

MARKET TIMING OPTIMIZATION QUANT FINANCE MODEL

Quantitative Finance Guide

By [@deepthinksfinance](#)

STEP-BY-STEP PROMPT GUIDE

For Complete Beginners — Google Colab Edition

Build your own AI-powered model that tells you WHEN to invest in the market

WHAT YOU WILL BUILD

A complete Quantitative Finance model using Python that:

- Downloads 8 years of real NIFTY 50 market data automatically
- Calculates Momentum, Volatility & Drawdown signals
- Creates a composite Entry Score to time the market
- Runs a backtest comparing your strategy vs Buy & Hold
- Shows interactive charts of portfolio growth & buy zones
- Outputs Total Return, CAGR & Max Drawdown metrics

SECTION 1 — INTRODUCTION

1.1 What Is Market Timing?

Market timing is the practice of deciding WHEN to enter or exit the stock market based on data, rather than staying fully invested at all times. Instead of asking 'Which stock should I buy?', this model answers:

"When is the best time to put my money into the market?"

1.2 What Is This Model? (Plain English)

This model reads NIFTY 50 price data every day and calculates three signals:

Signal	What It Measures
Momentum	Is the market trending upward or downward?
Volatility	How wild / risky is the market right now?
Drawdown	How far has the market fallen from its peak?

It then combines these into a single Entry Score (0 to 1). A high score means 'good time to invest heavily.' A low score means 'be cautious.'

1.3 Why Use Google Colab?

Google Colab is a free, browser-based Python notebook. You do not need to install anything on your computer. All you need is a Google account.

- Free to use — no software installation required
- Runs in your browser — works on any device
- All Python libraries are pre-installed or easily added
- You can share your notebook with anyone via a link

1.4 Prerequisites — What You Need Before Starting

You do NOT need prior coding experience. This guide will walk you through every step. However, make sure you have:

1. A Google Account (Gmail) — free to create at google.com
2. Access to Google Colab — visit colab.research.google.com
3. A stable internet connection (downloads market data live)
4. About 30 minutes of uninterrupted time

SECTION 2 — SETTING UP GOOGLE COLAB

2.1 Opening Google Colab

Follow these steps exactly to open a new notebook:

5. Open your browser and go to: `colab.research.google.com`
6. Sign in with your Google account if prompted
7. Click 'New Notebook' (blue button, top left)
8. A new untitled notebook will open — you will see an empty code cell

2.2 Understanding the Colab Interface

Colab is made up of 'cells'. Each cell can contain code or text. Here is what you need to know:

Part of Interface	What It Does
Code Cell	Where you type/paste Python code
Play Button (▶)	Runs the code in that cell
+ Code	Adds a new code cell below
Runtime menu	Controls the Python environment
Output area	Shows results, charts, and errors below each cell

TIP: To run a cell, click the Play (▶) button on its left side, or press Shift + Enter on your keyboard.

2.3 Installing Required Libraries

Before running the main model, you need to install one library that is not pre-installed in Colab. Create your FIRST code cell and paste this exactly:

```
# CELL 1 — Install required library
# Paste this into your first Colab cell and press the Play (▶) button

!pip install yfinance --quiet
```

After you run this cell, you should see a success message. The other libraries (numpy, pandas, matplotlib, plotly) are already available in Colab by default.

IMPORTANT: You must run Cell 1 first, every time you open a new Colab session. The libraries reset when you close the tab.

SECTION 3 — THE MASTER PROMPT (COPY & PASTE)

3.1 What Is a Prompt?

In this context, a 'prompt' is a complete set of instructions you give to an AI (like Gemini, ChatGPT, or Claude) to generate Python code for you. You do NOT write the code yourself — the AI writes it based on the prompt.

Alternatively, you can paste the prompt directly into Google Colab as a comment and let Colab AI (powered by Gemini) generate the code for you inline.

