1. Data Handling and Exploration:

1.1 After loading the dataset, the initial observations are,

- The dataset consists of trading transactions with 14 columns.
- The symbol column contains XAUUSD, indicating that the dataset is related to gold trading.
- The presence of negative pips and profit values suggest losing trades .
- The dataset includes key trading metrics like open_price, close_price, Stop loss, take profit, pips, volume and profit.

1.2 Initial Exploratory Data Analysis,

- There are no missing values in the dataset. Since, `print(df.isnull().sum())` shows output zero. No need to handle missing values.
- There are no duplicate rows in the dataset too.
- The 'type' column contained inconsistent capitalization for the values, 'Buy', 'Sell', 'sell', and 'buy'. The 'reason' column contained numerical values [4,0,3,16,2,1,17,5].
- The placeholder check returned zero, it means all values in close_time column and other columns are valid.
- The open_time and close_time are currently stored as objects(strings), but they should be converted to datetime for proper time based analysis. The column 'reason' is stored as int64 but it represents categorical data, it might be better as object or category. Because, Pandas stores categorical variables more efficiently than integers.
- There are **26,970 rows** with negative pips and **26,968 rows** with negative profit. It looks like a financial dataset so negative values are expected. For better understanding, outliers have been identified. Based on the boxplot and descriptive statistics it's clear that the pips and profit columns contain extreme negative and positive values. These extreme values could indicate outliers or errors in the dataset. From the descriptive statistics, it can be seen that the mean and median (50th percentile) for both pips and profit are relatively small compared to the min and max values, indicating that

extreme values are pulling the mean away from the median. The large standard deviation suggests data is highly spread out. To check whether the extreme values are valid or not, investigate further by checking the volume column to see if extreme values align with large trade sizes and examine the 'reason' column to see if extreme losses have valid reasons. After investigating, the extreme values appear to be valid. Because they align with larger trade sizes and valid reasons.

1.3 Data Cleaning and Preprocessing for analysis,

- No missing values,
- No duplicate rows,
- Correct data types,
- No Invalid values,
- Outliers handled

2. Profitability Analysis: Profitability analysis involves evaluating the financial performance of different traders (logins) to identify,

- Most Profitable Traders – Those consistently generating high profits.
- Least Profitable Traders – Those incurring frequent or significant losses.
- Cumulative Profit Trends – Analyzing how profits accumulate over time per login.
- Key Contributing Factors – Identifying patterns in trader behavior, market conditions, or trading strategies that lead to profitability.

Recommended Approach - Keep the extreme values.

Extreme values (both positive and negative) are critical for accurately calculating cumulative profits and identifying the most and least profitable logins. Capping or removing extreme values could distort the true profitability picture.For example, a login with a few extreme losses might still be profitable overall, and this should be reflected in the analysis.

2.1 **Most and Least Profitable Logins**: The goal of this analysis is to identify which logins contributed the highest and lowest profits to understand profitability trends.

Key Findings-

i. Most Profitable Logins:

Login ID	Cumulative Profit (in currency)
13378390	53,891.98
55009560	28,475.44
13088202	27,848.61
13205503	27,049.34
13070589	27,023.68
55008451	27,021.14
13205506	26,494.85
13361147	25,136.16
11173702	24,301.54
55010677	24,265.33

The highest profit was recorded by login 13378390, with a cumulative profit of 53,891.98, significantly higher than the rest.

ii. Least Profitable Logins:

Login ID	Cumulative Profit (in currency)
13103928	-14,778.82
13333728	-13868.00
55011482	-12,215.00
13018096	-12,194.31
13251499	-11,405.24
13410127	-10,571.86
55009211	-10,102.92
13276691	-10,010.77
13054222	-9,635.14
13131614	-9,573.61

The least profitable login is 13103928, with a total loss of -14,778.82, followed closely by login 13333728 with -13,868.00.

