Generative AI Predictive Maintenance Applications

Using generative AI to achieve maintenance excellence

June 2024

MongoDB.

Table of Contents

Abstract

In manufacturing operations, unanticipated equipment breakdown can result in line stoppage and substantial throughput losses, potentially leading to millions of dollars in revenue loss. To avoid such losses, the manufacturing industry strives to implement predictive maintenance excellence programs. Generative AI can play a significant role in creating an optimal maintenance strategy for manufacturers.

This white paper highlights the benefits of predictive maintenance, including increased productivity, reduced downtime, and lower maintenance costs (as shown with a real-life example). It then digs into applying AI to the four stages of predictive maintenance:

1. **Machine prioritization:** This stage prioritizes machines for the maintenance excellence program using a retrieval-augmented generation (RAG) system that takes in structured and unstructured data related to maintenance cost and past failures.
2. **Failure prediction:** Machine sensor data is analyzed to predict machine failures using MongoDB Atlas Stream Processing and machine learning models.
3. **Repair plan generation:** This stage generates work orders with instructions and resource allocation using large language models (LLMs) and data from various sources.
4. **Maintenance guidance generation:** Generative AI is used to integrate service notes and additional information with the repair plan, providing enhanced guidance for technicians.

Enhancing—and revolutionizing—maintenance operations with AI can lead to substantial cost savings, significant efficiency improvements, and heightened productivity, giving businesses a competitive edge.

Generative AI in predictive maintenance

Smart manufacturing use cases are revolutionizing many organizations, and a key driver of this is the incorporation of artificial intelligence into manufacturing processes.

Many firms have embarked on significant digital transformation journeys in the past two years with the goal of improving efficiency and resilience. However, a concerning gap exists between tech adoption and return on investment. While 89% of organizations have begun digital and AI transformations, only 31% have seen the expected revenue lift, and only 25% have realized the expected cost savings. In some cases, situations have even worsened.

This highlights the importance of not just implementing new technologies, but implementing them strategically. In other words, simply deploying AI isn't a guaranteed path to success. Manufacturers need to carefully consider how AI can address their specific challenges, and then integrate it into existing processes effectively.

To better understand how gen AI can be used to build robust predictive maintenance solutions, let's dig into the characteristics of organizations that have successfully implemented AI.

The path to success

Successful organizations exhibit common traits across five key areas:

- **Identifying high-impact value drivers and AI use cases:** Efforts should be concentrated on domains where artificial intelligence yields maximal utility rather than employing it arbitrarily.
- **Aligning AI strategy with data strategy:** Organizations must establish a strong data foundation with a data strategy that directly supports their AI goals.

- **Continuous data enrichment and accessibility:** High-quality data, readily available and usable across the organization, is essential for the success of AI initiatives.
- **Empowering talent and fostering development:** By equipping their workforce with training and resources, organizations can empower them to leverage AI effectively.
- **Enabling scalable AI adoption:** Building a strong and scalable infrastructure is key to unlocking the full potential of AI by enabling its smooth and ongoing integration across the organization.

Maintenance strategy: A key value driver

A well-defined maintenance strategy can be a game-changer for any organization, driving significant revenue and cost savings. Here's how it works:

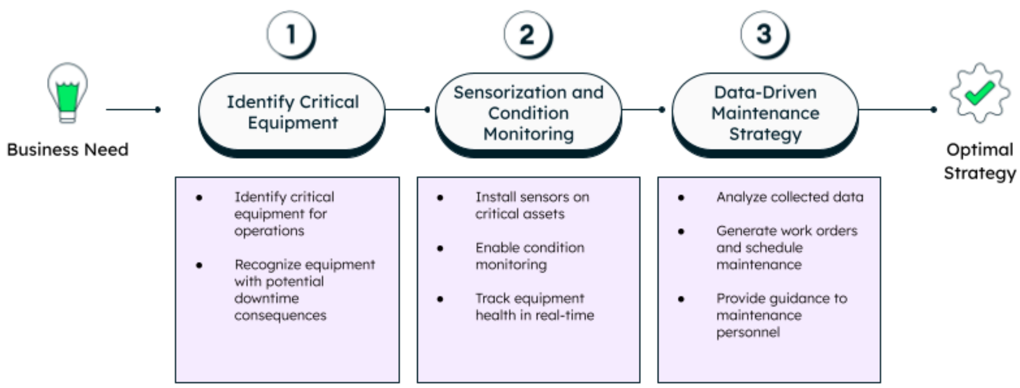
First, identify the equipment that is most crucial for your operations. Downtime for this equipment can lead to bottlenecks, halting production.

Second, equip these critical assets with sensors to enable condition monitoring. This allows you to monitor the health of the equipment in real time, identifying potential issues before they escalate into catastrophic failures.

Third, based on the prediction, the system can generate work orders, schedule maintenance activities, and even provide guidance to maintenance personnel. This ensures that maintenance is performed only when necessary, optimizing resource allocation.

This series of activities delivers tangible benefits. Costs are reduced through saved labor hours and extended machine lifespan. Additionally, revenue increases as your machines operate at optimal performance levels.

Figure 1. Steps required for an optimal maintenance strategy



What exactly is an optimal maintenance strategy?

An optimal maintenance strategy isn't a one-size-fits-all solution. It's about choosing the right blend of approaches based on your specific equipment and operational needs.

In today's processes, we see a spectrum of maintenance approaches. These methods range from highly complex and expensive at one end to simpler, more affordable options on the other:

- **Reactive maintenance:** This is the most basic approach in which maintenance is performed after a machine fails. While simple, it can lead to unexpected downtime and higher repair costs.
- **Preventive maintenance:** This is a proactive approach that involves scheduling maintenance tasks based on predetermined time intervals or usage metrics. This helps prevent breakdowns but can be inefficient as machine conditions can vary. Thresholds for these tasks may need to be adjusted due to factors like aging equipment, changes in processes, or different materials being used.
- **Condition-based maintenance (CBM):** This approach continuously monitors the health of the machine through sensors and data analysis. Maintenance is then

triggered based on the actual condition of the equipment rather than a set schedule. This is more efficient than preventive maintenance as it avoids unnecessary maintenance. Threshold-based alerting systems are often used with CBM.

- **Predictive maintenance:** This is the most advanced approach, using data analytics to predict potential equipment failures before they occur, which allows for proactive maintenance and minimizes downtime. Predictive maintenance requires significant upfront investment in sensors and data analysis tools.

How predictive maintenance boosts efficiency and saves money

Predictive maintenance uses data analysis to identify problems in machines before they fail. This allows organizations to schedule maintenance at the optimal time, maximizing machine reliability and efficiency.

Here's how predictive maintenance can benefit manufacturing operations, according to Deloitte:

3 to 5% Reduction in New Equipment Costs	**5 to 20%** Increase in Labor Productivity	**15 to 20%** Reduction in Facility Downtime
10 to 30% Reduction in Inventory Levels	**5 to 20%** Reduction in Carrying Costs	

Predictive maintenance is constantly evolving. We've moved beyond basic threshold-based monitoring to advanced techniques like machine learning (ML) models. These models can not only predict failures but also diagnose the root cause, allowing for targeted repairs.

The latest trend in predictive maintenance is automated strategy creation. This involves using AI to not only predict equipment breakdowns but also to generate repair plans, ensuring the right fixes are made at the right time.

