# The Evolution of Data Products

Abdelkrim Alfalah

13/12/2024

# Traditional Data Products as Scoring Engines

in the ML Era (before the advent of LLMs), most data products were designed to process structured data and generate **structured outputs**, such as scores, classifications, and recommendations. Either using Supervised or Unsupervised techniques.

These outputs were concise, quantitative, and task-specific, representing the culmination of complex algorithms applied to curated datasets.

# What Are Scoring Engines?

A **scoring engine** is a system or model designed to produce a numerical or categorical output that serves as a decision-making tool. The output often represents a prediction, ranking, or classification derived from input features.

Examples include:

- **Risk Scores**: Predicting the likelihood of loan default or insurance claims.
- **Propensity Scores**: Calculating the likelihood of a user purchasing a product or clicking on an ad.
- **Recommendations**: Suggesting items, movies, or songs based on a user's history.

# Key Features of Traditional Scoring Engines

1. **Structured Inputs**: predefined, static, and carefully engineered, fixed schema with limited flexibility.
2. **Task-Specific Models**: specific purpose, Models were narrow, not adaptable to other tasks without retraining.
3. **Explainable Outputs**: easy to interpret and use for decision-making.
4. **Minimal Context Awareness**: Context was handled externally through feature engineering or preprocessing.
5. **Deterministic Nature**: Given the same input, consistent output.

# Examples of Traditional Data Products

- **Credit Scoring Systems**:
  - Used by banks to assess the likelihood of a borrower repaying a loan.
  - Inputs: Payment history, income, debt-to-income ratio.
  - Output: A single numerical score.
- **Recommendation Engines**:
  - Used by platforms like Netflix or Amazon to suggest items based on user behavior.
  - Inputs: Viewing history, ratings, item similarity.
  - Output: A ranked list of recommended items.
- **Fraud Detection Models**:
  - Used to identify fraudulent transactions in real-time.
  - Inputs: Transaction amount, location, device information.
  - Output: A binary classification (fraudulent or not).
- **Customer Churn Prediction**:
  - Used to predict if a customer is likely to leave a service.
  - Inputs: Usage frequency, customer support interactions, billing data.
  - Output: A probability score indicating churn likelihood.

# Limitations of Traditional Scoring Engines

1. **Rigid Feature Engineering**:
   - Data scientists spent significant time curating and engineering features to extract predictive power.
   - Models were brittle and couldn't adapt to new data formats or sources.
2. **Lack of Contextual Understanding**:
   - Scoring engines could not dynamically adapt to unstructured or real-time inputs.
3. **Narrow Use Cases**:
   - Each engine was built for a specific problem and required retraining or redesign for any new application.
4. **Limited User Interaction**:
   - Outputs were presented as static results, leaving little room for nuanced, interactive explanations or dialogue.

# Transition to LLMs: From Scores to Stories

- **LLMs as Generative Engines**:
  - Large Language Models (LLMs) represent a transformative leap in AI, shifting the paradigm from generating structured, task-specific outputs (like scores or classifications) to producing **unstructured, context-rich outputs** such as natural language responses, insights, and creative content.
  - These models are capable of generating free-form text that is coherent, context-sensitive, and adaptive, making them powerful tools across a wide range of applications.

# From Structured to Unstructured

- While traditional data products excelled at producing structured outputs, LLMs generate **unstructured outputs** like text, conversation, and narratives.
- Example: An LLM augmented risk-scoring system might **try** to tell you *why* a certain decision was made in plain language, rather than just providing a number.

# Versatility of LLM Applications 1

LLMs are highly adaptable and have found applications in a variety of domains, proving their flexibility and transformative potential:

## Communication and Assistance

- **Chatbots**: Virtual assistants powered by LLMs can handle customer support, troubleshoot issues, or guide users through a product.
- **Customer Support**: Automating responses to frequently asked questions while escalating complex queries to human agents.

## Content Creation

- **Copywriting**: Generating marketing copy, social media posts, or ad content tailored to a specific audience.
- **Creative Writing**: Producing stories, poems, or even screenplays based on prompts.
- **Blog and Article Writing**: Assisting with drafting articles, especially for SEO or specific topics.

# Versatility of LLM Applications 2

## Personalised Education

- **Tutoring Systems**: Providing personalised explanations, answering questions, or generating study materials for learners.
- **Language Translation and Learning**: Real-time translation and language practice for global communication.

## Data Analysis and Interpretation

- **Summarisation**: Extracting and condensing information from long documents, reports, or meeting transcripts.
- **Business Insights**: Generating explanations or recommendations from structured data presented in natural language.

## Healthcare and Medicine

- **Medical Summaries**: Generating simplified explanations of medical reports for patients.
- **Clinical Decision Support**: Assisting doctors by synthesising and contextualizing patient data.

# The Power of Context in LLMs

One of the most revolutionary aspects of LLMs is their **ability to handle and adapt to context**. This feature sets them apart from traditional data products like scoring engines.

## Dynamic Context Handling

- LLMs use an internal **context window** to process all relevant information provided in the input.
- Example: In a legal assistant application, the model can consider the entire case file when drafting a response, tailoring the output to specific legal precedents or clauses.

