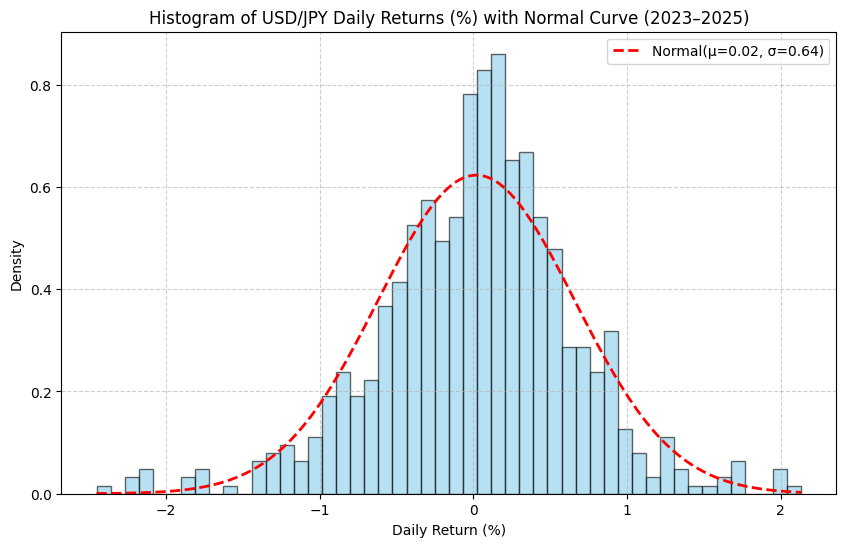
Minix Trading Project

# USD/JPY Swing Trading Strategy –

# Technical vs. ML-Enhanced

The project analyzes a swing trading strategy on the USD/JPY currency pair, comparing a pure technical-indicator approach against one augmented by machine learning. Swing trading targets short- to medium-term price moves (days to weeks) by entering near support/resistance turns and riding the dominant trend[[1]](https://www.investopedia.com/terms/s/swingtrading.asp#:~:text=What%20Is%20Swing%20Trading%3F). This case study calculates a wide range of technical indicators (e.g. moving averages, RSI, MACD, Bollinger Bands, etc.), generates entry/exit signals from logical rules, and then filters those signals using a Random Forest model. Finally, both strategies are backtested on historical USD/JPY data (2016–2025) to compare performance metrics such as total return, win rate, drawdown and Sharpe ratio.

Data and Pipeline



The histogram of daily USD/JPY returns from 2023 to 2025, overlaid with a fitted normal distribution, provides insights into the statistical properties of the underlying time series. The distribution of returns is centered around zero, with the fitted mean estimated at 0.02%. This is consistent with the general behavior of FX markets, where exchange rate changes tend to exhibit no persistent drift over long horizons once carry effects are excluded. The estimated daily volatility of 0.64% reflects the typical level of variation in USD/JPY and highlights the relatively liquid and stable nature of the currency pair.

While the fitted normal distribution provides a reasonable approximation of the central mass of the data, deviations are observed in the tails. The histogram reveals slightly fatter tails than the Gaussian curve, particularly on the downside. This indicates that extreme events, such as sharp policy moves or unexpected macroeconomic shocks, occur more frequently than would be predicted by a normal model. These tail risks are well documented in financial time series and highlight the importance of robust risk management frameworks for any trading strategy.

From a pairs trading perspective, the near-zero mean return suggests that directional bets on USD/JPY are unlikely to provide consistent profitability. Instead, performance must rely on identifying relative mispricings through statistical signals rather than through directional bias. The relatively low but non-negligible daily volatility makes the pair attractive for mean-reversion strategies, while the presence of fat tails reinforces the need to account for rare but impactful shocks. Overall, the analysis confirms that while USD/JPY returns exhibit a broadly normal distribution, reliance on Gaussian assumptions alone would underestimate risk, and strategies must incorporate measures that explicitly handle excess kurtosis and tail behavior.

