In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt

from statsmodels.tsa.statespace.sarimax import SARIMAX
```

In [2]:
```
df = pd.read_csv(r"ML471_S4_Datafile_Practice.csv")

df['Date'] = pd.to_datetime(df['Date'])
df = df.sort_values('Date')
df.set_index('Date', inplace=True)

df = df.fillna(method='ffill').fillna(method='bfill')
```

```
C:\Users\DURGESH S V\AppData\Local\Temp\ipykernel_10896\564892809.py:7: FutureWar
ning: DataFrame.fillna with 'method' is deprecated and will raise in a future ver
sion. Use obj.ffill() or obj.bfill() instead.
  df = df.fillna(method='ffill').fillna(method='bfill')
```

In [3]:
```
y = df['Close']

exog = df[['Close_diff', 'SMA_10', 'SMA_30', 'SES']]
```

In [4]:
```
train_size = int(len(df) * 0.8)

y_train = y.iloc[:train_size]
y_test = y.iloc[train_size:]

exog_train = exog.iloc[:train_size]
exog_test = exog.iloc[train_size:]
```

In [5]:
```
model = SARIMAX(
    y_train,
    exog=exog_train,
    order=(1, 0, 1),
    seasonal_order=(3, 1, 1, 12),
    enforce_stationarity=False,
    enforce_invertibility=False
)

results = model.fit(disp=False)
```

```
C:\Users\DURGESH S V\AppData\Roaming\Python\Python311\site-packages\statsmodels\t
sa\base\tsa_model.py:473: ValueWarning: No frequency information was provided, so
inferred frequency ME will be used.
  self._init_dates(dates, freq)
C:\Users\DURGESH S V\AppData\Roaming\Python\Python311\site-packages\statsmodels\t
sa\base\tsa_model.py:473: ValueWarning: No frequency information was provided, so
inferred frequency ME will be used.
  self._init_dates(dates, freq)
C:\Users\DURGESH S V\AppData\Roaming\Python\Python311\site-packages\statsmodels\b
ase\model.py:607: ConvergenceWarning: Maximum Likelihood optimization failed to c
onverge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

In [8]:
```
forecast = results.get_forecast(
    steps=len(y_test),
    exog=exog_test
).predicted_mean
```

```
forecast.index = y_test.index
```

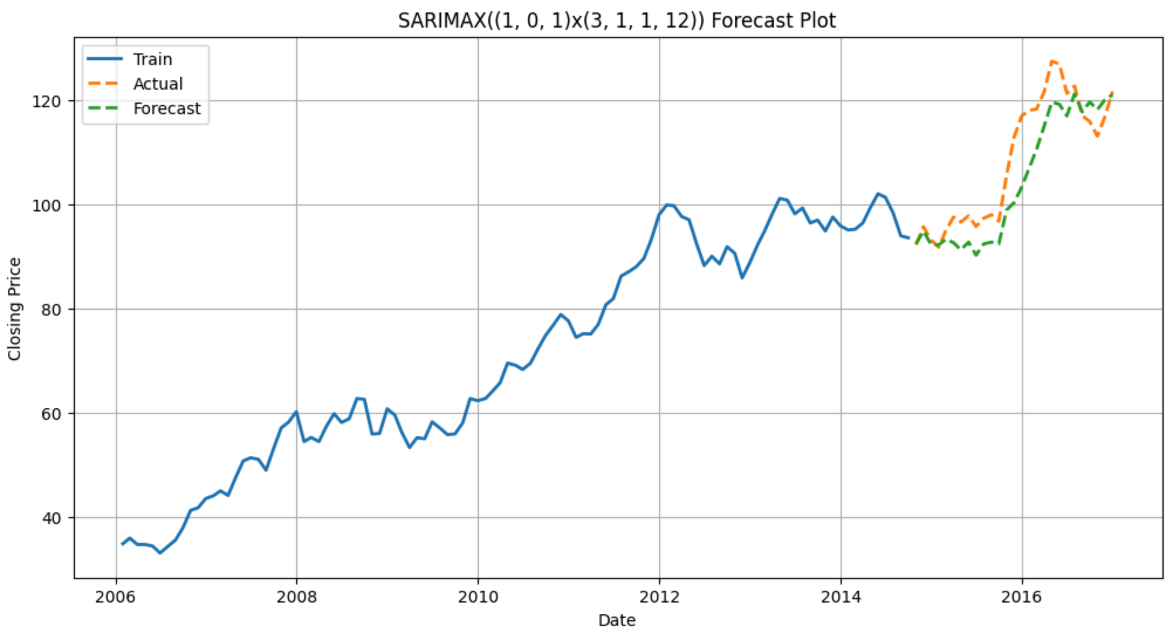
In [9]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))

plt.plot(y_train.index, y_train, label="Train", linewidth=2)
plt.plot(y_test.index, y_test, label="Actual", linestyle="--", linewidth=2)
plt.plot(forecast.index, forecast, label="Forecast", linestyle="--", linewidth=2

plt.title("SARIMAX((1, 0, 1)x(3, 1, 1, 12)) Forecast Plot")
plt.xlabel("Date")
plt.ylabel("Closing Price")
plt.legend()
plt.grid(True)

plt.show()
```



In [ ]: