Brent Crude Oil Prediction with ARIMA

The code snippet is a collection of functions that are used to model and forecast Brent Crude Oil prices using the ARIMA (Auto Regressive Integrated Moving Average) model. It includes functions for doing the following:

1. loading data
2. plotting data
3. testing stationarity
4. preprocessing data
5. checking seasonal differencing
6. plotting seasonal decomposition, plotting ACF (Auto Correlation Function) and PACF (Partial Auto Correlation Function) plots,
7. plotting seasonal data, evaluating stationarity, fitting the SARIMAX (Seasonal Auto Regressive Integrated Moving Average with exogenous regressors) model
8. forecasting future values
9. plotting the forecast with confidence intervals
10. calculating error metrics.

Example Usage

```python
data = load_data()
plot_data(data)
test_stationarity(data)
differenced_data = preprocess_data(data)
plot_differenced_data(differenced_data)
check_seasonal_differencing(data)
plot_seasonal_decomposition(differenced_data)
plot_acf_pacf_plots(differenced_data)
plot_seasonal_data(data)
evaluate_stationarity(differenced_data)
model = auto_arima_model(differenced_data)
plot_residuals(model)
results = fit_sarimax_model(differenced_data, model.order,
model.seasonal_order)
predictions, forecast_summary_90, forecast_summary_95 =
forecast_future_values(data['Price'], model.order, model.seasonal_order, 24)
plot_forecast_with_confidence_intervals(data['Price'], predictions,
forecast_summary_90, forecast_summary_95)
calculate_error_metrics(data['Price'], predictions)
```

Code Analysis

Inputs

- `data` : A pandas DataFrame containing the Brent Crude Oil prices from 2002 to 2022

Flow

1. The `load_data()` function is called to load the data from a CSV file.
2. The `plot_data()` function is called to plot the Brent Crude Oil prices.
3. The `test_stationarity()` function is called to test the stationarity of the data.
4. The `preprocess_data()` function is called to preprocess the data by differencing it.
5. The `plot_differenced_data()` function is called to plot the differenced data.
6. The `check_seasonal_differencing()` function is called to check if seasonal differencing is required.
7. The `plot_seasonal_decomposition()` function is called to plot the seasonal decomposition of the differenced data.
8. The `plot_acf_pacf_plots()` function is called to plot the ACF and PACF plots of the differenced data.
9. The `plot_seasonal_data()` function is called to plot the seasonal data.
10. The `evaluate_stationarity()` function is called to evaluate the stationarity of the differenced data.
11. The `auto_arima_model()` function is called to automatically determine the parameters for the ARIMA model.
12. The `plot_residuals()` function is called to plot the residuals of the ARIMA model.
13. The `fit_sarimax_model()` function is called to fit the SARIMAX model to the differenced data.
14. The `forecast_future_values()` function is called to forecast future values using the SARIMAX model.
15. The `plot_forecast_with_confidence_intervals()` function is called to plot the forecast with confidence intervals.
16. The `calculate_error_metrics()` function is called to calculate the error metrics (MAE and MAPE) for the forecasted values.

Outputs

- Various plots of the data, differenced data, seasonal decomposition, ACF and PACF plots, seasonal data, residuals, and forecast with confidence intervals.
- Logging messages with information about the stationarity of the data, the parameters of the ARIMA model, the summary of the SARIMAX model, and the error metrics.