

Storage Management Software

Milestone 04

Luis Bodenbach (3201837)
Thomas Scherer (3698925)
Berkay Baltaci (3174223)
Bilal El Kadri (3854428)
Amartya C. Dudhe (3928343)

1 Introduction

Lahn Inc. runs an online shop but doesn't have much space to keep products in the storage. That means every choice about what to order and when becomes important. To solve that issue, we were provided with a dataset of previous transactions, with over 110,000 sales between 2011 and 2016. Each transaction in this file contains information like the sale date, who bought the product (age, gender), where they live, what and how much they bought, and how much money was involved. Looking at this data, some patterns can directly be seen. Sales hit their highest points in May, June, and December. Sales didn't just stay steady; they grew a little every year. This makes predicting future demand necessary if Lahn Inc. wants to avoid empty shelves or a lot of unsold products. Three major product categories make up most of their business, with 18 subcategories underneath and more than 130 products overall. Some products make much more money than others. For example, "Bikes" bring in almost two-thirds of all profit, but "Clothing" makes up just a small slice. Most buyers are in the USA, especially California, but the company also ships to five other countries, which means needed stocking might change in each country. The problem is simple: Too much stock costs money and takes up space. Too little stock means lost sales. Both are bad for the business. Our project's goal is to predict what will sell and when by forecasting demand, so storage gets used better and customers can always find what they want.

Then we want to use AI to create a Business Report with recommendations, so the company can easily get insights in only a few minutes. With smarter stocking based on the patterns we see in the data, Lahn Inc. can cut down costs and boost profits, making the whole business more profitable, run smoother and satisfy customer needs.

2 Technical Solution

Our solution was a combination of machine learning models, forecasting algorithms, and software development techniques to meet Lahn Inc.'s needs. We followed the CRISP-DM methodology (Cross-Industry Standard Process for Data Mining) to structure the project phases. This provided a clear roadmap from business understanding and data understanding through data preparation, modeling, evaluation, and deployment.¹ Data Processing & Features: We did not perform that much data cleaning, we found around 1,000 duplicate records, but decided not to delete them, because these transactions did not have a Unique identifier, so the duplicates could have just come from different Persons with the same name, in the same country buying the same products and quantity. For predictive modeling, we implemented gradient boosting machine learning models using XGBoost. This is a high-performance libraries that can be used for regression and classification tasks. In our case, the primary prediction target was Order_Quantity (the number of units sold per transaction or aggregate period), treating it as a demand indicator. We trained the model on historical data to predict future demand

