Social Predict - Weighted Probability Adjustment Model (WPAM) for Market Pricing and Divergence-Based Payout Model (DBPM) for Payout Distributions

Patrick Delaney

November 2024

1 Overview

- **Overall Purpose:** The purpose of this document is to describe in clear detail with graphical illustrations how market pricing (probability) gets set and how points/money/units gets distributed from a series of historical bets/transactions.
- **Ab Initio:** Since our software is stateless, the market price and how a pool of points gets distributed should be able to be calculated from the starting point of a string of transactions. There is no need to cache or store calculations along the way.
- **Transparency and Unit Testing:** Given that pricing and distribution is fundamental to how the software works, it is essential to ensure the full algorithm is well understood in order to write proper unit tests and to communicate the rules to users and participants.

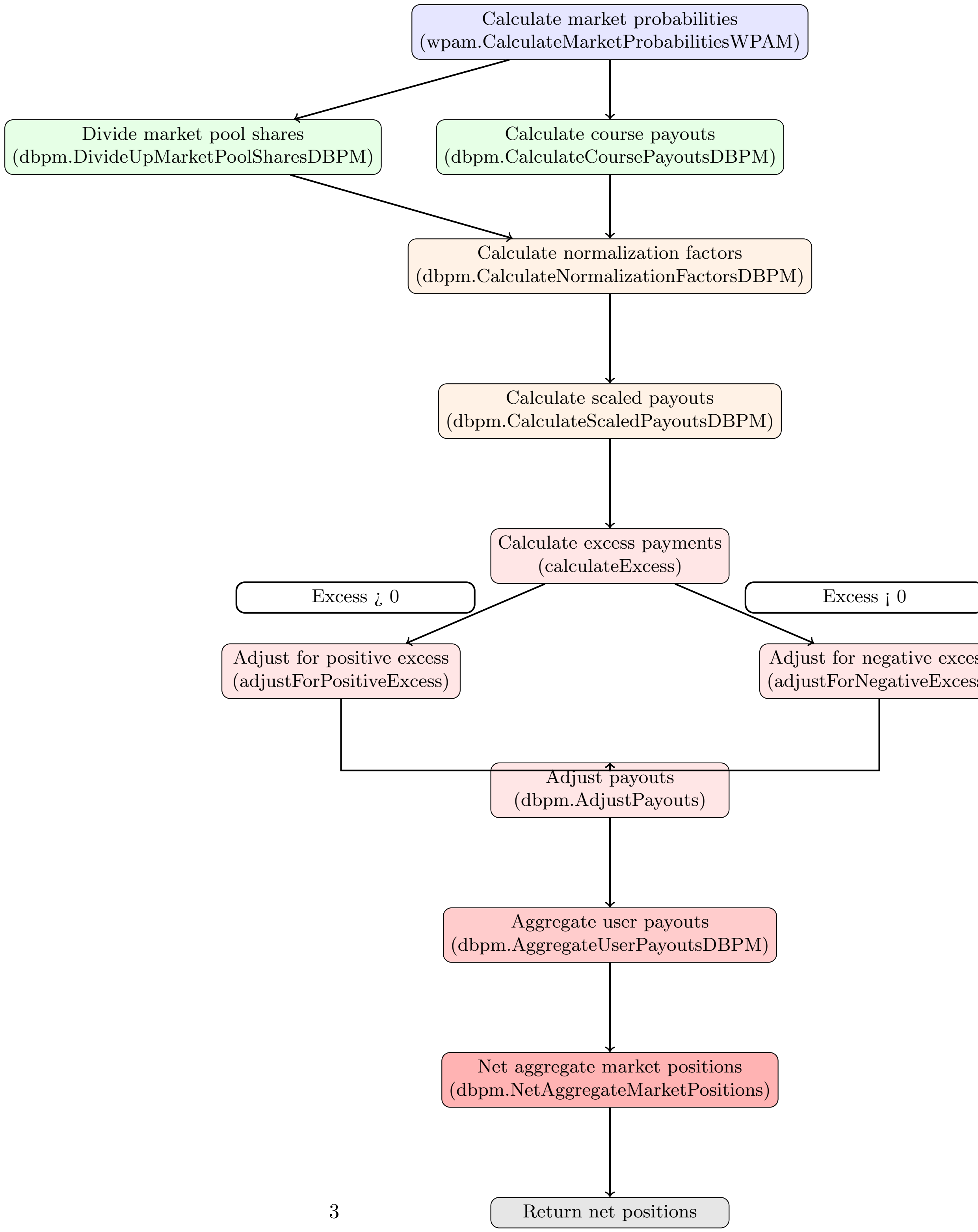
2 Conventions

- **Course Payouts Divide By Zero Condition:** When Course Payouts calculations and further calculations derived from Course Payouts may include an inf (divide by zero) condition, we will define the output as zero. This situation arises in which the Market Value is measured as zero, since everyone has exited the market. Technically the market Value is still equal to the initial investment value until the resolution has been fulfilled, so the Course Payouts, which simply represent a link between the probability and the payout at a given market condition, are not representative of the final state of the market. Thus, calling the Course Payouts and anything calculated from the Course Payouts zero makes sense.

- **Only NO or YES Positions Held:** By convention, a user can only hold either a NO or a YES position, but not both. There is a force sale in the selling function when one attempts to buy the opposite amount of shares currently held. One must neutralize one's own position prior to buying an opposite position.

