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 №5

Виконав:
студент групи КМ-93
Костенко О. А.

Керівник:
доцент
Олефір О. С.

Київ — 2022

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)
```

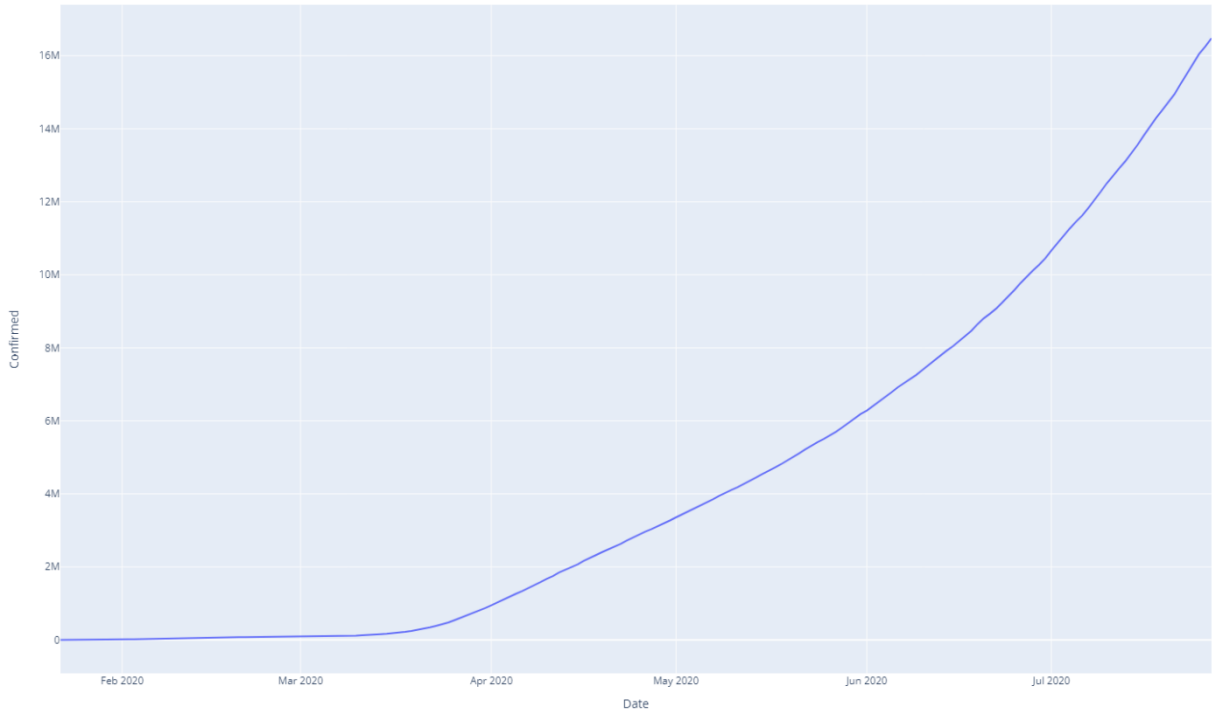
Опис результатів

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

Raw Data:

	Date	Confirmed
0	2020-01-22	555
1	2020-01-23	654
2	2020-01-24	941
3	2020-01-25	1434
4	2020-01-26	2118
5	2020-01-27	2927
6	2020-01-28	5578
7	2020-01-29	6166
8	2020-01-30	8234
9	2020-01-31	9927

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:

```
Predicted Values:
          Date      Forecast
131  2020-06-01  6.297768e+06
132  2020-06-02  6.415621e+06
133  2020-06-03  6.545412e+06
134  2020-06-04  6.679584e+06
135  2020-06-05  6.818049e+06
136  2020-06-06  6.961794e+06
137  2020-06-07  7.087377e+06
138  2020-06-08  7.211995e+06
139  2020-06-09  7.343850e+06
140  2020-06-10  7.484690e+06
Losses: Mean Squared Log Error: 0.0002473046672078623
```

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

Візуалізація результатів прогнозування: