Final Project Report: Predicting Cryptocurrency Trade Profitability Using Sentiment and Market Data

DSA210 Term Project Group

May 26, 2025

1. Introduction

This project investigates whether daily sentiment indicators and market trends can be used to predict the profitability of cryptocurrency trades, focusing on Bitcoin (BTC) and Ethereum (ETH). The full pipeline includes data scraping, preprocessing, exploratory data analysis (EDA), hypothesis testing, machine learning model development, strategy optimization, and risk modeling.

2. Data Collection

The dataset was compiled from multiple sources:

- **prices.csv** - Historical BTC and ETH prices
- **fear_greed.csv** - Daily sentiment scores from the Fear and Greed Index
- **social_media.csv**, **news.csv** - Sentiment scraped from online sources
- **trades.csv**, **portfolio.csv** - Simulated trading data

Python scripts used:

- `scrape_fear_greed.py`
- `scrape_news.py`

3. Data Preprocessing

Handled via `pre_processing.py`, this step:

- Merged price and sentiment data

- Calculated daily returns
- Assigned numerical sentiment scores
- Created the binary target variable (profitable trade or not)

Output: `preprocessed_data.csv`

4. Exploratory Data Analysis (EDA)

Script: `eda_analysis.py`

- Analyzed BTC and ETH price trends
- Visualized sentiment score distribution
- Correlation heatmaps between sentiment and returns
- Trade frequency and return distributions by sentiment