The pipeline begins by loading and cleaning the USD/JPY daily price data (2016–08-15 to 2025–08-13). Price fields (Open/High/Low/Close) are converted to numeric, and percentage changes normalized. We then compute a series of technical indicators to capture trend, momentum, volatility and volume signals. These indicators are added as columns to the data frame; overall, 29 new indicator columns were generated in the code. Next, we apply rule-based logic to produce swing trading signals:

* **Long signal:** e.g. MACD histogram positive, RSI moderate (30–70), price above 20-day MA, stochastic not too high, price near lower Bollinger band, etc., *and* 20-day MA above 50-day MA (strong uptrend).
* **Short signal:** opposite conditions (MACD negative, RSI not oversold, price below 20-day MA, etc.) with 20-day MA below 50-day (downtrend).

Exit signals are also defined (e.g. RSI extreme or MACD crossing back) to close positions. All signals and indicator values are stored in the prepared data.

This pipeline reflects common swing trading practice: indicators like moving averages identify trend (price above its MA suggests uptrend), RSI flags overbought/oversold (RSI>70 often overbought, <30 oversold), MACD tracks momentum shifts, Bollinger Bands measure volatility (price at upper band often “overbought”), ATR measures volatility of price range, and OBV (On-Balance Volume) accumulates volume flow to gauge buying/selling pressure. Stochastic %K and swing highs/lows help detect short-term reversals. The strategy requires multiple conditions to align, aiming to filter out false signals and trade only in clear trend/momentum setups.

Technical Indicators

This pipeline combines multiple indicators to improve signal quality and reduce noise.

1. **Moving Averages (MA):** These smooth price data highlight overall trend direction. Price trading above its moving average generally suggests an uptrend, while price below the average suggests a downtrend [[2]](https://www.oanda.com/us-en/trade-tap-blog/trading-knowledge/identify-trends-with-moving-averages/#:~:text=Why%20use%20moving%20averages%20in,trading).
2. **Relative Strength Index (RSI):** A momentum oscillator that tracks the speed and change of price moves. RSI values above 70 are often considered *overbought* (potential pullback), while values below 30 are considered *oversold* (potential rebound) [[3]](https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/RSI#:~:text=The%20Relative%20Strength%20Index%20,used%20to%20identify%20the%20general).
3. **MACD (Moving Average Convergence Divergence):** Measures momentum shifts by comparing two moving averages of price. A bullish signal occurs when the MACD line crosses above its signal line, while a bearish signal occurs when it crosses below [[4]](https://www.oanda.com/us-en/learn/indicators-oscillators/determining-entry-and-exit-points-with-macd/#:~:text=The%20Moving%20Average%20Convergence%20Divergence,entering%20and%20exiting%20a%20trade).
4. **Bollinger Bands:** Built around a moving average with upper and lower bands set by standard deviations, these capture volatility. Price reaching the upper band can indicate an “overbought” market, while touching the lower band may indicate “oversold” [[5]](https://www.investopedia.com/terms/b/bollingerbands.asp#:~:text=match%20at%20L350%20,Bollinger%20Band%2C%20it%20might%20be).
5. **ATR (Average True Range):** Captures market volatility by averaging the range of recent price movements. Higher ATR values reflect larger swings, while lower values suggest calmer markets [[6]](https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/atr#:~:text=Average%20True%20Range%20,use%2020%20to%2050%20periods).
6. **OBV (On-Balance Volume):** Accumulates trading volume by adding volume on up days and subtracting it on down days. Rising OBV signals buying pressure; declining OBV signals selling pressure [[7]](https://www.investopedia.com/terms/o/onbalancevolume.asp#:~:text=On,They%20are).
7. **Stochastic Oscillator (%K):** Compares the closing price to its recent trading range. High readings near 100 mean prices closed near the highs of the range, while readings near 0 mean prices closed near the lows—useful for spotting potential reversals.
8. **Swing Highs and Lows:** These are recent price peaks (highs) and troughs (lows). Traders use them to identify breakouts, breakdowns, or reversal levels, adding structure to entry and exit decisions.

