TIME SERIES ANALYSIS USING PROPHET

COVID19 CASE PREDICTIONS WITH PROPHET MODEL

This project aims to analyze COVID-19 data using the Prophet time series model to identify trends and predict future confirmed cases. The model Prophet's capabilities, and key patterns like trends are explored. The goal is to evaluate the model's performance and understand its predictive power through metrics and residual analysis.

```
# Import Necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load Data
data = pd.read_csv('covid_19_clean_complete.csv')
data.head(3)
```

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	Active	WHO Region
0	NaN	Afghanistan	33.93911	67.709953	2020-01-22	0	0	0	0	Eastern Mediterranean
1	NaN	Albania	41.15330	20.168300	2020-01-22	0	0	0	0	Europe
2	NaN	Algeria	28.03390	1.659600	2020-01-22	0	0	0	0	Africa

DATA PREPROCESSING

The dataset was preprocessed by inspecting data types and dimensions, converting the 'Date' column to datetime format, and verifying unique dates. Data was grouped by 'Date' to calculate daily totals. The 'Date' and 'Confirmed' columns were renamed to 'ds' and 'y' for Prophet compatibility, and non-essential columns were removed to focus on confirmed cases.

```
# Data Types
data.dtypes
# Total Columns and Rows
data.shape
```

(49068, 10)

```
# Check Null Values
data.isnull().sum()
```

Province/State	34404
Country/Region	0
Lat	0
Long	0
Date	0
Confirmed	0
Deaths	0
Recovered	0
Active	0
WHO Region	0

```
# Unique Dates
data['Date'].nunique()
```

188

```
# Group Data by Dates
data2 = data.groupby(['Date'])[['Confirmed', 'Deaths', 'Recovered',
'Active']].sum().reset_index()
data2['Date'].head(3)
```

	Date
0	2020-01-22
1	2020-01-23
2	2020-01-24

```
data2.shape
```

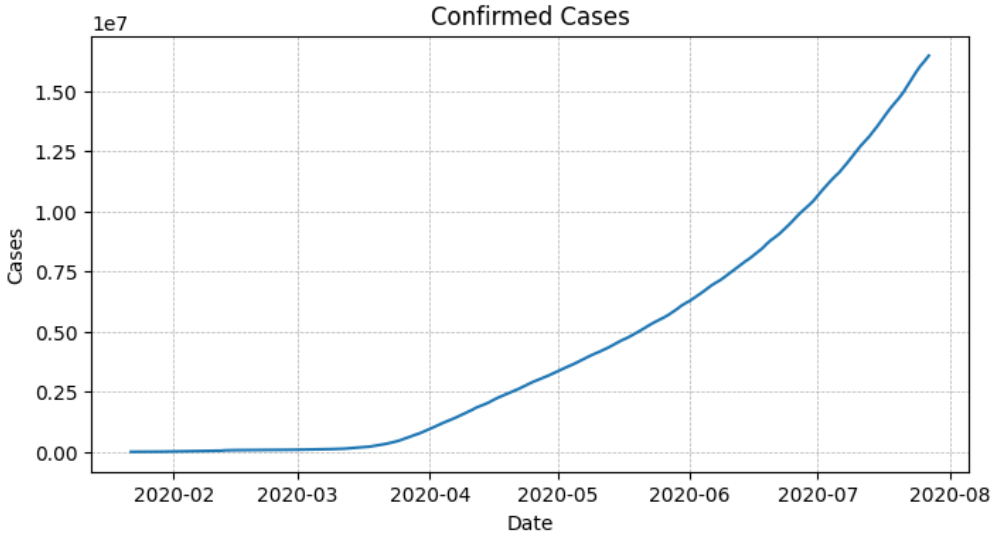
(188, 5)

```
# Rename Columns as ds and y
df = data2.rename(columns={'Date': 'ds', 'Confirmed': 'y'})
# Drop Deaths, Recovered and Active Column
df.drop(['Deaths', 'Recovered', 'Active'], axis=1, inplace=True)
df.head(3)
```

	ds	y
0	2020-01-22	555
1	2020-01-23	654
2	2020-01-24	941

PLOT THE DATA

```
plt.figure(figsize=(8,4))
plt.plot(df['ds'], df['y'])
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.title('Confirmed Cases', fontsize=12)
plt.xlabel('Date', fontsize=10)
plt.ylabel('Cases', fontsize=10)
```



The plot shows the trend of confirmed COVID-19 cases over time, demonstrating a clear exponential growth pattern during the observed period.