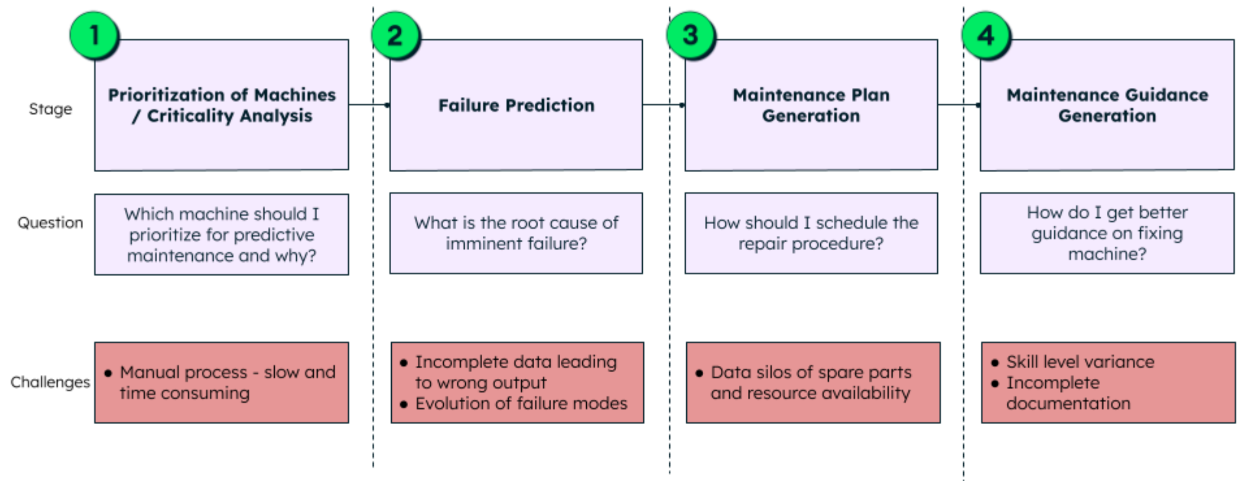
Automated strategy creation requires substantial investment in R&D, along with deep industry knowledge, access to relevant data, and practical operational experience. The question is, can generative AI help?

Generative AI and implementation challenges

The answer is yes, generative AI can help. But there are challenges at each stage of implementation that organizations must consider. Each stage involves a key question and

associated challenges, highlighting the steps and issues faced in predictive maintenance and machinery upkeep.

Figure 2. Different challenges seen at each stage of the predictive maintenance deployment



Now, let's envision a factory containing both automated and manual machines. Let's call it Gear Transmission Systems Ltd, whose primary output is gearboxes.

Within this factory, we have an array of equipment: cutting machines, milling machines, measurement devices, and more. As its general manager, you are tasked with managing the budget allocated for maintenance processes and improving strategies. One pressing question you must address is: which machines should take priority for the predictive maintenance projects, and why? This involves consulting with the maintenance managers and leaders and conducting quantitative analyses, a rather manual process.

Once you've identified the machines, the next step is to install sensors and train the machine learning model. However, two major challenges arise. First, you lack sufficient "run to failure" data to effectively train the model. Secondly, machine health deteriorates over time, leading to evolving failure modes with the age of the machine.

Assuming you manage to overcome these hurdles, the next phase involves maintenance scheduling and execution. You're faced with a myriad of data silos, including inventory data

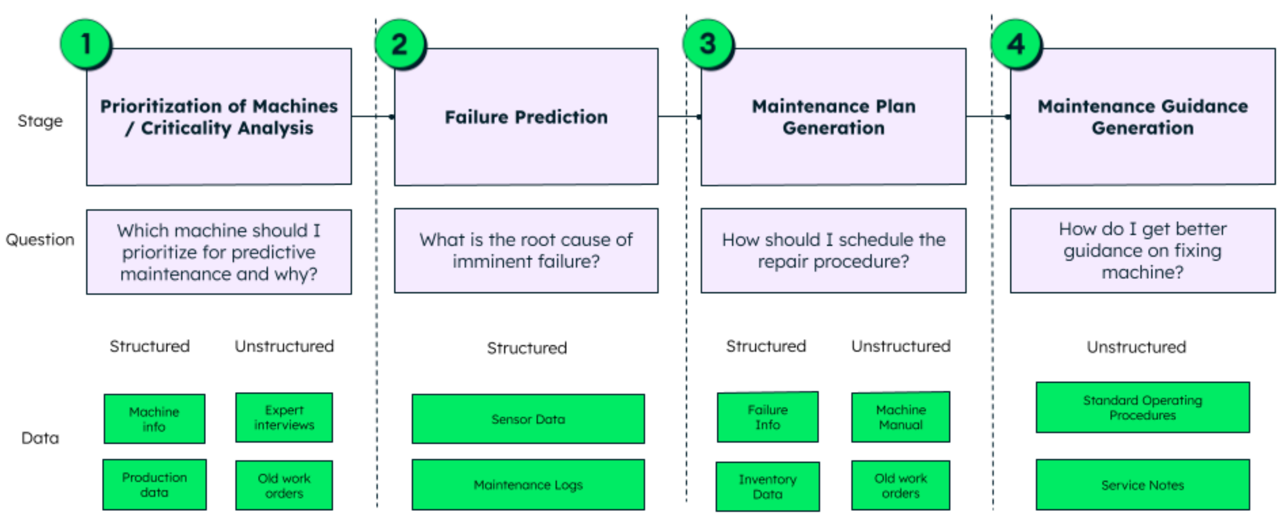
and resource availability data, which need to be integrated to formulate a comprehensive repair plan. Furthermore, it's essential to ensure that operators are adept at addressing minor machine issues to reduce reliance on external experts. While complex issues may still require OEM or SI support, internal troubleshooting capabilities are invaluable. Therefore, developing easy-to-follow documentation tailored to the skill levels of our staff is important.

Indeed, navigating these challenges presents a daunting task.

As Figure 3 shows, different data is required for solving above listed challenges:

- **Prioritization of machines/criticality analysis:** At this stage, we require both structured and unstructured data. We need previous machine failure data as well as expert analysis/opinion on which machines to prioritize for predictive maintenance and why.
- **Failure prediction:** This stage involves structured data such as sensor data and maintenance logs to identify the root cause of imminent failure.
- **Maintenance plan and guidance generation:** In both of these stages, we deal with both structured and unstructured data. The objective is to combine this data to generate an optimal repair plan and operator guidance.

Figure 3. Different data requirements at each stage



Now, let's build a four-stage application using MongoDB Atlas to demonstrate its capabilities.

Dive deep: Implementing predictive maintenance using MongoDB Atlas

MongoDB Atlas is the only multi-cloud developer data platform designed to accelerate and simplify how developers work with data. Using MongoDB Atlas, developers can power end-to-end value chain optimization with AI/ML, advanced analytics, and real-time data processing for innovative mobile, edge, and IoT applications.

Stage 1: Machine prioritization

Current machine prioritization for predictive maintenance relies heavily on manual analysis. Factory personnel gather historical and current machine data on utilization losses due to breakdowns. This data is then reviewed alongside the experience of maintenance managers and leaders. Based on this combined analysis, a roadmap for the predictive maintenance project is recommended, highlighting which machines should be prioritized.

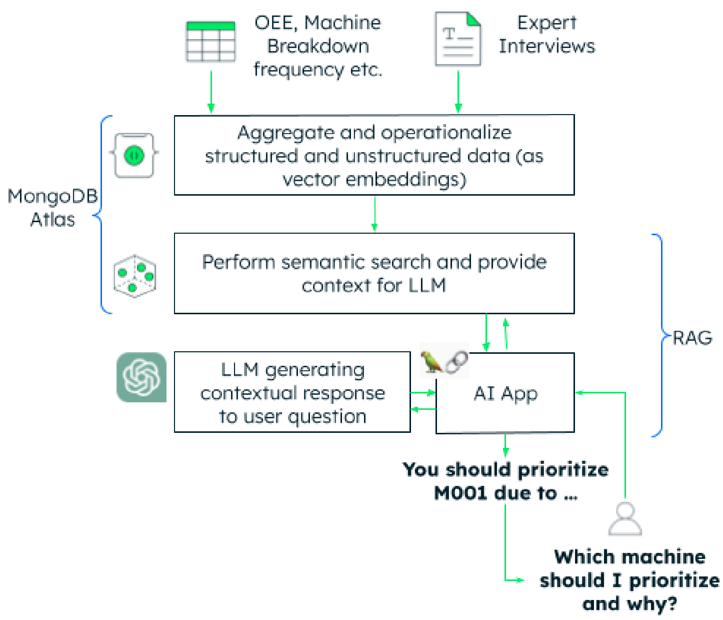
However, this approach has limitations. A reliance on manual analysis can be time-consuming and may not always capture the full picture of the maintenance project due to the limited use of quantitative data sources. Additionally, inconsistencies in interpretation can lead to an overdependence on institutional knowledge, which in turn can result in false analyses that impact the project's return on investment (ROI).

