# Capstone Project Final Report

***Optimizing FashionWorld's Retail Packaging Processes***

---

**SUBMITTED BY**

**Group Name:** *Capgemini 3*
**Program:** *Master in Business Analytics & Big Data*
**Institution:** *IE University – School of Science & Technology*
**Mentor:** *Angela Chen*
**Submission Date:** *July 6, 2025*


---

**SIGNATURES**

*I hereby certify that this report and the accompanying presentation is my own original work in its entirety, unless where indicated and referenced.*
Maria Do Carmo Vieira De Fonseca de Brito e Abreu

*I hereby certify that this report and the accompanying presentation is my own original work in its entirety, unless where indicated and referenced.*
Paul Morcos Douaihy

*I hereby certify that this report and the accompanying presentation is my own original work in its entirety, unless where indicated and referenced.*
Sarina Ratnabhas

*I hereby certify that this report and the accompanying presentation is my own original work in its entirety, unless where indicated and referenced.*
Enrico Miguel Cordero Tajanlangit

*I hereby certify that this report and the accompanying presentation is my own original work in its entirety, unless where indicated and referenced.*
Joy Zhong

## TABLE OF CONTENTS

## EXECUTIVE SUMMARY

Fashion World Retail (FWR), a global leader in the textile retail industry, engaged Capgemini to design an AI-powered solution to modernize and optimize its packaging operations. The initiative was launched in response to mounting challenges including fragmented packaging standards, operational complexity, and rising costs driven by manual workflows and siloed data systems.

We as consultants from Capgemini have delivered a fully integrated, data-driven solution aligned with FWR's strategic priorities: operational efficiency, regulatory compliance, and sustainability. Leveraging advanced machine learning and predictive modelling, the solution enhances decision accuracy, automates key packaging processes, and delivers real-time insights across the supplier network.

Through intelligent recommendations based on product attributes, seasonality, and shipment profiles, the system replaces reactive supplier behaviour with a proactive, standardized approach. The result is a significant reduction in packaging inefficiencies, improved responsiveness, and greater control over packaging compliance and cost.

Crucially, the solution is designed for scalability, adaptability, and continuous improvement, enabling Fashion World to embed long-term automation into its supply chain and sustain a competitive edge in an increasingly dynamic market.

---

## BACKGROUND & CONTEXT

Fashion World Retail (FWR) is a global leader in the textile industry, operating in over 30 countries with a vast supplier and distribution network. As a major player, FWR helps shape industry standards and must continuously adapt to evolving consumer expectations, regulatory pressures, and sustainability demands.

Fashion World's packaging process plays a strategic role in their supply chain operations, from cost control and logistics to environmental impact and brand perception. However, their current operations face key challenges that are hindering them from full efficiency and scalability.

- ***Manual, fragmented packaging processes*** that lead to inconsistent standards and costly inefficiencies.
- ***Scattered data systems*** that slow down decision-making and reduce packaging traceability.
- ***Reactive supplier behaviour***, increasing delays and hindering strategic packaging optimization.
- ***High operational costs***, particularly from outdated and non-standardized packaging practices.

Given these pain points, optimizing packaging is not just a cost-saving opportunity, it is a strategic necessity, as these issues are driving up costs, reducing sustainability efforts, and creating operational bottlenecks. Our solution directly addresses these pain points through packaging automation, standardization, and real-time data intelligence.

As Capgemini consultants, we were engaged to lead a data-driven transformation of FashionWorld's packaging operations, from manual and reactive, to automated, intelligent, and scalable. By applying our expertise in AI, machine learning, cloud architecture, and supply chain analytics, our mission is to design a tailored solution that not only solves today's inefficiencies, but sets the foundation for continuous improvement.

Our approach followed a five step consulting-style, end-to-end methodology that ensures cross-functional alignment and seamless integration into FashionWorld's operations:

*DISCOVERY → ANALYSIS → MODELING → PROTOTYPING → DELIVERY*

Through this approach, we have designed a solution that is designed to deliver tangible, scalable value through four core components:

i. ***Centralized & Clean Packaging Data*:** Establishing a reliable foundation for analytics and automation.
ii. ***Automated Packaging Density Report Generator*:** Replacing manual calculations with ML-driven recommendations.
iii. ***Supplier KPI Dashboard*:** Providing real-time insights into supplier compliance and performance
iv. ***Scalable Cloud-Ready Architecture*:** Ensuring performance, flexibility, and readiness for future expansion.

Together, these deliverables will enable FashionWorld to achieve packaging excellence through operational efficiency, sustainability gains, and digital leadership.

---

## PHASE 1: DISCOVERY

This phase focused on identifying key internal stakeholders, understanding their primary pain points, and outlining data requirements aligned with their operational goals. Our discovery was grounded in structured conversations, document reviews, and collaboration with domain experts from FashionWorld's supply chain, quality, data, and procurement teams. We have identified the following main stakeholders and their needs as a reference starting point to understand what our solution needed to deliver in order to ensure that each functional team receives relevant insights in a format that supports their day-to-day and strategic decision-making:

| | |
|---|---|
| *Operations & Logistics* | End-to-end visibility of order status, supplier responsiveness, and validation timeline. |
| *Data Engineers* | Traceability of errors/issues with incoming goods, focus on reprocessing and pattern recognition. |
| *Data Analysts* | Machine Learning |
| *Procurement Team* | Supplier compliance, cost management, vendor performance tracking. |
| *Business Leadership & Management* | Overall efficiency, strategic decision-making, value from digitization. |

Now that our stakeholder desires are aligned, we proceeded to begin a comprehensive analysis on understanding their current operational environment, data landscape, and end-to-end packaging process in order to secure a strong foundation for our solution.

***Data Landscape***

In order to conduct a deep analysis into their current operations, we had to begin by understanding the data itself by performing a thorough exploratory data analysis (EDA). This step was key in order to better understand the structure, quality, and patterns within the packaging-related datasets provided by FashionWorld - Density Report, Product Attributes, Supplier Scorecard, & Historical Incidents.

***Data Cleaning***

Our initial discovery revealed that packaging data was fragmented across non-centralized sources, Excel files, emails, PDFs, and manual logs, lacking standardization and consistent formatting, hindering accessibility and collaboration across teams. Therefore, our first step was to perform comprehensive data cleaning to ensure accuracy and consistency across datasets before proceeding with any further analysis. The following steps were taken for each report to prepare the data for reliable integration and insight generation:

i. **Density Report:**
- Removed all ProductReference entries not matched in the ProductAttributes table to eliminate inconsistencies.
- Standardized ProductReference values by stripping trailing characters (e.g., "X") to align with base product codes.
- Normalized supplier names by fixing casing and applying a standard naming convention.
- Dropped records with unrealistic ProposedUnitsPerCarton values (e.g., -3, 0, 9999).
- Removed rows missing the ProposedFoldingMethod and harmonized naming across valid entries.
- Standardized formats in the ProposedLayout column.
- Cleaned and unified PackagingQuality labels for consistent model training.
  - Final dataset reduced to **478,846 rows** (4.23% reduction)

ii. **Supplier Scorecard:**
- Cleaned column names by removing spaces and special characters to improve query compatibility.
- Standardized SupplierName values by stripping whitespace, converting to lowercase, and mapping known naming inconsistencies.

**iii. Product Attributes:**
- No cleaning was required.
- All data integrity checks were passed for key fields: ProductReference, ProductName, GarmentType, Material, and Size.

**iv. Historical Incidents:**
- Removed unmatched ProductReference values (e.g., codes from PRD00 to PRD09) due to inconsistencies.

- Standardized ProductReference entries by removing trailing "X" for consistency with base product codes.
- Cleaned SupplierName values by stripping whitespace, normalizing casing, and correcting known variations and typos

***Exploratory Data Analysis (EDA)***

Now that our data is clean, we proceeded with the EDA phase. Our analysis revealed several critical insights & inefficiencies into the relationship between product attributes, packaging methods, and supplier performance:

- **Raw incident volume is misleading**: High incident counts alone do not reflect poor supplier performance especially for high-volume suppliers. Normalizing by order volume is essential to fairly assess efficiency and error rates.

- **Systematic risk in layout + method combinations**: Certain combinations consistently correlate with poor packaging outcomes. Notably, Method 3 appears across a majority of high-risk cases regardless of layout, suggesting a structural flaw or misuse. Layout E is also frequently associated with problematic pairings.

- **Product complexity impacts packaging success**: Garments like coats, suits, and hoodies show disproportionately high failure rates, particularly when paired with high-risk configurations. This indicates a need for more tailored handling procedures for bulky or structured garments.

- **Material matters**: Wool, denim, linen, and silk are consistently overrepresented in poor packaging incidents, highlighting that delicate or less flexible fabrics may require specialized packaging setups.

- **Packaging success varies widely**: Bad packaging rates ranged from 16% (best) to 34% (worst), a nearly twofold difference. This confirms that packaging configuration can significantly influence overall packaging quality.

- **Supplier benchmarking opportunities**:
  - **Suppliers D and H** exhibit persistent underperformance and may require intervention or additional training.
  - **Supplier G** demonstrates strong performance at lower volumes and could serve as a benchmark for best practices.

While our exploratory analysis uncovered several clear patterns and actionable insights, it also surfaced important limitations and open questions that warrant deeper investigation:

- **Interaction effects**: Deeper analysis is needed to explore how issue types interact with product attributes (e.g., garment type, material, size) and packaging elements (e.g., layout, folding method).

- **Data linking challenge**: Currently, historical incidents cannot be reliably linked to other datasets due to **missing or inconsistent join keys**—limiting longitudinal or root cause analysis.

