Cryptocurrency Trading Performance Analysis

Machine Learning-Driven Trading Strategy

Workflow: I split the project into four clear phases across two notebooks for easy understanding.

Jupyter Notebook 1:

Phase 1:

1. Cleaned Fear/Greed Data(fear_greed_data.csv)
2. Cleaned Trading Data(historical_data.csv)
3. Feature Engineering(net_pnl, pnl_per_dollar, is_profitable, direction)
4. Merged datasets on date feature(trade_date = date)

Phase 2: EDA

1. Explored the distribution of the merged data.
2. Seen the trading behaviour patterns.
3. Visualized using the histograms, boxplots.
4. Saved the cleaned and merged dataset .

Jupyter Notebook 2: Firstly**,** I Loaded the merged dataset.

Phase 3:

1. Hypothesis Testing: Here I have 3 tests using Mann-Whitney-test because the is not normalized
   I. Comapared average pnl_per_dollar in Fear vs Greed
   II. Compared pnl_per_dollar for pro_sentiment vs contrarian
   III. Compared is_profitable across groups using the chi-square test to compare the observed frequency distributions
2. I have conducted a level-based analysis like side level, Account level, Coin Level analysis
3. In the I examined whether traders go long (buy before selling) or short (sell before buying). In the account level I examined that weather the traders are consistently better at reading sentiment(fear/greed) and in the coin level I have examined weather the coins react different to sentiments.
4. The modelling is also done here where I took the variables and One-Hot-encoded the categorical variables and scaled the USD size variable(numerical) and took the target variable into y(is_profitable) which is taken from the net_pnl feature in notebook1.
5. Here I used Random Forest(Bagging) and XGBoost(Gradient Boosting) for the classification of the target variable(is_profitable).
6. Evaluated the models according to my predicted and actual labels.

Phase 4:

1. Took the key performance metrics using the Best performing coins using the feature importance and took the coin-specific performance for each coin. Likewise I got low and high win rate of coins
2. Printed the reports of every insight I have got in the recommendations.

Why My model Works

Random Trading Results:

- 50% accuracy
- 18,046 wrong trades
- $45,116 in losses

Model Results:

- 75% accuracy (real intelligence)
- 9,023 wrong trades
- $22,558 in losses

Key results:

- $22,558 saved in prevented losses on $1.8M trading volume
- 1.25% better returns compared to random trading
- Clear strategy for which coins to trade and which to avoid

High Win Rate features(Coins):

- BERA: 57.1% win, 82.5% accuracy
- JELLY: 84.7% win, 85.3% accuracy
- PURR: 72.1% win, 78.3% accuracy

Trader Performance Patterns:

- Trade Distribution: Primarily small trades ($0-100 range)
- Challenging Areas: Traditional cryptos (BTC 33.3%, ETH 37.1% win rates)
- Optimal Trade Size: Under $100 for best risk-adjusted returns

Sentiment-Performance Relationship:

- High Social Buzz = High Accuracy: FARTCOIN (83.7% accuracy), MELANIA (93.4% accuracy)
- Contrarian Opportunity: High accuracy coins with low win rates (FTT, MELANIA) offer value

Smart Trading Strategy

- 50% in top performers (JELLY, PURR, BERA, ONDO)
- 30% in moderate coins (FARTCOIN, PENDLE, WLD)
- 10% in high-accuracy contrarian plays (MELANIA, FTT)

In Summary my ML model to predict crypto trades with 75% accuracy, saving $22,558 on $1.8 M and revealing meme coins like JELLY and PURR as top picks while avoiding BTC and ETH.