Bank customer churn prediction with Logistic Regression and Decision Tree

Description and motivation of the problem

- Building two supervised learning models (Logistic Regression and Decision Tree) for a classification question (predicting customer churn prediction).
- Comparing and analysing the performance of those models from predicting bank customer churn (exited = 1 vs exited = 0).
- Comparing the results to previous research using the same dataset by Manas Rahman in 2020.

Initial analysis of the data set including basic statistics

- The dataset Bank Customer Churn Prediction from Kaggle Datasets.
- The original dataset consists of 10000 rows and 14 columns, 13 of which are features and 1 is the label (binary, with 1 = exited and 0 = not exited). This dataset does not have any missing values.
- Table 1 shows the descriptive statistics of this dataset. Irrelevant features (i.e. 'RowNumber', 'CustomerId', 'Surname') have been removed and categorical features ('Geography', 'Gender') have been coded using one-hot encoding.
- Churned records account for 20% and unchurned records account for 80% (Figure 1). We have an imbalanced class problem in this dataset.
- The correlation heatmap (figure 2) shows the correlation between features and the label.
- The box plots (Figure 3) show the distribution of each feature. Churned customers tend to have lower credit scores, more balance, and are more likely to be in Germany.

Figure 1

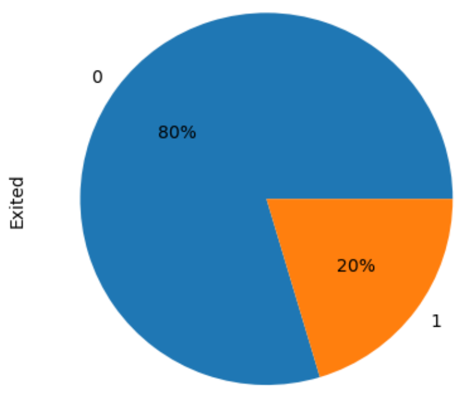


Table 1

	count	mean	std	min	25%	50%	75%	max
CreditScore	10000	650.5	96.7	350	584	652	718	850
Age	10000	38.9	10.5	18	32	37	44	92
Tenure	10000	5	2.9	0	3	5	7	10
Balance	10000	76485.9	62397.4	0	0	97198.5	127644.2	250898.1
NumOfProducts	10000	1.5	0.6	1	1	1	2	4
HasCrCard	10000	0.7	0.5	0	0	1	1	1
IsActiveMember	10000	0.5	0.5	0	0	1	1	1
EstimatedSalary	10000	100090.2	57510.5	11.6	51002.1	100193.9	149388.2	199992.5
is_Germany	10000	0.3	0.4	0	0	0	1	1
is_Spain	10000	0.2	0.4	0	0	0	0	1
is_male	10000	0.5	0.5	0	0	1	1	1
Exited	10000	0.2	0.4	0	0	0	0	1