The most profitable login (13378390) has a significantly higher profit (53,891.98) compared to the second most profitable (28,475.44). This suggests that a few users contribute disproportionately to overall profitability.The losses among the least profitable logins range from -9,573.61 to -14,778.82, indicating a relatively consistent negative impact.

2.2 **Cumulative Profits per Login**: The dataset includes 600 unique logins, with cumulative profits ranging from -14,778.82 to +53,891.98, indicating significant variability in performance.

Key Findings-

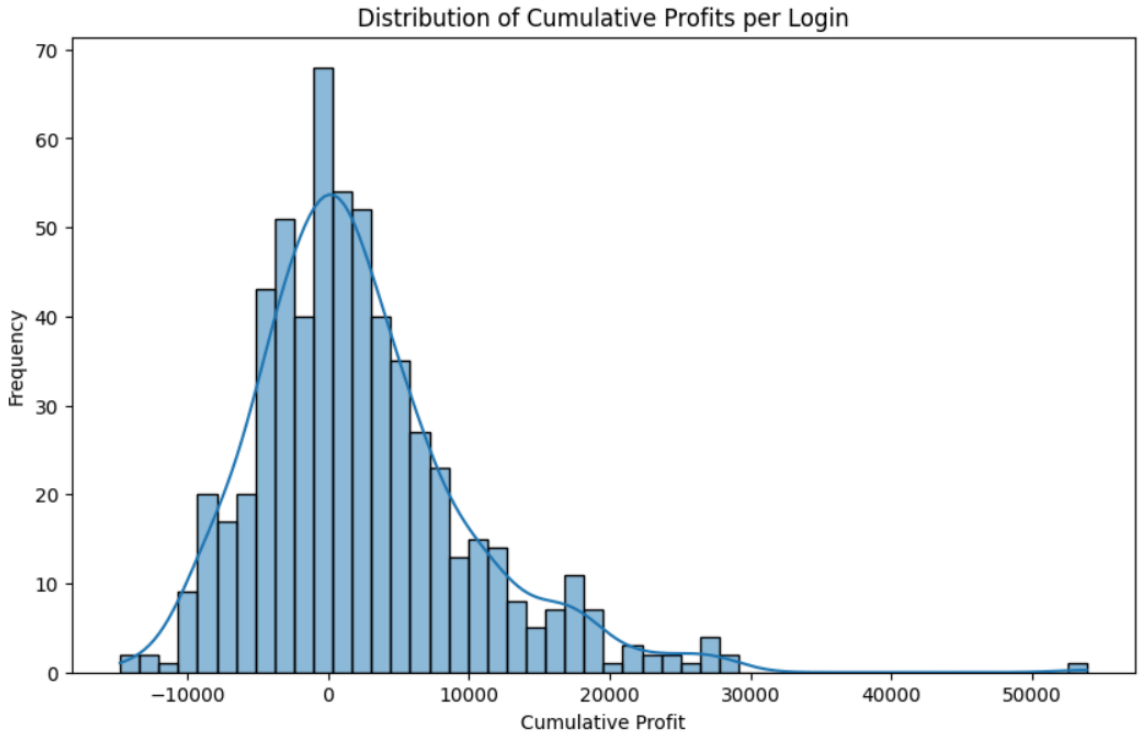
i. Distribution of Profits:

- 31 logins (top 5% of the ranked list) have cumulative losses exceeding -10,000.
- The worst-performing login (13103928) incurred a loss of -14,778.82.
- The majority of logins are profitable, with 25% of the top-ranked logins generating profits above +27,000.
- The highest-performing login (13378390) achieved a profit of +53,891.98, significantly outperforming others.