3.2 Two Ways to Use This Prompt

Method	How It Works
Method A — AI Chatbot	Copy the prompt → paste into ChatGPT / Gemini / Claude → copy the generated code → paste into Colab
Method B — Colab AI	Open Colab → click the AI button (sparkle icon) → paste the prompt → let Colab write code for you directly

3.3 The Complete Master Prompt

Copy EVERYTHING below (from the first line to the last line) and paste it into your AI tool of choice:

```
Act as a Quantitative Finance Researcher.
Build a Market Timing Optimization Model for NIFTY 50 using 8 years of daily data.

Use Python with: yfinance, numpy, pandas, matplotlib, plotly

OBJECTIVE:
Create a model that tells WHEN to invest, not WHAT to buy.

STEP 1 — DATA:
Download 8 years of daily data for ^NSEI (NIFTY 50) using yfinance.

STEP 2 — FEATURES:
```

Calculate:

- 50-day moving average (MA50)
- 200-day moving average (MA200)
- Momentum Strength = $(\text{Price} / \text{MA200}) - 1$
- 20-day rolling volatility (std dev of daily returns), normalized
- Rolling maximum price
- Drawdown = $(\text{Price} - \text{Rolling Max}) / \text{Rolling Max}$

STEP 3 – ENTRY SCORE:

Create a composite Entry Score by combining:

- Momentum component (positive momentum = higher score)
- Volatility component (lower volatility = higher score)
- Drawdown component (deeper drawdown = buying opportunity)

Normalize all 3 components to 0-1 range. Combine them equally.

STEP 4 – POSITION SIZING STRATEGY:

Based on Entry Score (no binary buy/sell signals):

- High Score (above 0.6) --> 100% invested
- Medium Score (0.3-0.6) --> 50% invested
- Low Score (below 0.3) --> 15% invested

STEP 5 – BACKTEST:

Simulate INR 100,000 starting capital.

Calculate daily portfolio value based on position sizing.

Compare Strategy vs Buy & Hold over the same period.

STEP 6 – VISUALS (use plotly for all charts):

Plot 1: Portfolio growth – Strategy vs Buy & Hold

Plot 2: Entry Score over time with threshold lines

Plot 3: Drawdown chart for both strategies

Plot 4: NIFTY price with Buy Zones highlighted (green = high score)

STEP 7 – OUTPUT:

Print a performance summary showing:

- Total Return % (Strategy vs Buy & Hold)
- CAGR % (Strategy vs Buy & Hold)
- Max Drawdown % (Strategy vs Buy & Hold)
- Sharpe Ratio (Strategy vs Buy & Hold)

BONUS (include this): Add ipywidgets sliders for:

- Momentum weight (0.0 to 1.0)
- Volatility weight (0.0 to 1.0)
- Drawdown weight (0.0 to 1.0)

Weights should auto-normalize. Show updated results interactively.

Output clean, well-commented Python code ready to run in Google Colab.

Use clear section headers as comments in the code.

SECTION 4 — STEP-BY-STEP EXPLANATION FOR BEGINNERS

This section explains each part of the model in plain English so you understand what is happening at every stage.

STEP 1

Downloading NIFTY 50 Data

What Happens

The model uses yfinance (Yahoo Finance library) to automatically download 8 years of NIFTY 50 closing prices. The ticker symbol for NIFTY 50 on Yahoo Finance is ^NSEI.

Why 8 Years?

8 years of data gives the model enough history to capture multiple market cycles — bull runs, corrections, crashes, and recoveries — making the strategy more reliable.

Beginner Prompt Tip

If the data download fails, try asking the AI: *"Fix the yfinance download error for ^NSEI and add error handling."*

STEP 2

Calculating the Three Key Features

Feature 1 — Momentum

Momentum measures the strength of the market trend. We calculate two Moving Averages (MA):

Indicator	Meaning
MA50 (50-day)	Average price of last 50 days — short-term trend
MA200 (200-day)	Average price of last 200 days — long-term trend
Momentum Strength	$(\text{Current Price} / \text{MA200}) - 1$ = how far above/below trend

If Momentum Strength is positive (price is above MA200), the market is in an uptrend. If negative, the market is below its long-term average.

Feature 2 — Volatility

Volatility measures how much the market is swinging up and down. We calculate the standard deviation of daily returns over the last 20 days. High volatility = risky, uncertain market. Low volatility = stable, calmer market.

Note: We INVERT the volatility signal — lower volatility gets a HIGHER score, because it represents a less risky entry point.

Feature 3 — Drawdown

Drawdown measures how far the market has fallen from its recent peak. Formula:

$$\text{Drawdown} = (\text{Current Price} - \text{Rolling Maximum Price}) / \text{Rolling Maximum Price}$$

A deep drawdown (e.g., -30%) means the market has crashed significantly. Historically, deep drawdowns represent buying opportunities for patient investors.