Figure 2

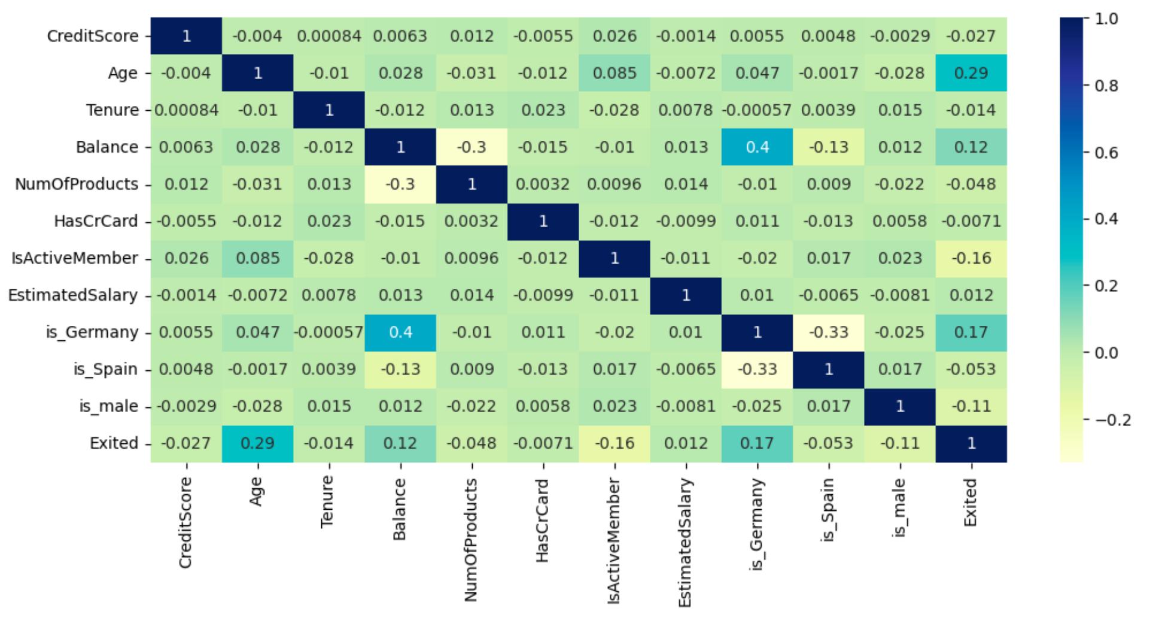
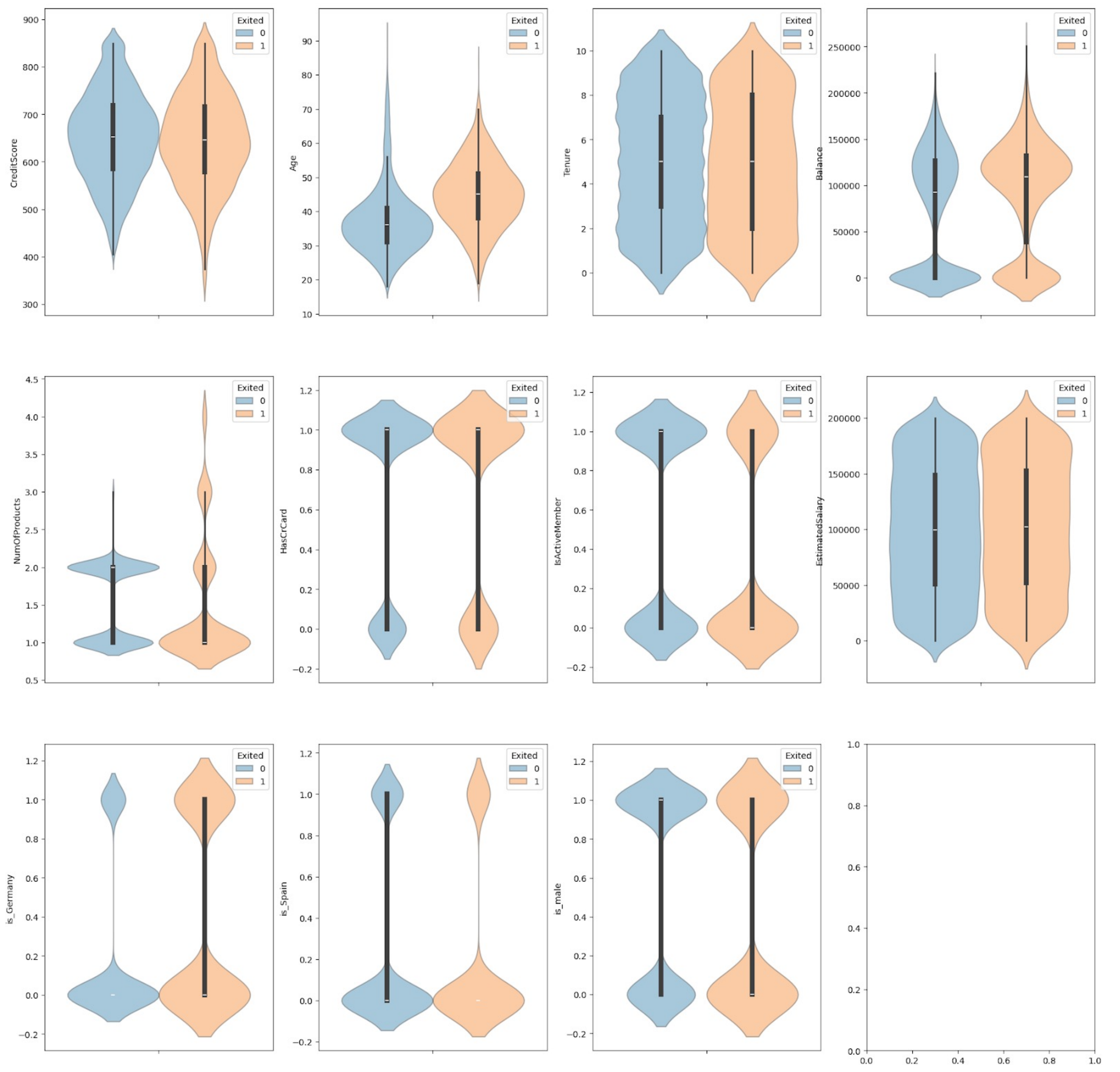


