Analysing a Churn Banking Dataset to Predict Customer Loss Using A Tuned Decision Tree

720091272

1 Introduction

Financial management strategy is a core structure of business strategy management. A component of this is customer retention, which reduces loss. Customer retention is noted to be at least 5x less expensive than customer acquisition on average [8]. To test this using machine learning, we must ask, "Can we accurately predict customer churn and fine tune it using a decision tree model?". This project covers results from experimentation and fine tuning a decision tree model in 3 configurations for a banking dataset [2]. Before that, we will go over the dataset, applied techniques, and the model itself.

Developing a model to predict churn based on existing features will help provide a structural foundation for both financial and business strategy, and equip banks (or other organisations) with a tool to predict customer loss based on numerical and categorical data of the customers.

2 Dataset

The dataset is composed of bank customer information, and is broken down into 13 features (including the target, Exited), 10 of which are used in the experiment. Excluding customer id and surname, there is CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, and EstimatedSalary. This dataset is medium sized with 10,000 rows of data, and has both numerical and categorical features to challenge and improve the performance of the model. There is one core issue with this dataset, however.

The dataset is highly imbalanced. Of predicted target feature Exited, 79.6% of rows are class 0, and 20.4% are class 1 (See Figure 4a). When a dataset is imbalanced, as noted by Guo et al. (2008), accuracy is heavily weighted to the majority class, causing the classifier model to perform poorly on the minority class. They also cite a decrease in precision and recall of the minority class performance [7]. As such, necessary techniques to improve model performance, particularly one that will fix data imbalance are necessary.

3 Machine Learning Techniques

By nature of the dataset and target y (Exited) being a binary feature (0,1), the problem at hand is classification. Due to it being a classification problem, supervised learning is the next step. Supervised learning as defined by Cunningham et al. (2008), is the use of labeled data for training a machine learning model in order to generalise against unlabeled “unseen” data [5].

GridSearchCV (Grid Search Cross-Validation), as defined by (Yan et al. 2022), is an optimisation tool to tune hyperparameters and model selection for machine learning models [13]. In the model used, the hyperameters that are tuned using GCV is max_depth, min_samples_split, min_samples_leaf, class_weight. The tested parameters are then fit to the model and evaluated with train and test accuracy, precision, recall, f1-score. This is used to fine tune the model hyper parameters as an extension of other techniques.

Cost-complexity pruning (CCP) is a technique to reduce the size of a tree, which lowers its complexity. This reduces overfitting and hence ability to generalise unseen data. Cost Function provided by Ravi et al. (2017)[11] and Friedman et al. (2009)[12]:

$$R_\alpha(T) = R(T) + \alpha \cdot |\text{Leaves}(T)| \tag{1}$$

This equation is the core of CCP, as it balances accuracy with tree minimisation and simplicity, and is used in the model to calculate the best alpha to be fit alongside GCV for further tuning. CCP is used to prune the tree, the effect of which is shown in Figure 1 and Figure 2

Two models use two different oversampling techniques to evaluate the best performance. The first is Synthetic Minority Over-sampling Technique (SMOTE) which generates synthetic minority class samples to equalise the class imbalance as defined by Blagus et al. (2013)[4]. The second approach is RandomOverSampler (ROS) which instead randomly duplicate the instances of minority classes, as defined by (Benala et al. 2023)[3]

4 Machine Learning Models

Decision trees (DTs), as defined by (Kotsiantis et al. 2007)[9], are a machine learning model that classify instances by sorting them beginning from the root node based on feature values. Each node in a tree is a feature in an instance to be classified, and a branch is a value that the node can assume.

DTs work with numerical and categorical data directly, whereas linear classifiers or neural networks cannot, which is significant for the mixed data types in the dataset. It should be noted that the data is preprocessed (dropped duplicates and null rows) and transformed using label encoder but only for consistency across oversampling techniques. Moreover, DTs don't rely on data relativity but thresholds, removing the need to scale data (Günlük et al. 2021)[6]. Additionally, DT's are strong at handling nonlinear relationships which is important for transformed categorical data (Alshammari et al. 2024) [1].

5 Results and Analysis

Random state of 0 to seed runs and train test split of 0.4 is used across all models and configurations.