But with the arrival of generative AI, things have changed. A gen AI-based machine prioritization tool can be created to reduce the time manufacturing experts spend on manual analysis, and to decrease the risk of poor investments.

To leverage gen AI, experts need a data store capable of storing and operationalizing both structured and unstructured data. Having such a data store will allow them to perform semantic searches and to provide the right context to the large language model, ensuring it generates responses based on factory data without hallucinating. Such a system can result in positive business outcomes.

Here's how the system can look with MongoDB Atlas as the AI data store:

Figure 4. Generative AI-powered machine prioritization



This retrieval-augmented generation (RAG) system consists of four parts. First, an AI data store aggregates and operationalizes structured and unstructured data. In Figure 4, machine breakdown history and operational parameters are represented as structured data, while expert interviews are stored as unstructured PDF files. The PDF files are vectorized and stored in MongoDB Atlas. Atlas Vector Search is then utilized to perform semantic searches and to find meaningful context from the PDF embeddings.

[Atlas Vector Search](#) can be triggered using an AI application, connecting to MongoDB Atlas to retrieve the right context, which is then fed into the large language model to answer questions like "*Which machine should I prioritize and why?*" The response might

suggest prioritizing Machine M001 or M002 due to certain reasons, including but not limited to the criticality of the machine, high maintenance cost, etc.

MongoDB Atlas streamlines RAG implementation as it handles everything under the hood. App data, metadata, context data, and vectors are all stored in the same place. As the app evolves, the document model is inherently flexible and ideal for storing structured and unstructured data. You can add data to the same collections inside the database as needed, vectorize it, and store the vectors alongside it.

Once the data is stored, vector search capabilities are provided right out of the box, and search operations can be optimized using dedicated search nodes.

With MongoDB Atlas, it's just one query in one technology, one language, and one infrastructure to manage and scale, with no data duplication, ultimately leading to a lower total cost of ownership and a unified developer experience.

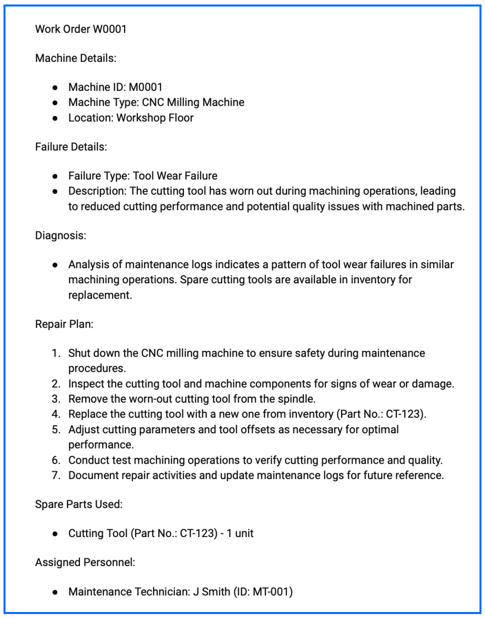
Figure 5 (a-c) shows some structured and unstructured data examples that we can use to build a machine criticality analysis tool. We have some structured data, as indicated by the red boxes showing a list of machines with planned work order percentage, unplanned work order percentage, cycle time per part (in minutes), and more.

Figure 5a. Structured data showing machine metrics

Raw material warehouse / Part Fabrication process						Assembly Station
Machine Type and Process Flow	EDM / Wire cutting	CNC / Lathing	CNC / Milling	Shot peening / Shot peening	CMM / Metrology Measurement	
Machine ID	M0005	M0002	M0001	M0003	M0004	
No of machine/line	3	2	3	3	3	
Cycle Time per part (mins)	21	14	25	20	21	
OEE, % (non idle time - losses)	89	83	83	88	85	
Bottleneck Index (cycle time/(OEE x no of machines)	8	8	10	8	8	
Planned Work order, %	70	60	50	60	70	
Unplanned Work order, %	30	40	50	40	30	
Annual maintenance cost/line, k CAD	15	20	65	35	21	

Then, we have unstructured information, like interviews with maintenance managers providing insights on which machines to prioritize, as well as work orders from previous breakdowns.

Figure 5b. Work Order



Work Order W0001

Machine Details:

- Machine ID: M0001
- Machine Type: CNC Milling Machine
- Location: Workshop Floor

Failure Details:

- Failure Type: Tool Wear Failure
- Description: The cutting tool has worn out during machining operations, leading to reduced cutting performance and potential quality issues with machined parts.

Diagnosis:

- Analysis of maintenance logs indicates a pattern of tool wear failures in similar machining operations. Spare cutting tools are available in inventory for replacement.

Repair Plan:

1. Shut down the CNC milling machine to ensure safety during maintenance procedures.
2. Inspect the cutting tool and machine components for signs of wear or damage.
3. Remove the worn-out cutting tool from the spindle.
4. Replace the cutting tool with a new one from inventory (Part No.: CT-123).
5. Adjust cutting parameters and tool offsets as necessary for optimal performance.
6. Conduct test machining operations to verify cutting performance and quality.
7. Document repair activities and update maintenance logs for future reference.

Spare Parts Used:

- Cutting Tool (Part No.: CT-123) - 1 unit

Assigned Personnel:

- Maintenance Technician: J Smith (ID: MT-001)

Figure 5c. Interview with Maintenance Manager

Interview with Maintenance Manager

Interviewer: Good morning, Taylor, how are you?

Taylor Smith: Good morning. I am good, hope you are doing well too.

Interviewer: Taylor, Thank you for giving us your time. Today we wanted to understand the maintenance strategy at this site. Could you please start by explaining your role here and what it entails?

Taylor Smith: Certainly. As the Maintenance Manager, I oversee all maintenance activities within the facility. This involves developing maintenance plans, scheduling repairs, managing maintenance budgets, and ensuring all equipment is running efficiently to support production operations.

Interviewer: That sounds like a lot to manage. How do you typically plan maintenance activities for the year?

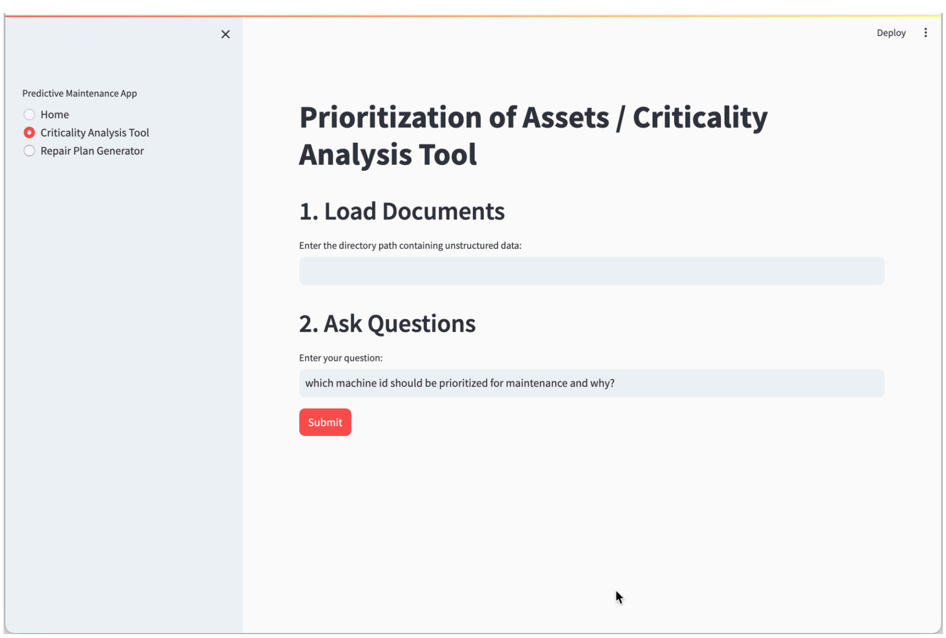
Taylor Smith: It's definitely a comprehensive process. At the start of each year, we conduct a thorough assessment of all our machinery and equipment. Based on historical data, maintenance logs, and input from our maintenance technicians, we prioritize which machines require attention and develop a maintenance schedule for the year.

