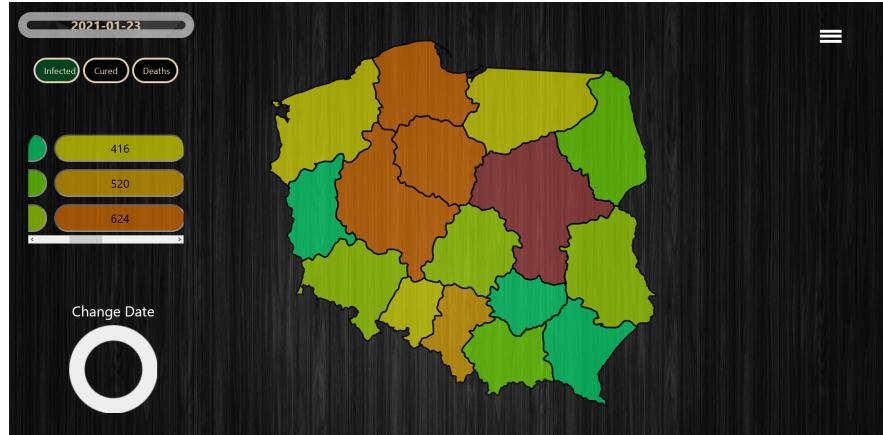
2.12 Predictions

Pozostało poruszyć ostatnią kwestię, a mianowicie responsywność. Po poszukiwaniach najlepszego sposobu, ostatecznie udało się uzyskać dostosowaną ergonomicznie do wizualnego odbioru aplikację, która działa zarówno na urządzeniach mobilnych, jak i desktopowych. Oto fragmenty testów responsywności:

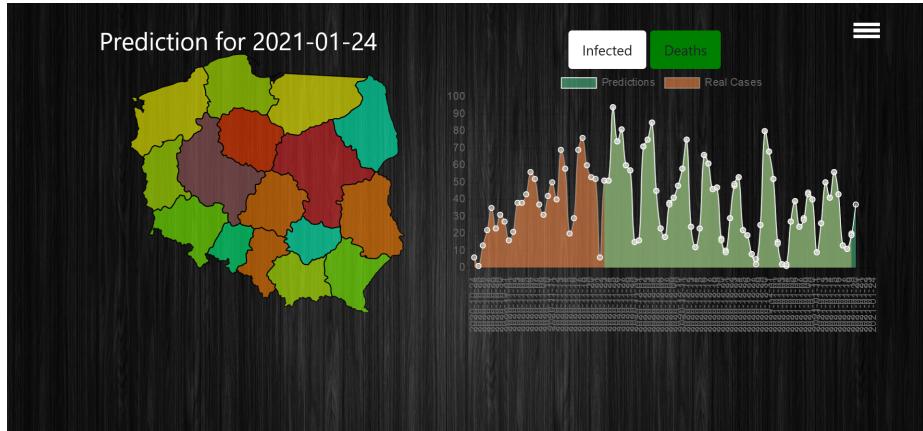


2.13 Responsivity test

Badanie znacznego przybliżenia w oknie przeglądarki:



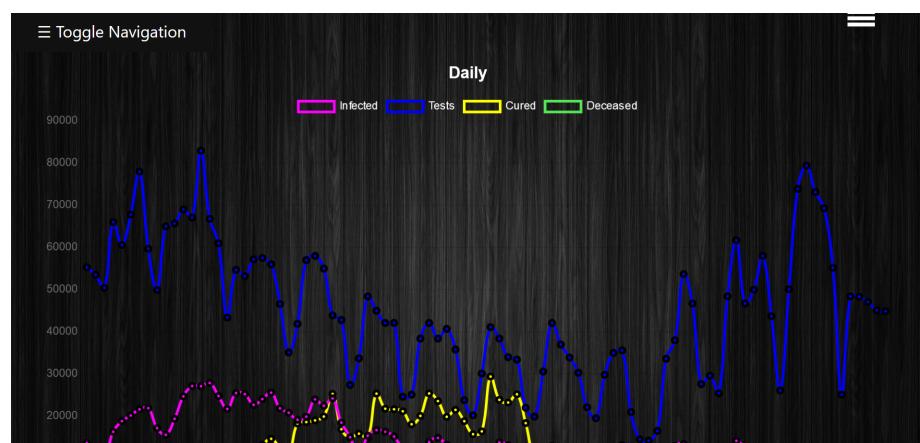
2.14 Responsivity test



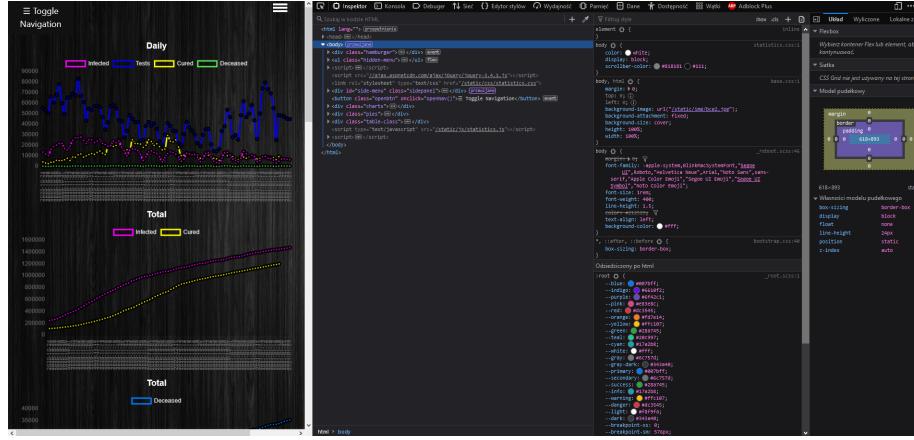
2.14 Responsivity test



2.15 Responsivity test



2.15 Responsivity test



2.16 Responsivity test

2.2 Podsumowanie

Przeprowadzone testy funkcjonalności uwzględniały skrajne przypadki użytkowania aplikacji, szczególnie zaimplementowana responsywność, pozwala na utrzymanie całości programu w formie umożliwiającej korzystanie z jego własności, takich jak odczyt i interpretacja zwizualizowanych danych, a także kontrola ram czasowych. Dzięki temu aplikacja ma szansę być wykorzystana zarówno przez użytkowników komputerów stacjonarnych jak i przenośnych, ale także przez tych korzystających z urządzeń mobilnych, gdzie rozdzielcość ekranów jest znacznie mniejsza.

3 Testy jednostkowe

[LINK](#)