BUA 451 Final Project

Yiming TONG | Business Data Analytics & Finance, Syracuse University
Dataset: bigquery-public-data.covid19_nyt.us_counties
Date: 2025-04-29

1. Executive Summary

This project focuses on the county-level COVID-19 epidemic in the United States, extracts and analyzes the "cumulative confirmed cases" and "daily new" trends from the public BigQuery dataset, and then constructs a binary classification model to predict whether the next day will enter the "high incidence" state.

** - Business problem ** : Help public health departments prioritize medical resources and provide early warning of new spikes.

** - Key findings - ** :

1. As of the latest date, the five most affected counties are Los Angeles, New York City, Cook, Miami-Dade, Maricopa.
2. The new peak in Los Angeles County is mainly concentrated in July 2020 and January 2021.

** Model performance ** : ~74% precision and 70% + recall on the test set based on Logistic Regression with data added in the last 7 days.

** Management implications ** : This can be used to dynamically deploy care and supplies and to develop intervention strategies in advance of high-risk periods.

2. Dataset Description

Data Source: Google BigQuery Public Dataset

Table ID: bigquery-public-data.covid19_nyt.us_counties

Origin: The New York Times COVID-19 repository (county-level daily counts of cases and deaths)

Created: April 9, 2020 Last Modified: April 28, 2025

Location: US (no table expiration)

Schema

Column Type Description

-date DATE Report date

-county STRING County name

-state_name STRING State name

-county_fips_code STRING FIPS geographic identifier for the county

-confirmed_cases INTEGER Cumulative number of confirmed COVID-19 cases to date

-deaths INTEGER Cumulative number of confirmed COVID-19 deaths to date

- Table Description: County-level time series of COVID-19 confirmed cases and deaths published by The New York Times (source: https://github.com/nytimes/covid-19-data).
- Record Count: ~2.7 million rows (all U.S. counties, daily from 2020-01-21 through 2025-04-28).

This dataset supports both exploratory analysis and time-series predictive modeling.

```
!pip install pandas-gbq --quiet
!pip install google-cloud-bigquery pandas
```

```
from google.colab import auth
auth.authenticate_user()
```

```
from google.cloud import bigquery
client = bigquery.Client(project = 'silver-harmony-457719-s1')
```

```
import pandas as pd
from pandas.io import gbq
import matplotlib.pyplot as plt
import seaborn as sns
from google.cloud import bigquery
```

```
!pip install plotly ipywidgets --quiet

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.io as pio
```

```
pio.renderers.default = "colab"
```

```
project_id = 'bigquery-public-data.covid19_nyt.us_counties'
client = bigquery.Client(project = 'bigquery-public-data.covid19_nyt.us_counties'
```

3. EDA Results and Visuals

3.1 - Insight 1: Top 5 Counties for “Cumulative Number of Confirmed Diagnoses” by Latest Date

Business Implications: Helps decision makers quickly target counties with the worst outbreaks and most need for resource investment.

retrieve data

```
query1 = """
WITH latest AS (
  SELECT MAX(date) AS max_date
  FROM `bigquery-public-data.covid19_nyt.us_counties`
)
SELECT
  county,
  state_name AS state,
  confirmed_cases
FROM `bigquery-public-data.covid19_nyt.us_counties` AS t
JOIN latest AS l
  ON t.date = l.max_date
ORDER BY confirmed_cases DESC
LIMIT 5;
"""
df_top5 = client.query(query1).to_dataframe()
print(df_top5)
```