Table 1 shows a lack of performance difference when dropping the least important feature compared to an undropped feature run when both are not oversampled. Both of these models perform poorly on the minority class (1) in most cases decreasing the precision, recall, and f1 score up to 48% of majority class performance, due to the imbalanced dataset. RandomOverSampler causes an initial spike in accuracy at baseline and drops 1.59% (89.36->87.77) when using GCV, and no change when using the best alpha. The model still overfits with this method as the train accuracy is 0.97% after GCV+CCP compared to 0.88% test accuracy. SMOTE prevents this and the accuracy increases with both GridSearch and cost complexity pruning, at over 2% increase with each additional configuration. Oversampling improves the performance both with ROS and SMOTE, with the exception of overfitting the training set with ROS. The most significant effect of these techniques is the improved precision, recall, and f1 scores on the minority class 1 which fall within 0.08 of class 0 for SMOTE, and 0.1 for ROS. From no dropped features to SMOTE there was +0.18, +0.38, +0.31 in minority class performance for precision, recall, and f1 respectively.

6 Limitations

The limitation of the experiment is the imbalanced dataset. While fixed with oversampling techniques, RandomOverSampling creates random duplications of existing data and SMOTE Synthetic data. The generated samples may be unrealistic and not as reliable nor authentic as the original dataset, which decreases the applications of the model for real world scenarios with imbalanced data where the pattern recognition of the model is weighted with generated data.

7 Conclusion

4 models with 3 configurations using GCV and CCP were used to fine tune a decision tree model for customer loss prediction. Results found no large discrepency between dropped and undropped features, but both performed poorly on the minority class without oversampling. Oversampling improved the model in both cases, but SMOTE proved to be the best model as, unlike ROS, it did not overfit the training set.

8 Figures

Table 1: Decision Tree Evaluation Across 4 Variations with 3 Configurations Each

EstimatedSalary Dropped						
Model Configuration	**Training**	**Testing**	**Precision**	**Recall**	**F1 Score**	**Support**
Baseline	1.0	0.78575	0 : 0.88 1 : 0.48	0 : 0.85 1 : 0.53	0 : 0.86 1 : 0.50	0 : 3185 1 : 815
GCV	0.902	0.83125	0 : 0.87 1 : 0.61	0 : 0.92 1 : 0.47	0 : 0.90 1 : 0.53	0 : 3185 1 : 815
GCV + CCP Alpha	0.8545	0.85475	0 : 0.87 1 : 0.73	0 : 0.96 1 : 0.46	0 : 0.91 1 : 0.56	0 : 3185 1 : 815
No Dropped Features						
Baseline	1.0	0.7935	0 : 0.88 1 : 0.49	0 : 0.86 1 : 0.53	0 : 0.87 1 : 0.51	0 : 3185 1 : 815
GCV	0.905	0.824	0 : 0.88 1 : 0.58	0 : 0.91 1 : 0.50	0 : 0.89 1 : 0.54	0 : 3185 1 : 815
GCV + CCP Alpha	0.8695	0.8595	0 : 0.88 1 : 0.74	0 : 0.96 1 : 0.47	0 : 0.92 1 : 0.58	0 : 3185 1 : 815
No Dropped Features + SMOTE						
Baseline	1.0	0.8572	0 : 0.86 1 : 0.85	0 : 0.85 1 : 0.86	0 : 0.86 1 : 0.86	0 : 3186 1 : 3185
GCV	0.89	0.879	0 : 0.88 1 : 0.88	0 : 0.87 1 : 0.88	0 : 0.88 1 : 0.88	0 : 3186 1 : 3185
GCV + CCP Alpha	0.899	0.8898	0 : 0.86 1 : 0.92	0 : 0.93 1 : 0.85	0 : 0.89 1 : 0.89	0 : 3186 1 : 3185
No Dropped Features + RandomOverSampler						
Baseline	1.0	0.8936	0 : 0.95 1 : 0.85	0 : 0.84 1 : 0.95	0 : 0.89 1 : 0.90	0 : 3185 1 : 3186
GCV	0.9684	0.8777	0 : 0.96 1 : 0.85	0 : 0.83 1 : 0.97	0 : 0.89 1 : 0.90	0 : 3185 1 : 3186
GCV + CCP Alpha	0.9684	0.8777	0 : 0.93 1 : 0.83	0 : 0.81 1 : 0.94	0 : 0.87 1 : 0.89	0 : 3185 1 : 3186

