DataRobot

Real Generative AI Use Cases You Can Build Right Now

Explore the true power of generative AI with real use cases that you can build with DataRobot

Introduction

The generative AI market is changing quickly and rapid advancements are being made. Many underestimate the groundbreaking impact it will soon have on the enterprise. EY projects a 50% to 100% lift in productivity growth due to generative AI in the coming decade[1]. New use cases are being implemented every day. It's possible that the one use case which will transform your organization hasn't yet been devised. That's why it's important to stay on top of the innovations and actively "hunt" for that missing piece of the puzzle that might make all the difference for you.

But the paradox is that, even with this pace, it's still hard to build real-world generative AI solutions that deliver value. There aren't any AI teams out there that have "10 years of experience" with generative AI. According to a Google survey, more than half of responding executives said that their organizations lack the most critical skills to execute their AI strategy[2]. Yet the promise of generative AI is just too tempting, as 64% of those surveyed said they feel a high sense of urgency to adopt generative AI.

Many of DataRobot customers are already exploring generative AI with our platform, supported by a decade of our applied AI expertise. More than a few already have production-grade generative AI projects, poised to create value through efficiency gains, better response times, as well as improved consumer and business insights that drive business decisions.

This applied experience allows us to better understand technical, infrastructure, and organizational challenges, unique to generative AI and the associated risks that come with it.

All of these considerations can paralyze teams that are trying to identify, execute, and pilot near-term opportunities for value creation with generative AI. We're helping our customers overcome these challenges and mitigate risks every day to get their generative AI projects into production, where they can deliver real value to an organization.

Generative AI is the equivalent of going from an abacus to a Texas Instruments calculator. It eliminates tedium, supercharges productivity, and creates more space for creativity where teams fully embrace its potential. Those that do it early, will reap the biggest rewards.

That's why we enlisted our applied AI experts to put together this collection of practical generative AI use cases that can create real impact quickly for you and your organization. These are not "AI fairytales."
These are stories from the field where the impact is already happening.

You'll walk away with a better understanding of generative AI opportunities, their tangible business impact, as well as some of the technical considerations that can help you implement these use cases more confidently.

Common Generative AI Risks

AI hallucinations | Toxic prompts | Prompt injections | Data leaks | Hidden costs

[1] EY, "How tech disruptions can inform the economic impact of AI", Gregory Daco, 29 February 2024

[2] Google, "The Prompt: We asked business leaders what they're expecting from generative AI", 27 May, 2023

Before We Dive In

This complexity of generative AI requires a strategic framework to keep your development process aligned with desired goals and outcomes. It's important to eliminate guesswork while supporting quality assurance, cost management, and achievable use case ROI.

Strategic Considerations and Pitfalls to Avoid

RAG supplements the power of generative AI with information retrieved from supplemental internal or external databases to inform the LLM, leading to more accurate and up-to-date responses.	**Choose Between Pure LLM or RAG**. We find that Retrieval-Augmented Generation (RAG) is currently the most common approach to enterprise-grade generative AI. It's less expensive than fine-tuning an LLM, while still enabling hyper-specific outputs for any given use case. Yet, there are cost, data sourcing, security, scalability, and other issues that you need to consider.
Guard Models are typically represented by predictive models that are trained to monitor generative AI outputs, creating an additional "layer of defense" against incorrect outputs.	**Don't Blindly Trust the AI**. Consider a framework that will allow you to monitor generated responses to ensure their accuracy, prevent anomalous outputs, and improve performance through built-in feedback loops. Such a framework is usually a part of a broader Large language Model Operations (LLMOps) infrastructure.
LLM Playground (specific to DataRobot) is an environment that allows you to interact with LLM blueprints (LLMs and their associated settings), compare the responses of each to help determine use case fit.	**Shop Around. Rigorously**. It might be tempting to go with the LLM that's familiar to your team or the only one you've got access to. But this doesn't guarantee that it's the best choice for a particular use case and business problem. That's why it's important to test different LLMs to compare their responses. This also applies to system and user prompting strategies, "temperature" settings, embedding models, and chunking strategies - all of these different inputs can deliver meaningfully different responses.
LLM Safeguards may include, but are not limited to monitoring toxicity, readability, stop words, and personally identifiable information (PII) leaks.	**Implement Content Guardrails**. Be sure to include content safeguards related to competitor brand mentions, hate speech, and other unwanted content to avoid brand safety issues and problematic outputs. One of the important aspects to consider is how to automatically intervene to stop these types of data from being exposed or served to the user.
LLM Costs scale with usage patterns, thus creating budgetary risks that can directly impact the ROI and viability of your solution. Understanding "cost leakage" can help you optimize the solution, like updating the prompting strategy to minimize the amount of processed tokens.	**Create Cost Efficiencies**. If you run a generative AI chatbot, it might be getting the same types of questions over and over again. Instead of incurring a cost for generating the same answer every time, you may consider caching common answers to avoid increasing the usage rate of your generative AI solution. You can also block 'nonsense' or harmful responses before they happen or consider lower-cost components or LLMs.
Small-Scale, Internal use cases, with a limited number of users, are your best bet at the start of the journey.	**Start Small and Prove ROI Before Scaling**. When a small group of people is using your solution, the usage cost is small. Share it with a million customers, though, and those costs will soar. Make sure you understand which use cases are financially viable at your stage of the journey. Build the foundation before building the house.
Custom Metrics will be necessary to accurately assess the ROI, as generative AI often affects areas where there are no set KPIs. Without them, your understanding of actual value will be limited.	**Define Success Early**. From your ROI metrics to usage volume to costs, all stakeholders should be aligned in how your organization will evaluate the success of your use case. ROI is often tied to the complexity of the use case. Look for use cases that offer decent ROI, but aren't too complex for your first wins.

Common Types of Generative AI Use Cases

The potential applications of generative AI are virtually limitless. But most of them tend to fall into one or more of the following categories.

By Audience

Internal use cases can range from HR chatbots, to network security monitoring, to legal tools. Their impact is meant to stay within the company.

They offer limited risks (no exposure to users outside of the company) and are easier to control (access and usage controls), thus are a good place to start for any organization, no matter their AI prowess.

Externally-Facing use cases can range from automated communications to compliance tools.

Their impact extends beyond the organization, as employees share their outputs with external parties, like customers or vendors.

The risks go up with these use cases, as the generated answers are being shared outside of the organization, though still often have the opportunity to manual internal review.

External use cases are designed to directly interact with external parties, like customers or vendors (customer-facing chatbots, shopping assistants, support interfaces).

Given this, as well as the broadest audience, they carry even more risks (reputational, security, etc.). But they also represent potentially greater ROI due to their impact on revenue.

By Impact

Efficiency/Optimization use cases aim to increase productivity of existing teams and lower costs.

It's relatively easy to start calculating ROI for such use cases. For example, your sales organization knows that it takes them X amount of hours to respond to complete a specific task and how that might have changed with generative AI.

Growth use cases can open new revenue streams or drive more topline outcomes.

These use cases include increasing conversion rates and improving customer satisfaction.

But they are harder to quantify and measure, while also potentially exposing the organization to more risks, as these use cases are usually external in nature.

Strategic Sustainment use cases often don't fit the other two categories and revolve around long-term organizational goals that don't necessarily have a direct, easily measurable impact on productivity or the bottom line.