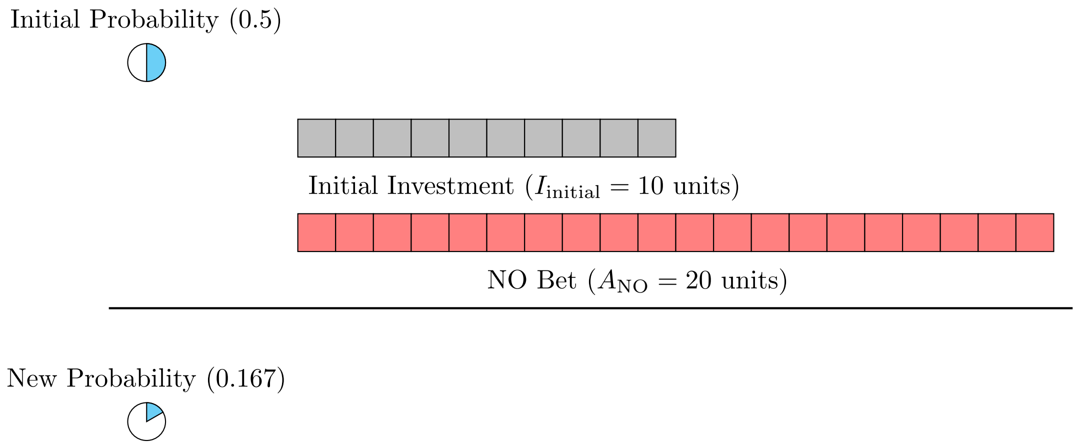
3 Positions Flow Chart

- **Exchange, Transform, Load:** The net positions that every trader has on any given market are calculated through what is basically an Exchange, Transform and Load (ETL) pipeline. We Exchange the data in the form of transactions, Transform it through a series of steps as shown in the chart below, and finally we Load the net positions as an object.
- **Works Anytime:** The ETL pipeline shown in the chart works at any point in the market's history, including after the resolution. Positions could be calculated for a market resolution at any given percentage between 0 pc and 100 pc or at 0 (NO) or 100 (YES) pc resolutions.


Calculate market probabilities
(wpam.CalculateMarketProbabilitiesWPAM)
Divide market pool shares
(dbpm.DivideUpMarketPoolSharesDBPM)
Calculate course payouts
(dbpm.CalculateCoursePayoutsDBPM)
Calculate normalization factors
(dbpm.CalculateNormalizationFactorsDBPM)
Calculate scaled payouts
(dbpm.CalculateScaledPayoutsDBPM)
Calculate excess payments
(calculateExcess)
Excess ¿ 0
Excess ¡ 0
Adjust for positive excess
(adjustForPositiveExcess)
Adjust for negative exces
(adjustForNegativeExces
Adjust payouts
(dbpm.AdjustPayouts)
Aggregate user payouts
(dbpm.AggregateUserPayoutsDBPM)
Net aggregate market positions
(dbpm.NetAggregateMarketPositions)
Return net positions

4 First Transaction

- **Initial Investment:** Every market is assumed to have an initial investment of at least 1. This investment comes from a fee assessed to the market creator. This initial investment goes both in the numerator and denominator of the probability calculation.
- **Initial Probability:** Every market starts out at a probability of 0.5 which means even odds between a YES and NO outcome.
- **First Transaction:** In our sample below, the first transaction is a bet of 20 Units in the NO direction.


Initial Probability (0.5)
Initial Investment ($I_{\text{initial}} = 10$ units)
NO Bet ($A_{\text{NO}} = 20$ units)
New Probability (0.167)

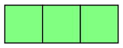
$$P_{\text{new}} = \frac{P_{\text{initial}} \times I_{\text{initial}} + A_{\text{YES}}}{I_{\text{initial}} + A_{\text{YES}} + A_{\text{NO}}} = \frac{0.5 \times 10 + 0}{10 + 0 + 20} = \frac{5}{30} \approx 0.167$$

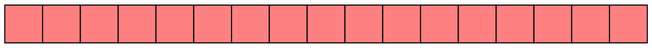
Market Share Division

- **Banker's Rounded Share Pool:** First we look at how many total shares of YES and NO exist in the market pool as a whole by using the WPAM. The total share pool is equal to the Total Market Volume, not including the initial investment. The total share pool times the Probability will be the YES shares, while 1-(Probability) times the total share pool will be the NO shares.
- **Recuperating Initial Investment:** The initial investment paid in by the market maker can be recuperated at the end after the resolution has been made. In effect, the initial market investment is held in escrow in exchange for a fee by the computer. After the resolution, the market maker bonus can be paid out and the initial investment will be redistributed to all market participants.

$$S = \text{Total Market Volume} = 20 \text{ units}$$

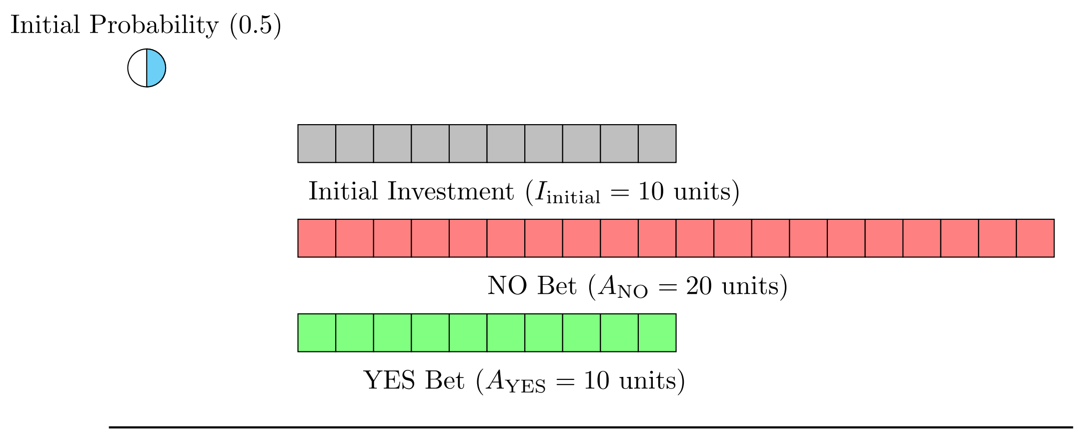
$$\begin{aligned} S_{\text{YES}} &= \lfloor S \times P_{\text{new}} \rceil = \lfloor 20 \times 0.167 \rceil = 3 \\ S_{\text{NO}} &= \lfloor S \times (1 - P_{\text{new}}) \rceil = \lfloor 20 \times 0.833 \rceil = 17 \end{aligned}$$

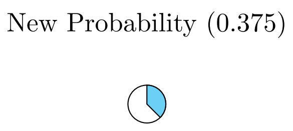

YES Shares ($S_{\text{YES}} = 3$)


NO Shares ($S_{\text{NO}} = 17$)

5 Second Transaction

- **Second Transaction:** A different bettor comes in and places 10 units in the YES direction.