(a) Unpruned Undropped feature model tree

(b) CCP Pruned Undropped feature model tree

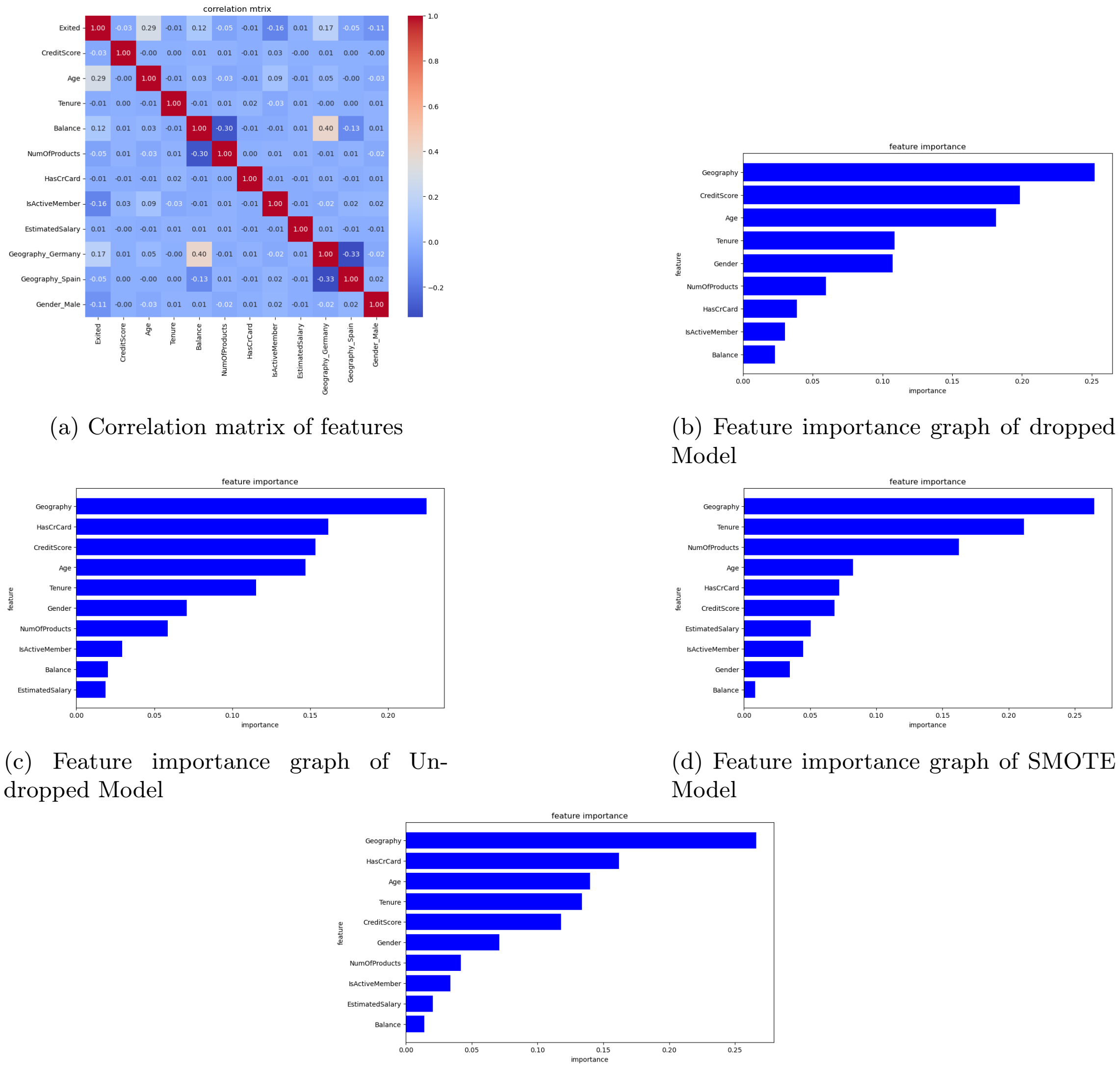
Figure 1: Cost-Complexity pruning tree visualisation on Undropped features run (no oversampling)

(a) Unpruned SMOTE model tree

(b) CCP pruned SMOTE model tree

Figure 2: Cost-Complexity pruning tree visualisation on SMOTE model run (oversampled)