Interviewer: How do you prioritize maintenance tasks?

Taylor Smith: We consider several factors when prioritizing maintenance tasks. One key factor is the criticality of the machine to our production process. Machines that are essential for maintaining production efficiency and meeting customer demands are given higher priority. Additionally, we consider the frequency and severity of past failures, as well as the potential impact of downtime on production schedules.

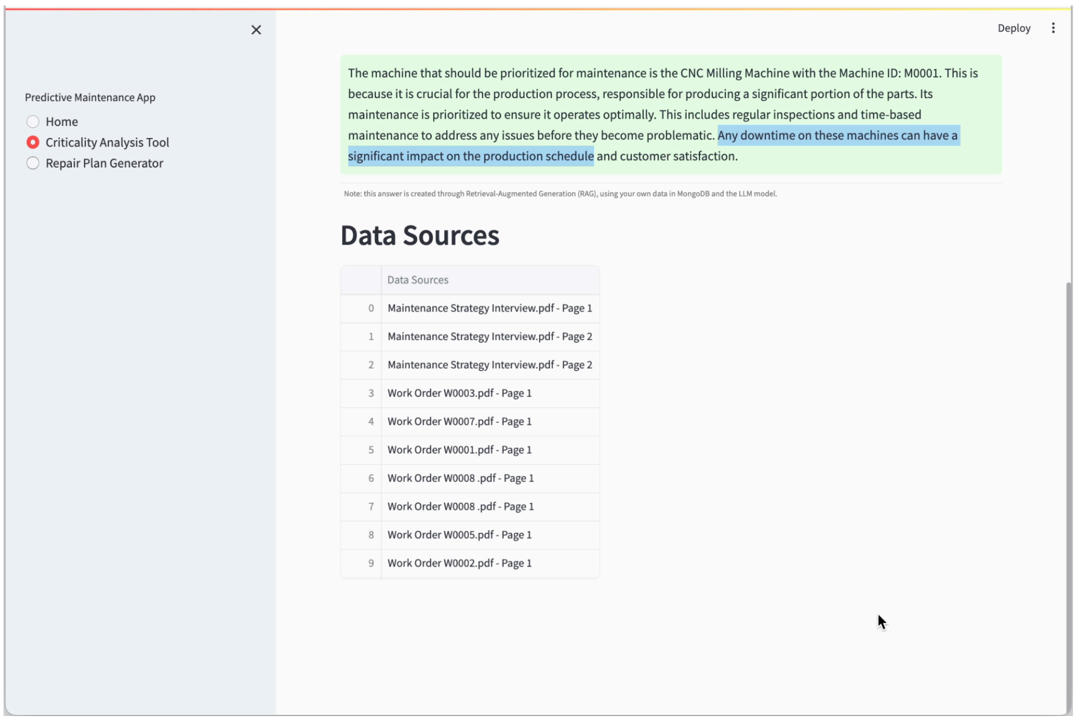
Let's see all of this together in a series of images below. Once all of this data is incorporated using the architecture shown previously and stored inside MongoDB, we can ask questions like, "Which machine ID should be prioritized for maintenance and why?"

Figure 6. Prioritization of Assets / Criticality Analysis Tool



The AI application will run the semantic search, retrieve the right context, feed it into the large language model, and generate a response specific to the use case. In this case, it suggests prioritizing the CNC Milling Machine with the Machine ID: M0001. It also provides reasons from structured and unstructured data sources, including maintenance strategy interviews and work orders, which contributed to generating the response. Thus, it provides a thorough response based on the available data.

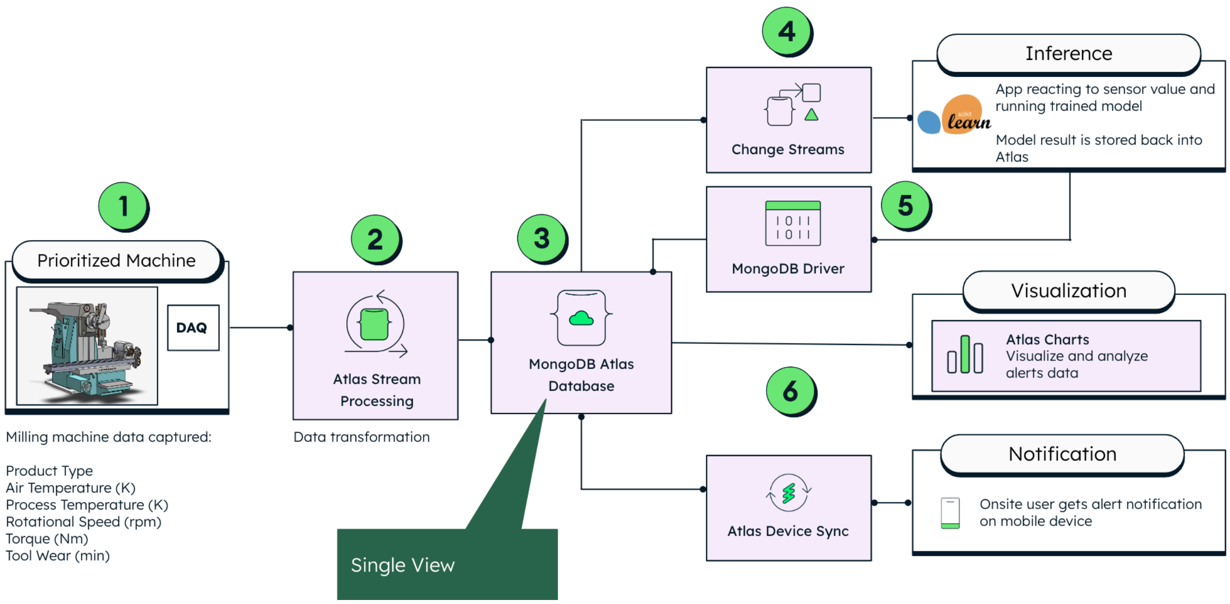
Figure 6a. Results and Data Sources from the Analysis Tool.



Stage 2: Failure prediction

Now that we've discussed prioritizing equipment, let's move on to failure prediction. MongoDB Atlas provides all the necessary building blocks or tools to implement failure prediction. By providing a unified view of operational data, real-time processing capabilities, integrated monitoring and alerting, and seamless compatibility with machine learning tools, MongoDB Atlas enables organizations to optimize machine performance and minimize downtime.

Figure 7. Smart milling machine uses real-time data to predict failures



As seen in this workflow, we have our prioritized machine, which is a milling machine with attached sensors that collect data such as air temperature, rotational speed, torque, and tool wear. This data will be processed through Atlas Stream Processing, enabling the processing of streams of complex data using the same data model and Query API used in Atlas databases. Atlas Stream Processing enables developers to build aggregation pipelines to continuously operate on streaming data without the delays inherent to batch processing. Results can be continuously published to MongoDB Atlas or to a Kafka topic. This allows data transformation and enrichment before it even lands in the database.