Figure 3



Brief summary of the two ML methods with their pros and cons

Logistic regression

- Logistic regression is a simple and efficient supervised learning algorithm for binary classification problems.
- Logistic regression maps the dependent variable as a sigmoid function of independent variables, and the function only returns values between 0 and 1 representing the probability.
- Logistic regression models are sometimes prone to overfitting. Regularization is the technique that can help reduce this issue [1].
- Pros:
  - Simplicity: Logistic regression models are simpler than other machine learning algorithms.
  - Speed: Due to their mathematical simplicity, logistic regression models require less computational capacity. It is thus an 'agile' method to start with and serve as a baseline model.
  - Interpretability: Logistic regression models are relatively easy to interpret.
- cons
  - Logistic regression models are usually less accurate than more complex algorithms.
  - Logistic regression models are prone to overfitting when there are too many features.
  - Logistic regression models are less accurate if the number of records is too small.

Decision tree

- Decision tree is another supervised learning algorithm but for both classification and regression problems.
- A decision tree is a tree structure where each node represents a feature, a branch represents a rule for splitting and a leaf node represents the output [2].
- Pros:
  - Decision tree models are easy to follow and understand as they have a flowchart-like structure and adhere to humans' decision-making processes.
  - Decision tree models are easy to interpret and visualise.
  - This algorithm can handle nonlinearity.
  - This algorithm doesn't require a rigorous data-cleaning process.
- Cons:
  - Decision tree models can create over-complex trees that lead to overfitting problems.
  - This algorithm can create a biased result if some classes dominate [3].
  - Decision trees are less stable. Small variations in the dataset might lead to a totally different model [3].

Hypothesis statement

- Both models will perform significantly better than random guessing in customer churn prediction.
- DT performs slightly better than LR.
- LR will have a shorter training time.
- The DT model can generate accuracy similar to that of the previous study using the same dataset [4].

Methodology: Description of the choice of training and evaluation methodology

1. Randomly split the preprocessed dataset into a training set (70%) and a test set (30%). Then, split the features (X) and the label (y).
2. Fit LR and DT models using the default hyperparameter as the baseline models for each algorithm.
3. Perform hyperparameter tuning on LR and DT models.
4. Fit the final models with the optimised hyperparameters.
5. Test LR and DT model on unseen data and check their generalisation using metrics.
6. Models with and without oversampling techniques, feature engineering and feature selection will be trained, evaluated and compared.
7. Metrics for model performance evaluation are accuracy, precision, recall and F1 score.

Choice of parameters and experimental results

Logistic regression:

- The baseline model generated good enough accuracy but low precision, recall and F1 score.
- Hyperparameters of 'Solver', 'Regularization' and 'Lambda' (regularization rate, controls the amount of regularization applied [5]) were tuned to improve model performance. Ridge regularization was applied to reduce overfitting, and 'Solver' was set to 'bfgs'. The optimization process is shown in Figure 4.
- The best model is the one trained on an oversampled training set with feature normalisation.
- The best-estimated lambda is 0.000083017.
- The experimental results can be found in Table 2.
- time for fitting the baseline LR model was 1.3150, time for tuning LR model was 24.4860, and time for fitting the optimised LR model was 0.0533. The confusion matrix is shown in Figure 5.

Decision tree:

- Gini's diversity index was used for the splitting criterion.
- The hyperparameter tuning process adjusted the depth of the tree and leaf size by tuning 'MaxNumSplits' and 'MinLeafSize' (using a grid search approach).
- The better model is the one trained on a non-oversampled training set, with feature selection and without normalisation.
- The final choice of hyperparameters are MaxNumSplits = 50 and MinLeafSize = 20.
- The experimental results are in Table 3, and the confusion matrix (Figure 6).
- Time for fitting baseline DT model was 1.4773, time for tuning DT model was 5.9305, and time for fitting opt DT model was 0.0166.