9 Appendix



(a) Correlation matrix of features

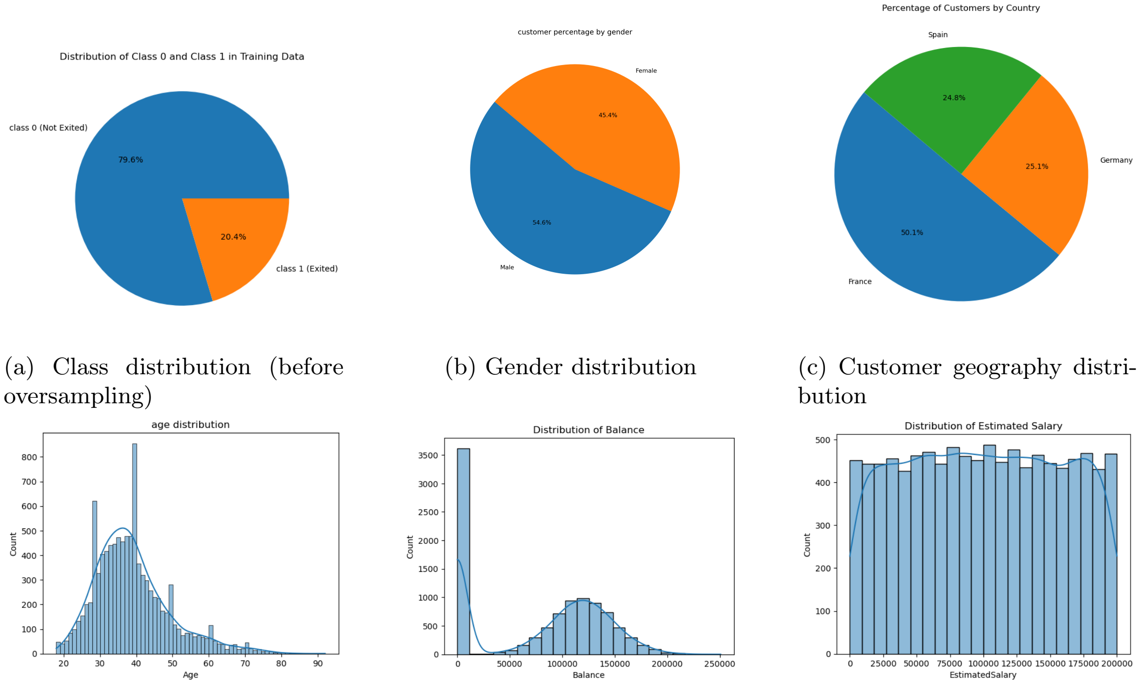
(b) Feature importance graph of dropped Model

(c) Feature importance graph of Undropped Model

(d) Feature importance graph of SMOTE Model

(e) Feature importance graph of ROS Model

Figure 3: Feature correlation and importance 4 modelsgraphs



(a) Class distribution (before oversampling)

