Betsson Task

RQ: For a given day, predict whether a customer will call the Customer Service in the following 14 days?

Jake Bonnici

June 28, 2023

1 Objective

The objective of this report is to propose a solution for the problem of predicting which customers will contact the Customer Service within the next 14 days, based on their past behavior. By achieving this goal, the overall operational efficiency of Betsson can be enhanced. To accomplish this, the report delves into data analysis, including exploratory data analysis, as well as feature engineering and selection. Additionally, a machine learning algorithm is employed to tackle the binary classification problem at hand.

2 Methodology

The target variable in this case is presented in encoded manner, ”0” and ”1”, where 0/1 represents the customer not contacting/contacted customer support in the next 14 days, respectively. The class label are segmented as 98.30 % and 1.70 %, for 0 and 1, respectively. Therefore, providing a heavily imbalanced dataset structure. The data consisted of a total of 270 columns. The customer identifier column was used to merge the 270 features with the separate csv having the customer‘s country. The columns were then feature engineered to replace the variables which were recorded weekly transformed to their mean reducing the total amount of features to 46.

Principal component analysis was then performed on the appropriate columns in order to reduce the dimensionality of the feature space. Figure 1, shows the cumulative variance ration and explained variance ratio. The dotted green line represents the cumulative variance required to have a confidence level of 95 % with the intersecting vertical line showing the respective required number of components.

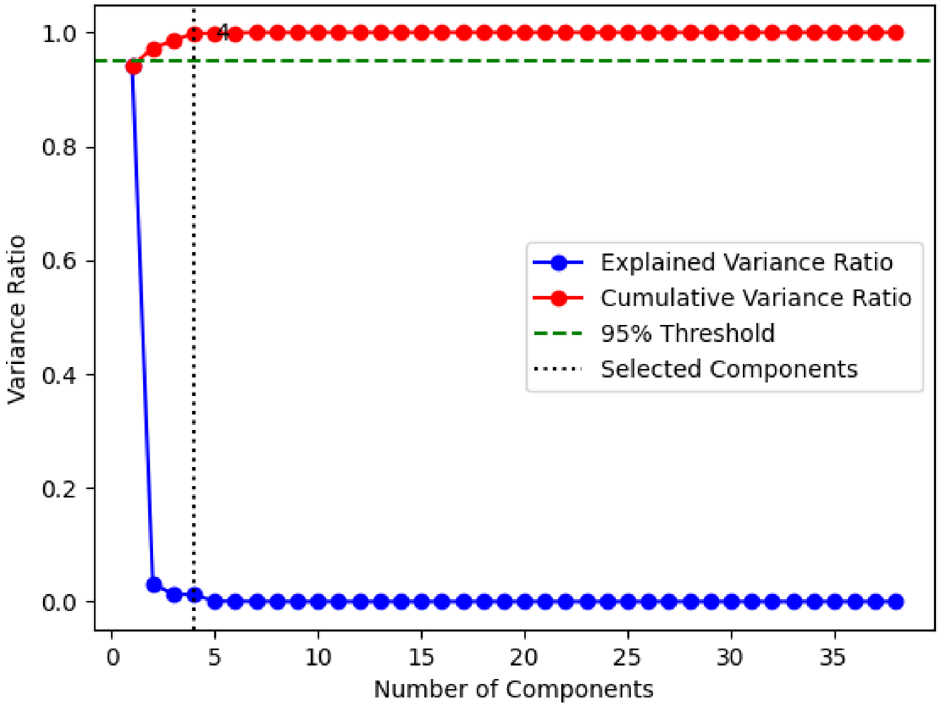

Figure 1: PCA performed on 38 features

The provided train data set was then separated into an 80/20 split for training and validation purposes.

Various machine learning algorithms were employed to address the problem at hand. The algorithms utilized in the analysis include Voting Classifier, Bagging Classifier, Balanced Bagging Classifier, Balanced Random Forest, ADA, Random Forest, Logistic Regression, XGBoost, and SVM. Each algorithm serves a distinct purpose, such as ensemble methods, boosting, and traditional classifiers. Additionally, feature selection techniques were applied alongside these algorithms to identify the most relevant features for the prediction task. Parameter grids were defined to enable hyper parameter tuning during the training phase. By leveraging this diverse set of algorithms, the objective was aimed to explore different modeling approaches and identify the most effective solution for the given problem.

