Details of ML model

1. Model 1: Multistep Hidden Markov Model and Gaussian Mixture Model

   The first component in the pipeline is a multistep Hidden Markov Model combined with a Gaussian Mixture Model (HMM-GMM), which is responsible for identifying and forecasting market regimes—underlying conditions such as bullish, bearish, or neutral phases that drive observable market behaviors.

   The Hidden Markov Model (HMM) is particularly effective at uncovering latent states from time-series data. In this context, it interprets patterns in on-chain metrics, fear and greed index, and market data to detect shifts between regimes. These hidden states reflect underlying market sentiment and risk conditions that aren't directly observable.

   To improve expressiveness, each hidden state is modeled using a Gaussian Mixture Model (GMM). This enables the model to capture complex, multimodal distributions of data within each regime—for example, distinguishing between shallow and deep bear phases within a broader bearish regime.

   The "multistep" capability of this setup means the model doesn't just classify the current regime but also predicts the sequence of future regimes across multiple time steps. This forward-looking view helps the system anticipate transitions in market dynamics—critical for proactive trading decisions.

   Expected output:
   - Step 1: Shallow Bear
   - Step 2: Shallow Bear
   - Step 3: Deep Bull
   - Step 4: Shallow Bull
   - Step 5: Neutral

2. Model 2: FinBert

   FinBERT is a financial language model that extracts sentiment scores from unstructured text sources like Twitter, Reddit, and news headlines. It outputs a sentiment score ranging from 0 to 1 for negative, positive and neutral, capturing the emotional tone of market participants. However, sentiment alone doesn't tell the full story—a strong sentiment signal without traction may be less impactful than a neutral signal that's widely discussed.

To account for this, we incorporate a buzz factor—a measure of how frequently an asset is mentioned across social and news platforms within a given time window. The buzz factor reflects market attention or hype, which is a key driver of short-term volatility and liquidity. It's typically computed as a normalized count of mentions per hour or day, adjusted for historical baselines to detect spikes.

By combining sentiment score and buzz factor, we form a richer representation of public opinion:

- A high sentiment + high buzz may indicate a FOMO-driven bull run.
- A negative sentiment + high buzz may signal panic or potential dumps
- Low buzz + strong sentiment may suggest weak conviction or non-actionable optimism.

This pair of features is then passed to the DL aggregator model to inform downstream trading decisions. It helps the system distinguish between emotional noise and market-moving sentiment backed by strong crowd momentum.

Expected Output:

- Sentiment Score: 0 to 1 (reflecting negative, positive and neutral)
- Buzz Factor: 0 to 1 (reflecting discussion intensity or hype strength)

3. Model 3: Supervised Deep Learning Model (Attention-Based Aggregator)

This model acts as a feature aggregator with features from multistep HMM-GMM model, sentiment score from FinBert and whale activity score and price changes which act as a labeled datasets.

The objective of this deep learning (DL) model is not to directly generate trading signals, but to learn complex nonlinear relationships between market regime patterns, sentiment shifts, whale behavior, and future price movements. The model is trained in a supervised learning setting, where each input feature vector corresponds to a target label—price change.

By aggregating diverse sources of information, the DL model captures richer contextual patterns that may not be easily represented by rule-based or shallow statistical methods. For instance, a specific combination of whale activity spike and bearish sentiment might indicate an upcoming dump, but only under certain regime intensities. The DL model can learn to identify such conditions over time.

Expected output:
Aggregated vector with meaningful data: [0.45, 0.1, 0.3,etc…]

This architecture ensures that low-level feature extraction and aggregation are handled by the deep learning model, while higher-level policy control and risk constraints (like Sharpe ratio or drawdown targets) can be optimized separately via reinforcement learning or rule-based postprocessing.

4. Model 4: Reinforcement Learning Model (Proximal Policy Optimization (PPO))

This model is a decision maker. It receives the aggregated feature vector from the deep learning model—containing encoded signals about market regime intensity, sentiment, whale behavior, and price changes—and learns to take optimal trading actions such as Buy, Hold, or Sell. The reinforcement learning (RL) agent is trained using a reward function that aligns with trading objectives such as maximizing the Sharpe Ratio, minimizing maximum drawdown, and ensuring sufficient trade frequency.

Over time, the RL agent learns to associate certain feature patterns with favorable or unfavorable outcomes. For example, it may learn that entering trades during shallow regimes often leads to lower returns after accounting for trading fees, and therefore chooses to hold. Conversely, when the feature vector signals a high-intensity bullish regime supported by positive sentiment and strong whale accumulation, the agent learns to confidently execute a buy. By continuously interacting with the environment and receiving feedback, the RL model adapts its policy to optimize long-term performance under dynamic market conditions.

Expected output: Buy, Sell or Hold