Automated CIRF Framework Analysis - Complete Python Implementation

PROJECT OVERVIEW

I'm conducting doctoral research on the Cultural Innovation Resilience Framework (CIRF) and need to systematically collect and analyze 500+ documented cultural entrepreneurship failures. I need a complete Python implementation that can automate data collection, analysis, and CIRF component scoring.

CIRF FRAMEWORK (13 Components)

Operational Pillars (1-4):

1. Economic Value Creation - Sustainable revenue, financial viability
2. Cultural Integrity - Authentic cultural preservation
3. Adaptability - Response to changing conditions
4. Social Empowerment - Community capacity building

Community Control Filters (5-9): 5. Community Benefit - Positive community impact 6. Cultural Protection - Heritage safeguarding 7. Community Relevance - Alignment with community needs 8. Sustainable Development - Long-term viability 9. Dignity & Empowerment - Community self-determination

Resilience Capacities (10-13): 10. Protective Capacity - Asset protection abilities 11. Adaptive Capacity - Flexibility while maintaining identity 12. Transformative Capacity - Innovation while preserving culture 13. Generative Capacity - New opportunity creation

REQUIRED PYTHON IMPLEMENTATION

PHASE 1: AUTOMATED DATA COLLECTION

Task 1: Multi-Database Web Scraper

Create Python scripts to systematically scrape:

Academic Databases:

- Google Scholar (using `scholarly` library)
- JSTOR (API access where available)
- ResearchGate (web scraping)
- Academia.edu (web scraping)
- Government repositories (various APIs)

Search Terms to Automate:

```python
primary_terms = [
    "cultural entrepreneurship failure",
    "indigenous business closure",
    "traditional craft business failed",
    "cultural tourism failure",
    "ethnic minority business bankruptcy",
    "social enterprise collapse cultural",
    "community enterprise failure",
    "cultural heritage business closed"
]

geographic_modifiers = [
    "Canada", "Australia", "New Zealand", "United States",
    "Africa", "Asia", "Europe", "Latin America"
]

sector_modifiers = [
    "tourism", "crafts", "artisan", "heritage", "museum",
    "festival", "cooperative", "social enterprise"
]
```

Task 2: Automated Data Extraction

Create NLP pipeline to extract from academic papers:

- Business/enterprise name and type
- Geographic location and cultural context
- Failure type and timeline
- Key stakeholders involved
- Reasons for failure
- Evidence quality assessment

Task 3: Database Management System

Create SQLite database with proper schema:

```sql
```

```
CREATE TABLE failures (
    id INTEGER PRIMARY KEY,
    source_type TEXT,
    citation TEXT,
    title TEXT,
    authors TEXT,
    publication_date DATE,
    url TEXT,
    location_country TEXT,
    location_region TEXT,
    cultural_context TEXT,
    sector TEXT,
    failure_date DATE,
    failure_type TEXT,
    description TEXT,
    detailed_analysis TEXT,
    evidence_quality INTEGER,
    cirf_1 INTEGER,  -- Economic Value Creation
    cirf_2 INTEGER,  -- Cultural Integrity
    -- ... all 13 components
    cirf_13 INTEGER, -- Generative Capacity
    total_score INTEGER,
    percentage REAL,
    confidence_score REAL,
    notes TEXT,
    extraction_date DATE
);
```

PHASE 2: AUTOMATED CIRF ANALYSIS

Task 4: NLP-Based CIRF Component Scoring

Create machine learning model to automatically score CIRF components:

```python
def analyze_cirf_components(text_content):
    """
    Analyze text content and score each CIRF component
    Returns: dict with component scores (0, 1, or 0.5 for unclear)
    """
    # NLP analysis for each component
    # Use keyword matching, sentiment analysis, and pattern recognition
    # Return structured scoring
```

Component Analysis Keywords:

- **Economic Value Creation:** revenue, profit, income, sales, market, financial, sustainable, viable
- **Cultural Integrity:** authentic, traditional, heritage, cultural, preserve, respect, accurate
- **Adaptability:** flexible, adapt, change, respond, evolve, modify, adjust
- **Social Empowerment:** community, capacity, skills, leadership, participation, empowerment
- **Community Benefit:** benefit, positive, improve, help, support, local, community
- **Cultural Protection:** protect, safeguard, preserve, maintain, heritage, tradition
- **Community Relevance:** relevant, appropriate, needed, wanted, aligned, suitable
- **Sustainable Development:** sustainable, long-term, environmental, future, viable
- **Dignity & Empowerment:** dignity, respect, self-determination, autonomy, control
- **Protective Capacity:** protect, defend, safeguard, secure, shield, guard
- **Adaptive Capacity:** adapt, flexible, responsive, adjust, modify, resilient
- **Transformative Capacity:** transform, innovate, change, evolve, develop, create
- **Generative Capacity:** generate, create, produce, develop, build, establish

Task 5: Pattern Recognition and Analysis

Automated statistical analysis:

- Frequency analysis of component violations
- Geographic pattern identification
- Sector-specific vulnerability analysis
- Temporal trend analysis
- Correlation analysis between components
- Cluster analysis of failure types

PHASE 3: QUALITY ASSURANCE & VALIDATION

Task 6: Automated Quality Control

- Cross-reference validation across multiple sources
- Confidence scoring for each analysis
- Outlier detection and flagging
- Missing data identification
- Bias detection algorithms

Task 7: Interactive Dashboard

Create web dashboard using Streamlit or Dash:

- Real-time data collection monitoring
- CIRF component analysis visualization
- Geographic mapping of failures
- Sector analysis charts
- Pattern recognition results
- Export functionality for thesis writing

PHASE 4: RESEARCH OUTPUT GENERATION

Task 8: Automated Report Generation

Generate formatted outputs:

- Statistical summary reports
- Data visualization charts
- Academic-ready tables
- Geographic distribution maps
- Temporal trend analysis
- Component correlation matrices

Task 9: Thesis Integration Tools

- Export to LaTeX tables
- Generate APA citations automatically
- Create appendices with full datasets
- Produce supplementary statistical analyses

IMPLEMENTATION REQUIREMENTS

Code Structure:

```
cultural_entrepreneurship_analysis/
├── src/
│   ├── data_collection/
│   │   ├── scrapers/
│   │   ├── apis/
```

```
│   │   └── extractors/
│   ├── analysis/
│   │   ├── nlp_analysis.py
│   │   ├── cirf_scoring.py
│   │   └── statistical_analysis.py
│   ├── database/
│   │   ├── models.py
│   │   └── operations.py
│   └── visualization/
│       ├── dashboard.py
│       └── charts.py
├── data/
│   ├── raw/
│   ├── processed/
│   └── results/
├── notebooks/
│   ├── exploratory_analysis.ipynb
│   └── validation_analysis.ipynb
├── config/
│   ├── database_config.py
│   └── api_keys.py
└── requirements.txt
```

Key Libraries:

```
pandas>=1.5.0
numpy>=1.24.0
requests>=2.28.0
beautifulsoup4>=4.11.0
selenium>=4.8.0
scrapy>=2.8.0
nltk>=3.8.0
textblob>=0.17.0
scikit-learn>=1.2.0
matplotlib>=3.6.0
seaborn>=0.12.0
plotly>=5.13.0
streamlit>=1.20.0
sqlalchemy>=2.0.0
scholarly>=1.7.0
python-dotenv>=1.0.0
```

Deliverables:

1. **Complete Python codebase** with all functionality

2. **Automated data collection** running continuously
3. **SQLite database** with 500+ documented failures
4. **Interactive dashboard** for real-time analysis
5. **Statistical analysis reports** ready for thesis
6. **Visualizations** for academic publication
7. **Documentation** for reproducible research

Academic Rigor Requirements:

- **Version control** with Git for all code
- **Reproducible methodology** with detailed logging
- **Confidence scoring** for all automated analyses
- **Manual validation** protocols for quality assurance
- **Comprehensive documentation** for peer review
- **Export capabilities** for academic writing

Timeline:

- **Week 1-2:** Core scraping and database infrastructure
- **Week 3-4:** NLP analysis and CIRF scoring algorithms
- **Week 5-6:** Statistical analysis and pattern recognition
- **Week 7-8:** Dashboard development and visualization
- **Week 9-10:** Quality assurance and validation
- **Week 11-12:** Report generation and thesis integration

SUCCESS METRICS:

- Successfully collect 500+ documented failures
- Achieve >85% accuracy in CIRF component scoring
- Generate comprehensive statistical analysis
- Create publication-ready visualizations
- Develop reproducible methodology for peer review
- Complete integration with thesis writing process

This automated approach will provide doctoral-level rigor while handling the scale of analysis required for comprehensive CIRF framework validation.