Telco Customer Churn Prediction
A comparison study of Logistic Regression and Random Forest

INM431. Machine Learning Coursework - ***Elnara Mammadova*** *(210026788)*

CITY UNIVERSITY OF LONDON
EST 1894

1. DESCRIPTION AND MOTIVATION

Customer Churn Prediction is a binary classification problem, where the objective is to predict those customers who intend to leave the service provider ahead of time. Customers retaining is the most important asset for any business as it is stated that "gaining a new customer is more costly than retaining an existing one" [1], and "by retaining a mere 5% more customers, e-companies can boost profits by 25% to 95%" [2]. Predictive models can provide correct identification of possible churners in the near future, so as to give companies an opportunity to provide appropriate retention solutions.

In this poster, we will summarize a comparative study on rate of churning of customers using Logistic Regression and Random Forest algorithms and evaluate the performance of the methods through their prediction accuracy.

3. HYPOTHESIS TESTING

Churn prediction is a topic of great complexity and is prone to produce some interesting results. Based on the previous studies [4][5][6] both models are expected to produce comparative results, with accuracies around 0.80, with Random Forest producing slightly better accuracies. Since our data is imbalanced, a weighted random forest could possibly generate better results [7][8].

Random Forest(bootstrap aggregation) is expected to perform much better on unbalanced training data, since we won't be using `RUSBoost` (multiclass classification for skewed or imbalanced data), while Logistic regression might require training on balanced data set (SMOTE).

2. INITIAL ANALYSIS OF DATA SET AND BASIC STATISTICS

2.1. Context

The Telco Churn Dataset was acquired from IBM Watson Accelerator Catalogue [3]. The dataset contains 7043 samples with 10 continuous and 22 discrete features. The target response variable ("Churn") contains two classes and represents imbalance between classes (No – 5174 vs, Yes-1869). The dataset is without missing or non-applicable values, however, there were outliers within continuous variables, that was addressed during initial data preparation.

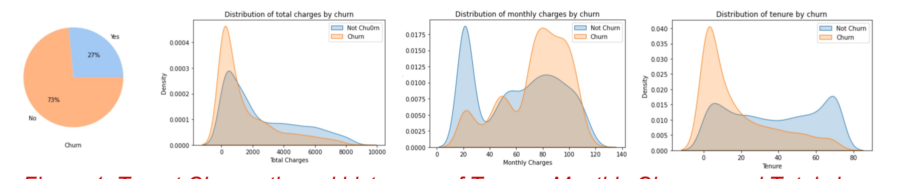

Figure 1. Target Class ratio and histogram of Tenure, Monthly Charges and Total charges.

2.2. Data Exploration

Basic statistical analysis and visualization techniques were applied for deeper data understanding. Some features showed strong indicators of churned customers while other were not indicative of churn.

- Customers who are more likely to churn are : senior citizens, customers with no partner or dependents, as well as customers with either Offer-E, Fibre Optic internet service, Unlimited data, Paperless billing and/or short term contracts
- Attributes such as gender and streaming services are not indicative of churn
- Customers who have churned have high monthly charges and are generally new customers.

2.3. Data Manipulation

After checking for null values, and removing outliers, several feature engineering methods were applied to the original dataset before modelling (despite the fact that Random Forest did not require feature scaling).

- Dummy features from categorical attributes with more than 2 unique values
- Encoded all the categorical features (0, 1 values)
- Applied standardization (scaling) to all numeric features ([0 1] intervals)