(b) Gender distribution

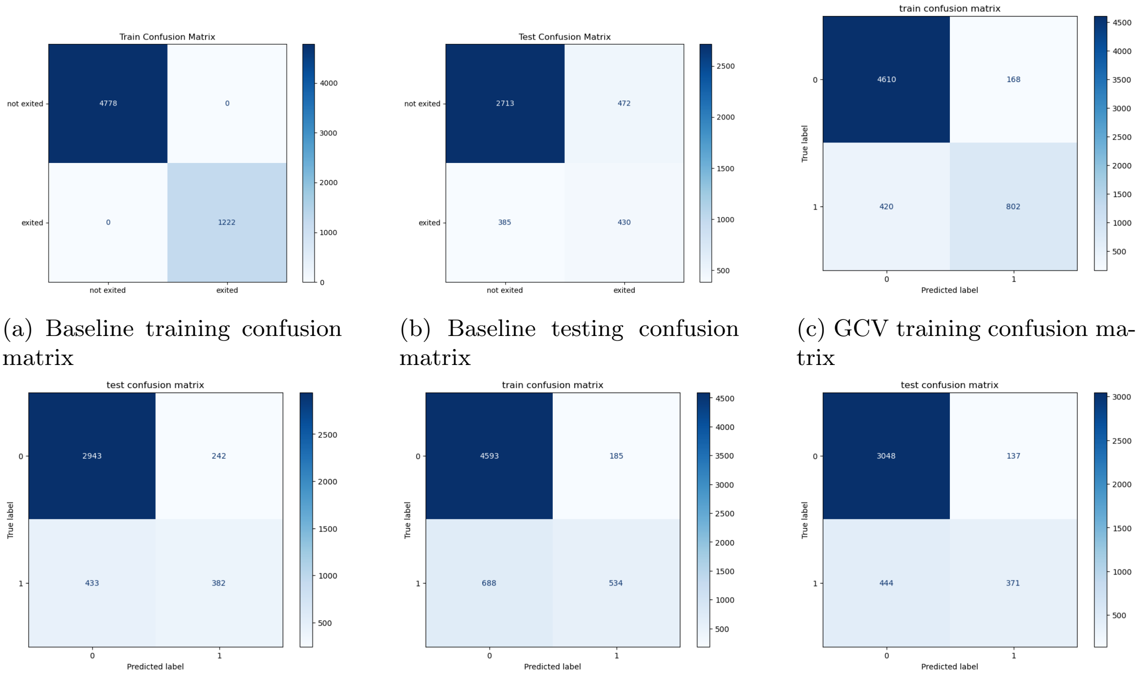
(c) Customer geography distribution

(d) Customer age distribution

(e) Balance distribution among customers

(f) EstimatedSalary distribution among customers

Figure 4: Data statistics and visualisations



(a) Baseline training confusion matrix

(b) Baseline testing confusion matrix

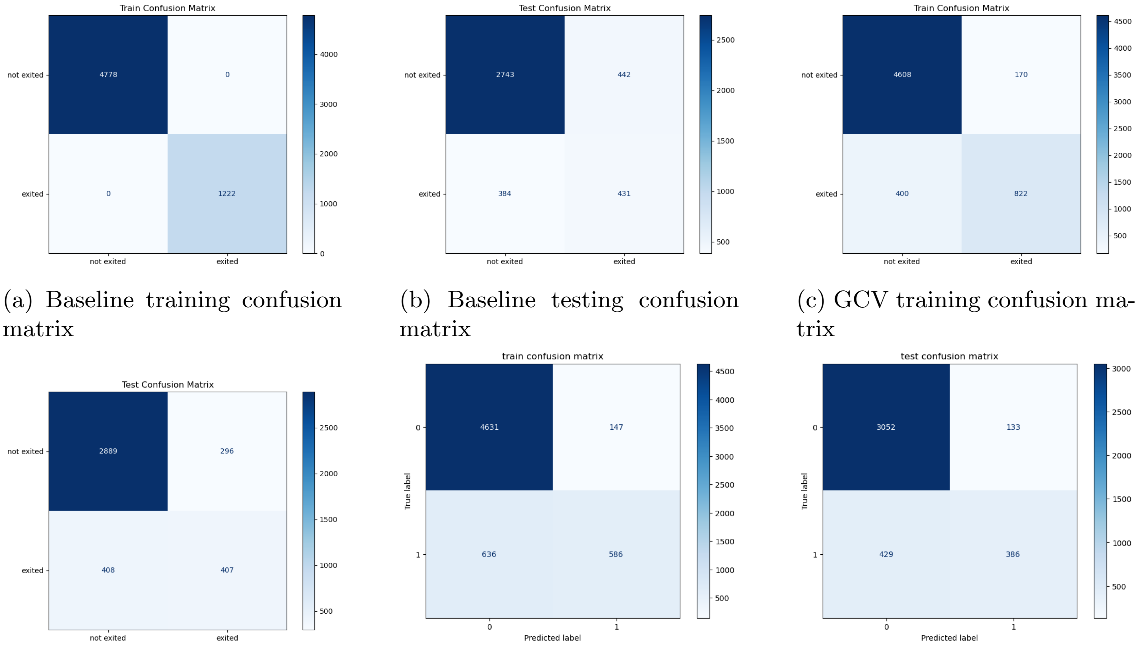
(c) GCV training confusion matrix

(d) GCV testing confusion matrix

(e) GCV and ccp alpha training confusion matrix

(f) GCV and ccp alpha testing confusion matrix

Figure 5: EstimatedSalary dropped model performance