Figure 4

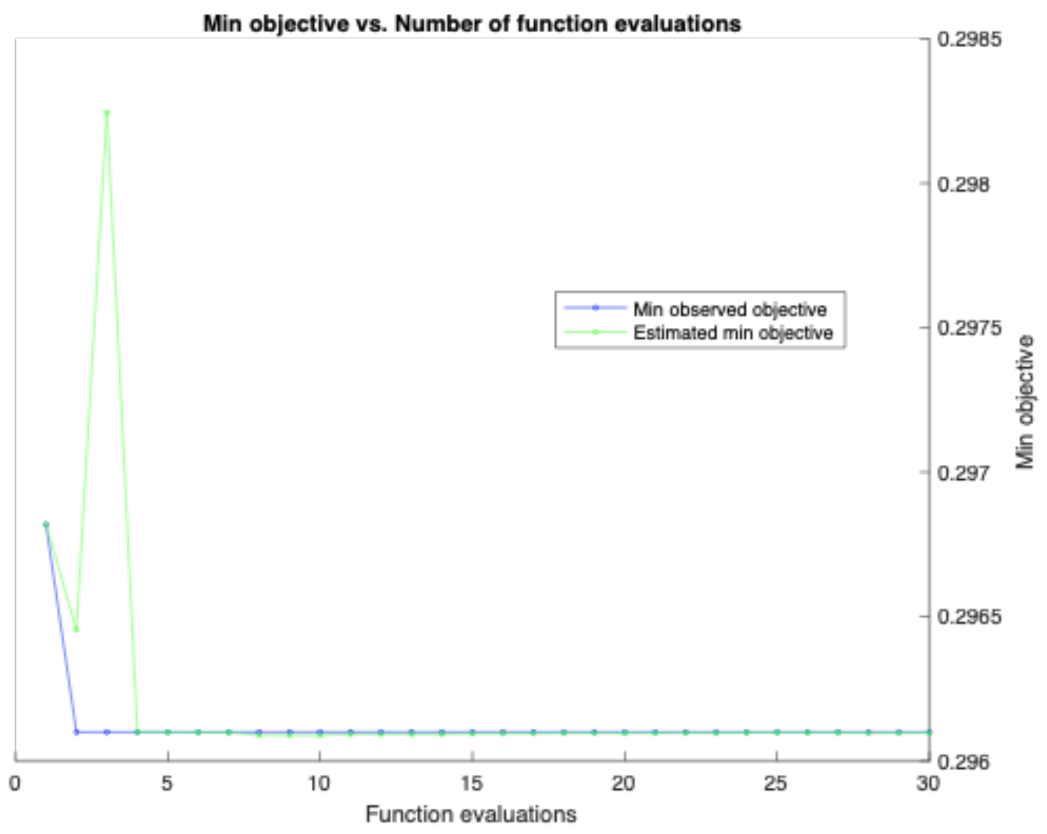


Table 2

LR	With oversampling		
	base	After tuning	test
Acc	0.7047	0.7054	0.6410
Precision	0.7076	0.7077	0.3336
Recall	0.6977	0.6999	0.7937
F1	0.7026	0.7038	0.4697

Table 3

DT: no_os, no_normalisation, selection			
	base	After tuning	test
Acc	0.7997	0.8633	0.8580
Precision	0.5114	0.7513	0.7204
Recall	0.5327	0.4986	0.4759
F1	0.5218	0.5994	0.5731

Figure 5

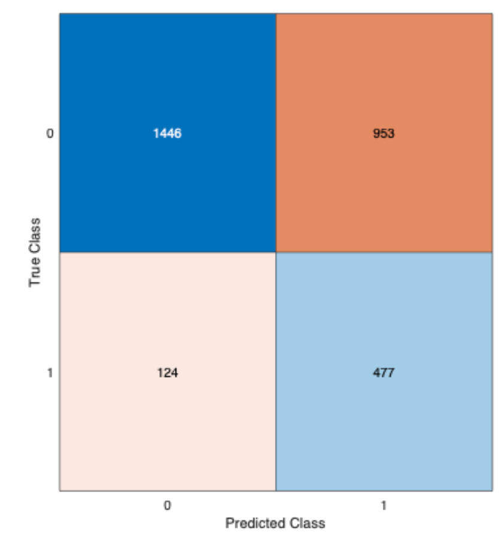
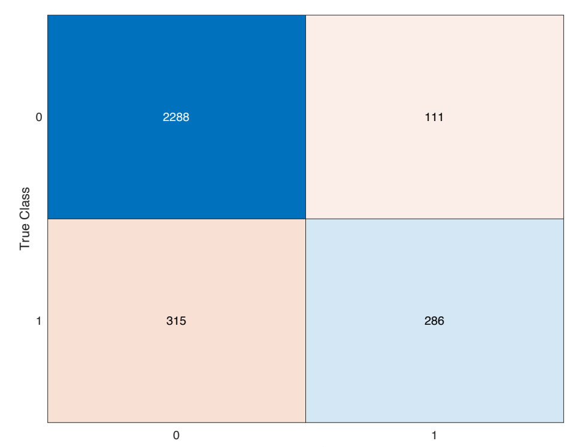