Once the data is in MongoDB, another application can react to sensor values and run a trained model designed to predict failures. The model results can be stored back into Atlas (between steps 4 and 5/Inference in Figure 7). These results can then be visualized using Atlas Charts. Finally, Atlas Device Sync can be used to push notifications to mobile devices for onsite users.

By identifying failure types based on stored data in the database, we can visualize them with Atlas Charts, which provides out-of-the-box visualization without the need for third-party tools. Additionally, Atlas device sync enables the pushing of notifications,

alerts, and messages to various platforms, including mobile apps, tablets, and web applications. This establishes an end-to-end system for failure prediction.

Figure 8. Machine generates data, which gets analyzed for failure, and alerts are sent out to relevant personnel

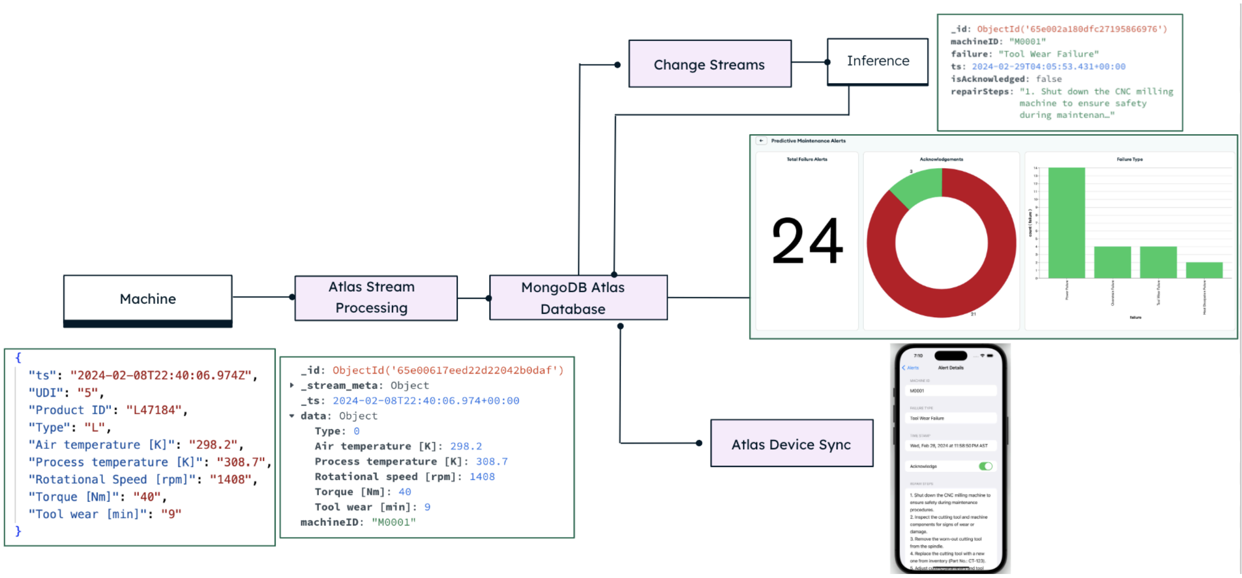
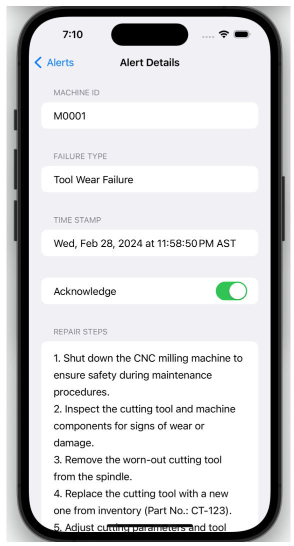


Figure 8 shows a machine that is constantly generating sensor data. It's noteworthy that much of this data appears as strings, even though they represent numerical values. This formatting issue provides a use case for the utilization of Atlas stream processing to convert these strings into numerical data. Subsequently, we will transform and consolidate this data into a structured data object, as depicted, and append a machine ID. This constitutes the dataset that will be stored in the database.

Figure 8a. Atlas Stream processing transforming string into numerical values

Following this, another inference process will analyze all sensor data for indications of failure, and the result of the failure can be stored inside MongoDB in a collection labeled *machine_failure*. This collection can be hooked up to Atlas Charts for visualization and analysis purposes. At the same time, we can enable Atlas Device Sync to synchronize the alerts to a mobile device, completing the end-to-end process.

Figure 8b. Alert Details on mobile



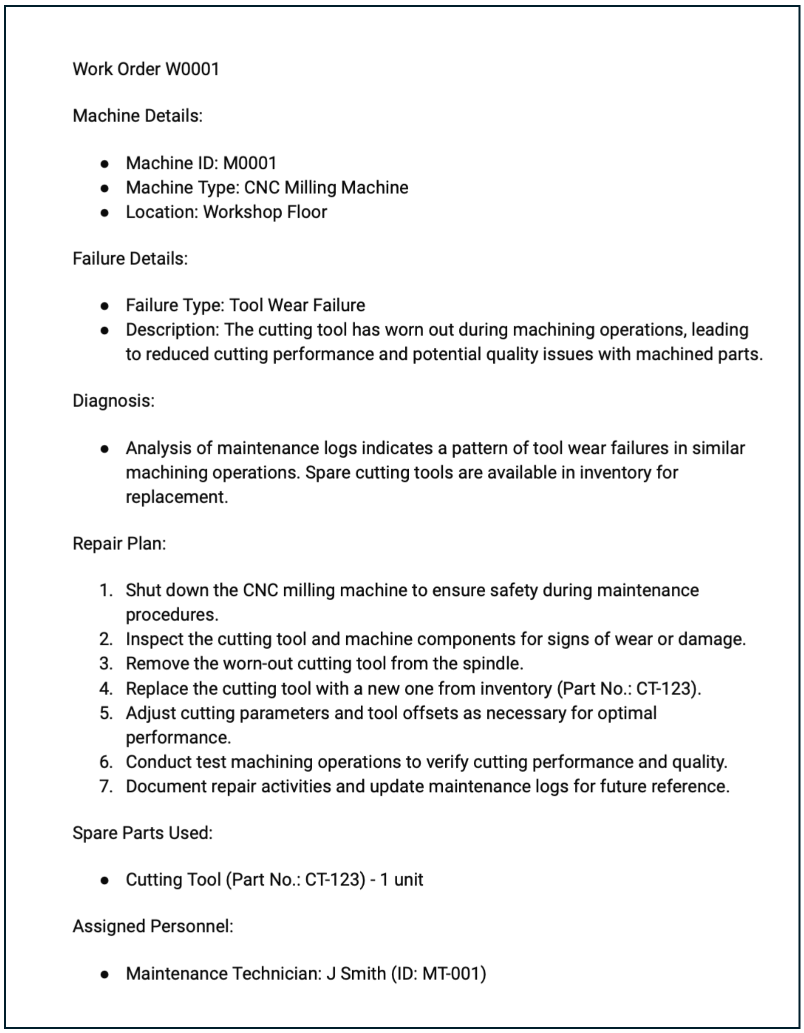
Stage 3: Repair plan generators

Having identified the nature of the equipment failures, the implementation of a comprehensive repair strategy becomes paramount. First, we have to generate a maintenance work order. This order should include repair instructions, spare parts needed, schedule, and resource availability information. In this case, both structured and unstructured data are involved. The repair instructions will come from the machine manual.

For this process, MongoDB Atlas acts as the operational data layer, seamlessly integrating structured and unstructured data.

Imagine if you needed to generate a work order using a highly specific template (as shown in Figure 9). This work order must encompass repair instructions, required spare parts, scheduling, and resource availability, all stemming from machine manuals. Let's say the impending failure is due to tool wear, necessitating tool change operations.