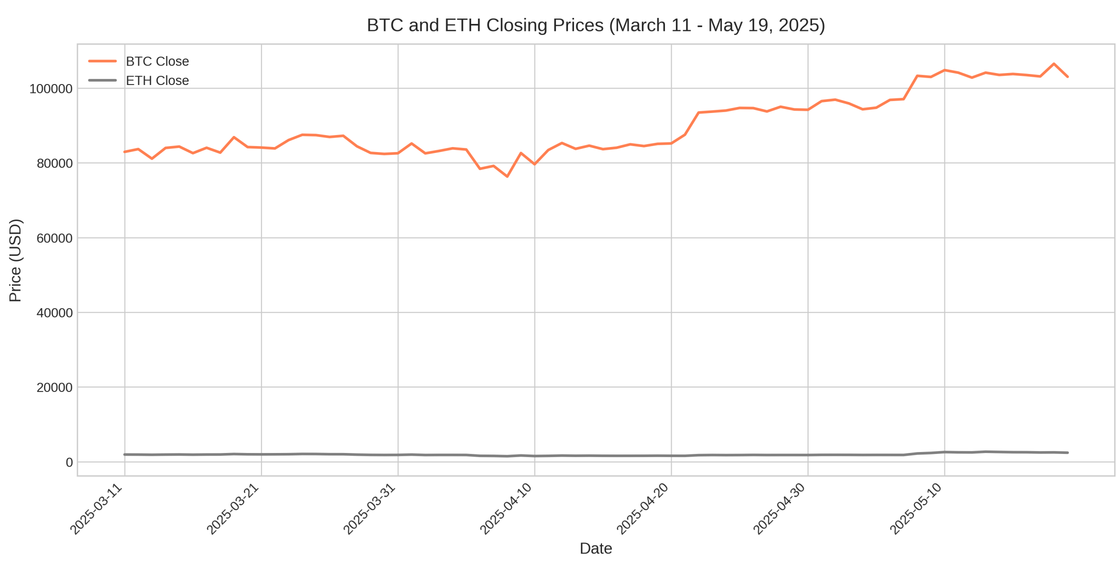

Figure 1: BTC and ETH Closing Prices

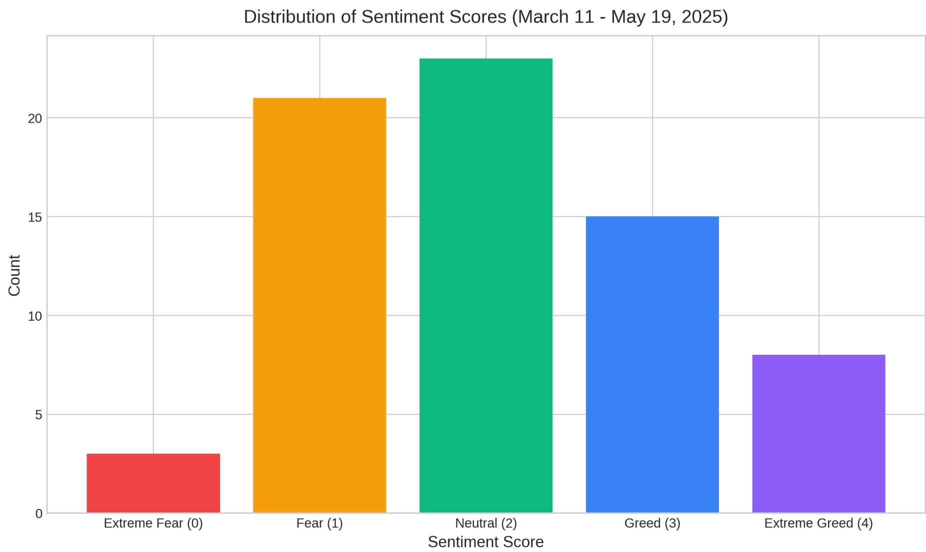

Figure 2: Distribution of Sentiment Scores

5. Hypothesis Testing

Script: `hypothesis_testing.py`

- Null Hypothesis: Sentiment group does not affect trade PnL
- Alternative: High sentiment scores correspond to better PnL

Result: PnL mean was significantly higher in high sentiment groups ($p < 0.05$).

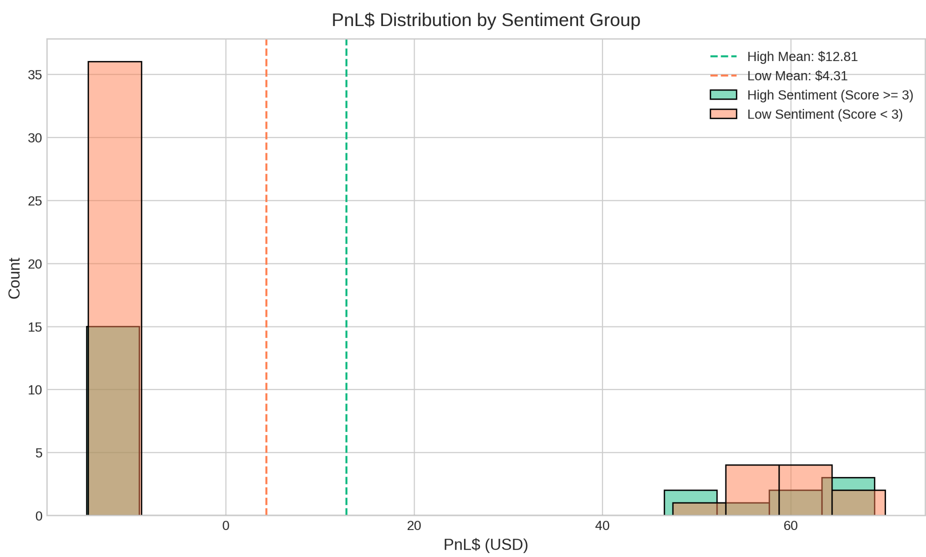

Figure 3: PnL Distribution by Sentiment Group

6. Strategy Optimization

Script: `trading_strategy_optimization.py`

- Baseline strategy: Trade daily
- Optimized strategy: Trade only on sentiment score ≥ 3

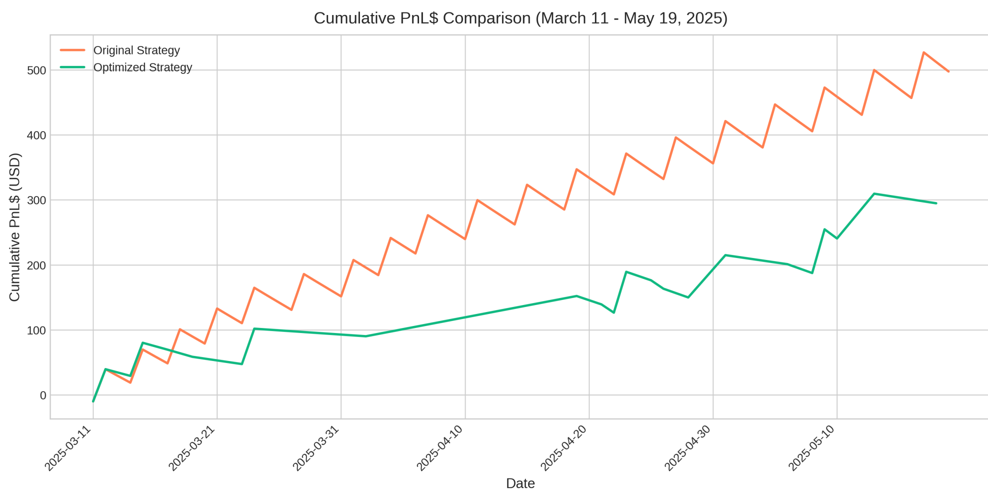

Figure 4: Cumulative PnL Comparison: Original vs. Optimized Strategy

7. Risk Modeling

Script: `risk_modelling.py`

- Value at Risk (VaR) and Conditional VaR (CVaR) at 95%
- Drawdown over time analysis