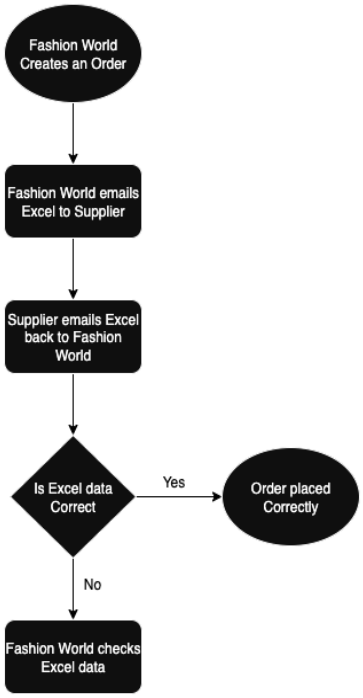
- **Class imbalance**: Some issue types or packaging failures are underrepresented in the data, which could bias model training or skew performance metrics.

- **Supplier-level anomalies**: Additional investigation into outlier behaviors, especially in suppliers with extreme scores (both high and low), may reveal best practices or systemic risks.

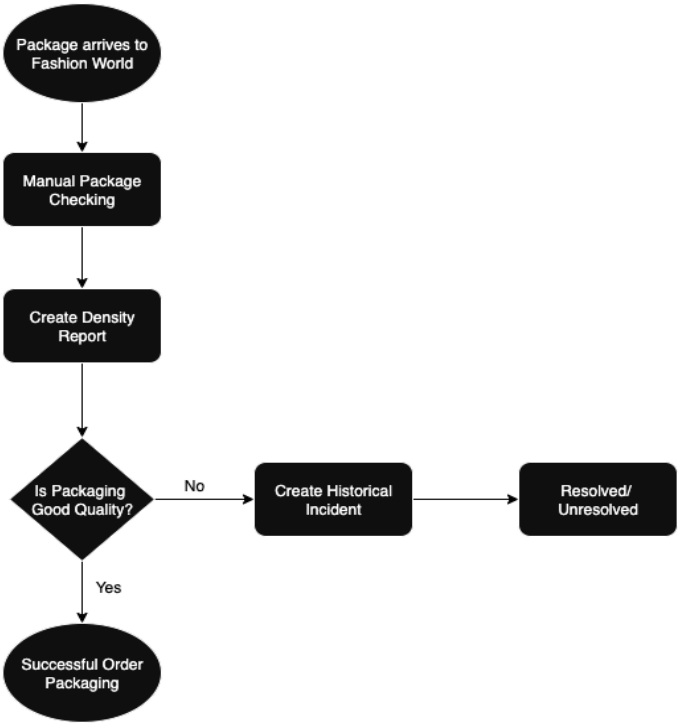
***Packaging Process***

Next, in order to effectively address the root causes of inefficiencies in FashionWorld's packaging operations, we then sought to gain a deep understanding of their end-to-end ordering and packaging workflow. This discovery phase was guided by stakeholder interviews and informed by analysis of the available datasets. Through this process, we reconstructed the operational pipeline into two key streams:

**i. Order Creation**




**ii. Order Delivery**




Mapping these workflows provided a holistic view of FashionWorld's packaging process, enabling us to identify key bottlenecks including a heavy reliance on spreadsheets, fragmented data sources, reactive quality checks, and the absence of real-time feedback loops. Incident records were handled retrospectively, often disconnected from order-level packaging data, making root cause analysis difficult and time-consuming.
This lack of automation and centralized tracking resulted in delays, inconsistencies in packaging decisions, and limited visibility into supplier performance, ultimately contributing to increased operational costs and missed sustainability opportunities. Recognizing these gaps, we defined our solution objectives around four key goals:

i. ***Data centralization and cleaning*** to unify packaging information across sources
ii. ***Real-time ingestion and processing*** to replace delayed, manual workflows
iii. ***Automated decision support via machine learning***, particularly for packaging quality

iv. ***KPI-driven dashboards*** tailored to each stakeholder team to ensure traceability, transparency, and accountability

These priorities informed the design of our scalable "to-be" architecture, which will automate the ingestion, processing, and serving of data, replacing manual effort with intelligent, real-time decision-making and performance monitoring.

---

## PHASE 2: ANALYSIS

Following our exploratory data analysis (EDA), we transitioned into the analytical phase, where we aligned key metrics with stakeholder needs to uncover the most pressing inefficiencies and opportunities within FashionWorld's packaging process. This step allowed us to move from surface-level observations to measurable insights that directly informed our solution design.

Using consolidated packaging, shipment, and product attribute data, we established a set of performance KPIs mapped to the needs of each stakeholder group. These KPIs served as the basis for identifying gaps, measuring supplier performance, and uncovering recurring issues in the packaging workflow:

| *Stakeholder* | *Relevant KPIs* | *ANALYSIS* |
|---|---|---|
| **Operations & Logistics Team** | ▪ % Orders Approved On-Time<br>▪ Avg. Time to Validate Order<br>▪ % Orders With Manual Intervention<br>▪ Order Stage Breakdown | These metrics revealed frequent delays in order validation and a high incidence of manual interventions. This pointed to inconsistent data handoffs between FashionWorld and suppliers, as well as the need for automation in tracking order status and packaging transitions. |
| **Quality Team** | ▪ % Orders Requiring Reprocessing<br>▪ Incident Types Breakdown<br>▪ Recurrent Supplier Issues<br>▪ Anomaly Rate per Supplier | Our analysis surfaced clusters of recurring errors tied to specific suppliers and products. Visualization of incident patterns enabled root cause identification and highlighted opportunities for supplier coaching or process redesign. |
| **Data Team** | ▪ Model Version Performance Score<br>▪ Deployment Frequency<br>▪ Feature Drift Score<br>▪ Drifted Features Count<br>▪ Performance Deviation<br>▪ Prediction Latency (ms)<br>▪ Best vs. Baseline Metric Delta<br>▪ Run Success Rate<br>▪ Hyperparameter Impact<br>▪ Experiments Metric | These indicators ensured we could track the long-term performance and adaptability of the machine learning models powering packaging recommendations. Identifying model drift or underperformance early is crucial for sustaining long-term value. |
| **Procurement Team** | ▪ Supplier Compliance Score<br>▪ Average Cost per Incident<br>▪ Bad Packaging Rate<br>▪ Supplier Ranking by KPI | The procurement analysis highlighted a small set of underperforming suppliers contributing disproportionately to quality issues and cost escalations. This created a |

| | | |
|---|---|---|
| | | clear opportunity for targeted supplier engagement and contract revision based on data-backed evidence. |
| **Business Leadership & Management** | ▪ Global Reprocessing Rate<br>▪ Average Validation Time Trend<br>▪ Total Monthly Cost Impact<br>▪ Compliance Trend Over Time | Executive-level metrics validated the broader business impact of operational inefficiencies and guided prioritization of initiatives. By tracking trends over time, we were able to project cost savings and ROI for the proposed solution. |

To effectively measure the impact of our solution, we will leverage these KPIs as baseline metrics to monitor performance and track progress over time. In addition, we have developed a fully integrated, real-time dashboard tailored to each stakeholder group. These dashboards will provide automated, visual insights into relevant processes and metrics, ensuring that every team has continuous visibility into performance, risk areas, and improvement opportunities.

---

## PHASE 3: MODELLING

With a strong business foundation and a consolidated dataset prepared during the discovery and analysis phases, we moved into the modeling phase with the goal of developing a predictive system capable of identifying defective packaging before it reaches the customer or distribution center. At this stage, our focus was twofold: first, to build a high-performing machine learning model aligned with FashionWorld's operational priorities, and second, to ensure that the model would be interpretable, scalable, and easily integrated into the real-time decision-making process. Our modeling strategy followed a rigorous, step-by-step approach, from defining the business problem and structuring the data, to selecting appropriate algorithms and tuning for performance and operational risk. What follows is a breakdown of how we transformed raw data into a production-ready predictive tool that supports FashionWorld's shift from reactive quality control to proactive quality assurance.

### ***Business Understanding***

The central objective of this project was to develop a machine learning model that could predict defective packaging before the product reaches a customer or distribution node. At FashionWorld, minimizing the occurrence of bad packaging is critical to ensuring customer satisfaction, sustaining brand equity, and reducing operational costs related to returns, reprocessing, and delayed shipments. Given the high volume of daily shipments and the relatively small proportion of defective units, this challenge requires a predictive system that is both highly sensitive to rare events and operationally scalable. To meet this objective, we framed the problem as a supervised binary classification task in which each packaging unit was labeled as either "Good" (0) or "Bad" (1). Our business priority was not only to maximize predictive performance but also to ensure the solution could support real-time decision-making and be interpretable enough to guide corrective action. Most importantly, our modeling strategy was explicitly aligned with the understanding that false negatives(failing to detect a defective package)represent a significantly greater operational risk than false positives.

***Data Understanding***

Our data originated from four cleaned datasets: density reports, product attributes, supplier scorecards, and historical incidents. Each dataset contributed unique contextual signals. The density reports captured physical packaging events and dimensions. Product attributes provided static product-level metadata such as garment type, size, and material. Supplier scorecards offered a temporal record of supplier performance, while the historical incidents dataset logged packaging-related operational failures over time. Using structured join logic, we integrated all four sources into a unified modeling table. Merges were performed using common keys such as ProductReference and SupplierCode, with validations to ensure proper cardinality. We prioritized many-to-one relationships to preserve the granularity of packaging events. Duplicate and redundant columns were eliminated during this merge process to avoid information leakage and multicollinearity. The resulting dataset offered a high-resolution view of packaging quality outcomes, with sufficient operational, supplier, and product-level context for effective predictive modeling.

