Universidad Central del Ecuador

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Sistemas De Información

Documento de Selección del Modelo

Estudiante:

- Luis Angel Gaona Cumbicus lagaona@uce.edu.ec
- Raul Alexander Pazos Erraez rapazos@uce.edu.ec
- Cristian Daniel Toca Rocha cdtoca@uce.edu.ec
- Marlon Josue Espinosa Mancero mjespinosam@uce.edu.ec

Docente:

- PhD, Jefferson Tarcisio Beltrán Morales jtbeltran@uce.edu.ec

Asignatura: Minería de datos

Paralelo: S8-P2

Fecha: sábado 13 de julio de 2024

Generative AI-Powered Economic Impact Analysis System (GEIA)

Fecha:*13/07/2024*

Contenido

HOJA DE CONTROL

Organismo	Universidad Central Del Ecuador		
Proyecto	Generative AI-Powered Economic Impact Analysis System (**GEIA**)		
Entregable	Documento de selección del modelo		
Autor	Luis Angel Gaona Cumbicus		
Versión/Edición	V1.0	Fecha Versión	13/07/2024
Aprobado por		Fecha Aprobación	…../…../…..
		N.º Total de Páginas	15

Historial de Cambios

Fecha	Autor	Organización	Descripción

Introducción

En el proyecto de implementación de un modelo predictivo para estimar los daños económicos en diferentes sectores de la economía ecuatoriana causados por eventos adversos, se requiere seleccionar un modelo adecuado para el análisis de series de tiempo interrumpidas. Para este propósito, se comparan dos modelos prominentes: **Prophet y CausalImpact.**