Machine Learning Enhancement – Random Forest

To refine the signals, a Random Forest model is trained on historical features and price movements. We derive 23 predictive features from the indicator data (e.g. current RSI, MACD values, Bollinger position, ATR/price, recent returns lagged 1–5 days, RSI lagged, volatility ratios, etc.). The model has two parts: a classifier predicting the next-day price direction (up or down), and a regressor predicting the next-day return. Training uses 80% of the data. Random Forests are known for high accuracy and robustness (ensemble of decision trees) and provide *feature importance* rankings[[8]](https://blog.quantinsti.com/random-forest-algorithm-in-python/#:~:text=,provide%20insights%20into%20the%20underlying). In our run the direction model achieved ~54% accuracy, and the top features included normalized ATR, past returns, volatility ratio and Bollinger band width.

After training, we apply the model to generate ML-enhanced signals. Specifically, an original long signal is accepted only if the model’s predicted up probability exceeds 60% *and* the predicted return exceeds 0.2%. Similarly, short signals require low up-probability (<40%) and predicted return < –0.2%. This filter drastically reduces trades (from 125 original longs to ~20 ML-long signals), aiming to improve the win rate. In practice, Random Forests *can* improve strategy performance by highlighting when indicator signals are likely to succeed[[8]](https://blog.quantinsti.com/random-forest-algorithm-in-python/#:~:text=classification%20and%20regression%20tasks%2C%20as,insights%20into%20the%20underlying%20relationships). The feature importance output (e.g. ATR\_Normalized, return lag) also gives insight into which factors drive predictions.

Backtesting and Performance Metrics

Both strategies are backtested on the same historical data using a simple entry/exit simulation. Each trade uses all capital (no leverage) and re-enters only after exit. Key performance metrics are computed for each strategy: total return, number of trades, win rate, average win/loss, profit factor, maximum drawdown and Sharpe ratio.

1. **Win Rate:** Fraction of trades with positive P/L. A higher win rate (closer to 1) means more frequent wins[[9]](https://www.quantifiedstrategies.com/trading-performance/#:~:text=,free%20rate%20to%20standard%20deviation).
2. **Profit Factor:** Ratio of gross profit to gross loss; >1 indicates profitable overall. It measures how many dollars are earned per dollar lost[[9]](https://www.quantifiedstrategies.com/trading-performance/#:~:text=%2A%20Sharpe%20Ratio%3A%20Assesses%20risk,profitability%20per%20unit%20of%20risk).
3. **Max Drawdown:** The largest peak-to-trough decline in equity, indicating risk.
4. **Sharpe Ratio:** Risk-adjusted return (excess return per unit volatility)[[9]](https://www.quantifiedstrategies.com/trading-performance/#:~:text=What%20is%20the%20Sharpe%20Ratio%2C,it%20widely%20used%20in%20trading); higher is better.