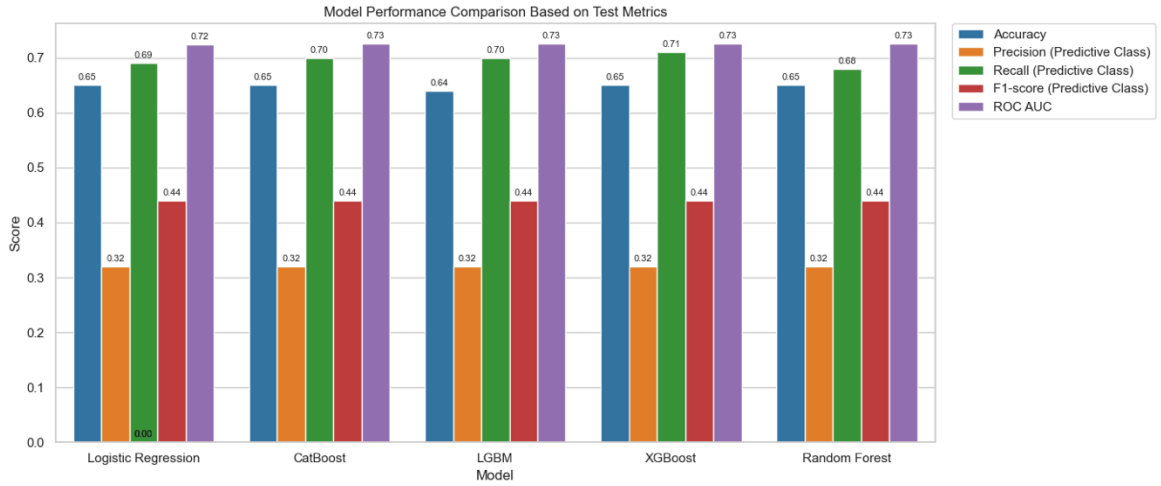
***Data Preparation***

Our preprocessing phase focused on preparing features for modeling while retaining maximum interpretability. We applied consistent encoding practices across the entire pipeline. All categorical variables were encoded using OrdinalEncoder, which ensured compatibility with gradient boosting and ensemble models. For CatBoost, native categorical processing was retained by passing categorical column names directly, in accordance with algorithm-specific best practices. We engineered three domain-informed binary flags that captured supplier and delivery risk: is_low_delivery_perf, is_high_bad_rate_supplier, and is_bad_incident_history. These indicators were constructed using mean-based thresholds rather than percentile cutoffs, ensuring that they remained interpretable and meaningful regardless of class balance shifts. Several features were excluded after careful evaluation. Specifically, is_peak_season, is_weekend_reported, is_high_unit_density, and is_large_package were removed due to low variability or weak correlation with the target variable. The final dataset was clean, enriched, and ready for fair model comparison.

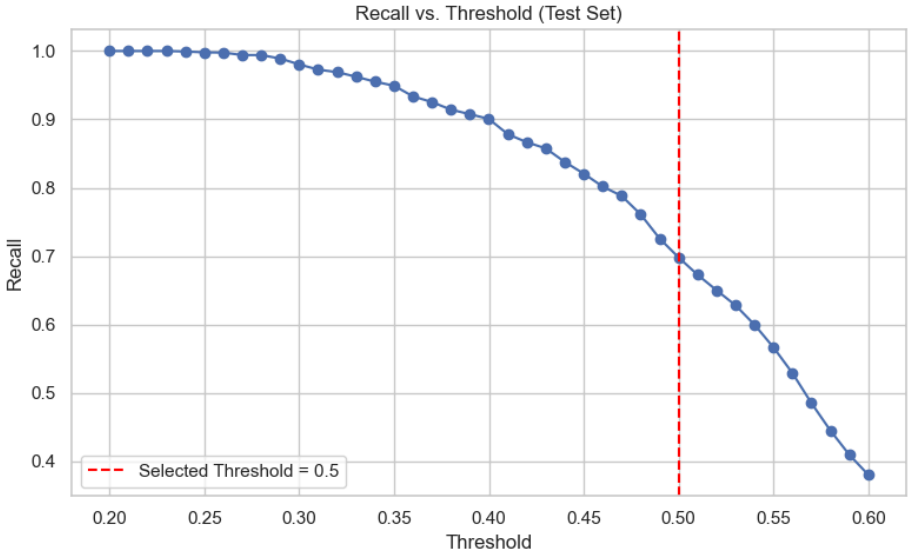
***Modelling Strategy***

We compared five supervised learning models: Logistic Regression, CatBoost, LightGBM (LGBM), XGBoost, and Random Forest. To ensure fairness, all models were trained and validated using the same encoded dataset and evaluated using a consistent five-fold StratifiedKFold cross-validation strategy. The primary evaluation metric during this initial phase was ROC AUC, chosen for its robustness to class imbalance and its ability to assess discriminative power across various probability thresholds. Hyperparameter tuning was performed using GridSearchCV, with dedicated parameter grids tailored to each algorithm's architecture while keeping grid complexity comparable. All models respected their encoding constraints and architectural properties. For example, Logistic Regression was provided with strictly numeric inputs, while CatBoost retained raw categorical features. These efforts ensured that model comparisons were both methodologically sound and practically valid.

## ***Model Selection and Threshold Strategy***

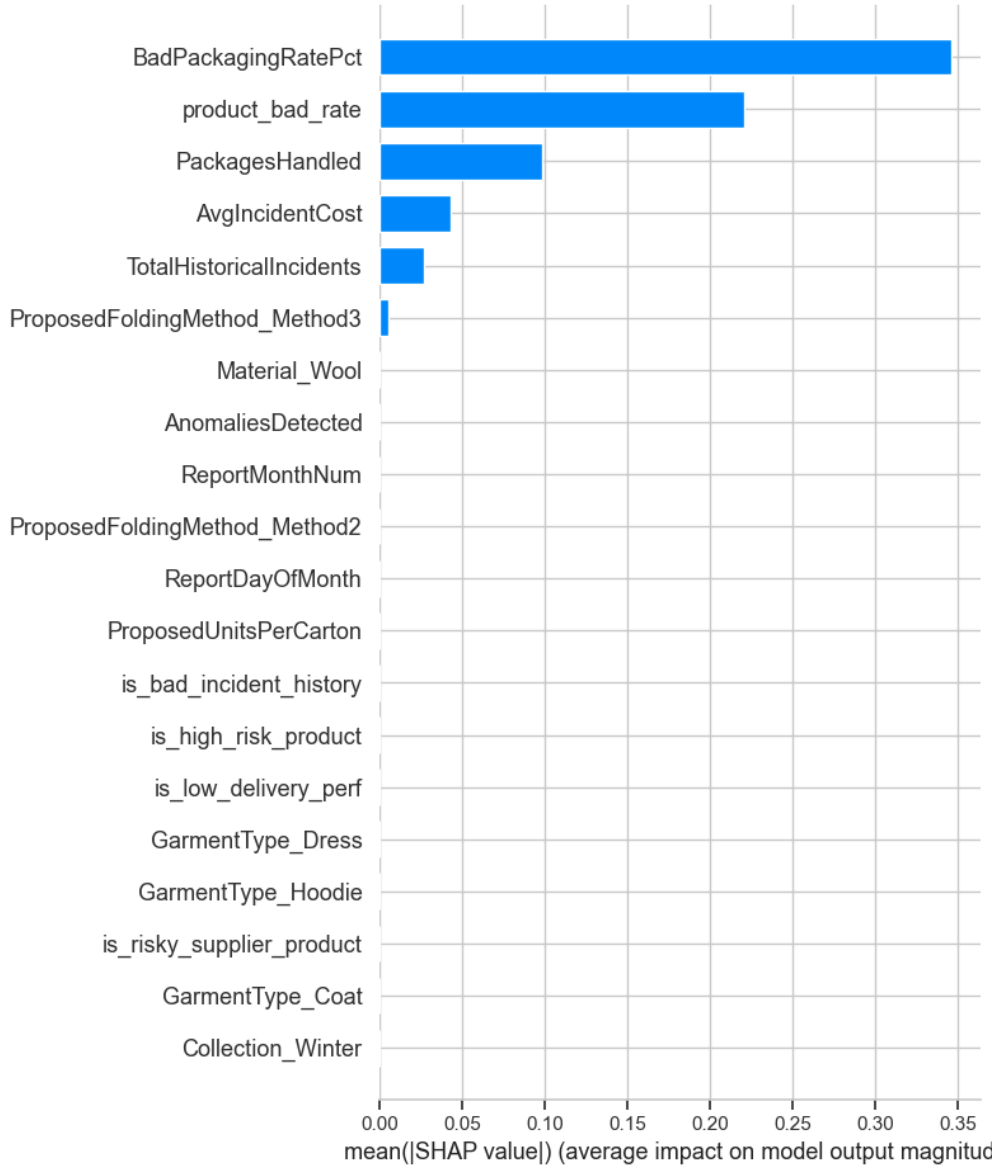



Our comparative evaluation revealed that XGBoost consistently outperformed the other models in capturing the minority class. Despite Random Forest achieving the highest overall accuracy, its recall on defective packaging was extremely low, rendering it unreliable in operational contexts. CatBoost, LightGBM, and Logistic Regression performed reasonably well but failed to match XGBoost's recall, which reached 0.71 on the test set. After selecting XGBoost as the final model, we re-optimized it with a new scoring focus: recall for the predictive class. During this final tuning phase, we applied a custom scoring function within GridSearchCV that averaged recall across thresholds ranging from 0.3 to 0.5. This threshold range was selected to reflect FashionWorld's operational preference for early and sensitive detection of defective units.




Subsequent analysis revealed that this newly threshold-calibrated model outperformed its earlier version in terms of minority class recall. At a threshold of 0.40, the model achieved a recall as high as 0.90, while still maintaining acceptable overall accuracy and a stable ROC AUC score. These results confirmed the effectiveness of our tuning strategy in aligning the model's behavior with FashionWorld's operational risk tolerance.

***Interpretability and Operational Integration***




To ensure that the model's predictions could be explained and acted upon by FashionWorld's operational teams, we applied SHAP (Shapley Additive Explanations) to the final XGBoost model and analyzed the raw feature importance generated by the same model. These dual interpretability strategies allowed us to validate that the model was prioritizing variables that were not only statistically significant but also operationally meaningful.

