Automatic Detection of Eligibility for Loan

Job Thomas Thekkekara (R00195427)
Munster
Technological
University, Cork,
Ireland
Job.thekkekara@mycit.ie

Abstract

Institutions like banks and insurance corporations lend money to people as loans. The dataset obtained from [1] is from an insurance company. The company wants to automate the loan eligibility process. Hence the main objective is to build a machine learning model that takes customer information as features and predicts if the customer has the eligibility to get the loan or not. The dataset went through several preprocessing steps. Imputation of missing values included multiple imputations by chained equations (MICE). Square root transformation in numerical variables to avoid skewness was also performed. Both label encoding and one hot encoding was used to encode categorical variables. Scaling of numerical variables was done using minmaxscaler. As 68% of the data belonged to one target class SMOTE technique was used to take care of the dataset imbalance. Logistic Regression, K Nearest Neighbor, Decision Tree, Support Vector Machine and Random Forest classifiers were initially considered for building the model. Out of these Random Forest classifiers gave the best training accuracy. It was further optimized using hyper parameter tuning. Finally, the model gave an accuracy of 75.6% with test data. It also gave a precision of 69%, recall of 71% and f1-score of 70% on test data.

1 Introduction

Data has the potential to transform business and data driven decision making is gaining more importance than ever in today's world. With the advancements in

computing capabilities, data analytics and machine learning are used across all industries to improve businesses. Financial industry is at the forefront of this starting from fraud detection to risk management. One such important application is credit risk scoring. Credit risk scoring is used by financial institutions to ease the decision-making process of accepting or rejecting a loan application. There are several advantages for this which include: faster decisions, less human intervention and bias, reduced cost and so on [2]. A credit risk scoring model is developed by checking the historical personal and financial data of customers. One of the main reasons for people to take loans is because they want to make an expensive purchase that they cannot afford. It is vital for the institutions that lend money to check the background of a customer before lending money. If the money is not given to the right customer who can pay it back, it may lead to further consequences that the institution has to bear. Another advantage of customer segmentation using data is that the institutions can find potential good customers who can take loans and specifically target them by providing customized advertisements.

The dataset obtained consists of 13 features and 614 data points. The features include Loan Id: A unique loan number assigned to each customer. Gender, Marital Status, Number of dependents, Education level, Employment type, property area: A customer's basic demographic information. Applicant Income, Co-applicant Income: Income of applicants and co-applicants. Loan Amount: The amount applied for loan. Credit History: Past history of credits by applicants. Loan Status: Whether the loan paid off or not. Here the target column loan status is clearly labeled and available. Hence it is a supervised learning problem. Also, since the objective is to classify future data points into Loan status 'yes' or 'No', it becomes a classification problem. The rest of the features are used for

training the model. Since the target has only two classes this becomes a binary classification problem.

2 Related Work

There is a number of research work done in this area. Loan approval prediction with basic machine learning algorithms was done by [3]. In 2016 loan prediction using ensemble modeling was done by [4]. This study mainly explores the Ada Boost algorithm and bagging ensemble to contrast with several other classifiers. In 2020 the same dataset was used by [5] to build loan approval prediction models by using algorithms such as logistic regression, KNN etc. Also in 2021 a similar study was conducted by [6]

3 Research

There are several steps in preprocessing data before building a machine learning model. Data imbalance is one of the major issues faced by a machine learning model. That is, the target class has substantial difference in the number of data points in each category[7]. In this dataset, 68% of the data belonged to the target class: Loan Status 'Yes'. Majority of the people available in the dataset were eligible for a loan. Only the remaining 32% belonged to the minority class 'No'.