STEP 3 Calculating the Entry Score

The Entry Score combines all three features into a single number between 0 and 1. Here is how it works:

Score Range	Market Interpretation
0.7 — 1.0	Excellent time to invest (strong uptrend, low volatility, recovering from drawdown)
0.4 — 0.7	Moderate opportunity (mixed signals, partial investment)
0.0 — 0.4	Caution advised (high risk, high volatility, or weak trend)

Each component is normalized to a 0-1 scale and then averaged (or weighted if you use the sliders). This prevents any single feature from dominating the score.

STEP 4 Position Sizing Strategy

Instead of a binary Buy / Sell signal, the model uses position sizing — it adjusts HOW MUCH of your capital is invested based on the Entry Score:

Entry Score	Action Taken
Above 0.6 (High)	100% of capital invested — maximum confidence
0.3 to 0.6 (Medium)	50% of capital invested — cautious participation
Below 0.3 (Low)	15% of capital only — minimal risk exposure

KEY INSIGHT: You are always in the market to some degree (15% minimum). This prevents missing sudden sharp recoveries while still protecting you from major downturns.

STEP 5 Running the Backtest

The backtest simulates what would have happened if you used this model over the past 8 years with a starting capital of INR 1,00,000. It calculates:

- Daily portfolio value based on the Entry Score-driven position sizing
- A Buy & Hold portfolio (always 100% invested, no timing) for comparison
- The difference in final value between the two strategies

IMPORTANT DISCLAIMER: Past performance does not guarantee future results. This model is for educational purposes only and is not financial advice.

STEP 6 The Visualizations

The model generates 4 interactive charts using Plotly. You can zoom, pan, and hover over them in Colab:

Chart	What It Shows
Portfolio Growth	Strategy vs Buy & Hold value over time (INR)
Entry Score	The composite score over time with threshold lines at 0.3 and 0.6
Drawdown Chart	How much each strategy fell from its peak at any point
NIFTY Buy Zones	Price chart with green highlights showing when score was high

STEP 7 Performance Output

After the charts, the model prints a clean summary table. Here are the four metrics and what they mean:

Metric	Plain English Explanation
Total Return %	Total profit as a % of starting capital over the full period
CAGR %	Compounded Annual Growth Rate — your average yearly return
Max Drawdown %	Worst peak-to-trough loss — how much you lost before recovering
Sharpe Ratio	Return per unit of risk — higher is better (above 1.0 is good)

SECTION 5 — BONUS: INTERACTIVE SLIDERS

5.1 What Are the Sliders?

The bonus section adds ipywidgets sliders — interactive controls you can drag with your mouse directly inside the Colab notebook. They let you change the weight of each signal and instantly see how results change.

5.2 The Three Slider Weights

Slider	What It Controls
Momentum Weight	How much importance to give to trend direction
Volatility Weight	How much importance to give to market stability
Drawdown Weight	How much importance to give to buying-the-dip opportunities

The three weights are automatically normalized, so they always sum to 1.0. For example, if you set all three to 1.0, each gets 33.3% weight.

5.3 Experimenting With Weights — Suggested Presets

Try these configurations to understand how each signal affects performance:

Preset Name	Momentum / Volatility / Drawdown
Balanced (default)	1.0 / 1.0 / 1.0 — equal weight to all three
Trend Follower	2.0 / 0.5 / 0.5 — focus on uptrends only
Contrarian / Dip Buyer	0.5 / 0.5 / 2.0 — buy when market has crashed
Risk Averse	1.0 / 2.0 / 1.0 — avoid entering during volatile periods

SECTION 6 — TROUBLESHOOTING COMMON ERRORS

6.1 Common Issues and Fixes

If you run into problems, here are the most common errors and exactly what to type to the AI to fix them:

Error / Problem	What To Ask the AI
ModuleNotFoundError: yfinance	"Fix: install yfinance in Google Colab and re-run the download"
Download returns empty data	"The yfinance download for ^NSEI returned empty. Fix it with a date range from 2016 to today"
Chart does not display	"The plotly chart is not showing in Colab. Use fig.show() and add renderer='colab'"
Sliders not working	"The ipywidgets sliders are not interactive in Colab. Fix the widget display"
NaN errors in score	"I am getting NaN values in the Entry Score. Add .dropna() and forward-fill missing values"
Code runs but no output	"Add print statements after each section to confirm data is loading correctly"

