The Cloudy&Shiny Index: A Global Market Sentiment Analysis System

Project in Financial Data Analysis

arda@ardakgul.com

Arda Akgül

June 8, 2025

Abstract

Project Overview: This paper presents the Cloudy&Shiny Index, a student project that creates a global market sentiment indicator by analyzing international financial markets. The system integrates market data from multiple countries to provide a unified sentiment measurement.

Implementation: The project combines technical analysis with sentiment indicators from major markets including the United States, China, Japan, Hong Kong, Germany, France, and Turkey. A weighted scoring system processes 13 financial instruments to generate a sentiment score from 0-100.

Methodology: The system uses a distance-based scoring algorithm that measures how far current prices deviate from moving averages, combined with external sentiment indicators like the Fear & Greed Index and news sentiment analysis.

Results: The implementation successfully processes real-time market data and generates sentiment classifications ranging from "Extreme Cloudy" to "Extreme Shiny" with approximately 11-13 active components depending on market availability.

Educational Value: This project demonstrates practical application of financial data analysis, international market correlation studies, and real-time data processing techniques learned during university coursework.

Conclusion: The Cloudy&Shiny Index serves as an educational tool for understanding global market dynamics and provides hands-on experience with financial programming and data analysis.

Keywords: Student project, market sentiment, international finance, technical analysis, Python programming

Contents

1 Introduction

1.1 Project Background and Motivation

As a student studying the intersection of computer science and economics, I was motivated to explore how global financial markets influence each other and whether it's possible to create a unified measure of international market sentiment. Traditional market analysis tools often focus on individual markets or regions, but in our interconnected global economy, understanding cross-market relationships has become increasingly important.

This project, the **Cloudy&Shiny Index**, represents my attempt to create a comprehensive system that analyzes multiple international markets simultaneously and produces a single sentiment indicator that reflects global market conditions.

1.2 Learning Objectives and Scope

The primary learning objectives of this project include:

- Understanding how to work with real-time financial data APIs
- Learning to implement technical analysis algorithms in Python
- Exploring correlations between international financial markets
- Practicing data processing and sentiment analysis techniques
- Gaining experience with web development for data visualization

The project scope covers major markets from seven countries/regions: United States, China, Japan, Hong Kong, Germany, France, and Turkey, providing exposure to both developed and emerging market dynamics.

2 System Design and Implementation

2.1 Technical Architecture

The Cloudy&Shiny Index system is built using Python and follows a modular architecture that separates data collection, processing, and presentation layers. The main components include:

- **Data Collection Module:** Fetches real-time market data using the yfinance library
- **Calculation Engine:** Processes technical indicators and computes sentiment scores
- **Sentiment Analyzer:** Integrates external sentiment sources
- **Web Interface:** Displays results in a user-friendly format

2.2 Core Algorithm Implementation

The main calculation follows this approach:

$$\text{CS\_Index} = \sum_{i=1}^{n}(W_i \times S_i) \times 0.9 + \frac{(FGI + NSA)}{2} \times 0.1 \tag{1}$$

Where:

$$W_i = \text{Dynamic weight of component } i \tag{2}$$
$$S_i = \text{Distance-based score of component } i \tag{3}$$
$$FGI = \text{Fear \& Greed Index} \tag{4}$$
$$NSA = \text{News Sentiment Analysis} \tag{5}$$

Distance-Based Scoring Method

The core innovation of this project is the distance-based scoring algorithm that converts price movements into sentiment scores:

$$S_i = 50 + \left(\frac{\min(\max(\frac{P_i - MA_{30,i}}{MA_{30,i} \times 0.20}, -1), 1) \times 50}{1} \right) \tag{6}$$

Where:

$$P_i = \text{Current price of component } i \tag{7}$$
$$MA_{30,i} = \text{30-day moving average of component } i \tag{8}$$
$$0.20 = \text{Maximum deviation threshold (20\%)} \tag{9}$$

This algorithm converts price deviations from moving averages into standardized sentiment scores, making it possible to compare sentiment across different markets and instruments.

