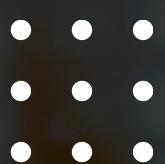
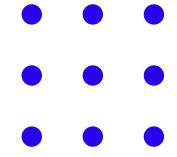




UM Hackathon 2025

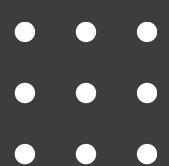
Domain 2: Alpha Strategies Using HMM
Presentation by Wonder Girls (82)





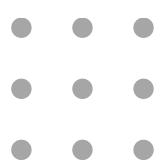
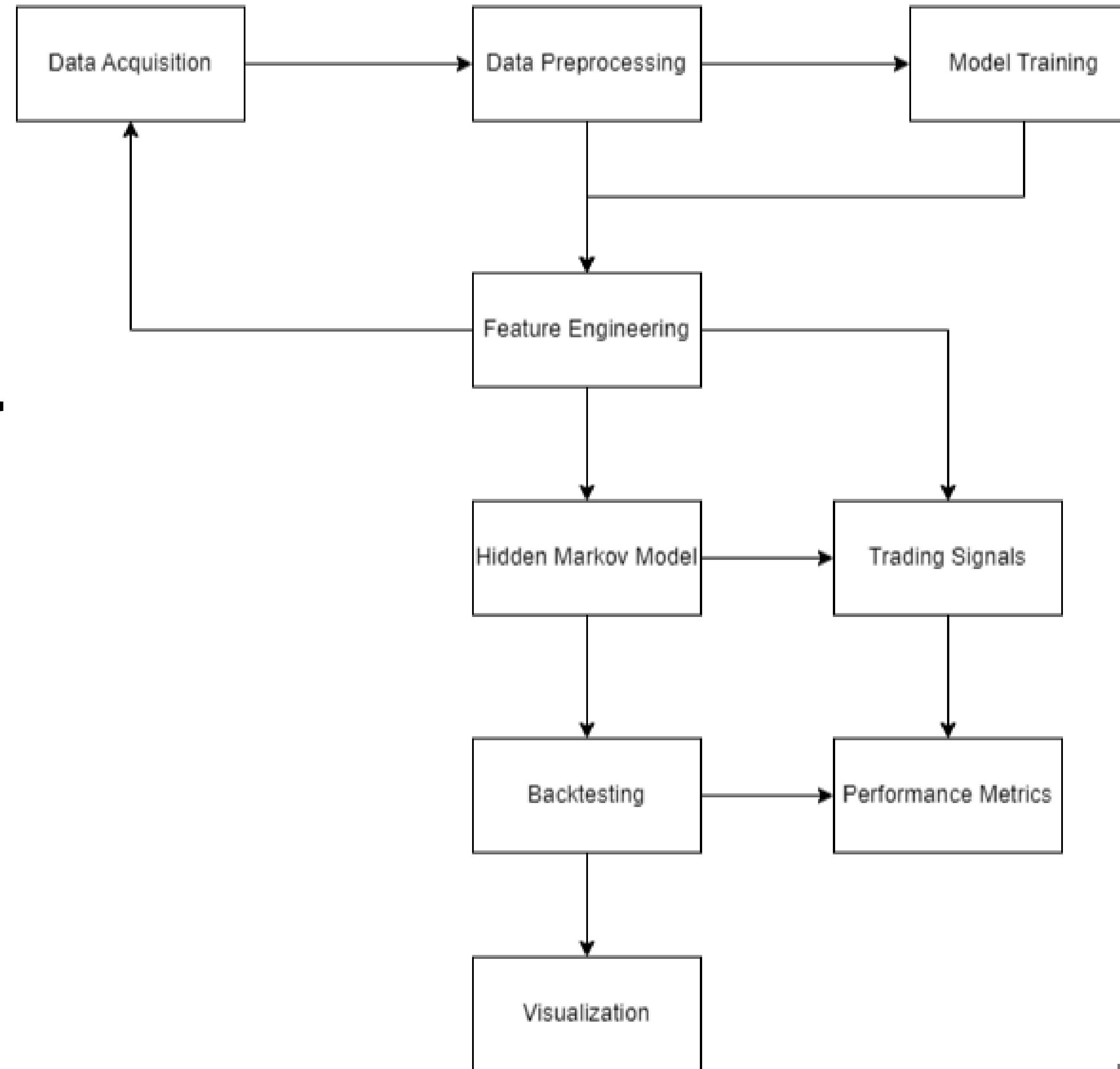
Objectives