¹ Chapman et al., 2000

67 levels. XGBoost was chosen for its ability to 116 make the output of our analyses more accessible
 68 handle nonlinear relationships, being very quick 117 we integrated a Large Language Model (LLM) to
 69 and feature interactions effectively, while also 118 generate polished reports. Specifically, we used
 70 providing useful outputs like feature importance 119 Ollama which is a runtime environment for
 71 scores for interpretability. The result was a 120 deploying LLMs locally that uses a model such as
 72 predictive model that can forecast short-term 121 Phi-3 or LLaMA 3.2 ³ and even the newest
 73 demand for products, helping inform how much of 122 Deepseek R1 (which goes up to 671B and is around
 74 each item to stock in upcoming periods. However, 123 400GB)⁴. The pipeline works as follows: after
 75 XGBoost did not deliver high quality outputs, 124 performing, for example a demand forecast for
 76 which sometimes were really questionably, so we 125 Germany next June, the system passes the
 77 decided to use Prophet for time Series Forecasting.² 126 forecasted quantity for each sub-category as a table
 78 Prophet is a robust forecasting library that models 127 to the LLM as our variable part. The prompt also
 79 trend, seasonality, and holidays, and is well-suited 128 includes some fixed, guided commands on how the
 80 for business time series data. The runtime of the 129 pdf file should be structured and which
 81 prophet model is between 1 and 2 minutes, instead 130 components have to occur in the Report (for
 82 of 10 seconds for the XGBoost approach. This is 131 example a business tip for the worst 3 performing
 83 because for instance, Prophet quantified the yearly 132 categories and the instruction, to create a
 84 seasonal effect (peaks around mid-year and 133 professional report for a business audience) to the
 85 December) and could extrapolate trend growth. 134 LLM. The LLM then produces a well-written
 86 The forecasts outputs of Prophet were reliable 135 summary. By including an LLM, we essentially
 87 looking predictions which we were very happy 136 added an AI “analyst” to our system that can take
 88 with. By predicting a few months ahead, Lahn Inc. 137 the quantitative findings and express them in
 89 can time its reorders to ensure stock arrives before 138 natural language, tailored to support business
 90 demand surges and avoid excess ordering when a 139 decision-making. This automated report generation
 91 dip is expected. Prophet’s ability to incorporate 140 was implemented as a final step in our FastAPI
 92 seasonality made it ideal for capturing the patterns 141 pipeline. Once analysis is done, the backend calls
 93 identified. We validated the forecasts by manually 142 the local LLM with the prepared prompt, then
 94 creating a Power Pivot table in excel, extracting the 143 returns the generated report (PDF) to the user.
 95 information we needed to compare with the
 96 forecasts. System Architecture: The backend was
 97 built with FastAPI, a modern Python web
 98 framework, which served our trained models and
 99 forecasting logic as RESTful endpoints. This
 100 backend could ingest new data or parameters and
 101 return predictions, forecasts, and recommended
 102 actions, but we only used it to transfer two
 103 variables *when* and *where* from frontend to
 104 backend. On top of this, we created a simple front-
 105 end interface in two forms: a lightweight web UI
 106 using HTML/CSS (served via the FastAPI
 107 backend) for easy access through a browser, and a
 108 desktop UI built with Tkinter (very minimalistic
 109 and optically not very appealing) for demonstration
 110 purposes. The user, like an inventory manager, can
 111 input the country and the month, that he wants
 112 forecasts about. The backend function is then
 113 called with those parameters, and results are
 114 directly moved into an output folder as a pdf file.
 115 Automated Reporting with LLM Integration: To

² Taylor & Letham, 2018

³ Vaswani et al. (2017)

⁴ Online-Website: <https://ollama.com/library> (last visited: 15/07/25 at 09:25 am)

3 Insights from the Data

Looking at our sales data, we immediately see that certain products always appear to be in the top 3 performing sub-category. For example, based on several test project reports generated with our application, we see a high performance of “Tires and Tubes”, which is constantly at the top. Showing strong and steady demand. When we use Prophet for demand forecasting, our ForecastGenerator class always provides the same output, which looks like the following table:

Sub_Category	Forecast
Tires and Tubes	1624
Bottles and Cages	1103
Helmets	917
Caps	502
Jerseys	433
Fenders	187
Cleaners	111
Gloves	100
Hydration Packs	81
Vests	78
Touring Bikes	63
Mountain Bikes	55
Road Bikes	51
Socks	21
Bike Stands	2
Bike Racks	1

The table ranks each sub-categories demand by forecasted demanded unit. This table goes directly, how it is, into our LLM prompt, together with some instruction for output generation for the LLM. When it comes to managing and optimizing inventory storage, it only makes sense that we prioritize good performing categories. However: It is obvious, that more tires and tubes are getting sold, than actual bikes. Still, 60 sold bikes make more profit for the company than 2000 tires, which means prioritization based only on demand quantity is nonsense. Back to the data, we can see that there is a pattern that repeats every year: sales ticking upward as spring turns to summer, with May and June especially busy. People seem to be getting ready for warmer weather. Then, December brings another rush, probably tied to holiday shopping. Sales slow down in July, August, January, and February. So,