# The Power of Context in LLMs 2

## Context Awareness:

- **Context retention**:
  - LLMs retain and build upon context in dialogues, enabling more natural and coherent interactions.
  - Example: A customer support bot can recall a user's query history to provide seamless assistance.
- **Contextual Relevance**:
  - By analysing the input holistically, LLMs focus on the most relevant parts, filtering out irrelevant details.
  - Example: In summarisation, they can distill the most important aspects of a 50-page document into a few sentences.

# The Power of Context in LLMs 3

**Flexibility Across Domains**:

- Unlike traditional scoring engines, which require domain-specific tuning, LLMs are versatile and general-purpose, capable of adapting their context handling to various fields without retraining.

# Challenges and Risks in the Post-LLM Era

The introduction of Large Language Models (LLMs) has unlocked immense possibilities, but it also brings new challenges and risks that must be addressed to ensure their effective and responsible use.

# Computational and Storage Costs

- **High Resource Requirements**:
  - Training, fine-tuning, and running LLMs require significant computational power, often relying on expensive GPUs .
  - Serving LLMs in real-time applications further increases costs due to the need for low-latency, high-throughput infrastructure.
- **Scaling Challenges**:
  - As context windows grow larger (e.g., 32k tokens in GPT-4), the computational cost per query increases substantially.
  - Storage requirements for fine-tuned models and logging massive volumes of input-output pairs add further overhead.
- **Mitigation Strategies**:
  - Use **retrieval-augmented generation (RAG)** to reduce input size by dynamically fetching only the most relevant context.

# Complexity in Evaluation

- **Beyond Traditional Metrics**:
  - Traditional metrics such as accuracy, precision, and recall are insufficient for evaluating LLMs, as they produce unstructured and highly variable outputs.
  - New metrics are needed, such as:
    - **Relevance**: How well the output aligns with the input query.
    - **Coherence**: Logical consistency and fluency of the response.
    - **Engagement**: How effectively the output interacts with users.
    - **Bias Detection**: Identifying and quantifying unintended biases in outputs.

# Complexity in Monitoring

- **Continuous Monitoring**:
  - LLMs require constant monitoring to ensure performance, accuracy, and ethical adherence.
  - Real-time applications, such as chatbots, necessitate **always-on observability** to handle evolving inputs and contexts.
- **Dynamic Performance Monitoring**:
  - LLM performance can degrade over time due to issues like **data drift** or changes in user behaviour.
  - Outputs can be inconsistent depending on input phrasing, requiring ongoing validation.

# Mitigation Strategies for Monitoring and Evaluation

- **Mitigation Strategies**:
  - Implement a robust observability stack, leveraging tools like **OpenTelemetry** for tracing and **Evidently AI** for drift detection.
  - Regularly conduct **A/B testing** of prompt variations and analyse user feedback.
  - Automate monitoring workflows using tools like **Grafana** for metrics visualisation and **Loki** for log aggregation.
  - Implement CI/CD pipelines for model updates, enabling seamless deployment and rollback mechanisms.

# Ethics and Bias

- **Risks of Misinformation**:
  - LLMs may produce **hallucinated content**—plausible-sounding but factually incorrect information.
- **Bias Amplification**:
  - LLMs trained on large datasets may inherit and amplify biases present in the data, leading to outputs that are discriminatory or offensive.
- **Accountability and Transparency**:
  - LLMs lack inherent explainability, making it difficult to trace the reasoning behind their outputs.
- **Mitigation Strategies**:
  - Regularly audit training datasets to identify and mitigate biases.
  - Implement post-processing filters to flag or correct harmful content.
  - Use **explainable AI (XAI)** techniques to make LLM decision-making more transparent.

# Legal and Regulatory Challenges

- **Data Privacy**:
  - Handling sensitive data raises concerns about compliance with regulations such as **GDPR** or **CCPA**.
  - Example: Ensuring that user data used for personalisation is not stored or misused.
- **Content Liability**:
  - Organisations may be held accountable for harmful or incorrect outputs generated by LLMs.
  - Example: A chatbot providing incorrect medical advice leading to harm.
- **Mitigation Strategies**:
  - Implement strong data governance policies to ensure compliance.
  - Use guardrails to restrict LLM outputs in high-stakes domains (e.g., finance, healthcare).

# User Expectations and Trust

- **Over-Reliance on LLMs**:
  - Users may place undue trust in LLMs, treating their outputs as authoritative without question.
  - Example: Believing an LLM's answer to a complex legal or medical query without validation.
- **Maintaining Engagement**:
  - If outputs are repetitive, irrelevant, or offensive, user engagement may decline.
- **Mitigation Strategies**:
  - Educate users about the limitations of LLMs to manage expectations.
  - Incorporate feedback loops to continuously improve user experience.

# Future Directions: Building the Next Generation of Data & AI Products

- **Hybrid Models**: How combining scoring engines and LLMs can produce both structured and unstructured insights (e.g., risk assessment with explanations).
- **Human-AI Interaction Design**: Emphasis on designing interfaces that effectively combine structured and unstructured outputs.