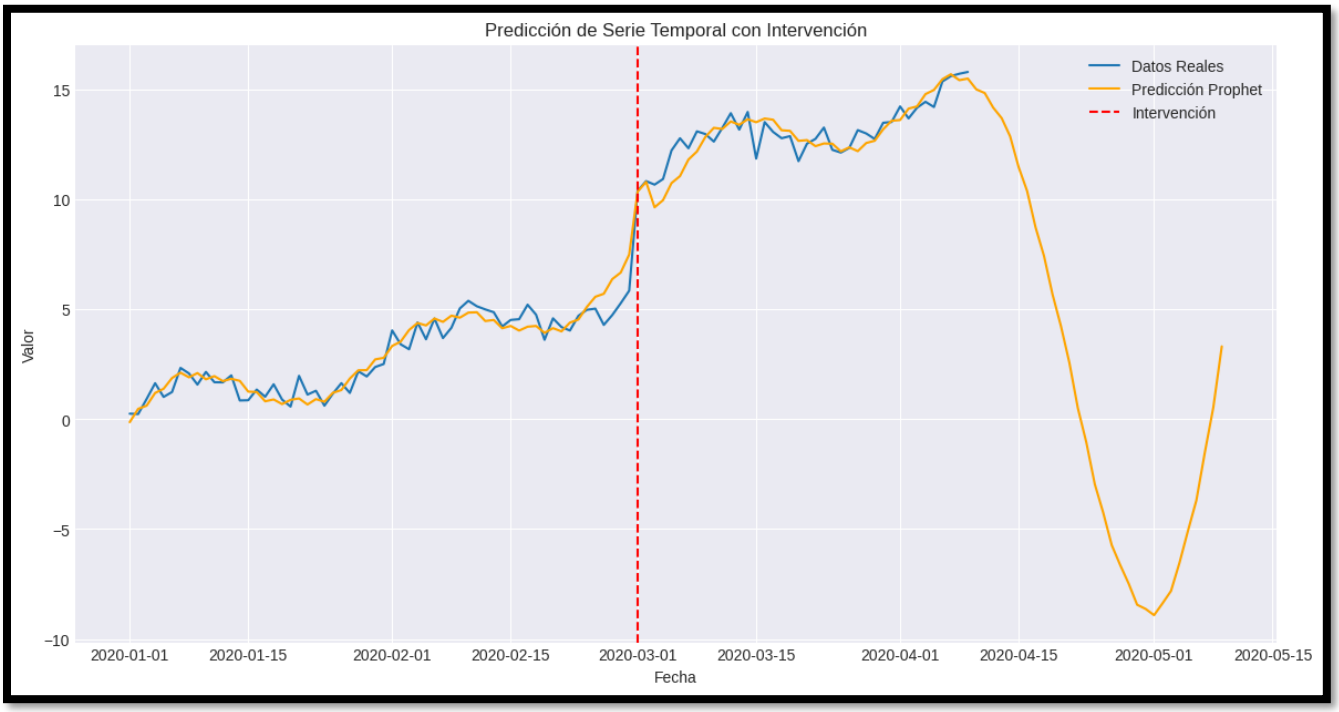
Modelos Considerados

1. Prophet

Prophet es un modelo desarrollado por Facebook para la predicción de series de tiempo que manejan estacionalidad diaria, semanal y anual. Es especialmente útil para datos con tendencias estacionales y cambios abruptos. Prophet modela las tendencias y estacionalidades con componentes aditivos y puede manejar cambios en la estacionalidad a lo largo del tiempo.

```
# Instalación de la librería Prophet (si no está instalada)
!pip install prophet

# Importar librerías
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from prophet import Prophet

# Configuración de gráficos
plt.style.use('seaborn-darkgrid')

# Generar fechas
np.random.seed(42)
dates = pd.date_range(start='2020-01-01', periods=100, freq='D')

# Generar datos sintéticos
data = np.sin(np.linspace(0, 20, 100)) + np.linspace(0, 10, 100) + np.random.normal(scale=0.5, size=100)

# Introducir una intervención a partir del día 60
data[60:] += 5

# Crear DataFrame para Prophet
df = pd.DataFrame({'ds': dates, 'y': data})

# Crear DataFrame para eventos
events = pd.DataFrame({
    'ds': [dates[60]],  # Fecha de la intervención
    'holiday': ['Intervención']
})

# Definir los días festivos para el modelo Prophet
holidays = pd.DataFrame({
    'holiday': 'Intervención',
    'ds': pd.to_datetime([dates[60]]),
    'lower_window': 0,
    'upper_window': 1,
})
```

```python
# Definir y ajustar el modelo Prophet
model = Prophet(holidays=holidays, yearly_seasonality=True, weekly_seasonality=True, daily_seasonality=False)
model.fit(df)

# Crear DataFrame para predicciones futuras
future = model.make_future_dataframe(periods=30)
forecast = model.predict(future)

# Extraer predicciones
predicciones = forecast[['ds', 'yhat']].copy()

# Determinar el punto de intervención
intervencion_fecha = dates[60]
intervencion_indice = predicciones[predicciones['ds'] >= intervencion_fecha].index[0]

# Extraer valores antes y después de la intervención
antes_intervencion = predicciones.loc[predicciones['ds'] < intervencion_fecha, 'yhat']
despues_intervencion = predicciones.loc[predicciones['ds'] >= intervencion_fecha, 'yhat']

# Calcular el impacto promedio
impacto_promedio = despues_intervencion.mean() - antes_intervencion.mean()

# Mostrar impacto
print(f"Impacto promedio debido a la intervención: {impacto_promedio:.2f}")

# Visualizar resultados
plt.figure(figsize=(14, 7))
plt.plot(df['ds'], df['y'], label='Datos Reales')
plt.plot(forecast['ds'], forecast['yhat'], label='Predicción Prophet', color='orange')
plt.axvline(x=intervencion_fecha, color='red', linestyle='--', label='Intervención')
plt.title('Predicción de Serie Temporal con Intervención')
plt.xlabel('Fecha')
plt.ylabel('Valor')
plt.legend()
plt.show()
```



2. CausalImpact

CausalImpact es una herramienta de Google para la evaluación del impacto de intervenciones en series de tiempo. Utiliza un modelo bayesiano para estimar el impacto de un evento en una serie temporal comparando el comportamiento real con un contrafactual, es decir, cómo se habría comportado la serie temporal en ausencia del evento.

