НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Курсовий проект

із дисципліни «Алгоритми і системи комп’ютерної математики»

На тему

«Прогнозування кількості хворих на COVID-19»

Етап №6

Виконав: Керівник:

студент групи КМ-93 доцент

Костенко О. А. Олефір О. С.

Київ — 2022

Зміст

[Опис обраного математичного методу 3](#_Toc123500914)

[Архітектура програмних засобів 6](#_Toc123500915)

[Формати вихідних даних 6](#_Toc123500916)

[Блок-схема алгоритму 7](#_Toc123500917)

[Код програми 8](#_Toc123500918)

[Опис результатів 10](#_Toc123500919)

[Висновки 12](#_Toc123500920)

# Опис обраного математичного методу

Підхід авторегресійного інтегрованого ковзного середнього (англ. autoregressive integrated moving average, ARIMA) до часових рядів полягає в тому, що в першу чергу оцінюється стаціонарность ряду. Різними тестами виявляються наявність поодиноких коренів і порядок інтегрованості часового ряду (зазвичай обмежуються першим або другим порядком). Далі, при необхідності, (якщо порядок інтегрованості більше нуля) ряд перетворюється взяттям різниці відповідного порядку і вже для перетвореної моделі будується деяка ARMA-модель, оскільки передбачається, що отриманий процес є стаціонарним, на відміну від вихідного нестаціонарного процесу (разностно-стаціонарного або інтегрованого процесу порядку d).

ARIMA - модель і методологія аналізу часових рядів. Є розширенням моделей ARMA для нестаціонарних часових рядів, які можна зробити стаціонарними взяттям різниць деякого порядку від вихідного часового ряду (так звані інтегровані або різносно-стаціонарні часові ряди). Модель 𝐴𝑅𝐼𝑀𝐴(𝑝,𝑑,𝑞) означає, що різниці часового ряду порядку d належать моделі 𝐴𝑅𝑀𝐴(𝑝,𝑞).

Модель 𝐴𝑅𝐼𝑀𝐴(𝑝,𝑑,𝑞) для нестаціонарного часового ряда 𝑋𝑡 має вигляд:,

де – це зпрогнозоване значення на період 𝑡;

– константа, зазвичай для спрощення рівняється нулю;

– параметри моделі, коефіцієнти авторегресії;

– параметри моделі, ковзного середнього;

– оператор різниці часового ряду порядку 𝑑 (послідовне взяття 𝑑 раз різниць першого порядку - спочатку від тимчасового ряду, потім від отриманих різниць першого порядку, потім від другого порядку і т.д.)

– білий шум.

Також, дана модель інтерпретується як 𝐴𝑅𝑀𝐴(𝑝+𝑑,𝑞) модель з 𝑑 одиничними розв'язками. При 𝑑 = 0 маємо звичайну ARMA модель.

ARIMA-моделі дозволяють моделювати інтегровані або разностностаціонарні часові ряди (DS-ряди, diference stationary). Часовий ряд називається інтегрованим порядку 𝑘, якщо різниці ряду порядку ∆𝑘𝑥𝑡, тобто є стаціонарними, в той час як різниці меншого порядку (включаючи нульового порядку, тобто сам тимчасової ряд) не є стаціонарними щодо деякого тренда рядами (TS-рядами, trend stationary ). В зокрема 𝐼(0) – це стаціонарний процес. Порядок інтегрованості часового ряду і є порядок 𝑑 моделі.

**Методологія Бокса-Дженкінса.** Методологія Бокса-Дженкінса підбору ARIMA моделі для даного ряду спостережень складається з 5 кроків.

**Крок 1.** Отримання стаціонарного ряду. Ми тестуємо ряд на стаціонарність, використовуючи описані вище методи: візуальний аналіз графіка, візуальний аналіз ACF і PACF, тести на одиничні коріння. Якщо виходить стаціонарний ряд, то переходимо до наступного пункту, якщо немає стаціонарності ряду, то застосовуємо оператор взяття послідовної різниці і повторюємо тестування. Напрактиці послідовна різниця береться, як правило, не більше двох разів.