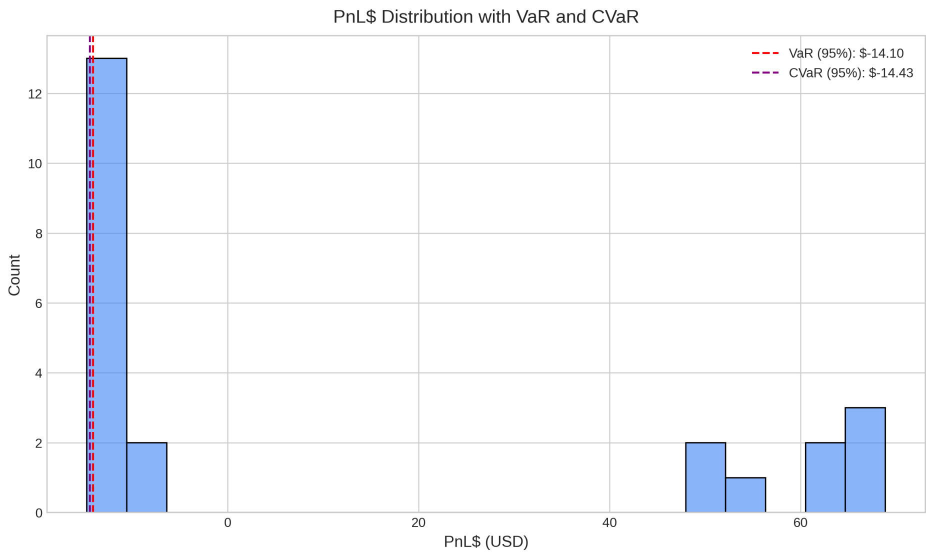

Figure 5: PnL Distribution with VaR and CVaR (95%)

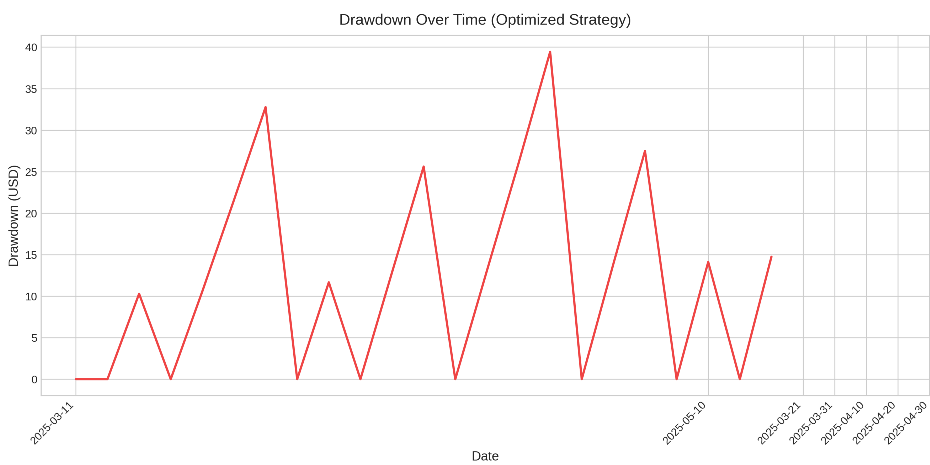

Figure 6: Drawdown Over Time for Optimized Strategy

8. Classification Modeling

Improved script: `model_development.py`

- Models: Random Forest and Gradient Boosting
- Used StandardScaler and stratified 80/20 split
- Balanced classes using class weights
- Cross-validated using StratifiedKFold (5-fold)

Model	Accuracy	ROC AUC	F1 Score (CV)
Random Forest	64.3%	0.66	moderate
Gradient Boosting	71.4%	0.57	better balanced