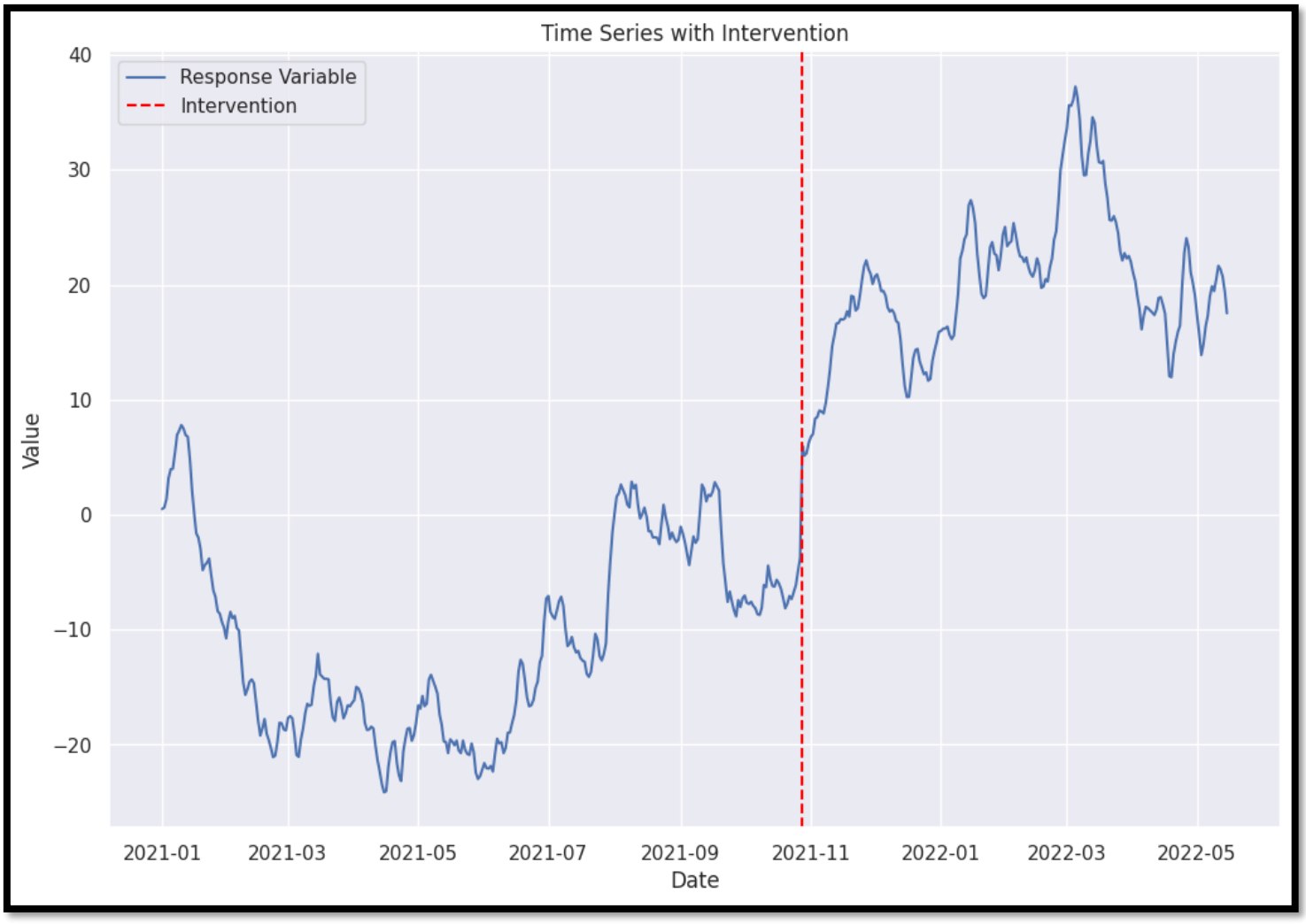
TIPO DE MODELOS

1. ***Modelo univariable***

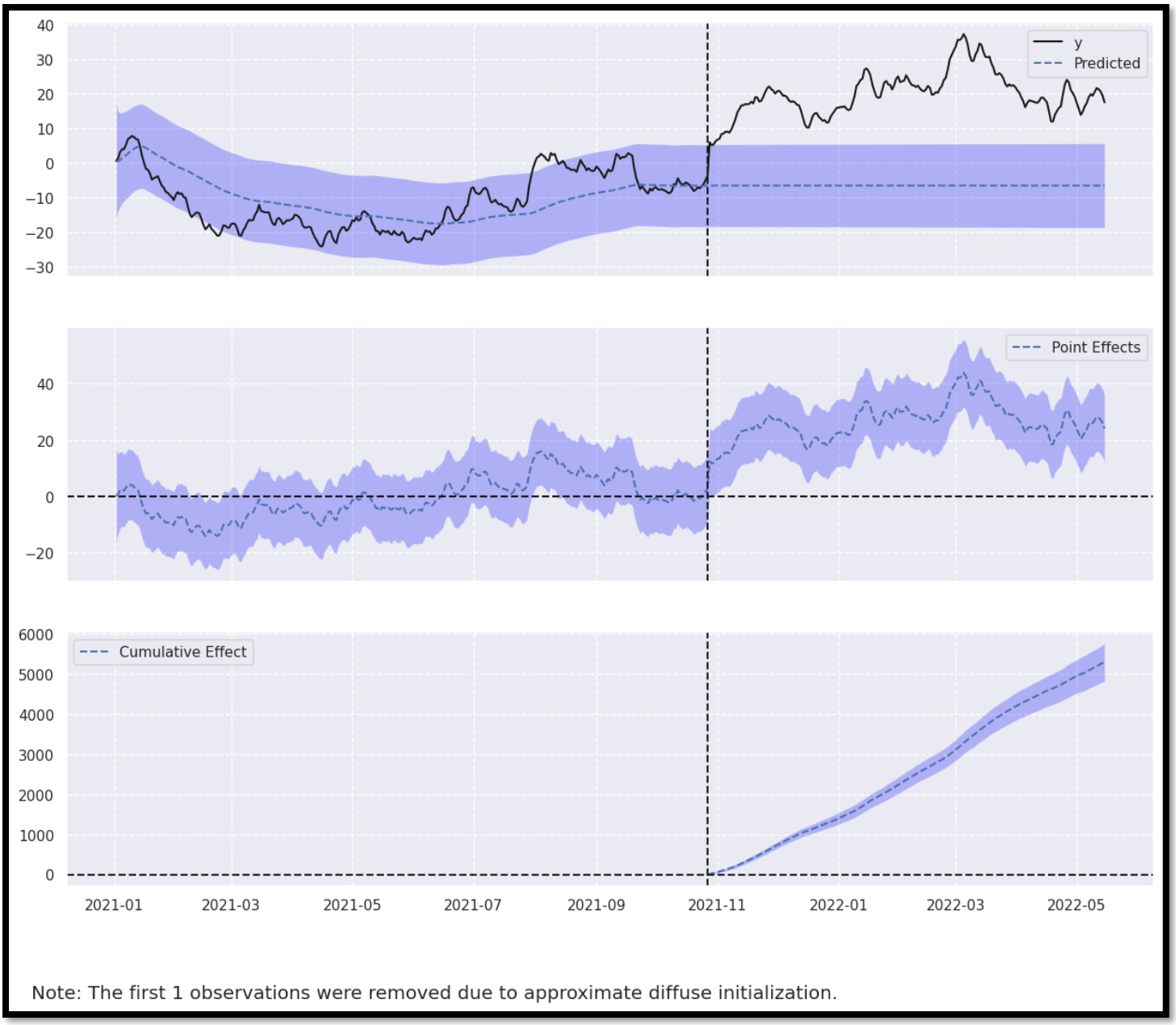
```python
# Data processing
import pandas as pd
import numpy as np
from datetime import datetime

# Create synthetic time-series data
from statsmodels.tsa.arima_process import ArmaProcess

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Causal impact
from causalimpact import CausalImpact

# Set up a seed for reproducibility
np.random.seed(42)

# Autoregressive coefficients
arparams = np.array([.95, .05])

# Moving average coefficients
maparams = np.array([.6, .3])

# Create an ARMA process
arma_process = ArmaProcess.from_coeffs(arparams, maparams)

# Create the time-series data
y = arma_process.generate_sample(nsample=500)

# Add the true causal impact
y[300:] += 10

# Create dates
dates = pd.date_range('2021-01-01', freq='D', periods=500)

# Create dataframe
df = pd.DataFrame({'dates': dates, 'y': y}, columns=['dates', 'y'])

# Set dates as index
df.set_index('dates', inplace=True)
```

```python
# Take a look at the data
print(df.head())

# Print out the time series start date
print(f'The time-series start date is: {df.index.min()}')

# Print out the time series end date
print(f'The time-series end date is: {df.index.max()}')

# Print out the intervention start date
print(f'The treatment start date is: {df.index[300]}')

# Visualize data using seaborn
sns.set(rc={'figure.figsize':(12,8)})
