Contents

[1. Data Loading and Preprocessing 2](#_Toc208770058)

[1. EDA, daily summaries 4](#_Toc208770059)

[2. EDA, individual trades 6](#_Toc208770060)

[3. Recommendation on target variable for ML model 8](#_Toc208770061)

[4. EDA, feature predictive power 9](#_Toc208770062)

[5. Composite quintile model evaluation 13](#_Toc208770063)

[6. Some analytics on trade vs skip days, OOS data 14](#_Toc208770064)

[7. Alternative approach: continuous sizing strategy 16](#_Toc208770065)

[8. Recommended next steps 18](#_Toc208770066)

## Data Loading and Preprocessing

**Setup & Checks**

* Loaded database credentials with load\_dotenv().
* Created a SQLAlchemy engine and verified schema/row counts for the three main tables.

**Data extraction & alignment**

* Pulled full data from market\_metrics, daily\_summaries, and individual\_trades.
* Converted timestamps to datetime and added a common date field to align tables.

**Reshape & visualization**

* Pivoted market metrics into wide format (one column per metric, by date), merged with daily summaries, and sorted chronologically.
* Generated interactive Plotly charts to explore the behavior of 40+ pre-market indicators across the 5-year history.

**Notes on Data Quality**

While visualizing the data, I noticed that some metrics are noisy, inconsistent, or lack informational value. These will be excluded from the analysis:

* FOMC\_Minutes and FOMC\_Rate are always 0.

A screenshot of a graph

Descripción generada automáticamente

* gamma support and resistance level: should this be compared with the QQQ open of that day to make it a stationary feature? I understand the absolute value by itself doesn’t say much.

A graph of a stock market

Descripción generada automáticamente con confianza media

* IV\_skew\_25\_d\_put\_minus\_call looks like a very noisy signal; I’d say the data quality for this metric isn’t very good.

A graph showing the growth of the stock market

Descripción generada automáticamente

* Options\_0DTE\_volume\_pct is a non-stationary signal with a clear shift, and it also looks too noisy, doesn’t seem like high-quality data, as many days are just 0.

A graph showing the growth of the stock market

Descripción generada automáticamente

* I have identified and removed couple of outliers from GEX\_aggregate\_bn (interpolated).

A graph with a line

Descripción generada automáticamente

Other signals look fine, but some are non-stationary. For those, z-scores could be used to normalize.

A graph showing a graph of a graph

Descripción generada automáticamente con confianza media

A graph showing a line

Descripción generada automáticamente

## EDA, daily summaries

As a first step, I plotted the number of trades per day against the daily win rate and total PnL. The results show a clear inverse correlation: as the trade count increases, the win rate falls, and the PnL becomes negative (regression is not needed here, patter is obvious).

A graph with blue bars

Descripción generada automáticamente

A graph with a line graph

Descripción generada automáticamente

A graph showing a line graph

Descripción generada automáticamente con confianza media

**Win Rate Analysis**

To calculate win rate, I aggregated all wins and losses across days within each trade count bucket (rather than averaging daily percentages).

|  |  |  |
| --- | --- | --- |
| Trades per day | Win rate (my calculation) | Win rate (stated in your analysis) |
| 1-3 | 54.8% | 91% |
| 4+ | 17.7% | 35% |

Although my absolute numbers differ, the conclusion is the same: days with fewer than 4 trades are systematically better than days with 4 or more trades. This suggests the finding is **robust**, regardless of calculation method.

**PnL Analysis**

I also calculated aggregated PnL for each trade count bucket:

A graph with blue and purple bars

Descripción generada automáticamente

* **When n\_trades > 4**, the total PnL is consistently negative, confirming that high-trade, reversal-driven days destroy profitability.
* **Between 1 and 4 trades**, there is an **inverse relationship**: the fewer the trades, the higher the total PnL. In other words, days with only 1–2 clean trades generate the bulk of profits, while profitability steadily erodes as the number of trades increases.