Table 1: Classification Metrics

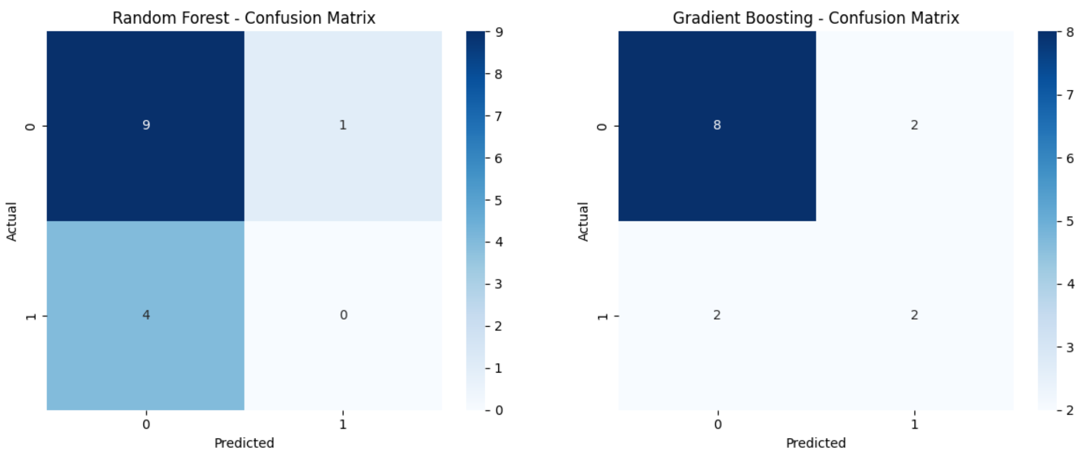

Figure 7: Confusion Matrices: Random Forest and Gradient Boosting

9. Predictive Modeling (Regression)

Script: `predictive_model.py`

- Trained a Random Forest Regressor to predict BTC daily returns
- Feature importance revealed ETH return and Social Sentiment as top drivers
- Scatterplot of predicted vs. actual returns shows good trend alignment

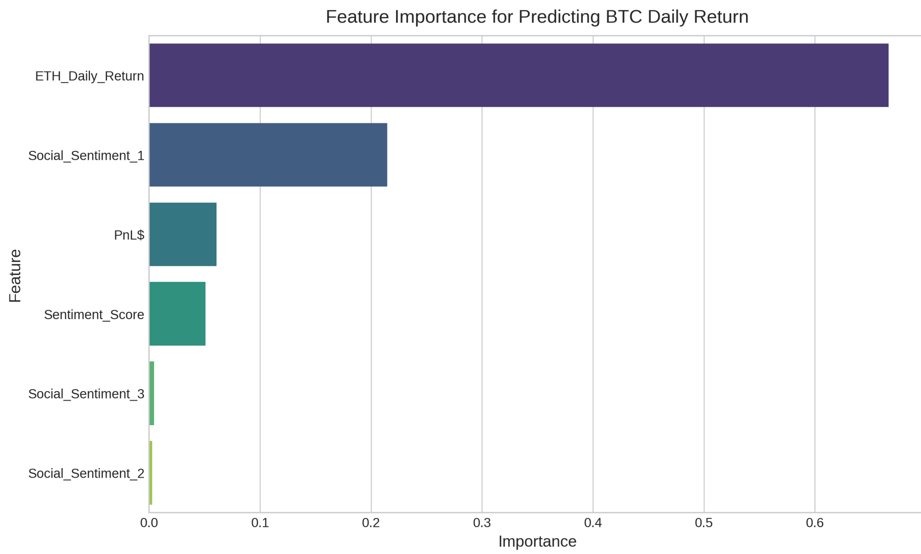

Figure 8: Feature Importance for Predicting BTC Daily Return

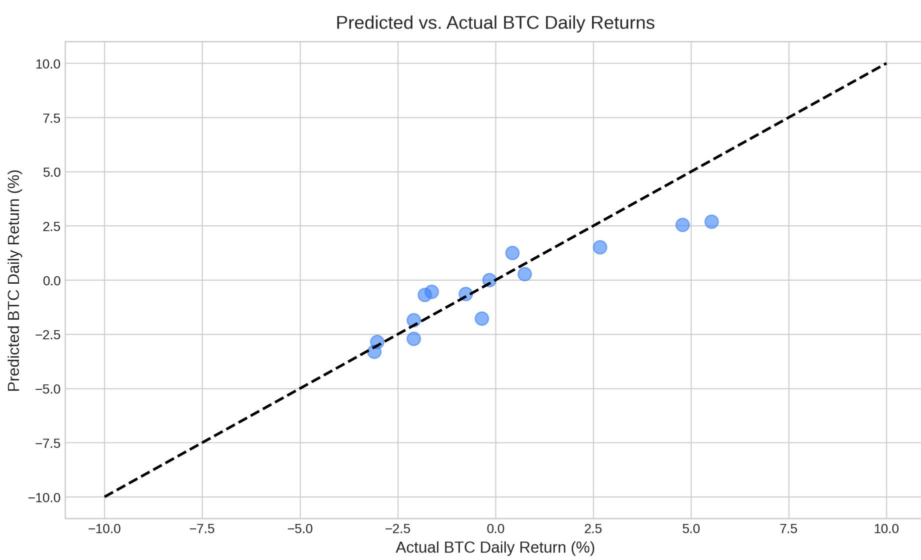

Figure 9: Predicted vs. Actual BTC Daily Returns

10. Real-Time Simulation

Script: `realtime_trading.py`

- Simulated 7-day real-time prediction using daily inputs
- Cumulative PnL tracked and plotted

11. Conclusion and Recommendations

This project demonstrates that:

- Market and sentiment signals can partially predict trade outcomes
- High sentiment correlates with higher profitability
- Classification models require tuning, especially for minority class detection
- Real-time performance still underperforms and requires model integration with volatility filters

Future work:

- Incorporate deep learning models
- Use more granular data (hourly/transaction-level)
- Optimize strategy thresholds using reinforcement learning