sns.lineplot(x=df.index, y=df['y'], label='Response Variable')
plt.axvline(x=df.index[300], color='red', linestyle='--', label='Intervention')
plt.legend()
plt.title('Time Series with Intervention')
plt.xlabel('Date')
plt.ylabel('Value')
plt.grid(True)
plt.show()

# Set pre-period
pre_period = [str(df.index.min())[:10], str(df.index[299])[:10]]

# Set post-period
post_period = [str(df.index[300])[:10], str(df.index.max())[:10]]

# Print out the values
print(f'The pre-period is: {pre_period}')
print(f'The post-period is: {post_period}')

# Calculate the pre-daily average
pre_daily_avg = df['y'][:300].mean()

# Calculate the post-daily average
post_daily_avg = df['y'][300:].mean()

# Print out the results
print(f'The pre-treatment daily average is: {pre_daily_avg}.')
print(f'The post-treatment daily average is: {post_daily_avg}.')
print(f'The raw difference between the pre and the post treatment is: {post_daily_avg - pre_daily_avg}.')
```

```
# Visualization
impact.plot()

# Causal impact summary
print(impact.summary())
print(impact.summary('report'))
```



```
The pre-period is: ['2021-01-01', '2021-10-27']
The post-period is: ['2021-10-28', '2022-05-15']
The pre-treatment daily average is: -10.803942129631345.
The post-treatment daily average is: 19.978180589987222.
The raw difference between the pre and the post treatment is: 30.782122719618567.
```