**Крок 2.** Після того, як отримано стаціонарний часовий ряд, будуються його вибіркові ACF і PACF, які, як було показано вище, є своєрідними «відбитками пальців» ARMA(p, q) процесу і дозволяють сформулювати кілька гіпотез про можливі порядки авторегресії 𝑝 і змінного середнього 𝑞. Зазвичай рекомендується використовувати моделі можливо більш низького порядку, як правило, з 𝑝 𝑞 3 (якщо немає сезонної компоненти).

**Крок 3.** Для кожної з обраних на першому етапі моделей оцінюються їх параметри і обчислюються залишки.

**Крок 4.** Кожна з моделей перевіряється, наскільки вона відповідає даним. З моделей, адекватних даними, вибирається найпростіша модель, тобто з найменшою кількістю параметрів.

**Крок 5.** Прогнозування. Після того, як обрана модель, можна будувати прогноз на один або кілька кроків за часом і оцінювати довірчі інтвервали прогнозованих значень.

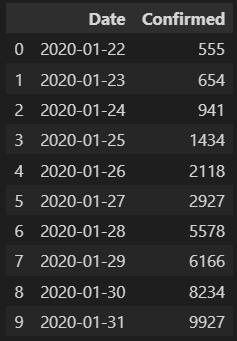
Модель S𝐴𝑅𝐼𝑀𝐴(𝑝,𝑑,𝑞, P, D, Q) - seasonal autoregressive integrated moving average, дуже схожа на модель *𝐴𝑅𝐼𝑀𝐴*(𝑝,𝑑,𝑞), за винятком того, що вона бере до уваги сезонність даних, з чого випливає додатковий набор компонентів авторегресії та ковзного середнього, що компенсуються частотою сезонності:

Модель S𝐴𝑅𝐼𝑀𝐴X(𝑝,𝑑,𝑞, r, P, D, Q), окрім сезонності, також враховує й екзогенні змінні, іншими словами, використовує зовнішні данні в прогнозі:

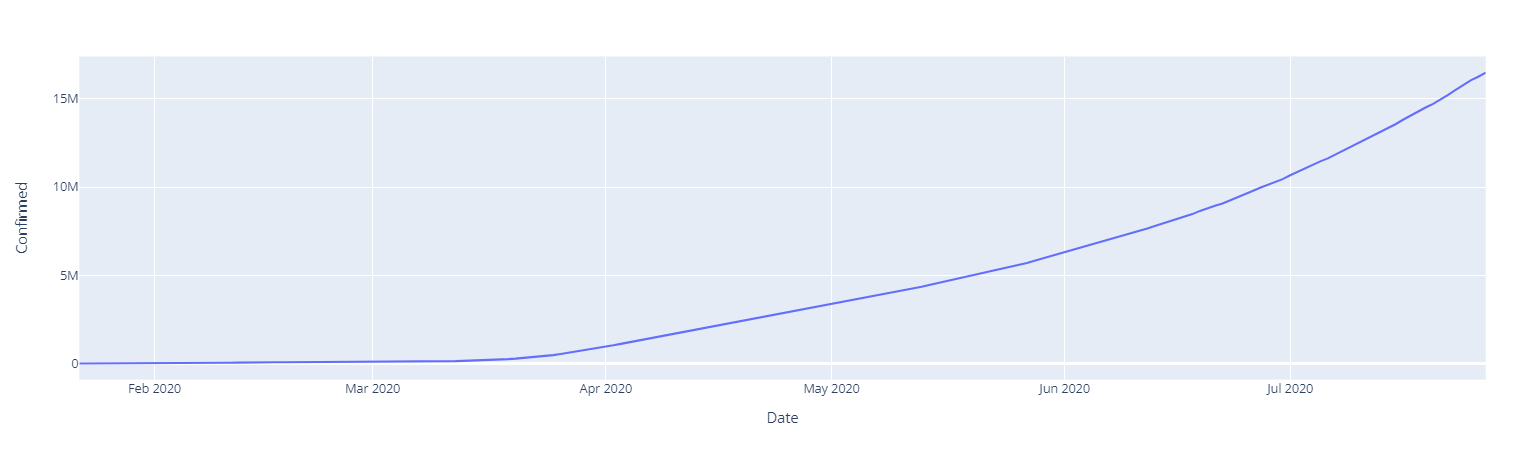
Контрольний приклад

Завантажимо дані про COVID-19 з файлу .csv. Залишивши в ньому тільки інформацію про дату та кількість хворих, отримуємо часовий ряд кількості хворих на COVID-19.