BUILDING THE PROPHET MODEL

Fitting the Model: The Prophet model is initialized and fitted to the training dataset to capture trends and seasonality patterns.

```
# Fit the Prophet Model
from prophet import Prophet
from prophet.plot import plot_plotly, plot_components_plotly, add_changepoints_to_plot

m = Prophet()
model=m.fit(df)
```

Making Predictions: A future dataframe is created for the next 30 days, and the model predicts the confirmed cases for this period.

```
# Create Future Dataframe
future = model.make_future_dataframe(periods=30, freq='D', include_history=False)
forecast = model.predict(future)

# View Predictions
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']]
```

	ds	yhat	yhat_lower	yhat_upper
0	2020-07-28	1.632020e+07	1.621924e+07	1.643263e+07
1	2020-07-29	1.652997e+07	1.642275e+07	1.663320e+07
2	2020-07-30	1.674391e+07	1.663175e+07	1.685091e+07
3	2020-07-31	1.695910e+07	1.684709e+07	1.706350e+07
4	2020-08-01	1.716676e+07	1.704431e+07	1.727728e+07
5	2020-08-02	1.736430e+07	1.725063e+07	1.747884e+07

```
forecast.shape
```

(30, 16)

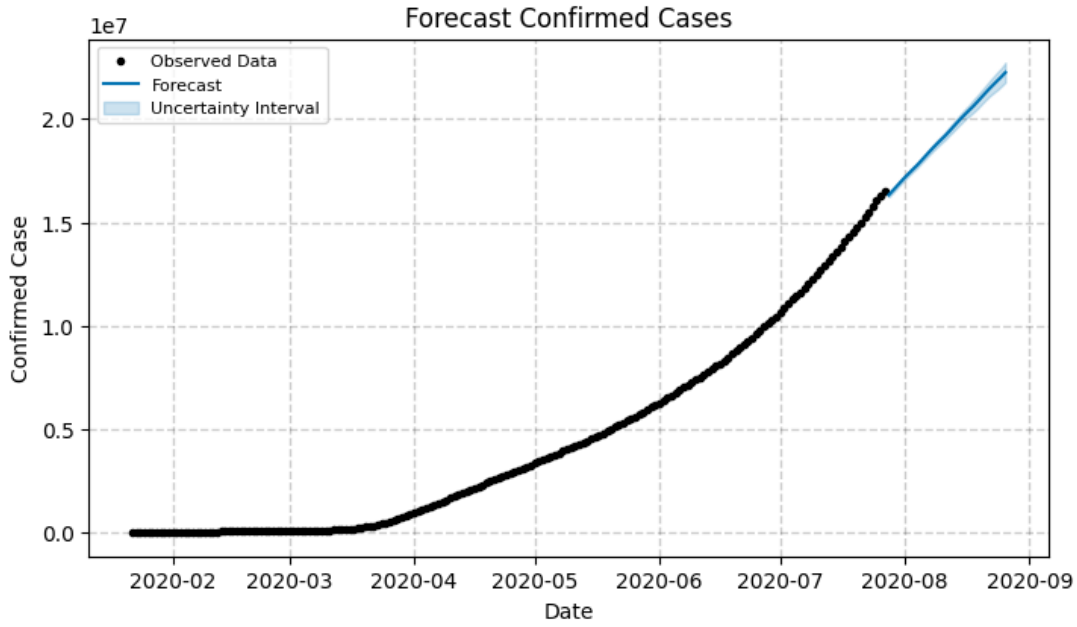
FORECAST VISUALIZATION

The plot below illustrates the observed data alongside the forecasted confirmed cases. It also highlights the uncertainty intervals, providing insights into potential variations in future trends.

```
    # Generate the Prophet Plot
    fig = model.plot(forecast)

    ax = fig.gca()
    fig.set_size_inches(7, 4)

    ax.grid(True, linestyle='--', color='black')
    ax.set_title('Forecast Confirmed Cases', fontsize=12)
    ax.set_xlabel('Date', fontsize=10)
    ax.set_ylabel('Confirmed Case', fontsize=10)
    ax.legend(['Observed Data', 'Forecast', 'Uncertainty Interval'], loc='upper left',
fontsize=8)
    plt.show()
```



