Big Data Analysis and Project

Assignment 1: Part C (Modelling)

Possakorn a1873765

Part 1: Problem Description

Expanding upon the summary provided in Part B, a comprehensive dataset incorporating order transactions, customer profiles, geolocation data, payment details, and customer reviews sourced from a leading Brazilian e-commerce platform was methodically transformed to facilitate an in-depth analysis. The primary focus of this examination was to identify **Which key platform features have the potential to most effectively optimize customer churn**.

Subsequent to the transformation, a meticulous process of feature engineering was undertaken to derive potent attributes that could significantly contribute to the analysis. The derived attributes and their characteristics have been outlined in the following **Table01**.

Table 1: Feature Attribute by data sources

data_sources	feature_name
customer review	avg_review_score
	num_negative_review
	num_review
	review_duration
geolocation	avg_delivery_dis
	delivery_diff_duration
	delivery_duration
	expect_delivery_duration
	handle_diff_duration
	order_same_city
	order_same_state
order transaction	avg_basket_size
	freight_cost
	frequency
	gross_sales
	monetary
	net_sales
	num_items
	order_last_180_d
	order_last_90_d
	recency
payment history	average_installments
	payment_duration
	payment_method
	voucher_used

Part 2: Data Pre-processing

The preliminary data necessitated a restructuring process, facilitated by join operations and aggregations at the customer level. Considering the pronounced class imbalance within churn classification, numerous potential models have been identified for output prediction. These will be detailed in the subsequent sections, however, the pre-processing procedures are enumerated below, inclusive of the corresponding code references:

- **The normalization of the feature space**, labelled as 'step_normalize', was actualized through feature scaling.
- **Encoding of categorical variables**, referred to as 'step_dummy', facilitated the execution of linear models.
- **The elimination of zero-variance predictors**, or 'step_zv', was implemented to enhance model performance.
- Some algorithms are susceptible to outliers; hence, **an outlier handling strategy**, 'step_YeoJohnson', was devised.
- **A collinearity reduction**, or 'step_corr', was employed to identify and eliminate predictors exhibiting correlations with other predictors, thereby preventing destabilization of model estimates.

It is noteworthy that tree-based models inherently handle categorical variables without encoding. Lastly, due to the extreme imbalance ratio, neither undersampling of the majority class nor oversampling of the minority class was conducted. This was to preclude the introduction of significant bias or overfitting.

Part 3: Model Selection

Addressing non-linear complexities in our dataset necessitated a diverse array of machine learning models:

- **Logistic Regression (logistic_reg)**: Primarily applied in linear settings, this model can cater to non-linear scenarios through effective feature engineering, acting as a robust baseline with comprehensible interpretability(Jain, Khunteta and Srivastava, 2020).

- **Support Vector Machine with Radial Basis Function Kernel (svm_rbf)**: Known for managing non-linear relationships via the kernel trick, this specific kernel proves adept at dealing with high-dimensional space data and complex decision boundaries(Y., 2022).

- **Light Gradient Boosting Machine (lgbm)**:A gradient-boosting framework using tree-based algorithms, LGBM excels on imbalanced datasets, capturing complex non-linear relationships efficiently, and naturally offers feature importance rankings(Ke *et al.*, 2017).

- **eXtreme Gradient Boosting (xgboost)**: This model, known for its speed and efficiency, offers a unique approach to handling non-linearities in data. Like random forests, it also uses an ensemble of decision trees, but it improves upon this concept by minimising loss function, thereby enhancing performance(Chen and Guestrin, 2016).

Considering the class imbalance and non-linear nature of the dataset for churn prediction, these models, each uniquely equipped to tackle these challenges, enable nuanced and precise predictions despite the inherent complexity.

Part 4: Model Refinement

Model refinement, consisting of hyperparameter tuning and judicious data splitting, was crucial in our analysis. We adopted a 'time slicing' approach, segregating data into separate time windows for model training and testing, a strategy that is particularly effective for temporal data such as customer churn or time series prediction(Gattermann-Itschert and Thonemann, 2021) as shown in **Figure01**. This setup mirrored real-world circumstances, enhancing the models' predictive utility, which setting the feature window for 12 months and label window for 3 months as out-of-period.
And, the tuning of hyperparameters also provide through the k-fold cross validation to control the behavior of the machine learning algorithm to optimize its performance following **Table02**.