Top Performers vs Underperformers,

Metric	**Worst 5 login (Losses)**	**Best 5 Logins (Profits)**
Cumulative Profit Range	-14,778.82 to -11,405.24	+27,023.68 to +53,891.98

2.3 Profit Distribution Visualization:



2.4 **Exploring the Key Factors of Profitability**:

i. Correlation between Profit and other Numerical Features:

The intensity and direction of the link between profitability(profit) and other numerical characteristics in the dataset is shown by the correlation analysis. Despite the link being slightly weak, the output shows that the characteristic "pips" has the largest positive correlation with profit (0.137911), meaning that when the value of pips rises, profitability usually follows as well. Features including "volume," "ticket," and "take profit" show kind of poor positive correlations with profit, indicating no effect on profitability. On the other hand, features such as "stop loss," "open_price," "close_price," and "login" exhibit weak negative correlations.

ii. Factors Influencing Trading Profitability:

The workflow and output analysis provide a comprehensive understanding of the factors influencing profitability in the dataset. Since categorical columns cannot be correlated directly, it has to be encoded into numerical values. Pips (0.137911) is the most significant positive factor affecting profit. A higher number of pips tends to increase profitability. Volume (0.011749) is slightly positive but has a very weak correlation. This suggests that the trade size (lot size) does not strongly determine profitability. Ticket (0.009388), Open Time (0.008821), Take Profit (0.005967), Symbol (0.005432), Close Time (-0.004877), Login (-0.005339), Close Price (-0.008343), Open Price (-0.009115) have extremely low correlation with profit (close to 0.00).This means that they do not have a significant direct impact on profitability. A boxplot visualizing the profitability distribution by trade type (buy/sell) indicates that the median profitability for "buy" trades is higher than for "sell" trades, with "buy" trades also showing a wider interquartile range, suggesting greater variability in profitability. The second bar plot showing the total profit per trading symbol highlights that certain symbols, such as "XAUUSD" and "EURUSD", contribute significantly to overall profitability, while others like "USDJPY" and "EURCAD" have lower contributions. This analysis reveals that

"pips" and trade type are key factors affecting profitability, with certain symbols also playing a role.

3. Feature Engineering & Predictive Modeling: The goal is to engineer new features, preprocess the data, and build a predictive model to identify profitable traders. Analyzing extreme losses separately to ensure the model is robust and not skewed by outliers.

3.1 Engineer New Feature:

i. Profitability Flag: A binary flag indicating whether a trader is profitable (1) or not (0).

ii. Win Rate: The ratio of profitable trades to total trades.

iii. Average Profit per Trade: The average profit of all trades.

iv. Average Loss per Trade: The average loss of all trades.

v. Extreme Loss Flag: A binary flag indicating whether a trader has experienced extreme losses (e.g., losses below a certain threshold).

vi. Trade Frequency: The number of trades executed by a trader.

vii. Volume Weighted Profit: The total profit weighted by trade volume.

The final DataFrame now includes these additional features, providing a comprehensive set of performance indicators for each trader.

3.2 Preprocess Data:

i. Handling Missing Values: The first step in the preprocessing pipeline is to drop rows with missing target values (profitability_flag). This ensures that the dataset used for modeling is clean, with no missing values in the target column. Missing values in the feature columns are handled using different strategies for numerical and categorical features:

- Numerical Features: For numerical features (win_rate, avg_profit, avg_loss, trade_frequency, volume_weighted_profit), missing values are imputed using the mean of the respective feature. This is achieved using the SimpleImputer with the strategy set to 'mean'.
- Categorical Features: For categorical features (extreme_loss_flag), missing values are filled using the most frequent value of that feature. This is also done using SimpleImputer with the strategy set to 'most_frequent'.

After filling missing values, the DataFrame X is updated with fillna(0, inplace=True) to ensure that any remaining missing values are replaced by zero.

ii. Feature Selection: The features and target are defined as,

Features (X): These columns are used as inputs for modeling:

- win_rate: Represents the percentage of profitable trades.
- avg_profit: The average profit from profitable trades.
- avg_loss: The average loss from unprofitable trades.
- extreme_loss_flag: A binary flag indicating extreme losses.
- trade_frequency: The number of trades executed by each user.
- volume_weighted_profit: The profit weighted by trade volume.

iii. Preprocessing Pipeline: A pipeline for preprocessing is created using ColumnTransformer and Pipeline,

- Numerical Features:

  Missing values are imputed with the mean of the respective feature using SimpleImputer.