Furthermore, specific techniques were employed to enhance the performance of the machine learning models. Firstly, class weights were utilized to address class imbalance within the dataset and adjust its learning accordingly. Class weight assignment gives greater importance to samples from the minority class during training, mitigating the potential bias towards the majority class and improving the models' ability to accurately predict both classes. Additionally, the model applied the SMOTE (Synthetic Minority Oversampling Technique) and Tomek links method (identifies and removes data points that are close to the decision boundary between different classes), which involves oversampling the minority class and undersampling the majority classes. This technique helps alleviate the impact of class imbalance and can lead to better performance in detecting the minority class instances.

Additionally, feature selection methods were applied to identify the most relevant features for the prediction task. The Recursive Feature Elimination with Cross-Validation (RFECV) technique was utilized, which iteratively removes less important features while evaluating the model's performance through cross-validation. RFECV helps to identify the subset of features that contribute the most to the prediction accuracy and can reduce dimensionality effectively.

Another feature selection technique used was Recursive Feature Elimination (RFE), which follows a similar iterative process to RFECV but does not involve cross-validation. RFE eliminates less important features based on their individual importance, allowing for dimensionality reduction. This approach is beneficial when computational resources are limited or when working with smaller datasets.

Lastly, the SelectKBest method was employed, which selects the top k features based on statistical tests or scoring methods. This approach evaluates the relevance of each feature individually and ranks them accordingly. SelectKBest allows for the selection of a fixed number of features based on their individual merit, helping to reduce the dimensionality of the dataset and focus on the most informative features for the prediction task.

By incorporating these techniques, the aim was to enhance the models' performance by addressing class imbalance and selecting the most relevant features, ultimately improving the accuracy and efficiency of the predictions for customer behavior.

The pipeline creation and grid search process formed a crucial part of the methodology. By constructing pipelines for each machine learning algorithm and conducting grid searches, an extensive exploration of various model configurations was performed. This approach allowed for thorough hyperparameter tuning and feature selection to identify the most effective combination of techniques for the prediction task.

The initial step of applying the StandardScaler ensured that all features were appropriately scaled, addressing potential discrepancies in their magnitudes. This preprocessing step contributed to improved model performance by normalizing the data.

The subsequent implementation of the SMOTETomek algorithm was particularly valuable in tackling the issue of class imbalance within the dataset. By generating synthetic samples for the minority class and removing Tomek links, this resampling technique helps to achieve a more balanced class distribution. This step was essential for training models that can adequately capture patterns from both the majority and minority classes, leading to more accurate predictions.

The feature selection techniques, such as RFECV, RFE, and SelectKBest, play a crucial role in dimensionality reduction and identifying the most informative features for the prediction task. By evaluating the relevance of each feature individually or iteratively, these methods help to focus the models' attention on the most influential aspects of the data. This approach not only improves prediction accuracy but also reduces computational complexity by reducing the number of features considered.

The grid search, performed using GridSearchCV, allows for an extensive exploration of hyperparameter configurations for each pipeline. By defining a parameter grid for each algorithm, the grid search systematically evaluates multiple combinations of hyperparameters using cross-validation. The scoring metric, ROC AUC, provides a reliable measure of model performance in classification tasks, enabling the identification of the best-performing model for predicting customer behavior.

Overall, the combination of pipelines, including data preprocessing, resampling, feature selection, and hyperparameter tuning through grid search, forms a comprehensive approach to tackle the assignment's objectives. Through this systematic methodology, the objective was to identify the most effective modeling techniques and hyperparameter settings to optimize the accuracy and efficiency of customer behavior predictions.

In addition to the created pipelines and conducted grid searches, evaluation metrics such as ROC AUC, accuracy, precision, recall, and F1 score were used to assess the performance of the machine learning models. These metrics provided a comprehensive understanding of the models' predictive capabilities, accounting for their discrimination ability, overall accuracy, and the balance between precision and recall. By considering multiple evaluation metrics, the objective was to evaluate and compare the models' performance across different aspects, providing a robust assessment of their effectiveness in predicting customer behavior. Only some of these metrics were considered, as discussed in the results, Section 3.

Finally, the threshold values were adjusted to find the best threshold that maximizes the recall score for class 1. The best model, along with its associated metric scores and threshold, were stored for further analysis. Then, the model was used to predict probabilities for the validation data, and the threshold is adjusted using the best threshold found during training.

3 Results

The best model selected for the given use case was a pipeline consisting of several steps: data scaling with StandardScaler, synthetic minority oversampling with Tomek links (SMOTE-Tomek), feature selection with SelectKBest (using 8 best features), and classification with RandomForestClassifier with balanced class weights. The best threshold value determined during training was of 0.6.