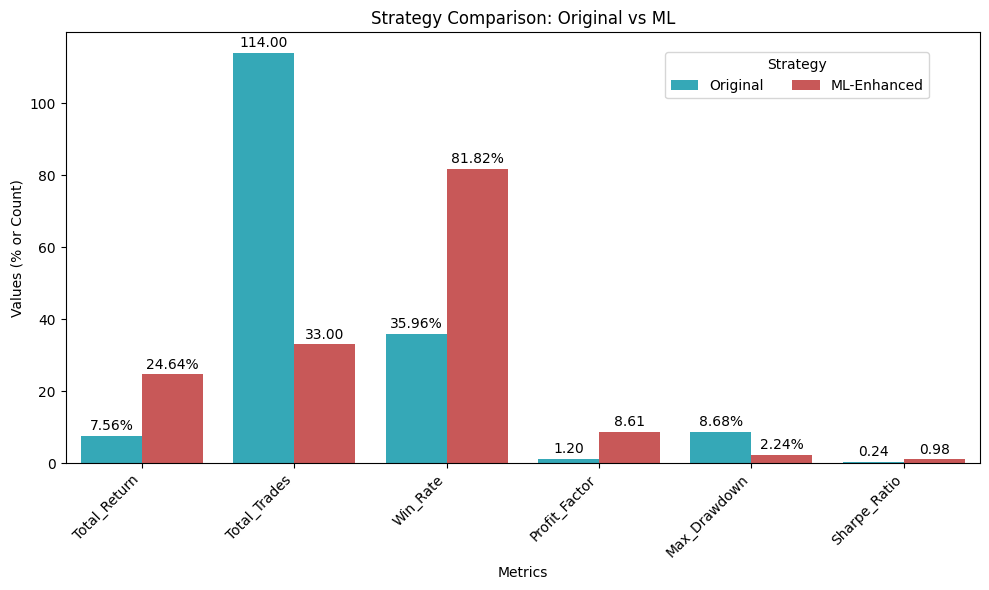
The code’s backtest (starting capital $10,000) yielded:

**Original Technical Strategy:** Total return ~7.56%, 114 trades, win rate ~35.96%, profit factor ~1.20, max drawdown ~8.68%, Sharpe ~0.24.

**ML-Enhanced Strategy:** Total return ~24.64%, 33 trades, win rate ~81.82%, profit factor ~8.61, max drawdown ~2.24%, Sharpe ~0.98.

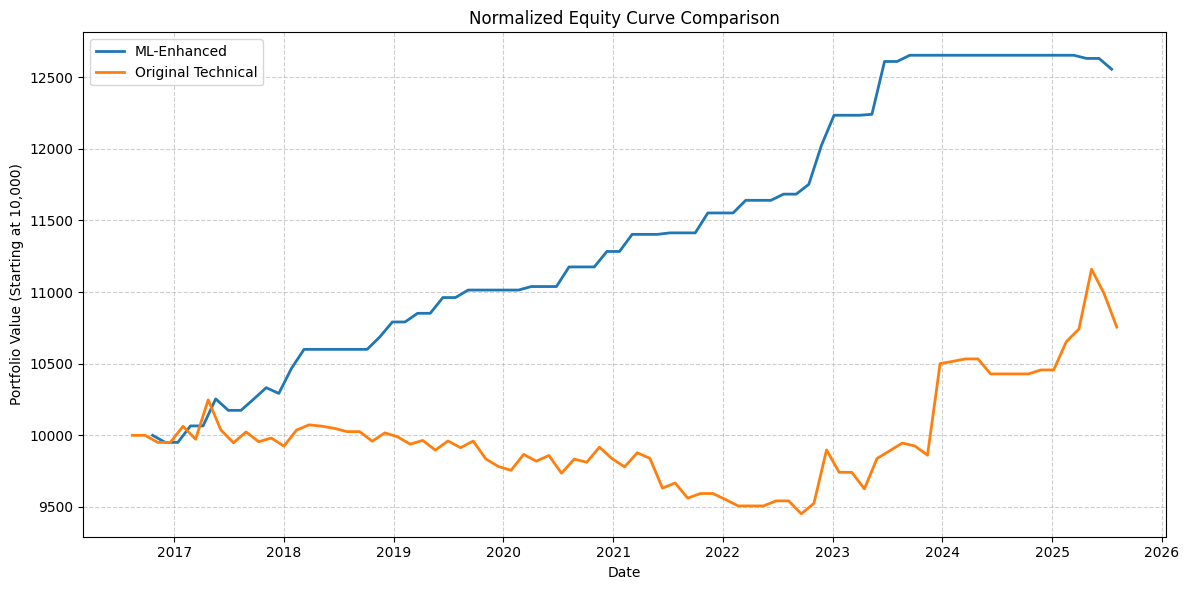
These results (see metrics table below) show the ML filter greatly improved performance: far higher win rate and profit factor, lower drawdown, and much higher Sharpe[[10]](file://file-L6chpQoFWYW4oZiDHgUdN2#:~:text=,981358). For example, profit factor jumped from ~1.20 to ~8.61 (each $1 lost vs $8.61 gained[[9]](https://www.quantifiedstrategies.com/trading-performance/#:~:text=%2A%20Sharpe%20Ratio%3A%20Assesses%20risk,profitability%20per%20unit%20of%20risk)), and drawdown fell by ~75%. The equity curve (portfolio value over time) for the ML strategy climbs steadily above that of the original strategy.