The top-ranking features in the model included BadPackagingRatePct, PackagesHandled, and product_bad_rate. These variables directly reflect the historical performance and scale of both suppliers and individual products. BadPackagingRatePct captures supplier-level packaging failure rates over time, serving as a reliable indicator of supplier consistency. PackagesHandled represents operational volume and exposure, while product_bad_rate highlights failure trends at the Product Reference level. Their prominence within the model confirms that historical and scale-based risk signals are highly predictive of future quality issues.

Beyond supplier performance, material and packaging methods such as ProposedFoldingMethod_Method3 was also found to be highly influential. These findings were reinforced by our earlier exploratory data analysis. Specifically, we observed that heavier garments, especially those made from wool, showed a significantly higher rate of packaging failures. Further analysis revealed that certain folding and layout methods were disproportionately applied to these heavier products. When combined with the structural demands of wool materials, these methods resulted in increased vulnerability to packaging defects. These patterns were consistent with insights shared by warehouse teams and suggest that the model is correctly capturing real-world fragility risks embedded in packaging routines.

Together, these interpretability outputs strengthen confidence in the model's alignment with operational reality. They also provide direct pathways for intervention. Quality assurance teams can use these insights to review and retrain folding techniques for wool-based products, monitor high-risk supplier behavior, and adapt layout configurations for heavier shipments. By enabling transparent predictions and traceable rationale, the model supports FashionWorld's goal of transitioning from reactive quality control to proactive quality prevention.

***Limitations, Mitigations, & Future Recommendations***

While the current implementation has demonstrated strong predictive capabilities and strategic alignment with FashionWorld's operational needs, it is important to acknowledge certain limitations that could affect long-term performance and generalizability.

A key concern lies in the quality of the target labels used to train the model. At present, we have limited visibility into the exact standards or operational definitions applied during the labeling of packaging outcomes. There is no formal validation pipeline ensuring that "Bad" labels were assigned under consistent and auditable criteria. This uncertainty introduces noise into the learning process and may partially explain why the boosting-based models, XGBoost, LightGBM, and CatBoost, all achieved comparable performance metrics. It is possible that with cleaner and more consistently labeled data, clearer performance differentials would emerge.

To mitigate this, we strongly recommend that the model be launched as part of a broader MLOps pipeline that includes automated retraining on a curated dataset. This cleaner dataset is expected to become available through our recently implemented data architecture, which now incorporates stricter validation, deduplication, and governance protocols. As higher-quality data flows into the pipeline, we anticipate improved stability in performance metrics and greater confidence in the model's generalization capabilities.

From a modeling perspective, LightGBM remains a promising alternative. It achieved results similar to XGBoost while offering significantly faster training and lower memory usage. These characteristics make it highly suitable for scenarios requiring frequent retraining or deployment at scale. CatBoost, although more computationally intensive, continues to offer practical advantages, particularly its ability to handle categorical variables without extensive preprocessing. This makes it attractive for scenarios where data pipelines are still evolving or require greater flexibility.

Additionally, we recognize that due to limited computational resources, our hyperparameter tuning relied on relatively compact grid searches. In future iterations, we recommend

allocating dedicated infrastructure, either via cloud-based compute clusters or internal GPU resources, to enable broader search spaces and more robust optimization strategies, such as Bayesian optimization or random search. These enhancements would allow for deeper exploration of each model's capacity and further refine performance, especially on the minority class.

Finally, we suggest incorporating a model monitoring framework to continuously track recall, precision, and ROC AUC over time, flagging any performance drift that may result from changes in upstream data quality or packaging processes. This will ensure that the predictive system remains accurate, stable, and aligned with evolving business needs.

Through a rigorously structured CRISP-DM approach, we developed a high-performing, interpretable model to predict defective packaging. By systematically evaluating five algorithms, applying domain-informed feature engineering, and integrating both performance and operational considerations into our selection and thresholding strategy, we delivered a solution aligned with FashionWorld's quality control objectives. The final XGBoost model, combined with SHAPE-based interpretability, now serves as a trusted tool for proactive packaging quality assurance. The system is ready for operational deployment and can be extended with real-time scoring and active learning modules.

---

## PHASE 4: PROTOTYPING

Our proposed solution architecture transitions FashionWorld's packaging quality management from a manual, reactive, and fragmented system to a centralized and automated pipeline leveraging Azure technologies.

### ***Why Azure ML***

Azure Machine Learning offers a robust, enterprise-grade platform for managing the full ML lifecycle, from data ingestion to model deployment and performance monitoring. Its integration with Azure Blob Storage allows seamless access to packaging reports and ensures training data is always current.

- **Data Connectivity and Automation:** Its native integration with Azure Blob Storage ensures seamless access to newly arriving data, such as packaging reports stored in the "orders" container, enabling training pipelines to operate on the most recent and validated information. Azure Event Grid enables event-driven retraining, ensuring adaptability to business needs and data freshness.

- **Pipeline Management and Experimentation:** Azure ML Pipelines support modular, version-controlled workflows defined in Python or YAML, making the entire training process reproducible and clear. The platform's built-in hyperparameter tuning engine supports grid search, random search, and Bayesian optimization at scale. Every model run, along with its parameters and metrics, is automatically logged using MLflow APIs, eliminating the need for separate tracking infrastructure.

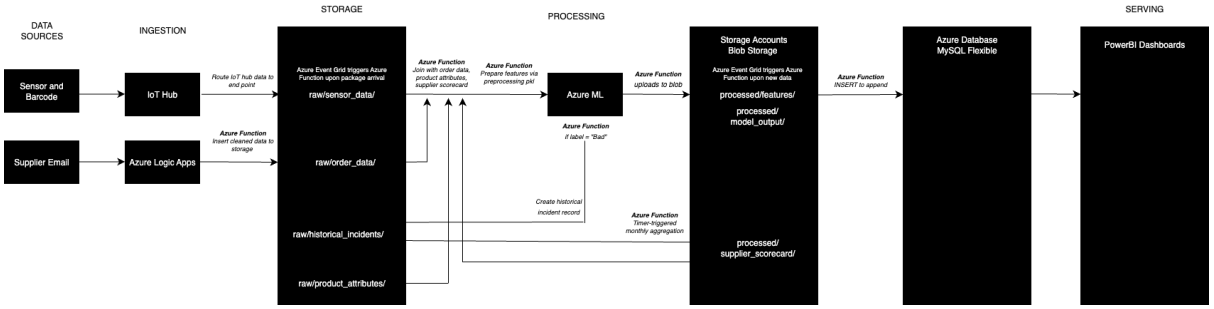
- **Model Performance Monitoring:** Real-time monitoring for critical metrics such as recall, precision, and ROC AUC is supported natively. This is essential for FashionWorld, where the cost of false negatives in defective packaging prediction is high. Azure ML

also supports SHAP-based explainability, model versioning, and drift monitoring to ensure continued alignment with operational and data quality standards.

- **MLOps Full Solution Tool Package**: Azure ML offers a unified MLOps solution that eliminates the need for third-party tools like Hydra, MLflow, or Weights & Biases. It provides robust support for configuration management, retraining, experiment tracking, and decision-grade transparency, all of which are essential for maintaining long-term model performance in a real-world production environment.

## ***To-Be Architecture w/ Azure***

The Azure technology services we selected for our solution is compatible with FashionWorld's existing architecture, minimizing disruption and maximizing operational efficiency:




Using Azure Blob Storage, Event Grid, IoT Hub, Logic Apps, Functions, SQL Database, and Power BI, the architecture creates a single source of truth for packaging, supplier, and incident data, enabling real-time insights and automated operational workflows across packaging quality management.

## ***Solution Objective 1: Eliminates Manual, Time-Intensive Processes***

Azure Logic Apps replace the manual back-and-forth of Excel files by automatically sending suppliers pre-filled packaging instructions, including the optimal folding method and layout, based on best practices and benchmarks. The incoming data is validated automatically for data quality and completeness. When data quality issues are flagged, suppliers are automatically re-emailed for corrections. Finally, clean, structured order data is inserted into the system via Azure Function Apps. This reduces turnaround time, minimizes manual data quality checks, and ensures consistent data ingestion across all suppliers.

Importantly, this design preserves suppliers' existing manual Excel workflows, placing the responsibility for data quality validation on FashionWorld. This ensures high data integrity while maintaining strong supplier relationships and minimizing resistance to the new system.

In addition, IoT sensors are proposed at receiving stations to detect package arrivals automatically to improve traceability and capture key package characteristics such as dimension and weight in real-time. These sensors trigger immediate ingestion and initiate the machine learning inference process, eliminating the need for manual package arrival logging, enabling stakeholders to trace discrepancies, and monitor supplier adherence.

Once a new order instance is uploaded to the designated order blob in the cloud, it automatically triggers the MLOps pipeline. This incoming data row is then passed through the preprocessing pipeline, where all necessary transformations—such as encoding and metric updates—are applied. The preprocessed row is subsequently fed into the trained machine learning model, which predicts whether the packaging quality of the incoming order is likely to be "Good" or "Bad". This prediction is instantly visualized on a Power BI dashboard and, in the case of a "Bad" prediction, an automated alert is sent to the Quality Assurance (QA) team, prompting them to inspect the package upon its arrival.

Previously, this process was entirely manual, requiring QA personnel to inspect every incoming package regardless of risk. The transition to a fully automated, cloud-hosted pipeline not only updates the dataset in real-time—using Python functions to recalculate supplier performance metrics and other relevant indicators—but also enables the system to improve over time as more accurate data is accumulated. Ultimately, this Azure-based ML inference workflow significantly reduces human intervention, enhances operational efficiency, and decreases both labor costs and avoidable lead times.

***Solution Objective 2: Automated, Consistent Performance and Quality Tracking***

The proposed architecture automates the creation of historical incident records whenever a package is flagged as “Bad”. This ensures data consistency between operational quality labels and incident records. Simultaneously, supplier scorecards are transformed into time-triggered monthly aggregations of incidents, bad packaging rates, and delivery metrics, eliminating manual compilation.

