STAGE 5 TEAM Task1

April 26, 2023

[8]:
```
!pip install dash
!pip install plotly
```

```
Requirement already satisfied: dash in c:\users\ashdh\anaconda3\lib\site-
packages (2.9.3)
Requirement already satisfied: dash-html-components==2.0.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (2.0.0)
Requirement already satisfied: dash-core-components==2.0.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (2.0.0)
Requirement already satisfied: plotly>=5.0.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (5.9.0)
Requirement already satisfied: dash-table==5.0.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (5.0.0)
Requirement already satisfied: Flask>=1.0.4 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (1.1.2)
Requirement already satisfied: Werkzeug>=0.15 in
c:\users\ashdh\anaconda3\lib\site-packages (from Flask>=1.0.4->dash) (2.0.3)
Requirement already satisfied: click>=5.1 in c:\users\ashdh\anaconda3\lib\site-
packages (from Flask>=1.0.4->dash) (8.0.4)
Requirement already satisfied: Jinja2>=2.10.1 in
c:\users\ashdh\anaconda3\lib\site-packages (from Flask>=1.0.4->dash) (2.11.3)
Requirement already satisfied: itsdangerous>=0.24 in
c:\users\ashdh\anaconda3\lib\site-packages (from Flask>=1.0.4->dash) (2.0.1)
Requirement already satisfied: tenacity>=6.2.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from plotly>=5.0.0->dash) (8.0.1)
Requirement already satisfied: colorama in c:\users\ashdh\anaconda3\lib\site-
packages (from click>=5.1->Flask>=1.0.4->dash) (0.4.5)
Requirement already satisfied: MarkupSafe>=0.23 in
c:\users\ashdh\anaconda3\lib\site-packages (from
Jinja2>=2.10.1->Flask>=1.0.4->dash) (2.0.1)
Requirement already satisfied: plotly in c:\users\ashdh\anaconda3\lib\site-
packages (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from plotly) (8.0.1)
```

[2]:
```
# Importing required libraries
import dash
#from jupyter_dash import JupyterDash
```

```
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objs as go
from datetime import date
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from scipy.optimize import curve_fit
```

[3]:
```
#reading the data for the confirmed number of COVID cases and displaying them
confirmedCasesNew = pd.read_csv(r"..\..\DATASETS\COVID␣
 ↪DATASETS\covid_confirmed_usafacts.csv")
a = confirmedCasesNew.iloc[:,4:].diff(axis=1).loc[:, '2022-05-30':'2023-01-01']
confirmedCases = pd.concat([confirmedCasesNew.iloc[:,:4], a], axis=1)
#confirmedCases
```

[4]:
```
#reading the data for the confirmed number of COVID cases and displaying them
confirmedCasesNew = pd.read_csv(r"..\..\DATASETS\COVID␣
 ↪DATASETS\covid_confirmed_usafacts.csv")
a = confirmedCasesNew.iloc[:,4:].diff(axis=1).loc[:, '2022-05-30':'2023-01-01']
confirmedCases = pd.concat([confirmedCasesNew.iloc[:,:4], a], axis=1)
confirmedCases



# reading the confirmed number of COVID Deaths and displaying them
confirmedDeathsNew = pd.read_csv(r"..\..\DATASETS\COVID␣
 ↪DATASETS\covid_deaths_usafacts.csv")
b = confirmedDeathsNew.iloc[:,4:].diff(axis=1).loc[:, '2022-05-30':'2023-01-01']
confirmedDeaths = pd.concat([confirmedDeathsNew.iloc[:,:4], b], axis=1)
```

[5]:
```
id_vars = ['countyFIPS', 'County Name', 'State', 'StateFIPS']
data1 = pd.melt(confirmedCases, id_vars= ['countyFIPS', 'County Name', 'State',␣
 ↪'StateFIPS'], var_name='date', value_name='cases')
data1 = data1[data1['County Name'] != 'Statewide Unallocated']

data2 = pd.melt(confirmedDeaths , id_vars= ['countyFIPS', 'County Name',␣
 ↪'State', 'StateFIPS'], var_name='date', value_name='deaths')
data2 = data2[data2['County Name'] != 'Statewide Unallocated']
data2

data = data1.merge(data2, on=['countyFIPS', 'County Name', 'State',␣
 ↪'StateFIPS', 'date'])
```