Table 1: Best Metric Scores for RandomForestClassifier

Class	Precision	Recall	F1-Score	Accuracy
0	0.993	0.898	0.944	0.894
1	0.1	0.654	0.173	0.894

The ROC-AUC score, which measures the model's ability to distinguish between positive and negative instances, was 0.852.

The following are the metrics and confusion matrix of the RandomForestClassifier before the threshold adjustment.

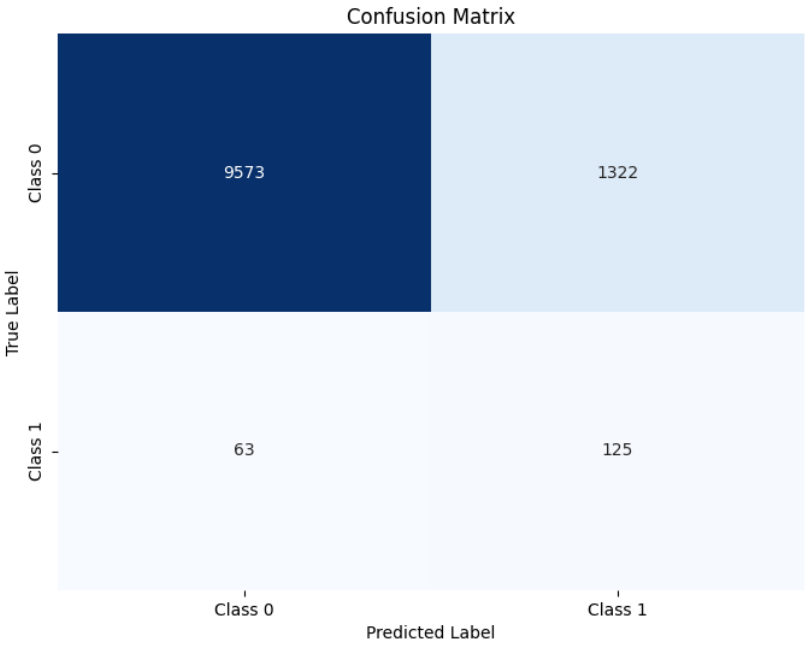

Figure 2: Random Forest Classifier Confusion matrix after threshold adjustment.

4 Discussion

The primary objective of this project was to gain a comprehensive understanding of customer behavior and identify those most likely to contact customer support within the next 14 days. To achieve this goal, the evaluation process involved selecting appropriate metrics, with a particular focus on recall.

Recall, also known as sensitivity or true positive rate, measures the model's ability to correctly identify positive instances, in this case, the customers who require assistance. A high recall score indicates that the model is successful in capturing a significant portion of dissatisfied customers, minimizing the number of false negatives. False negatives represent cases where customers in need of support go undetected, potentially leading to a poor customer experience and lost opportunities for intervention.

While maximizing recall is crucial, it is important to consider the trade-off with precision. Precision measures the proportion of correctly identified positive instances among all instances labeled as positive. In other words, it indicates how many of the predicted customer support requests are genuine. A lower precision score may result from including some satisfied customers in the predictions, leading to false positives. False positives occur when customers who do not require support are mistakenly identified as needing assistance. This trade-off implies that allocating resources to customers who are already satisfied could be a potential drawback. In the prototype stage, the emphasis is on capturing a subset of genuinely dissatisfied customers, even if it means including some satisfied customers in the predictions. This approach ensures that a larger number of dissatisfied customers are addressed and their concerns are appropriately handled. As the model evolves, fine-tuning can be performed to strike a better balance between recall and precision, minimizing both false negatives and false positives.

By prioritizing recall, the model aims to optimize the identification of dissatisfied customers and provide timely support, thereby improving customer satisfaction, loyalty, and overall business performance.

5 Conclusion

The current model has shown promising results, but there is still significant room for improvement through further analysis, increased computational power, and additional time investment. One potential avenue for enhancement is through more extensive feature engineering, which involves carefully selecting and transforming input variables to better capture the underlying patterns in the data. Additionally, employing feature selection methods like Recursive Feature Elimination with Cross-Validation (RFECV) could help identify the most informative features, although this approach can be computationally intensive.

To further optimize the model's performance, a larger parameter grid can be explored during the hyperparameter tuning process. This allows for a more thorough search of the parameter space to find the optimal combination of model settings that maximizes performance.

Overall, by conducting a deeper analysis, leveraging advanced techniques for feature engineering and selection, and fine-tuning the model's parameters, there is potential to enhance its performance and achieve even better results. However, these improvements require careful consideration of computational resources, as they may involve increased computational complexity and longer training times.