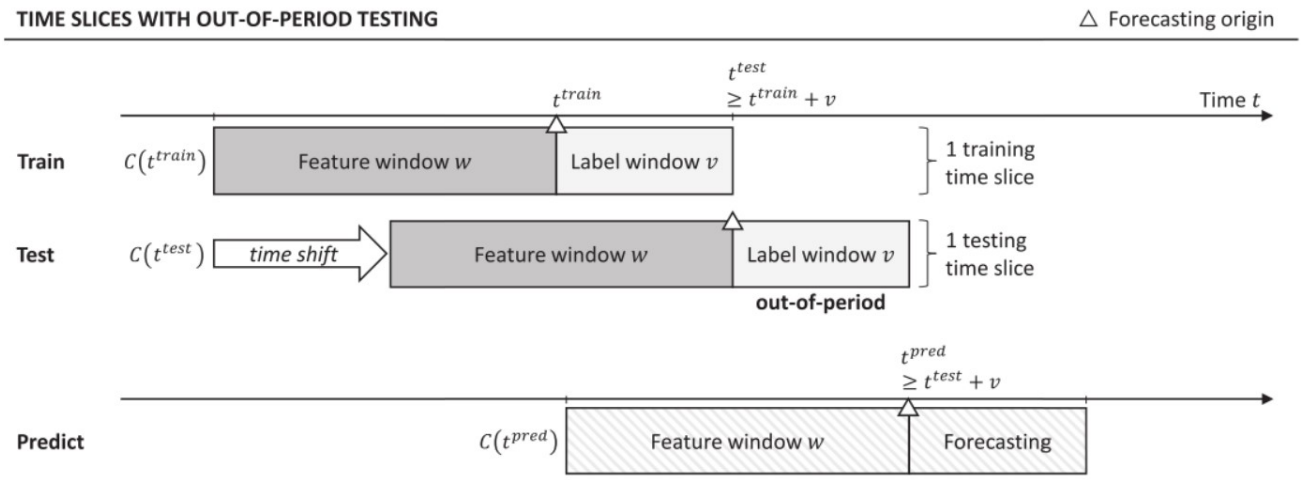

Figure 1: Time slice concept illustrated with out-of-period testing.

Table 2: Hyperparameters and ranges used during grid search

Classifier	Hyperparameter	Values	tuned_values
Logistic Regression	Regularization	[L1, L2 regularization]	L2
	Penalty term	tune()	1e-10
Suport Vector Machine	Kernal	Radial Basis Function(RBF)	
	Cost	tune()	2.378
	Rbf_sigma	tune()	1e-05
LightGBM	Trees	default = 1000	
	mtry	range[1, floor(sqrt(number of feature))]	1
	min samples per leaf	range[1, 10]	5
	learning rate	range[0.01, 0.3]	1.023
XGboost	Trees	default = 1000	
	mtry	range[1, floor(sqrt(number of feature))]	3
	min samples per leaf	range[1, 10]	1
	learning rate	range[0.01, 0.3]	1.023

Part 5: Performance Evaluation

To accurately evaluate our models on the imbalanced dataset, we employed two key performance metrics: accuracy and Area Under the Receiver Operating Characteristic curve (ROC AUC). Accuracy assesses the proportion of correct predictions, while ROC AUC, being insensitive to class imbalance, measures the ability of the model to distinguish between classes across varying thresholds. Comparisons were drawn between models pre and post hyperparameter tuning following **Table03**, ensuring fairness in evaluation and ascertaining the effectiveness of the tuning process in model performance improvement.

Table 3: Model performance comparing before and after tuning

	Before Tuning			After Tuning		
Model	Accuracy - train	Accuracy - test	ROC AUC	Accuracy - train	Accuracy - test	ROC AUC
lgbm	1.000	0.994	0.477	0.989	0.989	0.502
logistic	0.993	0.994	0.399	0.993	0.994	0.398
svm	0.993	0.994	0.494	0.993	0.994	0.427
xgboost	0.993	0.994	0.430	0.993	0.994	0.447

Part 6: Results Interpretation

"Upon analysing the model results, the LightGBM (Lgbm) stood out, demonstrating superior accuracy and the highest ROC AUC, making it apt for dealing with imbalanced classes, both before and after parameter tuning (see **Table02**).

In contrast, XGBoost, although powerful, exhibited lesser performance than LGBM due to increased computational demands with larger datasets. Likewise, SVM and Logistic Regression models posed challenges; SVM's computational requirements restricted its effectiveness with substantial datasets, and Logistic Regression had difficulty capturing complex data patterns. Hence, LGBM appeared to be the optimal model, attaining first place in performance metrics while excelling in speed and flexibility with sizeable, imbalanced datasets. Still, the overall performance of the report was somewhat curtailed due to insufficient data capturing customer behaviour, particularly regarding one-time customers, and constraints stemming from highly specific and private data sources.

To answer the research question—**"What are the key platform features that most effectively minimize customer churn?"**—we turn to **Table04**. Here, geolocation stands out as a pivotal feature affecting churn, mirroring customer behaviour across different locations. Also, transaction information, specifically recency, net sales, payment duration, and customer review factors, such as the average review score, significantly influence churn.

Table 4: Individual feature importance on each attribute (top features).

Variable	Importance
recency	0.301
net_sales	0.196
payment_duration	0.194
average_installments	0.175
state_SP	0.080
avg_review_score	0.031
order_same_state	0.024

Reference

Chen, T. and Guestrin, C. (2016) "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. New York, NY, USA: Association for Computing Machinery (KDD '16), pp. 785–794. Available at: https://doi.org/10.1145/2939672.2939785.

Gattermann-Itschert, T. and Thonemann, U.W. (2021) "How training on multiple time slices improves performance in churn prediction," *European Journal of Operational Research*, 295(2), pp. 664–674. Available at: https://doi.org/https://doi.org/10.1016/j.ejor.2021.05.035.

Jain, H., Khunteta, A. and Srivastava, S. (2020) "Churn prediction in telecommunication using logistic regression and logit boost," *Procedia Computer Science*, 167, pp. 101–112. Available at: https://doi.org/https://doi.org/10.1016/j.procs.2020.03.187.

Ke, G. *et al.* (2017) "LightGBM: A highly efficient gradient boosting decision tree," in I. Guyon et al. (eds.) *Advances in neural information processing systems*. Curran Associates, Inc. Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Y., T.V.A.S., Nguyen Nhu AND Ly (2022) "Churn prediction in telecommunication industry using kernel support vector machines," *PLOS ONE*, 17(5), pp. 1–18. Available at: https://doi.org/10.1371/journal.pone.0267935.