Loan Default Modeling

ISA 491/591 | Group 1 – Adam Kahle and Ellis French

Business Problem:

Financial Institutions are looking to reduce risks and improve decision-making in loan approvals. Our goal is to build a predictive model that classifies loan applicants into two categories: Those who are likely to default on their loan and those who are not.

Analytics Problem:

Develop a classification model that predicts the `loan_status` variable as *Charged Off* or *Fully Paid*. We explored Decision Tree, Bootstrap Aggregation, Random Forest, Logistic, XGBoost, Gradient Boost, and Neural Network models trained on an undersampled and complete dataset.

Champion Model: XGBoost

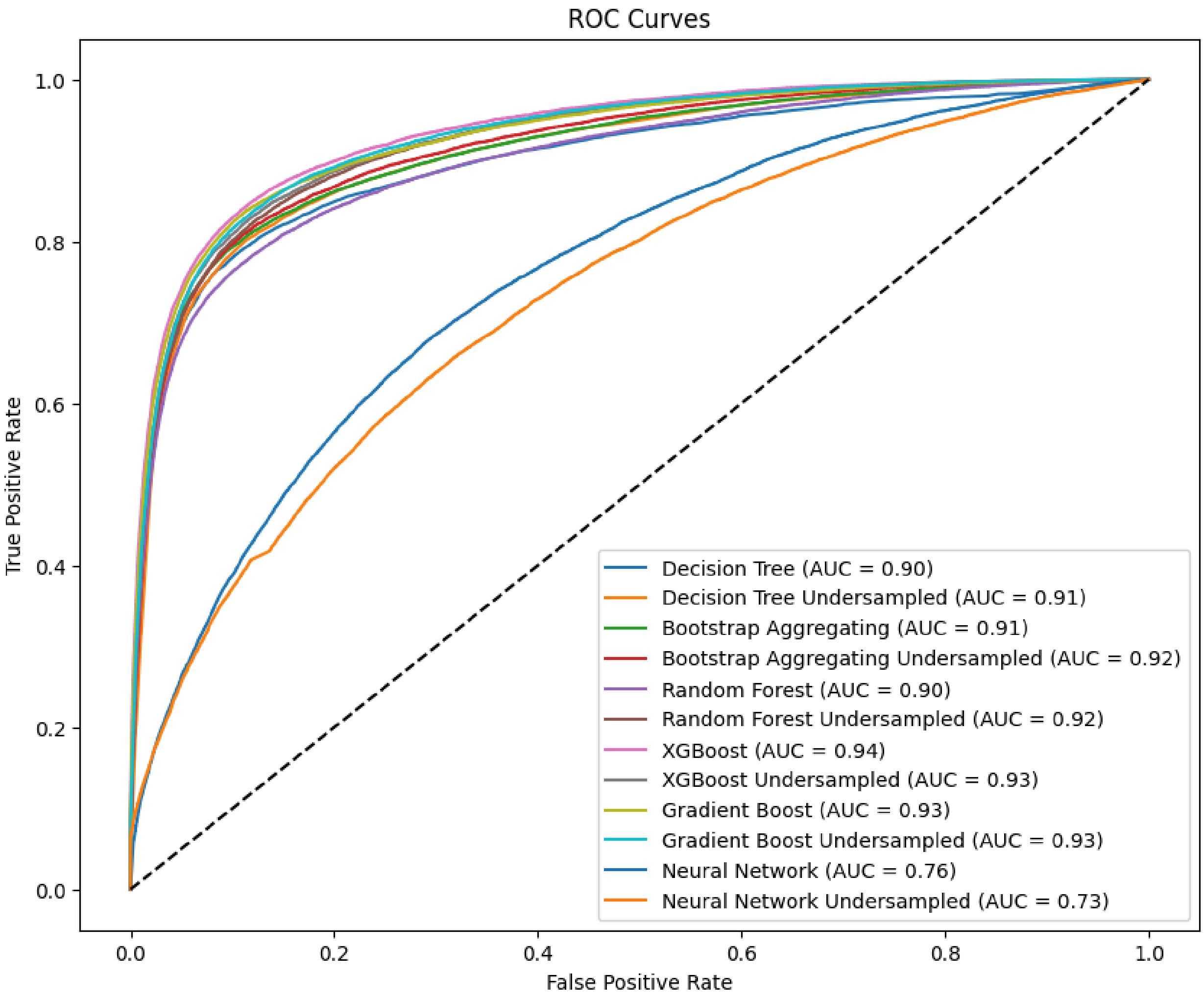
The standard **XGBoost** model is our champion model. The XGBoost model performed the best in nearly every category. *(Appendix A and B)*

The XGBoost Model had the best area under the curve, F1 score, and accuracy of all models. The XGBoost Boost Model had the second-best precision score (slightly lower than the Random Forest). It is important to note that the oversampled models had relatively poor recall scores compared to their balanced counterparts. The XGBoost had a recall score of 0.701008, while its balanced counterpart had a recall score of 0.816154.

F1 is an important metric for loan default models because it balances precision and recall, ensuring the model effectively identifies defaults while minimizing false positives and false negatives. This balance is crucial in financial applications where misclassifications can lead to significant economic impacts or missed opportunities.

We selected the XGBoost Model due to its strong F1 Score performance supplemented with high performance in all other classification metrics.