Figure 9. Example of a maintenance work order



Work Order W0001

Machine Details:

- Machine ID: M0001
- Machine Type: CNC Milling Machine
- Location: Workshop Floor

Failure Details:

- Failure Type: Tool Wear Failure
- Description: The cutting tool has worn out during machining operations, leading to reduced cutting performance and potential quality issues with machined parts.

Diagnosis:

- Analysis of maintenance logs indicates a pattern of tool wear failures in similar machining operations. Spare cutting tools are available in inventory for replacement.

Repair Plan:

1. Shut down the CNC milling machine to ensure safety during maintenance procedures.
2. Inspect the cutting tool and machine components for signs of wear or damage.
3. Remove the worn-out cutting tool from the spindle.
4. Replace the cutting tool with a new one from inventory (Part No.: CT-123).
5. Adjust cutting parameters and tool offsets as necessary for optimal performance.
6. Conduct test machining operations to verify cutting performance and quality.
7. Document repair activities and update maintenance logs for future reference.

Spare Parts Used:

- Cutting Tool (Part No.: CT-123) - 1 unit

Assigned Personnel:

- Maintenance Technician: J Smith (ID: MT-001)

Figure 10. MongoDB Atlas as the operational data layer for structured and unstructured data

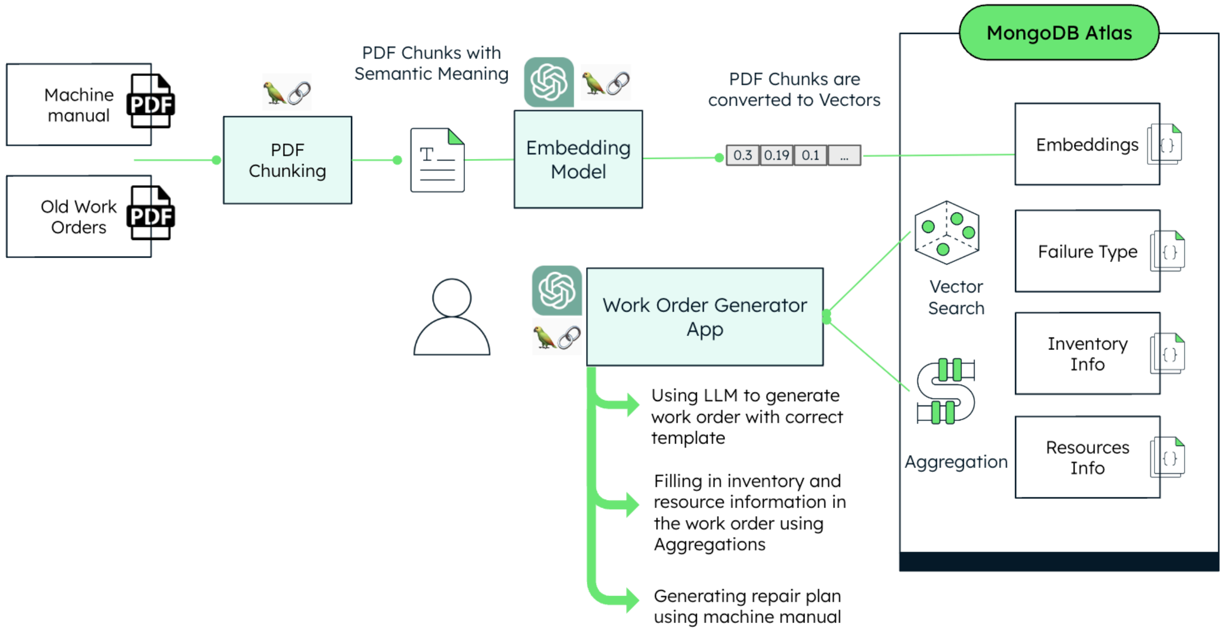


Figure 10 shows the process of work order generation using generative AI. First, we must extract chunks of information from a milling machine's operating and repair manual, as well as from old work orders stored as PDF files, and convert them into vectors. These embeddings are then stored in MongoDB Atlas. MongoDB's versatility allows for the storage of both structured and unstructured data within the same database. Leveraging Atlas Vector Search and Aggregation pipelines, we can integrate this data to feed into a large language model (LLM) powering a work order generator application. The LLM analyzes the data to generate the appropriate work order and template, drawing from past examples. It populates inventory and resource details using aggregation techniques and structured data. Finally, it generates a repair plan similar to the one shown in Figure 9.

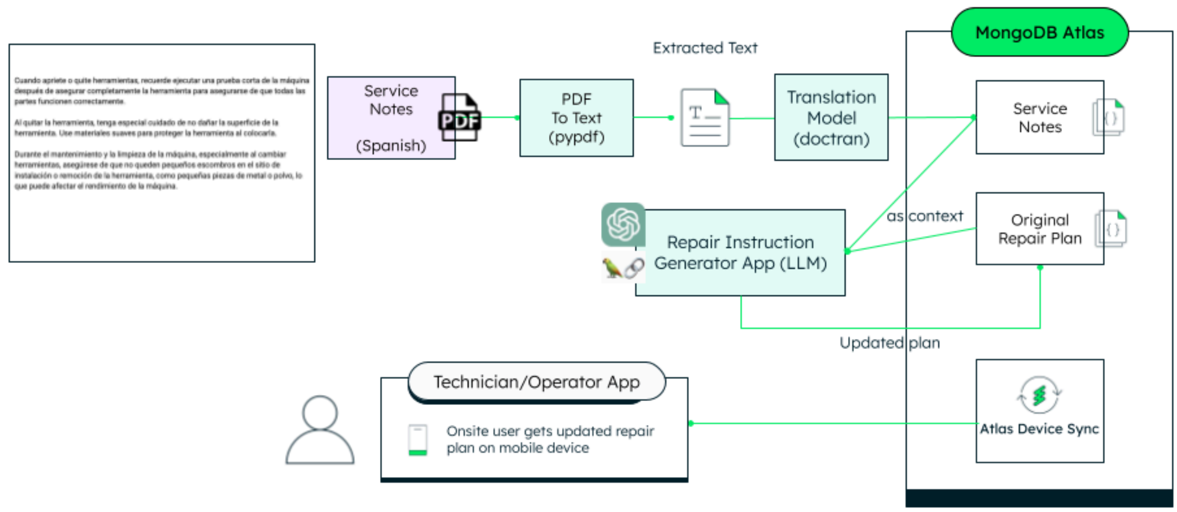
What sets this approach apart is the ability to use the same MongoDB database to store structured data such as failure types, spare parts inventory, and resource information. By employing the aggregation framework to extract relevant information from structured data and vector search to glean insights from vectors, the LLM within the work order generator application gains contextual understanding.

This application seamlessly utilizes the LLM to generate work orders with the correct template, filling in inventory and resource details through aggregations, and ultimately creating repair plans based on machine manuals. This application can run inside a central maintenance management system.

Stage 4: Maintenance guidance generation

So we come to the last step: How can we use gen AI to enhance the operator or technician guidance to maintain the machine?

