8.ARIMA model for time series forecasting

Aim:

To implement a python program to Create an ARIMA model for time series forecasting.

Procedure:

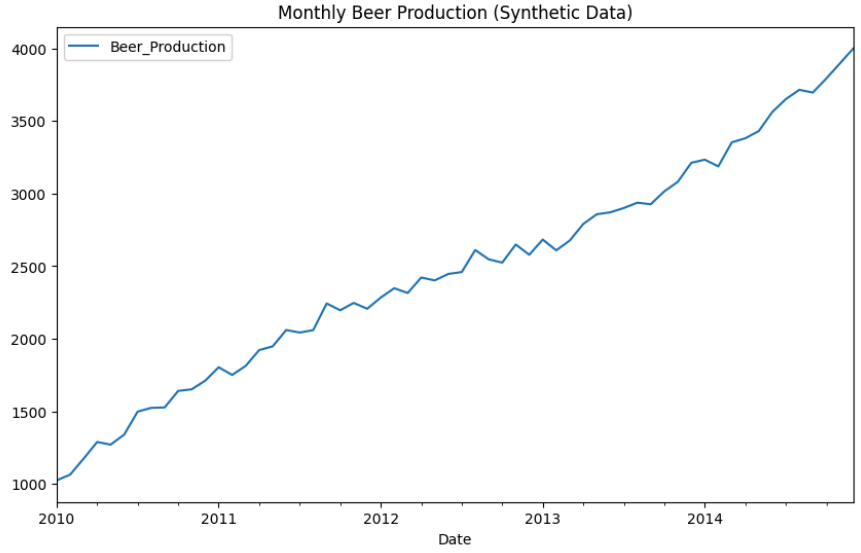
1. Collect or generate data (monthly beer production).
2. Preprocess data (parse dates, check for missing values).
3. Visualize the data to identify trends and seasonality.
4. Test for stationarity and apply differencing if needed.
5. Select ARIMA parameters using ACF and PACF plots.
6. Fit the ARIMA model to the data.
7. Make forecasts for future beer production.
8. Evaluate the model by comparing historical and forecasted values.

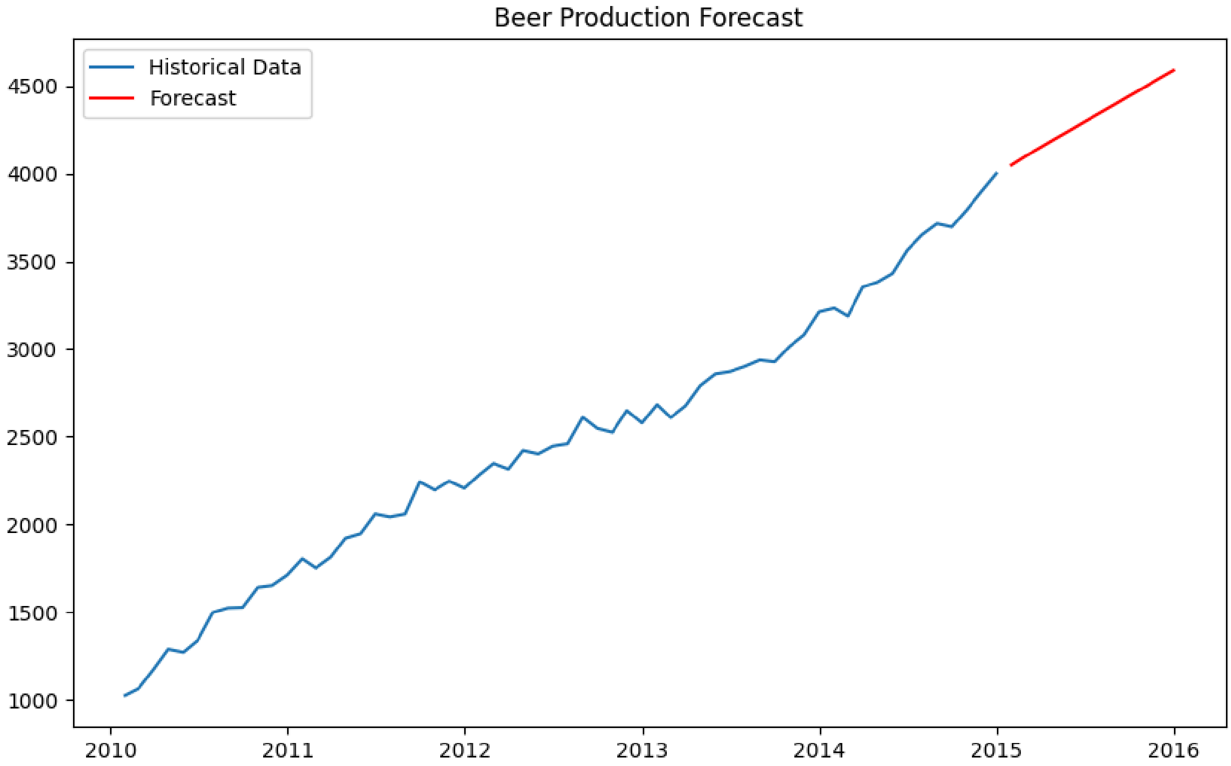
Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# 1. Generate Synthetic Monthly Beer Production Data (5 years)
np.random.seed(42)
date_range = pd.date_range(start='2010-01-01', periods=60, freq='M')
beer_production = 1000 + 50 * np.arange(60) + 200 * np.sin(np.linspace(0, 2 * np.pi, 60)) + np.random.normal(0, 50, 60)

# Create DataFrame
df = pd.DataFrame({'Date': date_range, 'Beer_Production': beer_production})
df.set_index('Date', inplace=True)

# 2. Plot the data
df.plot(figsize=(10,6))
```

```
plt.title('Monthly Beer Production (Synthetic Data)')
plt.show()

# 3. Fit ARIMA model (Order p=1, d=1, q=1)
model = ARIMA(df['Beer_Production'], order=(1, 1, 1))
fitted_model = model.fit()

# 4. Forecast the next 12 months
forecast = fitted_model.forecast(steps=12)

# 5. Plot historical data and forecast
plt.figure(figsize=(10,6))
plt.plot(df.index, df['Beer_Production'], label='Historical Data')
plt.plot(pd.date_range(df.index[-1], periods=13, freq='M')[1:], forecast,
label='Forecast', color='red')
plt.title('Beer Production Forecast')
plt.legend()
plt.show()
```

Output:





Result:

The above program has been executed successfully.