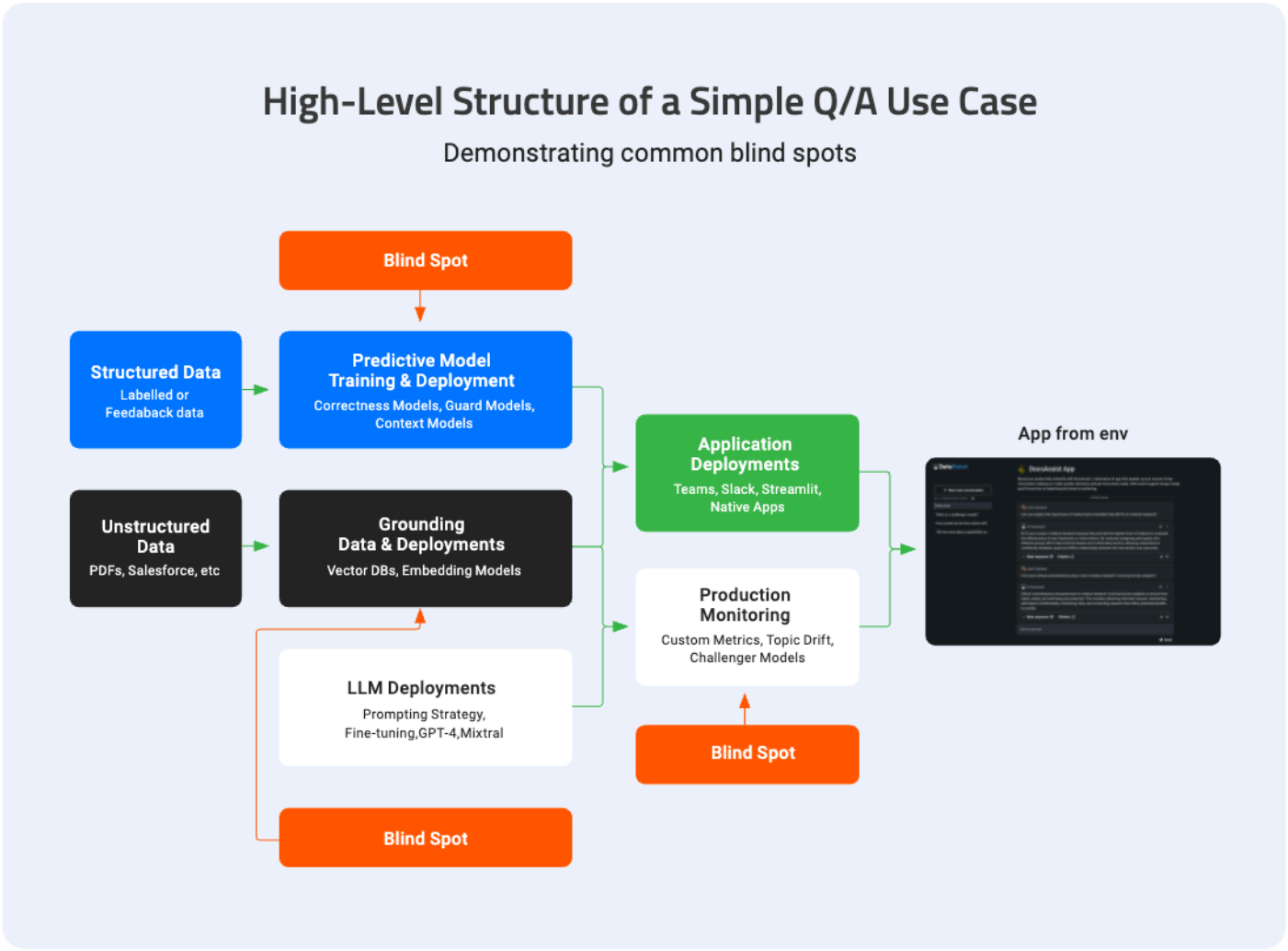
These use cases may revolve around sustainability, DEI initiatives, employee satisfaction, and more.

Expanding the Focus

Most organizations right now are myopically focused on the LLM (the model). This is where they invest most of their resources. But there are so many other components that are required to develop a production-grade generative AI solution.

We wanted to showcase where most organizations have a blind spot in these areas. By tackling these gaps and addressing the lifecycle more holistically, organizations can ensure long-term success with generative AI.

The diagram below represents a straightforward Q&A RAG use case, showing how many considerations go into creating a quality, enterprise-grade application even for a simple solution - and highlighting areas commonly being overlooked by organizations today. We're obfuscating a lot of the involved steps and operations, like all of the elements that go into the RAG custom model setup, to simplify it.



When you build out a use case, it's important to keep all of these components in mind. Generative AI, depending on the complexity of the use case, requires rigorous governance, monitoring, and fine-tuning. We tried to highlight these areas in the use cases, as well as offer some mitigation tactics for associated risks.

Addressing Potential Blind Spots

There are a few things that you will run into on your path to production-grade generative AI that you need to pay very close attention to in the process.

System Prompts

Invest resources into fine-tuning and setting up the system prompt, as it guides the generative AI model in tasks like transforming content into narratives, converting data-based requests into database queries, and generating responses in the right tone. Include specific guidance for formatting, structure, length, and style of responses. Ensure the prompt enables proper data manipulation and understanding by the LLM, while also considering potential cost implications of complex queries to avoid excessive token consumption and increased expenses.

Below is an example of the first iteration for the prompting approach in a use case around forecasting that you'll see in this ebook. The User Prompt section illustrates how intricate, detailed, and even confusing prompts can become for complex use cases. Figuring out how to obfuscate a lot of these steps as part of the system prompt and hide them behind the UI might be the difference between success and failure.

System Prompts

- You are a helpful assistant working to explain a machine learning model and its inputs.

User Prompt

- Our predictive model specializes in forecasting {ABC} for the {XYZ} market in {geo}.
- To unlock the intricate dynamics of the {XYZ} market, we've curated key insights into the top five influential features.
- (Insert columns for prediction explanations you'll be feeding the LLM)
- Now, let's embark on a journey to decode what these data values reveal about the broader {XYZ} landscape in {geo}.
- We invite you to transcend the numbers and offer profound interpretations.
- Explain not just what the data is, but what it signifies about the {XYZ} market in {geo}.
- Share insights that empower {XYZ} professionals to make informed decisions and navigate the unique characteristics of {geo}.
- First, I am going to provide you instructions to follow. Follow these numbered instructions exactly:
    - (Insert instructions for how to respond)
- Now, I am going to provide you examples to follow. Follow the format of these examples exactly:
    - (Insert examples for how to respond)
    - (Insert data for the prediction explanations - a dataframe converted to a string and injected into the prompt)
- Start now.

Databases and LLMs

Thoroughly test vector databases and LLMs, reviewing parameters like chunking, embedding, and temperature to ensure high-quality and efficient outputs. Practical testing may reveal superior performance of newer LLM versions like gpt-4-32k over older ones. Invest in establishing a review process to maintain the relevance of information in the vector database, particularly if your use case depends on up-to-date data.

Generative + Predictive

Combine generative and predictive AI as a powerful tool that bridges the gap between AI and users. Such an approach brings out the best of both worlds when you use a predictive model to make a prediction and then utilize generative AI to explain the context for that prediction. This can involve more upfront work to shape the problem into a dataset of previously defined examples and clear-cut prediction outputs, but can result in tangible outcomes and reduced hallucinations. For this, you'll need a predictive AI solution or framework that is able to provide context for its findings, "explaining" each prediction, row-by-row, which could then be served to the generative AI model for processing.

Data

Investigate your data. A well-structured and, if necessary, well-labeled database of internal documents is essential to facilitate knowledge retrieval for many generative AI use cases. They might also require rigorous data preprocessing to properly become the source of information for the LLM. Understanding data pipelines is also vital, as outputs often rely on multiple data sources that need to be combined for the solution to function effectively. Deep understanding of all enterprise databases, along with appropriately labeled datastores providing context for the LLM, is often essential for optimal performance. Do not discount the importance of data engineering for the success of generative AI.

Security

Consider the security implications that are crucial in finalizing the architecture of the solution, particularly given the potentially sensitive nature of the information involved. To minimize security risks, a locally hosted solution may be necessary. A less complex alternative might be using an LLM with enterprise-grade security protocols, like Azure OpenAI, Google Bison, and Amazon Bedrock. However, utilizing external LLMs via an API still has some potential to jeopardize highly sensitive data. Therefore, opting for a locally hosted LLM might be the preferred approach. Even for the testing phase. This ensures that data transmission remains internal to the organization.

Good understanding of governance and access workflows is also crucial, as it might be important to keep the information siloed, if it should only be accessible to a certain group of employees.

Processes and Workflow

Commit to the required organizational change to make generative AI successful. You'll need to build a clear and transparent process that includes important points, like how to use the generative AI solution or how to escalate interactions to a real person, if it's an externally-facing solution. Internal training might be required to help employees understand the system, its limitations, and associated risks. If the user doesn't know how to prompt it, interact with it, or gauge the answers, they may not be utilizing its full potential. Training can be expanded to executives and leadership to improve their familiarity with the new technology, its capabilities, as well as its limitations. To take it a step further, teams that build AI can also be enrolled into training programs to improve their understanding of what's available and let them test out new functionality and capabilities hands-on.

Have a clear vision about the internal workflows this solution will augment. Sometimes, a generally available app, accessible to everyone in the company, might not be the best immediate solution, as it may lead to higher costs or disrupt workflows if it doesn't work properly for everyone. Consider implementing it for a specific use case, for a specific team to "iron out the kinks."

Interface

Be flexible with how you see users interacting with the solution. The interface can make or break the solution, but it's vital that you don't overextend your team by pursuing an elaborate integration with your existing systems and tools. Sometimes, a standalone application can take you 90% there. And in other cases, a full integration with your CMS or ERP might be required to unlock the ROI. Sometimes, starting with a reasonably scoped MVP will also help you clear the initial hurdles before you tackle the broader solution. So it might be worth applying the crawl, walk, run methodology for more complex projects.