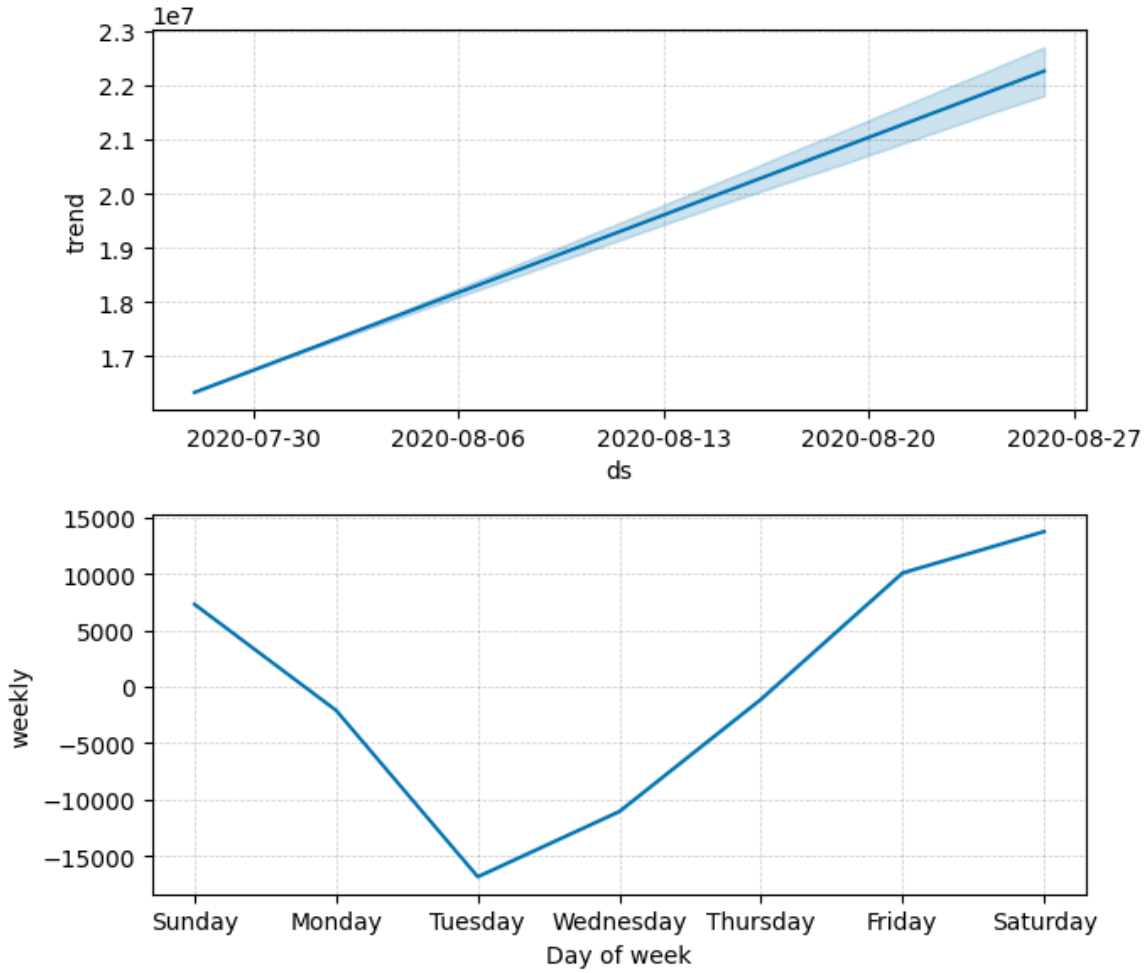
Component analysis plot below provides insights into the components influencing the forecast.

```
    # Generate the Prophet components plot
    fig = model.plot_components(forecast)

    axes = fig.get_axes()
    fig.set_size_inches(7, 6)

    for ax in axes:
        ax.grid(True, linestyle='--', linewidth=0.5, color='black')
        ax.title.set_fontsize(12)
        ax.xaxis.label.set_fontsize(10)
        ax.yaxis.label.set_fontsize(10)

    fig.tight_layout()
    plt.show()
```



Trend: The upper panel shows a steady exponential increase in confirmed cases over time, reflecting the overall growth pattern of the pandemic.
Weekly Seasonality: The lower panel highlights weekly fluctuations, indicating that cases tend to drop on Mondays and steadily rise toward the weekend.

CROSS VALIDATING THE MODEL

To evaluate the model's performance, cross-validation was performed using Prophet's diagnostic tools. The dataset was divided into training and testing splits dynamically by the cross_validation function. A rolling origin approach was used to forecast for a horizon of 30 days with a training period of at least 150 days, and evaluations were conducted at intervals of 30 days.

```
from prophet.diagnostics import cross_validation
cv = cross_validation(model, horizon='30 days', period='30 days', initial='150 days')
cv.sample(3)
```

	ds	yhat	yhat_lower	yhat_upper	y	cutoff
20	2020-07-18	1.250476e+07	1.232679e+07	1.270887e+07	14292198	2020-06-27
11	2020-07-09	1.130617e+07	1.121923e+07	1.139831e+07	12273063	2020-06-27
27	2020-07-25	1.342757e+07	1.315339e+07	1.374951e+07	16047190	2020-06-27

```
cv.shape
```

(30, 6)

```
from prophet.diagnostics import performance_metrics
performance = performance_metrics(cv)
performance
```