Figure 6



Analysis and critical evaluation of results

- DT generated better results than random guessing, but LR does not. Decision tree models perform better than logistic regression in terms of accuracy, precision, and F1 score. This might be due to the non-linear relationships between features and the label in this dataset and the non-normal distribution of many features (e.g. 'Balance'). DT tends to be superior over LR algorithm in handling non-linearity and non-normal distribution as it is a non-parametric estimator. Another reason could be due to zero values in the dataset (e.g. many zero values in 'Balance' column). DT model splits the data based on the feature values so it can handle sparse data with many missing or zero values. LR, however, assumes the features do not have missing or zero values.
- Compared with the baseline models, DT has a more significant improvement after hyperparameter tuning (around 6% in terms of accuracy).
- For LR, the argument 'OptimizeHyperparameters' was set to 'lambda', so the fitclinear function will optimise the hyperparameter 'lambda'. The regularisation and the solver were manually set to 'ridge' and 'bfgs' for the purpose of optimising accuracy [6]. However, the optimised LR model doesn't significantly improve accuracy. This is because regularization is not for improving algorithm performance. It is a penalty against the complexity, so the model doesn't pick up too much 'noise' from training set but has a good generalisation and achieves good accuracy on unseen data [7]. From the results (table 2), we can see LR model achieved a good enough generalisation.
- For DT, the hyperparameters 'MaxNumSplits' and 'MinLeafSize' were tuned during training process. The former specifies the maximum number of branch nodes, and the latter sets the minimum number of observations that should be present in each node of the tree [6]. A small value of MinLeafSize might result in a high variance that contributes to a high performance of the training set but compromises the performance of unseen data and also requires a longer training time. When MaxNumSplits = 50 and MinLeafSize = 20, our DT model achieves good accuracy on both validation and training sets. Precision also significantly improved after tuning, but this improvement did not show on recall and F1 score. The lower recall and F1 score might be due to the imbalanced labels.
- The time for fitting baseline LR and DT were similar, but the time for tuning and fitting the optimised DT model was significantly shorter than LR. Considering the higher performance of DT, DT is the more effective and efficient choice for this classification problem.
- The accuracy score of our DT model (85.8%) is better than the previous study using the same dataset (78.99%). In a previous study, the authors found DT achieved a significant improvement in accuracy score after oversampling the minority class [4]. This improvement was not replicated in this coursework. When using oversampling, our DT model using oversampling achieved an improvement in the training set but did not have a good generalisation on unseen data. One of the reasons could be the previous study conducted oversampling before splitting training set and test set, so there could be some data leakage issues.

Lessons learned and future work

- When there are imbalance problems in dataset, accuracy is not enough to evaluate the performance of a model. Imbalance. We should also look at metrics such as precision, recall and F1 score.
- We should choose learning algorithms not only based on research purpose and labels (continuous, multiclass, or binary) but also on the distribution of features and the relationship between features and the label.
- Future work:
- Try to use other learning algorithms that can better handle non-linearity or are more robust when there are outliers or other noise in dataset.
- Explore more preprocess techniques and evaluate how this contributes to model improvement, such as generating synthetic samples for oversampling.
- Try to tune other hyperparameters and see if this helps with improving performance, especially precision and recall.

References

[1] F. Salehi, E. Abbasi, and B. Hassibi, 'The Impact of Regularization on High-dimensional Logistic Regression', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019. Accessed: Dec. 18, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/hash/ab49ef78e2877bfd2c2bfa738e459bf0-Abstract.html

[2] S. Suthaharan, 'Decision Tree Learning', in *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, S. Suthaharan, Ed., in Integrated Series in Information Systems. , Boston, MA: Springer US, 2016, pp. 237–269. doi: 10.1007/978-1-4899-7641-3_10.

[3] '1.10. Decision Trees', scikit-learn. Accessed: Dec. 04, 2023. [Online]. Available: https://scikit-learn/stable/modules/tree.html

[4] M. Rahman and V. Kumar, 'Machine Learning Based Customer Churn Prediction In Banking', in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India: IEEE, Nov. 2020, pp. 1196–1201. doi: 10.1109/ICECA49313.2020.9297529.

[5] 'Generalized Linear Model (GLM) — H2O 3.44.0.2 documentation'. Accessed: Dec. 05, 2023. [Online]. Available: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glm.html#regularization

[6] 'Fit binary decision tree for multiclass classification - MATLAB fitctree - MathWorks United Kingdom'. Accessed: Dec. 08, 2023. [Online]. Available: https://uk.mathworks.com/help/stats/fitctree.html#namevaluepairs

[7] S. Raschka, 'Does regularization in logistic regression always results in better fit and better generalization?', Sebastian Raschka, PhD. Accessed: Dec. 08, 2023. [Online]. Available: https://sebastianraschka.com/faq/docs/regularized-logistic-regression-performance.html