Figure 11. Using the RAG Approach for Operator Work Instructions



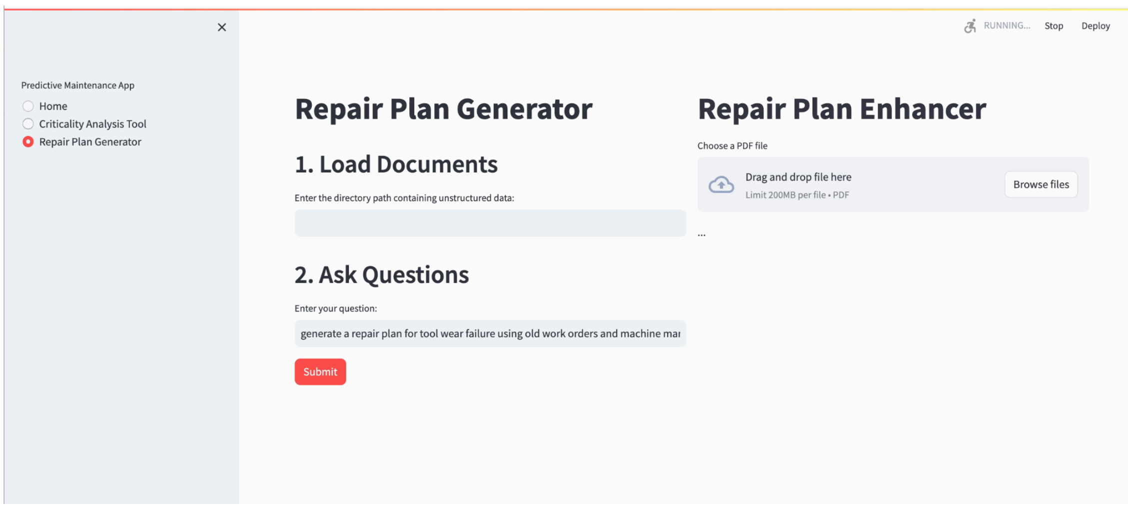
Let's walk through an example scenario here. The repair plan was generated in the last step. Now, the computerized maintenance management system (CMMS) has found some service notes uploaded to the platform by another technician, but they're written in another language, let's say Spanish. We can use the RAG architecture again to intelligently merge these service notes with the repair instructions generated in the previous step.

We first need to extract text from the PDF, translate it into English since our other data is in English, and then provide the service notes as well as the repair plan to the LLM as context. So, we have the original plan from the previous steps, and we combine it integrally using the LLM with the service notes obtained in this step. Note that we're not

performing vector search here. Once the plan is updated, then we can push it down to the technician's mobile app with Atlas device sync.

Let's see how it all looks in action. We have the repair plan generator on the left-hand side and the repair plan enhancer on the right-hand side. Using all the data we've discussed, we're going to generate a repair plan for tool wear failure using old work orders and machine manuals.

Figure 12. Repair Plan Generator



The system will generate step-by-step instructions (Figure 12a) on how to change the tool by looking at all work orders and different pages of the machine manual (Figure 12b). So, we don't have to go through all these 100+ pages of machine manuals to extract the right text based on your question.

Figure 12a. Answers from the Repair Plan Generator

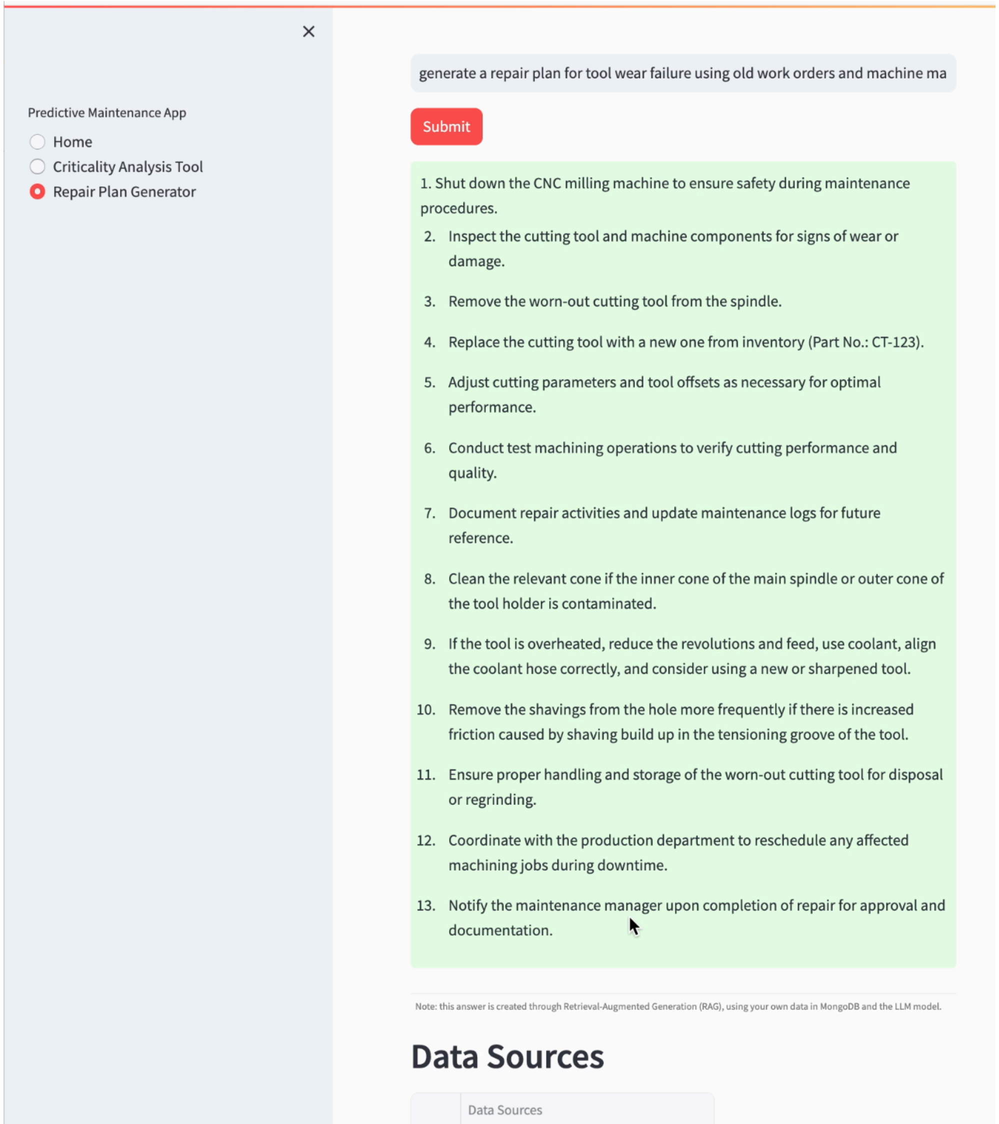
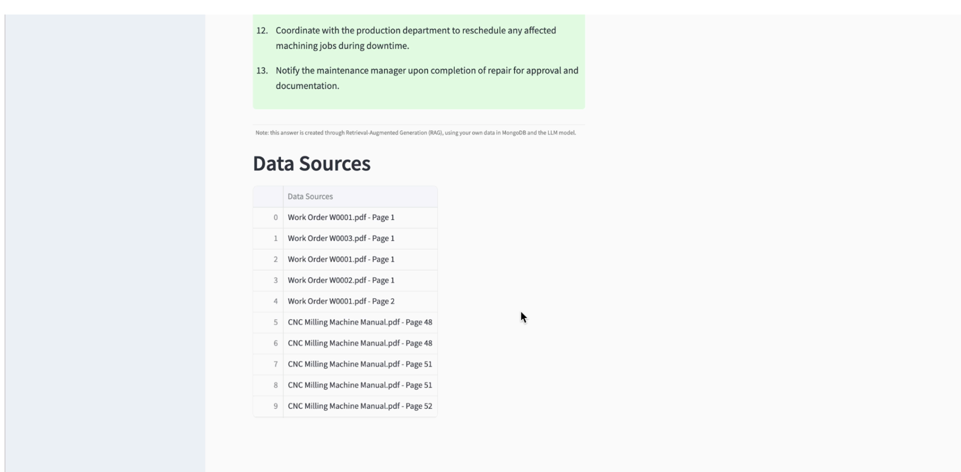
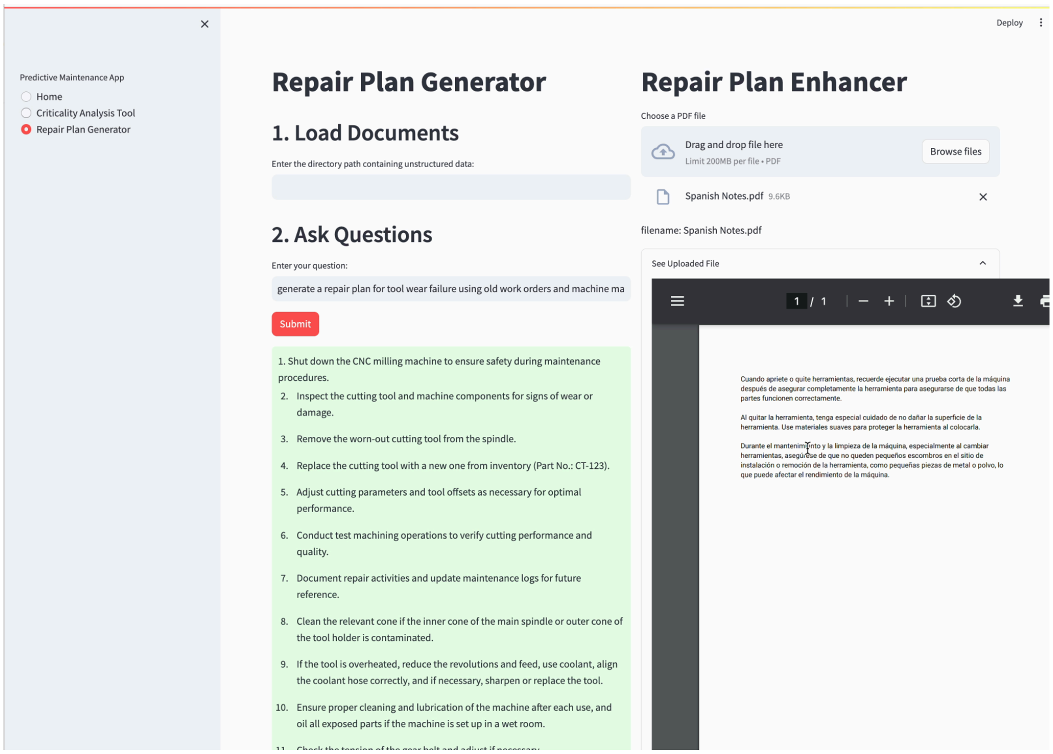