  Scaling: The numerical features are scaled using StandardScaler, which standardizes them by transforming the data to have a mean of 0 and a standard deviation of 1. This ensures that numerical features with different ranges (e.g., win_rate and avg_profit) are treated equally during model training.

- Categorical Features:

  Missing values are imputed with the most frequent value using SimpleImputer.

  One-Hot Encoding: The categorical feature extreme_loss_flag is encoded using OneHotEncoder. This transforms the binary flag into a one-hot encoded feature, which converts the categorical variable into separate binary columns (0 or 1), making it suitable for machine learning algorithms that cannot handle categorical variables directly.

The preprocessing pipeline is applied to the feature set (X), which processes both numerical and categorical features. This step transforms the data by applying the imputation, scaling, and encoding transformations as specified in the pipeline. After preprocessing, the transformed features are stored in X_preprocessed_df. The preprocessing pipeline successfully handles missing values, scales numerical features, and encodes categorical variables.

3.3 Build a Predictive Model:

A Logistic Regression model was trained to predict profitable traders based on engineered features and preprocessed data. The target variable (profitability_flag) indicates whether a trader is profitable (1) or not profitable (0). The model uses various features, including win_rate, avg_profit, avg_loss, extreme_loss_flag, trade_frequency, and volume_weighted_profit.

i. Model Evaluation: The model was evaluated using 5-fold cross-validation to ensure stability and robustness. The mean accuracy obtained across different validation splits is 65.6%, with a standard deviation of 1.8%, indicating some variability across different data partitions.

- Accuracy: 65.6%
- Classification Report: The detailed classification report includes the precision, recall, and F1-score for both classes (profitable and non-profitable traders).

ii. Metrics Breakdown:

Metric	Class 0 (Non-profitable Traders)	Class 1 (Profitable Traders)	Macro Average	Weighted Average
Accuracy	65.6%	65.6%	-	-
Precision	.6487	.6604	.65	.65
Recall	.5507	.7475	.64	.65
F1- Score	.59	.70	.64	.65

The logistic regression model successfully predicts profitable traders with good precision and recall, especially for profitable traders.

3.4 Validation of Model-Predicted Profitability Against Cumulative Profit Calculations:

1. Prediction of Profitability Flag: The model's predict method is applied to the preprocessed feature set (X_preprocessed_df) to predict the profitability flag for each trader. The predicted profitability flag (predicted_profitability_flag) is added as a new column to the DataFrame (df), where 1 indicates a profitable trader and 0 indicates a non-profitable trader.

- Login 11173702 has alternating predictions: 1 (profitable) for the first two rows, then 0 (non-profitable) for the next two, and again 1 for the last row.

2. Cumulative Profit Calculation: The cumulative profit for each trader is calculated using the cumsum() function, which computes the cumulative sum of the profit column grouped by login.The cumulative profit indicates the total profit for a trader up until each trade.

- Login 11173702 has progressively increasing cumulative profits (5578.40, 7498.40, 5100.40, etc.).

3. Validation of Predictions: The prediction_correct column is created to validate the correctness of each prediction,

If the model predicts profitable (1) and the cumulative profit is positive, the prediction is considered correct (1).

If the model predicts non-profitable (0) and the cumulative profit is non-positive, the prediction is also considered correct (1).

Otherwise, the prediction is considered incorrect (0).

4. Accuracy Calculation: The alignment_accuracy is calculated as the mean of the prediction_correct column, which gives the proportion of predictions that align with the actual cumulative profit. The percentage of correctly aligned predictions was calculated, resulting in 61.02% alignment accuracy.

The validation step was successfully accomplished by ensuring that the model's predicted profitable traders aligned with their cumulative profit calculations.

3.5 Analysis of Extreme Loss Traders:

The goal of this analysis is to identify and analyze traders with extreme losses, as flagged by the extreme_loss_flag feature. By separating these traders, one is able to comprehend trading behavior, profitability, and other important indicators to decide whether they call for particular management during the predictive modeling process.

The analysis of extreme loss traders revealed the following insights:

i. Profit Distribution:

- Out of 365 extreme loss traders, some traders achieved positive cumulative profits, while others incurred significant losses.

ii. Trade Frequency:

- Extreme loss traders exhibited a wide range of trade frequencies, from as low as 3 trades to as high as 309 trades.

iii. Win Rate:

- The win rate among extreme loss traders varied significantly, ranging from 27.83% to 60.71%.

The analysis of extreme loss traders highlights the importance of understanding the underlying behavior of this segment. The extreme_loss_flag is a helpful metric, but it falls short in capturing the complexities of trader profitability.

4.Clustering & Trader Segmentation:

The goal is to segment traders into meaningful groups (Losers, Profitable Traders, and Outliers) based on their trading behavior and performance metrics. The dataset used for this analysis contains trading data for multiple traders, with each record representing a single trade. The dataset includes various features that describe the trading behavior and performance of each trader. For this analysis, the primary feature used for clustering was cumulative profit, as it provides a clear and direct measure of a trader's overall performance. For clustering, the K-Means clustering algorithm has been used, which is a popular unsupervised machine learning technique for grouping data into clusters based on similarity.

Key Findings,

- Extracted the cumulative_profit feature for clustering.
- Handled missing values using SimpleImputer.
- Scaled the data using StandardScaler.
- Applied K-Means clustering with 3 clusters.
- Labeled the clusters based on their average cumulative profit.

4.1 Cluster Distribution:

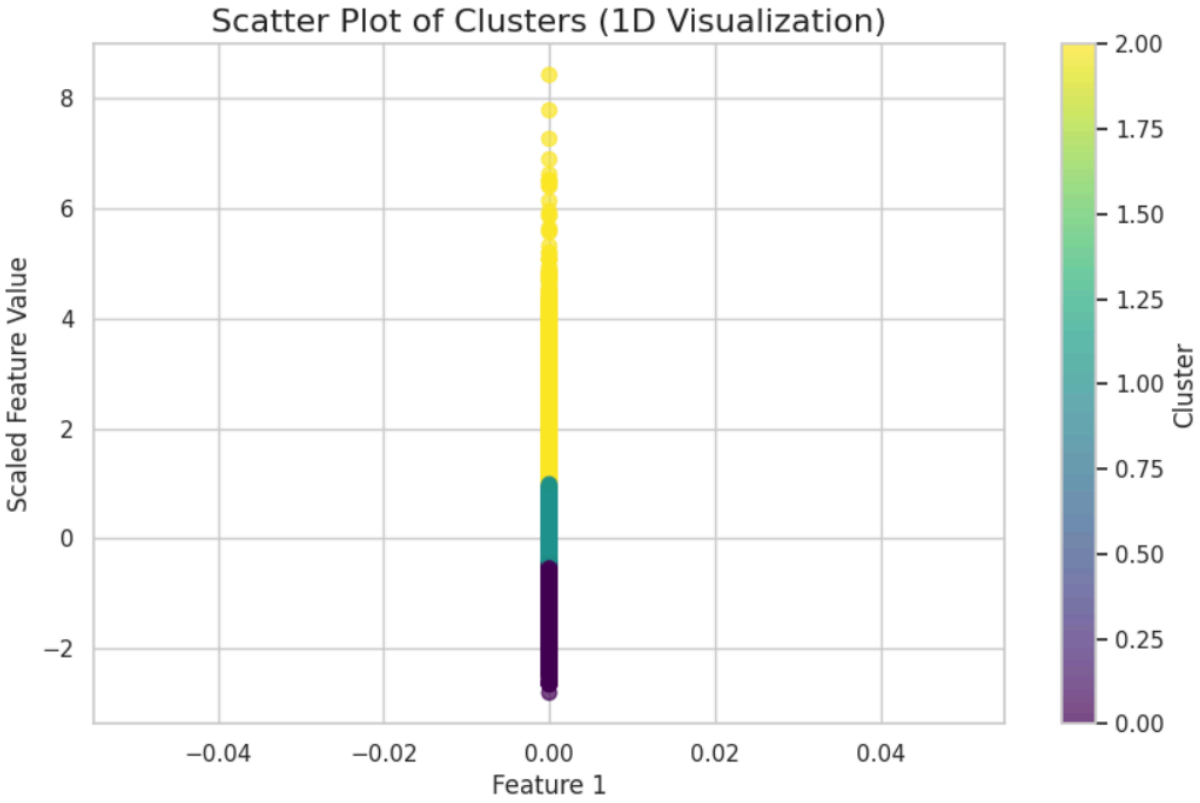
- Losers: Traders with negative cumulative profits.
- Neutral: Traders with mixed or near-zero cumulative profits.

