WESTMINSTER UNIVERSITY

Progress Report Feb 19th

Data Capstone - SAP Backorder Prediction | Pipeline Summary, Modeling Plan, Next Steps

Purpose: Condensed progress report for professor review: clear pipeline summary, full modeling plan, sprint timeline, and closing reflections.

Scope: ETL summary, modeling plan, next steps, ultimate goal

Pipeline Summary (ETL)

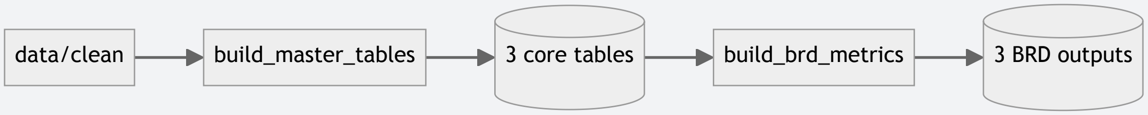
Short, high-level overview of the data pipeline.

What I Built: A two-phase ETL pipeline that turns SAP ERP clean CSVs into master tables and BRD-aligned metrics (backorders, weeks of coverage). Aligned with EnableCV BRD logic.

Directory Structure

- `data/clean/main/` : Transactional tables (orders, delivery, billing, inventory, PO)
- `data/clean/supporting/` : Reference tables (plant, company_code, sales_org)
- `data/processed/` : Master tables + BRD outputs (6 CSV files)
- `src/data/` : build_master_tables.py, build_brd_metrics.py, run_pipeline.py

ETL Flow



Step 1 - Core Master Tables: `build_master_tables.py` reads from `data/clean/main/` → produces: master_order_fulfillment, master_inventory_material, master_purchase.
Step 2 - BRD Metrics: `build_brd_metrics.py` reads core tables + delivery data → produces: master_order_fulfillment_brd, shipment_history, master_woc.

Output Tables

Table	Rows	Description
`master_order_fulfillment`	~52k	Order-to-cash: SO + delivery + billing + material + customer
`master_order_fulfillment_brd`	~52k	Same + outstanding_qty, saleable_inventory, backorder_units/amount, aging
`master_inventory_material`	~66k	Inventory by material/plant: unrestricted_stock, blocked, etc.
`master_purchase`	~862k	Purchase orders: PO + vendor + material
`shipment_history`	~12k	Material × week shipments for AWD (rolling 24 weeks)

Table	Rows	Description
`master_woc`	~26k	Weeks of Coverage: SI, net_available, AWD, WOC, woc_low_flag

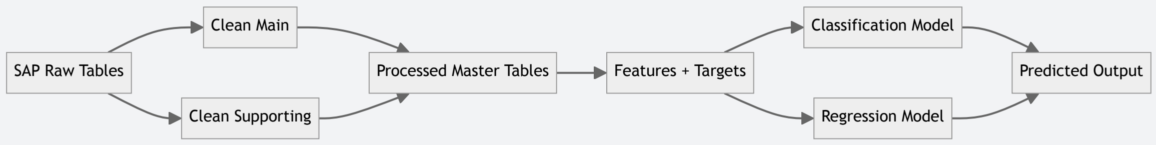
Modeling Plan

SAP Backorder Prediction: Regression & Classification Models

> **Purpose:** Model selection for backorder prediction, regression (magnitude) and classification (risk). Primary + Plan B for each.
>
> **Scope:** Predict backorder magnitude from historic sales trends and inventory; classify backorder risk.
>
> **Status:** Plan, to be refined after EDA and baseline results

Data Flow: From Raw to Prediction

How data moves from the SAP repository through ETL to final predictions.



Stages

- **SAP Raw:** Sales, inventory, delivery, billing, material master, etc.
- **Clean Main / Supporting:** Normalized, deduplicated, joined by keys
- **Processed Master:** master_order_fulfillment_brd, master_inventory_material, master_woc
- **Features + Targets:** Sales trends, inventory, WOC, backorder flags
- **Models:** Classification (yes/no) and regression (magnitude)

Final Prediction Output

Each row is a material/plant with predicted backorder risk and inventory context.

Material	Plant	Predicted Backorder	WOC	Saleable Inventory
000000000000002733	1000	Yes	2.1	45
000000000000002480	1000	No	8.3	120
000000000000001107	1000	Yes	0.5	12

Regression: Magnitude of Backorder

Predict numeric backorder (such as `backorder_units`, shortfall, demand) from historic sales trends and inventory.
Primary: Ridge / ElasticNet Regression

> **Rationale:** Linear, interpretable, and robust to multicollinearity (common in sales + inventory features). Use layered feature groups (sales trends, inventory, lead time) for interpretability.