the business has its busy stretches in early summer and again at the year’s end, then quiet spells right in the middle of summer and just after New Year’s. When we ran demand forecasts using Prophet, those trends became even clearer. The model picked up on the spikes in sales around mid-year and December and the slower months in between. Just looking at average sales wouldn’t show these swings. If Lahn Inc. planned only for “typical” months, they might run out of stock during peak times or end up with too much inventory sitting around when things slow down. This is why data analysis is very important. There’s a sharp split when you look at profits by category. Bicycles make up about 64% of profits, accessories about 27%, and clothing just 9%. These numbers come from profit data collected between 2011 and 2016. So, bicycles are the big moneymakers. Clothing, while it’s one of the main categories, only brings in a little money. Sales by region also paint an interesting picture. The United States leads the way, especially California. This is not a big surprise because it is a big state and the company has a strong presence in America. But other regions, including parts of Europe, have their own sales patterns. Local holidays and weather seem to play a role. In Australia and United States, demand for bikes is especially high and makes up a large slice of all bicycle sales (10.000 – 11.000 bike sales compared to 2.000 – 4.000 in the other countries). So, using the same stocking plan everywhere wouldn’t work. Different countries need different strategies. Here for a broad overview a table on the sum of total sold units (every product) for each country, making it clear where the main markets and therefore the main focus of Lahn Inc. should be:

Country	Sum of sold items
Australia	263.585
Canada	192.259
France	128.995
Germany	125.720
United Kingdom	157.218
United States	477.539

To sum up, forecasting demand with Prophet helped Lahn Inc. see not just when sales go up and down, but how that varies by category and location. These details can help the company make smarter inventory choices and stay ready for what customers want, when they want it.

4 Business Recommendations

The above insights guided us to propose the following business recommendations for Lahn Inc. to improve their inventory management and overall profitability:

Predict and Prepare for Seasonal Peaks: Prioritize stock availability in high-demand months. The company must increase its inventory of popular products (especially bikes and related accessories) before May, June and December of each year. This may involve increasing orders from suppliers in the preceding months and making sure there's enough space in the warehouse for all new products. In addition, aligning the marketing campaigns and promotions with these peak periods can boost sales when customers are most interested. On the other hand, in slower months (like January and February, or late summer), Lahn Inc. can reduce inventory levels to avoid overstock.

Focus on High-Profit Categories, Rationalize Low-Profit Stock: The data clearly shows that there are differences between the contributions of categories to profit. We recommend shifting inventory investment toward the Bikes category and top-performing sub-categories. These are the ones that bring in the most profits. Ensure that new models or variants of bikes, which historically sell well, are always stocked to our application's forecast. This reduces the chance that they need to restock, which can be really costly. On the other hand, they should re-evaluate the Clothing category: with only ~9% of profit coming from clothing, Lahn Inc. should consider reducing the range or depth of inventory in this category. However, not having stocked a cheap product can also backfire into customer dissatisfaction and effect the sales of other products, so they need to be careful here.

Implement Regional Inventory Strategies: adapt inventory levels and assortment to regional demand patterns. Given that the USA (especially California) is a sales driver, warehouses serving that region should maintain higher inventory levels for the top-selling products (such as the most popular bike models) and prepare for the significant increase in demand during the summer and holidays in that region. In regions where demand is lower or where seasonal patterns differ, adjust the stock accordingly. This could involve a slightly later build-up for holiday inventory in Europe if their peak is more concentrated in December than November.

Adopt Demand-Driven Reordering (Automation): We suggest adding our forecasting system to the reordering process. Instead of always ordering manually, Lahn Inc. should use the results from our model to trigger reorders. For example an automated alert system: if the forecast for the next N weeks shows that the stock of Product X will fall below a critical level before the next shipment can arrive, an order is suggested or automatically placed with the supplier. This kind of just-in-time inventory management will keep inventory levels just right, never too high which costs money and never so low that it causes stockouts. It may also be helpful to shorten supplier lead times or find other suppliers for important items (especially those in the Bikes category). This will help respond to sudden increases in demand. Over time, this demand-driven approach can be improved to include safety stock calculations for products with forecast uncertainty. The goal is a semi-automated system where the data triggers inventory actions. This frees up the managers to handle exceptions and strategy instead of manual monitoring. Should Lahn Inc. decide to move forward with this project, we would be more than ready to take on the assignment.

