ARIMA Model for Time Series Forecasting

Week 1 Assignment — Report

Introduction

This report summarises the implementation and results of an ARIMA-based time series forecasting exercise performed on historical financial data (daily closing prices). The aim was to understand ARIMA modeling components and practical steps needed to build, validate, and interpret an ARIMA forecast pipeline.

ARIMA: components and assumptions

ARIMA (AutoRegressive Integrated Moving Average) is denoted ARIMA(p, d, q):

- **AR(p)**: auto-regressive part — the model regresses current value on p lagged observations.
- **I(d)**: differencing order — number of times the series is differenced to induce stationarity.
- **MA(q)**: moving-average part — model expresses current value as a linear combination of q past forecast errors.

Key assumptions:

- The (transformed/differenced) series should be stationary: constant mean, variance and autocovariance over time.
- Residuals after fitting should be (approximately) white noise (no autocorrelation and near-normality for some inferential uses).

Stationarity, differencing, and tests

Stationarity is central: AR and MA components assume a stationary process. In practice:

- I used the Augmented Dickey–Fuller (ADF) test to test for a unit root. A p-value < 0.05 suggests stationarity.
- If ADF indicates non-stationarity, first differencing ($d = 1$) is commonly applied and re-tested. Higher-order differencing is rarely needed and risks overdifferencing.

Differencing removes trends and stabilizes the mean; sometimes a log transform is used first to stabilize variance.

ACF/PACF and parameter selection

Autocorrelation function (ACF) and partial-autocorrelation function (PACF) plots guide selection of p and q:

- ACF cutting off after lag q and decaying PACF suggests an MA(q) process.
- PACF cutting off after lag p and decaying ACF suggests an AR(p) process.

In the notebook I:

1. Plotted ACF and PACF on the training series (and on differenced series if differencing was applied).
2. Performed a grid-search over small ranges of p and q (e.g. 0–3) and d (0–2), choosing the model with the lowest AIC.
3. Optionally cross-checked with `pmdarima.auto_arima` when available.

Implementation notes and choices

Important practical steps included:

- **Data preprocessing:** downloaded data (Yahoo Finance), ensured `DatetimeIndex`, resampled to business-day frequency, and forward-filled missing values.
- **Train/test split:** chronological split with 80% training and 20% testing.
- **Model fitting:** used `statsmodels`' ARIMA implementation and selected the best (p, d, q) by AIC during grid search.
- **Forecasting:** used the fitted model to forecast the test period; computed confidence intervals when available.
- **Evaluation:** MAE, MSE and RMSE were computed to quantify forecast errors.
- **Walk-forward validation:** implemented an optional one-step-ahead walk-forward retraining to obtain a more realistic evaluation of rolling forecasts.
- **Diagnostics:** residual plots, residual ACF, QQ-plot and Ljung–Box test were used to check residual whiteness and normality.

Interpretation of plots and diagnostics

(Place the corresponding plots from the notebook near these captions in your submission.)

- **Time series plot:** Shows trend, volatility, and any visible seasonality. A clear trend motivated differencing.
- **ACF/PACF plots:** Guided initial guesses for p and q. For example, a slow decay in ACF suggested differencing; a PACF spike at lag 1 suggested $p \geq 1$.

- **Forecast vs actuals:** Visual comparison shows how well the model tracks test values; divergence during volatile periods indicates model limitations.
- **Residual plot and residual ACF:** Residuals should scatter around zero with no obvious pattern; residual ACF should show no significant autocorrelations.
- **QQ-plot and Ljung–Box test:** QQ-plot assesses approximate normality of residuals; Ljung–Box tests for autocorrelation in residuals. Failure of these checks means model assumptions may be violated.

Challenges, observations and learnings

- **Non-stationarity:** many financial series require differencing and sometimes variance stabilization (log transform).
- **Model selection:** ACF/PACF rules-of-thumb are helpful but not definitive; AIC-based grid search provides a data-driven selection.
- **Overfitting risk:** larger p, q can fit training data well but perform poorly on test data; parsimony is important.
- **Residual structure:** when residuals exhibit autocorrelation, consider alternative specifications (higher orders, seasonal terms, or exogenous regressors).
- **Computation and robustness:** ARIMA fitting can fail for some parameter combinations; robust code must skip non-converging fits.

Conclusions and possible improvements

ARIMA is a solid, interpretable baseline for forecasting stationary or near-stationary series and produces reliable short-term forecasts when assumptions hold. However, its linear form limits modeling of complex patterns, non-linearities, and regimes. Suggested improvements:

- **Seasonality:** use SARIMA or SARIMAX when seasonality is present.
- **Exogenous features:** include external regressors (volume, indicators) via SARIMAX.
- **Non-linear models:** consider machine learning / deep learning models (e.g., LSTM, XGBoost) or hybrid approaches.
- **Model selection automation:** use `auto_arima` for a fast search across candidate models.
- **Robust validation:** use rolling/walk-forward evaluation for realistic performance estimates.

Overall: the exercise reinforced theoretical understanding of ARIMA and exposed practical aspects of building, validating, and diagnosing time series models. The next steps would include exploring seasonal extensions and comparing ARIMA to non-linear models (LSTM) for richer pattern capture.