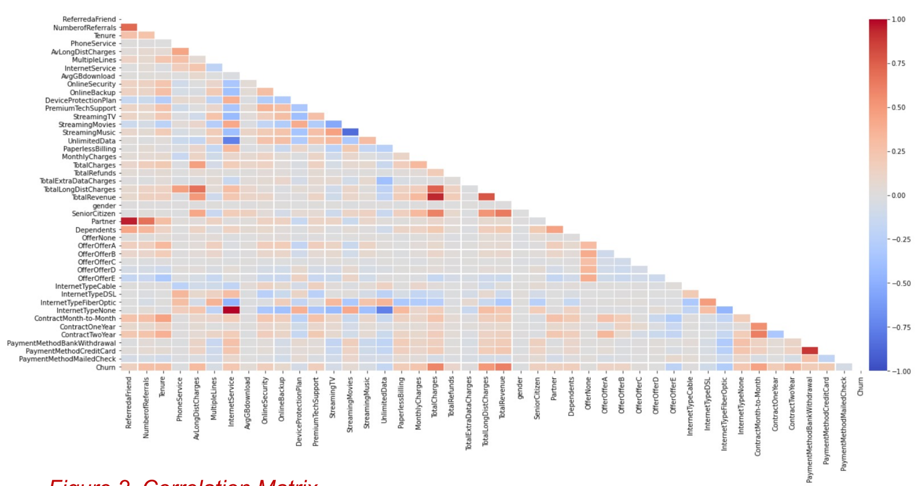

Figure 2. Correlation Matrix.

4. TRAINING AND EVALUATION METHODOLOGY

4.1. Training Methodology

- Subsequent to feature engineering, the transformed dataset was split randomly in 80:20 ratio of training-test data.
- Apply SMOTE (Synthetic Minority Oversampling Technique) on training set alone for Logistic Regression to fix for the class imbalance (new target class ratio 50/50). For Random Forest the training set was left unbalanced.
- Use 10-fold cross-validation during Bootstrap aggregation hyperparameter selection and lasso regularization.

4.2. Evaluation Methodology

- Getting the best performing lambda for logistic regression, and tree depth, min leaf size, maximum number of splits for random forest on the cross-validated training data and evaluate the performance on the test set.
- Getting accuracy, precision, recall, F!-Score, AUC and confusion matrix on test data for both models to evaluate overall classification performance. For additional evaluation of unbalanced classification, F2-Score, Precision-Recall AUC and G-Mean were calculated

5. COMPARISON OF MACHINE LEARNING ALGORITHMS

5.1. Logistic Regression

Logistic regression is a generalized linear regression analysis model. It is used to regress categorical and numeric variables into a binary outcome variables. It explains the relationship between one dependent variable and one or more independent variables by estimating probabilities using a cumulative distribution function of logistic distribution. The log odds of the target variable (logit) is being predicted. The regression coefficients are estimated using maximum likelihood estimation.

5.1.1. PROS

- Logistic regression is easier to implement, interpret, and is computationally efficient.
- The predicted parameters (trained weights) give measure of importance of each feature and are highly interpretable.
- It is a linear model and less prone to overfitting in low dimensional dataset with adequate training examples.
- It provides model flexibility in high dimensional datasets where the overfitting may be easily controlled using regularization techniques (L1, L2)

5.1.2. CONS

- It relies on the assumption of linearity between the dependent variable and the independent variables (which is highly unlikely in real world)
- It is more likely to experience problems capturing complex relationships between variables.
- It requires higher number of observations than that of features, otherwise it may result in over-fitting.
- It can only be used to predict discrete functions.

5.2. Random Forest

Random Forest regression generates an ensemble of decision trees, which uses Bootstrapped aggregation (bagging, which was also applied in this study) reduces the effects of overfitting and leads to "improvements for unstable procedures" including classification trees [9]. Each tree uses a subset of existing features selected at random for each decision split and gets trained with a bootstrap replicate of the original training set.

5.2.1. PROS

- It can handle large dataset with higher dimensionality and does not require feature scaling.
- It provides more optimization options where one can tweak hyperparameters and improve their machine learning model.
- It is less likely to overfit and provides better accuracy (less variance) than other classification algorithms.
- Random Forest regression is flexible in modelling a multitude of underlying predictor-response mapping functions.

