FAO Analytics Platform - Project Plan

Executive Summary

This plan outlines the development of a comprehensive FAO data analytics platform that begins as a learning project and evolves toward commercial viability. The platform will provide multiple analytical capabilities including trade network analysis, anomaly detection, food security assessment, and market intelligence.

Project Vision

Primary Goals

1. **Learning**: Master large-scale data processing, network analysis, and machine learning on real-world agricultural data
2. **Commercial Potential**: Build valuable insights that could become a subscription service, API, or consultancy tool
3. **Flexibility**: Architecture that supports multiple analysis types without being locked into one approach

Target Customers (Future)

- Agricultural commodity traders
- Food security organizations
- Government policy makers
- Agricultural insurance companies
- Supply chain risk managers
- Research institutions

Technical Architecture

Project Structure

```
fao-analytics-platform/
├── README.md
├── .gitignore
├── requirements.txt
├── config/
│   ├── __init__.py
│   ├── settings.py          # Configuration management
│   └── data_catalog.yaml    # Track which FAO datasets we use
│
├── data/                    # Not in git
│   ├── raw/                 # Original FAO CSV files
│   │   ├── prices/
│   │   ├── trade/
│   │   └── production/
│   ├── processed/           # Parquet files (columnar format)
│   │   ├── bronze/          # Raw data in parquet
│   │   ├── silver/          # Cleaned, normalized
│   │   └── gold/            # Analysis-ready marts
│   └── cache/              # Temporary processing files
│
├── src/
│   ├── __init__.py
│   ├── core/               # Shared infrastructure
│   │   ├── data_access.py   # FAO data client
│   │   ├── warehouse.py     # Data warehouse operations
│   │   ├── processors.py    # Common transformations
│   │   └── models.py        # Pydantic models for data
│   │
│   ├── ingestion/          # Data acquisition
│   │   ├── download_fao.py  # Get data from FAO
│   │   ├── csv_to_parquet.py # Convert to efficient format
│   │   └── validate_data.py  # Quality checks
│   │
│   ├── analytics/          # Analysis modules
│   │   ├── __init__.py
│   │   ├── trade_network/   # Network analysis
│   │   │   ├── build_graph.py
│   │   │   ├── vulnerability.py
│   │   │   └── shock_propagation.py
│   │   │
│   │   ├── anomaly_detection/
│   │   │   ├── detector.py
│   │   │   ├── price_anomalies.py
│   │   │   └── trade_anomalies.py
│   │   │
│   │   ├── market_intelligence/
│   │   │   ├── price_integration.py
│   │   │   ├── arbitrage.py
│   │   │   └── forecasting.py
│   │   │
│   │   └── food_security/
│   │       ├── dependency_scores.py
│   │       └── risk_assessment.py
│   │
│   └── api/                # Future API layer
│       ├── __init__.py
│       └── endpoints.py
│
├── notebooks/              # Jupyter notebooks
│   ├── explorations/       # Try out ideas
│   ├── tutorials/          # Document learnings
│   └── reports/           # Polished analyses
│
├── outputs/               # Results
│   ├── visualizations/
│   ├── reports/
│   └── models/           # Saved ML models
│
├── tests/
│   ├── unit/
│   └── integration/
│
└── docker/               # Future containerization
    └── Dockerfile
```

Implementation Phases

Phase 1: Foundation (Weeks 1-3)

Goal: Set up infrastructure and understand the data

Tasks:

1. Set up GitHub repository with this structure
2. Create data ingestion pipeline
   - Download key FAO datasets (start with 5-10 tables)

  - Convert CSV to Parquet format
  - Build data catalog documenting what we have

3. Explore data in notebooks
   - Understand relationships between tables
   - Identify data quality issues
   - Find interesting patterns

Learning Focus:

- Parquet and columnar storage
- Data warehouse design patterns
- FAO data structure and quirks

Deliverables:

- Working data pipeline
- Data quality report
- Initial exploration notebooks

Phase 2: First Analysis Module (Weeks 4-6)

Goal: Build trade network analysis as proof of concept

Tasks:

1. Create trade network builder
   - Graph construction from trade matrix
   - Network metrics (centrality, clustering)
   - Visualization tools
2. Vulnerability assessment
   - Import dependency calculations
   - Single points of failure detection
   - Risk scoring algorithm
3. Build first data marts
   - Aggregated trade flows
   - Country dependency metrics

Learning Focus:

- NetworkX for graph analysis
- Graph algorithms
- Data aggregation strategies

Deliverables:

- Trade vulnerability dashboard (notebook)
- Network visualization tool
- Dependency risk scores by country

Phase 3: Anomaly Detection (Weeks 7-9)

Goal: Add ML-powered anomaly detection

Tasks:

1. Implement time series anomaly detection
   - Price spike detection
   - Unusual trade patterns
   - Production anomalies
2. Context-aware filtering
   - Distinguish real events from errors
   - Historical event correlation
3. Alert system design
   - Severity scoring
   - Notification framework

Learning Focus:

- Time series analysis
- Unsupervised ML techniques
- Feature engineering

Deliverables:

- Anomaly detection pipeline
- Historical anomaly analysis
- Alert configuration system

Phase 4: Platform Integration (Weeks 10-12)

Goal: Combine modules into cohesive platform

Tasks:

1. Build unified data warehouse
   - Design mart schema
   - Implement update pipelines
   - Optimize query performance
2. Create orchestration layer
   - Scheduled updates
   - Dependency management
   - Error handling
3. API design (FastAPI)
   - RESTful endpoints
   - Authentication system
   - Rate limiting

Learning Focus:

- Data warehouse best practices
- API design
- System architecture

Deliverables:

- Integrated analytics platform
- API documentation
- Performance benchmarks

Phase 5: Commercialization Prep (Weeks 13-15)

Goal: Prepare for potential commercial launch

Tasks:

1. Build demonstration interface
   - Streamlit dashboard

    - Interactive visualizations
    - Use case examples
2. Package insights products
    - Weekly vulnerability reports
    - Anomaly alerts
    - Network analysis summaries
3. Pricing and deployment strategy
    - Cloud deployment options
    - Cost analysis
    - Subscription tiers

Learning Focus:

- Product development
- Cloud deployment
- Business model design

Deliverables:

- Demo application
- Business plan
- Deployment guide

Data Prioritization Strategy

Tier 1: Essential Tables (Start Here)

1. **trade_detailed_trade_matrix** - Core network data
2. **production_crops_livestock** - Supply capacity
3. **prices** - Market signals
4. **food_balance_sheets** - Supply/demand picture
5. **area_codes** - Geographic reference
6. **item_codes** - Commodity reference

Tier 2: Enhanced Analysis

7. **climate_change_temperature** - Weather impacts

8. **emissions_agriculture** - Sustainability metrics
9. **food_security_indicators** - Vulnerability data
10. **exchange_rates** - Currency normalization

Tier 3: Advanced Features

- Consumer price indices
- Investment flows
- Population data
- Land use statistics

Technology Stack

Core Technologies

- **Python 3.11+** - Primary language
- **PostgreSQL** - Local development database
- **DuckDB** - In-process analytical queries
- **Parquet** - File storage format

Key Libraries

- **Data Processing**: pandas, polars, dask
- **Network Analysis**: networkx, graph-tool
- **ML/Anomaly Detection**: scikit-learn, prophet, pytorch
- **Visualization**: plotly, altair, kepler.gl
- **API**: FastAPI, pydantic
- **Orchestration**: Prefect or Airflow

Cloud Services (Future)

- **Storage**: AWS S3 or Google Cloud Storage
- **Compute**: AWS Lambda or Google Cloud Functions
- **Database**: Supabase or AWS RDS
- **API Hosting**: Railway or Fly.io

Success Metrics

Technical Metrics

- Query performance: <1s for standard queries
- Data freshness: <24 hours from FAO updates
- System uptime: 99.9%
- Test coverage: >80%

Learning Metrics

- Technologies mastered
- Algorithms implemented
- Insights discovered
- Code quality improvements

Business Metrics (Future)

- Number of valuable insights generated
- Potential customer interviews conducted
- MVP user feedback collected
- Revenue model validated

Risk Mitigation

Technical Risks

- **Data Volume**: Start with filtered subsets, expand gradually
- **Complexity**: Build incrementally, one module at a time
- **Performance**: Use Parquet and data marts for speed

Business Risks

- **Market Fit**: Validate with potential users early
- **Competition**: Focus on unique insights from network analysis
- **Data Rights**: Ensure compliance with FAO data usage terms

Next Steps

Week 1 Checklist

☐ Create GitHub repository
☐ Set up Python environment
☐ Download first FAO dataset (suggest: prices)

- □ Convert to Parquet format
- □ Create first exploration notebook
- □ Document learnings in README

Quick Start Commands

```bash
# Clone and setup
git clone https://github.com/[your-username]/fao-analytics-platform
cd fao-analytics-platform
python -m venv venv
source venv/bin/activate  # or venv\Scripts\activate on Windows
pip install -r requirements.txt

# Download first dataset
python src/ingestion/download_fao.py --dataset prices --years 2020-2023

# Convert to Parquet
python src/ingestion/csv_to_parquet.py data/raw/prices.csv

# Launch notebook
jupyter notebook notebooks/explorations/01_explore_prices.ipynb
```

Learning Resources

Recommended Learning Path

1. **Parquet & Data Warehousing**: Apache Arrow documentation
2. **Network Analysis**: NetworkX tutorials
3. **Time Series Anomaly Detection**: Facebook Prophet docs
4. **MLOps**: MLflow or Weights & Biases tutorials
5. **API Development**: FastAPI course

Useful References

- FAO Data Documentation: [link to FAO docs]
- Modern Data Stack: dbt, Airbyte, Superset
- Graph Analytics: Neo4j Graph Academy
- Agricultural Economics: FAO publications

Commercial Considerations

Potential Revenue Models

1. **SaaS Subscription**: $500-5000/month per organization
2. **API Access**: Usage-based pricing for data/insights
3. **Custom Reports**: $5000-50000 per analysis
4. **Consulting**: Implementation and training services

Competitive Advantages

- Comprehensive FAO data integration
- Network-based insights (unique angle)
- Real-time anomaly detection
- Academic rigor with commercial applicability

MVP Features for Commercial Launch

- Trade vulnerability dashboard
- Weekly anomaly reports
- API with 3 key endpoints
- Basic user authentication
- Usage analytics

Conclusion

This project offers an excellent learning opportunity while building toward commercial value. The key is to start simple, iterate based on discoveries, and maintain flexibility to pivot based on what you learn about the data and market needs.

Remember: Every major AgTech company started by solving one specific problem really well. Your journey begins with understanding the data, then finding the most valuable insights hidden within it.