Oversight

As you'll see from many of the use cases, guard models are often essential to a secure and safe generative AI solution. Guard models that score every output for completeness, relevance, and confidence can aid the users, but they can also prevent anomalous outputs and hallucinations. For example, a guard model can serve as a guardrail that prevents outputs that mention an organization's competitors. It can also block unrelated, toxic, or malicious prompts that increase the costs of the solution and carry security risks.

They can also be used to improve performance through built-in feedback loops, where users can score provided answers and also edit them to improve outputs overtime. However, this might require a more complex workflow.

10+ Generative AI Use Cases You Can Implement Today

According to Jay Schuren, Chief Customer Officer at DataRobot, about 75 percent of all generative AI use cases that he sees organizations explore right now are efficiency plays, since these are lower risk internal use cases with easily defined ROI. Our list of use cases is representative of this, as most of them are focused on improving various internal processes, like procurement, legal, marketing, and sales.

This is also why most of these use cases are industry-agnostic, as they're focused on functions universal to practically any industry. We guarantee that you will find use cases immediately applicable to your business.

This list should help you break away from the all-too-common cycle of user behavior, where first, people overestimate generative AI, placing almost human-like expectations on it, which then leads into "frustration with the machine" as users discover the actual boundaries of the technology.

We talked to our applied AI experts, who work directly with customers, to give you a better understanding of the business impact, potential value, as well as some of the technical, infrastructure, and risk implications associated with each one of these generative AI use cases. We hope that this inspires you to expand your horizons by giving you a better understanding of what's possible and pushes you towards your own path to success with generative AI.

1 Save Time on Suspicious Activity Reporting (SAR)

FAST FACTS

Ease of implementation: Low
Value: High
Impact type: Efficiency/ Optimization

Primary users: Internal
Type: RAG, Summarization
Includes Predictive AI: Yes

What It Is

Financial organizations monitor suspicious activity as part of the industry's regulatory framework. The industry is already using predictive AI to improve accuracy of suspicious activity detection, but the suspicious activity reports (SARs) that have to be prepared for each incident still require a lot of scrupulous work. Fraud analysts spend a lot of time preparing these reports for delivery to regulators.

By combining the existing predictive AI workflows around suspicious activity monitoring with a generative AI component, the reporting process can be streamlined, improving the efficiency of Fraud/BSA/AML analysts.

How It Works

Prediction Explanations (a quantitative indicator of the effect variables have on the predictions) from the predictive model are fed to the generative AI model. This data is formatted as a JSON dictionary for the generative AI model to process. As such, there's no vector database that the generative AI model interacts with. The JSON file is the source of truth for the LLM. A system prompt in the background controls the format of the output, based on the predefined template for how the financial institution needs to format their reports. The simplified transformation of predictive insights to the narrative within the report looks like this: variable A exceeds the threshold of X, thus it indicates this transaction as fraudulent.

The analyst can then review the report and make necessary amendments, like elaborate on certain values that the predictive model highlighted, before passing the report along to the Financial Crimes Enforcement Network (FinCEN), which then routes it to the appropriate law enforcement agency for further investigation.

User Experience

The fraud investigator in this scenario would interact with a pre-built application which connects to their existing suspicious activity alerting system, retrieves known relevant context such as transaction history and check images, and generates output from a separate predictive model based on the results of the alerting system.

For each alert that has been flagged, the analyst would read a summary, created by generative AI, in natural language, detailing why an alert was tripped, its likelihood to truly be suspicious activity, and a compilation of pre-prepared relevant documents. Using human expertise and potentially seeking out additional information, the analyst reviews the case and makes the final determination on whether any given alert was correct or not. Once the analyst has confirmed the decision, the generative AI will construct the final narrative using all information given and format it into the predefined template that is consistent for the bank, explaining why a certain alert was truly suspicious or not.

Once the report is generated within the app, the user can review it, make amendments if needed, and then generate the file in the format necessary for the reporting workflow. This file is then submitted into a different system, tied into FinCEN.

Why It Might Benefit Your Business

A single suspicious activity report can take hours to write, review, and finalize, especially due to the variety of all the information that requires review. Large financial institutions receive tens of thousands of suspicious activity alerts every day.
So even a small improvement in how much time it might take to process one report, can have an incredibly powerful cumulative effect that can save tens of thousands of work hours. This also allows analysts to devote more of their time to actual investigation, not being pushed by the growing backlog of alerts to review, which improves their performance and minimizes human error. A reduction in processing time would also expedite approval for legitimate transactions.

Risks and Mitigation Tactics

Risks associated with this use case span both, generative and predictive AI components of the solution.

- Inaccurate flagging of transactions can result in inaccurate reports.
- Generated reports may misrepresent predictive data or draw incorrect conclusions.
- A poorly tuned system prompt can produce reports with unconventional wording and structure, requiring extensive manual amendments by analysts.

Baseline Mitigation Tactics

- Model monitoring for the predictive solution to ensure that it flags only the most relevant transactions.
- Extensive pre-production testing of the LLM, its parameters, like the system prompts or the temperature of responses.
- Consider a retraining regiment that uses grounding data to improve the model's outputs. This might require a new process, where the user can amend the automed report, which is then fed into a retraining database.

2 Streamline Request for Quote (RFQ) Processing

FAST FACTS

Ease of implementation: Medium
Value: High
Impact type: Efficiency/ Optimization, Growth

Primary users: Internal
Type: Labeling, Classification
Includes Predictive AI: No

What It Is

When an organization receives a request for a quote (RFQ) from a prospect, responsible teams (Sales, BDR, etc.) comb through the information in the RFQ to break down the request, match it with the right inventory, and then create an appropriately priced quote. The introduction of generative AI into this process eliminates a lot of the manual work necessary to match items in the RFQ and the internal SKU database.

How It Works

It's a multistep process, where the generative AI model first "cleans up" the data from the provided quote, transforming it into structured data that matches the internal database of the company, outputting a JSON "blob" for each line item in the quote. There's no vector database involved, as the model is called directly via the API and is asked to process the presented text.

The second step in the process is asking the model to match the extracted and structured items with the internal database that contains all of the product SKUs with all of their nomenclature and pricing information.

The output of the process is a breakdown of the quote, with matching items and their related pricing. Since there might be multiple similar SKUs within the same product categories, the user has the ability to manually select the correct one out of the list.

Simultaneously, a structured table is generated that includes all of the relevant product data, which the sales or business development user can review and then copy into an email or other format to include in their response to the RFQ.

All of this process is controlled by a system prompt and the only prompting elements that the user would be asked to perform would be to paste the original RFQ contents into the application and then to choose the most appropriate extracted SKU from the database that matches the original SKU in the RFQ.

User Experience

The user logins to the internal RFQ application (Streamlit, etc.). They paste the quote from the RFQ and run the application. The app processes the RFQ (multistep process described above) and generates the list of items from the RFQ with dropdowns underneath each item, which allow the user to select the most appropriate match from the database.

After all of the extracted RFQ items are matched to extracted products and pricing from the database, the user then runs the application again, which generates a structured table. They can copy the table and paste it to whatever system is needed to complete the RFQ and send it over to the prospect.

Why It Might Benefit Your Business

A single suspicious activity report can take hours to write, review, and finalize, especially due to the variety of all the information that requires review. Large financial institutions receive tens of thousands of suspicious activity alerts every day. So even a small improvement in how much time it might take to process one report, can have an incredibly powerful cumulative effect that can save tens of thousands of work hours. This also allows analysts to devote more of their time to actual investigation, not being pushed by the growing backlog of alerts to review, which improves their performance and minimizes human error. A reduction in processing time would also expedite approval for legitimate transactions.

Risks and Mitigation Tactics

- Inaccuracies in parsing the original quote may lead to incorrect model outputs, which will then cascade into errors in the product marching process. This problem scales with the complexity of the quote.
- A system prompt that's not fine-tuned may lead to inconsistent outputs.
- The specific LLM chosen for the use case by default may not be the best for parsing data. For example, OpenAI's API offers robust JSON mode support, but it's not guaranteed that it's the best for your use case or your prompting strategy.

Baseline Mitigation Tactics

- Extensive pre-production testing of the LLM, its parameters, like the system prompts.
- A guard model that monitors the generative AI output and detects and prevents erroneous SKU quotes from being added to the final quote.
- Monitoring custom metrics that focus on costs. You want to make sure the costs aren't ballooning when, for example, you start receiving more complex quotes that require more tokens for processing.
- Monitoring custom metrics around the generative AI output that focus on response drift. If your LLM keeps returning high amounts of quotes that the user needs to update manually, it might be time to review the process and the model itself to ensure its continued efficiency.