3 Market Coverage and Weight Distribution

3.1 Component Selection Rationale

The selection of market components was based on several criteria: - Global economic significance - Market liquidity and data availability - Regional representation - Different asset classes (equities, bonds, commodities, currencies)

The weight distribution attempts to balance US market influence with international diversification:

Table 1: Market Weight Distribution

Market Category	**Weight (%)**	**Rationale**
US Markets	40%	Largest global economy
International Markets	35%	Diversification across regions
Risk & Volatility	15%	Market stability indicators
Safe Havens & Commodities	10%	Crisis response measures

3.2 International Market Components

The system integrates major global markets with assigned weightings based on market capitalization and regional representation:

Table 2: International Market Components

Market	Region	Weight (%)
Shanghai Composite	China	8%
Nikkei 225	Japan	7%
Hang Seng	Hong Kong	6%
DAX	Germany	5%
BIST 100	Türkiye	5%
CAC 40	France	4%

4 Sentiment Classification System

4.1 Five-Tier Sentiment Scale

The Cloudy&Shiny Index uses a five-tier sentiment classification system:

Table 3: Sentiment Classification Scale

Range	Classification	Symbol	Economic Interpretation
75-100	Extreme Shiny	★★★	Strong global optimism with positive market conditions
51-74	Shiny	★★	Bullish sentiment with favorable market outlook
50	Neutral	○	Balanced market conditions
25-49	Cloudy	●●	Bearish sentiment with market concerns
0-24	Extreme Cloudy	■■■	Strong pessimism with significant market stress

5 Implementation and Algorithm

5.1 Core Algorithm Implementation

The system is implemented have possibility to use Python algorithms:

Algorithm 1 Cloudy&Shiny Index Calculation

```
Initialize global market components
Fetch real-time market data for all components
for each component i in components do
    Calculate technical indicators (MA5, MA20, MA30, RSI)
    Compute distance-based score S_i
    Apply inverse logic if component is risk indicator
    Calculate weighted contribution W_i × S_i
end for
Fetch Fear & Greed Index (FGI)
Analyze news sentiment (NSA)
Calculate final index using Equation (1)
Classify sentiment using five-tier scale
Return comprehensive sentiment analysis
```

5.2 Distance-Based Scoring Implementation

Listing 1: Distance-Based Scoring Implementation

```
def distance_based_score(current, ma, max_deviation=0.20):
    """
    Calculate sentiment score based on price distance from moving
        average
    """
    if ma == 0:
        return 50.0, 0.0

    # Calculate percentage difference from moving average
    diff_ratio = (current - ma) / ma

    # Normalize to maximum deviation threshold
    normalized_diff = diff_ratio / max_deviation
    normalized_diff = max(-1, min(1, normalized_diff))

    # Transform to 0-100 sentiment score
    score = 50 + (normalized_diff * 50)

    return float(score), float(diff_ratio)
```

6 System Implementation Status

6.1 Currently Implemented Features

The following features have been successfully implemented and tested:

- **Core Formula:** CS_Index = $(\text{Wi} \times \text{Si}) \times 0.9 + (\text{FGI} + \text{NSA})/2 \times 0.1$
- **International Markets:** 7 regional markets with 13 total components
- **Technical Analysis:** Moving averages, RSI, and distance-based scoring
- **External Sentiment:** Fear & Greed Index and basic news sentiment
- **Real-time Processing:** Live data fetching and calculation
- **Web Interface:** HTML dashboard for result visualization
- **Data Storage:** JSON/CSV export functionality
- **Basic Cryptocurrency:** Bitcoin and Ethereum price tracking

6.2 Features Not Yet Implemented

The following features are mentioned in research literature but not currently implemented:

- **Machine Learning Models:** No ML libraries or prediction algorithms
- **Advanced NLP:** News sentiment uses basic keyword matching only
- **DeFi Integration:** No decentralized finance protocol analysis
- **NFT Metrics:** No non-fungible token market tracking
- **Dynamic Weights:** Market weights are manually assigned, not calculated
- **Backtesting Framework:** No systematic historical performance validation

7 Results and Analysis

7.1 System Performance

The implemented system successfully processes market data and generates sentiment scores in real-time. Key observations from the project include:

Table 4: System Performance Metrics

Metric	Value
Average Processing Time	< 5 seconds
Market Components Analyzed	13 instruments
Countries/Regions Covered	7 markets
Typical Active Components	11-13 (varies by market hours)
Data Update Frequency	Real-time during market hours

7.2 Sample Output Analysis

A typical system output shows the integration of multiple international markets:

Listing 2: Live System Output - June 8


```
{
  "timestamp": "2025-06-08T16:31:47.610323+00:00",
  "index_value": 59.83,
  "sentiment": "Shiny",
  "active_components": 11,
  "total_components": 13,
  "external_indicators": {
    "fear_greed_index": {"value": 62, "classification": "Greed"},
    "news_sentiment": 52.0
  },
  "regional_breakdown": {
    "us_markets_contribution": 23.63,
    "international_markets_contribution": 20.89,
    "risk_indicators_contribution": 8.75,
    "safe_havens_contribution": 6.56
  }
}
```


