Carry Premium Project

from AlgorithmImports import \*

from datetime import timedelta

class BondCarryOptimization(QCAlgorithm):

def Initialize(self):

self.SetStartDate(2015, 1, 1)

self.SetEndDate(2024, 11, 8)

self.SetCash(100000)

# Add bond ETFs

self.bond\_etfs = {

'TLT': self.AddEquity('TLT', Resolution.Daily),

'LQD': self.AddEquity('LQD', Resolution.Daily),

'JNK': self.AddEquity('JNK', Resolution.Daily),

'EMB': self.AddEquity('EMB', Resolution.Daily)

}

# Monthly rebalance

self.Schedule.On(self.DateRules.MonthStart('TLT'), self.TimeRules.AfterMarketOpen('TLT'), self.Rebalance)

def Rebalance(self):

# Get yields for each ETF (approximation using dividend yield or price changes as a proxy)

yields = {symbol: self.GetYield(symbol) for symbol in self.bond\_etfs.keys()}

# Calculate weights based on optimization logic

# Here, we prioritize high yield (carry) while keeping diversification

total\_yield = sum(yields.values())

weights = {symbol: yield\_val / total\_yield for symbol, yield\_val in yields.items()}

# Set portfolio weights

for symbol, weight in weights.items():

self.SetHoldings(symbol, weight)

self.Debug(f"Setting {symbol} weight to {weight:.2%}")

def GetYield(self, symbol):

# Retrieve the dividend history for the past year

dividend\_history = self.History(Dividend, symbol, timedelta(days=365), Resolution.Daily)

# Check if dividend history has data and contains a 'dividends' column

if not dividend\_history.empty and 'dividends' in dividend\_history.columns:

total\_dividends = dividend\_history['dividends'].sum()

current\_price = self.Securities[symbol].Price

return total\_dividends / current\_price if current\_price > 0 else 0

else:

# Return a placeholder yield if no dividend data is available

return 0.03 # Placeholder default yield

def OnData(self, data):

# Handle data events, check for new rebalance, etc.

pass

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated