Trading Performance & Market Sentiment Analysis

Bitcoin Market Sentiment (Fear vs Greed) vs Hyperliquid Trader Performance

Prepared for: Vinay Rathore

Date: August 8, 2025

This report summarizes methodology, key findings, visualizations to include, and recommended next steps.

Executive Summary

Executive Summary

This report examines the relationship between Bitcoin market sentiment (Fear vs Greed) and trader performance (closed PnL) using Hyperliquid historical trade data.

Key deliverables you should include in your assignment:

1. Data cleaning & alignment of trade timestamps to IST and matching each trade to daily sentiment labels.

2. Descriptive statistics for per-trade PnL and aggregated daily PnL by sentiment.

3. Visualizations: distribution plots, boxplots, time-series of daily PnL, correlation heatmap, and feature importance from a Random Forest model.

4. Statistical testing to check whether PnL differs significantly between 'Fear' and 'Greed' days (e.g., Welch's t-test or Mann–Whitney U test when assumptions fail).

High-level conclusion (placeholder):

- Replace the placeholders below with results from your analysis. Example statements you may find:

  • "Average Closed PnL on Greed days was higher than on Fear days (mean Greed = X, mean Fear = Y)."

  • "Win rate during Greed days = A% vs Fear days = B%."

  • "Statistical test (Welch's t-test) resulted in p = PVAL, indicating [a significant / no significant] difference."

Data & Methodology

Data & Methodology

Datasets used:

- Bitcoin Market Sentiment Dataset: Daily records labeled with 'Fear' or 'Greed' (or numeric index).

- Hyperliquid Historical Trades: Per-trade records including account, symbol, execution price, size, side, timestamp, closed PnL, leverage, and related fields.

Preprocessing steps (done in your notebook):

1. Load both CSV files and parse timestamps. Convert all trade times to 'Asia/Kolkata' timezone (IST).

2. Extract trade_date from the IST-converted timestamps and merge trades to sentiment by date.

3. Standardize column names (Closed PnL to numeric, leverage to numeric, size to numeric).

4. Create derived fields: 'win' (Closed PnL > 0), 'absPnL', and daily aggregates (total_pnl, trades_count).

Analysis steps performed:

- Descriptive statistics by sentiment and by coin.

- Visual analysis (histograms, boxplots, time-series).

- Statistical testing (compare distributions between Fear and Greed).

- A Random Forest classifier to estimate feature importance for win prediction.

Key Findings

Key Findings (fill with your computed values)

1) Dataset overview:

- Total trades analyzed: [TOTAL_TRADES]

- Date range: [START_DATE] to [END_DATE]

2) Summary statistics by sentiment: (replace placeholders)

- Mean Closed PnL on Greed days: [MEAN_GREED]

- Mean Closed PnL on Fear days: [MEAN_FEAR]

- Win rate on Greed days: [WINRATE_GREED]

- Win rate on Fear days: [WINRATE_FEAR]

3) Statistical testing:

- Test used: Welch's t-test (or Mann–Whitney U if non-normal)

- t-statistic / U-statistic: [TEST_STAT]

- p-value: [P_VALUE]

- Interpretation: [CONCLUDE_SIGNIFICANCE]

4) Feature importance (from Random Forest classifier predicting 'win'):

- Top features (example): [FEATURE_1], [FEATURE_2], [FEATURE_3] with importances [IMP1, IMP2, IMP3].

Visualizations & Captions

Visualizations to Include (and captions)

1) Distribution of Closed PnL by Sentiment (Histogram + KDE).

Caption: 'Shows distribution and tail behaviour of trade profits/losses under different market sentiment regimes.'

2) Boxplot of Closed PnL by Sentiment.

Caption: 'Highlights medians, IQR, and outliers; useful to compare variability.'

3) Time-series of Daily Total PnL with Sentiment Overlay (shade or color by daily sentiment).

Caption: 'Displays how aggregate performance evolves and whether positive/negative spells align with sentiment.'

4) Win Rate by Sentiment (bar chart).

Caption: 'Quick metric to compare success rates under different market moods.'

5) Correlation Heatmap of numeric features.

Caption: 'Shows relationships between variables like size, leverage, and PnL.'

6) Feature Importance from Random Forest.

Caption: 'Identifies which trade features are most predictive of winning trades.'

Recommended Actions & Next Steps

Recommended Actions & Next Steps

1) Risk management tuning: If results show greater downside on Fear days, implement stricter stops and lower leverage on those days.

2) Strategy specialization: If certain coins perform better during Greed or Fear, design separate allocation strategies per sentiment.

3) Expand features: Add on-chain or other technical indicators (volatility, volume spikes, order book imbalance) and re-evaluate models.

4) Model experiments: Try regression models for predicting PnL magnitude and classification models for win probability. Use cross-validation and time-series aware splits.

5) Live testing: Backtest and then paper-trade a sentiment-aware allocation strategy before deploying real capital.

6) Reporting: Include a dashboard with the following panels: daily PnL, recent trades list, sentiment timeline, and alerts for threshold breaches.

Appendix: Code Snippets

Appendix: Key Code Snippets

Merge trades to daily sentiment (example):

```
trades['Timestamp IST'] = pd.to_datetime(trades['Timestamp IST'], errors='coerce')

trades['Timestamp IST'] = trades['Timestamp IST'].dt.tz_localize('UTC').dt.tz_convert('Asia/Kolkata')  # if naive

trades['Date'] = trades['Timestamp IST'].dt.date

merged = pd.merge(trades, fear_greed[['Date','classification']], on='Date', how='inner')
```

Compute per-sentiment aggregates:

```
agg = merged.groupby('classification').agg(trades_count=('Closed PnL','count'), total_pnl=('Closed PnL','sum'),
                                           avg_pnl=('Closed PnL','mean'), win_rate=('Closed PnL', lambda x: (x>0).mean()))
```

Welch's t-test (Fear vs Greed):

```
from scipy import stats

fear = merged[merged['classification'].str.lower()=='fear']['Closed PnL'].dropna()

greed = merged[merged['classification'].str.lower()=='greed']['Closed PnL'].dropna()

t_stat, p_val = stats.ttest_ind(fear, greed, equal_var=False)
```

Save figures to PDF (example using PdfPages):

Closing Notes

Closing Notes

This report was generated as a template containing structured sections and clear placeholders. Run the provided notebook on your actual datasets to fill numeric values and export figures.

If you'd like, I can:

 - Automatically populate this report with results and charts if you provide the merged dataframe (merged_df) here,

 - Or modify the report to include charts embedded directly from your dataset.

Tell me how you'd like to proceed — I can generate the fully-populated PDF next.