***Solution Objective 3: Dashboard Explanation***

To ensure our solution delivers measurable impact and meets the unique needs of each stakeholder group, we designed a suite of tailored dashboards as the primary interface for data consumption. These dashboards transform complex datasets into actionable insights, enabling teams to monitor KPIs, track real-time performance, and make informed decisions across the packaging workflow. Below is an overview of how each team will leverage their respective dashboards to support their specific responsibilities and goals:

**i. Operations & Logistics Team:**

- ***NEED*:** This team needs full visibility into order processing to monitor status updates, supplier responsiveness, and validation timelines.
- ***DASHBOARD*:** We’ve built a Power BI dashboard focused on tracking the order pipeline in real time. It will include visualizations like stage transition breakdowns and validation duration heatmaps to help them pinpoint bottlenecks and improve throughput.

**ii. Quality Team:**

- ***NEED***: The Quality Team is primarily concerned with traceability and issue resolution in incoming goods. They require detailed views on reprocessing events, anomaly rates, and patterns in recurring supplier issues.
- ***DASHBOARD***: Their Power BI dashboard will offer detailed charts on incident types, root cause pattern recognition, and supplier performance metrics, enabling fast root cause identification and quality improvements.

**iii. Data Team**

- ***NEED***: For the Data Team, the focus is on tracking the performance and reliability of machine learning models used in automation or prediction.
- ***DASHBOARD***: We leverage Azure ML Service to surface reports on model health, training schedules, and deployments. A dedicated model health dashboard is being scoped to give them real-time visibility into key indicators like model drift, training cycles, and performance metrics across models.

**iv. Procurement Team:**

- ***NEED***: This team needs data to drive supplier performance evaluations and cost control.
- ***DASHBOARD***: We leverage Power BI to create dashboards that consolidate compliance scores, incident-related costs, packaging error rates, and supplier rankings. These dashboards are designed for easy comparison across vendors and highlight cost drivers to support strategic sourcing decisions.

**v. Business Leadership / Management:**

- ***NEED***: Leadership is focused on high-level efficiency, value realization, and strategic trends.
- ***DASHBOARD***: Their dashboards consolidate global KPIs such as reprocessing rates, cost impact summaries, and compliance over time. The executive views are designed to support digital transformation narratives and operational governance, offering a clear view of whether the organization is improving against its core KPIs.

---

## Phase 5: DELIVERY

With a fully developed prototype in place, we transitioned into the delivery phase by finalizing a scalable solution package and deployment roadmap, ensuring FashionWorld is equipped for seamless enterprise-wide implementation. Our architecture was intentionally designed with operational compatibility and organizational scalability in mind, using native Azure services to integrate smoothly with FashionWorld's existing cloud environment while minimizing disruption to suppliers and stakeholders. To ensure seamless integration, we've created the following documents to help guide technical deployment, standardize data onboarding, manage user access, and support organization-wide adoption of the solution.

### ***Deployment Blueprint***

**OBJECTIVE:** To provide technical teams with a step-by-step guide for deploying the proposed solution architecture in FashionWorld's Azure environment.

**i. STEP 1: Environment Configuration Guidance**

- Subscription setup and resource group creation
- Networking and security considerations (VNets, firewalls, identity access)
- Azure regions for performance and compliance

**ii. STEP 2: Cloud Resource Provisioning**

- Azure Blob Storage: structured directory for raw and processed data
- Azure Event Grid: trigger logic for pipeline automation

- Azure IoT Hub: sensor integration for real-time tracking
- Azure Logic Apps: supplier data ingestion workflows
- Azure Functions: custom processing logic and model inference
- Azure SQL Database: structured data storage for dashboards
- Azure ML: model deployment and monitoring
- Power BI: dashboard setup and gateway configuration

**iii. STEP 3: CI/CD Recommendations**

- Organize full code repository
- Use YAML for all Azure ML jobs and environments
- Setup Azure Devops
- Lint Python and YAML files
- Run unit tests with pytest
- Validate Azure ML YAML files
- Register datasets and components using Azure ML CLI v2
- Submit training jobs via Azure ML job creation
- Auto-register models using MLflow tracking
- Deploy model to Azure ML managed online endpoint
- Configure endpoint autoscaling and authentication
- Invoke and test endpoint with sample payload
- Enable logging via Application Insights
- Unit tests for data prep and feature logic
- Integration tests for ML jobs and data flows
- Deployment tests for live endpoint scoring
- Chain preprocessing, training, and deployment in Azure ML Pipelines
- Trigger retraining on schedule or blob data arrival
- Track metrics and artifacts using Azure ML Experiments
- Monitor data drift, prediction drift, and endpoint health
- Log model lineage and versioning automatically

***User Access Framework***

**OBJECTIVE:** Define stakeholder-based access control for Power BI and Azure resources.

| *Role* | *Platform Access* | *Permission Level* |
|---|---|---|
| **Operations & Logistics** | Power BI | Read-only (Pipeline Tracker) |
| **Quality Team** | Power BI | Read-only (Incidents & Heatmaps) |
| **Data Science Team** | Azure ML, Power BI | Read/Write (Model Monitoring) |
| **Procurement Team** | Power BI | Read-only (Scorecards) |
| **Business Leadership** | Power BI | Read-only (Executive Dashboards) |
| **Azure Admin** | All Azure Resources | Contributor/Admin |

***Change Management Guide***

**OBJECTIVE:** Facilitate smooth adoption of the new system across internal teams and suppliers.

i. **Internal Adoption Strategy:**
   *STEP 1*: Internal stakeholder training sessions (per role group)
   *STEP 2*: Power BI onboarding tutorials and quick-reference guides
   *STEP 3*: FAQ documentation and IT helpdesk alignment

ii. **Supplier Engagement Strategy:**
   *STEP 1*: Supplier introduction sessions and walkthrough demos
   *STEP 2*: Supplier-facing Excel template guide
   *STEP 3*: Support channel (email + escalation procedure)

iii. **Communication Timeline:**
   *WEEK 1*: Executive and team-level announcements
   *WEEK 2-3*: Training and Q&A sessions
   *WEEK 4*: System go-live with monitoring and support ramp-up

By pairing our technical solution with a comprehensive operational enablement package, we've ensured that FashionWorld is not only prepared for deployment but also positioned for long-term success. The documentation we've delivered, spanning deployment, data ingestion, access control, and change management, supports a smooth transition from pilot to enterprise scale. This foundation equips internal teams and suppliers with the tools, guidance, and governance they need to fully adopt and sustain the new system, ensuring measurable value and continuous improvement across packaging operations.

---

## CONCLUSION

Through a structured, data-driven approach, our team successfully delivered a comprehensive solution to optimize FashionWorld's packaging quality management. From identifying key inefficiencies in the current manual and fragmented process to designing and deploying a fully automated, scalable architecture built on Azure technologies, this project lays the groundwork for long-term transformation.

The integration of machine learning models into operational workflows, combined with role-specific dashboards, data governance tools, and a robust change management strategy, ensures measurable impact across supplier performance, packaging accuracy, and real-time decision-making. By empowering both internal teams and external partners with actionable insights and automation, FashionWorld is now equipped to reduce reprocessing costs, improve supply chain resilience, and uphold its commitment to quality and sustainability at scale.

Our solution is ready for enterprise deployment, with built-in flexibility to evolve alongside FashionWorld's future operational and business needs. This project not only solves an immediate operational challenge, but also positions the organization as a digital leader in retail packaging optimization.

# APPENDIX

## i. Azure Blob Storage

Azure Blob Storage is a scalable object storage solution, it supports structured and unstructured data including raw and processed data files including CSV, Excel, and ML feature outputs. It serves as the data lake in the pipeline, retaining raw inputs and historical records for traceability. Because Blob Storage is highly durable and can handle virtually unlimited data, FashionWorld can reliably archive every quality incident (e.g. images of defective packaging or measurement logs) without worrying about capacity, data loss or cost. Blob storage supports lifecycle management policies and tiered storage (hot, cool and archive) which optimizes costs for infrequently accessed historical data. Additionally, Blob storage provides geo-redundant storage and encryption-at-rest by default. Therefore, all quality data is durably stored with high availability guarantees and compliance features, supporting FashionWorld's traceability and governance requirements.

***CONFIGURATION***: Provision a storage account with dedicated blob containers for raw and processed data. Azure Blob Storage natively integrates with Event Grid, to enable this, Storage Event Publishing should be chosen on the containers so new file uploads trigger an event notification. Event-driven configuration ensures as soon as a device uploads a new file, downstream processes are immediately notified.

***SOURCE***:
https://learn.microsoft.com/en-us/azure/architecture/example-scenario/monitoring/monitoring-observable-systems-media

## ii. Azure Event Grid

Azure Event Grid is a fully managed serverless event routing service that decouples event producers and consumers in our architecture. It uses a publish-subscribe model to deliver events from sources like Blob Storage and IoT Hub to downstream handlers like Functions and Logic Apps. Event Grid enables real-time, serverless orchestration of quality control workflows. The service can also be scaled to multiple subscribers, ensuring the solution remains responsible even as IoT devices scale up. Event Grid guarantees at-least-once delivery and implements retry and dead-lettering policies, so even if a subscriber is temporarily down, the event will be delivered reliably.

***CONFIGURATION***: Create event subscriptions for relevant events on our sources. One subscription will listen to Blob Storage Events (the filter) and trigger Azure Functions (the endpoint) to start the pre-processing pipeline for Machine Learning inferences.

***SOURCES***:
https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-event-grid
https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-event-overview

## iii. Azure IoT Hub

Azure IoT Hub is a managed cloud gateway that enables secure, bi-directional communication between IoT devices and the cloud. It is the ingestion backbone of FashionWorld's quality management system, gathering real-time data from packaging line

sensors. It can reliably ingest millions of events per second from a fleet of devices while maintaining per-device security and connection management. This makes it well-suited for automated quality management where potentially hundreds of packaging machines and sensors will stream data (e.g. weight, dimensions, images, environmental conditions) that needs to be analyzed. IoT Hub can scale to millions of simultaneous device connections and events, enabling future integration with computer vision technologies for enhanced package quality monitoring. The service also integrates natively with Event Grid, enabling instantaneous triggering of processes when important events occur.