|  |  |  |
| --- | --- | --- |
| Metric | Original | ML-Enhanced |
| Total Return | 0.0756 | 0.2464 |
| Total Trades | 114 | 33 |
| Win Rate | 0.3596 | 0.8182 |
| Profit Factor | 1.2018 | 8.6102 |
| Max Drawdown | 0.0868 | 0.0224 |
| Sharpe Ratio | 0.2373 | 0.9814 |

**

*(All metrics from backtest; ML-enhancement dramatically improves risk-adjusted returns*[*[10]*](file://file-L6chpQoFWYW4oZiDHgUdN2#:~:text=,981358)*.)*

Results and Analysis



The ML-enhanced strategy clearly outperformed the pure technical strategy. The large improvement in win rate (≈82% vs 36%) means the ML model filtered out many losing trades. The profit factor jumped to 8.61, indicating very few losses compared to gains, whereas the original strategy barely broke even (PF≈1.20). Lower drawdown (2.24% vs 8.68%) shows the ML approach kept equity near new highs with fewer deep drops. The Sharpe ratio (~0.98) is much higher, reflecting better return per unit risk[[9]](https://www.quantifiedstrategies.com/trading-performance/#:~:text=What%20is%20the%20Sharpe%20Ratio%2C,it%20widely%20used%20in%20trading).

Importantly, the ML-enhanced approach achieved this with far fewer trades (33 vs 114), meaning it skipped many marginal setups. The Random Forest essentially learned to trust only the strongest signals (high predicted return and probability). This is a key benefit of combining machine learning: even with just ~54% classification accuracy, filtering rules dramatically boosted outcomes by reducing unprofitable trades (ensemble models like RF are known to handle complex patterns and rank features by importance[[8]](https://blog.quantinsti.com/random-forest-algorithm-in-python/#:~:text=,provide%20insights%20into%20the%20underlying)).

Conclusion

In summary, the case study shows that layering ML classification/regression on top of a well-defined swing trading strategy can significantly improve performance. The technical strategy alone produces modest gains, whereas the ML filter yields higher returns and much better risk metrics (higher win rate, profit factor, Sharpe ratio; lower drawdown)[[10]](file://file-L6chpQoFWYW4oZiDHgUdN2#:~:text=,981358)[[9]](https://www.quantifiedstrategies.com/trading-performance/#:~:text=%2A%20Sharpe%20Ratio%3A%20Assesses%20risk,profitability%20per%20unit%20of%20risk). Key components included computing standard indicators (moving averages, RSI, MACD, etc.) and applying logical entry/exit rules, then training a Random Forest to learn when those signals are most reliable.

This exercise illustrates a typical quant workflow: feature engineering (indicators), rule-based signals, machine learning for signal selection, and rigorous backtesting. Each metric helps quantify the effect. For example, a Sharpe ratio near 1.0 is generally considered good for a trading strategy[[9]](https://www.quantifiedstrategies.com/trading-performance/#:~:text=What%20is%20the%20Sharpe%20Ratio%2C,it%20widely%20used%20in%20trading), and an 8.6 profit factor is outstanding. Although the ML model had moderate raw accuracy (~0.54), its use of probability thresholds and predicted returns meant only high-confidence trades were taken, which greatly boosted overall efficiency.