Note: The first 1 observations were removed due to approximate diffuse initialization.

```
Posterior Inference {Causal Impact}
                          Average            Cumulative
Actual                    19.98              3995.64
Prediction (s.d.)         -6.54 (1.2)        -1308.33 (240.77)
95% CI                    [-8.83, -4.11]     [-1765.94, -822.13]

Absolute effect (s.d.)    26.52 (1.2)        5303.97 (240.77)
95% CI                    [24.09, 28.81]     [4817.76, 5761.58]

Relative effect (s.d.)    -405.4% (18.4%)    -405.4% (18.4%)
95% CI                    [-440.38%, -368.24%][-440.38%, -368.24%]

Posterior tail-area probability p: 0.0
Posterior prob. of a causal effect: 100.0%

For more details run the command: print(impact.summary('report'))
Analysis report {CausalImpact}
```

Durante el periodo post-intervención, la variable respuesta tuvo un valor medio de aprox. 19,98. Por el contrario, en ausencia de una intervención, hubiéramos esperado una respuesta promedio de -6,54. El intervalo del 95% de esta predicción contrafactual es [-8,83, -4,11]. Restando esta predicción de la respuesta observada se obtiene una estimación del efecto causal que tuvo la intervención sobre la variable de respuesta. Este efecto es 26,52 con un intervalo del 95% de [24.09, 28.81]. Para una discusión sobre la importancia de este efecto, vea abajo.

