

# Executive Summary: LLM-Powered Automated Report Generation System

## System Overview

This system automates business report generation from structured sales data using Large Language Models (LLMs) orchestrated through LangGraph. It analyzes datasets, extracts insights, generates visualizations, and compiles professional reports in multiple formats (PDF, HTML, Word, Markdown). The architecture is provider-agnostic, supporting local inference (Ollama) and cloud LLMs (OpenAI, Anthropic, Gemini).

**Value Proposition:** Transform raw data into actionable intelligence reports through multi-agent LLM orchestration, reducing manual analysis from hours to minutes with automated quality evaluation.

## Architecture & Design Approach

### 1. Stateful Workflow Orchestration with LangGraph

The system uses **LangGraph** for multi-step workflow orchestration with providing state management. The workflow consists of **9 sequential nodes**:

```
load_and_profile → analyze →  
plan_viz → generate_plots →  
write_summary → write_analysis → write_recommendations →  
compile → evaluate
```

Each node is a function that receives centralized **ReportState**, performs operations (LLM calls, data processing, file generation), and returns updated state.

### 2. Multi-Agent LLM Architecture

The system employs **specialized agents** with domain-specific capabilities:

- **Data Analysis Agent:** Uses LangChain's `create_agent()` with structured tools (`inspect_data_columns`, `get_statistical_summary`, `analyze_relationships`) to explore data
- **Visualization Planning Agent:** Plans chart types based on data characteristics
- **Report Writing Agents:** Generate summaries, analysis sections, and recommendations with role-specific prompts

**Rationale:** Specialized agents with tool-calling capabilities and unique prompts yield higher quality outputs than monolithic prompts.

### 3. Type-Safe Configuration

Configuration uses **Pydantic-Settings** with nested models for validation, type safety, and automatic environment variable binding. Field validators ensure provider/model compatibility at startup, preventing runtime errors. Sensitive keys are automatically masked in logs.

### 4. Intelligent Data Pipeline

The **BMWDataLoader** employs **fuzzy column name matching** to handle diverse Excel schemas (e.g., "Date", "sale\_date", "TransactionDate" → "date"). Combined with **StatisticalAnalyzer**, it generates comprehensive data profiles (temporal trends, regional distributions, model performance) as part of the LLM analysis.

---

## Evaluation Framework

### Overview

The **ReportEvaluator** provides **automated quality assessment** without human intervention, addressing the challenge of validating LLM-generated content at scale. Evaluation runs as the final workflow node, scoring reports across multiple dimensions.

### Metrics

1. **Coverage Score (40% weight)** — Measures whether insights span 4 key dimensions:

- Temporal (growth trends, time-series patterns)
- Regional (geographic distribution)
- Product (model-level analysis)
- Performance (revenue, volume, pricing)

Implementation uses keyword matching with expanded vocabularies.

2. **Completeness Score (30% weight)** — Validates presence of required sections (Executive Summary, Trends, Recommendations) and minimum length (1,000 characters). Formula:  $0.7 \times \text{section\_score} + 0.3 \times \text{length\_score}$

3. **Readability Score (30% weight)** — Assesses sentence length (optimal: 15-20 words) and document structure (headings, lists). Formula:  $0.7 \times \text{sentence\_score} + 0.3 \times \text{structure\_score}$

4. **Overall Score** — Weighted composite:  $0.4 \times \text{coverage} + 0.3 \times \text{completeness} + 0.3 \times \text{readability}$