Розглянемо вхідні дані. Перші 10 записів файлу:

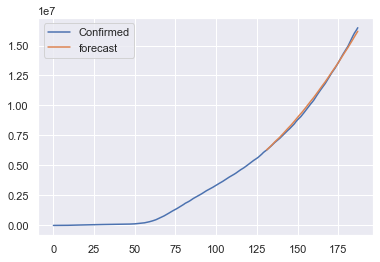


Побудова завантажених даних:



Для прогнозування кількості хворих на COVID-19 використаємо бібліотеку *statsmodels*, а саме модель *SARIMAX.* Розділимо дані на тренувальну та тестову вибірки з відношенням 70:30 та навчимо модель на тренувальних даних.

Результат прогноз моделі з урахуванням тестової виборки:



# Архітектура програмних засобів

Для забезпечення можливості прогнозування кількості хворих на COVID-19, була написана програма мовою Python, з використанням бібліотек:

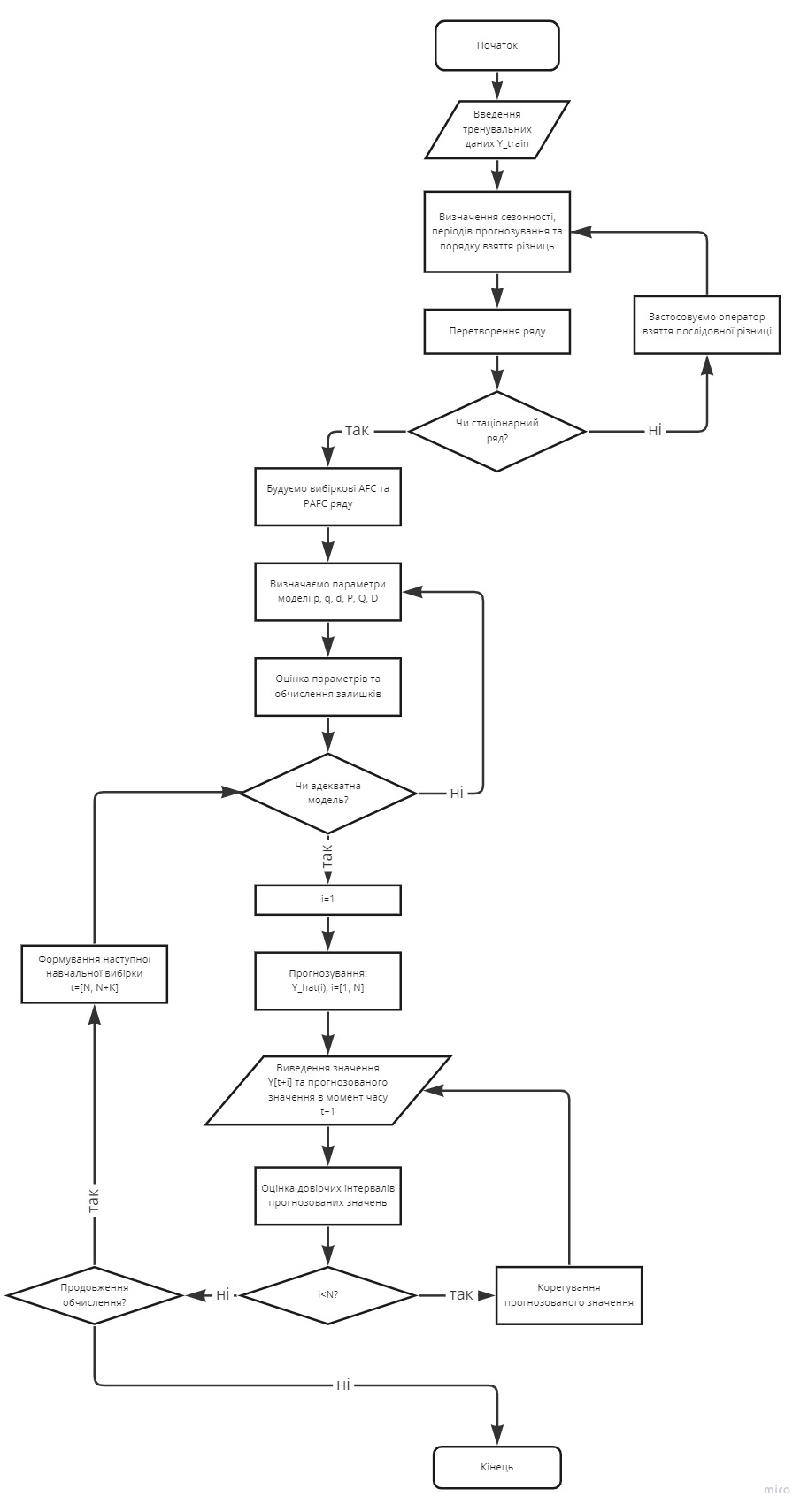
* pandas – зчитування даних формату .csv, та робота з дата фреймом;
* statsmodel – побудова моделі SARIMAX;
* plotly – побудова інтерактивних графіків.

