CEROP: Cross-Border Enterprise Risk & Operations Predictor

Week 5: Model Selection and Shortlisting

Group 4

Arthunya Kanoklertwongse

Aswin Shriram Thiagarajan

Divya Gopal

Ankit Kumar

Westcliff University

AIT 506: Machine Learning

Professor Oueichek Ibba

August 1, 2026

Model Selection and Shortlisting

Week 4 delivered two cleaned, leakage-checked datasets, the financial distress layer (Taiwan Bankruptcy Prediction) and the operations layer (DataCo Smart Supply Chain), each with its own target and its own risk profile. This week picks up where that left off: three candidate models were trained per layer, evaluated on the held-out Week 4 test split, and shortlisted using the same evaluation framework across both, accuracy, precision, recall, F1-score, and ROC-AUC, with the shortlist decision resting on whichever metric actually reflects the cost of a wrong prediction rather than on accuracy alone. Model training and evaluation code for both layers is available in the shared project notebook (see the Method note at the end of each layer section for the direct link).

The two layers are deliberately kept separate below. They do not share a target, a feature set, or even a comparable class balance, so folding them into one narrative would blur two distinct decisions into one. What they do share, where tree-based models beat linear ones, and where recall took priority over precision, is picked back up in the Cross-Layer Synthesis at the end.

Model Selection – Financial Risk Layer

The financial layer picks up directly from the Week 4 preprocessing pipeline by using the saved train/test arrays (5,455 training rows, 1,364 test rows, 85 features) as the target for Bankrupt. Three models were trained and comparing to the Logistic Regression, Random Forest, and XGBoost.

Start with the imbalance, because everything else follows from this. Only 3.23% of firms in the training set are actually bankrupt A model that predicts "not bankrupt" for every single firm would score 96.8% accuracy without learning anything. Accuracy is close to useless here. All three

models were trained with explicit class weighting class_weight="balanced" for Logistic Regression and Random Forest, scale_pos_weight for XGBoost and evaluation leaned on precision, recall, F1, and ROC-AUC instead.

Table 1. Financial layer model comparison (test set, n = 1,364, 44 bankrupt firms)

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.878	0.186	0.818	0.303	0.902
Random Forest	0.966	0.478	0.500	0.489	0.943
XGBoost	0.970	0.535	0.523	0.529	0.950

Note. Values from week5_financial_model_comparison.csv. Class weighting applied to all three models; see Section 2 of the notebook for the imbalance analysis.

At the first glance, Logistic Regression looks like the strongest choice for catching bankruptcies with 36 of 44 true cases (81.8% recall) against XGBoost's 23 (52.3%). But that comparison is misleading. It compares the three models at their own default 0.5 decision threshold, and thresholds aren't fixed properties of a model and it can be moved. Hold all three to the same 81.8% recall and the gap reverses, such as XGBoost catches the same 36 bankruptcies while raising only 82 false alarms, versus 140 for Logistic Regression. a 41% fewer unnecessary reviews for identical detection. Logistic Regression's apparent edge is a threshold artifact, not better separation between classes. Its precision at the default threshold, 18.6%, means roughly five out of every six firms it flags aren't actually going bankrupt.

Random Forest is close behind XGBoost on every metric (F1 0.489 vs. 0.529, ROC-AUC 0.943 vs. 0.950), which makes it a solid fallback if XGBoost proves harder to tune or explain to a non-technical audience.

Recommendation. XGBoost is the shortlisted model for the financial layer, with Random Forest retained as a backup. The call rests on ROC-AUC and the threshold-adjusted comparison above is not accuracy. Since accuracy cannot tell a useful model apart from one that just guesses the majority class here. Logistic Regression is dropped from the shortlist with 85 correlated financial ratios, a linear decision boundary can't separate the classes cleanly, and its 18.6% precision is exactly what that failure looks like.

One caveat worth stating plainly with only 44 bankrupt firms in the test set, a single misclassified firm shifts recall by about 2.3 percentage points. The differences between models are directionally reliable. They're not precise to the decimal point.