3 Transform Invoice Anomaly Detection and Processing

FAST FACTS

Ease of implementation: Medium
Value: Medium
Impact type: Efficiency/ Optimization

Primary users: Internal
Type: Summarization
Includes Predictive AI: Yes

What It Is

Manual invoice processing is costly, time-consuming, and prone to error. While predictive AI models can identify patterns and incongruencies in an organization's invoicing data, generative AI can augment the validation process to generate concise summaries of all detected anomalies and improve invoice approvals.

By crafting clear narratives around these invoice anomalies, generative AI summarization can help better communicate underlying causes and improve certainly around approval or rejection of any given invoice.

This is another example of generative and predictive AI working together to deliver new efficiencies for an organization.

How It Works

The data from the internal invoicing system, like SAP Concur, gets into the organization's database (information filled out by the employee submitting the invoice). This information is then fed into

a predictive model that utilizes unsupervised learning for anomaly detection by comparing every invoice against historical data from previously labeled invoices (training data, where invoices are categorized as "anomalous" and "non-anomalous").

Using the previously described Prediction Explanations mechanism, a generative AI model is instructed through a system prompt to summarize the predictions for any given transaction in a concise and human-readable format, which are then fed back into the original invoicing system where the analyst or the financial manager is able to review the findings and make the decision (reject or approve). The process augments invoicing by improving anomaly detection rates for invoices and explaining the anomalies to the people making the decisions. It eliminates a lot of the manual steps required to review each and every invoice, by automating most of the reasoning processes involved in the review process.

Two simple python files can easily orchestrate this integration through simple functions and hooks that will be executed each time an invoice requires a prediction and its consecutive analysis. The first file has the credentials to connect with the generative AI model and contains the prompt to summarize the explanations and insights derived from the predictive model. The second file easily orchestrates the whole predictive and generative pipeline through a few simple hooks.

User Experience

The end user can interact with the invoice anomaly detection solution via the front end user interface. They consume the generated insights within their invoicing system to make the final decision on each individual invoice. Everything in the backend is handled by the predictive model and the generative AI solution.

Why It Might Benefit Your Business

By automating anomaly detection organizations can accelerate invoice processing workflows, reduce the human capital required to handle manual invoice reviews and minimize disruptions created by invoicing errors. The additional benefit of this process is improved communication with external parties, like employees submitting invoices. Fewer legitimate invoices are being flagged due to the predictive AI pipeline, while more illegitimate ones get through the review process. Those that do get flagged are accompanied by an appropriate narrative explaining the organization's decision to reject the invoice.

Depending on the size of the organization and its invoicing backlog, the solution can save dozens, hundreds, and even thousands of hours on invoice processing, while also saving the organization money by detecting more anomalous invoices.

Risks and Mitigation Tactics

Risks associated with this use case span both, generative and predictive AI components of the solution.

- An inaccurately flagged invoice may lead to an incorrect decision by the user if the system prompt for the generative AI is not fine-tuned to appropriately explain the prediction. In this case, a bad prediction may be masked by a bad summarization with generative AI. The output may look convincing and the users may choose to just trust it to make the decision.
- A system prompt that's not fine-tuned may lead to unconventionally worded and structured summaries for invoices, which complicates the review and may also impact the user experience. For example, the system outputs an explanation for the prediction that's too long for the invoicing system to display.

Baseline Mitigation Tactics

- Custom metrics monitoring for the generative AI model that tracks text quality parameters, like readability and complexity.
- Extensive pre-production testing of the solution, such as feature selection, prompt engineering, and various LLMs. This also requires a human in the loop, i.e. the end user should be involved in evaluating the quality of various pipelines to identify the optimal solution. Since such solutions are integrated with existing invoicing tools, it's important to make sure that the output of the model fits the UI of the system.
- A retraining regiment that uses grounding data to improve the model's outputs. However, this requires a new process, where the analyst can amend the automed report, which is then fed into a vector database (new infrastructure).

4 Automatically Resolve Basic Customer Service Inquiries

FAST FACTS

Ease of implementation: Low
Value: Medium-High
Impact type: Growth

Primary users: External
Type: RAG
Includes Predictive AI: No

What It Is

High-volume customer service queries are a resource drain for organizations—and undeniably a source of friction in the customer experience. Customers today expect solutions to their problems and they expect them quickly. That's why many organizations have large customer care departments, as well as elaborate customer support portals that facilitate communications. But it's a constant race between an optimal size of the organization and the ability to serve customers. The existing rule-based chatbot solutions simply can't deal with a variety of potential requests, leaving customers dissatisfied and looking for more support.

If they don't get what they're looking for from this first-line chatbot, they'll need to talk to a real person (either online or over the phone) which is expensive if the rate of these requests is high. The more of the requests could be automated away through the chatbot, the fewer of them will require human assistance, thus lowering the overall costs of customer care.

Such a solution, trained on the organization's knowledgebase, can immediately respond to actual customer questions, automating hundreds and even thousands of ongoing conversations with customers. For example, in banking, a generative model can be trained to perform complex and personalized tasks such as checking account balances and facilitating transfers and transactions. Retrieval-augmented generation can also fetch technical information from manuals and other documentation to provide customers with detailed answers to nuanced product-related questions.

Given the potential sensitivity of information that this solution might need to access, some organizations may prefer using a locally-hosted open source LLM. This way, the organization won't need to expose any information externally by sending data to a proprietary model.

How It Works

Context-relevant documents, like user manuals, can be combined into a vector database, then combined with a structured database with relevant customer information, like transaction data, that the generative AI solution will use for lookup and querying. When the user prompts the system (for example, a banking client asking for transaction history), the LLM model queries the transaction data database by transforming the original language query into an SQL request. It can do that because the LLM is originally supplied with the metadata about the database, so it understands the context of the database and the information available there. With static data, like manuals, which live in the vector database, the LLM will use the original user prompts to instantly generate the response, based on that grounding data.

The information is then extracted from the database and combined with any relevant context from the manuals, which is then fed back into the LLM, along with the original query to produce a human-like response that goes beyond the capabilities of a simple rules-based chatbot.

User Experience

Depending on the industry, a customer (user) can engage the generative AI solution in many places and through a variety of interfaces. This could be a personal assistant (remember Clippy?) accessible after the user logs into the system or an open chatbot application, available on the organization's support forum. Ultimately, the organization would need to assess the right touchpoints in the customer journey where interfacing with a chatbot like this would make sense.

The interaction would be no different from engaging a chatbot or a support professional through a chat window, where the user could ask questions, receive clarifications, or place requests. The biggest difference would be the granularity, speed, and robustness of the response.

Why It Might Benefit Your Business

Generative chatbots can automate a wide range of customer service queries that currently require the assistance of a human representative and can't be dealt with by rules-based chatbot systems. A generative AI solution can massively reduce capital expenditures for customer support by satisfying more requests that would otherwise have to be handled by a real customer support representative. The solution can also reduce customer service wait times, potentially improving customer satisfaction (CSAT), and subsequently, retention.

Risks and Mitigation Tactics

- There are numerous risks associated with a customer-facing solution, from inaccurate or off-topic responses (hallucinations) to toxic responses, as well as outputs that divulge sensitive information that the customer shouldn't have access to.
- A system prompt that's not fine-tuned may lead to unconventionally worded and structured responses, which don't satisfy the customer. This includes the initial prompt that might query the database, as well as the final prompt that processes the query to deliver an answer.
- A prompt that's off-topic to the system, as users might be trying to use the assistant like a generic GPT tool. It might often be the byproduct of the user not understanding how to prompt the solution to get the most out of it.
- "Jailbreaking" might be an issue too, when someone submits a malicious prompt that might break the system or divulge sensitive information to the user.
- Ballooning costs based on user prompts and generated responses, especially since you can't predict the length of submitted customer prompts and their complexity.

Baseline Mitigation Tactics

- Custom metrics monitoring for toxicity, readability, and complexity to ensure that customer-facing responses are appropriate. This also extends to operational custom metrics, like tokens/ cost monitoring to ensure the financial viability of the solution.
- Built-in suggested prompts or auto-filling prompts as a way to guide the user and help them understand the solution and its capabilities.
- Guard models that prevent unwanted generative AI outputs, based on the response parameters or halt any potential malicious prompts.
- Extensive pre-production testing of the LLM, its parameters, like the system prompts.

5 Simplify Request for Proposal (RFP) Generation

FAST FACTS

Ease of implementation: Medium-High
Value: Medium
Impact type: Efficiency/ Optimization
Primary users: Internal
Type: RAG
Includes Predictive AI: No

What It Is

An RFP is a document that puts forth a proposal to potential vendors and suppliers, published by organizations interested in procuring a product or a service. RFPs are standard practice in any industry where technical products, contractual bids, and other large-scale selling efforts take place. The document often includes many detailed questions designed to determine if the potential product or service is up to the specifications. Thus, responding to RFPs is a time-consuming, but necessary process.