6.2 How to Re-Run After Errors

9. Go to Runtime menu → Restart and Run All
10. If that fails, go to Runtime → Factory Reset Runtime
11. Then run Cell 1 (pip install) again before running the main code

SECTION 7 — UNDERSTANDING THE MODEL DEEPER

7.1 When Does the Model Invest?

The model invests heavily when ALL THREE of the following are true simultaneously:

- The market is above its 200-day moving average (uptrend / momentum is strong)
- Recent daily price swings are small (volatility is low — stable market)
- The market has already corrected from a recent peak (drawdown creates opportunity)

When all three conditions align, the Entry Score rises above 0.6 and the model puts 100% to work.

7.2 When Does the Model Avoid the Market?

- Market is below its 200-day MA (bearish trend — momentum is weak or negative)
- Very high day-to-day price swings (panic / uncertainty in the market)
- Market is at an all-time high with no recent correction (no margin of safety)

7.3 Why Do Volatility and Drawdown Matter?

Together, volatility and drawdown act as a safety filter:

High volatility often precedes or accompanies market crashes. By reducing exposure when volatility is high, the model avoids getting caught in sharp sell-offs.

Deep drawdowns, on the other hand, historically mark the best long-term buying opportunities. The 2020 COVID crash, the 2016 correction, and the 2022 bear market were all followed by strong recoveries. The model identifies these as high-score zones.

7.4 Limitations of This Model

- This model uses hindsight (historical data) — future results may differ significantly

- Transaction costs (brokerage, STT, slippage) are not included in the backtest
- The model does not account for dividends or corporate actions
- It works on index-level data only — not individual stocks
- No model is perfect — combine this with your own research and risk tolerance

SECTION 8 — NEXT STEPS & ADVANCED IDEAS

8.1 Modifications to Try

Once you have the model running, try these prompt additions to make it more advanced:

- Add RSI (Relative Strength Index) as a 4th feature to the Entry Score
- Add MACD signal line crosses as an additional confirmation filter
- Test the model on SENSEX (^BSESN) and compare results to NIFTY
- Add a rebalancing rule — only change position monthly, not daily
- Include transaction cost simulation (0.1% per trade both ways)

8.2 Prompts to Extend the Model

Here are ready-to-use follow-up prompts to improve your model:

Add RSI:

```
Add a 14-day RSI indicator to the existing market timing model.  
Include it as a 4th component of the Entry Score.  
RSI below 40 should increase the score (oversold = buying opportunity).  
RSI above 70 should decrease the score (overbought = reduce exposure).  
Normalize RSI to 0-1 and add a slider for its weight.
```

Add Transaction Costs:

```
Modify the backtest to include transaction costs.  
Charge 0.1% of portfolio value whenever position size changes by more than 10%.  
Show how total return changes with and without transaction costs.
```

8.3 Sharing Your Notebook

12. In Colab, go to File → Share
13. Set access to 'Anyone with the link can view'
14. Copy the link and share it with others

QUICK REFERENCE CARD

3-STEP QUICK START

Step	Action
1. Open Colab	Go to colab.research.google.com → New Notebook
2. Install Library	Paste: <code>!pip install yfinance --quiet</code> → Run
3. Run the Prompt	Copy Section 3.3 prompt → paste into AI → run generated code

GLOSSARY OF KEY TERMS

Term	Plain English Definition
yfinance	Python library to download stock price data from Yahoo Finance
Moving Average (MA)	Average price over last N days — smooths out noise
Momentum	Speed and direction of market price movement
Volatility	How much prices are swinging — higher = riskier
Drawdown	How far the market has fallen from its recent highest point
Entry Score	Combined 0-1 signal telling you how good the current entry point is
Position Sizing	Deciding what % of your capital to invest (not all-or-nothing)
Backtest	Testing a strategy on past data to see how it would have performed
CAGR	Compound Annual Growth Rate — average yearly % return
Sharpe Ratio	Measures return vs risk — higher is better
ipywidgets	Python library for interactive sliders in Jupyter / Colab notebooks

DISCLAIMER

This guide is for educational and informational purposes only. Nothing in this document constitutes financial advice. Always consult a SEBI-registered financial advisor before making investment decisions. Past model performance does not guarantee future results.