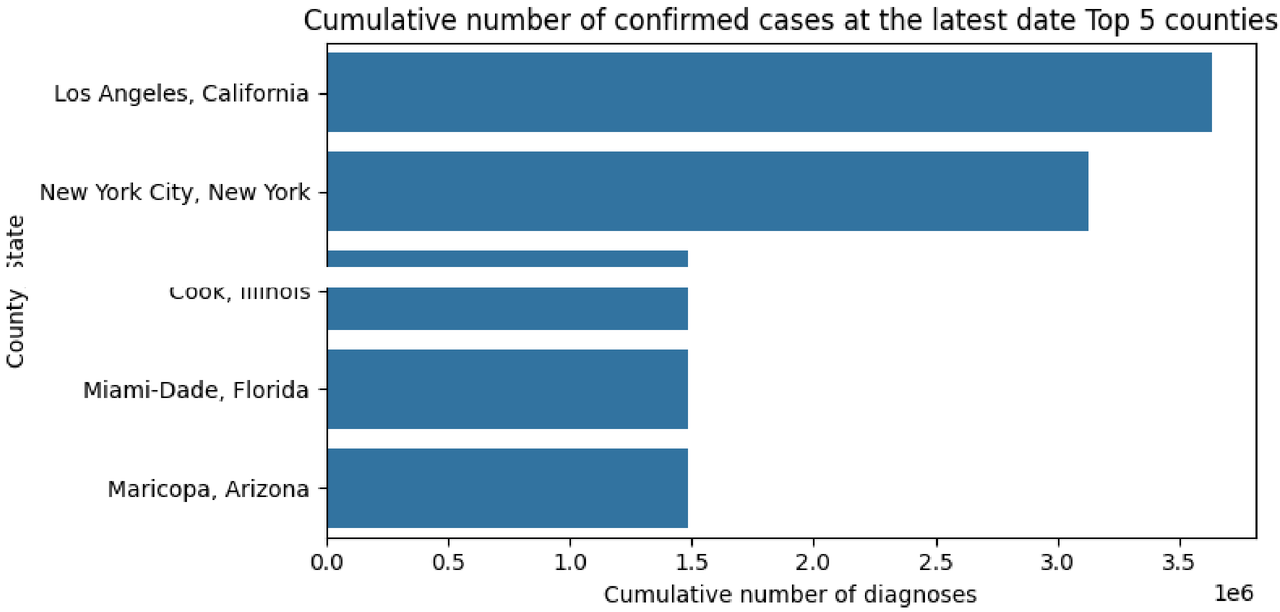
```
          county       state  confirmed_cases
0    Los Angeles  California          3632440
1  New York City    New York          3126782
2           Cook    Illinois          1487242
3     Miami-Dade     Florida          1487115
4       Maricopa     Arizona          1484296
```

Drawing

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8,4))
sns.barplot(
    data=df_top5,
    x='confirmed_cases',
```

```
    y=df_top5['county'] + ', ' + df_top5['state']
)
plt.title('Cumulative number of confirmed cases at the latest date Top 5 counties
plt.xlabel('Cumulative number of diagnoses')
plt.ylabel('County, State')
plt.tight_layout()
plt.show()
```



3.2 - Insight 2: Daily Trends in New Diagnoses in Los Angeles County

Business Implications: Focus on monitoring additions in high-risk counties to provide early warning of medical and material deployment needs.

```
query2 = """
SELECT
  date,
  confirmed_cases
FROM `bigquery-public-data.covid19_nyt.us_counties`
WHERE county='Los Angeles'
  AND state_name='California'
ORDER BY date
"""
df_la = client.query(query2).to_dataframe()

# Calculate "new diagnoses per day"
df_la['new_cases'] = df_la['confirmed_cases'].diff().fillna(0).astype(int)

print(df_la)
```

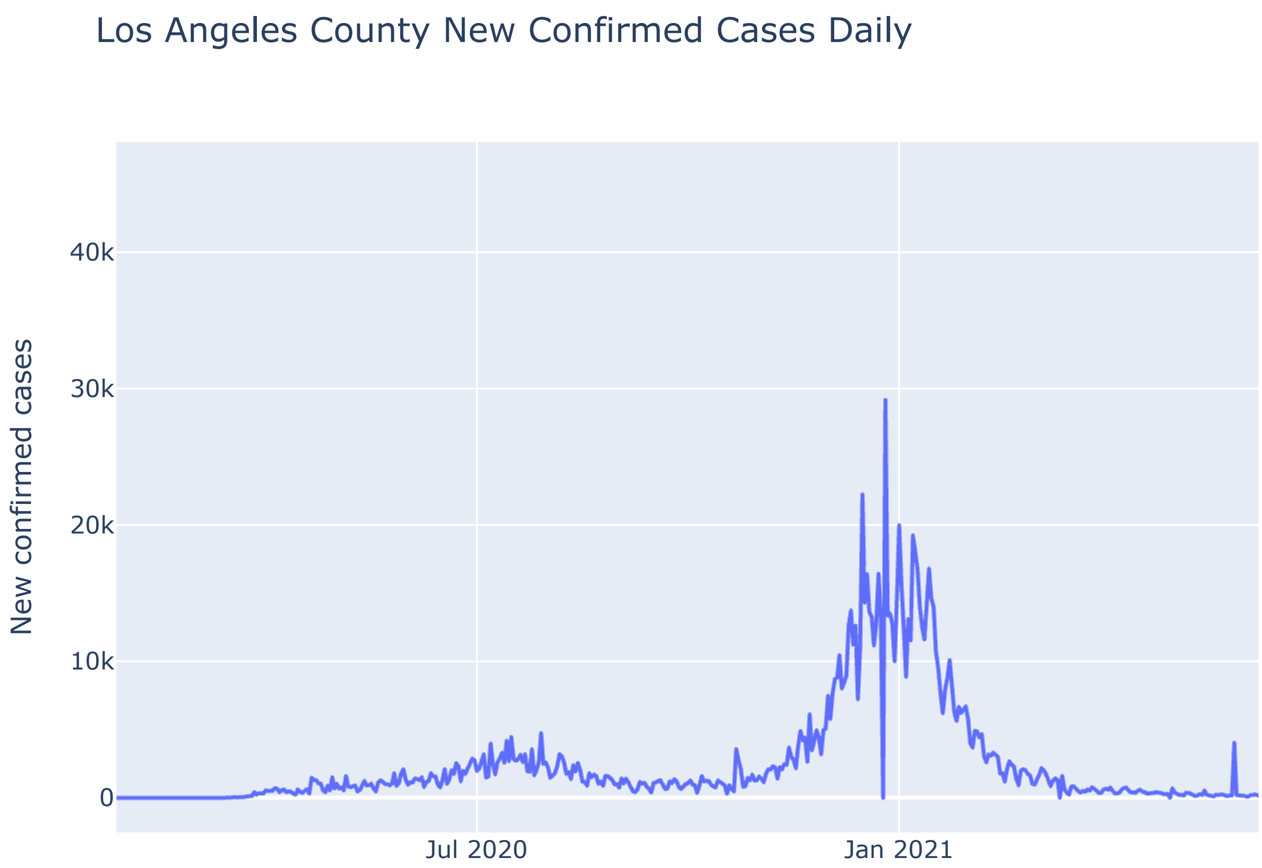
```
             date  confirmed_cases  new_cases
0     2020-01-26                1          0
1     2020-01-27                1          0
2     2020-01-28                1          0
3     2020-01-29                1          0
4     2020-01-30                1          0
...          ...              ...        ...
1066  2022-12-27          3622954       6958
1067  2022-12-28          3625123       2169
1068  2022-12-29          3629061       3938
1069  2022-12-30          3632440       3379
1070  2022-12-31          3632440          0

