

Junior Data Scientist – Trader Behavior Insights

My Assignment

Step-by-Step Explanation of the Analysis

The assignment was completed through a seven-step data science pipeline:

1. Data Loading and Cleaning

The first step imports necessary libraries (pandas, numpy, seaborn, sklearn, etc.) and loads the two datasets: historical_data.csv (trades) and fear_greed_index.csv (sentiment).

- **Normalization:** Column names in both datasets are standardized (lowercase, underscores).
- **Type Conversion:** The trade Timestamp (in milliseconds) and the sentiment date column are converted into proper datetime objects for accurate merging and time-series analysis.
- **Data Quality:** Key trade metrics (execution_price, closed_pnl, etc.) are converted to numeric, and a check confirms that both datasets are complete, with no missing values after initial cleaning.

2. Feature Engineering: Daily Performance Aggregation

Since the sentiment data is daily, the granular trade data is aggregated to a daily level to create comparable metrics.

- A new DataFrame (daily_trades) is created with key performance indicators for all traders combined for each day:
 - **daily_pnl:** Total Profit and Loss for all trades on that day.
 - **trade_count:** Total number of trades executed.
 - **win_rate:** The ratio of profitable trades to total trades.

3. Data Merging and Sentiment Scoring

The daily performance data is merged with the sentiment data using the trading date. The categorical classification (e.g., 'Fear', 'Greed') is mapped to an ordinal **sentiment_score** (0-100) to allow for quantitative analysis like correlation and linear modeling.

- *Mapping Example:* Extreme Fear maps to 0, Fear to 25, Neutral to 50, Greed to 75, and Extreme Greed to 100.

4. Exploratory Data Analysis (EDA) and T-Test

Initial insights are generated by analyzing performance across the different sentiment categories.

- **Average Performance by Sentiment:** The analysis reveals that the average daily PnL and average Win Rate are **highest during 'Fear' market conditions**.
 - Average Daily PnL during **Fear**: \$8458.56.
 - Average Daily PnL during **Greed**: \$4416.04.

- **T-Test:** An independent t-test is performed to compare the daily PnL between Extreme Fear and Extreme Greed conditions. The results suggest the difference is **not statistically significant** at the 5% level (P-value of 0.093).

5. Statistical Analysis: Correlation

A correlation matrix is calculated and visualized to quantify the linear relationship between the numeric sentiment score and the performance metrics.

- **Key Finding:** A negative correlation is observed between sentiment_score and both **daily_pnl (-0.15)** and **win_rate (-0.17)**. This is a strong indicator of a **contrarian performance** pattern, where lower sentiment (higher fear) is associated with better trading results.
- **Time-Series Visualization:** A 30-day rolling correlation plot is generated, showing how the relationship between market PnL and sentiment changes over time.

6. Trader Segmentation (K-Means Clustering)

To uncover hidden trading patterns, the individual trader accounts are profiled and segmented using **K-Means clustering** (with k=3) based on their average PnL, average win rate, and average trade count.

- **Cluster 1: Best Performance** (Highest Avg PnL: \$800k, Highest Avg Win Rate: 88.6%).
- **Cluster 0: High Volume, Moderate PnL** (Highest Avg Trade Count: 12k).
- **Cluster 2: Low Performance** (Lowest Avg PnL: \$64k, Lowest Avg Win Rate: 47%).

7. Predictive Modeling

A **Logistic Regression** model is built to predict the probability of the **next day being profitable** (next_profitable) using current-day sentiment and trade count as predictors.

- The model uses **Time Series Cross-Validation** (TimeSeriesSplit) to ensure the training data chronologically precedes the testing data, which is essential for any prediction involving time.
- The model achieved a cross-validation accuracy of **77%**, demonstrating that sentiment and trade volume offer a reasonable baseline for predicting short-term profitability.