***CONFIGURATION***: Register each packaging sensor or inspection device as an IoT Hub device (using the IoT Hub identity registry). Each device would use secure credentials (SAS token or X.509 cert) to send telemetry messages about packaging quality to the hub. IoT Hub's built-in message routing will be used to direct data streams appropriately: for example, set up a route that sends all raw telemetry into Blob Storage. Another route that forwards specific events (like package arrival) into Event Grid. IoT Hub also allows cloud-to-device messages, which could be used to send commands back to the line. For instance, to halt a production line for manual inspection if a critical defect is detected.

***SOURCES***:
https://learn.microsoft.com/en-us/azure/iot-hub/iot-concepts-and-iot-hub
https://devblogs.microsoft.com/azure-sql/build-your-full-paas-iot-solution-with-azure-sql-database/#:~:text=From%20device%20connectivity%20and%20management%2C%C2%A0IoT,management%2C%20predictable%20performance%20and%20availability

### iv. Azure Logic Apps

Azure Logic Apps is a cloud service for creating and running automated workflows and integration processes with little to no code. This service is well-suited for FashionWorld's supplier communication needs because it can easily connect to external systems (email, ticketing systems, etc.) and define multi-step workflows. It automates the generation, distribution, monitoring, validation, and follow-up of these emails, reducing manual administrative work and turnaround delays. Its native integration with Office 365 Outlook and Excel Online means it can securely handle corporate email accounts and Excel processing without requiring external tools.

***CONFIGURATION***: Azure Logic Apps will be configured to automatically generate and email suppliers pre-filled Excel order forms using the Excel Online and Office 365 Outlook connectors when a new order is triggered, populating fields such as product type, folding method, and carton layout based on benchmark data. It will monitor a shared mailbox for supplier responses, filter emails by supplier and order identifiers, and automatically download and validate the returned Excel attachments for data completeness and correctness. If issues are found, Logic Apps will automatically send follow-up emails with predefined dynamic templates requesting corrections, and if the data is valid, it will communicate with Azure Function Apps to perform the final data validation and upload to Blob storage for ingestion.

### v. Azure Functions

Azure Functions is a serverless compute service that allows running small pieces of code (functions) on demand, typically triggered by events. Functions are an excellent fit for this use case because they enable us to perform on-the-fly data processing and apply machine

learning in response to events – for example, calling a ML model as soon as a sensor reading comes in. These Azure Functions will utilize managed identities for secure service access, support scaling to handle high event volumes from IoT Hub and Event Grid, and ensure FashionWorld's pipeline remains modular, fully automated, and operationally resilient.

***CONFIGURATION***: Deploy one or more Azure Function Apps with targeted serverless functions for each processing step, using event-driven triggers for seamless integration with Blob Storage, Event Grid, IoT Hub, and Azure SQL

- For order data ingestion, ValidateOrderDataFunction will be triggered by Logic Apps via an HTTP request or Event Grid to perform schema validation and data quality checks. Before programmatically uploading clean files to the order data container.
- For feature preparation, a FeatureEngineeringFunction will be triggered by Event Grid upon new uploads in the sensor data container, loading the relevant order data, product attributes, and supplier scorecard from their respective raw and processed blobs using Blob input bindings, performing join operations and transformations via a pre-registered preprocessing .pkl pipeline, and preparing ML-ready features for the Azure ML Endpoint
- For ML inference, an MLPostProcessingFunction will receive model outputs from the ML endpoint, evaluate the returned label, and if labeled "Bad," it will automatically generate a structured JSON incident record, writing it to the historical incidents container using Blob output bindings for full traceability.
- For supplier performance tracking, a SupplierScorecardAggregationFunction will be scheduled using a Timer Trigger to run monthly, aggregating incident and quality data across the period, calculating supplier KPIs (e.g., bad packaging rates, on-time delivery percentages), and writing the updated scorecards to the supplier scorecard processed container.
- For analytics ingestion, an IngestToSQLFunction will be triggered by Event Grid upon new blobs appearing in feature and model output containers, parsing the structured outputs and appending validated records to the Azure MySQL Flexible SQL database using SQL output bindings or ADO.NET within the function.

***SOURCES:***
https://learn.microsoft.com/en-us/azure/architecture/example-scenario/monitoring/monitoring-observable-systems-media

### vi. Azure SQL Database

Azure SQL Database is a fully managed relational database service for storing structured data in the cloud. In FashionWorld's quality management pipeline, it plays the role of the central data repository for all structured information: quality inspection results, incident records, supplier information, sensor data and product data. It offers the robust consistency and query capabilities of SQL Server without the overhead of managing a server. It provides high performance, scalability, and built-in high availability, which means the data is safely stored and can be queried at any time for audits or analytics. Azure SQL Database integrates well with both Logic Apps and Power BI with native connectors.

***CONFIGURATION****:* Azure SQL Database will be configured on a suitable tier to handle both transactional inserts (from our Functions or Logic Apps writing new incidents) and analytical queries (from Power BI or analysts running reports). The Temporal Tables feature should be enabled to ensure changes are versioned over time for authoring and quality

management. Additionally, Azure AD authentication and role-based access should be used so that only authorized personnel can query or update the data, for data governance.

***SOURCES:***
https://devblogs.microsoft.com/azure-sql/build-your-full-paas-iot-solution-with-azure-sql-database/#:~:text=Azure%20SQL%20proven%20to%20be,series%20analysis

### vii. Azure SQL Database

Power BI is a suite of business analytics tools that enables interactive data visualization and reporting. In the packaging quality management pipeline, Power BI is used to turn all the collected data: incident logs, supplier performance metrics, and real-time sensor readings, into insights and dashboards for decision-makers. It provides a rich, visually immersive way to monitor quality KPIs and to drill into details for traceability.

***CONFIGURATION:*** In practice, Power BI dataset and reports will pull from the Azure SQL Database. Enabling direct query mode can allow for near real-time dashboards. As soon as an incident is inserted into SQL by the pipeline, it will reflect in the Power BI visuals (with minimal delay).

***SOURCES:***
https://learn.microsoft.com/en-us/azure/architecture/ai-ml/architecture/automate-pdf-forms-processing
https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview

# Machine Learning Technical Annex

## *Dataset Preparation*

The dataset used in model training was constructed through extensive preprocessing and domain-informed feature engineering. A total of 32 features were retained for modeling, encompassing numerical variables, engineered binary indicators, time-based fields, and encoded categorical features. Class labels were encoded such that "Good" equals 0 and "Bad" equals 1. The final dataset consisted of 477,055 observations, with approximately 19.9% labeled as "Bad," indicating class imbalance.

The dataset was partitioned into training and test sets using stratified sampling in an 80:20 ratio, resulting in 381,644 training samples and 95,411 test samples.

## *Features*

| Feature | Data Type | Source | Description | Feature Type | Purpose |
|---|---|---|---|---|---|
| Weight | float | Original | Product weight in kg | Numerical | Model input |
| ProposedUnitsPerCarton | float | Original | Proposed number of units per carton | Numerical | Model input |
| PackagesHandled | float | Merged | Monthly number of packages handled by supplier | Numerical | Operational load proxy |
| TotalIncidents | float | Merged | Number of incidents in the reporting month | Numerical | Supplier quality proxy |
| AnomaliesDetected | float | Merged | Monthly number of detected anomalies | Numerical | Operational risk proxy |
| BadPackagingRatePct | float | Merged | Percentage of bad packaging reported monthly | Numerical | Supplier quality risk |
| OnTimeDeliveryRatePct | float | Merged | On-time delivery rate for the supplier | Numerical | Delivery reliability |
| AverageCostPerIncidentEUR | float | Merged | Average financial cost of an incident | Numerical | Cost impact proxy |
| TotalHistoricalIncidents | int | Aggregated | Historical number of incidents per product-supplier | Numerical | Long-term risk |
| UnresolvedIncidents | int | Aggregated | Number of unresolved past incidents | Numerical | Operational risk |
| AvgIncidentCost | float | Aggregated | Average cost of past incidents | Numerical | Cost risk metric |
| ReportYear | int | Extracted | Year of packaging report | Categorical | Seasonality control |
| ReportMonthNum | int | Extracted | Month of report | Categorical | Seasonality control |
| ReportQuarter | int | Extracted | Quarter of the year | Categorical | Seasonality control |
| ReportDayOfMonth | int | Extracted | Day of month of report | Categorical | Cycle detection |
| ReportDayOfWeek | int | Extracted | Weekday index (0=Mon) | Categorical | Operational schedule |
| IsWeekend | int | Engineered | 1 if report submitted on weekend | Binary | Operational variation |
| supplier_bad_rate | float | Engineered | Historical bad rate for supplier | Numerical | Supplier reliability |
| is_high_risk_supplier | int | Engineered | 1 if supplier bad rate > median | Binary | Supplier flag |
| product_bad_rate | float | Engineered | Historical bad rate for product | Numerical | Product quality flag |
| is_high_risk_product | int | Engineered | 1 if product bad rate > median | Binary | Product flag |
| is_risky_supplier_product | int | Engineered | 1 if both supplier and product are high risk | Binary | Interaction effect |
| is_wool_product | int | Engineered | 1 if material is Wool | Binary | Material-specific risk |
| is_low_delivery_perf | int | Engineered | 1 if delivery rate < mean | Binary | Supplier delivery risk |
| is_high_bad_rate_supplier | int | Engineered | 1 if bad rate > mean | Binary | Supplier packaging issue |
| is_bad_incident_history | int | Engineered | 1 if incident cost > mean | Binary | Supplier cost risk |
| GarmentType_enc | int | Encoded | Garment type (ordinal encoded) | Categorical | Product identifier |
| Material_enc | int | Encoded | Material type (ordinal encoded) | Categorical | Product material |
| ProposedFoldingMethod_enc | int | Encoded | Folding method (ordinal encoded) | Categorical | Packaging proposal |
| ProposedLayout_enc | int | Encoded | Proposed packaging layout | Categorical | Layout strategy |
| Size_enc | int | Encoded | Product size | Categorical | Dimensional control |
| Collection_enc | int | Encoded | Seasonal collection | Categorical | Product grouping |

Extensive preprocessing and feature engineering were undertaken to construct a high-quality dataset suitable for predictive modeling. The initial raw data, comprising density reports, product attributes, supplier scorecards, and incident logs, was systematically merged to form a unified, observation-level dataset enriched with both operational and historical risk indicators. Several domain-informed features were engineered to reflect latent quality drivers, including binary flags for high-risk suppliers and products (based on historical defect rates), low delivery performance, elevated incident costs, and wool-based materials. Temporal features such as report year, month, day-of-week, and weekend indicators were extracted from timestamps to capture seasonality and operational cycle effects. Aggregated incident metrics were computed at the product–supplier level to quantify long-term risk, while interaction features captured compound effects between supplier and product risk.