- Profitable Traders: Traders with high positive cumulative profits.

4.2 Visual Representation:

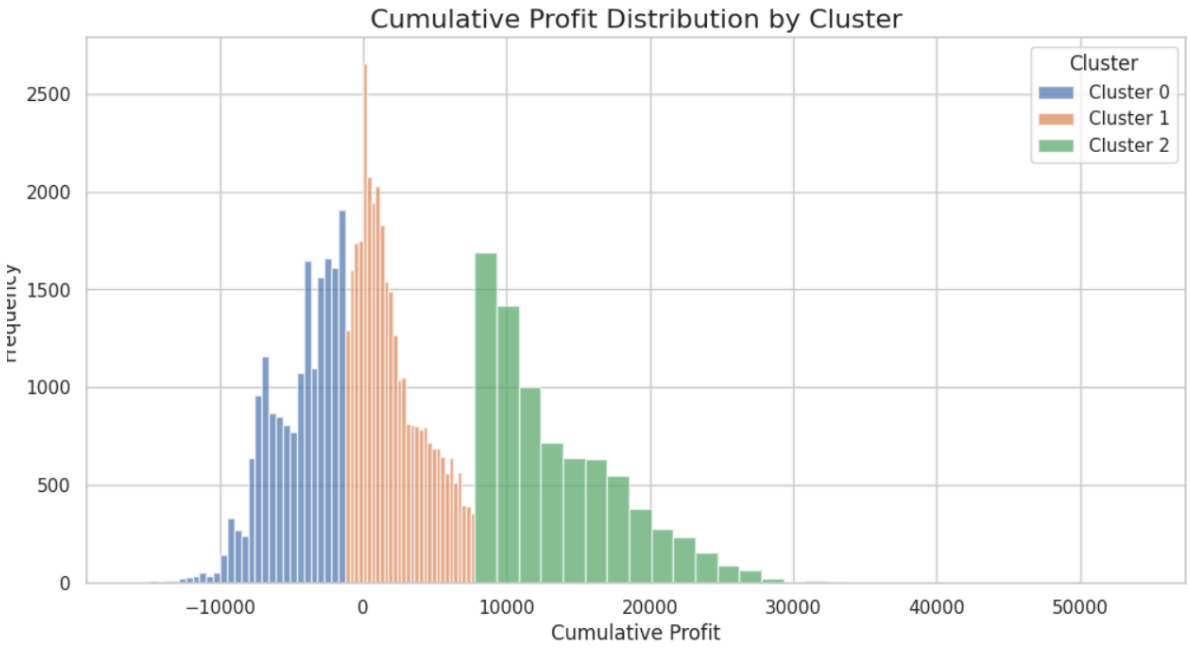
i. Scatter Plot: A scatter plot of cumulative profits, colored by cluster labels. The scatter plot provides a 1D visualization of the clustering results, where the x-axis represents Feature 1 (likely the scaled cumulative profit or another key feature used for clustering), and the y-axis represents the Cluster assignments (0, 1, or 2).

- Cluster 0 (likely Losers) is concentrated on the left side of the plot, suggesting lower values of Feature 1.
- Cluster 2 (likely Profitable Traders) is concentrated on the right side, indicating higher values of Feature 1.
- Cluster 1 (likely Neutral) is positioned in the middle, representing traders with near-zero or mixed performance.