Resumir los puntos de datos individuales durante la posintervención. período (que sólo a veces puede ser interpretado significativamente), el La variable respuesta tuvo un valor global de 3995,64. Por el contrario, si la intervención no hubiera tenido lugar, habríamos esperado una suma de -1308,33. El intervalo del 95% de esta predicción es [-1765,94, -822,13].

Los resultados anteriores se dan en términos de números absolutos. En relativo En términos generales, la variable respuesta presentó una disminución del -405,4%. el 95% El intervalo de este porcentaje es [-440,38%, -368,24%].

Esto significa que el efecto negativo observado durante la intervención período es estadísticamente significativo. Si el experimentador esperaba un efecto positivo, se recomienda para verificar si las anomalías en las variables de control pueden tener provocó una expectativa demasiado optimista de lo que debería haber sucedido en la variable respuesta en ausencia de la intervención.

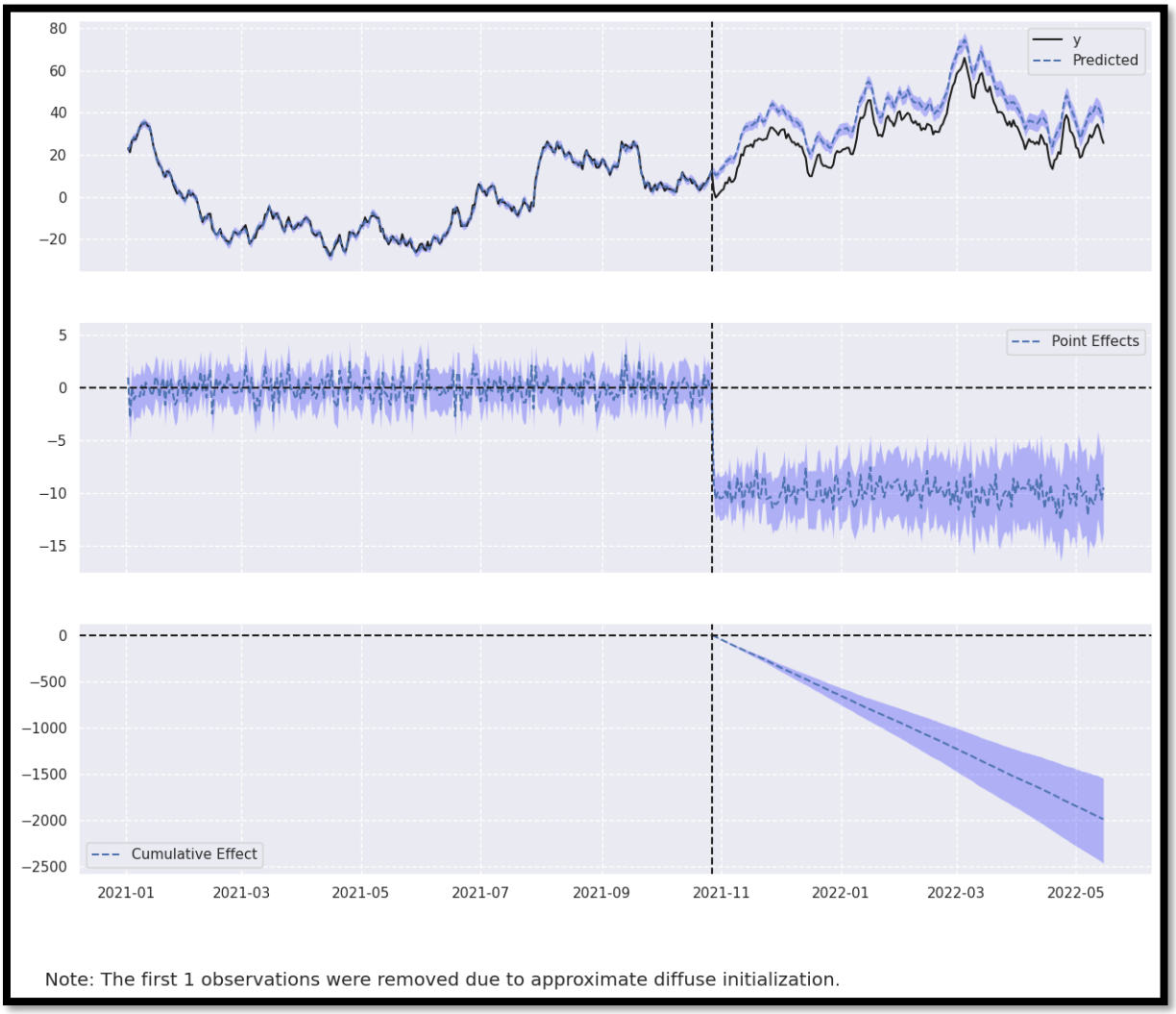
La probabilidad de obtener este efecto por casualidad es muy pequeña. (Probabilidad bayesiana del área de la cola unilateral p = 0,0). Esto significa que el efecto causal puede considerarse estadísticamente. significativo.

