Data Science Capstone: Final Proposal

Data Science Capstone: Final Proposal

Predicting Backorder and Overstock Risk, Plus Demand and Inventory Levels

Addy Cruz
Data Science Capstone

Abstract

This proposal addresses supply chain inefficiencies (unfulfilled orders due to backorders and excess inventory) using the SAP BigQuery dataset. The project will build a predictive pipeline that (1) classifies products at risk of backorder or overstock and (2) produces numeric outputs (demand forecasts, expected shortfall, excess units, recommended order quantities) to support production, labor, and resource decisions. Methodologies include exploratory data analysis on SAP sales, materials, and inventory tables; classification models (such as logistic regression, tree-based methods) for risk; and regression/forecasting for demand and inventory levels. The goal is to reduce revenue loss from backorders, cut waste from overstock, and deliver actionable metrics for replenishment and allocation.

Introduction

The project will deliver an end-to-end data science pipeline that turns enterprise SAP data into actionable predictions for backorder risk, overstock risk, and demand and inventory levels. Goals and objectives are: (1) to identify which products or materials are at risk of backorder or overstock using derived signals from order, delivery, and stock data; (2) to produce numeric estimates (demand forecasts, shortfall, excess, or recommended order quantities) so that stakeholders can reallocate production and resources; and (3) to document data provenance, modeling choices, and limitations so that the approach can be reproduced or adapted to company-approved data. The work is scoped to the SAP BigQuery dataset (Kaggle) as the sole data source, with outputs suitable for reports, dashboards, or downstream decision support.

Background

Background and rationale for the project

Why this issue needs to be investigated. Many organizations face simultaneous backorder risk (lost revenue, unfulfilled orders) and overstock (excess inventory, waste). Data-driven classification of at-risk products and numeric forecasts for demand and inventory can improve replenishment, safety stock, and production allocation.

What we will learn and gain. We will learn how to design a reproducible pipeline on real SAP-style data: building targets from document flow (order vs delivery vs billing), joining sales and inventory tables, and combining classification with regression/forecasting. We will gain a working pipeline and evidence of which factors drive backorder and overstock and how well predictions support decisions.

Why it is important. Better prediction of backorder and overstock, plus demand and inventory levels, supports concrete actions (shifting production, reallocating labor, reducing waste, and capturing revenue) which matter for operations and sustainability.

Existing software and related work. Supply chain analytics and demand-forecasting tools (such as ERP modules, dedicated planning software) often rely on similar inputs: sales orders, delivery and billing documents, and stock levels. This project uses a single, well-defined dataset (SAP BigQuery on Kaggle) to demonstrate a full pipeline from raw tables to classification and forecasting outputs, with clear documentation for reproducibility.

Methodological background

How the project will be developed. The project will be developed as a reproducible pipeline: data ingestion and cleaning, feature and target construction from SAP tables, exploratory data analysis (EDA), baseline and advanced models for classification and regression/forecasting, evaluation, and documentation. The pipeline will be implemented in Python (such as pandas, scikit-learn) with version-controlled code and clear separation of data, features, models, and reports.

Technologies and why they are the best choice. *Data:* SAP BigQuery dataset (Kaggle) provides sales documents (`vbak`, `vbap`, `vbep`), material master and stock (`mara`, `mard`), deliveries (`likp`, `lips`), billing (`vbrk`, `vbrp`), and purchasing (`ekko`, `ekpo`, `ekbe`, `eket`), which are sufficient for demand signals, inventory context, and order-to-delivery flow. *Modeling:* Classification—primary: XGBoost/LightGBM; Plan B: logistic regression. Regression (magnitude)—primary: Ridge/ElasticNet; Plan B: XGBoost/LightGBM. *Tooling:* Python, Jupyter for EDA and reporting, and a scripted pipeline (`run_pipeline.py`) for reproducibility. This stack is standard in data science and aligns with capstone scope and timeline.

Proposed Work

Specific tasks

1. **Data and targets:** Ingest and clean SAP BigQuery CSVs; define join keys (`mandt`, `vbeln`, `matnr`, `werks`, etc.); derive backorder/overstock targets or risk signals from order vs delivery timing, shortfall by material, and stock levels.
2. **Feature engineering:** Build features from sales orders, schedule lines, inventory, deliveries, and purchasing (such as lead time, order/delivery history, stock levels, material and customer attributes).
3. **EDA and baselines:** Perform EDA (distributions, missing values, correlation, document-flow analysis); train logistic regression and naive or regression-based demand baselines; document findings.
4. **Classification models:** Train and evaluate classifiers for backorder/overstock risk; handle class imbalance (such as weights, resampling, threshold tuning); report precision, recall, F1, ROC-AUC, and confusion matrix.
5. **Regression and forecasting:** Build demand forecasts and, where applicable, excess-inventory or recommended-order estimates; compare regression and time-series approaches; ensure outputs are actionable.
6. **Pipeline and documentation:** Implement a reproducible pipeline (scripts, configs); document methods, assumptions, and limitations; produce reports suitable for PDF or presentation.