Initial Probability (0.5)
Initial Investment ($I_{\text{initial}} = 10$ units)
NO Bet ($A_{\text{NO}} = 20$ units)
YES Bet ($A_{\text{YES}} = 10$ units)

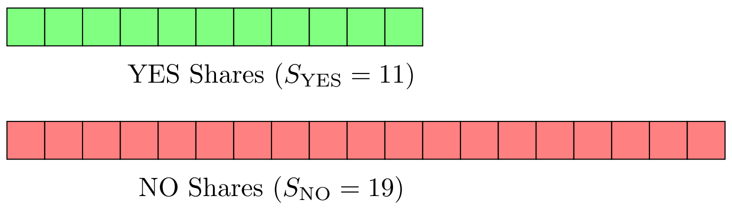


$$P_{\text{new}} = \frac{0.5 \times 10 + 10}{10 + 10 + 20} = \frac{15}{40} \approx 0.375$$

Market Share Division

$$S = \text{Total Market Volume} + I_{\text{initial}} = (10 + 20) = 30 \text{ units}$$

$$S_{\text{YES}} = \lfloor S \times P_{\text{new}} \rceil = \lfloor 30 \times 0.375 \rceil = 11$$
$$S_{\text{NO}} = \lfloor S \times (1 - P_{\text{new}}) \rceil = \lfloor 30 \times 0.625 \rceil = 19$$


YES Shares ($S_{\text{YES}} = 11$)
NO Shares ($S_{\text{NO}} = 19$)

Calculating Course Payouts

- **Course Payouts:** The purpose of the Course Payouts is to create a relationship between the probability, which is a float, rational positive numbers, and shares, which are of type int64, integers. We use a linear weight for each historical bet in the history of the market to create a reward factor.
- **Reward Factor:** Course payouts are calculated in two steps, first by calculating a reward factor d_i, which is the distance from the current probability of the market R, and the probability at which the bet was made p_i.
- **New Probability After Bet, Not Previous Probability:** The current probability on the market, R is the New Probability which was just calculated by our WPAM function, $wpam.CalculateMarketProbabilitiesWPAM$ given the bet that was just immediately made. This enforces the idea that there is no reward for simply moving the market oneself and then selling at a new probability created from that movement. In order to profit, someone else must come in and move the market further in one's favor to be able to sell.

Step One: Calculate Reward Factor for Each Bet

$$d_i = |R - p_i|$$

Step Two: Calculate Course Payout for Each Bet

$$C_i = d_i \times b_i$$

Example Calculations

Given:

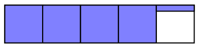
- Resolution Probability: $R = 0.375$
- Bet 0: Neutral, $b_0 = 10$ units, $p_0 = 0.5$
- Bet 1: NO, $b_1 = 20$ units, $p_1 = 0.167$
- Bet 2: YES, $b_2 = 10$ units, $p_2 = 0.375$

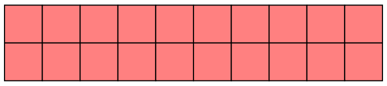
For Bet 1:

$$d_1 = |R - p_1| = |0.375 - 0.167| = 0.208$$
$$C_1 = d_1 \times b_1 = 0.208 \times 20 = 4.160 \text{ units}$$

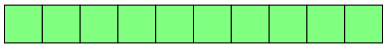
For Bet 2:

$$d_2 = |R - p_2| = |0.375 - 0.375| = 0$$
$$C_2 = d_2 \times b_2 = 0 \times 10 = 0 \text{ units}$$


Course Payout for Bet 1 ($C_1 = 4.16$ units)


Bet 1: NO ($b_1 = 10$ units at $p_1 = 0.167$)

Course Payout for Bet 2 ($C_2 = 0$ units)


Bet 2: YES ($b_2 = 10$ units at $p_2 = 0.375$)

Normalization Factor Calculation

The normalization factor ensures that course payouts are proportional to the total shares for each outcome, making sure that payouts align with the available market pool.

- The normalization factor F_{YES} and F_{NO} adjusts each outcome's payouts so that they fit within the total shares allocated for each outcome.
- This prevents payouts from exceeding the total market pool, ensuring fair distribution based on the market's share allocation.
- By scaling payouts, the normalization factor maintains balance between YES and NO outcomes, reflecting the distribution of bets and probabilities in the market.

Step Three: Calculate Normalization Factor

- At this point the normalization factor will weight everything toward NO because there are zero course payouts for YES.

Given:

- Total YES Shares: $S_{\text{YES}} = 11$
- Total NO Shares: $S_{\text{NO}} = 19$
- Course Payouts:
  - $C_{\text{NO},1} = 4.16$ units
  - $C_{\text{YES},2} = 0$ units
  - Total payout sums: $C_{\text{YES\_SUM}} = 0$, $C_{\text{NO\_SUM}} = 4.16$

Calculate YES Normalization Factor

$$F_{\text{YES}} = \begin{cases} \frac{S_{\text{YES}}}{C_{\text{YES\_SUM}}} & \text{if } C_{\text{YES\_SUM}} > 0 \\ 0 & \text{if } C_{\text{YES\_SUM}} = 0 \end{cases}$$

Since $C_{\text{YES\_SUM}} = 0$, we have:

$$F_{\text{YES}} = 0$$

Calculate NO Normalization Factor

$$F_{\text{NO}} = \begin{cases} \frac{S_{\text{NO}}}{C_{\text{NO\_SUM}}} & \text{if } C_{\text{NO\_SUM}} > 0 \\ 0 & \text{if } C_{\text{NO\_SUM}} = 0 \end{cases}$$

Since $C_{\text{NO\_SUM}} = 4.16$, we have:

$$F_{\text{NO}} = \frac{19}{4.16} \approx 4.56$$

Thus, the normalization factors are:

$$F_{\text{YES}} = 0, \quad F_{\text{NO}} \approx 4.56$$

Scaled Payout Calculation for Second Transaction

- **Purpose of Scaled Payouts:**
  - Scaled payouts use normalization factors to adjust the raw course payouts, ensuring they align proportionally with the total market shares for each outcome.
  - This prevents over-distribution of payouts and ensures fairness in the final allocations.
  - By scaling payouts, the process reflects market probabilities and bet contributions accurately while adhering to the market's constraints.