8 Challenges and Learning Outcomes

8.1 Technical Challenges Encountered

During the development of this project, several technical challenges were encountered and resolved:

- **Data Quality Issues:** Some international markets have different trading hours and holidays, requiring robust error handling
- **API Rate Limits:** Managing requests to avoid hitting Yahoo Finance API limitations
- **Currency Differences:** Handling markets traded in different currencies
- **Market Closures:** Dealing with weekends and holidays when markets are closed

8.2 Educational Insights Gained

This project provided valuable learning experiences in:

- Working with real-time financial data APIs
- Understanding international market correlations
- Implementing technical analysis algorithms
- Web development for data visualization
- Error handling and system reliability

9 Future Improvements and Extensions

9.1 Potential Enhancements

Several improvements could be made to extend this project:

- **Machine Learning Integration:** Adding ML models to improve sentiment prediction accuracy
- **More Markets:** Including additional emerging markets and asset classes
- **Historical Backtesting:** Implementing systematic backtesting capabilities
- **Mobile Application:** Creating a mobile app for real-time monitoring
- **Alert System:** Adding email/SMS notifications for significant sentiment changes

9.2 Academic Applications

This project could be extended for further academic research in:

- International finance correlation studies
- Behavioral economics research
- Market efficiency analysis
- Crisis prediction modeling

10 Current Limitations and Future Work

10.1 Current System Limitations

The current implementation has several limitations that could be addressed in future versions:

- **Basic Sentiment Analysis:** News sentiment analysis uses simple keyword matching rather than advanced natural language processing
- **Limited Cryptocurrency Coverage:** Only basic Bitcoin and Ethereum price tracking is implemented
- **No Machine Learning:** The system currently relies on traditional technical analysis without ML enhancement
- **Manual Weight Assignment:** Market weights are assigned manually rather than dynamically calculated

10.2 Proposed Future Enhancements

Future versions of the system could incorporate machine learning techniques:

$$\text{Enhanced\_Index} = \alpha \times \text{Traditional\_Score} + \beta \times \text{ML\_Prediction} + \gamma \times \text{Sentiment\_Analysis} \quad (10)$$

10.3 Additional Future Research Areas

As digital assets become more mainstream, future versions could explore:

- Advanced machine learning models for sentiment prediction
- Expanded cryptocurrency sentiment analysis
- DeFi protocol indicators (research needed)
- NFT market sentiment integration (experimental)
- Stablecoin adoption metrics (under development)

11 Conclusion

11.1 Project Summary

The Cloudy&Shiny Index project successfully demonstrates the implementation of a global market sentiment analysis system. Key achievements include:

1. **International Integration:** Successfully integrating 13 financial instruments from 7 different markets
2. **Real-time Processing:** Creating a system that processes live market data efficiently
3. **Technical Implementation:** Developing robust algorithms for sentiment calculation
4. **Educational Value:** Gaining practical experience with financial data analysis and programming
5. **Web Interface:** Building a user-friendly interface for visualizing results

11.2 Learning Outcomes

This project provided valuable educational experiences in:

11.2.1 Technical Skills

- Python programming for financial applications - API integration and data processing - Web development with HTML/CSS/JavaScript - Real-time data handling and error management

11.2.2 Financial Knowledge

- Understanding of international market dynamics - Technical analysis concepts and implementation - Market correlation and sentiment analysis - Cross-border financial relationships

11.2.3 Project Management

- Breaking down complex problems into manageable components - Version control and code organization - Documentation and presentation skills - Testing and debugging methodologies

11.3 Personal Reflection

Working on this project has deepened my understanding of global financial markets and the technical challenges involved in processing real-time financial data. The experience of integrating multiple international markets has highlighted the complexity of global finance and the importance of robust system design.

The project also demonstrated the value of combining theoretical knowledge with practical implementation, as many challenges only became apparent during the actual coding and testing phases.

12 Acknowledgments

I would like to thank my professors at TED University for providing the theoretical foundation that made this project possible. Special thanks to the instructors in the Computer Science and Economics departments for their guidance on financial programming and market analysis concepts.

I also acknowledge the open-source community for providing the libraries and tools used in this project, including Python, yfinance, pandas, and various web development frameworks.