Categorical variables including garment type, material, folding method, layout, size, and collection were transformed using ordinal encoding to maintain compatibility with tree-based models such as XGBoost and LightGBM. These encodings ensured efficient representation while preserving category identity. Additionally, all non-predictive identifiers (e.g., report ID, raw timestamps) and correlated or redundant features were excluded to reduce noise and avoid data leakage. Missing values in incident-related columns were imputed with domain-appropriate defaults, typically zeros, to retain low-risk cases without introducing bias. Finally, the target variable, *PackagingQuality*, was binarized into 0 for "Good" and 1 for "Bad" labels, clearly defining the supervised learning objective. This carefully curated and engineered dataset formed the foundation for all downstream modeling efforts.

***Modeling Strategy Overview***

Five classification algorithms were selected to cover a spectrum of complexity and modeling paradigms:

- Random Forest
- XGBoost
- LightGBM
- CatBoost
- Logistic Regression

Each model was initially trained with default parameters to establish baseline performance. Subsequently, hyperparameter tuning was performed via stratified five-fold cross-validation with ROC AUC as the primary optimization metric. Once the top-performing models were identified, they were further refined with a focus on maximizing recall to align with the business priority of minimizing undetected defective units.

***Model Evaluation Framework***

Each model was evaluated using the following metrics:

- **Accuracy**: Overall classification correctness
- **Precision** (for class 1): Proportion of predicted "Bad" instances that were correct
- **Recall** (for class 1): Proportion of actual "Bad" instances that were correctly identified
- **F1 Score** (for class 1): Harmonic mean of precision and recall
- **ROC AUC**: Threshold-independent area under the ROC curve

A particular focus was placed on recall for the minority class, as failing to detect a defective packaging case carries a high operational cost.

***1. Random Forest***

**Baseline Model (Default Parameters)**

The baseline Random Forest model achieved an accuracy of 79.6% on the test set. However, its performance on the minority class was suboptimal:

- **Precision (class 1)**: 0.421

- **Recall (class 1)**: 0.068
- **F1 Score (class 1)**: 0.117
- **ROC AUC**: 0.688

The low recall indicated poor detection of defective packaging, prompting further tuning and feature evaluation.

**Feature Engineering Impact**

The Random Forest model was retrained on a reduced set of raw features (excluding engineered variables). In this configuration:

- **Precision (class 1)**: 0.38
- **Recall (class 1)**: 0.07
- **F1 Score (class 1)**: 0.11

The degradation in recall and F1 confirmed the positive contribution of engineered features and justified their inclusion.

**Tuned Model (Best Parameters)**

Hyperparameter optimization identified the following optimal settings: n_estimators=300, max_depth=10, min_samples_split=2.

With these parameters:

- **Accuracy**: 65%
- **Precision (class 1)**: 0.32
- **Recall (class 1)**: 0.68
- **F1 Score (class 1)**: 0.44
- **ROC AUC**: 0.7252

The recall improved significantly, fulfilling the project's primary objective, though at the cost of reduced precision and overall accuracy.

***2. XGBoost***

**Baseline Model**

The initial XGBoost classifier, trained with fixed parameters, produced the following results:

- **Accuracy**: 65%
- **Precision (class 1)**: 0.32
- **Recall (class 1)**: 0.69
- **F1 Score (class 1)**: 0.44
- **ROC AUC**: 0.7252

These results already surpassed the Random Forest baseline in terms of recall and matched its tuned performance.

**Tuned Model**

Hyperparameter tuning yielded optimal values: n_estimators=100, max_depth=5, learning_rate=0.1.

The final model retained the same performance as the baseline:

- **Accuracy**: 64%
- **Precision (class 1)**: 0.32
- **Recall (class 1)**: 0.71
- **F1 Score (class 1)**: 0.44
- **ROC AUC**: 0.7252

These results matched those of the tuned Random Forest, with slightly better recall.

***3. LightGBM***

**Baseline Model**

LightGBM's default configuration produced results nearly identical to XGBoost:

- **Accuracy**: 64%
- **Precision (class 1)**: 0.32
- **Recall (class 1)**: 0.70
- **F1 Score (class 1)**: 0.44
- **ROC AUC**: 0.7252

**Tuned Model**

Grid search yielded best parameters: n_estimators=200, max_depth=15, learning_rate=0.05.

The tuned LightGBM model maintained the same performance:

- **Accuracy**: 64%
- **Precision (class 1)**: 0.32
- **Recall (class 1)**: 0.70
- **F1 Score (class 1)**: 0.44
- **ROC AUC**: 0.7252

***4. CatBoost***

CatBoost was configured to use raw categorical variables directly. Class imbalance was addressed using class weights [1, 4].

**Tuned Model**

The optimized CatBoost model achieved the following:

- **Accuracy**: 65%
- **Precision (class 1)**: 0.32

- **Recall (class 1)**: 0.70
- **F1 Score (class 1)**: 0.44
- **ROC AUC**: 0.7257

CatBoost marginally outperformed other models on ROC AUC, maintaining high recall and comparable precision. Its ability to handle categorical features natively makes it operationally efficient.

***5. Logistic Regression***

Logistic regression was applied after filtering highly correlated features and standardizing all numerical variables. Categorical features were one-hot encoded.

**Tuned Model**

Grid search tuning over penalty types (l1, l2) and regularization strength (C) led to a model achieving:

- **Accuracy**: 65%
- **Precision (class 1)**: 0.32
- **Recall (class 1)**: 0.69
- **F1 Score (class 1)**: 0.44

Despite being a linear model, logistic regression matched the recall of ensemble methods, confirming the quality of the engineered feature set.

**Final Model Selection after Hypertuning**

| Model | Precision (Bad) | Recall (Bad) | F1 Score (Bad) | Accuracy | ROC AUC |
|---|---|---|---|---|---|
| Random Forest | 0.32 | 0.68 | 0.44 | 0.65 | 0.7252 |
| XGBoost | 0.32 | 0.7 | 0.44 | 0.64 | 0.7252 |
| LightGBM | 0.32 | 0.7 | 0.44 | 0.64 | 0.7252 |
| CatBoost | 0.32 | 0.7 | 0.44 | 0.65 | 0.7257 |
| Logistic Regression | 0.32 | 0.69 | 0.44 | 0.65 | 0.725 |

The final model comparison was conducted after rigorous hyperparameter tuning for all candidate classifiers. Each model was optimized through stratified five-fold cross-validation using ROC AUC as the primary selection metric, followed by evaluation on a held-out test set. The primary objective was to maximize the recall for the minority class representing defective packaging events while maintaining acceptable levels of precision and overall discrimination power.

Across all tuned models, performance was consistent with a recall of approximately 0.70 for the minority class. XGBoost, LightGBM, and CatBoost all achieved this threshold, with CatBoost delivering a marginally higher ROC AUC of 0.7257. Random Forest and Logistic Regression closely followed with recall scores of 0.68 and 0.69, respectively. All models exhibited identical precision and F1 scores for the minority class (0.32 and 0.44,

respectively), indicating a shared trade-off between true and false positives under the class imbalance.

Despite near-identical performance across ensemble methods, XGBoost was ultimately selected for deployment due to its robust performance, efficient inference capabilities, and widespread production readiness. Its ability to achieve high recall while maintaining a balanced ROC AUC ensures that the system aligns with the business requirement of minimizing undetected defective units.