Step Four: Calculate Scaled Payout for Each Bet

Given:

- Normalization Factors: $F_{\text{YES}} = 0$, $F_{\text{NO}} = 6.01$
- Course Payouts:
  - $C_{\text{NO},1} = 4.16$ units
  - $C_{\text{YES},2} = 0$ units

For Bet 1 (NO):

$$\text{Scaled Payout}_{\text{NO},1} = \text{Round}(C_{\text{NO},1} \times F_{\text{NO}})$$

$$\text{Scaled Payout}_{\text{NO},1} = \text{Round}(4.16 \times 4.56) = \text{Round}(18.96) = 19 \text{ units}$$

For Bet 2 (YES):

$$\text{Scaled Payout}_{\text{YES},2} = \text{Round}(C_{\text{YES},2} \times F_{\text{YES}})$$

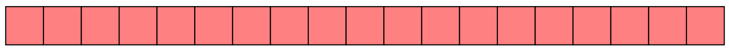
$$\text{Scaled Payout}_{\text{YES},2} = \text{Round}(0 \times 0) = \text{Round}(0) = 0 \text{ units}$$

Final Scaled Payouts:

Bet 1 (NO): 19 units

Bet 2 (YES): 0 units

Scaled Payout for Bet 1 (19 units)



Scaled Payout for Bet 2 (0 units)

Adjust Payouts Equations

Calculate Excess

The excess is the difference between the total scaled payouts and the available pool of funds in the market:

$$\text{Excess} = \sum_{i=1}^{n} \text{ScaledPayout}_i - \text{AvailablePool}$$

Where:

- ScaledPayout_i: The scaled payout for the i-th bet.
- AvailablePool: The total available pool of funds, calculated as:

$$\text{AvailablePool} = \text{MarketVolume}(\text{Bets})$$

Calculate Excess

Given:

- Scaled Payouts:
  - $\text{Scaled Payout}_{\text{NO},1} = 19$ units
  - $\text{Scaled Payout}_{\text{YES},2} = 0$ units
- Available Pool:

$$\text{Available Pool} = \text{Sum of Bets} = 20 + 10 = 30 \text{ units}$$

Excess Calculation:

$$\text{Excess} = \sum_{i=1}^{n} \text{Scaled Payout}_i - \text{Available Pool}$$

$$\text{Excess} = (19 + 0) - 30$$

$$\text{Excess} = 19 - 30 = -11 \text{ units}$$

Interpretation:
Since the calculated **Excess** is negative ($\text{Excess} < 0$), this indicates that the scaled payouts are 11 units below the available pool, requiring adjustments to redistribute this shortfall across the payouts to align with the total pool.

Adjust for Positive Excess

If the calculated excess is positive ($\text{Excess} > 0$), we adjust the payouts by deducting from the newest payouts (starting from the last):

```
If $\text{Excess} > 0$:
    For $i = n, n-1, \ldots, 1,$ while $\text{Excess} > 0$:
        If $\text{ScaledPayout}_i > 0,$ then:
            $\text{ScaledPayout}_i \leftarrow \text{ScaledPayout}_i - 1$
            $\text{Excess} \leftarrow \text{Excess} - 1$
```

The adjustment continues until $\text{Excess} = 0$.

In our scenario above, we have no positive excess so we don't need to calculate it.

Adjust for Negative Excess

If the calculated excess is negative ($\text{Excess} < 0$), we adjust the payouts by adding to the oldest payouts (starting from the first):

```
If $\text{Excess} < 0$:
    For $i = 1, 2, \ldots, n,$ while $\text{Excess} < 0$:
        $\text{ScaledPayout}_i \leftarrow \text{ScaledPayout}_i + 1$
        $\text{Excess} \leftarrow \text{Excess} + 1$
```

The adjustment continues until $\text{Excess} = 0$.

Adjust for Negative Excess

If the calculated excess is negative ($\text{Excess} < 0$), we adjust the payouts by incrementing all payouts cyclically, starting from the oldest, until the excess is resolved:

```
If $\text{Excess} < 0$:
    While $\text{Excess} < 0$:
        For $i = 1, 2, \ldots, n$:
            $\text{ScaledPayout}_i \leftarrow \text{ScaledPayout}_i + 1$
            $\text{Excess} \leftarrow \text{Excess} + 1$
```

Given:

- Initial Scaled Payouts:

$$\text{Scaled Payout}_{\text{NO},1} = 19, \quad \text{Scaled Payout}_{\text{YES},2} = 0$$

- Initial Excess:

$$\text{Excess} = -11$$

Step-by-Step Adjustment:

We cycle through the payouts, adding +1 to each in turn, until the excess is resolved:

- Cycle 1:

$$\begin{aligned}\text{Scaled Payout}_{\text{NO},1} &= 19 + 1 = 20\\ \text{Scaled Payout}_{\text{YES},2} &= 0 + 1 = 1\\ \text{Excess} &= -11 + 2 = -9\end{aligned}$$

- Cycle 2:

$$\begin{aligned}\text{Scaled Payout}_{\text{NO},1} &= 20 + 1 = 21\\ \text{Scaled Payout}_{\text{YES},2} &= 1 + 1 = 2\\ \text{Excess} &= -9 + 2 = -7\end{aligned}$$

- Cycle 3:

$$\begin{aligned}\text{Scaled Payout}_{\text{NO},1} &= 21 + 1 = 22\\ \text{Scaled Payout}_{\text{YES},2} &= 2 + 1 = 3\\ \text{Excess} &= -7 + 2 = -5\end{aligned}$$

- Cycle 4:

$$\begin{aligned}\text{Scaled Payout}_{\text{NO},1} &= 22 + 1 = 23\\ \text{Scaled Payout}_{\text{YES},2} &= 3 + 1 = 4\\ \text{Excess} &= -5 + 2 = -3\end{aligned}$$

- Cycle 5:

$$\begin{aligned}\text{Scaled Payout}_{\text{NO},1} &= 23 + 1 = 24\\ \text{Scaled Payout}_{\text{YES},2} &= 4 + 1 = 5\\ \text{Excess} &= -3 + 2 = -1\end{aligned}$$

- Final Cycle:

$$\begin{aligned}\text{Scaled Payout}_{\text{NO},1} &= 24 + 1 = 25\\ \text{Scaled Payout}_{\text{YES},2} &= 5 \quad \text{(unchanged)}\\ \text{Excess} &= -1 + 1 = 0\end{aligned}$$

Final Adjusted Scaled Payouts:

$$\text{Scaled Payout}_{\text{NO},1} = 25, \quad \text{Scaled Payout}_{\text{YES},2} = 5$$

Adjusted Excess:

$$\text{Excess} = (25 + 5) - 30 = 30 - 30 = 0$$

Interpretation:

The excess of -11 has been evenly redistributed across the payouts by cyclically adding $+1$ to each, starting with the oldest (NO,1), until the excess reached 0.

Final Adjusted Scaled Payouts:

$$\text{Scaled Payout}_{\text{NO},1} = 25, \quad \text{Scaled Payout}_{\text{YES},2} = 5$$

Adjusted Excess:

$$\text{Excess} = (25 + 5) - 30 = 30 - 30 = 0$$

Interpretation:

The negative excess has been resolved by distributing the excess equally between the payouts for NO and YES, ensuring the total scaled payouts match the available pool.