- Implementing **alpha trading strategy** which detects hidden signals or mispriced assets.
- The goal is to make profits that are higher than overall market returns.
- Our objective is **to find correlations of hidden market features to predict future market behavior.**



Conceptual diagram

Our
implementation
strategy



Data source: CryptoQuant

1. Retrieving datasource

- **exchange_netflow**; netflow_total
- **miner_to_exchange**; flow_total
- **exchange_reserve**; reserve_usd
- **price_ohlcv**; open, high, low, close, volume
- data interval; 1 hour for 1 year

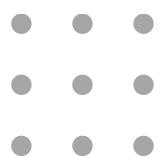


• • •
• • •
• • •

Data source: CryptoQuant

2. Data cleaning

- Check for missing values
- Align timestamps
- Feature engineering;
 - returns (percentage change)
 - rolling volatility
 - volume-weighted average price
 - miner sell pressure ratio





Implicit Market Indicators

01

Volume-weighted Average Price

Used to find hidden accumulation – where price may suddenly spike

02

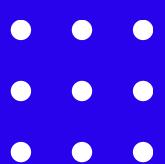
Miner sell pressure

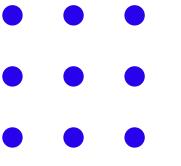
Ratio > 1 indicate miners are selling a lot, which could push the price down

03

Volatility

High volatility indicate price is moving a lot, where it could be a signal for breakouts and periods of low volatility usually followed by breakout

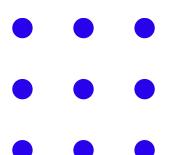




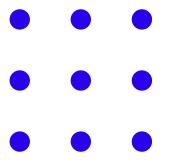
Model Training

Gaussian Hidden Markov Model (HMM)

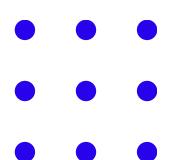
- Train the model on the engineering features; returns, volatility, VWAP, and miner sell ratio to find patterns and correlations.
- Predict hidden states using the trained HMM.
- Maps hidden states to trading signals (buy, sell, or hold).



Strategy

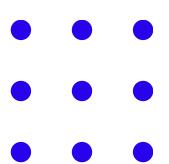


- Applying feature engineering
- Then, model will generate the hidden state trade signals:
 - Bearish (0)
 - Neutral (1)
 - Bullish (2)
- When the model detects state 2 → market is bullish → Buy.
- When the model detects state 0 → market is bearish → Sell.
- Otherwise → do nothing.



Backtesting plan

- Initialization
 - Start with \$10,000.
 - Position = 0 means no assets are held.
 - Each trade deducts 0.06% fee.
- Simulating Trades
 - Loops through your dataset row by row (time-series).
 - close_price is the asset price at each point in time.