This reinforces the earlier conclusion: multiple-trade days (especially those with 4+) correspond to choppy conditions and cascade losses, whereas low-trade days capture the profitable directional moves.

**Interpretation of Trade Count vs PnL**

The correlation between number of trades and PnL is expected given the mechanics of the strategy. The ORB system uses a 5:1 reward-to-risk setup (500-tick take profit vs. 100-tick stop loss):

* In **choppy markets**, price action tends to trigger stop losses repeatedly before any large move develops. This produces more trades (due to reversals) and leads to cumulative losses.
* In **trending markets**, the strategy can reach the 500-tick take profit cleanly, often within 1–3 trades. This results in fewer trades and significant profits.

Thus, the number of trades is not an independent driver of performance, but rather a symptom of the underlying market structure (trend vs. chop).

This exploratory analysis revealed an almost **linear inverse relationship between the number of trades and daily PnL** (both in terms of averages and totals).

A graph of blue squares

Descripción generada automáticamente

In practice, fewer trades correspond to clean, profitable breakouts, while more trades correspond to choppy, unprofitable sessions. Based on this, it makes sense to reframe the prediction problem around forecasting **the number of trades for the day** (1, 2, 3, or ≥4). This target is both well-defined and strongly correlated with profitability, making it a practical proxy for identifying which days should be traded versus skipped.

## EDA, individual trades

**Daily-Level Summary**

I analyzed the individual\_trades table at the daily level by iterating over all trading days and summarizing three key metrics:

* The total number of trades executed (n\_trades)
* The aggregated daily profit and loss (pnl)
* Whether the **first trade of the day** was profitable (first\_trade\_profitable). So, take\_profit was hitted before stop\_loss.

To compute this, I filtered trades by date, sorted them by entry time (datetime\_in), and extracted the relevant values. With this daily summary in place, I then split the dataset into two groups: days where the first trade was profitable and days where the first trade was a loss.

Finally, I plotted the distribution of the number of trades for each group. This allows me to compare how trade counts differ depending on whether the first trade of the day was a win or a loss.

A graph with numbers and a bar

Descripción generada automáticamente A graph of a number of trade

Descripción generada automáticamente con confianza media

**Position of First Profitable Trade**

I analyzed each trading day to record the total number of trades, the total PnL, and the position of the first profitable trade (1st, 2nd, 3rd, etc.). If no trade was profitable, the day was marked as 9 (the maximum observed value for first profitable trade on other days was 8).

Plotting n\_trade\_profitable against daily PnL reveals a clear pattern: the later the first win occurs, the worse the overall PnL tends to be. This captures the **cascade effect**: the more trades required, the choppier the market, the longer it takes to see the first profitable trade, and the lower the final PnL.

I then produced the same plot aggregated by bucket, showing the **total PnL per value of n\_trade\_profitable**, which reinforces the same conclusion.

A graph of a graph

Descripción generada automáticamente con confianza media

A graph with blue squares

Descripción generada automáticamente

**Key Findings**

From the distributions, I observe that:

* **When the first trade is profitable**, most days consist of only one trade. These days are typically clean breakout sessions where the strategy captures the 5:1 move quickly.
* **When the first trade is not profitable**, the day usually involves two or more trades, reflecting the reversal and chop dynamics of the strategy.

The performance difference is very clear:

|  |  |  |
| --- | --- | --- |
| Group | Avg. daily PnL | Total PnL |
| First trade profitable | +$1,928 | +$509,090 |
| First trade loss | -$279 | -$299,199 |

## Recommendation on target variable for ML model

Based on the analysis, I recommend exploring two alternative target definitions for the prediction model:

1. **Binary classification** – whether the first trade of the day is profitable (True / False).
   * Strong signal: +$1,928 avg. PnL when true vs. –$279 when false.
   * Clear separation: +$509k aggregate profits from “first trade wins” vs. –$229k from “first trade losses.”
   * Captures the essence of clean breakout vs. choppy reversal days.
2. **Regression / ordinal classification** – the number of trades in a day (1, 2, 3, 4+).
   * Supported by almost linear inverse relation between trade count and PnL.
   * Captures the cascade effect: more trades correspond to later first win and lower profitability.