By sourcing information from internal documentation (manuals, user guides, enablement materials, public documentation portals, etc.) — including past RFP responses — businesses in any industry can use generative AI to fill out RFPs more efficiently or answer questions from prospects more readily. Over time, this generative approach can be used to strengthen and update responses for improved RFP results.

Currently, for many organizations, the RFP process is manual and tedious. The responsible party (proposal teams or sales) needs to scour through internal documents and manuals to finalize their answers. The process would often involve other internal stakeholders and channels (Slack, JIRA, etc.) where people request clarifications from internal SMEs. This pipeline is full of tedium and is prone to human error, especially when someone might be referencing an outdated document to generate their responses manually.

How It Works

With generative AI, this process could be augmented by using grounding data (internal knowledge, like documentation) to inform an LLM, so it could then quickly generate responses to RFP questions.

In this case, a vector database should be set up, with a specific collection of internal documents, as well as previous RFP responses. It's important to maintain the database and ensure that only the most relevant documents reside there. For example, an RFP response from 5 years ago might not be the best source material since your products or services have most likely changed over this period of time.

The LLM can query the vector database when generating the answer. In many cases, this can be a standard publicly available LLM, called via an API, since the information supplied in the RFP is not sensitive. But, for cases when your RFPs contain very sensitive information (government contractors, etc.), a locally-hosted open source LLM will be a more secure choice. However, it's worth reiterating that many enterprise LLMs such as Azure OpenAI provide enterprise security protocols that will keep your data/prompts/context private.

Additionally, the overall workflow may additionally include a predictive AI guard model (trained on labeled RFP responses from the past), designed to monitor outputs and deliver confidence scores for each response. The lower the confidence score, the more attention the user needs to pay to amending the answer. The solution, as a whole, doesn't require guard models to function though.

NOTE: the same approach can be used to **Requests for Information (RFIs)**, as it has the same basic principles, requirements, data sources, and other parameters in common. It is a good framework for many workflows that include processing incoming information requests.

Why It Might Benefit Your Business

Streamlined, accelerated RFP generation reduces the burden placed on all individuals and/or teams involved in the RFP process. If your organization has a sales-led RFP motion, then you will especially benefit from this use case. The saved hours can be used by the sales reps to focus on their main tasks (engaging prospects, moving deals through the pipeline, etc.).

Additionally, this puts less pressure on the SMEs involved in the RFP review process, as they don't have to respond to questions. They only need to review and vet the generated answers. All in all, it's a huge productivity booster for any organization, especially if it deals with a lot of RFPs and suffers from an RFP backlog. On top of it all, the potential updates of the vector database with the most successful RFPs, can further improve conversions and efficiency over time and directly impact revenue outcomes as the solution learns to deliver only the most relevant and actionable responses.

User Experience

When the organization receives an RPF, the responsible party goes to the RFP solution to upload the questions. This could be a simple app that has a text box where they can paste the questions and submit them as the user prompt for the LLM. The solution also can allow them to add PDF files (common for RFPs). The PDFs are transformed into text (Python OCR, etc.) and sent to the LLM as the prompt. An alternative solution would be to integrate the generative AI solution with their existing communications tools, like a bot in Slack or Microsoft Team. Just as easily as having an app, you could have this run as a script that adds answers into an Excel sheet or even an MS Word document, which may be the format a stakeholder is looking for.

Once the LLM, informed by the vector database, produces responses, the user can copy them into the appropriate interface if necessary and proceed with the usual workflow, like sending to appropriate stakeholders before replying to the RFQ.

Risks and Mitigation Tactics

- There are numerous risks associated with an externally-facing solution, from inaccurate or off-topic responses (hallucinations) to toxic responses, as well as outputs that divulge sensitive information that the potential client shouldn't have access to.
- A system prompt that's not fine-tuned may lead to unconventionally worded and structured responses, which don't satisfy the user, leading to a lot of manual edits.

Baseline Mitigation Tactics

- Custom metrics monitoring for toxicity, readability (Flesch Reading Score), as well as informative/ uninformative responses to ensure that externally-facing responses are appropriate. This also extends to operational custom metrics, like tokens/ cost monitoring to ensure the financial viability of the solution.
- Guard models that prevent unwanted generative AI outputs, based on the response parameters or halt any potentially inaccurate answers from being outputted to the user.
- Extensive pre-production testing of the LLM, its parameters, like the system prompts or the vector database. This limits unwanted consequences, like responses that misrepresent a product or a capability.
- The users can also rate the answers or edit them, which then feeds back into the original vector database for future reference. However, this might require additional workflow and setup steps.

6 Quickly Query or Summarize Any Document with Just-In-Time Retrieval (JITR)

FAST FACTS

Ease of implementation: High
Value: Medium
Impact type: Efficiency/ Optimization

Primary users: Internal
Type: Modified RAG
Includes Predictive AI: No

What It Is

Accessing, understanding, and retrieving information from documents is central to countless processes across various industries. Whether it's finance, healthcare, government contracting, or practically any other industry - reading and summarizing lengthy business reports, research papers, legal documents, and other text-based content is necessary for a wide range of business functions.

In most cases, reviewing such documents to extract important information is a manual process. Employees spend countless hours digesting, annotating, summarizing, and interpreting the information.

Generative AI can be used to expedite these processes by analyzing the documents. Imagine having a personal assistant who tirelessly reads any paper you give to them to answer questions as they pop into your head. The solution can also be set up to simply summarize content to accelerate its review and processing. This approach is called Just-In-Time Retrieval (JITR).

How It Works

The JITR framework allows users to supply files (.pdf, .docx, .txt, etc.) and leverage LLMs to answer user queries about the content of those files, almost immediately. It's a modification of the standard RAG framework, where the LLM is informed by a vector database to provide context-aware answers to the user. With JITR, the vector database is created at runtime, based on the document that the user provides, unlike with standard RAG, where the vector database is created in advance.

A JITR solution (fully contained application) accepts appropriate file formats. It then transforms the content of the file into a base64-encoded string, which, after a series of manipulations, is turned into a small vectorstore, which can then be processed as a RAG workflow. This solution can work with multiple content types and process information in any language.. You can learn more about the technical side of JITR by reviewing the JITR AI Accelerator that contains the notebook with all of the necessary code to create a JITR Bot application that's ready for deployment. Since the deployment is done through an endpoint, the solution can be integrated into almost any application by just having the application make a simple API call.

User Experience

The process for interacting with a JITR Bot is straightforward, as all of the legwork is done on the backend. A user would access the application to interact with the bot. It could be an app in Slack, a standalone application, or any of the dozens of other available options.

The interface allows uploading a file in a specified format, along with associated questions, just like if you were sending a standard Slack message. After a few seconds (time frame depends on the size of the file), the bot notifies the user that the file was processed, while also providing answers to the questions. The user can then send follow up queries in the same thread to receive additional information.

For example:

> What were the Q3 earnings of Acme Inc.?

> What is this document about?

> Summarize this document

Why It Might Benefit Your Business

JITR is applicable to practically any industry and any department. Every organization consumes or deals with a variety of documents in one form or another. Often, retrieval of information for employees and teams can be time consuming. Utilization of JITR improves employee efficiency by reducing the review time of lengthy documents and providing instant and accurate answers to their questions.

This represents thousands upon thousands of saved working hours for legal, procurement, marketing, sales, and other teams, which directly and tangibly impacts the velocity of business operations. Especially if the organization deals with large documents.

Summarization also allows your business to expand its collective knowledge base by reading, interpreting, and summarizing a larger body of text-based information than what is manually feasible for your existing workforce.

But another key benefit is the democratization of generative AI. Many organizations may not have resources and expertise in software development to develop tools that utilize LLMs in their workflow. JITR enables teams and departments that are not fluent in Python to convert a text file into a vector database as context for an LLM. No knowledge of LLMs, Natural Language Processing (NLP), or vector databases is required.

Risks and Mitigation Tactics

- Risks associated with the generated output: potential toxicity or readability issues associated with the prompt, as well as any potential cost implications, based on the scale of the user base and the complexity of inputs and outputs. Additionally, there needs to be specific guidance around length and style of responses.
- Security risks for cases when the solution is used to process internal documents via an externally deployed LLM.

Baseline Mitigation Tactics

- Custom metrics monitoring for toxicity, readability (Flesch Reading Score), as well as informative/ uninformative responses to ensure that responses are appropriate. This also extends to operational custom metrics, like tokens/ cost monitoring to ensure the financial viability of the solution.