13 References

References

[1] Python Software Foundation. (2024). Python programming language. *Available at:* `https://www.python.org/`

[2] Aroussi, R. (2024). yfinance: Yahoo Finance market data downloader. *PyPI Package*. `https://pypi.org/project/yfinance/`

[3] McKinney, W. (2024). pandas: Powerful data structures for data analysis in Python. *Available at:* `https://pandas.pydata.org/`

[4] Investopedia. (2024). Technical analysis and market sentiment indicators. *Financial Education Resource*. `https://www.investopedia.com/`

[5] Murphy, J. J. (1999). Technical analysis of the financial markets: A comprehensive guide to trading methods and applications. *New York Institute of Finance*.

[6] CNN Business. (2024). Fear & Greed Index. *Market Sentiment Indicator*. `https://www.cnn.com/markets/fear-and-greed`

A Mathematical Proofs

A.1 Convergence Properties of Distance-Based Scoring

Let $S_i(t)$ be the sentiment score for component i at time t. The distance-based scoring function has the following convergence property:

For any market component i with price $P_i(t)$ and moving average $MA_{30,i}(t)$, the distance-based score $S_i(t)$ tends toward 50 (neutral) as price movements stabilize around the moving average.

Let $D_i(t) = \frac{P_i(t) - MA_{30,i}(t)}{MA_{30,i}(t)}$ be the normalized difference.

As markets reach equilibrium, $D_i(t) \to 0$ due to mean reversion tendencies.

Since $S_i(t) = 50 + \frac{D_i(t)}{0.20} \times 50$, we have:

$$\lim_{D_i(t) \to 0} S_i(t) = 50 + \frac{0}{0.20} \times 50 = 50$$

Therefore, the distance-based scoring converges to neutral sentiment (50) under stable market conditions. ■

B Complete Algorithm Implementation

Listing 3: Cloudy&Shiny Index Implementation - Student Project

```
class CloudyShinyIndexCalculator:
    """
    Student Project: Global Market Sentiment Analysis System
    TED University - Computer Science/Economics
    """

    def __init__(self):
        # Market components with assigned weights
        self.components = {
            # US Markets (40% total weight)
            'SPY': {'weight': 0.20, 'name': 'S&P 500', 'region': 'US'},
            'QQQ': {'weight': 0.12, 'name': 'NASDAQ 100', 'region': 'US'},
            'IWM': {'weight': 0.08, 'name': 'Russell 2000', 'region': 'US'},

            # International Markets (35% total weight)
            '000001.SS': {'weight': 0.08, 'name': 'Shanghai Composite', 'region': 'China'},
            '^N225': {'weight': 0.07, 'name': 'Nikkei 225', 'region': 'Japan'},
            '^HSI': {'weight': 0.06, 'name': 'Hang Seng', 'region': 'Hong Kong'},
            '^GDAXI': {'weight': 0.05, 'name': 'DAX', 'region': 'Germany'},
            '^FCHI': {'weight': 0.04, 'name': 'CAC 40', 'region': 'France'},
            'XU100.IS': {'weight': 0.05, 'name': 'BIST 100', 'region': 'Turkey'},

            # Risk & Volatility (15% total weight)
            'VIX': {'weight': 0.10, 'name': 'Volatility Index', 'inverse': True},
```

```
            'TLT': {'weight': 0.05, 'name': 'US 20Y Treasury', 'inverse
                ': True},

            # Commodities & Safe Havens (10% total weight)
            'GLD': {'weight': 0.06, 'name': 'Gold', 'inverse': True},
            'DXY': {'weight': 0.04, 'name': 'US Dollar Index'}
        }

    def calculate_index(self):
        """
        Main calculation method for global market sentiment
        Returns comprehensive sentiment analysis
        """
        # Implementation details provided in main codebase
        pass
```

C Project Implementation Details

C.1 System Architecture

The project follows a modular architecture with the following components:

Table 5: System Components

Component	Function	Technology
Data Fetcher	Retrieves real-time market data	yfinance library
Calculator Engine	Processes sentiment scores	Python algorithms
Web Interface	Displays results	HTML/CSS/JavaScript
Data Storage	Saves historical data	JSON/CSV files
Error Handler	Manages API failures	Exception handling

Final Note: This document presents a student project developed for educational purposes. The system demonstrates practical application of financial programming concepts and provides valuable learning experiences in international market analysis and real-time data processing. IT SHOULD NOT BE USED FOR FINANCIAL DECISIONS.