Bundle for complementary Products: use product pairings to create bundled offerings and improve the customer experience. When the data shows that people often buy a bike and a helmet together, Lahn Inc. can formalize that by selling the "Bike and Helmet Combo" at a small discount, or by just suggesting the helmet on the bike's product page. Bundling not only can increase sales of accessories but also ensures customers get everything they need at once, improving satisfaction.

5 Reflection on Plan vs. Implementation

At the beginning of the project, our goal was to build a system that could help improve inventory decisions using data. We planned to use machine learning models like XGBoost or LightGBM to predict demand and evaluate product profitability. The idea was also to build a user interface where people could select different types of analyses and get readable summaries, generated with the help of large language models such as LLaMA or Phi-3. We wanted the system to do more than just calculate numbers and it should explain them in a clear and helpful way.

As we got further into development, we adjusted our approach in a few areas. Instead of using

320 XGBoost or LightGBM for the demand forecasting
321 part, we switched to Prophet. This decision was
322 based mostly on practicality. Prophet works
323 especially well for time series forecasting, like
324 predicting future sales by month. It's easier to use
325 than the tree-based models we initially planned,
326 and it handles seasonal patterns and trends
327 automatically. That made it a good fit for the kind
328 of grouped data we were working with.

329 A notable advancement in comparison with our
330 initial proposal is the implementation of two
331 distinct user interface options: a desktop graphical
332 user interface and a web-based interface. Both
333 versions allows the user to select a country and a
334 month, after which the system processes the data,
335 generates a demand forecast, employs a language
336 model to create a structured business report, and
337 finally saves the result as a PDF. This dual interface
338 approach offers a high degree of flexibility in terms
339 of the system's application, depending on user
340 preference or technical configuration.

341 Part of the original plan that was implemented was
342 the use of a large language model. We connected
343 Ollama to the system and used the LLaMA 3.2
344 model to convert forecast data into a clear,
345 structured report. The generated text is included in
346 the final PDF file, making the results easier to
347 understand for non-technical users. This part of the
348 project worked well and largely met our original
349 expectations.

350 We ultimately decided not to implement a
351 profitability classifier or market basket analysis,
352 even though these were part of our original
353 concept. Given the time constraints and the
354 complexity involved, we chose to focus our efforts
355 on demand forecasting and report generation—
356 features that were more central to the project's main
357 goal. Financial attributes such as profit and price
358 were also left out in this phase, as they were more
359 relevant to profitability assessments than to the
360 kind of demand predictions we were prioritizing.

361 In the end, these choices proved to be reasonable.
362 With Prophet, it enabled us to quickly generate
363 meaningful forecasts, gives us more time to focus
364 on integrating the language model and building a
365 reliable reporting system. Although the final
366 version does not include all the features we initially
367 planned, it provides a clear and functional
368 prototype of the system. The core idea has been
369 fully implemented and is working well, and the
370 features we left out can still be added in future
371 iterations without the need for a complete redesign.

372 In conclusion, the project ended up in a different
373 place than we first expected, but it still achieves the
374 original goal: helping businesses plan better by
375 turning raw data into useful information. We
376 learned that it's important to be flexible, especially
377 when working with real data and limited time. The
378 tools we used changed, and some features were
379 made simpler or delayed, but we always kept the
380 main goal in mind. The end result is a working
381 system that makes forecasts, explains them clearly,
382 and provides a good starting point for further
383 development.

384 References

- 385 Chapman, P., Clinton, J., Kerber, R., Khabaza, T.,
386 Reinartz, T., Shearer, C. & Wirth, R. (2000). CRISP-
387 DM 1.0: Step-by-step data mining guide. SPSS Inc.
- 388 Taylor, S. J., & Letham, B. (2018). Forecasting at
389 Scale. *The American Statistician*, 72(1), 37-45.
- 390 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J.,
391 Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017).
392 Attention is all you need. *Advances in Neural*
393 *Information Processing Systems*, 30, 5998-6008.
- 394 Online-Website: <https://ollama.com/library>