Both targets align with the strategy’s design and mechanics. The first-trade profitability target is tightly connected to daily outcome quality, while the trade count target offers a more general proxy for chop vs. trend days.

## EDA, feature predictive power

**Prepare dataset for the analysis**

In this stage I first cleaned the feature set by removing low-value metrics such as FOMC\_Minutes and FOMC\_Rate, and set aside others like the gamma levels, option skew, and 0DTE volumes for later review.

For the remaining signals I transformed each one into a **quintile score (1–5)**, calculated in an expanding way using only past data to avoid leakage. This allows me to compare each day’s value against its own history and keep the signals stationary. I plotted the raw time series together with their quintile version to confirm the transformation.

**Correlation/Information Coefficient**

Next, I merged these normalized features with the daily trade outcomes and **split the dataset in time (first half for training, second half for testing).** On the training set I computed the **Spearman correlation (Information Coefficient) between each feature quintile and the total PnL**, then ranked the metrics by absolute correlation.

Finally, I visualized the relationship by plotting the PnL against feature quintiles in a grid of subplots, sorted by their IC values, with each title showing the metric name and its correlation. This gives both a quantitative ranking of predictive power and a qualitative view of how each signal behaves across its range.

A screenshot of a graph

Descripción generada automáticamente

From the IC ranking I selected the five strongest predictors, but I also checked the correlation matrix to avoid redundancy. Several metrics are highly correlated with each other (for example, VIX3M, VIX, and VIX9D), so including all of them would add little new information while increasing collinearity risk. To address this, I kept only one representative from each highly correlated group and then selected the top five **uncorrelated features**. This way, the final feature set captures the strongest signals while remaining diverse and non-redundant.

A screen shot of a computer screen

Descripción generada automáticamente

Selected features based on correlation with n trades:

* PDC ATR Delta
* Options Volume vs 20d avg
* PDC VOL 20MA
* NQ overnight gap pct

A screenshot of a computer

Descripción generada automáticamente

**Composite Feature-Based Trading Filter**

To simplify the analysis and test the predictive value of the strongest signals, I constructed an **aggregate quintile indicator**. This is defined as the average of the quintile scores of the top uncorrelated features. The idea is to capture a single composite measure of “market conditions” rather than relying on each feature individually.

For the first test, I will apply a simple rule:

* **If the aggregate quintile is 1** (i.e., the combined signal is in its historically low range), the model will **skip trading** for that day.
* Otherwise, the day is considered tradable.

This provides a straightforward way to evaluate whether clustering multiple features into a single normalized signal can effectively distinguish between favorable and unfavorable trading conditions.

**Results on In-Samle data (TRAIN)**

The results show that the composite quintile (defined as the average of the top uncorrelated features’ quintiles) achieves a much stronger Spearman correlation with the number of trades than any of the individual features on their own.

A graph with blue squares

Descripción generada automáticamente A graph of a bar

Descripción generada automáticamente con confianza media

This suggests that the aggregation is successfully capturing a more robust and informative signal about market conditions.

When plotting the average number of trades against the composite quintile, the relationship is clear:

* Lower composite quintiles correspond to more trades,
* while higher composite quintiles correspond to fewer trades.

**Results on Out-Of-Samle data (TEST)**

If we repeat this exercise with test data, results are SURPRINSINGLY GOOD! (I didn’t expect this tbh, I have to review every step of the process).

A graph with blue squares

Descripción generada automáticamente A graph of a bar chart

Descripción generada automáticamente con confianza media

## Composite quintile model evaluation

As a next step, I will evaluate the composite quintile as a **trading filter on out-of-sample data**. Specifically, I will exclude days where the combined quintile signal is equal to 1. This corresponds to filtering out roughly the bottom 20% of days.

