Multivariate_Vector_Autogression_VAR_Time_Series

September 5, 2020

```
In [1]: # IMPORT LIBRARIES
        import pandas as pd
        import numpy as np
        import sys
        import statsmodels.api as sm
        from statsmodels.tsa.api import VAR
        from sklearn import preprocessing

        pd.set_option("display.max_columns", None)
        pd.set_option("display.max_rows", None)
```

```
In [2]: # IMPORT DAILY CASE DATA
        str_file = 'daily.csv'
        df_cases = pd.read_csv(str_file)
        df_cases.head()
```

```
Out[2]:        date state  positive  negative  pending  hospitalizedCurrently  \
        0  20200607    AK     544.0   64360.0      NaN                    7.0
        1  20200607    AL   20500.0  239066.0      NaN                    NaN
        2  20200607    AR    9426.0  150847.0      NaN                  145.0
        3  20200607    AS       0.0     174.0      NaN                    NaN
        4  20200607    AZ   26889.0  254732.0      NaN                 1252.0

           hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
        0                     NaN             NaN              NaN
        1                  2022.0             NaN            615.0
        2                   844.0             NaN              NaN
        3                     NaN             NaN              NaN
        4                  3352.0           392.0              NaN

           onVentilatorCurrently  onVentilatorCumulative  recovered dataQualityGrade  \
        0                    1.0                     NaN      382.0                B
        1                    NaN                   364.0    11395.0                B
        2                   35.0                   143.0     6424.0                A
        3                    NaN                     NaN        NaN                C
        4                  248.0                     NaN     5517.0               A+
             lastUpdateEt          dateModified  checkTimeEt   death  hospitalized  \
        0  6/7/2020 00:00  2020-06-07T00:00:00Z  06/06 20:00    10.0           NaN
        1  6/7/2020 11:00  2020-06-07T11:00:00Z  06/07 07:00   692.0        2022.0
        2  6/7/2020 16:10  2020-06-07T16:10:00Z  06/07 12:10   154.0         844.0
        3  6/1/2020 00:00  2020-06-01T00:00:00Z  05/31 20:00     0.0           NaN
        4  6/7/2020 00:00  2020-06-07T00:00:00Z  06/06 20:00  1044.0        3352.0

                    dateChecked  fips  positiveIncrease  negativeIncrease   total  \
        0  2020-06-07T00:00:00Z     2                 8               995   64904
        1  2020-06-07T11:00:00Z     1               457             13465  259566
        2  2020-06-07T16:10:00Z     5               325              3191  160273
        3  2020-06-01T00:00:00Z    60                 0                 0     174
        4  2020-06-07T00:00:00Z     4              1438              8537  281621

           totalTestResults  totalTestResultsIncrease  posNeg  deathIncrease  \
        0             64904                      1003   64904              0
        1            259566                     13922  259566              3
        2            160273                      3516  160273              0
        3               174                         0     174              0
        4            281621                      9975  281621              2

           hospitalizedIncrease                                      hash  \
        0                   -48  62adbd451838656b7df7519e830d6439be0b5877
        1                    29  9040674078ce6afca363f8e95943845a032ab5d6
        2                     6  ef23d4d3f9e232bb5f58a59d79a27d2cb0797e2a
        3                     0  893135d0d7a9340a91aca139f4e3bb289f418f71
        4                    32  505a05efa5a9b912644a7ad16b2ab6f37330806b

           commercialScore  negativeRegularScore  negativeScore  positiveScore  score  \
        0                0                     0              0              0      0
        1                0                     0              0              0      0
        2                0                     0              0              0      0
        3                0                     0              0              0      0
        4                0                     0              0              0      0

           grade
        0    NaN
        1    NaN
        2    NaN
        3    NaN
        4    NaN
```

In [3]:

```
# FILTER DAILY CASE DATA FOR ONLY NECESSARY COLUMNS
lst_columns_to_keep = ['date', 'positiveIncrease']
df_cases = df_cases[lst_columns_to_keep]
df_cases.head()
```

Out[3]:

```
       date  positiveIncrease
0  20200607                 8
