🚀 Advanced Data Cleaning Agent - Setup Guide

Requirements (requirements.txt)

```txt
streamlit>=1.28.0
pandas>=1.5.0
numpy>=1.24.0
langgraph>=0.0.40
openai>=1.0.0
matplotlib>=3.7.0
seaborn>=0.12.0
```

Quick Start

1. **Install Dependencies**

```bash
pip install -r requirements.txt
```

2. **Run the Application**

```bash
streamlit run advanced_data_cleaning_agent.py
```

3. **Get Your OpenAI API Key**
    - Go to OpenAI Platform
    - Create an account and get your API key
    - Enter it in the sidebar of the app

Key Features That Make This Special

🧠 Memory System

- **Learns from Experience**: Remembers successful cleaning patterns
- **Avoids Mistakes**: Stores failed attempts to prevent repetition
- **Gets Smarter**: Improves recommendations over time

💰 Cost Optimization

- **Rule-Based First**: Uses Python logic before LLM calls
- **Smart Caching**: Avoids repeated expensive operations
- **Progressive Enhancement**: LLM only for complex cases

🔄 Robust Workflow

- **Error Recovery**: Handles failures gracefully
- **Validation Loop**: Ensures data integrity
- **Progress Tracking**: Shows exactly what's happening

🎯 User Experience

- **Visual Progress**: Real-time workflow execution
- **Before/After Comparison**: Clear quality metrics
- **Downloadable Results**: Clean data ready to use

Understanding the Architecture

Workflow Steps:

1. **Schema Analyzer** → Understands data structure (no LLM)
2. **Quality Assessor** → Identifies issues (smart rules + minimal LLM)
3. **Cleaning Planner** → Creates strategy (memory + rules)
4. **Code Generator** → Writes Python code (LLM when needed)
5. **Executor** → Runs code safely (no LLM)
6. **Validator** → Checks results (no LLM + learning)

Why This Beats Traditional Approaches:

- **10x Cost Reduction**: Smart LLM usage
- **Better Reliability**: Validation at every step
- **Learning Capability**: Improves with each dataset
- **Transparency**: Shows decisions and reasoning

Advanced Usage

Memory Management

The agent automatically stores:

- Successful cleaning patterns
- Failed attempts with reasons
- User preferences and feedback

Cost Tracking

Monitor your API usage:

- Total cost per session
- Cost per row processed
- Number of LLM calls made

Extensibility

Easy to add new cleaning strategies:

1. Add new issue types in `quality_assessor_node`
2. Create corresponding code generators
3. Add validation logic

Troubleshooting

Common Issues:

1. **"No module named 'langgraph'"** → Install with `pip install langgraph`
2. **"OpenAI API key not found"** → Enter key in sidebar
3. **"Workflow execution failed"** → Check data format and try again

Performance Tips:

- Use smaller datasets for testing
- Monitor cost in the sidebar
- Check memory stats to see learning progress

Next Steps

1. **Try Different Datasets**: See how memory improves performance
2. **Experiment with Settings**: Different models, strategies
3. **Extend Functionality**: Add custom cleaning rules

4. **Monitor Costs**: Track API usage and optimization

This agent represents a new paradigm in data cleaning - intelligent, cost-effective, and continuously learning. Enjoy exploring its capabilities! 🎉