Code Documentation

October 8, 2024

The model implemented is a lookback structure model that has three assets viz. USD returns of gold, GBP returns of the FTSE100 and INR returns of the NIFTY50. The model looks back every quarter to find the asset that performed the best and second best and allocates 67% and 33% to these respectively. These assets are then delta hedged daily using put options. The life cycle of one such product is 3 years. The study covers the time period from October 2000 till October 2024 and analyses the performance of this model for 8 such life cycles.

1 Assumptions

The following are some of the assumptions in the implementation of the model.

1. In order to delta hedge the long positions in the stocks, put options have been used. However, historical options chain data is notoriously difficult to find for free. As a result, synthetic data has been generated to generate deltas and prices for the put options. To generate synthetic data, 22 day rolling volatility, 3 month t-bill rates and weekly expiry has been used as inputs to the Black Scholes Merton option pricing formula for ATM puts. The delta has also been calculated using these inputs.
2. Free data sources, namely yfinance and investing.com have been used to obtain the daily prices of the Nifty 50 FTSE 100 index and spot gold. ETF prices have not been used since ETF's were introduced roughly 3 years after the start date of the analysis.
3. Transaction costs and dividends have not been considered, the market has been assumed to be able to fill as many orders without any volume constraints or slippage.

2 Code Files

2.1 Data.py

Data.py is used for the generation of the dataset that will contain 24 years worth of data.The resulting DataFrame contains the following columns:

- Date: Date of the data points.
- open_nifty, open_ftse, open_gold: Opening prices for Nifty, FTSE, and Gold.
- put_price_nifty, put_price_ftse, put_price_gold: Calculated put option prices.
- delta_nifty, delta_ftse, delta_gold: Deltas of the options.

Five libraries need to be imported in order to run the file. yfinance and yahooquery are used for data fetching, pandas and numpy for data manipulation, and scipy.stats.norm for statistical functions used in the Black-Scholes model.

Data.py contains the following functions:

2.1.1 parse_dates(date_str)

Convert date in str format to pd.Datetime object

2.1.2 fetch_data(ticker, start_date, end_date, existing_df=None)

This function fetches historical data for specified tickers using the yahooquery library. It handles both the case where a dataset exists already and where new data needs to be appended.

2.1.3 filter_dates(df1, df2, date_column='Date')

This function aligns the data sets by filtering for common dates, ensuring that all data frames have matching dates for accurate analysis.

2.1.4 convert_val(df)

The function cleans up the data by converting columns that contain , and are of the type str to numerical values.

2.1.5 calculate_black_scholes_options(df)

This function performs the Black-Scholes calculations to determine the put option prices and deltas for Nifty, FTSE, and Gold. It creates two helper functions, black_scholes_put and black_scholes_delta, to calculate these values.

2.2 Main.py

This file code is designed to analyze the aforementioned investment startegy in financial assets (Nifty, FTSE, and Gold) over specific time periods using delta neutral hedging and asset ranking methods. It calculates the returns, ranks the assets based on performance, allocates capital accordingly, and then evaluates the performance using metrics like Sharpe ratio, Sortino ratio, cumulative returns, and maximum drawdown.
The required libraries are (pandas, numpy,datetime,math) for data manipulation and computation, and (matplotlib) in order to plot returns.
Main.py contains the following functions:

2.2.1 calculations(df)

This function calculates the daily and quarterly returns of Nifty, FTSE, and Gold. It uses percentage change for daily returns and resampling to find quarterly returns. It Returns the calculated quarterly returns for each asset.

2.2.2 rank_returns(nifty, ftse, gold)

Ranks the assets based on their quarterly returns and allocates 0.67 to the best asset and 0.33 to the second-best. It sorts the returns for each quarter to assign ranks and weightings. It returns Data frame with 0.67 and 0.33 allocations for each quarter.

2.2.3 find_columns(row)

It identifies the assets that have been allocated 0.67 and 0.33 weights in each quarter. This is extremely useful for manipulation using fstrings in the latter half of the code It returns a Data frame with the names of assets allocated the highest weights.

2.2.4 settle_opt(u67,u33)

Calculates the profit or loss (PnL) from put option positions at the end of each week. Settles options based on their final prices compared to their strike prices and determines the position taken (buy/sell) in order to find the PnL accrued from each option in the dictionary of options. Returns the total PnL from the options.

2.2.5 calc(start,end,col_67,col_33,df,init_cap)

This function implements the main logic for the computation of the daily positions in the two chosen assets and put option positions for daily delta hedging. Returns a subset of the original dataframe to correspond to the start and end of the quarter after computing the positions and their daily PnLs.

2.2.6 analyze(returns,rf,prices)

Calculates various risk and performance metrics such as Sharpe ratio, Sortino ratio, cumulative returns, CAGR, and Maximum Drawdown. Prints the performance metrics and plots the cumulative returns over time.