Prompt Engineering with CURSOR AI

Created by: Fabio Classo

2025

E-commerce Analytics Project

A comprehensive Python-based e-commerce analytics system that generates synthetic data, performs data quality validation, creates metrics, and generates visualizations for business insights.

Data Generation

Create realistic synthetic e-commerce data including users, products, sales, payments, and sellers with configurable parameters and business rules.

Data Quality Validation

Comprehensive validation using Pydantic models with business rules, quality scoring, and detailed error reporting for data integrity.

Metrics & Visualization

Calculate key business KPIs and generate interactive charts and dashboards for comprehensive business insights and analysis.

🚀 Quick Start Guide

01

Generate Synthetic Data

Create realistic e-commerce data for analysis including 1000 users, 500 products, 5000 sales transactions, and payment records.

```
python generate_fake_data.py
```

02

Generate Business Metrics

Calculate key business metrics and KPIs including sales performance, customer behavior, and product analytics.

```
python generate_metrics_dataframes_refactored.py
```

03

Create Visualizations

Generate comprehensive analytics dashboards with sales overview, customer analysis, and data quality validation charts.

```
python ecommerce_analyzer.py
```

04

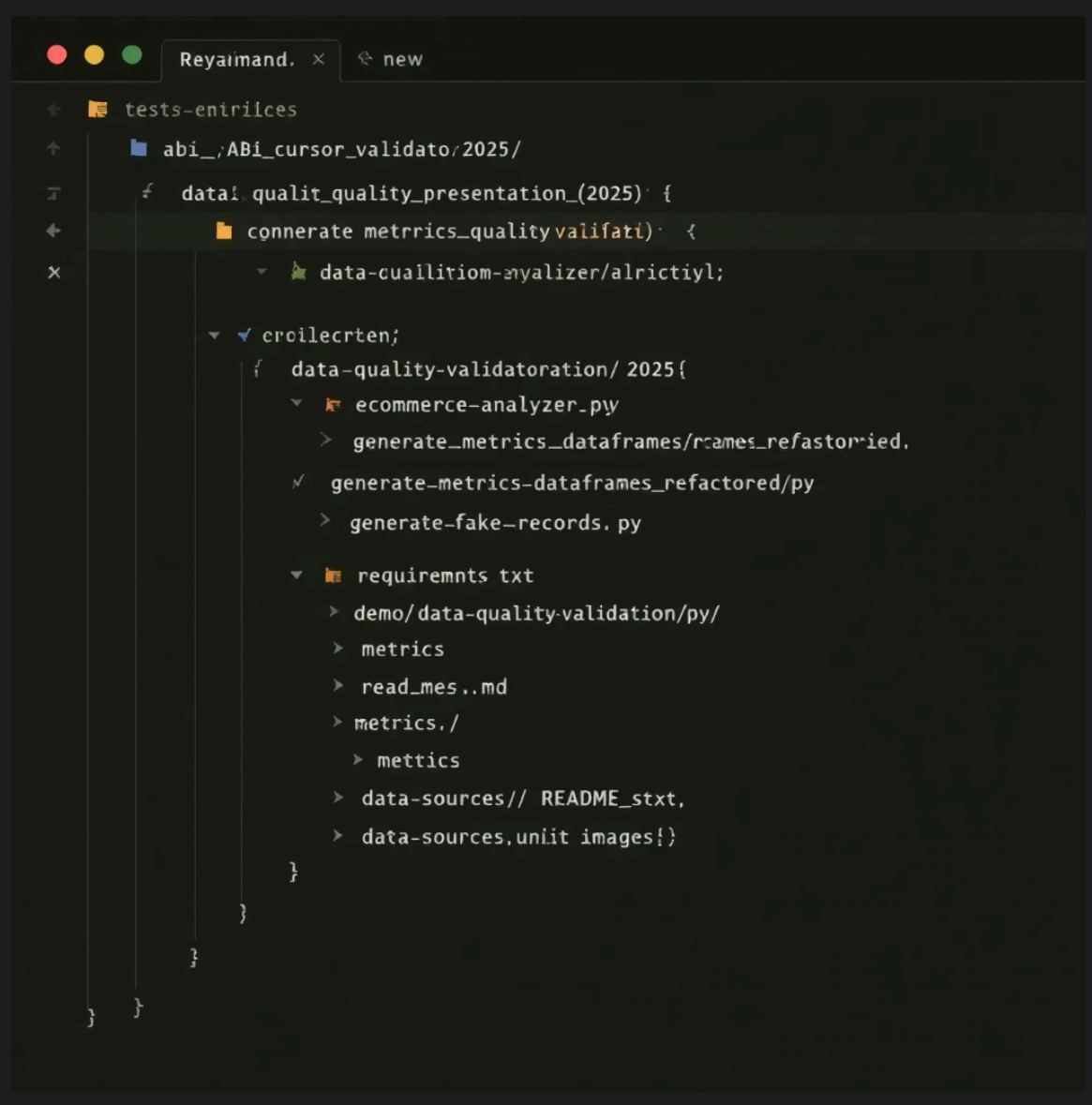
Validate Data Quality

Run comprehensive data quality validation demo with quality scoring, error reporting, and validation report exports.

```
python demo_data_quality_validation.py
```