Figure 12b. Data Sources



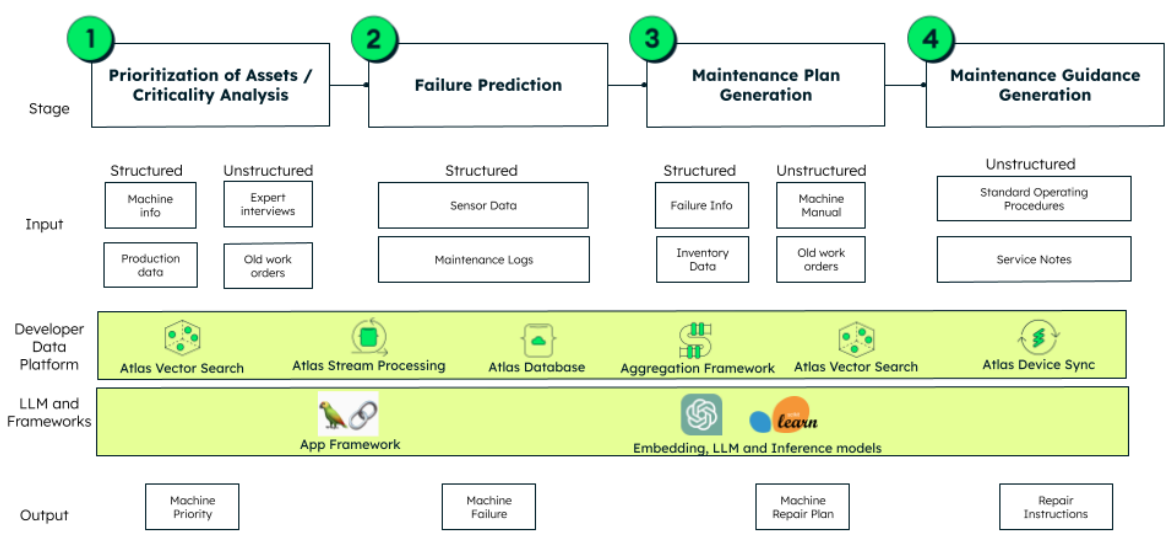
Next, we're going to ask to take into consideration additional notes taken by the technician on the repair plan enhancer application (Figure 12c), and then merge them with the original repair procedure. The enhanced plan can therefore have more detail based on the additional document we added.

Figure 12c. Repair Plan Enhancer



In summary, we are essentially integrating AI and gen AI apps to implement an end-to-end predictive maintenance strategy (Figure 13).

Figure 13. Model chaining with unified data store



Our input consisted of a combination of structured and unstructured data. We leveraged the various services offered by MongoDB Atlas Developer Data Platform, including Atlas Vector Search, Atlas Stream Processing, Atlas Device Sync, and, of course, the MongoDB database and aggregation framework. These features enabled us to provide the right context to the LLM and the appropriate data to the AI model.

Ultimately, we obtain the desired output at each stage, ranging from machine prioritization, failure type identification, and repair plan formulation, to instruction generation.

The future of maintenance is here

Advancements in AI are revolutionizing maintenance practices, potentially leading to a 50% reduction in machine downtime, a significant boost in labor productivity ~50%, and a 30-60% cut in maintenance costs. For manufacturers seeking maintenance excellence, a unified data store and a developer data platform are key enablers. These tools provide the foundation for integrating AI applications that can analyze sensor data, predict failures, and optimize maintenance schedules.

Curious about the possibilities of leveraging MongoDB Atlas for your app? Apply for an innovation workshop to explore the possibilities with our experts. You can also experience an example of acoustic fan diagnostics by diving into our GitHub repository.

You can also watch the complete recording of our MongoDB .local NYC 2024 presentation on this topic.

About MongoDB

MongoDB empowers innovators to unleash the power of software and data. Whether deployed in the cloud or on-premises, organizations use MongoDB for trading platforms, global payment data stores, digital end-to-end loan origination and servicing solutions, general ledger system of record, regulatory risk, treasury and many other back-office processes. At the core of our developer data platform is the most advanced cloud database service on the market, MongoDB Atlas, which can run in any cloud, or even across multiple clouds to get the best from each provider with no lock-in.

To learn more about MongoDB, visit MongoDB.com

About the authors

Dr. Humza Akhtar

Humza is a smart manufacturing and automotive expert at MongoDB. Prior to joining MongoDB, he worked at Ernst & Young Canada in the digital operations consultancy practice. Humza attained his Ph.D. at Nanyang Technological University, Singapore, and worked with the Singapore manufacturing industry for many years on Industry 4.0 research and implementation. He has spent most of his career enabling smart and connected factories for many manufacturing clients. His book *"Implementing Industry 4.0 - The Model Factory as the Key Enabler for the Future of Manufacturing"* is the first book of its kind detailing the real-world implementation of concepts related to digital manufacturing.

Sebastian Rojas Arbulu

Sebastian is a cross-industry specialist with experience in consulting, banking digital transformation, sales, data analysis, marketing, and teaching. He has worked with both startups and established companies. Before joining MongoDB, he was a Business Consultant at NTT DATA, specializing in banking digital transformation. Sebastian holds a Bachelor's Degree in Business Administration and is on a mission to empower businesses with the right tools to unlock the power of their data.

Resources

For more information, please visit mongodb.com or contact us at sales@mongodb.com.
Case Studies (mongodb.com/customers)
Presentations (mongodb.com/presentations)
Free Online Training (university.mongodb.com)
Webinars and Events (mongodb.com/events)
Documentation (docs.mongodb.com)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Enterprise Download (mongodb.com/download)
MongoDB Realm (mongodb.com/realm)

Legal Notice