```

```
        1  20200607                457
        2  20200607                325
        3  20200607                  0
        4  20200607               1438
```

```
In [4]: # MAKE DATE DATETIME
        df_cases["date"] = pd.to_datetime(df_cases["date"], format='%Y%m%d')
        df_cases.head()
```

```
Out[4]:         date  positiveIncrease
        0 2020-06-07                 8
        1 2020-06-07               457
        2 2020-06-07               325
        3 2020-06-07                 0
        4 2020-06-07              1438
```

```
In [5]: # SUM DATA FOR ALL OF USA
        df_cases = df_cases.groupby(['date'])['positiveIncrease'].sum()
        df_cases.tail()
```

```
Out[5]: date
        2020-06-03    20063
        2020-06-04    20540
        2020-06-05    28392
        2020-06-06    23062
        2020-06-07    19932
        Name: positiveIncrease, dtype: int64
```

```
In [6]: # IMPORT WEATHER DATA
        df_weather = pd.read_excel('external_datasets.xlsx', sheet_name='average_US_temperature
        df_weather["date"] = pd.to_datetime(df_weather["date"], format='%Y%m%d')
        df_weather.head()
```

```
Out[6]:         date  temperature
        0 2020-11-30         43.5
        1 2020-11-29         43.5
        2 2020-11-28         43.5
        3 2020-11-27         43.5
        4 2020-11-26         43.5
```

```
In [7]: # IMPORT BORDER CLOSING DATA
        df_border = pd.read_excel('external_datasets.xlsx', sheet_name='border_closing_2020')
        df_border["date"] = pd.to_datetime(df_border["date"], format='%Y%m%d')
        df_border.head()
```

```
Out[7]:         date  border_closed
        0 2020-11-30              0
        1 2020-11-29              0
        2 2020-11-28              0
        3 2020-11-27              0
        4 2020-11-26              0
```

```
In [8]: # IMPORT NUMBER OF TOURISTS
        df_tourists = pd.read_excel('external_datasets.xlsx', sheet_name='tourists_2020')
        df_tourists["date"] = pd.to_datetime(df_tourists["date"], format='%Y%m%d')
        df_tourists.head()
```

```
Out[8]:         date  tourists
        0 2020-11-30     30859
        1 2020-11-29     28840
        2 2020-11-28     26953
        3 2020-11-27     25190
        4 2020-11-26     23542
```

```
In [9]: # MAKE A LIST OF ALL DATES TO FORECAST
        from datetime import date, timedelta

        date_start = date(2020, 6, 8)
        date_end = date(2020, 11, 30)

        int_days_in_between = date_end - date_start

        lst_dates = []

        for i in range(int_days_in_between.days + 1):
            day = date_start + timedelta(days=i)
            lst_dates.append(str(day))
```

```
In [10]: # MODEL
         # manual loop to make the model multi-step
         for date in lst_dates:
             # merge with weather data
             df = pd.merge(
                 df_cases,
                 df_weather,
                 left_on=["date"],
                 right_on=["date"],
                 how="left",
             )
             # merge with border closing data
             df = pd.merge(
                 df,
                 df_border,
                 left_on=["date"],
                 right_on=["date"],
                 how="left",
             )

             # error in the model with this data which is why it is commented out
             # merge with tourist data
         #     df = pd.merge(
         #         df,
         #         df_tourists,
         #         left_on=["date"],
         #         right_on=["date"],
         #         how="left",
         #     )

             # normalize the data
             df = df.set_index('date')
             x = df[['positiveIncrease', 'border_closed']] #, 'tourists']].values #returns a n
             min_max_scaler = preprocessing.MinMaxScaler()
             df_scaled = min_max_scaler.fit_transform(x)

             # fitting the model
             model = VAR(df_scaled)#, freq='D')
             results = model.fit(maxlags=15, ic='aic') # maxlag of 15 is an arbitrary number

             # forecasting
             lag_order = results.k_ar
         #     forecast = results.forecast(df.values[-lag_order:], 1) # lag_order is the recom
             forecast = results.forecast(df_scaled[-15:], 1) # 15 is arbitrary, would need to

             # add the previous forecast to the dataset as a new observation
             df_cases.loc[date] = round(max(min_max_scaler.inverse_transform(forecast)[0][0],
             df_cases.index = pd.to_datetime(df_cases.index)

         df_cases.tail(10)
```

```
Out[10]: date
         2020-11-21     96296.0
         2020-11-22     98912.0
         2020-11-23    101937.0
         2020-11-24    104292.0