📁 Project Structure

- **data_quality_validator.py** - Pydantic data validation models
- **ecommerce_analyzer.py** - Analytics visualization generator
- **generate_metrics_dataframes_refactored.py** - Metrics generation with dependency injection
- **generate_fake_data.py** - Synthetic data generation
- **generate_bad_records.py** - Generate data with quality issues
- **demo_data_quality_validation.py** - Demonstration script
- **tests/** - Comprehensive test suite with pytest
- **images/** - Generated visualizations and charts



Prerequisites

- Python 3.8 or higher
- pip (Python package installer)
- Virtual environment (recommended)

Data Files Generated

Users: Personal information, demographics, contact details

Products: Catalog with pricing, categories, inventory

Sales: Transaction records with amounts, dates, status

Payments: Payment methods, amounts, transaction IDs

Sellers: Company information, ratings, verification status

Analytics & Metrics

Sales Metrics: Monthly sales, status distribution

Product Analytics: Top products by quantity and revenue

Customer Analytics: Demographics, purchase behavior

Payment Analytics: Method distribution, success rates

Geographic Analytics: Distribution by city, state, country

Visualizations Created

Sales Overview: Revenue trends, status distribution

Customer Analysis: Demographics, behavior patterns

Product Performance: Top sellers, revenue analysis

Payment Analysis: Method usage, success rates

Data Quality Dashboard: Validation results and quality scores

Advanced Features & Implementation

🔍 Data Quality Validation

The project includes comprehensive data quality validation using Pydantic models with business rules and quality scoring.

```
from data_quality_validator import DataQualityValidator
import pandas as pd

# Initialize validator
validator = DataQualityValidator()

# Load and validate data
df = pd.read_csv('your_data.csv')
results = validator.validate_dataframe(df, 'users')

# Check results
print(f"Quality Score: {results['data_quality_score']:.2%}")
print(f"Valid Records: {results['valid_records']}")
print(f"Invalid Records: {results['invalid_records']}")
```

🧪 Testing & Quality Assurance

Comprehensive test suite with pytest ensuring code quality and reliability across all components.

```
# Run all tests
python -m pytest

# Run specific test files
python -m pytest tests/unit/test_data_quality_validator.py

# Run with verbose output
python -m pytest -v

# Run with coverage
python -m pytest --cov=.
```

1000

User Records

Generated synthetic user data with demographics and contact information

500

Product Catalog

Diverse product inventory with pricing and category information

5000

Sales Transactions

Realistic transaction data with various statuses and payment methods

50

Seller Profiles

Company information with ratings and verification status

🎓 Learning Objectives & Technical Excellence

Data Engineering

Demonstrates synthetic data generation and ETL processes with realistic business scenarios and data relationships.

- Synthetic data generation with Faker library
- Data pipeline orchestration and dependency management
- ETL process implementation with pandas

Data Quality & Validation

Comprehensive validation, cleansing, and quality assessment using modern Python frameworks and best practices.

- Pydantic model validation with business rules
- Quality scoring and error reporting systems
- Data cleansing and transformation processes

Analytics & Visualization

Business metrics calculation, KPI tracking, and interactive dashboard creation for comprehensive data insights.

- Business metrics calculation and KPI tracking
- Interactive chart generation with matplotlib and seaborn
- Dashboard creation and data storytelling

🔧 Performance & Troubleshooting

Performance Tips

- **Large Datasets:** Use chunked processing and parallel processing where available
- **Faster Validation:** Validate data in batches and cache results
- **Better Visualizations:** Use appropriate chart types and optimize resolution

Common Issues

- **Import Errors:** Ensure all dependencies are installed via requirements.txt
- **File Not Found:** Generate data first using generate_fake_data.py
- **Memory Issues:** Reduce data size parameters for large datasets

Configuration

- **Data Generation:** Modify record counts and data ranges
- **Validations:** Customize business rules and constraints
- **Visualizations:** Adjust chart styles and output formats

🚀 Happy Analyzing!

This comprehensive e-commerce analytics project demonstrates professional-grade data engineering, quality validation, unit testing and visualization techniques.

Created by: Fabio Classo | ABInbev | 2025