2. *Modelo multivariable*

```
# Data processing
import pandas as pd
import numpy as np
from datetime import datetime

# Create synthetic time-series data
from statsmodels.tsa.arima_process import ArmaProcess

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Causal impact
from causalimpact import CausalImpact

# Set up a seed for reproducibility
np.random.seed(42)

# Autoregressive coefficients
arparams = np.array([.95, .05])

# Moving average coefficients
maparams = np.array([.6, .3])

# Create a ARMA process
arma_process = ArmaProcess.from_coeffs(arparams, maparams)

# Create the control time-series
X = 10 + arma_process.generate_sample(nsample=500)
```

```
# Create the response time-series
y = 2 * X + np.random.normal(size=500)

# Add the true causal impact
y[300:] -= 10

# Create dates
dates = pd.date_range('2021-01-01', freq='D', periods=500)

# Create dataframe
df = pd.DataFrame({'dates': dates, 'y': y, 'X': X}, columns=['dates', 'y', 'X'])

# Set dates as index
df.set_index('dates', inplace=True)

# Take a look at the data
df.head()


# Print out the time series start date
print(f'The time-series start date is :{df.index.min()}')

# Print out the time series end date
print(f'The time-series end date is :{df.index.max()}')

# Print out the intervention start date
print(f'The treatment start date is :{df.index[300]}')
```

```
# Visualize data using seaborn
sns.set(rc={'figure.figsize':(12,8)})
sns.lineplot(x=df.index, y=df['X'])
sns.lineplot(x=df.index, y=df['y'])
plt.axvline(x= df.index[300], color='red')
plt.legend(labels = ['X', 'y'])

# Set pre-period
pre_period = [str(df.index.min())[:10], str(df.index[299])[:10]]

# Set post-period
post_period = [str(df.index[300])[:10], str(df.index.max())[:10]]

# Print out the values
print(f'The pre-period is {pre_period}')
print(f'The post-period is {post_period}')
# Calculate the pre-daily average
pre_daily_avg = df['y'][:300].mean()

# Calculate the post-daily average
post_daily_avg = df['y'][300:].mean()

# Print out the results
print(f'The pre-treatment daily average is {pre_daily_avg}.')
print(f'The post-treatment daily average is {post_daily_avg}.')
print(f'The raw difference between the pre and the post treatment is {post_daily_avg - pre_daily_avg}.')

# Causal impact model
impact = CausalImpact(data=df, pre_period=pre_period, post_period=post_period)

# Visualization
impact.plot()
```



Note: The first 1 observations were removed due to approximate diffuse initialization.

```
Posterior Inference {Causal Impact}
                          Average            Cumulative
Actual                    30.08              6016.92
Prediction (s.d.)         40.03 (1.17)       8005.59 (234.99)
95% CI                    [37.81, 42.41]     [7561.42, 8482.55]

Absolute effect (s.d.)    -9.94 (1.17)       -1988.67 (234.99)
95% CI                    [-12.33, -7.72]    [-2465.63, -1544.5]

Relative effect (s.d.)    -24.84% (2.94%)    -24.84% (2.94%)
95% CI                    [-30.8%, -19.29%]  [-30.8%, -19.29%]

Posterior tail-area probability p: 0.0
Posterior prob. of a causal effect: 100.0%

For more details run the command: print(impact.summary('report'))
```