The clear separation between clusters suggests that Feature 1 is a meaningful metric for distinguishing trader performance.

ii. Histogram: A histogram of cumulative profits, with clusters highlighted is shown. The histogram provides a visual representation of the distribution of cumulative profits across three identified clusters. Each cluster represents a group of traders categorized based on their cumulative profit. The objective of this clustering is to segment traders into distinct profitability groups, allowing for better understanding and potential strategy development.



Key Findings-

- Cluster 0 (Blue - Losses Group): This cluster consists of traders with negative cumulative profits.
- Cluster 1 (Orange - Neutral Group): This group consists of traders with cumulative profits near zero, meaning they are either barely profitable or incurring small losses.The distribution is relatively symmetric and concentrated around zero.
- Cluster 2 (Green - Profitable Traders Group): This cluster consists of traders with high positive cumulative profits.

This histogram provides a clear segmentation of trader profitability, helping to differentiate high-performing traders from those incurring losses.

iii. Heatmap: The correlation heatmap provides a visual representation of the relationships between the features used for clustering. Understanding these relationships is crucial for identifying which features are strongly correlated and which are independent, helping to refine the clustering process and interpret the results.

Key Findings,

1. Win Rate:

   Positive Correlation: Win rate shows a moderate positive correlation with trade_frequency (0.35), indicating that traders who trade more frequently tend to have a higher win rate.

   Negative Correlation: Win rate has a weak negative correlation with avg_profit (-0.33) and avg_loss (-0.03), suggesting that higher win rates are not strongly associated with higher average profits or losses.

2. Average Profit:

   Negative Correlation: avg_profit has a strong negative correlation with avg_loss (-0.67), indicating that traders with higher average profits tend to have lower average losses.

   Weak Correlation: avg_profit shows weak correlations with other features, such as extreme_loss_flag (0.24) and trade_frequency (-0.40).

3. Average Loss:

   Positive Correlation: avg_loss has a moderate positive correlation with trade_frequency (0.42), suggesting that traders with higher average losses tend to trade more frequently.

Negative Correlation: avg_loss has a strong negative correlation with avg_profit (-0.67), reinforcing the inverse relationship between average profit and average loss.

4. Extreme Loss Flag:

   Weak Correlations: extreme_loss_flag shows weak correlations with all other features, with the highest being a weak positive correlation with avg_profit (0.24).