Note. Full training and evaluation code: Week5_Financial_Model_Selection.ipynb

Model Selection – Operations Risk Layer

The operations layer builds on the same Week 4 outputs (137,569 training rows, 34,393 test rows, 35 features) with Late_delivery_risk as the target. Three models went into the comparison: Logistic Regression, Random Forest, and a linear-kernel SVM.

The classes here are close to balance 54.8% of orders in the training set are late, 45.2% are on time. That makes accuracy a meaningful number on its own this time, and no class weighting was needed. Precision, recall, and F1 are still reported alongside it, mainly to keep the comparison consistent with the financial layer.

Table 2. Operations layer model comparison (test set, n = 34,393, 18,854 late deliveries)

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.726	0.887	0.573	0.696	0.774

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	0.752	0.873	0.640	0.739	0.855
SVM (linear)	0.725	0.887	0.570	0.694	0.774

Note. Values from week5_operations_model_comparison.csv. Class balance close to even, so no class weighting was applied.

Random Forest wins here, and cleanly. There's no threshold artifact to untangle this time, it leads on every metric that matters at the same default decision point. It correctly identifies 12,068 of the 18,854 late deliveries in the test set on 64.0% recall while holding 87.3% precision. ts F1 (0.739) sits well above both linear models (0.696 and 0.694).

Logistic Regression and the linear SVM land almost identically to each which makes sense given both fit a linear boundary decision over the same 35 features. Neither captures the non-linear interactions between shipping mode, order timing, and regional patterns that Random Forest's tree splits can pick up on naturally.

A 5 fold stratified cross validation confirms the ordering is not an artifact of one split. Random Forest holds the best CV F1 (0.728 ± 0.002), close with the test-set F1 of 0.739, and neither linear model closes the gap under repeated resampling.

Recommendation. Random Forest is the shortlisted model for the operations layer. Unlike the financial layer, there's no need to weigh F1 against recall separately. Random Forest wins both at once, so the statistical result and the business priority point to the same choice without conflict.

Recall matters more than precision here for a specific operational reason. A missed late delivery means losing the chance to warn the customer or step in before the order actually shows up late, while a false positive just triggers an unnecessary review, a minor cost by comparison.

One limitation to carry into Week 6: Random Forest still misses about 36.0% of late deliveries at it default decision threshold. Threshold tuning, the same kind of analysis done for the financial layer, is the obvious next step here, not a settled question.

Note. Full training and evaluation code: AIT506_Week5_Operational_Model_Selection.ipynb

Cross-Layer Synthesis

Two patterns hold across both layers despite the datasets have almost nothing else in common.

Tree based models beat linear ones in both case by similar margin. XGBoost outperform in Logistic Regression by 22.6 points of F1 on the financial layer (0.529 vs. 0.303). For the Random Forest outperform Logistic Regression by 4.3 points on the operations layer (0.739 vs. 0.696). These smaller gap on operations side, but same direction, and probably the same cause. A linear decision boundary cannot capture interactions between correlated financial ratios any better than it can capture interactions between shipping mode, order timing, and region.

Recall was prioritized over precision in both layers too but for different reasons. On the financial layer, a missed bankruptcy means an undetected failing counterparty. On the operations layer, a missed late delivery means losing the chance to warn the customer before the order shows up late. Different business, same shape of asymmetry, a false negative costs more than a false positive.

Where the two layers actually diverge is how much work it took to get to that recall priority. The financial layer's 3.23% bankruptcy rate meant the shortlist decision required active correcting

for a threshold artifact. The operations layer's nearly balance 54.8%/45.2% split meant no correction was needed. Severe imbalance does not just lower the achievable recall, it distorts model looks best until the comparison is done correctly.

Both shortlisted models, XGBoost for financial, Random Forest for operations, carry the same open item into Week 6. Neither is deployed at its optimal decision threshold yet.

References

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). https://doi.org/10.1145/2939672.2939785

Constante, F., Silva, F., & Pereira, A. (2019). *DataCo smart supply chain for big data analysis* [Data set]. Mendeley Data. https://doi.org/10.17632/8gx2fvg2k6.5

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html

Taiwanese Bankruptcy Prediction [Data set]. (2020). UCI Machine Learning Repository. https://doi.org/10.24432/C5004D