7 Accelerate Data Analysis and Reporting

FAST FACTS

Ease of implementation: Low
Value: Medium
Impact type: Efficiency/ Optimization

Primary users: Internal
Type: RAG, Summarization
Includes Predictive AI: No

What It Is

Internal analysis is an important aspect for almost any organization. For example, marketing teams routinely analyze past campaign performance to demonstrate ROI to internal customers and stakeholders. Banks and other financial institutions rely on financial performance analysis, and so on. Similar processes are often externally-facing, where organizations would need to analyze and share the performance of their products and services with their clients. All of these workflows are underpinned by laborious processes, where various analysts and managers would comb through performance metrics, uncover patterns, and communicate all of this information.

Generative AI can accelerate and enhance such analytical processes. It can be used to automate reporting and break down performance for external clients and internal stakeholders. At the same time, analysts can use LLM-powered chatbots to create additional queries based on such automated reports, quickly dig into the data to draw comparisons, and identify trends without manually going through all of the information. It's a multifaceted approach that involves summarization for automated reporting and then RAG for the chatbot element.

How It Works

In this case, the first step is preprocessing and appropriately labeling the data that will go into the vector database. This could be data from your CMS, your marketing performance tools, and other destinations, but the important aspect is making sure you provide enough context for the LLM. The data is then structured in what could be described as a JSON dictionary, embedded, and stored in the vector database. Alternatively, you can set up an agent to connect to an SQL database. This agent will be able to understand natural language questions and transform them into appropriate SQL queries, given relevant metadata about the underlying SQL database structure and columns.

From there, the LLM retrieves the information, according to its setup and system prompts, based on the desired output. For example, it can generate a weekly summary report around marketing performance, which then gets sent to a user's email. The critical component here is the system

prompt that delivers appropriate summarization of the data, related findings, and fits into the reporting structure that's useful for the end user.

The complementary step in this process is the interaction with a chatbot that is using the same vector database with the same data and the same LLM to respond to custom queries related to the data. This allows the user to get more intimate with the data, uncover additional insights, or build new narratives around the data for final consumers of the report.

Why It Might Benefit Your Business

This solution allows a variety of professionals to generate and deliver necessary reports quickly, especially those that contain time-sensitive data. An organization can cut out a large portion of manual reporting tasks, allowing analysts and managers to invest more time into evaluating and understanding the data to uncover deep, actionable insights.

An evolution of this solution could include a predictive model, integrated to deliver a variety of other reporting capabilities, like "what if" scenarios and more.

Overall, this generative AI workflow creates additional efficiency gains for anyone accessing and analyzing a variety of reporting data. This directly influences the velocity of business decisions, while saving, potentially, thousands of hours for those employees directly involved in the reporting process.

User Experience

Depending on the exact setup, reporting and chatbot capabilities can be integrated into practically any major business tool. For example, the report could be delivered via a Slack application/ channel and the user can ask follow up questions in the same thread. This could also be a standalone application that allows the user to browse through the library or previous reports and ask questions about them or an integration with existing tools, like PowerBI and Tableau.

Since this solution has broad applications, the users might be asking a variety of questions.

For example:

> How did the campaign performance change since last month?

> Which production KPIs saw the highest growth since last year?

> Provide an overview of our investment portfolio's performance in Q2.

Risks and Mitigation Tactics

- Risks associated with the generated output: potential toxicity or readability issues associated with the prompt, as well as any potential cost implications, based on the scale of the user base and the complexity of inputs and outputs. Additionally, there needs to be specific guidance around length and style of responses.
- Security risks for cases when the solution is used to process internal documents via an externally deployed LLM.
- Accuracy of responses will also be important, as users may be in the position to make bad decisions based on inaccurate generative AI outputs.

Baseline Mitigation Tactics

- Custom metrics monitoring for toxicity, readability (Flesch Reading Score), as well as informative/ uninformative responses to ensure that responses are appropriate. This also extends to operational custom metrics, like tokens/ cost monitoring to ensure the financial viability of the solution.
- Guard models that would prevent unwanted or unrelated outputs and permit the users to rate the outputs. This ensures that the final reports don't include any inaccurate/ hallucinated answers by the LLM and that the outputs can be improved over time with feedback from the users.
- Extensive testing of the LLM, its parameters, like the system prompts or the vector database.

8 Transform Forecasting Insights with Generative AI

FAST FACTS

Ease of implementation: Medium
Value: High
Impact type: Efficiency/ Optimization
Primary users: Internal
Type: Advanced Summarization
Includes Predictive AI: Yes

What It Is

In many industries, forecasts are still often created in spreadsheets for the sake of transparency and ease of forecast model interpretation. But these spreadsheet forecasts require a lot of manual effort, are resistant to backtesting, and are often difficult for non-technical stakeholders to understand.

For these reasons, many organizations are already utilizing machine learning for time series forecasting. However, one of the problems with these predictive models persists. It still may be hard for non-technical or business stakeholders without a quantitative background to interpret these insights, even with advanced explainability that predictive AI can provide.

Generative AI can bridge the gap by conveying predictions and methodologies for non-technical stakeholders, elevating visibility of this data and the underlying data science work across the organization. The solution would process the contextual information, as well as quantitative prediction insights, explaining the key drivers of these forecasts in human language and even learning to interpret and explain the underlying dynamics of the market at hand based on the data. This creates a seemingly all-knowing human-like assistant that's able to back its decisions with the highest degree of quantitative data possible.

How It Works

For this solution, prediction explanations (a quantitative indicator of the effect variables have on the predictions) from the forecasting model are fed to the generative AI model. Post-processed prediction data with prediction explanations for every time series is ingested and stored in a .csv format. Then it's converted to a string when injected into the LLM prompt. But could easily also be stored in a database table and converted to a string later.

Organizations may need to post-process the data and do aggregations up to the series-level in order to make it easier for the LLM to understand the prediction explanations, as the individual row level of predictions might be too granular, the series-level is easier to understand and work with. Since the generative AI model and its explanations are only as powerful as the underlying forecast, it's important to use as many features in the underlying forecasting model as possible.

Once the data is fed to the LLM, it summarizes prediction explanations into powerful narratives through an intricate prompting strategy.

User Experience

The user can interact with the model through a number of ways, depending on how the solution is set up. It can be a standalone application that has access to different forecasting reports from the predictive model, with all of the data already pre-processed for the LLM. This application can also include prompting templates for the user to choose from and modify if necessary since, as you can see above, the prompting strategy can be complicated, depending on the forecasting needs.

The organization may also choose to obfuscate some of the complicated prompting details by offering the user the ability to select necessary prompt elements via a dropdown, like the specific geos they might be interested in. This will then be automatically added to the final prompt. Once they've set up their prompt, they run the application and receive the final report within the application (text field, .pdf, or other formats).

Why It Might Benefit Your Business

Extremely powerful forecasting models built with predictive AI and backed by transparent explanations built with generative AI are extremely easy to understand. This improved decision-making processes by delivering reliable and understandable forecasts, which can have a multiplying positive effect on long-term business decisions, like investment choices and resource allocation. Getting the powerful predictive AI insights into the hands of consumers who otherwise would have been able to make decisions based on them can become a force multiplier for an organization.

Such a comprehensive solution also Increases efficiency and productivity of analytical teams by augmenting and automating forecasting processes. These teams spend a lot of time interpreting the data, but a significant investment is also made in storytelling to explain these findings to decision makers. Generative AI can simplify this process, while simultaneously improving the robustness of insights.

As a unified generative and predictive AI workflow, this can be a visible competitive advantage through improved velocity and accuracy of forecasting insights, as well as their transparency.

What You Need To Implement This Use Case Successfully

- A robust time series forecasting solution or framework that is able to provide context for its findings, "explaining" each prediction, row-by-row.
- An elaborate post-processing pipeline for getting the predictive insights into the LLM workflow.
- A solid prompting strategy that allows to shape the LLMs outputs in a digestible and useful manner.

Risks and Mitigation Tactics

- Risks associated with the generated output: potential toxicity or readability issues associated with the prompt, as well as any potential cost implications, based on the scale of the user base and the complexity of inputs and outputs.
- Data quality issues on the predictive side of the workflow, such as inaccurate or incomplete data, which can impact the accuracy and reliability of the predictions and lead to downstream effects for generative AI outputs.

Baseline Mitigation Tactics

- Custom metrics monitoring for toxicity, readability (Flesch Reading Score), as well as informative/ uninformative responses to ensure that responses are appropriate. Additional tokens/ cost monitoring ensures the financial viability of the solution.
- Guard models that would prevent unwanted or unrelated outputs and ensure that the final answers don't include any hallucinated answers by the LLM
- Extensive testing of the LLM, its parameters, like the system prompts.
- Ongoing monitoring of the underlying predictive model that supplies the forecasts (accuracy, data drift, etc.)