Adjust Payouts Function

The overall adjustment process is summarized as follows:

$$\text{Excess} \leftarrow \sum_{i=1}^{n} \text{ScaledPayout}_i - \text{AvailablePool}$$

$$\text{If Excess} > 0:$$

$$\text{Adjust payouts using AdjustForPositiveExcess.}$$

$$\text{Else if Excess} < 0:$$

$$\text{Adjust payouts using AdjustForNegativeExcess.}$$

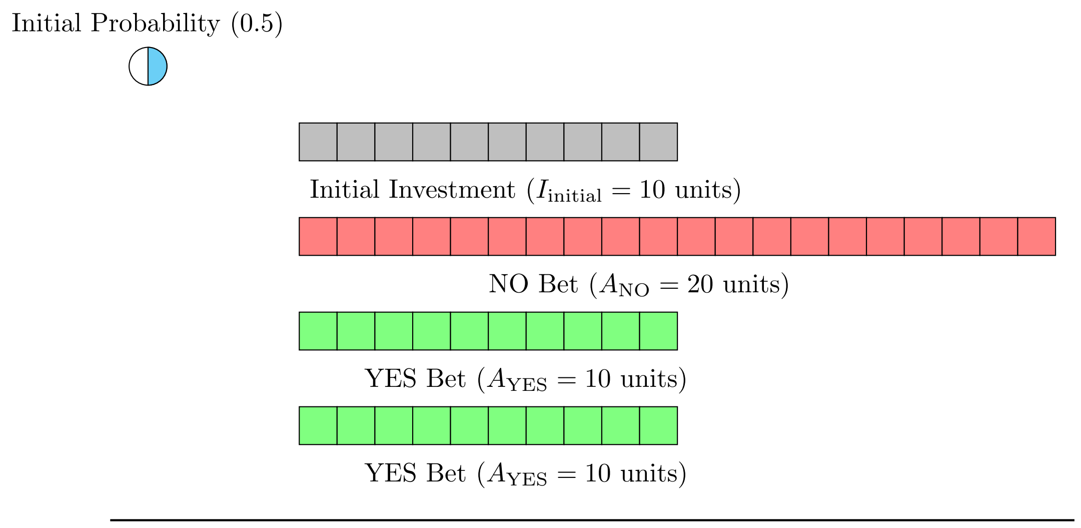
$$\text{Return Adjusted ScaledPayouts.}$$

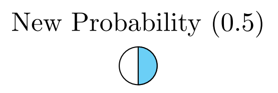
Aggregate User Payouts

We will defer calculating aggregated user payouts until the last transaction in this paper for simplicity sake.

6 Third Transaction

- **Third Transaction:** Another bettor places 10 units in the YES direction.


Initial Probability (0.5)
Initial Investment ($I_{\text{initial}} = 10$ units)
NO Bet ($A_{\text{NO}} = 20$ units)
YES Bet ($A_{\text{YES}} = 10$ units)
YES Bet ($A_{\text{YES}} = 10$ units)

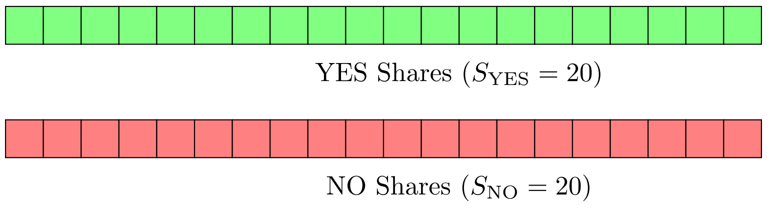


$$P_{\text{new}} = \frac{0.5 \times 10 + 20}{10 + 20 + 20} = \frac{25}{50} = 0.5$$

Market Share Division

$$S = \text{Total Market Volume} + I_{\text{initial}} = (20 + 20) = 40 \text{ units}$$

$$S_{\text{YES}} = \lfloor S \times P_{\text{new}} \rceil = \lfloor 50 \times 0.5 \rceil = 20$$
$$S_{\text{NO}} = \lfloor S \times (1 - P_{\text{new}}) \rceil = \lfloor 50 \times 0.5 \rceil = 20$$


YES Shares ($S_{\text{YES}} = 20$)
NO Shares ($S_{\text{NO}} = 20$)

Calculating Course Payouts

Step One: Calculate Reward Factor for Each Bet

$$d_i = |R - p_i|$$

Step Two: Calculate Course Payout for Each Bet

$$C_i = d_i \times b_i$$

Example Calculations for Third Transaction

Given:

- Resolution Probability: $R = 0.5$
- Bet 0: Neutral, $b_0 = 10$ units, $p_0 = 0.5$
- Bet 1: NO, $b_1 = 20$ units, $p_1 = 0.167$
- Bet 2: YES, $b_2 = 10$ units, $p_2 = 0.375$
- Bet 3: YES, $b_3 = 10$ units, $p_3 = 0.5$

For Bet 1:

$$d_1 = |R - p_1| = |0.5 - 0.167| = 0.333$$
$$C_1 = d_1 \times b_1 = 0.333 \times 20 = 6.660 \text{ units}$$

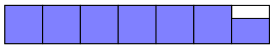
For Bet 2:

$$d_2 = |R - p_2| = |0.5 - 0.375| = 0.125$$
$$C_2 = d_2 \times b_2 = 0.125 \times 10 = 1.250 \text{ units}$$

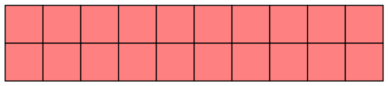
For Bet 3:

$$d_3 = |R - p_3| = |0.5 - 0.5| = 0$$
$$C_3 = d_3 \times b_3 = 0 \times 10 = 0 \text{ units}$$

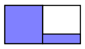
Course Payout for Bet 1 ($C_1 = 6.660$ units)



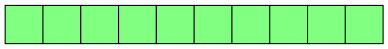
Bet 1: NO ($b_1 = 20$ units at $p_1 = 0.167$)



Course Payout for Bet 2 ($C_1 = 1.25$ units)

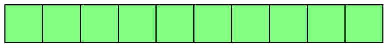


Bet 2: YES ($b_1 = 10$ units at $p_1 = 0.375$)



Course Payout for Bet 3 ($C_1 = 0.0$ units)

Bet 3: YES ($b_1 = 10$ units at $p_1 = 0.5$)



Normalization Factor Calculation for Third Transaction

Step Three: Calculate Normalization Factor

Given:

