ARIMA R/Quarto Pipeline

AUTHOR

Farooq Mahmud

Load time series

```
data <- read.csv(file.path(getwd(), "../data", "time_series.csv"))
data$pickup_date <- as.Date(data$pickup_date)
data <- data[order(data$pickup_date), ]

h <- 14          # 14-day forecast (same period as LSTM for comparison)
VAL_DAYS <- 30   # match LSTM: 30 validation + 14 test held out

# Training series: same as LSTM (first 322 days; then 30 validation + 14 test)
n_train <- nrow(data) - VAL_DAYS - h
ts_data_train <- ts(data$avg_duration_min[1:n_train], frequency = 366, start = c(2024, 1))
```

Model Selection

```
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```
auto_fit <- auto.arima(ts_data_train)
auto_fit
```

```
Series: ts_data_train
ARIMA(1,1,4)

Coefficients:
         ar1      ma1     ma2      ma3     ma4
      0.5180  -0.9430  0.0031  -0.2887  0.3019
s.e.  0.1747   0.1699  0.1054   0.0784  0.0584

sigma^2 = 0.852:  log likelihood = -428
AIC=868.01   AICc=868.27   BIC=890.63
```

14-Day Forecast

```
# Forecast 44 steps (validation + test) so last 14 match LSTM test period
forecast_result <- forecast(auto_fit, h = VAL_DAYS + h)
```

Table of Forecasted Values

```r
# Same 14-day period as LSTM (last 14 days of 2024) for fair comparison
forecast_full <- as.data.frame(forecast_result)
# Use last 14 of the 44-step forecast (steps 31:44 = LSTM test period)
forecast_summary <- forecast_full[(VAL_DAYS + 1):(VAL_DAYS + h), ]
forecast_dates <- data$pickup_date[(nrow(data) - h + 1):nrow(data)]

# Save forecast to CSV for comparison with LSTM

formatted_forecast <- cbind(Date = forecast_dates, forecast_summary)

formatted_forecast <- within(formatted_forecast, {
  `Point Forecast` <- round(`Point Forecast`, 2)
  `Lo 80` <- round(`Lo 80`, 2)
  `Hi 80` <- round(`Hi 80`, 2)
  `Lo 95` <- round(`Lo 95`, 2)
  `Hi 95` <- round(`Hi 95`, 2)
})

arima_forecast_df <- data.frame(
  date = forecast_dates,
  actual = data$avg_duration_min[(nrow(data) - h + 1):nrow(data)],
  forecast = formatted_forecast$`Point Forecast`
)

print(formatted_forecast)
```

```
                Date Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
2024.9617 2024-12-18          20.71 19.02 22.40 18.13 23.29
2024.9645 2024-12-19          20.71 19.01 22.41 18.12 23.31
2024.9672 2024-12-20          20.71 19.00 22.42 18.10 23.32
2024.9699 2024-12-21          20.71 18.99 22.43 18.09 23.33
2024.9727 2024-12-22          20.71 18.99 22.44 18.07 23.35
2024.9754 2024-12-23          20.71 18.98 22.45 18.06 23.36
2024.9781 2024-12-24          20.71 18.97 22.45 18.04 23.38
2024.9809 2024-12-25          20.71 18.96 22.46 18.03 23.39
2024.9836 2024-12-26          20.71 18.95 22.47 18.02 23.41
2024.9863 2024-12-27          20.71 18.94 22.48 18.00 23.42
2024.9891 2024-12-28          20.71 18.93 22.49 17.99 23.43
2024.9918 2024-12-29          20.71 18.92 22.50 17.97 23.45
2024.9945 2024-12-30          20.71 18.91 22.51 17.96 23.46
2024.9973 2024-12-31          20.71 18.90 22.52 17.95 23.47
```

```r
write.csv(arima_forecast_df, file.path(getwd(), "../data", "arima_forecast_14day.csv"), row.names
```

Evaluation: sMAPE and MASE

```r
library(Metrics)
```

```

Attaching package: 'Metrics'

The following object is masked from 'package:forecast':

    accuracy
```

```r
library(yardstick)
```

```

Attaching package: 'yardstick'

The following objects are masked from 'package:Metrics':

    accuracy, mae, mape, mase, precision, recall, rmse, smape

The following object is masked from 'package:forecast':

    accuracy
```

```r
train_vals <- data$avg_duration_min[1:n_train]
mae_train <- mean(abs(diff(train_vals)))

# Metrics::smape returns proportion; multiply by 100 for percentage
arima_smape <- Metrics::smape(arima_forecast_df$actual, arima_forecast_df$forecast) * 100

arima_mase <- yardstick::mase_vec(
  truth = arima_forecast_df$actual,
  estimate = arima_forecast_df$forecast,
  m = 1,
  mae_train = mae_train
)

cat("ARIMA sMAPE (%):", round(arima_smape, 4), "\n")
```

```
ARIMA sMAPE (%): 7.5833
```

```r
cat("ARIMA MASE:", round(arima_mase, 4), "\n")
```

```
ARIMA MASE: 1.9344
```