Appendix A: ROC-AUC Curves

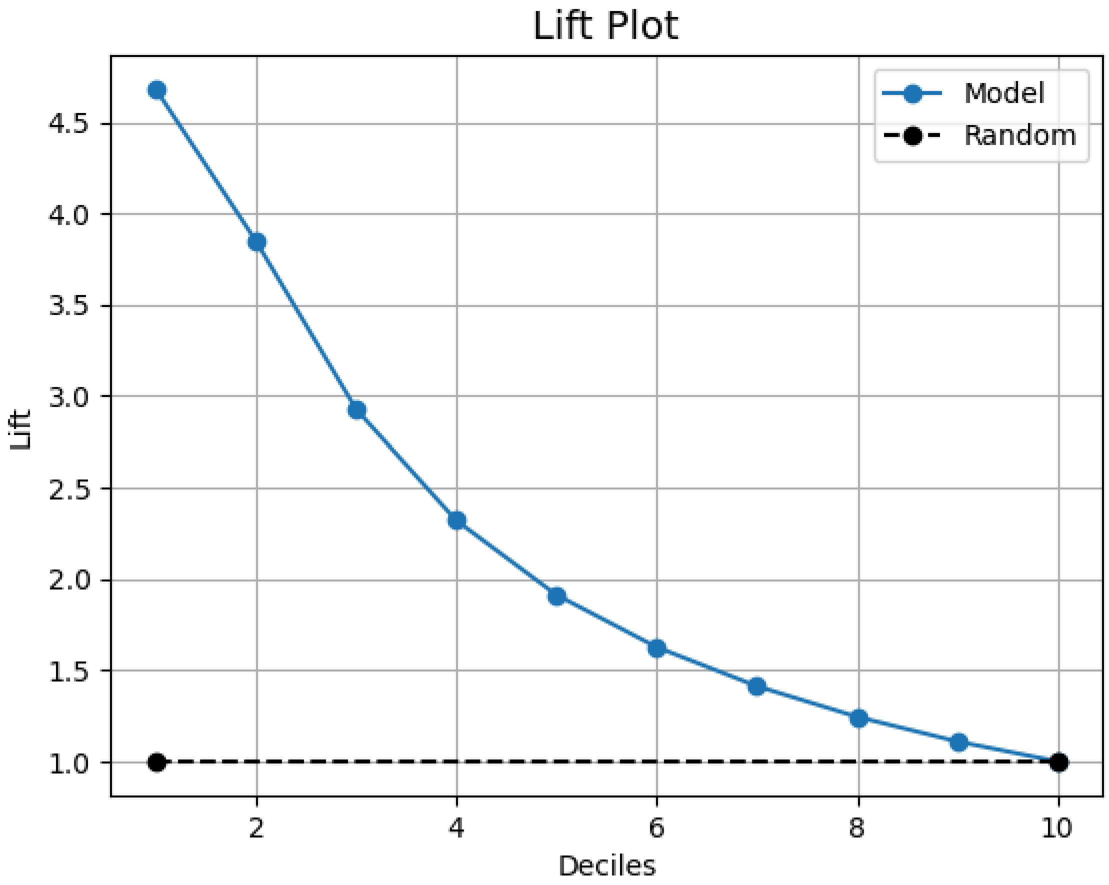


Appendix B: Model Performance Metrics

	Hyperparameters	F1 Score	Recall	Accuracy	Precision	AUC
Decision Tree	Criterion: entropy Max depth: 11 Min leafs: 10 Min Split: 50	0.728668	0.682398	0.900321	0.781669	0.9
Decision Tree (undersampled)	Criterion: gini Max depth: 9 Min leafs: 9 Min Split: 2	0.704801	0.803872	0.867922	0.627470	0.91
Bootstrap	Criterion: entropy Max depth: 11 Min leafs: 10 Min Split: 50	0.737052	0.684973	0.904139	0.797702	0.91
Bootstrap (undersampled)	Criterion: gini Max depth: 9 Min leafs: 9 Min split: 2	0.719998	0.800064	0.877947	0.654499	0.92
Random Forest	Criterion: entropy Max depth: 15 Min leaf: 6 Min split: 5	0.668566	0.553631	0.892337	0.843727	0.9
Random Forest (undersampled)	Criterion: entropy Max depth: 15 Min leaf: 6 Min split: 5	0.703812	0.831224	0.862778	0.610269	0.92
XGBoost	Learning rate: 0.1 Max depth: 8 Estimators: 100	0.756533	0.701008	0.911503	0.821610	0.94
XGBoost (undersampled)	Learning rate: 0.1 Max depth: 7	0.724563	0.816154	0.878294	0.651455	0.93

	Estimators: 100					
Gradient Boost	Learning rate: 0.1 Max depth: 8 Estimators: 50	0.750239	0.695699	0.909146	0.814057	0.93
Gradient Boost (undersampled)	Learning rate: 0.2 Max depth: 8 Estimators: 30	0.725355	0.826129	0.877294	0.646493	0.93
Neural Network	Batch size: 128 Hidden layer: (3,) Solver: relu Max iter: 1000	0.357600	0.262255	0.815190	0.561875	0.76
Neural Network (undersampled)	Batch size: 128 Hidden layer: (3,) Solver: relu Max iter: 1000	0.432652	0.454092	0.766412	0.413145	0.73

Appendix C: CHAMPION Model Lift Plot



Appendix D: Group 1 ISA491 Loan Default Jupyter Notebook

https://colab.research.goog491_finalproject_playground.ipynble.com/drive/13ULStp6VWDzlpc4qYjl-iixbXKYvfeRx?usp=sharing