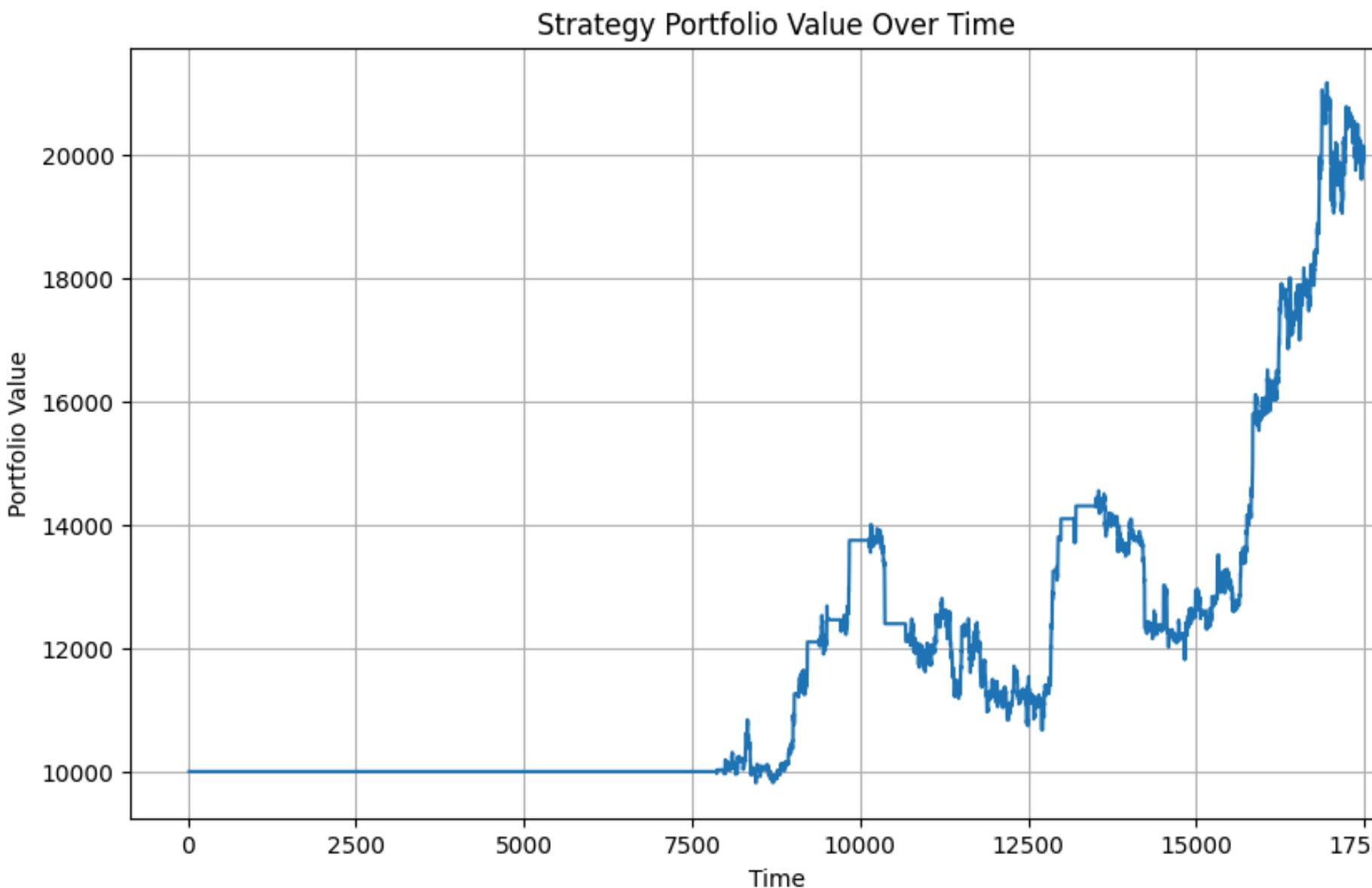
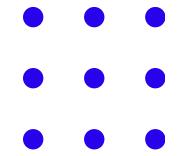
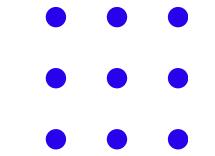


Backtesting plan

- Buy logic
 - If the signal is "Buy" and you don't own any:
 - Calculate how many units you can buy after deducting the fee.
 - Invest all available capital into the asset.
 - Set portfolio_value = 0 because all money is now in the asset.
- Sell logic
 - If the signal is "Sell" and you do hold assets:
 - Sell everything at the current close_price.
 - Deduct the fee.
 - Set your position back to 0 (all in cash).

⋮
⋮
⋮

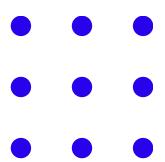
Expected Result



- By using the model, we expect a maximum drawdown of **- 23.77%**
- The sharpe ratio (SR) of ≥ 1.8 is expected

Suggestion/Plan for improvement

- Obtain more datasets and add more predictive features
- Feature engineer the predictive features
- Use HMM to detect bull/bear/sideways markets and adjust position sizing
- Performance boost: implementing parallel processing and caching
- Add compliance check (“Sharpe too low”, “Trade frequency < 3”)





Thank you