- **Pros:** Fast, stable, easy to explain to stakeholders
- **Cons:** Assumes mostly linear relationships

Plan B: Gradient Boosting Regressor (XGBoost / LightGBM)

> **Rationale:** Captures non-linear effects and interactions without manual feature engineering. Often best performance on tabular supply-chain data.

- **Pros:** Handles missing values and outliers; strong on complex relationships
- **Cons:** Less interpretable; needs tuning

Classification: Backorder Risk (Yes/No)

Predict backorder risk, binary or risk level classification.

Primary: XGBoost / LightGBM

> **Rationale:** Strong on tabular data; handles ~10% backorder class imbalance well. Built-in feature importance and interaction capture.

- **Pros:** Often best performance; handles imbalance; feature importance for reporting
- **Cons:** Less interpretable than logistic regression

Plan B: Logistic Regression

> **Rationale:** Interpretable baseline; coefficients show direction and relative importance of inventory, lead time, order/delivery timing.

- **Pros:** Simple, fast, easy to explain
- **Cons:** Assumes linear decision boundary; weaker if relationships are non-linear

Model Selection Summary

Task	Primary	Plan B
Regression (magnitude)	Ridge / ElasticNet	XGBoost / LightGBM Regressor
Classification (risk)	XGBoost / LightGBM	Logistic Regression

> **Workflow:** Start with Ridge/ElasticNet for regression and XGBoost/LightGBM for classification. If Ridge underperforms, switch to gradient boosting for regression. If XGBoost is hard to explain or overfits, fall back to logistic regression for classification.

Glossary

Term	Definition
AWD	Average Weekly Demand, rolling 24-week average of units shipped

Term	Definition
BRD	Business Requirements Document
ETL	Extract, Transform, Load
SAP	Systems, Applications, and Products (ERP software)
SI	Saleable Inventory, unrestricted stock available to fulfill orders
SO	Sales Order
WOC	Weeks of Coverage, net available inventory ÷ AWD

Next Steps: Sprint Schedule

Sprint timeline from Jan 20 through May 1. Target completion: May 1.

Sprint Milestones

Sprint	Dates	Focus
Sprint 1	Jan 20 – Feb 2	Data & targets: ETL pipeline, master tables, target definition (backorder/overstock)
Sprint 2	Feb 3 – Feb 16	EDA: Exploratory analysis, data quality checks, visualizations, document-flow analysis

Sprint	Dates	Focus
Sprint 3	Feb 17 – Mar 2	Baselines & classification: Logistic regression, tree-based models (XGBoost/LightGBM), class imbalance tuning
Sprint 4	Mar 3 – Mar 16	Regression & forecasting: Demand forecasts, inventory estimates, Ridge/ElasticNet, gradient boosting
Sprint 5	Mar 17 – Mar 30	Integration: Pipeline wiring, model evaluation, refinement, Plan B fallbacks if needed
Sprint 6	Mar 31 – Apr 13	Documentation: Report draft, methods, limitations, reproducibility
Sprint 7	Apr 14 – May 1	Final deliverables: Report polish, presentation, code freeze

Target completion: May 1: All development complete; final report, presentation, and deliverables ready.

Summary

Ultimate Goal

Build an end-to-end predictive pipeline that turns SAP ERP data into actionable predictions for backorder risk, overstock risk, and demand/inventory levels. The goal is to reduce revenue loss from backorders, cut waste from overstock, and deliver actionable metrics (demand forecasts, shortfall/excess estimates, recommended order quantities) for replenishment and allocation decisions.

Handling Complexities

The pipeline is inherently complex: SAP data spans multiple tables, document flows (order → delivery → billing), and organizational units. I handle this by:

- **Two-phase ETL:** Core master tables first, then BRD metrics: clear separation of concerns.
- **Normalized join keys:** Canonical document/item keys so different document formats (such as `4500000051` vs `51`) match.
- **Primary vs Plan B models:** Ridge/ElasticNet for interpretability; XGBoost/LightGBM for performance, with fallbacks if needed.
- **Documentation:** Pipeline report, modeling plan, and this progress report to keep the logic clear and reproducible.

Confidence

I am confident with this type of pipelining. I have experience building ETL pipelines from enterprise data, joining transactional and master tables, and deriving business metrics from document flows. The SAP BigQuery dataset is representative of real ERP data, and the pipeline structure (clean → processed → features → models) is well-tested and reproducible.

Excitement

I am excited about the introduction of a more advanced modeling structure. The dual classification (risk) + regression (magnitude) approach, with primary and Plan B options for each, gives flexibility to balance interpretability and performance. Moving from baselines to advanced models (XGBoost, LightGBM) and refining the feature set based on EDA will allow the pipeline to deliver real value for production and replenishment decisions.