9 Quickly Generate Legal and Compliance Answers

FAST FACTS

Ease of implementation: Medium
Value: High
Impact type: Efficiency/ Optimization
Primary users: Internal
Type: RAG
Includes Predictive AI: No

What It Is

It's hard to find more document-intensive processes than legal and compliance. Finding answers to complex questions about the law and policies can involve a lot of time spent combing through dense and complicated documents. Any uncertainty here can halt operational and business processes. This kind of work involves highly paid professionals, which additionally increases the costs, as such processes take up a lot of their valuable time.

Generative AI can address this pain point for legal and compliance professionals to scour documentation and find answers to pressing questions, in the exact context that these professionals require. After retrieving these relevant chunks of information, generative AI constructs and delivers a legible answer that the user can utilize in their decision making. The added benefit of this automation is that the LLM can uncover additional insights from those documents, things that a person "grinding" through the documents might miss.

How It Works

The process involves feeding legal and compliance documents into a vector database, which is then utilized by the LLM to retrieve information for the user, chatbot-style. Most organizations already have stores of legal documents, in places like Microsoft SharePoint, which can be used as the source of the information. An important part of the process is ensuring that all of the possible LLM and vector database parameters are tested thoroughly before deployment, given the specific nature of "legalese." Things like chunking and embedding strategies need to be reviewed rigorously.

In many RAG cases, a standard publicly available LLM, called via an API, could work. But since the information for legal and compliance purposes can be highly sensitive, a locally-hosted open source LLM will be a better, more secure choice.

The overall workflow will benefit from a predictive AI guard model, designed to monitor outputs and deliver confidence scores for each response, while also blocking unwanted, hallucinated outputs. The lower the confidence score, the more attention the user needs to pay to the answer. The users can also rate the answers or edit them, which then can be sent back to the original vector database for future reference.

User Experience

There are multiple ways of approaching the deployment of this solution, but the most common one is implementing the chatbot directly in the standard corporate communications environment, like Microsoft Teams.

The user would open the chat window and start asking questions, since the vector database already stores most of the necessary legal documents.

For example:

> What are the liability rules in the EU AI Act?

> What are the rules around filing of civil appeals in the appellate court of {insert_state}?

For this to work seamlessly, an additional internal data pipeline could be built to ensure that new documents could be added to the database quickly. For example, an automated solution that scans the location of legal files and automatically adds new ones to the vector database.

The responses come back according to the given system prompt settings (format, length, etc.). An important addition to the output here would be to automatically ask the model to link to specific source documents that it's referencing. This increases trust and streamlines the user review process even more.

Why It Might Benefit Your Business

Organizations spend a lot of resources or, even, pay a high hourly rate to send legal and compliance professionals searching through large libraries of information. Generative AI chatbots can significantly reduce the time and labor it takes to find relevant information, realizing cost savings while freeing up those professionals to focus on more important work.

This approach also reduces the risk of information being overlooked, resulting in faster, more comprehensive answers to the most pressing legal and compliance questions.

Risks and Mitigation Tactics

- There are numerous risks associated with this solution, given the sensitive nature of the information that's being processed and outputted. From inaccurate or off-topic responses (hallucinations) to toxic responses, as well as outputs that divulge sensitive information that other stakeholders shouldn't have access to.
- A system prompt that's not fine-tuned may lead to unconventionally worded and structured responses, which don't satisfy the user, leading to a lot of manual edits or direct and potentially costly errors.

Baseline Mitigation Tactics

- Custom metrics monitoring for toxicity, readability (Flesch Reading Score), as well as informative/ uninformative responses to ensure that the responses are appropriate. This also extends to operational custom metrics, like tokens/ cost monitoring to ensure the financial viability of the solution.
- Guard models that prevent unwanted generative AI outputs, based on the response parameters or halt any potentially inaccurate answers from being outputted to the user. Accuracy is paramount when legal language is involved and guard models can ensure that. This is also important to ensure that the users are utilizing the tool appropriately, as the guardrails in place can prevent users from sending irrelevant prompts, thus ballooning the costs of the solution.
- A feedback loop by which the user can rate the generated response and edit, if necessary. The edited version then gets added back to the vector database to inform future responses, thus improving the system as time goes on.
- Extensive testing of the LLM, its parameters, like the system prompts or the vector database to ensure that responses require minimal oversight and don't lead to answers that misrepresent legal or compliance norms.

10 LLM-Enhanced Smart Clustering

FAST FACTS

Ease of implementation: High
Value: Medium
Impact type: Growth

Primary users: Internal
Type: Clustering
Includes Predictive AI: Yes

What It Is

Clustering is an unsupervised machine learning method that groups similar data points together into their own segments, or "clusters".. Clustering has a wide range of applications, as you don't provide a target to be predicted; rather, you let the algorithm find cluster labels for you. From finding new groups of customers and identifying new product complaint types to sentiment analysis and anomaly detection, clustering is a powerful tool. But standard approaches to clustering have their own limitations.

One of the toughest parts of the workflow is explaining clusters to end users. In most scenarios, the users building the models might not have the subject matter expertise to tailor the cluster labels toward the users consuming the models. For example, a data scientist in healthcare will not have the same level of expertise around certain diseases to properly understand how and why they're being clustered in a certain way. Classic machine learning is only capable of providing simple, nondescript labels like "Cluster 1", "Cluster 2", "Cluster 3", and so on. The process of manually digging into the data points in each cluster, identifying specific cluster themes and deriving the corresponding human-intelligible cluster labels is time-consuming and complicated. It may leave many potentially useful insights untapped.

How It Works

Generative AI models have been trained on vast amounts of domain and business datasets, so they can, with some clever prompt engineering, understand and automatically label the clusters tuned for end user expertise. This specific approach utilizes the unique cluster insights available from any DataRobot clustering model as additional context for the LLM.

Once clusters are created, cluster insights are packaged into cluster summaries to be sent to the LLM, which is then asked to label the clusters. Below is the system prompt for the LLM, asking it to generate cluster labels. The results are then fed back into the original system of record to update cluster labels.

```
def get_cluster_names(cluster_info, label_types="human friendly and descriptive"):
    prompt = (
        'You are a business analyst. You have run a clustering model and the following text in double quotes shows the cluster level values."'
        + cluster_info
        + """. Please provide "
        + label_types
        + " cluster names for each cluster. Output format is json with fields cluster description, cluster name, cluster id."
    )
    response = get_completion(prompt)
    return prompt, response
```

User Experience

This whole process could be automated away with a few API calls to the point that when clustering is complete, in a matter of seconds, the cluster labels and descriptions are being updated in the original UI of your enterprise AI system of record. The end user only needs to run the clustering job in DataRobot.

Why It Might Benefit Your Business

LLM-enhanced clustering is beneficial in cases where the users building the models lack the subject matter expertise and/or lack the time to create labels that are useful to consumers of these models.

This approach can be useful across many industries and applications, from ecommerce to marketing audience segmentation to sports science. LLM-enhanced labeling introduces more speed and efficiency for clustering projects, allowing to simplify the communication of outcomes to ultimate consumers of this information.

Risks and Mitigation Tactics

- The primary risk in most clustering projects is that the clusters themselves will not be useful or actionable. Because we are not training to predict any target variable of interest, the cluster segments may not represent a meaningful segmentation of the data.
- Another set of risks associated with this cluster labeling approach revolves around the cost. Users may opt for a more involved process that first queries an LLM to pre-summarize every row of the dataset.

Baseline Mitigation Tactics

- Additional feature engineering–i.e., creating more features to help the clustering algorithm learn what makes the data points different–is likely to help ensure that clustering segments are useful and well defined. As alluded to above, one such approach may be to loop over every row in the dataset and prompt an LLM to take a first pass at summarized themes. These themes can be fed into the clustering algorithm as an additional feature.

Risks and Mitigation Tactics

While this can provide more meaningful clustering segments in the end, it can also dramatically balloon the cost of tokens and the time required to run this approach.

Baseline Mitigation Tactics

- Because the straightforward approach to cluster modeling and smart cluster labeling can be performed very quickly, with minimal LLM token costs, users can quickly review and iterate on clustering schemes and their corresponding smart labels before investing a lot of time and money into extensive feature engineering. In the case where users want to make an LLM call for every row to engineer LLM summaries for clustering, users should consider testing that pipeline on a smaller sample of the data to avoid iteration costs.

11 Improve Service with a Field Technician Chatbot

FAST FACTS

Ease of implementation: Medium
Value: Medium
Impact type: Efficiency/ Optimization

Primary users: Internal
Type: RAG
Includes Predictive AI: No

What It Is

When field technicians arrive on-site to service products, they often have little context regarding the issue and how to fix it. Modern hardware is complex and diverse. In such scenarios, the technicians would often refer to manuals and other internal documentation to get guidance for the repair process. This makes it difficult for field technicians to efficiently solve the problem, as the internal information might be represented by a large set of documents that are hard to query and make sense of.

