Enhanced Short Strangle NIFTY Options Trading System V4

Production-Quality Backtesting Edition

Overview

This is a complete options trading system with production-quality backtesting capabilities. It downloads historical data from Zerodha Kite API, applies your short strangle strategy, and provides detailed performance metrics.

📁 File Structure

```
project/
├── short_strangle_system_V4_PRODUCTION.py  # Main trading system
├── historical_data_manager.py              # Historical data downloader
├── access_token.txt                        # Kite API token (auto-generated)
├── trades_database.db                      # SQLite database for trades
├── strangle_trading.log                    # Detailed logs
├── backtest_cache/                         # Cached historical data
│   ├── instruments/                        # NFO instruments cache
│   ├── nifty/                              # NIFTY spot data cache
│   ├── vix/                                # VIX data cache
│   └── options/                            # Option prices cache
├── backtest_data_production.csv            # Complete backtest dataset
└── backtest_trades_*.csv                   # Exported trade results
```

🚀 Installation

Requirements

bash

```
pip install pandas numpy kiteconnect colorama tabulate requests
```

Setup

1. Place both Python files in the same directory
2. Update API credentials in `Config` class:

python

```
API_KEY = "your_api_key"
API_SECRET = "your_api_secret"
```

3. (Optional) Configure Telegram alerts for notifications

📊 Running a Backtest

Step 1: Start the System

bash

```
python short_strangle_system_V4_PRODUCTION.py
```

Step 2: Select Backtest Mode

```
Select Trading Mode:
1. Paper Trading (Default)
2. Live Trading
3. Backtest Mode
Enter choice (1/2/3): 3
```

Step 3: Enter Date Range

Enter start date (YYYY-MM-DD): 2024-01-01

Enter end date (YYYY-MM-DD): 2024-03-31

Force refresh cached data? (yes/no, default: no): no

Step 4: Authenticate with Kite

- Browser will open with Kite login URL
- Log in and authorize the app
- Copy the `request_token` from the redirect URL
- Paste it in the terminal

Step 5: Wait for Data Download

The system will:

1. Download NIFTY spot prices (minute data)
2. Download India VIX data
3. Find option contracts for each trading day
4. Download option prices for CE and PE
5. Cache all data locally for future use

Note: First-time download may take 15-30 minutes depending on date range. Subsequent runs use cached data and are much faster!

Step 6: Review Results

The system will display:

- Real-time dashboard during backtest simulation
- Daily P&L summaries
- Final performance metrics
- Exported CSV files with detailed trades

📈 Understanding the Results

Console Output

Metric	Value
Total Trades	45
Win Trades	28
Win Rate	62.2%
Cumulative P&L	₹2,45,000
Max Drawdown	₹45,000
Profit Factor	1.85
Sharpe Ratio	1.42
Rolled Positions	3

Key Metrics Explained

Win Rate: Percentage of profitable trades **Cumulative P&L**: Total profit/loss over the backtest period **Max Drawdown**: Largest peak-to-trough decline **Profit Factor**: Gross profits / Gross losses (>1 is profitable) **Sharpe Ratio**: Risk-adjusted returns (>1 is good, >2 is excellent) **Rolled Positions**: Number of times positions were adjusted

---

📁 Exported Files

1. `backtest_data_production.csv`

Complete minute-by-minute data used for backtesting

- NIFTY spot prices
- VIX values
- CE and PE option prices
- Selected strikes and symbols

2. `backtest_trades_[date_range].csv`

All executed trades with:

- Entry/exit times and prices
- Strikes and symbols
- P&L per trade
- Rolled position indicators

3. `backtest_daily_performance_[date_range].csv`

Daily performance summary:

- Daily P&L
- Number of trades
- Win rate
- Drawdown tracking

4. `trades_database.db`

SQLite database containing:

- Individual trade records
- Daily performance metrics
- Historical analysis data

🔧 Configuration Parameters

Strategy Parameters (in `Config` class)

python

```
# Capital & Position Sizing
CAPITAL = 1000000            # Total trading capital
BASE_LOTS = 50              # Standard lot size
REDUCED_LOTS = 25           # Lots during high VIX


# Strike Selection
OTM_DISTANCE_NORMAL = 250     # Points OTM in normal VIX
OTM_DISTANCE_HIGH_VIX = 350   # Points OTM in high VIX


# VIX Thresholds
VIX_THRESHOLD = 20.0         # VIX level to reduce position
VIX_LOW_THRESHOLD = 15.0      # Minimum VIX for entry
VIX_HIGH_THRESHOLD = 25.0     # Maximum safe VIX level


# Risk Management
PROFIT_TARGET_PCT = 0.50      # Take profit at 50% of premium
STOP_LOSS_PCT = 0.25         # Stop loss at 25% loss
TRAILING_STOP_PCT = 0.15      # Trail by 15% from peak profit
MAX_LOSS_ONE_LEG_PCT = 1.50   # Exit all if one leg loses 150%
ROLL_THRESHOLD_PCT = 0.75     # Roll at 75% loss


# Premium Limits
MIN_COMBINED_PREMIUM = 150    # Minimum total premium
MAX_COMBINED_PREMIUM = 300    # Maximum total premium


# IV Requirements
MIN_IV_PERCENTILE = 30       # Enter only if IV > 30th percentile
MAX_IV_PERCENTILE = 80       # Don't enter if IV > 80th percentile
```

🎯 Strategy Logic

Entry Conditions

1. VIX between LOW and HIGH thresholds
2. IV percentile in acceptable range (30-80%)
3. No existing active positions
4. Entry window (9:30 AM - 2:30 PM)
5. Combined premium within limits