5.2.2. CONS

- Random Forest model can easily get very large and are slow to build (computationally intensive).
- It is not suitable for linear methods with a lot of sparse features.
- Predictions are not easily interpretable, especially when deep decision trees are preferred in order to lower bias and variance.
- Constructed to minimize the overall error, so in highly imbalanced datasets, it tends to focus more on the prediction accuracy of the majority class, resulting in poor classification results for the minority class [8].

6. CHOICE OF PARAMETERS AND EXPERIMENTAL RESULTS

6.1. Logistic Regression *(implemented with **MATLAB** `lassoglm` & `fitglm` functions)*

6.1.1. Parameter selection

- A cross-validated lasso (L1 penalty) regularization of a generalized linear model [10] (lassoglm) using 25 Lambda (number of regularization coefficients) and 10-fold cross-validation was run on the balanced training data. Model coefficients corresponding to the Lambda with minimum expected deviance and nonzero model coefficients was chosen from cross-validated deviance of lasso fit. Sequence of models were used to identify and discard redundant (highly correlated) predictors.
- Trained the model on balanced training data. Fit the logistic binomial model (fitglm) using coefficients from the minimum-plus-one standard error point found in the previous step.
- The threshold was applied to the cut-off point in probability between the positive and negative classes, which by default for any classifier would be set at 0.5. However, setting the threshold to >=0.4, biased the predictive behaviour of a classification model. That way the model produced imbalanced binary prediction that is not biased to the majority class. *Table 1* shows the comparison of performance measures for Logistics regression model with thresholds set to >=0.5 vs. >=0.4.

6.1.2. Results

- From (*figure 1*) the minimum deviance is around 5000.
- Our residuals are slightly right skewed but otherwise pretty symmetrically distributed, tending to cluster towards the middle of the plot. Residuals from Lasso regularized model is the same concentration in the middle as logistic regression, but significantly higher (*figure 2*).
- Lasso Generalization gets 41 coefficients to optimize the model, of which DeviceProtectionPlan, StreamingTV and OfferA are the most valuable factors.

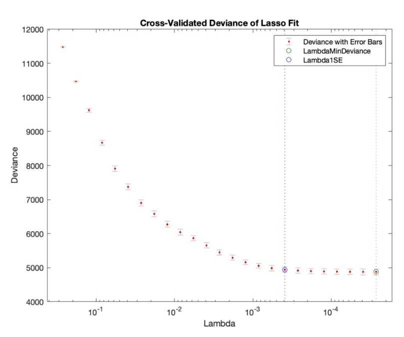

Figure 3. Logistic Regression Cross-Validates Lasso Optimizer using 25 Lambda

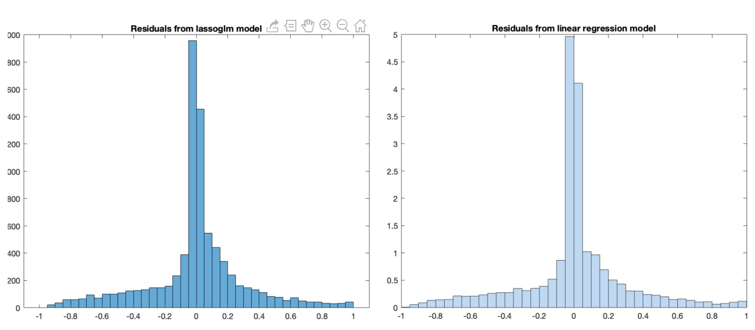

Figure 4. Residuals from models.

	LR=0.4 threshold	LR=0.5 threshold
Accuracy	0.86374	0.87014
Recall	0.91785	0.88137
Precision	0.82821	0.862
f1-score	0.87073	0.87158
f2-score	0.89841	0.87743

Table 1. Comparison of performance measures between probability thresholds.