(a) Baseline training confusion matrix

(b) Baseline testing confusion matrix

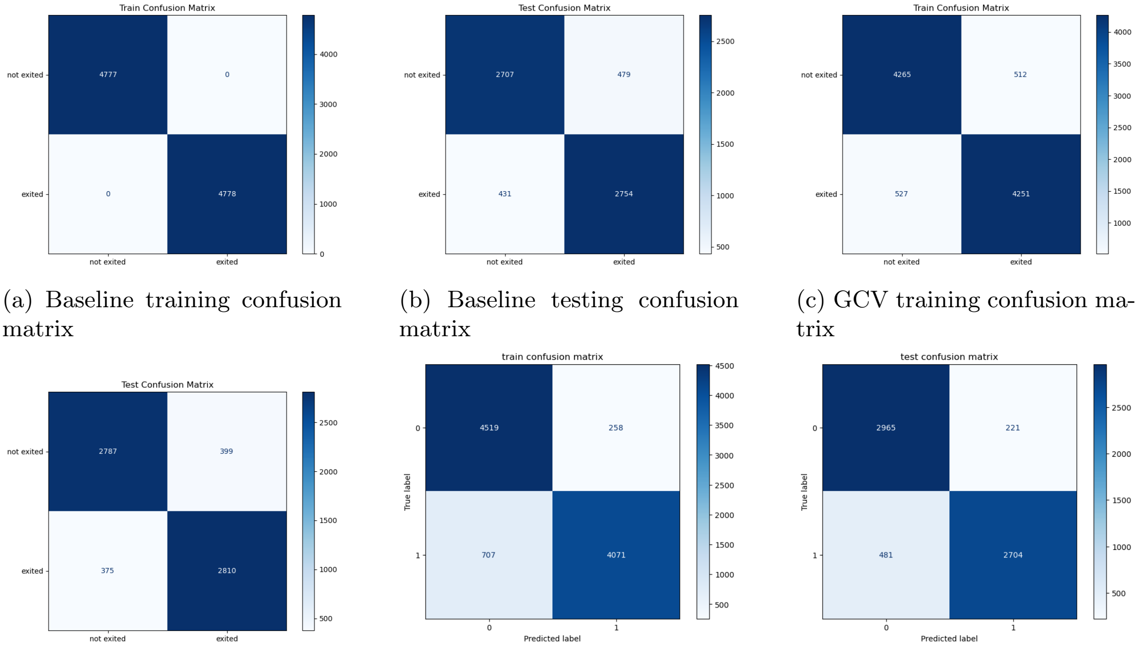
(c) GCV training confusion matrix

(d) GCV testing confusion matrix

(e) GCV and ccp alpha training confusion matrix

(f) GCV and ccp alpha testing confusion matrix

Figure 6: No dropped features model performance



(a) Baseline training confusion matrix

(b) Baseline testing confusion matrix

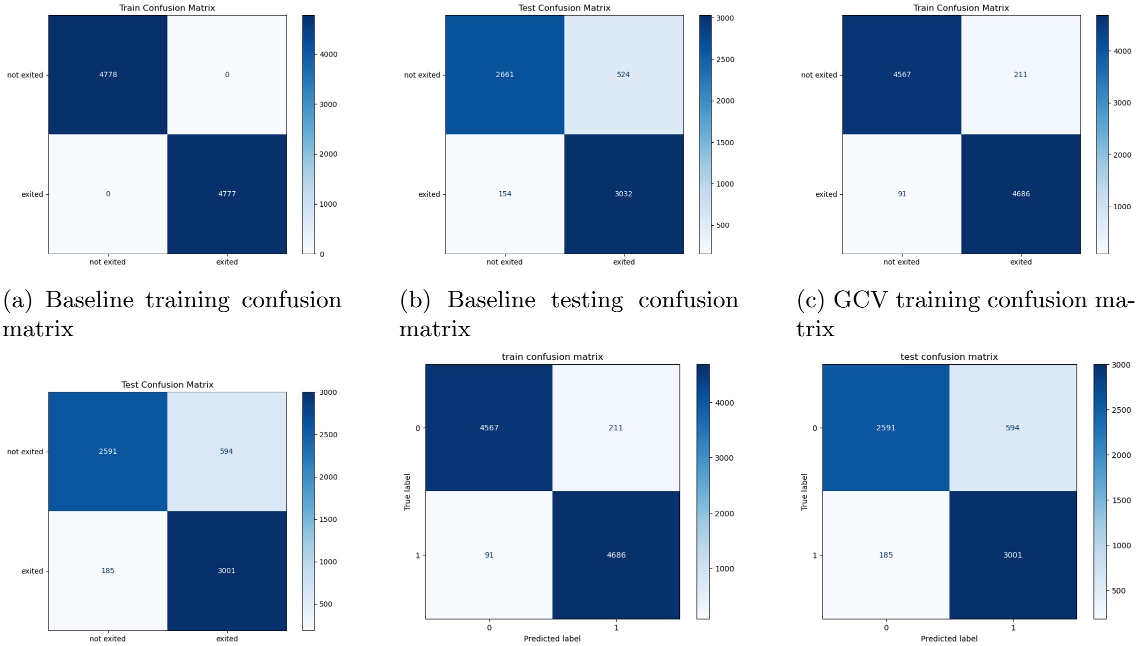
(c) GCV training confusion matrix

(d) GCV testing confusion matrix

(e) GCV and ccp alpha training confusion matrix

