Market Intelligence Publication System Documentation

Overview

The Market Intelligence Publication System is a comprehensive financial analysis platform that generates daily and weekly market intelligence reports by analyzing multiple data sources through a seven-framework contradiction analysis system.

Daily Publication Structure

1. Header Section

Purpose: Brand identity and date information **Data Source**: System-generated **Content**:

- Publication title: "Daily Market Pulse"
- Current date in formatted style
- Subtitle indicating comprehensive analysis scope

Status: ✅ Complete - No input files required

---

2. Hero Metrics Dashboard

Purpose: High-level system status and critical alerts **Input Files Required**:

- `market_trend_analysis_{date}.html` (for local sentiment)
- `market_sentiment_analysis_{date}.html` (for global sentiment)
- `hyg_report_{date}.html` (for credit data)
- `NiftyMRNPredictions_{date}.html` (for regime status)

Extracted Data:

```
🚨 System Alert Status: CRITICAL/WARNING/NORMAL
🌍 Global vs Local Sentiment: 6000% divergence
💳 Credit Data Integrity: 9.1% spread divergence
🔬 MRN Regime Status: 14/21 days (High transition probability)
```

Status: ⚠️ **Needs Fix** - Currently uses fallback data when files missing

---

3. Critical Divergence Alert

Purpose: Executive summary of most critical findings **Input Files**: All framework files (processed through contradiction analysis) **Content**:

- Bottom Line Up Front (BLUF) summary
- Number of critical frameworks
- Key contradictions identified
- Immediate action required notice

Status: ✅ Partially working - Dynamically generated from framework analysis

---

4. Seven-Framework Contradiction Analysis

Purpose: Detailed breakdown of each analytical framework

Framework 1: Global vs Local Sentiment

Input Files:

- `market_sentiment_analysis_{date}.html` → Global sentiment score
- `market_trend_analysis_{date}.html` → Local sentiment score

Calculation: `|(global_sentiment - local_sentiment) / local_sentiment| * 100` **Status**:
✅ Working with real data extraction

Framework 2: Economic Assessment vs Reality

Input Files:

- `economic_indicators_trend_{fy_start}_{date}.html`

Extracted Data:
- Risk index (35.9)
- Overall sentiment ("Strongly Bearish")
- Sentiment confidence ("High")
- Category scores (Market Fear: 50.0%, Interest Rates: 87.5%, etc.)
- Indicator distribution (3 bullish, 0 neutral, 13 bearish)

Calculation: Contradiction between bearish indicators vs optimistic assessment **Status**: ✅ Working with real data extraction

Framework 3: Credit Markets vs Fundamentals

Input Files:
- `hyg_report_{date}.html`

Extracted Data:
- Current HYG spread (3.23%)
- Calculated spread (2.96%)
- Divergence percentage
- Data quality score (80.6%)

Status: ✅ Working with real data extraction

Framework 4: Risk Assessment vs Market Activity

Input Files:
- `market_dashboard_{date}.html` (Global economic data)

Extracted Data:
- Overall market sentiment (-1.5)
- Bullish signals count (3)
- Bearish signals count (3)
- Neutral indicators (10)

Status: ✅ Working with real data extraction

Framework 5: Sector Sentiment Intelligence

Input Files:
- `sector_sentiment_allinone_{fy_start}_{date}.html`

Extracted Data:
- Overall assessment ("MODERATELY BEARISH AND DETERIORATING")
- Sector ratios (Power: 50.0, FMCG: 18.18, Telecom: -50.0)
- Turnaround alerts
- Top/avoid sectors

Status: ⚠️ **Needs Fix** - Sector cards generation hardcoded

Framework 6: Quantitative Regime Analysis

Input Files:
- `NiftyMRNPredictions_{date}.html`

Extracted Data:
- MI state ("ZERO")
- MI duration (14 days)
- Maximum duration (21 days)
- Transition probability

Status: ✅ Working with basic extraction

Framework 7: US Economic Backdrop

Input Files:

- Same as Framework 4 (`market_dashboard_{date}.html`)

Extracted Data:

- US-specific indicators identified
- Treasury stress signals
- Employment deterioration signals
- Fed policy uncertainty

Status: ✅ Working with real data extraction

---

5. Daily Stock Intelligence

Purpose: Actionable stock recommendations

Current Implementation (HARDCODED):

```
python

# Hardcoded in _get_fallback_dashboard_data()
"accumulation": ["APOLLOHOSP", "AUROPHARMA", "BAJAJFINSV", "BAJFINANCE", "BANDHANBNK", "BEL"
"distribution": ["ABFRL", "ALKEM", "ATUL", "AXISBANK", "BAJAJ-AUTO", "BALRAMCHIN"]
```

Required Implementation:

Input Files:

- `market_dashboard_{date}.html` → Extract from actual HTML tables

Real Data Structure:

```
html

<!-- From Bullish Stocks table -->
<table class="stock-table">
  <tr>
    <td>APOLLOHOSP</td>
    <td class="bullish">ACCUMULATION</td>
    <td>Healthcare</td>
    <td>₹5,847.30</td>
  </tr>
</table>

<!-- From Bearish Stocks table -->
<table class="stock-table">
  <tr>
    <td>ABFRL</td>
    <td class="bearish">SHORT</td>
    <td>Consumer</td>
    <td>₹285.45</td>
  </tr>
</table>
```