Strike Selection

- **Low VIX**: Strikes closer to ATM (200 points OTM)

- **Normal VIX**: Standard distance (250 points OTM)
- **High VIX**: Conservative strikes (450 points OTM)

Exit Conditions

1. **Profit Target**: Close at 50% of initial premium
2. **Trailing Stop**: Exit if profit falls 15% from peak
3. **Stop Loss**: Emergency exit at 25% loss
4. **Leg-Specific Stop**: Exit all if one leg loses 150%
5. **Square Off**: Close all positions at 3:15 PM

Position Rolling

- Triggered when loss exceeds 75% of premium
- Rolls to next safer OTM strike (100 points further)
- Only rolls once per position

🔄 Cache Management

Using Cached Data

Force refresh cached data? (yes/no, default: no): no

Uses existing cached data (fast, recommended for repeat backtests)

Refreshing Cache

Force refresh cached data? (yes/no, default: no): yes

Downloads fresh data from Kite API (slow, use when:)

- First-time backtest
- Data might be corrupted
- Want most recent contract prices

Clearing Cache Manually

Delete the `backtest_cache/` directory to start fresh

🐛 Troubleshooting

"No tokens for {symbol}" Warning

Cause: Historical option contracts not found in instruments list **Solution**:

- Check if date range is too far in the past
- Ensure Kite API has data for that period
- Try refreshing cache with `force_refresh=True`

"Failed to fetch NIFTY/VIX data"

Cause: API rate limits or authentication issues **Solution**:

- Wait a few minutes and retry
- Check if access token is valid
- Verify API credentials

Empty Backtest Data

Cause: No matching option contracts found **Solution**:

- Reduce date range to recent months
- Check strike calculation logic
- Review logs for specific errors

Slow Performance

Cause: Downloading large amounts of data **Solution**:

- Use cached data (`force_refresh=no`)
- Reduce date range
- Run overnight for multi-year backtests

📊 Sample Backtest Run

bash

```
$ python short_strangle_system_V4_PRODUCTION.py


================================================================================

  ENHANCED SHORT STRANGLE NIFTY OPTIONS TRADING SYSTEM
  Version 4.0 - Production Backtest Edition

================================================================================


Select Trading Mode:
1. Paper Trading (Default)
2. Live Trading
3. Backtest Mode
Enter choice (1/2/3): 3

BACKTEST MODE
Enter start date (YYYY-MM-DD): 2024-03-01
Enter end date (YYYY-MM-DD): 2024-03-31
Force refresh cached data? (yes/no, default: no): no

Downloading and preparing historical data...
Cache mode: USE EXISTING

2024-03-01: ✓ NIFTY data loaded from cache
2024-03-01: ✓ VIX data loaded from cache
2024-03-01: ✓ Found CE/PE strikes
2024-03-01: ✓ Option data loaded

... [processing all dates] ...

Backtest data saved to: backtest_data_production.csv
Total data points: 8,432
Date range: 2024-03-01 09:15:00 to 2024-03-29 15:30:00

Starting backtest simulation...

Trading Day: 2024-03-01
```

```
╔═══════════════════════════╦═══════════════════╗
║ NIFTY Spot        ║ 22,147.00    ║
║ India VIX         ║ 16.25        ║
║ Active Trades     ║ 2            ║
║ Daily P&L         ║ ₹12,500      ║
╚═══════════════════════════╩═══════════════════╝
```

... [daily results] ...

==

BACKTEST SUMMARY - 2024-03-01 to 2024-03-31

==

```
╔═══════════════════════════════╗ ╔════════════════════════════════╗
║ Total Trades        ║ 24        ║
║ Win Rate            ║ 66.7%     ║
║ Cumulative P&L      ║ ₹1,25,400 ║
║ Max Drawdown        ║ ₹22,100   ║
║ Profit Factor       ║ 2.14      ║
║ Sharpe Ratio        ║ 1.68      ║
║ Trading Days        ║ 21        ║
║ Avg Daily P&L       ║ ₹5,971    ║
╚═══════════════════════════════╝ ╚════════════════════════════════╝
```

Backtest completed successfully!

💡 Tips for Best Results

1. **Start Small**: Test with 1-3 months first
2. **Use Cache**: Always use cached data for multiple iterations
3. **Check Logs**: Review `strangle_trading.log` for detailed insights
4. **Optimize Parameters**: Adjust VIX thresholds based on results
5. **Verify Data**: Spot-check the CSV files for data quality
6. **Multiple Scenarios**: Run backtests across different market conditions

---

🔐 Security Notes

- Never commit API credentials to version control
- Keep `access_token.txt` private
- Use paper trading to validate before live trading
- Always test thoroughly in backtest mode first

📞 Support

For issues or questions:

1. Check the logs: `strangle_trading.log`
2. Review exported CSV files for data quality
3. Verify API credentials and authentication

4. Ensure sufficient API rate limits

⚠️ Disclaimer

This system is for educational and research purposes. Options trading involves substantial risk. Always:

- Thoroughly backtest before live trading
- Understand the strategy completely
- Use appropriate position sizing
- Never risk more than you can afford to lose
- Consult with financial advisors

🎉 Happy Backtesting!

The system is now production-ready with:

- ✅ Historical data caching
- ✅ Robust error handling
- ✅ Detailed performance metrics
- ✅ CSV exports for analysis
- ✅ Real-time dashboard
- ✅ Database persistence

Run your backtest and analyze the results to optimize your strategy!