# Формати вихідних даних

Вихідні дані: .csv файл, що містить інформацію про дату та кількість підтверджених випадків COVID-19, що відповідають цій даті.

Результат роботи програми: прогноз у вигляді .csv файлу, а також графік з фактичними та прогнозованими значеннями у форматі .png.

# Блок-схема алгоритму



# Код програми

import pandas as pd

import plotly.express as px

from statsmodels.tsa.statespace.sarimax import SARIMAX

from sklearn.metrics import mean\_squared\_log\_error

import plotly.graph\_objects as go

TRAINING\_SPLIT=0.7

def split(df):

    row\_number=df.shape[0]

    global TRAIN\_NUM

    TRAIN\_NUM=int(row\_number\*TRAINING\_SPLIT)

    df\_train=df.iloc[:TRAIN\_NUM, :]

    df\_test=df.iloc[TRAIN\_NUM:, :]

    return df\_train, df\_test

def losses(confirmed, df):

    actual=list(confirmed['Confirmed'])

    predcited=list(df['Forecast'])

    return mean\_squared\_log\_error(actual, predcited)

def plot\_results(df\_test, df):

    fig=go.Figure()

    fig.add\_trace(go.Scatter(

        x=df\_test['Date'],

        y=df\_test['Confirmed'],

        mode='lines',

        name='Actual values'

    ))

    fig.add\_trace(go.Scatter(

        x=df.iloc[TRAIN\_NUM:,0],

        y=df.iloc[TRAIN\_NUM:,2],

        mode='lines',

        name='Predicted values'

    ))

    fig.show()

def build\_model(df\_train):

    sarimax\_model=SARIMAX(df\_train['Confirmed'], order=(4, 2, 0), seasonal\_order=(0, 1, 1, 7))

    sarimax\_model\_fit=sarimax\_model.fit()

    return sarimax\_model\_fit

def sarimax\_predict(model, df\_test):

    predicted=pd.DataFrame()

    forecast\_test=model.forecast(len(df\_test))

    predicted['Date']=df\_test['Date']

    predicted['Forecast']=list(forecast\_test)

    return predicted

def sarimax(df):

    print(f'Raw Data:\n{df.head(10)}')

    fig=px.line(df, x='Date', y='Confirmed')

    fig.show()

    df\_train, df\_test=split(df)

    model=build\_model(df\_train)

    predicted=sarimax\_predict(model, df\_test)

    df['Forecast']=[None]\*TRAIN\_NUM+list(predicted['Forecast'])

    df.plot()

    print(f'Predicted Values: \n{predicted.head(10)}')

    plot\_results(df\_test, df)

    plot\_results(df, df)

    loss=losses(df\_test, predicted)

    print(f'Losses: Mean Squared Log Error: {losses}')

if \_\_name\_\_=='\_\_main\_\_':