Status: ⚠️ **CRITICAL FIX NEEDED** - HTML extraction logic exists but may need refinement

---

6. Sector Intelligence Dashboard

Purpose: Sector-wise investment guidance

Current Issue: `generate_dynamic_sector_html()` method exists but fallback data still used

Input Files:

- `sector_sentiment_allinone_{fy_start}_{date}.html`

Real Data Expected:

```html
<div class="top-sectors-box">
  <div class="top-sector">
    <h4>1. Power</h4>
    <p>Ratio: 50.0</p>
    <p>Bullish: 75.0% | Neutral: 25.0% | Bearish: 0.0%</p>
  </div>
</div>
```

Status: ⚠️ **Needs Testing** - Dynamic extraction implemented but may need validation

---

7. Immediate Action Items

Purpose: Next 4-hour actionable steps **Data Source**: Aggregated from all frameworks and stock picks **Content**:

- Master arbitrage positioning
- Specific stock actions
- Credit market plays
- Sector rotation guidance

Status: ✅ Dynamically generated from analysis

Weekly Publication Additional Sections

8. Weekly Stock Performance & Attribution Analysis

Purpose: Track previous week's recommendation performance

Current Implementation (HARDCODED):

```python
stock_performance = {
    "winners": [
        {"symbol": "NTPC", "return": 12.4, "thesis": "Power sector leader..."},
    ],
    "losers": [
        {"symbol": "ZOMATO", "return": -18.4, "thesis": "Consumer Services..."},
    ]
}
```

Required Implementation:

Input Files Needed:

1. `recommendations_{date-7}.json` → Previous week's picks
2. Price data files for performance calculation

New Data Flow:

```
Week 1 (June 5): Generate picks → Save to recommendations_20250605.json
Week 2 (June 12): Load recommendations_20250605.json → Calculate actual performance
```

Status: ❌ **MAJOR FIX NEEDED**

Input File Requirements Summary

Required Daily Files:

```
C:/Projects/apps/institutional_flow_quant/output/progressive_analysis/
├── market_trend_analysis_20250605.html
└── market_dashboard_20250605.html

C:/Projects/apps/newsagent/data/processed/
└── news_dashboard_2025-06-05.html

C:/Projects/apps/institutional_flow_quant/output/sectortrend/
└── sector_sentiment_allinone_20250401_20250605.html

C:/Projects/apps/globalindicators/
├── reports/market_sentiment_analysis_20250605.html
├── data/market_dashboard_20250605.html
└── output/economic_indicators_trend_20250401_20250605.html

C:/Projects/apps/CodeRed/reports/
└── hyg_report_20250605.html

C:/Projects/apps/institutional_flow_quant/
└── NiftyMRNPredictions_20250605.html
```

Required Price Data (New):

```
data/prices/
├── NTPC_daily.csv
├── ITC_daily.csv
├── APOLLOHOSP_daily.csv
└── ... (all 200+ stocks)

data/recommendations/
├── recommendations_20250605.json
├── recommendations_20250604.json
└── ...
```

Critical Fixes Pending

1. Stock Performance Calculation (PRIORITY 1)

Problem: Hardcoded stock performance in weekly reports **Fix Required**:

- Implement `PriceDataManager` class
- Create recommendation tracking system
- Calculate real 7-day returns

2. Stock Pick Validation (PRIORITY 2)

Problem: May be using fallback stock lists instead of HTML extraction **Fix Required**:

- Validate HTML extraction from `market_dashboard_{date}.html`
- Test with real files to ensure bullish/bearish tables parsed correctly

3. Sector Card Generation (PRIORITY 3)

Problem: Sector intelligence may revert to fallback **Fix Required**:

- Verify dynamic sector HTML generation works with real data
- Test sector mapping and ratio calculations

4. Configuration Management (PRIORITY 4)

Problem: Hardcoded thresholds and stock universe **Fix Required**:

- Move to JSON configuration files
- Implement dynamic stock universe management

Stock Performance Implementation Plan

Phase 1: Data Collection

```
python

class PriceDataManager:
    def get_price_change(self, symbol: str, start_date: str, end_date: str) -> float:
        """Calculate percentage change between two dates"""

    def save_daily_recommendations(self, date_str: str, picks: Dict):
        """Save today's picks for future performance tracking"""
```

Phase 2: Performance Tracking

```
python

def calculate_recommendation_performance(self, prev_recommendations: Dict, current_date: str
    """
    Input: Previous week's recommendations + current date
    Output: Actual performance of those recommendations
    """
```

Phase 3: Report Integration

- Replace hardcoded performance with calculated results
- Add success rate metrics
- Include attribution analysis ("We said BUY → Stock went up +12%")

Expected Output Quality

With Real Data:

- **Credible Performance**: "Our NTPC recommendation from June 5 gained +12.4%"
- **Loss Avoidance**: "We said AVOID ZOMATO - it fell -18.4%, saving you losses"
- **Success Metrics**: "83% success rate this week, +12.8% alpha generated"

Current Output (Hardcoded):

- **Fictional Performance**: Made-up numbers with no tracking
- **No Credibility**: Cannot verify recommendations
- **No Learning**: System doesn't improve from past performance

The system architecture is solid, but these fixes are essential for production credibility and real-world usage.