[1071 rows x 3 columns]
```

3.3 Interactive line graphs (Plotly)

```
import plotly.express as px

fig = px.line(
    df_la,
    x='date',
    y='new_cases',
    title='Los Angeles County New Confirmed Cases Daily',
    labels={'new_cases':'New confirmed cases','date':'Date'}
)
fig.update_layout(hovermode='x unified')
fig.show()
```



4. Predictive Modeling

4.1Feature engineering

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_sco

# (assuming df_la already contains date (datetime), confirmed_cases, new new_case
# df_la = client.query(query2).to_dataframe()
# df_la['date'] = pd.to_datetime(df_la['date'])
# df_la['new_cases'] = df_la['confirmed_cases'].diff().fillna(0).astype(int)

# Construct lags for the last 7 days
df_ml = df_la.copy()
for lag in range(1, 8):
    df_ml[f'lag_{lag}'] = df_ml['new_cases'].shift(lag)

# Construct target: Will it be added tomorrow >median
median_new = df_ml['new_cases'].median()
```

```
df_ml['target'] = (df_ml['new_cases'].shift(-1) > median_new).astype(int)

# Discard rows with NaN
df_ml = df_ml.dropna().reset_index(drop=True)

df_ml[['date','new_cases'] + [f'lag_{i}' for i in range(1,8)] + ['target']].head(
```

	date	new_cases	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	target
0	2020-02-02	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1	2020-02-03	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	2020-02-04	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
	2020-									

4.2 Delineation of training/testing sets

```
# features + label
X = df_ml[[f'lag_{i}' for i in range(1,8)]]
y = df_ml['target']

# Split with time series: no shuffling
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, shuffle=False)
```

4.3 Model training and evaluation

```
# Training
model = LogisticRegression(max_iter=500)
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Reporting
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.71875

Classification Report:
               precision    recall  f1-score   support

           0       0.69      0.63      0.66       137
           1       0.74      0.79      0.76       183

    accuracy                           0.72       320
   macro avg       0.71      0.71      0.71       320
```

```
weighted avg       0.72      0.72      0.72       320

Confusion Matrix:
 [[ 86  51]
 [ 39 144]]
```

5. Managerial Insights and Takeaways

1. ** Prioritizing resources ** : The Top 5 counties with the highest risk of severe illness and death are prioritized to receive medical supplies and human support.
2. ** Dynamic early warning ** : High incidence early warning based on model prediction, which can start emergency response 1-2 days in advance and optimize emergency material allocation.
3. ** Monitoring and Strategy ** : Combined with interactive visualization, real-time monitoring of peak periods and adjusting epidemic prevention policies (such as restricting aggregation and expanding detection) to cut peaks.