1 Machine Learning Assignment Report

Overview

The dataset has 20,132 observations, 159 features (42 continuous, 117 discrete), and an imbalanced target with 21% labeled as Class 1. This imbalance likely leads to prediction bias, as shown by a Dummy Classifier achieving 78.32% accuracy, which serves as the baseline. Given the large training dataset, we chose bias reduction as the main research direction to enhance the model's overall predictive performance.

Data processing

Features were categorized into continuous and discrete categories. For continuous features, we applied standardization followed by PCA, retaining the top four components to reduce dimensionality. Alternative preprocessing methods, such as log transformation, were explored but yielded sub-optimal results, sometimes even decreasing out-of-sample accuracy. After multiple feature combination trials, we discovered that using all discrete features without any processing resulted in the greatest performance improvement across various models, possibly due to the dependence of continuous features on discrete ones in causal inference.

Hyperparameter tuning

To find the optimal hyperparameters, we used grid search with cross-validation (GridSearchCV) where tuned parameters can be seen in Appendix. The hyperparameters were selected on the basis of maximizing the F1 score, which can make them suitable for the unbalanced data set. Additionally, "class_weight" and "bootstrap" were incorporated as tuned hyperparameters to help mitigate the risk of biased predictions. 5-fold cross-validation is used to reduce the risk of overfitting and provide more robust performance on the different data subsets.

Model research

In the model selection phase, we tested several classifiers detailed in Appendix. While more sophisticated models like neural networks were considered, research has shown they often underperform on inference tasks with tabular data (Borisov et al., 2024). Grinsztajn, Oyallon, and Varoquaux (2022) demonstrated tree-based methods typically outperform others in such scenarios. Given that our dataset is tabular data and based on the average F1 score with corresponding standard deviation, we opted for Random Forest. This choice leverages ensemble learning techniques and effectively handles diverse feature types. Consequently, we eliminated continuous features, which led to significant improvements in model performance. The Random Forest model not only achieved the highest F1 score with robustness, ensuring a strong balance of precision and recall, but also recorded the highest average accuracy.

2 Conclusion

Our final model, with optimized hyperparameters, achieved an out-of-sample F1 score of 64% and accuracy of 87% using 5-fold cross-validation, ensuring robust performance estimation. The distribution of predicted labels in the test set closely mirrored the label distribution in the training set. This similarity, assuming consistent data-generating processes for both sets, suggests that our algorithm has effectively captured the underlying patterns in the data generating process.

Appendix

Table 1: Random Forest Hyperparameters for GridSearchCV

hyperparameter	grid
n_estimators	{100,200,300,400,500}
criterion	{gini,entropy,log-loss}
bootstrap	{True, False}
class_weight	{balanced, balanced_subsample}

Table 2: Models and Hyperparameter Tuning Ranges

Model	**Tuning Parameter**	**Range**
SGDClassifier	penalty	{’l1’, ’l2’, ’elasticnet’}
	alpha	{0.001, 0.01, 0.1}
	learning_rate	{’constant’, ’optimal’, ’adaptive’, ’invscaling’}
	eta0	{0.001, 0.01, 0.1}
LinearSVC	penalty	{’l1’, ’l2’}
	loss	{’hinge’, ’squared_hinge’}
	C	{0.001, 0.01, 0.1, 1}
KNeighborsClassifier	n_neighbors	{3, 4, 5}
	weights	{’uniform’, ’distance’}
ExtraTreesClassifier	n_estimators	{100, 200, 300}
	criterion	{’gini’, ’entropy’, ’log_loss’}
	bootstrap	{True, False}
	oob_score	{True, False}
GradientBoostingClassifier	loss	{’log_loss’, ’exponential’}
	learning_rate	{0.001, 0.01, 0.1}
	n_estimators	{300}
	criterion	{’friedman_mse’, ’squared_error’}
	max_features	{’sqrt’, ’log2’, None}
XGBClassifier	booster	{’gbtree’, ’dart’}
	eta	{0.1, 0.3, 0.5}
	objective	{’binary:logistic’, ’binary:logitraw’}
LinearDiscriminantAnalysis	solver	{’svd’, ’lsqr’, ’eigen’}
	shrinkage	{’auto’, None}

References

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2024, June). Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(6), 7499–7519. Retrieved from `http://dx.doi.org/10.1109/TNNLS.2022.3229161` doi: 10.1109/tnnls.2022.3229161

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, *35*, 507–520.