Durante el periodo post-intervención, la variable respuesta tuvo un valor medio de aprox. 30.08. Por el contrario, en ausencia de una intervención, hubiéramos esperado una respuesta promedio de 40,03. El intervalo del 95% de esta predicción contrafáctica es [37,81, 42,41]. Restando esta predicción de la respuesta observada se obtiene una estimación del efecto causal que tuvo la intervención sobre la variable de respuesta. Este efecto es -9,94 con un intervalo del 95% de [-12,33, -7,72]. Para una discusión sobre la importancia de este efecto, vea abajo.

Resumir los puntos de datos individuales durante la posintervención. período (que sólo a veces puede ser interpretado significativamente), el La variable respuesta tuvo un valor global de 6016,92. Por el contrario, si la intervención no hubiera tenido lugar, habríamos esperado una suma de 8005,59. El intervalo del 95% de esta predicción es [7561,42, 8482,55].

Los resultados anteriores se dan en términos de números absolutos. En relativo En términos generales, la variable respuesta presentó una disminución del -24,84%. el 95% El intervalo de este porcentaje es [-30,8%, -19,29%].

Esto significa que el efecto negativo observado durante la intervención período es estadísticamente significativo. Si el experimentador esperaba un efecto positivo, se recomienda para verificar si las anomalías en las variables de control pueden tener provocó una expectativa demasiado optimista de lo que debería haber sucedido en la variable respuesta en ausencia de la intervención.

La probabilidad de obtener este efecto por casualidad es muy pequeña. (Probabilidad bayesiana del área de la cola unilateral p = 0,0). Esto significa que el efecto causal puede considerarse estadísticamente. significativo.

Resultados y Justificación

En este proyecto, hemos evaluado dos modelos prominentes para el análisis de series de tiempo interrumpidas: Prophet y CausalImpact. Ambos modelos tienen sus fortalezas y aplicaciones específicas.

Prophet es una herramienta poderosa para realizar predicciones en series temporales, especialmente en datos con tendencias estacionales y cambios abruptos. Sin embargo, Prophet está diseñado principalmente para la predicción de tendencias futuras y no se centra en la evaluación del impacto de eventos específicos en la serie temporal. Si bien es útil para modelar el comportamiento general de la serie, no ofrece una metodología directa para cuantificar cómo un evento adverso afecta la serie temporal en comparación con un escenario contrafactual (sin el evento).

Por otro lado, **CausalImpact** es un modelo diseñado específicamente para medir el impacto de un evento en una serie temporal, lo que lo hace ideal para el problema que queremos resolver en este proyecto. CausalImpact no solo permite realizar un análisis contrafactual, sino que también ofrece un reporte detallado y resumido del impacto del evento, proporcionando claridad sobre la diferencia entre el escenario con y sin el evento. Este nivel de detalle es crucial para entender cómo un evento adverso afecta los diferentes sectores de la economía ecuatoriana.

Además, CausalImpact ofrece varias ventajas adicionales, como la posibilidad de integrar información de múltiples variables explicativas y la capacidad de generar intervalos de confianza para las estimaciones del impacto, lo que mejora la robustez y la interpretabilidad de los resultados. Estas características hacen que CausalImpact sea la herramienta más adecuada para nuestro objetivo de estimar los daños económicos causados por eventos adversos.

Por estas razones, se ha decidido adoptar CausalImpact como el modelo principal para el análisis de series de tiempo interrumpidas en este proyecto. Este enfoque nos permitirá obtener una visión clara y detallada del impacto económico de eventos adversos, contribuyendo a la toma de decisiones informadas para mitigar dichos impactos en el futuro.