[6]:
```
#creating log values
data['log_cases']=np.log(data['cases'])
data['log_deaths']=np.log(data['deaths'])

#making the date column as a date type
data['date']=data['date'].astype('datetime64[ns]')

#removing all negative and non numeric values from the cases and deaths
data['cases'] = data['cases'].clip(lower=0)
data['deaths'] = data['deaths'].clip(lower=0)
data['log_cases'] = data['log_cases'].clip(lower=0)
data['log_deaths'] = data['log_deaths'].clip(lower=0)

data
```

```
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning: invalid value encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning: invalid value encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

[6]:

	countyFIPS	County Name	State	StateFIPS	date	cases	deaths	log_cases	log_deaths
0	1001	Autauga County	AL	1	2022-05-30	9	0	2.197225	0.0
1	1003	Baldwin County	AL	1	2022-05-30	55	1	4.007333	0.0
2	1005	Barbour County	AL	1	2022-05-30	1	0	0.000000	0.0
3	1007	Bibb County	AL	1	2022-05-30	9	0	2.197225	0.0
4	1009	Blount County	AL	1	2022-05-30	6	0	1.791759	0.0
…	…	…	…	…	…	…	…	…	…
681809	56037	Sweetwater County	WY	56	2023-01-01	0	0		
681810	56039	Teton County	WY	56	2023-01-01	0	0		
681811	56041	Uinta County	WY	56	2023-01-01	0	0		
681812	56043	Washakie County	WY	56	2023-01-01	0	0		
681813	56045	Weston County	WY	56	2023-01-01	0	0		

```
681809       0   0.000000         0.0
681810       0   0.000000         0.0
681811       0   0.000000         0.0
681812       0   0.000000         0.0
681813       0   0.000000         0.0

[681814 rows x 9 columns]
```

[7]:
```
data.info(5)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 681814 entries, 0 to 681813
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   countyFIPS   681814 non-null  int64
 1   County Name  681814 non-null  object
 2   State        681814 non-null  object
 3   StateFIPS    681814 non-null  int64
 4   date         681814 non-null  datetime64[ns]
 5   cases        681814 non-null  int64
 6   deaths       681814 non-null  int64
 7   log_cases    679581 non-null  float64
 8   log_deaths   680011 non-null  float64
dtypes: datetime64[ns](1), float64(2), int64(4), object(2)
memory usage: 52.0+ MB
```

[8]:
```
#ORIGINAL
app = dash.Dash(__name__)
df=data
states = data['State'].unique().tolist()
states.insert(0,"All States")
# Create dropdown options
dropdown_options = [{'label': state, 'value': state} for state in states]


app.layout = html.Div([
            html.H1("Covid-19 trend lines",style={'text-align':'center'}),
            dcc.DatePickerRange(
                id='my-date-picker-range',
                min_date_allowed=date(2022, 5, 30),
                max_date_allowed=date(2023, 1, 1),
                initial_visible_month=date(2022, 6, 1),
                start_date_placeholder_text='Start Date',
                end_date_placeholder_text='End Date',
                display_format='MM/DD/YYYY',
                month_format='MMMM, YYYY',
                start_date=date(2022, 6, 1),
```

```
            end_date=date(2023, 1, 1)
        ),
html.Br(),

#SETTING A DROP DOWN FOR STATE SELECTION
html.Div([
    html.Label('Select State:'),
    dcc.Dropdown(
        id='state-dropdown',
        options=dropdown_options,
        #multi=True,
        value="All States"  # Set default value
        )]),
html.Br(),
#SETTING TOGGLE BUTTON FOR CASES/DEATHS
 html.Div([
        html.Label("Cases Or Deaths:"),
        dcc.RadioItems(id="CorD",
            options=[
                {"label": "Cases", "value": "cases"},
                {"label": "Deaths", "value": "deaths"}
            ],
            value="cases", labelStyle={"display":"block"})]),
html.Br(),

#SETTING TOGGLE FOR LINEAR/LOG
html.Div([
html.Div([
        html.Label("Linear Or Log:"),
        dcc.RadioItems(id="LOrLog",
            options=[
                {"label": "Linear", "value": "linear"},
                {"label": "Log", "value": "log"}
            ],
            value="linear", labelStyle={"display":"block"})],
    style={'display': 'inline-block', 'margin-right': '20px'}),
html.Br(),

#SETTING CHECKLIST FOR LINEAR/NON-LINEAR REGRESSION LINE
html.Div([
        html.Label("Regression: "),
        dcc.Checklist(id='regression-checklist',
            options=[
                {'label': 'Linear Regression', 'value': 'linearReg'},
                {'label': 'Nonlinear Regression', 'value': 'nonlinearReg'}
            ],
            value=[],
```

```
                inputClassName='toggle-radio',
                inputStyle={'margin-right': '10px'}
                        )
    ],
    style={'display': 'inline-block', 'margin-right': '20px'})
    ],
    style={'display': 'flex', 'flex-direction': 'row'}),
    dcc.Graph(id='Trend_Plot')
    ])