- Total YES Shares: $S_{\text{YES}} = 25$
- Total NO Shares: $S_{\text{NO}} = 25$
- Course Payouts:
  - $C_{\text{NO},1} = 6.66$ units
  - $C_{\text{YES},1} = 1.25$ units
  - $C_{\text{YES},2} = 0$ units
  - Total payout sums: $C_{\text{YES\_SUM}} = 1.25$, $C_{\text{NO\_SUM}} = 6.66$

Calculate YES Normalization Factor

$$F_{\text{YES}} = \begin{cases} \frac{S_{\text{YES}}}{C_{\text{YES\_SUM}}} & \text{if } C_{\text{YES\_SUM}} > 0 \\ 0 & \text{if } C_{\text{YES\_SUM}} = 0 \end{cases}$$

Since $C_{\text{YES\_SUM}} = 1.25$, we have:

$$F_{\text{YES}} = \frac{20}{1.25} = 16$$

Calculate NO Normalization Factor

$$F_{\text{NO}} = \begin{cases} \frac{S_{\text{NO}}}{C_{\text{NO\_SUM}}} & \text{if } C_{\text{NO\_SUM}} > 0 \\ 0 & \text{if } C_{\text{NO\_SUM}} = 0 \end{cases}$$

Since $C_{\text{NO\_SUM}} = 6.66$, we have:

$$F_{\text{NO}} = \frac{20}{6.66} \approx 3.00$$

Thus, the normalization factors are:

$$F_{\text{YES}} = 16, \quad F_{\text{NO}} \approx 3.00$$

Scaled Payout Calculation for Third Transaction

- **Purpose of Scaled Payouts:**
  - Scaled payouts adjust raw course payouts proportionally based on normalization factors to ensure payouts align with the available shares in the market pool.
  - The adjustment ensures fairness, reflecting the market's allocation of shares while staying within constraints.
  - Scaled payouts provide accurate distributions of rewards based on contributions and probabilities in the market.

Step Four: Calculate Scaled Payout for Each Bet

Given:

- Normalization Factors: $F_{\text{YES}} = 20$, $F_{\text{NO}} \approx 3.75$
- Course Payouts:
  - $C_{\text{NO},1} = 6.66$ units
  - $C_{\text{YES},2} = 1.25$ units
  - $C_{\text{YES},3} = 0$ units

For Bet 1 (NO):

$$\text{Scaled Payout}_{\text{NO},1} = \text{Round}(C_{\text{NO},1} \times F_{\text{NO}})$$

$$\text{Scaled Payout}_{\text{NO},1} = \text{Round}(6.66 \times 3.00) = \text{Round}(20.000) = 20 \text{ units}$$

For Bet 2 (YES):

$$\text{Scaled Payout}_{\text{YES},2} = \text{Round}(C_{\text{YES},2} \times F_{\text{YES}})$$

$$\text{Scaled Payout}_{\text{YES},2} = \text{Round}(1.25 \times 16) = \text{Round}(20.00) = 20 \text{ units}$$

For Bet 3 (YES):

$$\text{Scaled Payout}_{\text{YES},3} = \text{Round}(C_{\text{YES},3} \times F_{\text{YES}})$$

$$\text{Scaled Payout}_{\text{YES},3} = \text{Round}(0 \times 20) = \text{Round}(0) = 0 \text{ units}$$

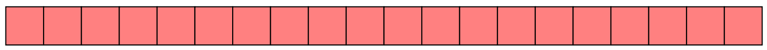
Final Scaled Payouts:

Bet 1 (NO): 20 units

Bet 2 (YES): 20 units

Bet 3 (YES): 0 units

Scaled Payout for Bet 1 (20 units)



Scaled Payout for Bet 2 (20 units)

Scaled Payout for Bet 3 (0 units)

Adjust Payouts Equations

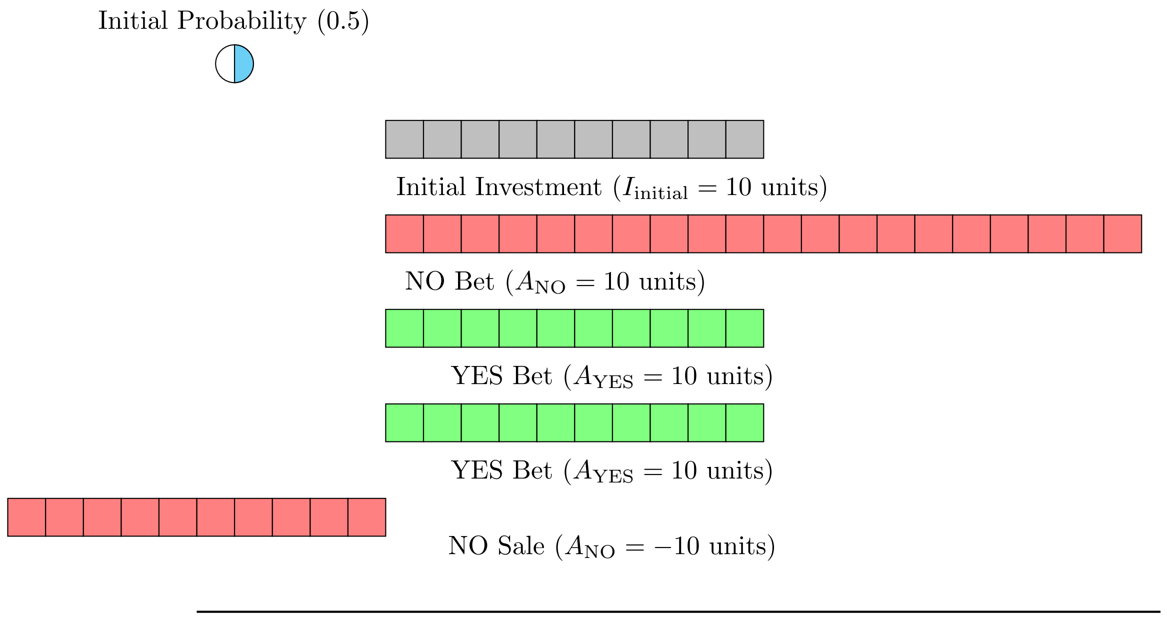
There are no excess shares in our scenario so payout adjustment has no effect at this point.

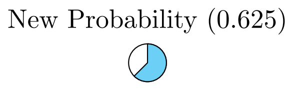
Aggregate User Payouts

We will defer calculating aggregated user payouts until the last transaction in this paper for simplicity sake.