By doing so, the strategy effectively “skips” trading on these low-quality sessions. I will then compare the performance metrics — total profit, average daily PnL, and win rate (%) — against the baseline results using the full dataset without filtering.

This analysis will show whether the composite quintile improves risk-adjusted returns by removing the days most likely to be unprofitable, while still capturing the majority of profitable breakout opportunities.

**Conclusion from overall results**

To evaluate the effectiveness of the composite quintile filter, I compared performance and accuracy metrics between the **days we trade** (quintile > 1) and the **days we skip** (quintile = 1), both in-sample and out-of-sample.

|  |  |  |
| --- | --- | --- |
| Data | DO trade aggregate PnL | DON’T trade aggregate PnL |
| In-Sample (TRAIN) | +$190k | -$53k |
| Out-of-Sample (TEST) | +$173k | -$30k |

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Accuracy (Hit rate) | Trade precision | Skip precision |
| In-Sample (TRAIN) | 0.37 | 0.26 | 0.79 |
| Out-of-Sample (TEST) | 0.38 | 0.25 | 0.78 |

The composite quintile filter shows a **consistent improvement in profitability** across both in-sample and out-of-sample periods. When restricting trades to days where the composite quintile is greater than 1, the strategy captured **strong positive aggregate PnL** (+$190k in-sample, +$173k out-of-sample). In contrast, the skipped days (quintile = 1) were **net losers**, with aggregate PnL of –$53k and –$30k, respectively. This demonstrates that the filter is successfully removing a segment of days that are, on average, unprofitable.

From a classification standpoint, the accuracy metrics are more modest. The overall hit rate is ~37–38%, with trade precision around 25–26% and skip precision near 78–79%. This reflects the asymmetric nature of the ORB strategy: a small proportion of winning days deliver most of the profits, while many days contribute small or moderate losses. As a result, **precision on “do trade” decisions is low**, but **precision on “don’t trade” decisions is high**.

Taken together, these results suggest that the composite quintile filter is a valuable tool. While it does not predict every profitable day with high accuracy, it effectively **filters out the worst-performing days**, thereby improving the aggregate profitability of the strategy in both training and testing. The key insight is that the filter works better as a **risk-control mechanism** (avoiding bad days) than as a precise selector of all good days.

## Some analytics on trade vs skip days, OOS data

Using the OOS (test) split, I compare days **skipped by the composite signal** (“don’t trade”) vs. **not skipped** (“trade”). I plot (1) the distribution of the **number of trades per day** and (2) the distribution of **total daily PnL**, and report the **% of days where the first trade is profitable** in each group.

**Plot 1 – Distribution of number of trades (test data)**

Days **skipped** by the signal show a **clear right shift** toward **more trades per day**. This matches prior findings: higher trade counts are symptomatic of choppy/reversal sessions that trigger multiple stops.

A graph of different colored bars

Descripción generada automáticamente

**Plot 2 – Distribution of total PnL (test data)**

Skipped days have a worse PnL distribution: losses are more frequent and larger in magnitude. In contrast, days not skipped exhibit a distribution skewed toward more frequent profitable outcomes, consistent with cleaner, directional moves.

A graph showing the value of a company

Descripción generada automáticamente con confianza media

**First trade hit rate**

* **Skipped days:** 17.5% of days have the first trade profitable.
* **Not skipped days:** 26.8% of days have the first trade profitable.