         2020-11-25    105500.0
         2020-11-26    105908.0
         2020-11-27    106379.0
         2020-11-28    107763.0
         2020-11-29    110179.0
         2020-11-30    112946.0
         Name: positiveIncrease, dtype: float64
```

```
In [11]: # ALL FORECASTS
         pd.options.display.float_format = '{:,}'.format
         print('all forecasts for daily new cases')
         df_cases[date_start:date_end]
```

```
all forecasts for daily new cases
```

```
Out[11]: date
         2020-06-08    21,260.0
         2020-06-09    18,544.0
         2020-06-10    19,888.0
         2020-06-11    24,674.0
         2020-06-12    22,902.0
         2020-06-13    23,015.0
         2020-06-14    22,279.0
         2020-06-15    17,909.0
         2020-06-16    18,693.0
         2020-06-17    20,890.0
         2020-06-18    21,184.0
         2020-06-19    23,168.0
         2020-06-20    22,745.0
         2020-06-21    19,419.0
         2020-06-22    18,575.0
         2020-06-23    18,113.0
         2020-06-24    18,465.0
         2020-06-25    21,027.0
         2020-06-26    21,738.0
         2020-06-27    20,532.0
         2020-06-28    19,221.0
         2020-06-29    17,163.0
         2020-06-30    16,360.0
         2020-07-01    17,779.0
         2020-07-02    19,008.0
         2020-07-03    19,694.0
         2020-07-04    19,319.0
         2020-07-05    17,257.0
         2020-07-06    15,468.0
         2020-07-07    15,033.0
         2020-07-08    15,549.0
         2020-07-09    16,873.0
         2020-07-10    17,660.0
         2020-07-11    16,809.0
         2020-07-12    15,130.0
         2020-07-13    13,435.0
         2020-07-14    12,563.0
         2020-07-15    13,122.0
         2020-07-16    14,201.0
         2020-07-17    14,630.0
         2020-07-18    13,982.0
         2020-07-19    12,277.0
         2020-07-20    10,451.0
         2020-07-21     9,618.0
         2020-07-22     9,865.0
         2020-07-23    10,630.0
         2020-07-24    11,007.0
```

2020-07-25	10,218.0
2020-07-26	8,483.0
2020-07-27	6,693.0
2020-07-28	5,632.0
2020-07-29	5,623.0
2020-07-30	6,162.0
2020-07-31	6,271.0
2020-08-01	5,383.0
2020-08-02	3,619.0
2020-08-03	1,690.0
2020-08-04	451.0
2020-08-05	166.0
2020-08-06	357.0
2020-08-07	225.0
2020-08-08	0.0
2020-08-09	0.0
2020-08-10	0.0
2020-08-11	0.0
2020-08-12	0.0
2020-08-13	0.0
2020-08-14	270.0
2020-08-15	388.0
2020-08-16	240.0
2020-08-17	166.0
2020-08-18	150.0
2020-08-19	147.0
2020-08-20	239.0
2020-08-21	296.0
2020-08-22	293.0
2020-08-23	278.0
2020-08-24	141.0
2020-08-25	11.0
2020-08-26	4.0
2020-08-27	31.0
2020-08-28	55.0
2020-08-29	44.0
2020-08-30	0.0
2020-08-31	0.0
2020-09-01	0.0
2020-09-02	0.0
2020-09-03	0.0
2020-09-04	0.0
2020-09-05	0.0
2020-09-06	0.0
2020-09-07	0.0
2020-09-08	0.0
2020-09-09	0.0
2020-09-10	0.0

```
2020-09-11         0.0
2020-09-12         0.0
2020-09-13         0.0
2020-09-14         0.0
2020-09-15         0.0
2020-09-16         0.0
2020-09-17         0.0
2020-09-18         0.0
2020-09-19         0.0
2020-09-20         0.0
2020-09-21         0.0
2020-09-22         0.0
2020-09-23         0.0
2020-09-24         0.0
2020-09-25         0.0
2020-09-26         0.0
2020-09-27         0.0
2020-09-28         0.0
2020-09-29         0.0
2020-09-30         0.0
2020-10-01         0.0
2020-10-02         0.0
2020-10-03         0.0
2020-10-04         0.0
2020-10-05     7,905.0
2020-10-06     6,347.0
2020-10-07     6,190.0
2020-10-08     9,311.0
2020-10-09     8,190.0
2020-10-10     9,936.0
2020-10-11    15,294.0
2020-10-12    19,627.0
2020-10-13    23,212.0
2020-10-14    24,557.0
2020-10-15    23,196.0
2020-10-16    24,761.0
2020-10-17    27,620.0
2020-10-18    30,881.0
2020-10-19    35,614.0
2020-10-20    38,843.0
2020-10-21    39,662.0
2020-10-22    40,051.0
2020-10-23    40,490.0
2020-10-24    42,438.0
2020-10-25    46,707.0
2020-10-26    50,597.0
2020-10-27    53,297.0
2020-10-28    54,748.0
```

```
2020-10-29    54,797.0
2020-10-30    55,265.0
2020-10-31    57,456.0
2020-11-01    60,768.0
2020-11-02    64,542.0
2020-11-03    67,462.0
2020-11-04    68,509.0
2020-11-05    68,752.0
2020-11-06    69,368.0
2020-11-07    71,109.0
2020-11-08    74,289.0
2020-11-09    77,832.0
2020-11-10    80,388.0
2020-11-11    81,648.0
2020-11-12    81,956.0
2020-11-13    82,400.0
2020-11-14    84,113.0
2020-11-15    86,998.0
2020-11-16    90,199.0
2020-11-17    92,731.0
2020-11-18    93,945.0
2020-11-19    94,265.0
2020-11-20    94,789.0
2020-11-21    96,296.0
2020-11-22    98,912.0
2020-11-23   101,937.0
2020-11-24   104,292.0
2020-11-25   105,500.0
2020-11-26   105,908.0
2020-11-27   106,379.0
2020-11-28   107,763.0
2020-11-29   110,179.0
2020-11-30   112,946.0
Name: positiveIncrease, dtype: float64
```

```
In [ ]:
```