7 Fourth Transaction

- **Fourth Transaction:** The original bettor holding NO sells 10 units of NO.


Initial Probability (0.5)
Initial Investment ($I_{\text{initial}} = 10$ units)
NO Bet ($A_{\text{NO}} = 10$ units)
YES Bet ($A_{\text{YES}} = 10$ units)
YES Bet ($A_{\text{YES}} = 10$ units)
NO Sale ($A_{\text{NO}} = -10$ units)

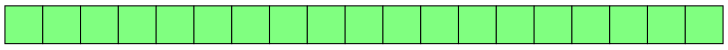


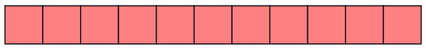
$$P_{\text{new}} = \frac{0.5 \times 10 + 20}{10 + 20} = \frac{25}{40} = 0.625$$

Market Share Division

$$S = \text{Total Market Volume} + I_{\text{initial}} = (20 + 10) = 30 \text{ units}$$

$$S_{\text{YES}} = \lfloor S \times P_{\text{new}} \rceil = \lfloor 30 \times 0.625 \rceil = 19$$
$$S_{\text{NO}} = \lfloor S \times (1 - P_{\text{new}}) \rceil = \lfloor 30 \times 0.375 \rceil = 11$$


YES Shares ($S_{\text{YES}} = 19$)


NO Shares ($S_{\text{NO}} = 11$)

Calculating Course Payouts

Step One: Calculate Reward Factor for Each Bet

$$d_i = |R - p_i|$$

Step Two: Calculate Course Payout for Each Bet

$$C_i = d_i \times b_i$$

Example Calculations for Fourth Transaction

Given:

- Resolution Probability: $R = 0.625$
- Bet 0: Neutral, $b_0 = 10$ units, $p_0 = 0.5$
- Bet 1: NO, $b_1 = 20$ units, $p_1 = 0.167$
- Bet 2: YES, $b_2 = 10$ units, $p_2 = 0.375$
- Bet 3: YES, $b_3 = 10$ units, $p_3 = 0.5$
- Bet 4: NO Sale, $b_4 = -10$ units, $p_4 = 0.625$

For Bet 1:

$$d_1 = |R - p_1| = |0.625 - 0.167| = 0.458$$
$$C_1 = d_1 \times b_1 = 0.458 \times 20 = 9.16 \text{ units}$$

For Bet 2:

$$d_2 = |R - p_2| = |0.625 - 0.375| = 0.25$$
$$C_2 = d_2 \times b_2 = 0.25 \times 10 = 2.5 \text{ units}$$

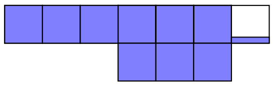
For Bet 3:

$$d_3 = |R - p_3| = |0.625 - 0.5| = 0.125$$
$$C_3 = d_3 \times b_3 = 0.125 \times 10 = 1.25 \text{ units}$$

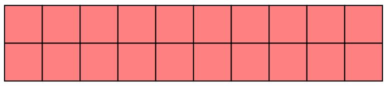
For Bet 4:

$$d_4 = |R - p_4| = |0.625 - 0.625| = 0$$
$$C_4 = d_4 \times b_4 = 0 \times (-10) = 0 \text{ units}$$

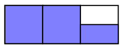
Course Payout for Bet 1 ($C_1 = 9.16$ units)



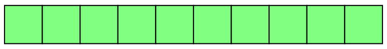
Bet 1: NO ($b_1 = 20$ units at $p_1 = 0.167$)



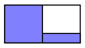
Course Payout for Bet 2 ($C_1 = 2.5$ units)



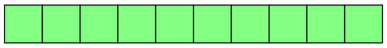
Bet 2: YES ($b_1 = 10$ units at $p_1 = 0.375$)



Course Payout for Bet 3 ($C_1 = 1.25$ units)

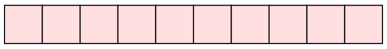


Bet 3: YES ($b_1 = 10$ units at $p_1 = 0.5$)



Course Payout for Bet 4 ($C_1 = 0.0$ units)

Bet 4: NO ($b_1 = -10$ units at $p_1 = 0.625$)



Normalization Factor Calculation for Fourth Transaction

Step Three: Calculate Normalization Factor

Given:

- Total YES Shares: $S_{\text{YES}} = 19$
- Total NO Shares: $S_{\text{NO}} = 11$
- Course Payouts:
  - $C_{\text{YES},1} = 2.5$ units
  - $C_{\text{NO},1} = 9.16$ units
  - $C_{\text{YES},2} = 1.25$ units
  - $C_{\text{NO},2} = 0.0$ units
  - Total payout sums: $C_{\text{YES\_SUM}} = 3.75$, $C_{\text{NO\_SUM}} = 9.16$

Calculate YES Normalization Factor

$$F_{\text{YES}} = \begin{cases} \frac{S_{\text{YES}}}{C_{\text{YES\_SUM}}} & \text{if } C_{\text{YES\_SUM}} > 0 \\ 0 & \text{if } C_{\text{YES\_SUM}} = 0 \end{cases}$$

Since $C_{\text{YES\_SUM}} = 3.75$, we have:

$$F_{\text{YES}} = \frac{19}{3.75} = 5.067$$

Calculate NO Normalization Factor

$$F_{\text{NO}} = \begin{cases} \frac{S_{\text{NO}}}{C_{\text{NO\_SUM}}} & \text{if } C_{\text{NO\_SUM}} > 0 \\ 0 & \text{if } C_{\text{NO\_SUM}} = 0 \end{cases}$$

Since $C_{\text{NO\_SUM}} = 9.16$, we have:

$$F_{\text{NO}} = \frac{11}{9.16} \approx 1.200$$

Thus, the normalization factors are:

$$F_{\text{YES}} \approx 5.067, \quad F_{\text{NO}} \approx 1.200$$

Scaled Payout Calculation for Fourth Transaction

- **Fourth Transaction:**
  - We can now see the effects of the course payout on scaling toward older transactions. While both Bet 2 and Bet 3 were of equal amounts, because Bet 2 was made at a point when the YES outcome was less likely based upon the price, there is proportionally more weight given to this bet.
  - The fourth bet is a negative amount because it was a sale. Because the fourth bet occurs at the resolution price R, the Scaled Payout is zero. However in a future transaction, the Scaled Payout for this transaction could become negative, which would reduce the user's final payout, because they are no longer participating in the market at this amount.