**Equity curve**

The following plots show accumulated profits through time (something equivalent to an equity curve), for both the original strategy and the strategy if we skip days when composite quintile is 1, for both TRAIN and TEST datasets:

A graph showing the growth of the stock market

Descripción generada automáticamente

A graph showing the growth of the stock market

Descripción generada automáticamente

The following table shows the risk-adjusted returns, both in terms of Sharpe and Calmar, for the original strategy and the strategy skipping trading days when the composite quintile is 1, and for both Train and Test datasets. It can be observed that returns are increased, and volatility and drawdown are reduced. So, risk-adjusted returns are significantly better if we skip those trading days.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data | Annualized return | Annualized Vol | Max Drawdown | Sharpe | Calmar |
| In-Sample (TRAIN) | $57k | $32k | $22k | 1.77 | 2.53 |
| In-Sample (TRAIN), skip if q == 1 | $81k | $29k | $19k | 2.74 | 4.09 |
| Out-Of-Sample (TEST) | $61k | $33k | $22k | 1.83 | 2.78 |
| Out-Of-Sample (TEST), skip if q==1 | $74k | $29k | $17k | 2.54 | 4.20 |

## Alternative approach: continuous sizing strategy

Forward returns appear approximately monotonic with the **combined quintile** (1–5), so instead of a hard “skip” rule I test a continuous sizing rule that scales exposure with the signal.

**Sizing rule.** For each day t, the size is calculated as follows:

This anchors the baseline at s=1 when q=3, down-weights low-quality days and up-weights high-quality days. The bounds can be tightened/loosened if desired.

**Implementation.** I keep the strategy mechanics unchanged (same ORB entries, 500/100 tick TP/SL). Daily PnL scales linearly with the composite quintile ​ (i.e., notional/leverage only is adjusted). No days are skipped in this variant; it is a continuous alternative to the discrete skip filter.

I report the same set of outputs as for the previous strategies, for TRAIN and TEST.

Interpretation guide. If the monotonic relation holds OOS, the scaled variant should (i) tilt exposure toward better days, (ii) defuse risk on poorer days, and (iii) potentially smooth the equity curve and improve risk-adjusted returns versus both baseline and the binary skip rule.

A graph showing the growth of the stock market

Descripción generada automáticamente

A graph showing the growth of the stock market

Descripción generada automáticamente

The following table shows the risk-adjusted returns:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data | Annualized return | Annualized Vol | Max Drawdown | Sharpe | Calmar |
| In-Sample (TRAIN) | $57k | $32k | $22k | 1.77 | 2.53 |
| In-Sample (TRAIN), adjust size | $108k | $34k | $20k | 3.13 | 5.35 |
| Out-Of-Sample (TEST) | $61k | $33k | $22k | 1.83 | 2.78 |
| Out-Of-Sample (TEST), adjust size | $96k | $34k | $20k | 2.79 | 4.68 |

## Forward Walk Optimization: discrete weights for each quintile

To move beyond a linear sizing rule, my first Forward Walk Optimization (FWO) assigns a discrete weight to each value of the composite quintile (1–5). Instead of interpolating between a MIN\_SIZE and a MAX\_SIZE, the daily position size is read directly from a 5-element vector (“combo”):  
q=1 → w1, q=2 → w2, …, q=5 → w5.

Candidate weight grid (with a monotone prior)

* w1 ∈ {0, 0.25, 0.5, 0.75}
* w2 ∈ {0.5, 0.75, 1}
* w3 ∈ {1}
* w4 ∈ {1, 1.25, 1.5}
* w5 ∈ {1.25, 1.5, 1.75, 2}

I generate the Cartesian product of these sets and keep only schedules that are non-decreasing (w1 ≤ w2 ≤ w3 ≤ w4 ≤ w5). This encodes the intuition that higher-quality days should never receive a smaller weight than lower-quality days.

Walk-forward setup

* Windows: 18 months for training, followed by 3 months for testing, rolled forward through time.
* Signal: the composite quintile is computed using expanding transforms (past-only), preventing look-ahead.
* For each training window and for every candidate combo:
  1. Map daily size by quintile: size on each day equals the weight assigned to that day’s quintile.
  2. Reweight daily PnL, build the equity curve, and compute annualized return, annualized volatility, and Sharpe on the training window.
  3. Select the single best-Sharpe combo for that training window.
  4. Apply the selected combo only to the subsequent 3-month test window and store the five weights used out-of-sample.