Rationale

Tasks are ordered so that data and targets are fixed first, then features and EDA inform modeling. Classification and regression/forecasting are separated because they address different outputs (risk vs numeric levels) but share the same data and feature base. Baselines before advanced models provide a clear performance comparison. A single pipeline script and documentation ensure the work can be reproduced and extended.

Plan of work

- **Data and targets:** Load SAP tables; define and implement backorder/overstock targets (such as from `vbup`, delivery shortfall, stock thresholds); create master analysis tables.
- **Features and EDA:** Build feature set from orders, deliveries, inventory, and purchasing; run EDA notebooks; summarize data quality and document flow.
- **Classification:** Implement logistic regression and tree-based models; tune for class imbalance; evaluate with standard and business-oriented metrics.
- **Regression/forecasting:** Implement demand and inventory-level models; produce numeric outputs; compare methods and document choices.
- **Integration and docs:** Wire pipeline end-to-end; write methods and limitations; generate final report (MD/HTML/PDF).

Deployment strategy

The completed project will be made available as a **reproducible pipeline and report**, not a live deployed service. **Where and how:** Code and data references live in the capstone repository; the pipeline can be run locally or in a standard Python environment (such as conda/venv). Outputs include tables, figures, and a written report (MD/HTML) that can be converted to PDF. **Intended users:** Instructors, evaluators, and (if shared) stakeholders who want to reproduce or adapt the analysis. **Considerations:** No deployment to cloud or external users is required; the focus is on reproducibility, clarity, and correct use of the SAP dataset. If the project were extended, the same pipeline could be run on company-approved SAP exports with minimal changes to code.

Timeline

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 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.

Work is planned in phases: (1) data and targets + EDA; (2) baselines and classification; (3) regression/forecasting and integration; (4) documentation and report. The sprint timeline allows for iterative refinement based on EDA and baseline results.

Preliminary work (optional)

The repository already contains SAP-derived data in `data/` (raw, clean, processed), a two-phase ETL pipeline in `src/data/` (`build_master_tables.py`, `build_brd_metrics.py`, `run_pipeline.py`), target construction in `src/features/build_targets.py`, and notebooks for EDA and modeling (`01_eda_targets.ipynb`, `02_modeling.ipynb`, `03_conclusion.ipynb`). The pipeline produces six master tables: `master_order_fulfillment`, `master_order_fulfillment_brd`, `master_inventory_material`, `master_purchase`, `shipment_history`, and `master_woc`. This demonstrates feasibility of loading and joining SAP tables and running initial analyses. The final proposal builds on this by formalizing targets, full feature set, model comparison, and report structure.

Evaluation

Technical performance: Success will be evaluated by (1) correct derivation of backorder/overstock targets and demand/inventory signals from SAP data; (2) classification metrics (such as precision, recall, F1, ROC-AUC) and interpretability of feature importance; (3) quality and actionability of numeric outputs (demand forecasts, shortfall/excess estimates); and (4) reproducibility of the pipeline and clarity of documentation.

Ethical and social impact: The project uses a public Kaggle dataset (no confidential or personal data). Considerations include: avoiding overclaiming predictive accuracy; clearly stating assumptions and limitations; and documenting how the approach would need to be validated on company-specific data before operational use. No user or confidential data will be used or shared.

Feedback: Results will be reviewed against capstone rubric and any feedback from advisors or peers; the report will be revised accordingly before final submission.

Conclusion

This proposal outlines a data science capstone that addresses real supply chain problems (backorder and overstock risk, plus demand and inventory levels) using the SAP BigQuery dataset. The approach combines classification (who is at risk?) with regression/forecasting (how much shortfall or excess? what order quantity?) to produce actionable outputs for production and replenishment decisions. The work is scoped to a single, well-documented data source and a reproducible Python pipeline, with clear sprint milestones (Jan 20 – May 1) and deliverables (pipeline, report in MD/HTML/PDF). Success will be measured by technical quality of models and outputs and by the clarity and reproducibility of the final report.

References and data

- **SAP BigQuery Dataset (Kaggle):** SAP Dataset | BigQuery Dataset by Mustafa Keser. Used for demand signals, inventory context, and order-to-delivery flow.
- **Repository:** Data and code in this capstone repository; pipeline entry point in `README.md` and `run_pipeline.py`; reports in `reports/md/` and `reports/html/`.