Keytakeaways

Technical indicators (RSI, MACD, Bollinger Bands, ATR, OBV, MAs) serve as the foundation of the swing strategy[[3]](https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/RSI#:~:text=The%20Relative%20Strength%20Index%20,used%20to%20identify%20the%20general)[[5]](https://www.investopedia.com/terms/b/bollingerbands.asp#:~:text=match%20at%20L350%20,Bollinger%20Band%2C%20it%20might%20be). A Random Forest can then rank features (ATR, past returns, volatility measures) and filter signals to improve profitability[[8]](https://blog.quantinsti.com/random-forest-algorithm-in-python/#:~:text=,provide%20insights%20into%20the%20underlying). Metrics like win rate, profit factor and Sharpe ratio provide clear evidence that the ML-enhanced strategy is superior in this backtest. These findings would be highlighted when presenting this project, emphasizing how data-driven methods and machine learning can refine traditional trading rules.

[[1]](https://www.investopedia.com/terms/s/swingtrading.asp#:~:text=What%20Is%20Swing%20Trading%3F) [What Is Swing Trading?](https://www.investopedia.com/terms/s/swingtrading.asp)

[[2]](https://www.oanda.com/us-en/trade-tap-blog/trading-knowledge/identify-trends-with-moving-averages/#:~:text=Why%20use%20moving%20averages%20in,trading) [Moving averages for trend-following trading strategies | Tools and Strategies](https://www.oanda.com/us-en/trade-tap-blog/trading-knowledge/identify-trends-with-moving-averages/)

[[3]](https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/RSI#:~:text=The%20Relative%20Strength%20Index%20,used%20to%20identify%20the%20general) [What is RSI? - Relative Strength Index - Fidelity](https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/RSI)

[[4]](https://www.oanda.com/us-en/learn/indicators-oscillators/determining-entry-and-exit-points-with-macd/#:~:text=The%20Moving%20Average%20Convergence%20Divergence,entering%20and%20exiting%20a%20trade) [Moving Average Convergence Divergence (MACD) | Learn to Trade](https://www.oanda.com/us-en/learn/indicators-oscillators/determining-entry-and-exit-points-with-macd/)

[[5]](https://www.investopedia.com/terms/b/bollingerbands.asp#:~:text=match%20at%20L350%20,Bollinger%20Band%2C%20it%20might%20be) [Understanding Bollinger Bands: A Key Technical Analysis Tool for Investors](https://www.investopedia.com/terms/b/bollingerbands.asp)

[[6]](https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/atr#:~:text=Average%20True%20Range%20,use%2020%20to%2050%20periods) [What Is Average True Range? - Fidelity](https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/atr)

[[7]](https://www.investopedia.com/terms/o/onbalancevolume.asp#:~:text=On,They%20are) [On-Balance Volume (OBV): How It Works and How to Use It](https://www.investopedia.com/terms/o/onbalancevolume.asp)

[[8]](https://blog.quantinsti.com/random-forest-algorithm-in-python/#:~:text=,provide%20insights%20into%20the%20underlying) [Random Forest Algorithm In Trading Using Python](https://blog.quantinsti.com/random-forest-algorithm-in-python/)

[[9]](https://www.quantifiedstrategies.com/trading-performance/#:~:text=,free%20rate%20to%20standard%20deviation) [Trading Performance: Strategy Metrics, Risk-Adjusted Metrics, And Backtest - QuantifiedStrategies.com](https://www.quantifiedstrategies.com/trading-performance/)

[[10]](file://file-L6chpQoFWYW4oZiDHgUdN2#:~:text=,981358) [Source code](https://github.com/vkorde3/pairs-trading/blob/main/pairs_trading.ipynb)