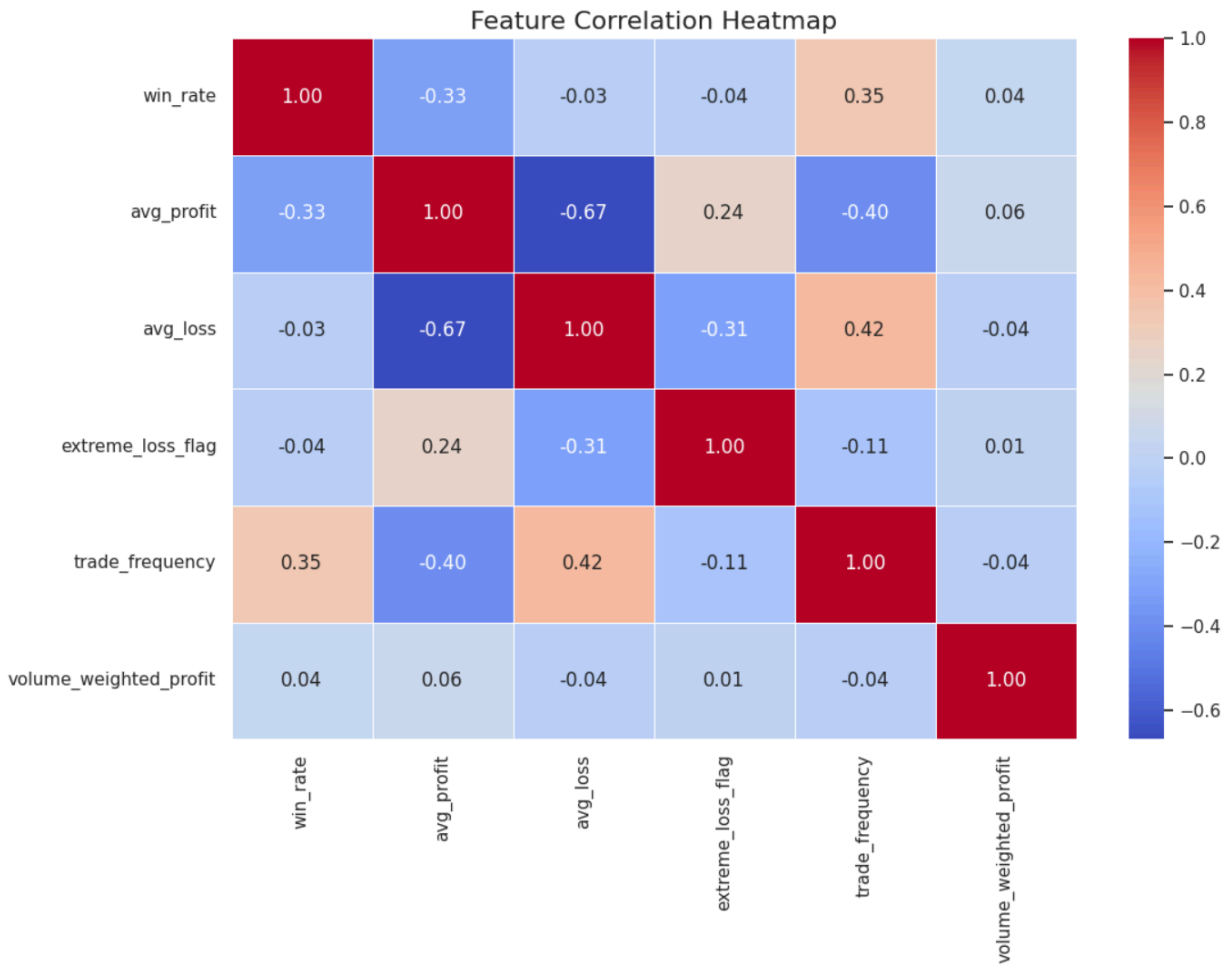
5. Trade Frequency:

   Positive Correlation: trade_frequency has a moderate positive correlation with win_rate (0.35) and avg_loss (0.42), indicating that more frequent trading is associated with higher win rates and higher average losses.

   Negative Correlation: trade_frequency has a weak negative correlation with avg_profit (-0.40).

6. Volume Weighted Profit:

   Weak Correlations: volume_weighted_profit shows very weak correlations with all other features, suggesting it is relatively independent of the other metrics.



The correlation heatmap reveals important relationships between the features used for clustering. The strong inverse correlation between avg_profit and avg_loss highlights the trade-off between profit and loss in trading.

Conclusion

This analysis of trading data provided valuable insights into trader profitability, key influencing factors, and trader segmentation. Through extensive data cleaning and exploratory analysis, the dataset's integrity and identified important patterns in trading behavior.

The profitability analysis revealed significant variability in trader performance, with a few traders contributing disproportionately to overall profitability. Key factors influencing profitability included the number of pips and trade type, while

trade volume showed little impact. Predictive modeling using logistic regression demonstrated a reasonable ability to classify profitable traders, achieving 65.6% accuracy.

Further segmentation using clustering techniques categorized traders into three groups—Profitable Traders, Neutral Traders, and Losers—providing a clear distinction in trading behavior and performance. The correlation heatmap highlighted meaningful relationships between trade frequency, win rate, profit, and losses, reinforcing the trade-off dynamics in trading strategies.

Future work could involve refining predictive models, incorporating additional features, and exploring advanced machine learning techniques for improved forecasting accuracy.