6.2. Random Forest *(implemented with **MATLAB** `fitcensemble` and `fitglm` functions)*

6.2.1. Parameter selection

- The selected Random Forest model is a Bootstrap aggregation method ('Method' is 'Bag') with decision tree template, which predict by averaging all individual trees' decisions using weak learners.
- Unlike Logistic Regression the model was run on unbalanced data, using Prior probabilities (26.5, 73.5) and custom misclassification cost [0 3; 1 0]. Assigning prior helped handle the imbalanced data problem with cost-sensitive approach, so that it can reduce misclassified data [11].
- The optimized Random Forest model is trained by hyperparameters which were selected as result of a custom nested for-loop grid search analysis. Best grid-search results were found at the following hyperparameters without overfitting: minimum leaf size of 4, and 150 maximum splits.
- Final hyperparameter optimization was done on Number of Learning Cycles by generating 10-fold cross-validated ensemble on training data and by examining the cross-validation classification (0.188) as a function of the number of trees in the ensemble. The optimization produced Optimal number of trees being at 68.

6.2.2. Experimental Results

- Based on the home-made grid search, it was obvious that Maximum Number of Splits and Minimum Leaf Size are the main controllers of the model's total-error.
- The model accuracy results in a plateau with increasing Minimum Leaf Size after 150 Maximum Number of Splits.
- Test and Cross-validation classification loss produce comparable results.
- The variance of the results are very high for different leaf sizes

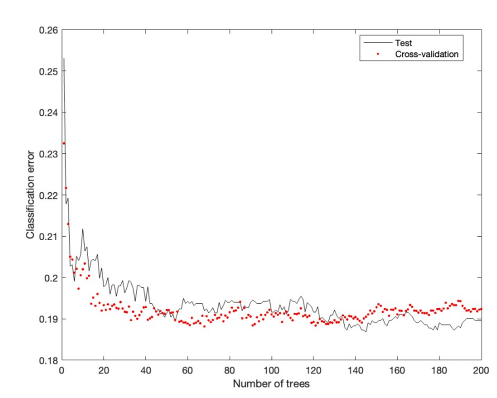

Figure 5. Classification loss, cross-validation vs. test

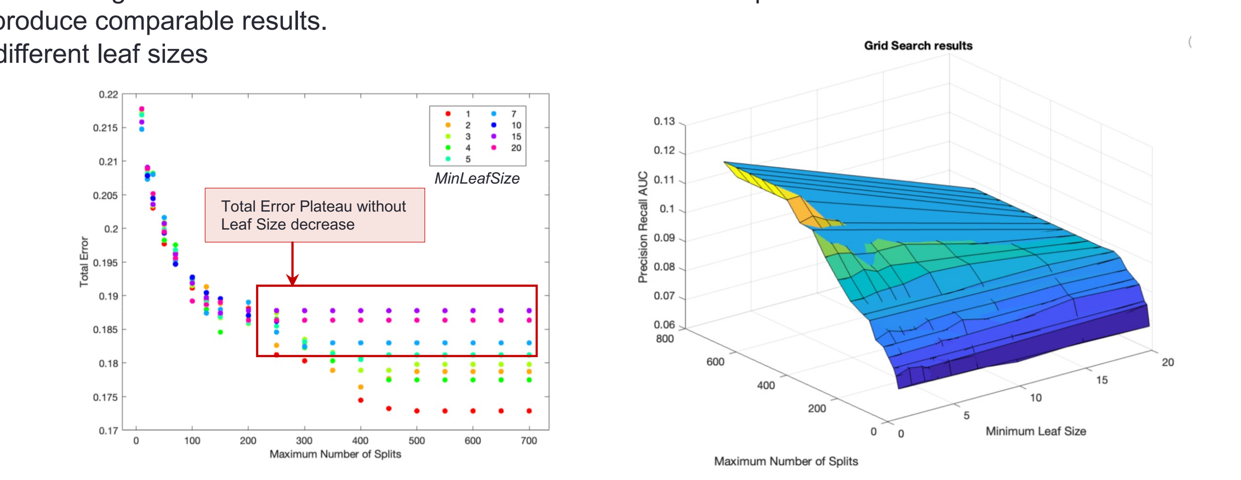

Figure 6. Home-made Grid Search results w.r.t. Total Error and Precision Recall AUC