#NON REGRESSION PART
@app.callback(
    Output(component_id='Trend_Plot', component_property='figure'),
    [Input('my-date-picker-range', 'start_date'),
    Input('my-date-picker-range', 'end_date'),
    Input('CorD', 'value'),
    Input('LOrLog', 'value'),
    Input('state-dropdown', 'value'),
    Input('regression-checklist', 'value')]
)
def updated_graph(start_date, end_date, CorD_value, LorLog_value,␣
 ↪selected_state,regression):
    df_state = df[df['State'] == selected_state]
    df_state = df_state.sort_values(by='date')
    df_state['log_cases'] = np.log(df_state['cases'])
    df_state['log_deaths'] = np.log(df_state['deaths'])
    if selected_state == "All States":
        df_state = data.groupby("date").sum().reset_index()
    else:
        df_state = data[data['State']== selected_state].groupby("date").sum().
 ↪reset_index()
        '''dfs = {}
        for state in selected_state:
            dfs[state] = data[data['State'] == state].sort_values(by='date').
 ↪groupby("date").sum().reset_index()
            #dfs[state] = data[data['State'].isin(selected_state)].
 ↪groupby("date").sum().reset_index()
        df_state = pd.concat(dfs)
        '''

    if LorLog_value == 'linear':
        if start_date is not None:
            df_1 = df_state[(df_state['date'] >= start_date) &␣
 ↪(df_state['date'] <= end_date)]
        else:
```

```
            df_1 = df_state

        if CorD_value == 'cases':
            fig_c = px.scatter(df_1, x='date', y='cases', color='cases')
        elif CorD_value == 'deaths':
            fig_c = px.scatter(df_1, x='date', y='deaths', color='deaths')

    elif LorLog_value == 'log':
        if start_date is not None:
            df_1 = df_state[(df_state['date'] >= start_date) &␣
↪(df_state['date'] <= end_date)]
        else:
            df_1 = df_state

        if CorD_value == 'cases':
            fig_c = px.scatter(df_1, x='date', y='log_cases', color='log_cases')
        elif CorD_value == 'deaths':
            fig_c = px.scatter(df_1, x='date', y='log_deaths',␣
↪color='log_deaths')

    # Add regression lines if selected
    if "linearReg" in regression:
        X = np.array(df_1.index).reshape(-1, 1)
        y = np.array(df_1[CorD_value])
        model = LinearRegression()
        model.fit(X, y)
        y_pred = model.predict(X)
        fig_c.add_trace(go.Line(x=df_1['date'], y=y_pred, mode='lines',␣
↪name='Linear Regression'))

    if "nonlinearReg" in regression:
        # Define a nonlinear regression function here
        X = np.array(df_1.index).reshape(-1, 1)
        y = np.array(df_1[CorD_value])
        poly = PolynomialFeatures(degree=4)
        X_poly = poly.fit_transform(X)
        model = LinearRegression()
        model.fit(X_poly, y)
        y_pred = model.predict(X_poly)
        fig_c.add_trace(go.Line(x=df_1['date'], y=y_pred, mode='lines',␣
↪name='Polynomial Regression'))


    fig_c.update_traces(mode='lines+markers')
    fig_c.update_layout(
        title=f"{selected_state} COVID-19 {CorD_value.capitalize()} Trend␣
↪({start_date} to {end_date})",
```

```
        xaxis_title="Date",
        yaxis_title=f"{CorD_value.capitalize()}",
        legend_title=f"{CorD_value.capitalize()}",
    )

    return fig_c

# Run the app
if __name__ == '__main__':
    app.run_server(debug=False)
```

```
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production
deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
127.0.0.1 - - [26/Apr/2023 18:06:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:06:51] "GET /_dash-layout HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:06:51] "GET /_dash-dependencies HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:06:51] "GET /_dash-component-
suites/dash/dcc/async-datepicker.js HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:06:51] "GET /_dash-component-
suites/dash/dcc/async-dropdown.js HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:06:51] "GET /_dash-component-
suites/dash/dcc/async-graph.js HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:06:51] "GET /_dash-component-
suites/dash/dcc/async-plotlyjs.js HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:06:52] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:07:03] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:07:49] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:09:20] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:09:22] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
```

```
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:09:27] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:09:30] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:09:42] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:09:45] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:09:57] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:09:59] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:10:00] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:10:41] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:10:42] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:10:44] "POST /_dash-update-component HTTP/1.1" 200
```

```
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:10:48] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:10:51] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

127.0.0.1 - - [26/Apr/2023 18:10:56] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:11:00] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:11:42] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


127.0.0.1 - - [26/Apr/2023 18:13:37] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.
```

```
C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


127.0.0.1 - - [26/Apr/2023 18:13:38] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


127.0.0.1 - - [26/Apr/2023 18:13:41] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
```

```
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


127.0.0.1 - - [26/Apr/2023 18:13:43] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


127.0.0.1 - - [26/Apr/2023 18:13:51] "POST /_dash-update-component HTTP/1.1" 200
-
C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.


C:\Users\ashdh\anaconda3\lib\site-
packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.
```

```
127.0.0.1 - - [26/Apr/2023 18:14:16] "POST /_dash-update-component HTTP/1.1" 200
-
```

[ ]: