INNOFarms.AI Data Analyst Intern

Technical Assignment

Agricultural Market Intelligence & Financial Analysis System

📋 Assignment Overview

Duration: 24 hours

Objective: Build a data pipeline that scrapes, analyzes, and provides financial insights on agricultural commodities

Part 1: Data Collection & Analysis

1.1 Web Scraping

Task: Scrape agricultural price data from **3 different sources**

Recommended Sources:

1. **AGMARKNET** (https://agmarknet.gov.in/) - Indian market prices
2. **USDA NASS** (https://quickstats.nass.usda.gov/) - US agricultural data
3. **Investing.com** (https://www.investing.com/commodities/agricultural) - Commodity futures

Requirements:

- Scrape at least **5 commodities** (Wheat, Rice, Corn, Tomato, Onion)
- Extract: commodity name, price, unit, date, source
- Handle errors, rate limiting (2-5 sec delays)
- Save raw data as JSON/CSV

Deliverable:

```
src/scrapers/
  ├── agmarknet_scraper.py
  ├── usda_scraper.py
  └── investing_scraper.py
data/raw/
  └── [source]_[date].json
```

1.2 Database & ETL

Task: Design database and build ETL pipeline

Required Tables:

- `commodities:` commodity_id, name, category
- `sources:` source_id, name, url, reliability_score
- `price_data:` price_id, commodity_id, source_id, price, unit, date
- `daily_aggregates:` commodity_id, date, avg_price, min, max, std_dev

ETL Requirements:

- Normalize all prices to common unit (e.g., INR/quintal)
- Remove duplicates
- Detect anomalies (price > 3 standard deviations)
- Create daily aggregates

Deliverable:

```
sql/schema.sql
src/etl/etl_pipeline.py
database/market_data.db
```

1.3 Analysis & Forecasting

Task: Build price analysis and forecasting system

Required Analysis:

- Moving averages (7-day, 15-day, 30-day)
- Trend detection (upward/downward/stable)
- Price volatility (coefficient of variation)

Forecasting Models (implement 2):

4. **Moving Average** - Simple baseline
5. **ARIMA/Prophet** - Time series model

Evaluation Metrics:

- MAPE (Mean Absolute Percentage Error)
- RMSE (Root Mean Square Error)

Output Format:

```
{
  "commodity": "Wheat",
  "current_price": 245.50,
  "7day_avg": 242.30,
  "trend": "Upward",
  "volatility": "Low (3.2%)",
  "forecast_7day": [246, 247, 248, 249, 250, 251, 252],
  "model_accuracy": {"mape": 2.3, "rmse": 5.8}
}
```

Deliverable:

```
src/analytics/
  ├── trend_analyzer.py
  └── forecasters.py
reports/analysis_output.json
```

💰 Part 2: Financial Analysis

2.1 Cost & Profitability Calculator

Task: Calculate production costs and profit projections

Cost Structure (create for 5 commodities):

```
wheat:
  cycle_days: 120
  yield_per_hectare: 40  # quintal
  costs:
    seeds: 3000
    fertilizers: 8000
    labor: 12000
    total: 45000
```

Calculations Required:

```
# Per hectare metrics
revenue = market_price × yield
profit = revenue - total_cost
profit_margin = (profit / revenue) × 100
roi = (profit / total_cost) × 100

# Efficiency metrics
profit_per_day = profit / cycle_days
cycles_per_year = 365 / cycle_days
annual_profit = profit × cycles_per_year
```

Deliverable:

```
config/production_costs.yaml
src/financial/cost_calculator.py
src/financial/profit_calculator.py
```

2.2 Opportunity Ranking

Task: Rank commodities by profitability and risk

Scoring Criteria:

```
weights = {
    'profit_margin': 0.30,
    'roi': 0.25,
    'profit_per_day': 0.20,
    'price_stability': 0.15,
    'demand_trend': 0.10
}
```

Risk Levels:

- **Low:** Volatility < 10%
- **Medium:** Volatility 10-20%
- **High:** Volatility > 20%

Deliverable:

```
src/financial/
  ├── opportunity_scorer.py
  └── risk_analyzer.py
reports/opportunity_rankings.json
```

2.3 Scenario Planning

Task: Build break-even and sensitivity analysis tools

Break-Even Analysis:

```
breakeven_price = total_cost / yield
breakeven_yield = total_cost / market_price
margin_of_safety = (market_price - breakeven_price) / market_price × 100
```

Sensitivity Scenarios:

```
scenarios = {
    'pessimistic': current_price × 0.80,  # -20%
    'realistic': current_price,
    'optimistic': current_price × 1.20    # +20%
}
# Calculate profit for each scenario
```

Portfolio Allocation:

Given 100 hectares, suggest crop allocation to maximize profit. Consider: diversification, risk, expected returns

Deliverable:

```
src/financial/scenario_planner.py
```

```
reports/scenario_analysis.json
reports/portfolio_recommendation.json
```

📦 Submission Requirements

Required Files:

```
project/
├── README.md
├── requirements.txt
├── config/
├── sql/
├── src/
├── database/
├── reports/
└── logs/
```

Documentation:

README.md must include:

- Installation steps
- How to run the pipeline
- Data sources used
- Key findings (top 3 commodities with justification)

Submission Package:

6. **GitHub Repository Link**
7. **Video Walkthrough (5-10 min)** showing:
    - Scraper running
    - Database tables
    - Analysis outputs
    - Key recommendations
8. **Technical Report (2-3 pages PDF):**
    - Data sources & challenges
    - Cost assumptions
    - Model selection reasoning
    - Top 3 commodity recommendations
    - Challenges faced & solutions

⚖️ Evaluation Criteria

Category	Points
Web Scraping (3 sources, error handling)	20
Database Design & ETL	20
Analytics & Forecasting	20
Financial Modeling	20
Code Quality & Documentation	15
Deliverables Complete	5
TOTAL	**100**

Bonus Points (+10):

- Unit tests
- Visualizations
- Advanced features

📤 How to Submit

Email to: hiring@innofarms.ai

Include:

9. GitHub repository link
10. Video walkthrough link (YouTube/Loom)
11. Technical report PDF
12. ZIP file with database + reports

Deadline: 24 hours from assignment receipt

💡 Tips

- Start with AGMARKNET (easiest to scrape)
- Focus on Part 1 first, then Part 2
- Document assumptions clearly
- Test incrementally, don't wait till the end
- Simple working solution > complex broken one
- Handle errors gracefully

Questions? Email: Ashu.Pal@INNOFarms.AI (clarifications only, no debugging help)

Good luck! 🚀

INNOFarms.AI - Building the AWS of Future Food Systems