(f) GCV and ccp alpha testing confusion matrix

Figure 7: SMOTE model performance



(a) Baseline training confusion matrix

(b) Baseline testing confusion matrix

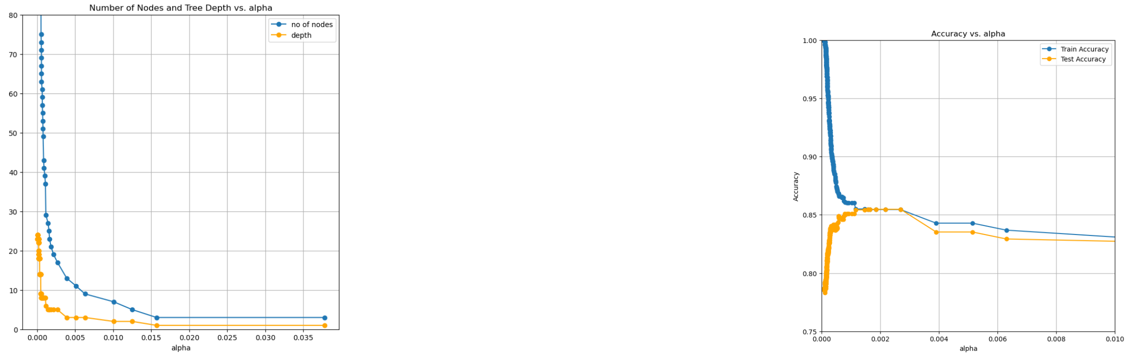
(c) GCV training confusion matrix

(d) GCV testing confusion matrix

(e) GCV and ccp alpha training confusion matrix

(f) GCV and ccp alpha testing confusion matrix

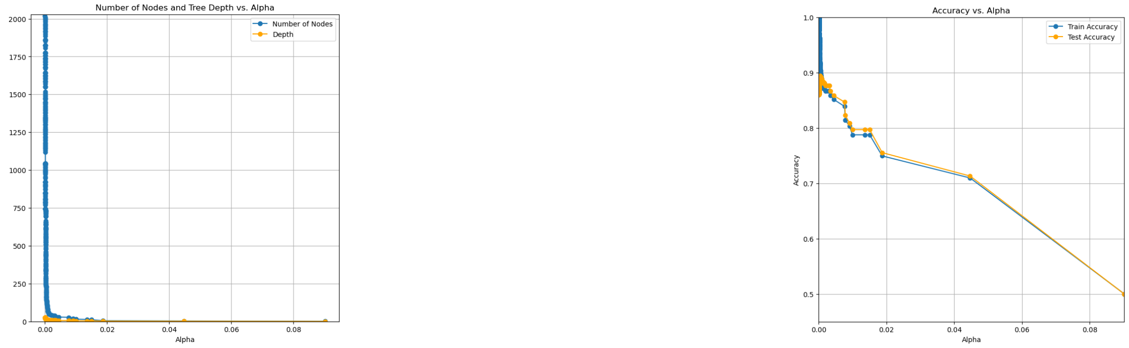
Figure 8: RandomOverSampler dropped model performance



(a) nodes and tree depth vs alpha

(b) test/train accuracy by alpha

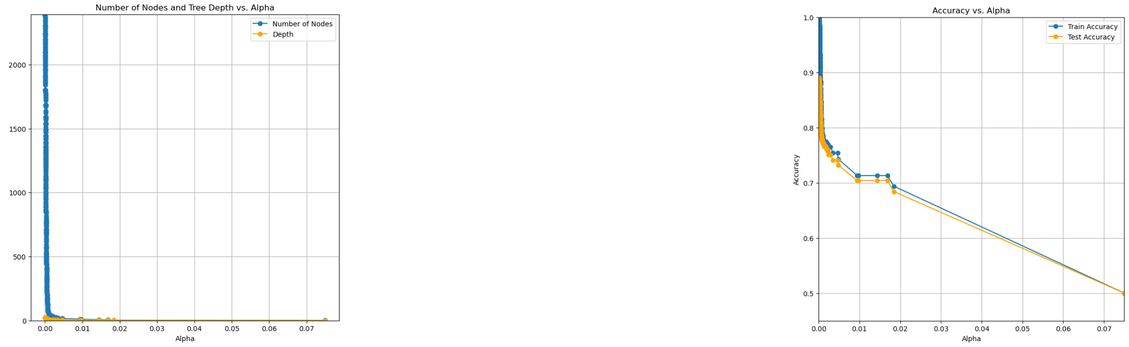
Figure 9: No dropped features CCP graphs