7. ANALYSIS AND CRITICAL EVALUATION RESULTS

	Logistic Regression	Random Forest
Accuracy	0.80767	0.8105
Sensitivity/Recall	0.81283	0.80214
Specificity/TNR	0.8058	0.81353
Precision	0.60198	0.60852
f1-score	0.6917	0.69204
f2-score	0.75962	0.69204
G-Mean	0.80931	0.80781
AUC	80.932	80.783
Precision-Recall AUC	74.372	73.964

Table 2. Performance metrics for both models (testing)

LR_classloss = 0.1923
R_classloss = 0.1895

- Consistent with the literature and supporting our hypothesis, the experiments demonstrated that both models achieve similar results: Random Forest with slightly higher accuracy, and Logistic regression exhibiting slightly higher performance when it comes to more important metrics.
- Since the dataset is imbalanced, Recall and F2-Score are more important threshold metrics. Getting lower false negatives (predicting churned customer as non-churn) resulted in better accuracy comparison of the models.
- Despite the fact of Random Forest providing higher Accuracy and lower classification loss, Logistic Regression outperforms Random Forest when it comes to Recall, F2-Score, G-mean and Precision Recall AUC.
- Because Random Forest was trained on biased data, when it encounters considerable negative class in the test set, it is prone to predict a greater proportion of false negatives and a lower proportion of false positive instances. This was rewarded by test set negative bias, resulting in greater accuracy but the weaker positive predictive values is penalized by F2-score.
- In order to not get penalized by the F2-score in our Logistic Regression predictions, setting the probability threshold to >=0.4 was rewarded by the greater test set positive class bias (bringing F2-score from 0.72 to 0.76, Recall from 0.746 to 0.812 while reducing Accuracy from 0.83 to 0.808).