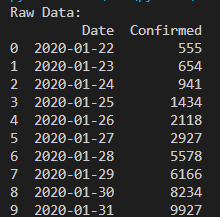
    df=pd.read\_csv('day\_wise.csv')

    covid\_df=df[['Date', 'Confirmed']].dropna()

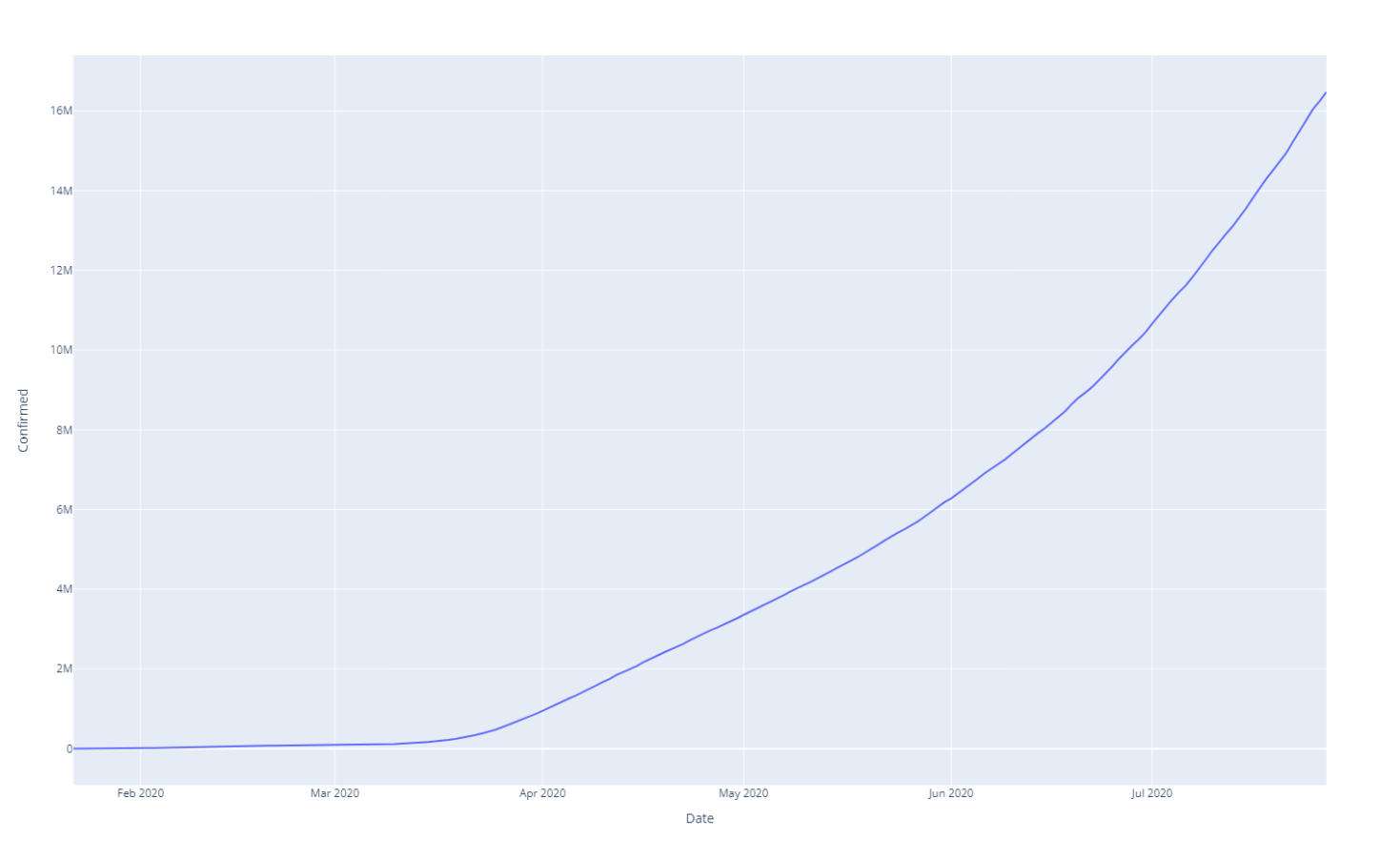
    sarimax(covid\_df)