Out-of-sample construction and reporting

* Concatenate all test windows to form the full OOS period.
* Assign daily size from the stored weights and build the OOS equity curve.
* Compare against the baseline (original strategy, unscaled PnL).
* Report: Annualized return, Annualized volatility, Max Drawdown, Sharpe, and Calmar.
* Visualize: (i) baseline vs. FWO-discrete OOS equity curves, and (ii) the time series of chosen weights w1…w5 to show how sizing adapts over time.

Equity curve is shown in the following plot:

A graph of a graph of a graph

Descripción generada automáticamente con confianza media

Results from backtest:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CAGR | RV | MaxDD | Sharpe | Calmar |
| Original strategy | $53k | $33k | $22k | 1.57 | 2.41 |
| FWO strategy | $127k | $39k | $21k | 3.22 | 5.97 |

The discrete FWO backtest shows a **clear step-up in performance** compared to the original strategy. Annualized returns more than doubled (from about $53k to $127k) while volatility stayed at a similar level, resulting in a Sharpe ratio that improves from 1.57 to 3.22 and a Calmar ratio from 2.41 to 5.97. In other words, this approach not only increased profitability but also delivered substantially better risk-adjusted returns, with lower drawdown relative to gains.

## Forward Walk: linear interpolation between MIN\_SIZE and MAX\_SIZE parameters

As an alternative to the first, discrete-weight FWO, I also tested a **continuous sizing** rule parameterized by two bounds (MIN\_SIZE and MAX\_SIZE). The setup mirrors the previous walk-forward design:

1. **Composite quintile** built from the average of the top features’ quintiles, discretized to 1–5 using an expanding percentile transformation (past-only).
2. **Walk-forward windows:** optimize MIN\_SIZE and MAX\_SIZE on an 18-month training window, then apply the selected parameters to the subsequent 3-month test window.
3. **Selection criterion:** evaluate annualized return, annualized volatility, and Sharpe; instead of taking the single best pair, take the **median of the top 5%** Sharpe solutions for robustness.
4. **OOS application:** apply the chosen parameters only to the test window and compare the resulting equity curve with the baseline (fixed sizing).

**Result:** this continuous sizing FWO nearly doubles the out-of-sample annualized return (from ~$53k to ~$96k) and improves risk-adjusted metrics (Sharpe 1.57 → 2.64, Calmar 2.41 → 4.62), confirming that scaling exposure with the composite quintile enhances performance versus the baseline.

A graph of a graph of a graph

Descripción generada automáticamente con confianza media

Results from backtest:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CAGR | RV | MaxDD | Sharpe | Calmar |
| Original strategy | $53k | $33k | $22k | 1.57 | 2.41 |
| Continuous sizing, FWO | $96k | $36k | $20k | 2.64 | 4.62 |

## Note from FWO analysis

Overall, the evidence favors a **non-linear** mapping from signal to size: the discrete, monotone quintile-weight FWO outperforms the linear MIN/MAX variant across out-of-sample metrics—CAGR ≈ **$127k vs $96k**, Sharpe **3.22 vs 2.64**, Calmar **5.97 vs 4.62**—with **comparable drawdown** (~$21k vs ~$20k). The monotone discrete approach captures tail effects the linear rule misses while remaining intuitive and interpretable, so it is the stronger choice when feasible.

## Recommended next steps

* **Target & labels** – Define the target as **first-trade success** (whether the first trade of the day hits take-profit before stop-loss), using strictly **previous-day features** to avoid leakage.
* **Model & validation** – Train models such as **XGBoost/GBM** with proper **walk-forward splits (18m→3m)** and report **out-of-sample accuracy** on 2024–2025.
* **Decision policy** – Turn probabilities into a simple **trade/skip rule** (trade only if confidence ≥ threshold), and optionally test a **three-bucket policy** (win / recoverable loss / cascade loss).
* **Robustness analysis** – Run **ablation tests, regime slices, and rolling IC/IR checks** to confirm stability and ensure results are not regime-specific or driven by a single feature.