	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	3 days	8.991621e+10	2.998603e+05	2.982883e+05	0.028980	0.028752	0.029409	0.0
1	4 days	1.262131e+11	3.552648e+05	3.513753e+05	0.033511	0.032267	0.034092	0.0
2	5 days	1.777171e+11	4.215651e+05	4.168035e+05	0.038999	0.039512	0.039789	0.0
3	6 days	2.427774e+11	4.927245e+05	4.896636e+05	0.044965	0.045217	0.046009	0.0
4	7 days	3.088267e+11	5.557218e+05	5.535089e+05	0.049922	0.050166	0.051208	0.0
5	8 days	3.810181e+11	6.172666e+05	6.152759e+05	0.054552	0.054384	0.056088	0.0

The average performance metrics from cross-validation are as follows.

```
# Average Performance Metrics
print("MSE:", performance['mse'].mean().round(3))
print("RMSE:", performance['rmse'].mean().round(3))
print("MAE:", performance['mae'].mean().round(3))
print("MAPE:", performance['mape'].mean().round(3))
```

MSE: 2339143902993.477; RMSE: 1347913.031; MAE: 1345658.575; MAPE: 0.097

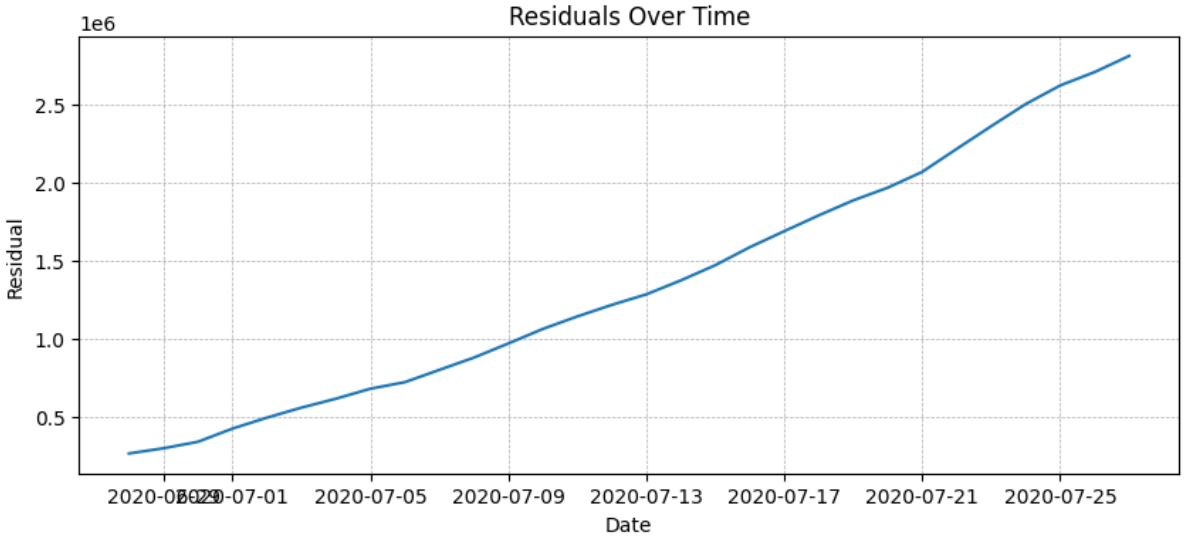
MSE: Extremely high due to the large scale of the data, making it less interpretable.
RMSE: Approximately 1,347,913 cases, indicating significant prediction errors in absolute terms.
MAE: Around 1,345,659 cases, reflecting the typical size of errors.
MAPE: 9.7%, showing that predictions deviate on average by 9.7% of actual values, which demonstrates strong relative accuracy.

Overall, while the absolute errors (RMSE and MAE) highlight challenges in predicting large values accurately, the MAPE demonstrates that the model performs well in capturing relative trends and variations in the data. This suggests the model is suitable for trend forecasting but could benefit from further optimization for more precise case predictions.

RESIDUAL ANALYSIS

The residual plot shows the differences between the actual values and the predicted values over time.

```
cv['residual'] = cv['y'] - cv['yhat']
plt.figure(figsize=(10,4))
plt.plot(cv['ds'], cv['residual'])
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.title("Residuals Over Time")
plt.xlabel("Date")
plt.ylabel("Residual")
plt.show()
```



The residual plot shows a steady increase in errors over time, indicating that the model tends to underestimate the actual values as the dataset progresses. This upward trend suggests that the model struggles to fully capture the exponential growth in cases. The residuals reach up to 2.5 million, aligning with the high RMSE observed in performance metrics. This highlights the need for further tuning or incorporating additional factors to improve the model's accuracy.

Notebook Link:
https://github.com/suranjitpartho/timeseries_prophet_covid19/blob/main/timeseries_prophet_covid19.ipynb

Data Source:
The dataset used for this analysis was downloaded from Kaggle. It contains daily COVID-19 statistics.
https://www.kaggle.com/datasets/shriyasingh900/covid19-dataset