Step Four: Calculate Scaled Payout for Each Bet

Given:

- Normalization Factors: $F_{\text{YES}} \approx 5.067$, $F_{\text{NO}} \approx 1.200$
- Course Payouts:
  - $C_{\text{NO},1} = 9.16$ units

- $C_{\text{YES},2} = 2.5$ units
- $C_{\text{YES},3} = 1.25$ units
- $C_{\text{NO},2} = 0.0$ units

For Bet 1 (NO):

$$\text{Scaled Payout}_{\text{NO},1} = \text{Round}(C_{\text{NO},1} \times F_{\text{NO}})$$
$$\text{Scaled Payout}_{\text{NO},1} = \text{Round}(9.16 \times 1.200) = \text{Round}(15.03) = 11 \text{ units}$$

For Bet 2 (YES):

$$\text{Scaled Payout}_{\text{YES},2} = \text{Round}(C_{\text{YES},2} \times F_{\text{YES}})$$
$$\text{Scaled Payout}_{\text{YES},2} = \text{Round}(2.5 \times 5.067) = \text{Round}(16.68) = 13 \text{ units}$$

For Bet 3 (YES):

$$\text{Scaled Payout}_{\text{YES},3} = \text{Round}(C_{\text{YES},3} \times F_{\text{YES}})$$
$$\text{Scaled Payout}_{\text{YES},3} = \text{Round}(1.25 \times 5.067) = \text{Round}(8.34) = 6 \text{ units}$$

For Bet 4 (NO):

$$\text{Scaled Payout}_{\text{NO},2} = \text{Round}(C_{\text{NO},2} \times F_{\text{NO}})$$
$$\text{Scaled Payout}_{\text{NO},2} = \text{Round}(0.0 \times 1.200) = \text{Round}(0) = 0 \text{ units}$$

Final Scaled Payouts:

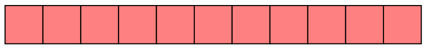
Bet 1 (NO): 11 units

Bet 2 (YES): 13 units

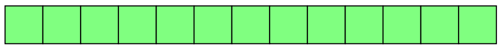
Bet 3 (YES): 6 units

Bet 4 (NO): 0 units

Scaled Payout for Bet 1 (11 units)

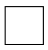


Scaled Payout for Bet 2 (13 units)



Scaled Payout for Bet 3 (6 units)

Scaled Payout for Bet 4 (0 units)



Adjust Payouts Equations

There are no excess shares in our scenario so payout adjustment has no effect at this point.

Aggregate User Payouts

Aggregation is simply taking the list of adjusted payouts and assigning it to the original user. The aggregation of user payouts is defined as follows:

Input Definitions

- $\text{Bets} = \{\text{Bet}_i \mid i = 1, 2, \ldots, n\}$: A set of bets, where each bet Bet_i has the following attributes:
  - Username_i: The username of the user who placed the bet.
  - $\text{Outcome}_i \in \{\text{YES}, \text{NO}\}$: The outcome of the bet.
- $\text{FinalPayouts} = \{P_i \mid i = 1, 2, \ldots, n\}$: The final scaled payouts for each bet.

Aggregation Equation

For each user User_u, the total shares for YES and NO outcomes are calculated as follows:

$$\text{YesSharesOwned}_u = \sum_{\substack{i \\ \text{Username}_i = \text{User}_u \\ \text{Outcome}_i = \text{YES}}} P_i$$

$$\text{NoSharesOwned}_u = \sum_{\substack{i \\ \text{Username}_i = \text{User}_u \\ \text{Outcome}_i = \text{NO}}} P_i$$

Negative Share Adjustment

If any aggregated share count is negative, it is adjusted to zero:

$$\text{YesSharesOwned}_u = \begin{cases} \text{YesSharesOwned}_u & \text{if YesSharesOwned}_u \geq 0 \\ 0 & \text{if YesSharesOwned}_u < 0 \end{cases}$$

$$\text{NoSharesOwned}_u = \begin{cases} \text{NoSharesOwned}_u & \text{if NoSharesOwned}_u \geq 0 \\ 0 & \text{if NoSharesOwned}_u < 0 \end{cases}$$

Output

The final output is a set of market positions:

$$\text{Positions} = \{\text{MarketPosition}_u \mid u = 1, 2, \ldots, m\}$$

where each market position is represented as:

$$\text{MarketPosition}_u = (\text{Username}_u, \text{YesSharesOwned}_u, \text{NoSharesOwned}_u)$$

Normalize Market Positions

The normalization of market positions ensures that for each user, only one of YesSharesOwned or NoSharesOwned is greater than zero, with the other being set to zero. The resulting value represents the net difference between the two. This is in line with the convention that users can only hold YES or NO shares but not both simultaneously on the same market.

Input Definitions

- $\text{Positions} = \{\text{MarketPosition}_u \mid u = 1, 2, \ldots, m\}$: A set of market positions, where each position MarketPosition_u has the following attributes:
  - Username_u: The username of the user.
  - YesSharesOwned_u: The total YES shares owned by the user.
  - NoSharesOwned_u: The total NO shares owned by the user.

Normalization Equation

For each user User_u, the normalized market position is calculated as:

$$\text{If } \text{YesSharesOwned}_u > \text{NoSharesOwned}_u :$$
$$\text{YesSharesOwned}_u^{\text{Normalized}} = \text{YesSharesOwned}_u - \text{NoSharesOwned}_u$$
$$\text{NoSharesOwned}_u^{\text{Normalized}} = 0$$

$$\text{If } \text{NoSharesOwned}_u \geq \text{YesSharesOwned}_u :$$
$$\text{NoSharesOwned}_u^{\text{Normalized}} = \text{NoSharesOwned}_u - \text{YesSharesOwned}_u$$
$$\text{YesSharesOwned}_u^{\text{Normalized}} = 0$$

Output

The final output is a set of normalized market positions:

$$\text{NormalizedPositions} = \{\text{MarketPosition}_u^{\text{Normalized}} \mid u = 1, 2, \ldots, m\}$$

where each normalized market position is represented as:

$$\text{MarketPosition}_u^{\text{Normalized}} = \begin{cases} (\text{Username}_u, \text{YesSharesOwned}_u^{\text{Normalized}}, 0) & \text{if YesSharesOwned}_u > \text{NoSharesOw} \\ (\text{Username}_u, 0, \text{NoSharesOwned}_u^{\text{Normalized}}) & \text{if NoSharesOwned}_u \geq \text{YesSharesOw} \end{cases}$$