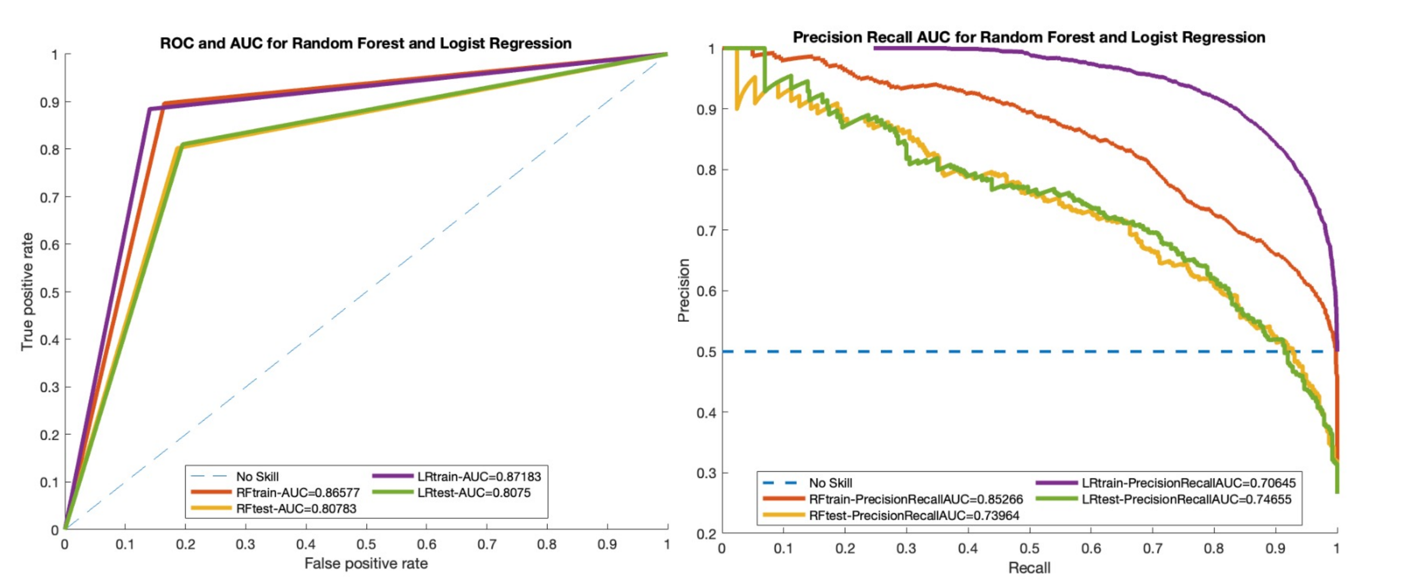

Figure 7. ROC AUC and PR AUC curves

- For ranking metrics, like the ROC AUC, calculating the area under the curve as a score and using that score to compare classifiers is preferred. In this case, the focus on the minority class makes the Precision-Recall AUC more useful for imbalanced classification problems. ROC and PR curves tell a different story, as PR curve focuses on the minority class, whereas the ROC curve covers both classes. Both ROC AUC and PR AUC show very similar behaviour for the models, only difference being the Precision Recall AUC for training data. As seen in the figure Logistic Regression is showing higher AUC for training dataset. Both models performed in similar way in ROC AUC and PR AUC.

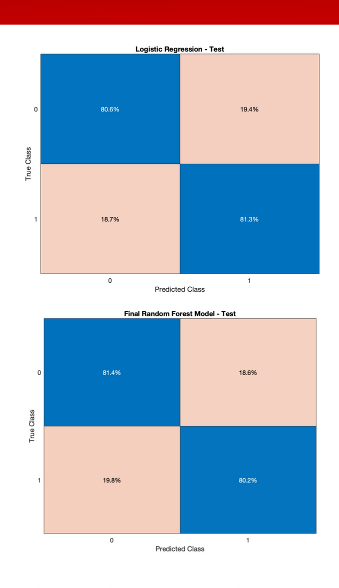

Figure 8. Confusion Matrix

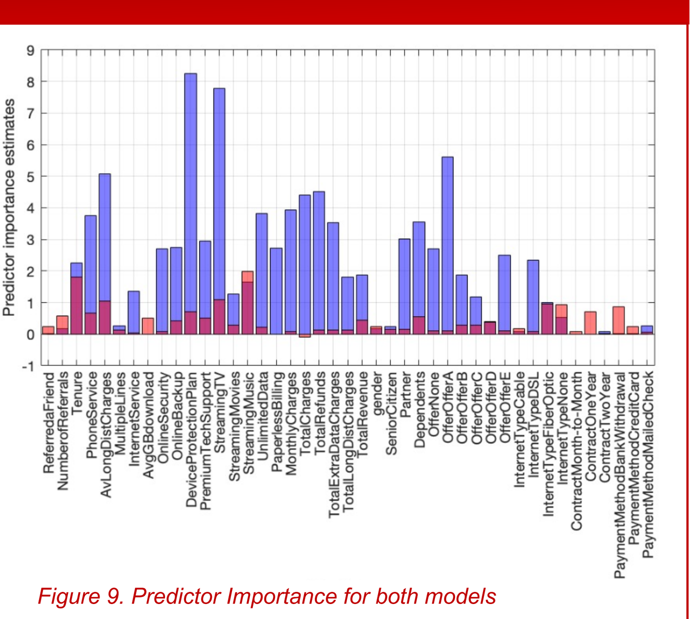

Figure 9. Predictor Importance for both models

- There is a significant difference in predictor importance for the two models used, which deserves a further investigation down the road.
- It was surprising to achieve such similar accuracies in testing for both models, which is believed to be the consequence of how the bias was managed during both models' training.
- Training and testing of Logistic regression was computationally very efficient which makes it more applicable for large dataset.