(a) nodes and tree depth vs alpha

(b) test/train accuracy by alpha

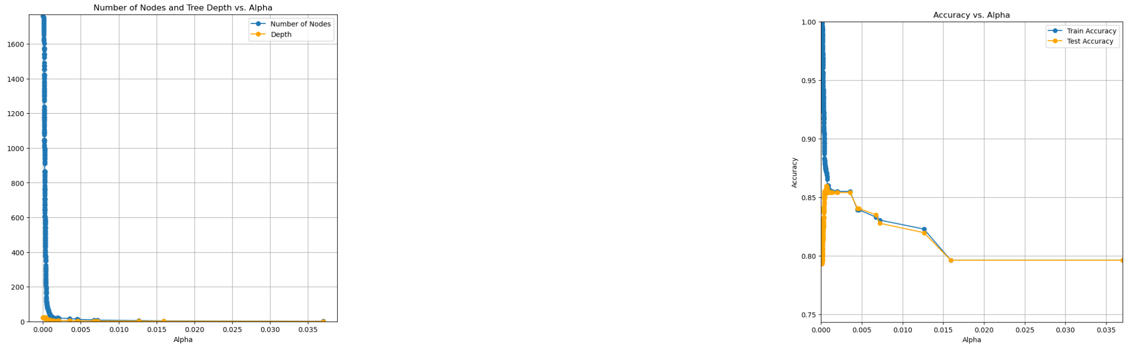
Figure 10: SMOTE CCP graphs



(a) nodes and tree depth vs alpha

(b) test/train accuracy by alpha

Figure 11: RandomOverSampler CCP graphs



(a) nodes and tree depth vs alpha

(b) test/train accuracy by alpha

Figure 12: EstimatedSalary dropped CCP graphs

References

[1] Talal Alshammari. Using artificial neural networks with gridsearchcv for predicting indoor temperature in a smart home. *Engineering, Technology & Applied Science Research*, 14(2):13437–13443, 2024.

[2] BarelyDedicated. Bank customer churn modeling. https://www.kaggle.com/datasets/barelydedicated/bank-customer-churn-modeling, 2021. Accessed: 2024-11-29.

[3] Tirimula Rao Benala and Karunya Tantati. Efficiency of oversampling methods for enhancing software defect prediction by using imbalanced data. *Innovations in Systems and Software Engineering*, 19(3):247–263, 2023.

[4] Rok Blagus and Lara Lusa. Smote for high-dimensional class-imbalanced data. *BMC bioinformatics*, 14:1–16, 2013.

[5] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008.

[6] Oktay Günlük, Jayant Kalagnanam, Minhan Li, Matt Menickelly, and Katya Scheinberg. Optimal decision trees for categorical data via integer programming. *Journal of global optimization*, 81:233–260, 2021.

[7] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou. On the class imbalance problem. In *2008 Fourth international conference on natural computation*, volume 4, pages 192–201. IEEE, 2008.

[8] Christopher W Hart, James L Heskett, and W Earl Sasser Jr. The profitable art of service recovery. *Harvard business review*, 68(4):148–156, 1990.

[9] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[10] OpenAI. Chatgpt: Generative pre-trained transformer. https://openai.com/chatgpt, 2024. Accessed: 2024-11-28.

[11] Kiran Bangalore Ravi and Jean Serra. Cost-complexity pruning of random forests. *arXiv preprint arXiv:1703.05430*, 2017.

[12] Jerome Friedman Trevor Hastie, Robert Tibshirani. *The elements of statistical learning: data mining, inference, and prediction.* Springer, 2009.

[13] Tao Yan, Shui-Long Shen, Annan Zhou, and Xiangsheng Chen. Prediction of geological characteristics from shield operational parameters by integrating grid search and k-fold cross validation into stacking classification algorithm. *Journal of Rock Mechanics and Geotechnical Engineering*, 14(4):1292–1303, 2022.

Generative AI Statement

AI-supported/AI-integrated use is permitted in this assessment. I acknowledge the following uses of GenAI tools in this assessment:

- (NO) I have used GenAI tools for developing ideas.
- (NO) I have used GenAI tools to assist with research or gathering information.
- (NO) I have used GenAI tools to help me understand key theories and concepts.
- (NO) I have used GenAI tools to identify trends and themes as part of my data analysis.
- (YES) I have used GenAI tools to suggest a plan or structure for my assessment.
- (NO) I have used GenAI tools to give me feedback on a draft.
- (NO) I have used GenAI tool to generate image, figures or diagrams.
- (NO) I have used GenAI tools to proofread and correct grammar or spelling errors.
- (NO) I have used GenAI tools to generate citations or references.
- (YES) Other: [please specify]: I have used GenAI tools to debug code.

I declare that I have referenced use of GenAI outputs within my assessment in line with the University referencing guidelines.