# Опис результатів

Вхідні дані є .csv файл, що містить інформацію про дату та кількість підтверджених випадків COVID-19, що відповідають цій даті.

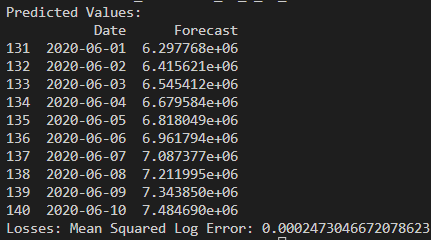


Візуалізація вхідних даних:

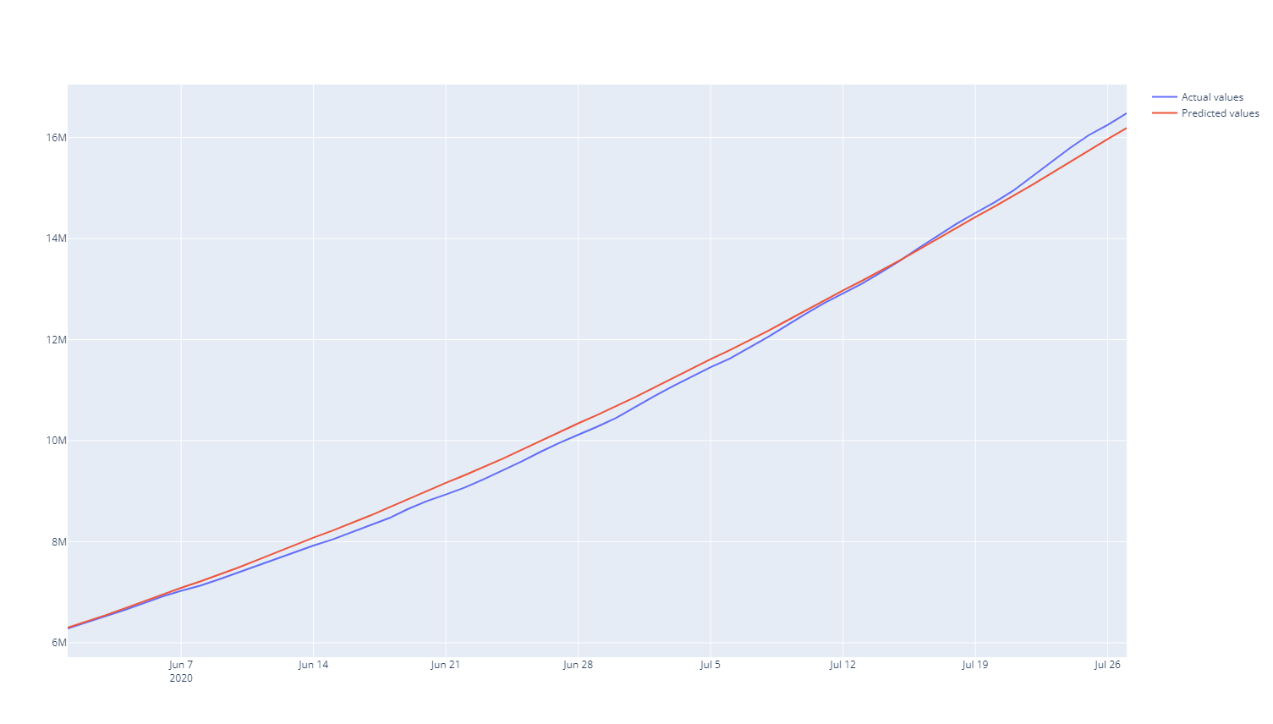


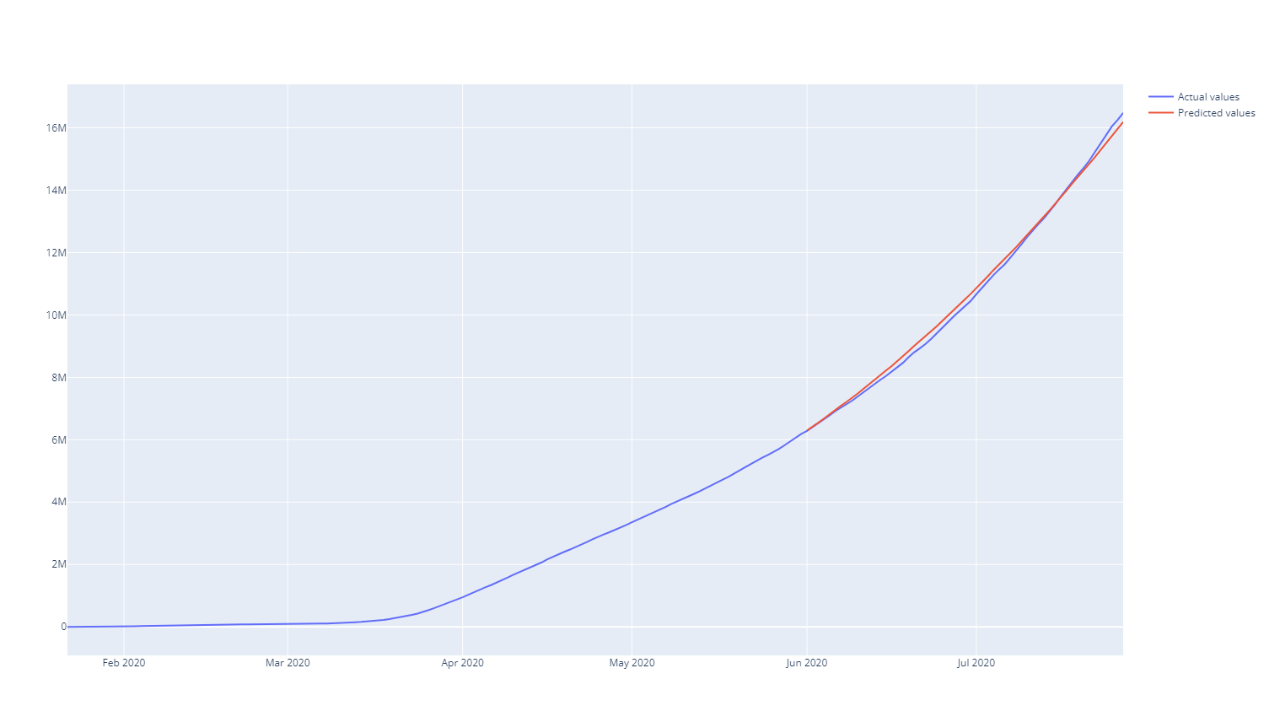
Розділимо вхідні дані на навчальну та тестову вибірки, де навчальна вибірка складає 70% від вхідного набору даних та використовується для навчання моделі SARIMAX, а тестова для перевірки результатів прогнозування.

В результаті прогнозування було отримано такі результати:



Похибка прогнозування складає 0,0002473 на тестових даних, що означає, що модель дає досить точні результати.

Візуалізація результатів прогнозування:



# Висновки

В ході курсової роботи було розглянуто підхід авторегресійного інтегрованого ковзного середнього та його варіацій до часових рядів, в результаті чого було створено, навчено модель сезонного авторегресійного інтегрованого ковзного середнього SARIMAX. Дана модель була застосована для прогнозування кількості хворих на COVID-19.

Тестування моделі дало похибку, що становить 0,0002473, яка означає, що модель дає достатньо точні результати для подальшого використання у майбутньому. Натомість, вхідні дані потрібно оновлювати, а модель навчати заново, підібравши нові параметри.

Результати роботи програми представлено графічно.