Multiply that by the complexity of the specific product lineup and you might end up in a situation where the technician spends hours just to find the right information, like the relevant error code or a specific troubleshooting routine. Only the most experienced technicians are able to complete such tasks with relative efficiency, but that's also not guaranteed.

Generative AI can be used to power a "field technician assistant" that leverages the whole pool of different text-based manuals and product-related documents to help technicians find the information they need to quickly and accurately fix the issue.

How It Works

In this case, all of the internal text-based documentation is transferred to a vector database, which is then used as context by the LLM to respond to technician queries. Since this solution mostly uses documentation that, in most cases, is externally available, this solution uses an API to send queries to an external LLM.

However, if the specific application involves transferring sensitive information (military or dual-purpose hardware, specific patented know-how, etc.), it's better to set up the solution with security in mind. It would then be based on a locally-hosted LLM that the organization has full control over.

One of the crucial elements in this solution is the ability of the LLM to properly reference the various tables available in the documents. Testing is important to ensure that the solution doesn't confuse various error codes and properly understands the contents of the tables. Additional steps might be required to append table headers before they're being fed into the vector database, so that the LLM would be able to better understand the context of each column in the table.

User Experience

The user would be able to access the solution in a number of ways, like a company-issued portable device with the application installed or a secure web portal through any sort of authenticated mobile device.

The user is able to directly ask questions about specific products and service-related workflows.

For example:

> How do I troubleshoot the hydraulic faults on {product name], {model number}?

> What is the error associated with error code #000D1 on {product name], {model number}?

> Show me step-by-step instructions for fixing the cooling system on {product name], {model number}.

The LLM processes the query and generates a response within the same interface.

The solution can also be set up to append the answers with links to related documents, as well as confidence scores that will allow the technical personnel to gauge the trustworthiness of the responses.

Why It Might Benefit Your Business

Slow field servicing reduces the daily productivity of each technician and creates longer servicing wait times that often frustrate device owners. Additionally, in any industry suffering from labor shortages and turnover, the solution would help organizations decrease the amount of time that a new technician needs to be mentored by a senior technician, then sent out on their own, with this assistant helping them retain the information that they learned during their initial apprenticeship.

The field technician chatbot can help accelerate servicing and improve the quality of fieldwork by making it easier for technicians to quickly service devices due to easy access to the correct information. It also improves performance of newer employees in these roles, as the solution helps them to accelerate their knowledge retrieval. This can significantly reduce ramp up times and thus affect the efficiency of the whole service and maintenance pipeline.

Risks and Mitigation Tactics

- Inaccurate or off-topic responses (hallucinations), as well as toxic responses may derail the servicing and repair process.

Baseline Mitigation Tactics

- Custom metrics monitoring for toxicity, readability (Flesch Reading Score), as well as informative/ uninformative responses to ensure that the responses are appropriate.
- Guard models that prevent unwanted generative AI outputs, based on the response parameters or halt any potentially inaccurate answers from being outputted to the user, which is important in a technical environment The guardrails can prevent users from sending irrelevant prompts, thus ballooning the costs of the solution.
- A feedback loop by which the user can rate the generated response and edit them, if necessary. The edited version then gets added back to the vector database to inform future responses, thus improving the system as time goes on.
- Extensive pre-production testing of the LLM, its parameters, like the system prompts or the vector database to minimize unwanted consequences, like answers that lead to technical complications or even accidents.

12 Democratise Access to Internal Datastores with a Query Chatbot

FAST FACTS

Ease of implementation: High
Value: High
Impact type: Efficiency/ Optimization

Primary users: Internal
Type: RAG
Includes Predictive AI: No

What It Is

Access to data isn't always straightforward in organizations. Some of it is available through dashboards and various other reporting interfaces. But rather frequently, the various non-technical stakeholders might require access to information that's stored in one or many of the internal databases. Many organizations have processes in place to retrieve such information. For example, ticketing systems or portals that allow users to submit requests, which are then processed by data experts, like data engineers. This process might have different forms, but often it would entail a data professional translating the request into a database query and retrieving the information.

It can be a cumbersome and slow process, which can slow down workflow and impact business decisions. The larger the organization, the bigger the problem may be, as these requests pile up in backlogs of SalesOps, MarketingOps, Product Insights, and other teams tasked with retrieving the information. Generative AI can be particularly good at translating natural language data requests into database queries and then retrieve that data from its storage location.

How It Works

This process can be highly reproducible, as the LLM is interacting with specific data and the results are much more consistent, unlike with text outputs that might slightly change every time they're generated. It's not a classic RAG problem, where an LLM uses a vector database as a source. Instead, it queries the database directly and uses available database metadata (descriptors, column names, etc.) to match the data with the original query in plain text.

In this case, an LLM translates the plain English request into an SQL query for proper execution, executes the query, and outputs the data with a high degree of accuracy and consistency. The core of the solution is a complex system prompt that includes a description of the schema for the data source, as well as other instructions on how to navigate the data. But once all of that is mapped out and fine-tuned, the solution can execute extremely complex SQL queries, processing multiple tables, joining them, etc.

User Experience

The user of this solution would be able to access the interface through a variety of solutions, but the most flexible one is a standalone internal application that provides a space to submit queries and receive generated answers, based on the structure dictated by the system prompt.

The user can select data sources from a list or, as an additional step, can have the solution suggest the best data source for the query first. For the latter option, the databases available would have to include extensive metadata to describe the data that's being stored there.

Since the solution processes natural language, there are multiple intuitive ways to query the database.

For example:

> How many customers do we have who defaulted on a loan in 2023?

> How many existing customers are at risk of churning?

> How many employees login to the system every day?

The questions depend on the nature of the data.
The system would then output the answer, in natural language.

For example:

> We have 147 customers who defaulted on a loan in 2023

Why It Might Benefit Your Business

A lack of access to self-service data creates unnecessary workloads for data professionals and vendors. It can also impact business operations when non-technical personnel, such as lawyers, marketers, and sales professionals, are held up waiting for their requests to be processed.

This solution also accelerates and democratizes access to meaningful business insights. Data is at the core of any modern business in any industry. Its availability and interpretability are vital to decision making. This solution unlocks these insights and allows less data- and tech-savvy employees to access important insights on their own.

Risks and Mitigation Tactics

- A system prompt that's not fine-tuned may lead to unconventionally worded, inaccurate, or "expensive" responses that require too many tokens to process. This is important because the original plain English input may be a complex request represented by a complicated SQL query that requires a very high number of tokens the LLM needs to process on the intake side.
- Data access requirements have to be well understood and taken into account. There might be data management processes that require additional people in the process to safeguard the overall security of the database. For example, EMEA employees shouldn't have access to the database that contains US Federal clients' data, available only to US employees with appropriate clearance. Those EMEA employees should not be able to query the database for that information. Thus, the initial access permissions and limitations have to be carefully thought through.

Baseline Mitigation Tactics

- Extensive testing of the LLM, its parameters, like the system prompts. A solution in this case may be to test the LLM on a small sample of questions that you also have definitive responses to, so you could gauge the output, its format, and accuracy.
- Elaborate and comprehensive workflow and access controls to ensure that the queryable data is appropriately segregated based on employees' data access privileges.

Embrace Experimentation on The Road to Value-Driven Generative AI

The use cases that you see here are coming directly from our applied AI experts, who are working with customers all over the world to bring their generative AI vision into reality. It's imperative to move from theory to practice now to secure long-term success with generative AI.

Every organization is unique, so it's imperative to tackle your unique roadblocks on the path of generative AI. You might not even know them until you actually start, just like you might not know about potholes on a road until you actually drive it. Organizations that we work with are already seeing business impact, but what preceded that was, often, months of scrupulous planning, ideation, and technical work. The longer you wait, the bigger the schism is going to be between your organization and its competitors.

Generative AI represents a new frontier of value creation for innovative organizations. Given the right infrastructure and tools, collaborative experimentation can quickly lead to powerful use cases that generate new value with speed, and at scale.

The DataRobot AI Platform provides all of the tools and workflows you need to effectively build, operate and govern both generative and predictive AI to enable new capabilities and efficiencies within your organization. This includes the DataRobot Generative AI Playground that offers everything you need to visualize, iterate, and experiment with different generative AI applications.

For up-and-coming generative AI adopters, our Generative AI Catalyst Program lets you collaborate with our AI experts to align on a clear roadmap, build lasting skills to scale independently with generative AI, and accelerate delivery of high priority generative AI use cases.