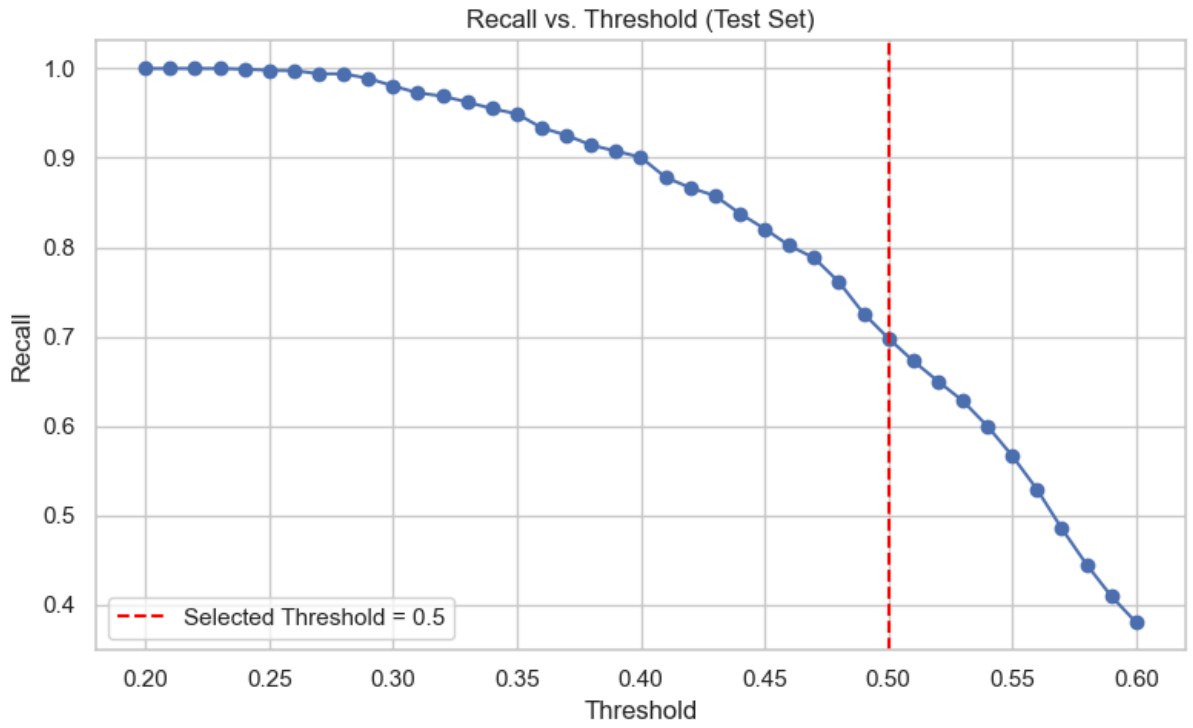
**Final Model Recall Tuning**

The final selected model, XGBoost, was subject to a second round of targeted hyperparameter optimization with recall as the primary evaluation criterion. This strategic focus on recall was driven by the nature of the business problem: the need to minimize the risk of undetected defective packaging units. In the context of quality control, failing to identify a bad packaging instance has a higher operational and reputational cost than falsely flagging a good one. Therefore, maximizing recall ensures that the majority of true defective cases are captured, even at the expense of precision or false positives, an acceptable trade-off for FashionWorld's risk management policy.

To formalize this requirement during model selection, a custom recall-based scoring function was constructed using the make_scorer utility from scikit-learn. This scoring method evaluated recall specifically for the positive class (label 1), which corresponds to defective packaging. By explicitly focusing the optimizer on this metric, the hyperparameter search space was explored with the goal of enhancing the model's sensitivity to the minority class. This approach contrasts with traditional ROC AUC, which evaluates overall discrimination, or accuracy, which tends to favor the majority class in imbalanced datasets.

| {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 100} | | | | |
|---|---|---|---|---|
| Metric | Precision | Recall | F1 Score | Support |
| Class 0 (Good) | 0.89 | 0.63 | 0.74 | 76471 |
| Class 1 (Bad) | 0.32 | 0.7 | 0.44 | 18940 |
| Accuracy | | | 0.65 | 95411 |
| Macro Avg | 0.61 | 0.67 | 0.59 | 95411 |
| Weighted Avg | 0.78 | 0.65 | 0.68 | 95411 |

The recall scoring pipeline integrated seamlessly with stratified five-fold cross-validation to ensure stable and generalizable performance across all segments of the data. As a result, the tuned XGBoost model demonstrated not only high recall (0.70) but also retained acceptable ROC AUC (0.7252), confirming its ability to differentiate effectively between good and bad packaging cases while prioritizing early detection of defects. This metric-centric tuning strategy directly supports the organization's shift from reactive defect resolution to proactive quality assurance.




The plot titled **"Recall vs. Threshold (Test Set)"** illustrates how the recall of the final XGBoost model varies as the classification threshold is adjusted between 0.20 and 0.60. This curve plays a critical role in aligning model outputs with business objectives, especially in scenarios where recall is prioritized, such as proactive quality control.

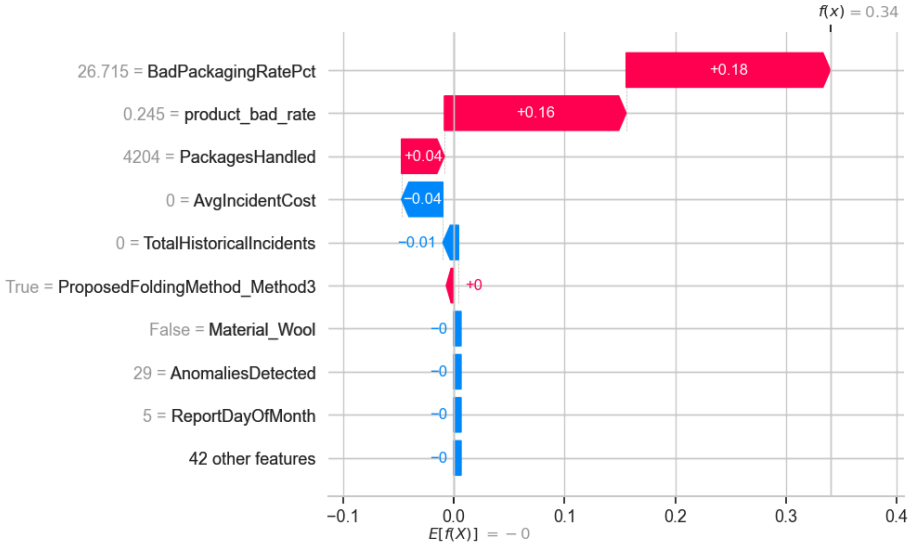
**Analysis:**

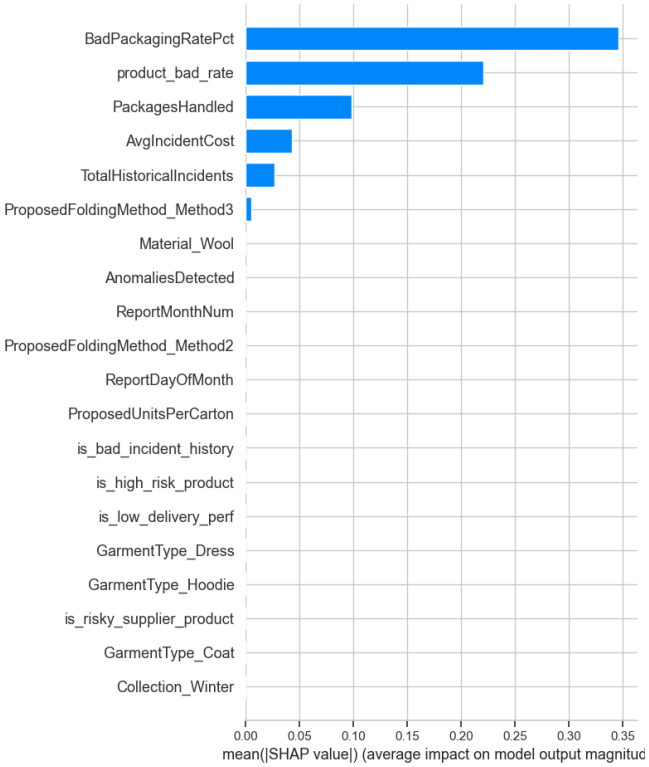
**Recall Trend**: As the threshold increases, the recall decreases monotonically. This is expected, as higher thresholds result in fewer positive predictions, thereby missing more true positives (bad packaging cases). At very low thresholds (e.g., 0.20), the model captures nearly all defective cases, achieving a recall close to 1.0. However, this comes at the cost of a potentially high false positive rate.

**Default Threshold (0.50)**: The red dashed line indicates the standard decision threshold of 0.5. At this point, the recall is approximately 0.70. This threshold provides a balanced trade-off between capturing defective cases and limiting false positives, which aligns with the previously reported model metrics.

**Operational Implication**: Lowering the threshold could further increase recall (moving above 0.90), which may be suitable for applications with zero tolerance for undetected defects. However, this would significantly inflate the false positive rate, potentially leading to excessive manual inspections. Conversely, increasing the threshold would reduce false alarms but risk missing critical defect cases.

## *SHAP ANALYSIS AND INTERPRETATION :*




The SHAP summary plot (beeswarm) provides a high-resolution view of how individual features influence the XGBoost model's predictions across all samples. Each dot represents a SHAP value for a single prediction, with colors indicating the original feature value (red for high, blue for low). The horizontal spread of each feature shows the variation in its effect across the dataset. Features are ranked by their mean absolute SHAP value, indicating overall importance. BadPackagingRatePct and product_bad_rate are the two most influential features, with high values consistently pushing the model toward predicting the defective class (label 1). Operational volume, captured through PackagesHandled, also plays a strong role, suggesting that supplier workload is associated with elevated defect likelihood. Other top-ranked contributors include AvgIncidentCost, TotalHistoricalIncidents, and binary flags such as Material_Wool and ProposedFoldingMethod_Method3, indicating that certain materials and folding techniques correlate with poor packaging outcomes.

The SHAP bar plot offers a global summary of feature importance by averaging the absolute SHAP values for each feature across the entire test set. This representation quantifies the magnitude of impact, independent of direction. The plot reaffirms that BadPackagingRatePct dominates model behavior, followed closely by product_bad_rate and PackagesHandled. These three features contribute the majority of model variance, confirming the hypothesis that defect rates and operational scale are critical predictors of packaging failure. Intermediate contributors like AvgIncidentCost and TotalHistoricalIncidents reflect cost and reliability metrics, while encoding variables for folding method and material type offer additional nuance without dominating the prediction logic.

The SHAP force plot decomposes a single prediction to show the additive contribution of each feature toward the final output score. In the illustrated case, the model's baseline (expected value) is shifted upward by strong positive contributions from a high BadPackagingRatePct and elevated product_bad_rate, collectively increasing the predicted probability of a defect. Smaller contributions come from operational indicators such as PackagesHandled, while some features like AvgIncidentCost and TotalHistoricalIncidents slightly pull the prediction downward, reflecting neutral or risk-reducing effects. Categorical conditions such as the use of ProposedFoldingMethod_Method3 are also visible, with a zero net effect in this instance. The waterfall-style breakdown enables stakeholders to audit individual decisions and trace model reasoning back to specific data points.