8. LESSONS LEARNED AND FUTURE WORK

Both Logistic Regression and Random Forest performed well in classifying churn and little hyperparameter tuning and feature engineering was required. I believe that this analysis should also lay a good foundation for the other churn prediction problems, where the proposed methods should be applicable with little or no modifications. Part of the observations during Random forest training was that the accuracies obtained with the balanced datasets were lower than that of the imbalanced datasets (when considering the *F2*-scores, those of the balanced datasets were higher than those of the imbalanced dataset). However, further work should include reducing the dimensions of input features through feature selection, and deeper understanding of predictor importance for two models. It would be interesting to see what will be the effect on both models if we apply the feature selection method by assigning weights to the features. Since our data is imbalanced, it would also be interesting to see how multiclass classification with Random Forest performs with RUSBoost or cost-sensitive learning methods by modifying the model weight of AdaBoost. Additional work could be done for Logistic Regression, where instead of L1 (lasso), construct L2 (ridge) generalized linear model to combat the bias-variance trade-off by assigning weights to variables, such that our model maintains all the variables and at the same time gives more importance to the important ones [12]. In further work, to correct class bias other methods such as ADASYN (instead of SMOTE) can be applied to test if this improves prediction results. Additionally, further work can be conducted on the finding the reasons of the inconsistency of misclassification cost and false positive cost. Another future direction that can be foreseen from the proposed work is the implementation of more comprehensive study with other types of models and compare our results, with further data acquired. Additionally, neural networks and different variations of deep learning models could be assessed for the customer churn problem in the telecommunications industry.

9. REFERENCES

[1] Xu, M. and Walton, J. (2005) "Gaining customer knowledge through analytical CRM," *Industrial management + data systems*, 105(7), pp. 955–971.
[2] Jones, T. O. (1996) "Why satisfied customers defect," *Journal of management in engineering*, 12(6), pp. 11–11
[3] IBM Data. *https://www.ibm.com/communities/analytics/watson-analytics-blog/predictive-insights-in-the-telco-customer-churn-data-set*
[4] Ebrah, K. and Elnasir, S. (2019) "Churn prediction using machine learning and recommendations plans for telecoms," *Journal of computer and communications*, 07(11), pp. 33–53.
[5] Pamina, J. *et al.* (2020) "Inferring machine learning based parameter estimation for telecom churn prediction," in *Computational Vision and Bio-Inspired Computing*. Cham: Springer International Publishing, pp. 257–267.
[6] Essam Shaaban. *et al.* (2012). "A Proposed Churn Prediction Model," *International Journal of Engineering Research,* 02(4), pp.693-697
[7] Effendy, V., Adiwijaya and Baizal, Z. K. A. (2014) "Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest," in *2014 2nd International Conference on Information and Communication Technology (ICoICT)*. IEEE.
[8]Amin, A. *et al.* (2016) "Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study," *IEEE access: practical innovations, open solutions*, 4, pp. 7940–7957.
[9] Breiman, L. (1996) "Bagging predictors," *Machine learning*, 24(2), pp. 123–140.
[10] Pan, X. and Xu, Y. (2021) "A safe feature elimination rule for L1-regularized logistic regression," *IEEE transactions on pattern analysis and machine intelligence*, PP, pp. 1–1.
[11] C. Chen, A. Liaw and L. Breimenn, "Using Random Forest to Learn Imbalanced Data," Statistics Department of University of California, Berkeley, 2004.
[12] Pereira, J. M., Basto, M. and Silva, A. F. da (2016) "The logistic lasso and ridge regression in predicting corporate failure," *Procedia economics and finance*, 39, pp. 634–641.