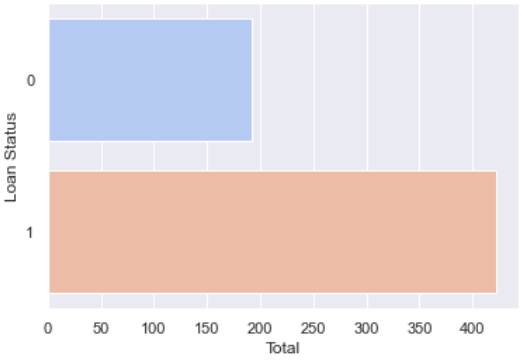

Fig1: Imbalance in target class

This indicates a clear skewness or bias in the data. Imbalanced datasets pose a challenge to the machine learning algorithms as most of them were designed considering equal class distribution [8]. These imbalances can be caused due to several reasons. It could occur due to sampling bias. If the data was collected from a particular location or in a particular time without accounting for the generalization imbalance might occur. It could also occur due to error in data collection. Another major reason is the problem domain itself. In certain scenarios such as credit card fault detection, there would always be very few people who defaulted compared to the ones who did not. Or in disease prediction, there would always be a greater number of people who did not have the disease. In such situations class imbalance is unavoidable.

There are two types of class imbalance. Slight imbalance and severe imbalance. If the distribution of data points is uneven by a small amount, then there is a slight imbalance. But if the distribution is uneven on a large scale, then it is a severe imbalance. In this problem the class imbalance is not severe as there are sufficient data points in the minority class. The class in an imbalanced dataset having more examples is called a majority class and the class in an imbalanced dataset having less examples is called a minority class. In this scenario more concentration has to be given to the minority class. It will be harder for a model to make predictions on minority class as there are fewer data points available for the model to learn. Model will not have sufficient data points to learn the characteristics of the minority class. But as there are more data points on the majority class the model can learn it well and can make accurate predictions on it. Thus, the accuracy of the model will not be affected as most of the data points belong to the majority class and the model predicts them correctly. This becomes more devastating when the interest is on the prediction of the minority class. Example- in credit card fault detection the interest lies in predicting the customers who can default. Since the number of defaulters will be less in training data, the model cannot do accurate predictions on this. Meanwhile the overall accuracy of the model will still be high as it can predict the people who do not default. Hence accuracy will not be a right metric here to evaluate the model. Other metrics such as precision, recall and f1-score have to be considered here.

There are several methods available to overcome the problems of class imbalance. Resampling techniques are widely used for this. It includes undersampling and oversampling techniques. In random under sampling some observations are randomly

removed from the majority class. Sometimes a subset of the majority class is also taken in under sampling technique. The samples are picked with or without replacement. It has the advantage that the run time of algorithms gets improved as the number of data points decreases. But it has several disadvantages too. One major disadvantage is the information loss. As random samples are removed from the dataset important data points might get discarded. Also, the sample remaining after under sampling might be biased and can cause inaccuracies. In random over sampling more copies of data points are randomly added to the minority class with or without replacement. It usually outperforms under sampling as there is no information loss. But oversampling can sometimes lead to overfitting of the model as the generalization of test data is lost. Synthetic Minority Oversampling Technique (SMOTE) is another widely used method. This method creates synthetic points between two samples of minority class, and it overcomes the overfitting problem caused by random over sampling [9] [10]. SMOTE algorithm works in the below steps:

- A minority class data point is chosen as an input vector
- K nearest neighbors is found for the input vector (default 5)
- One of the distance metrics such as Euclidean distance is used to find the distance between input vector and the randomly selected neighbor
- The distance between is multiplied by a random value (0,1] and the result is added to the input vector to get new vector
- Thus, synthetically new data points are interpolated from existing ones.

This approach is very effective as the newly created synthetic samples are relatively close to the existing samples of minority class. Also, this procedure can be used to produce as many samples as possible for the minority class. One general downside for this method is that the minority samples are generated without considering the majority class and can lead to the creation of ambiguous data points if there is a strong overlap between majority and minority classes. It is to be noted that sampling technique is applied only to the training data and usually testing, and validation datasets are kept as it is so that it remains unbiased. Tomek Links is another method of under sampling that removes majority classes in places where there is significant overlap between classes. Removing observations with overlap increases the distance between the classes which further aid the classification process. Often Tomek Links technique is used with SMOTE in case of class imbalance. There are also algorithms that increase the cost of classification mistakes on minority classes. Such algorithms are called penalized learning algorithms. Penalized-SVM is an important algorithm of this nature. Finally, the most important thing is to change the performance metric in case of class imbalance. Confusion matrix, precision, recall, f1-score, area under ROC curve can all be considered as metrics instead of accuracy. Fig 2 shows the results of training data after applying SMOTE technique.

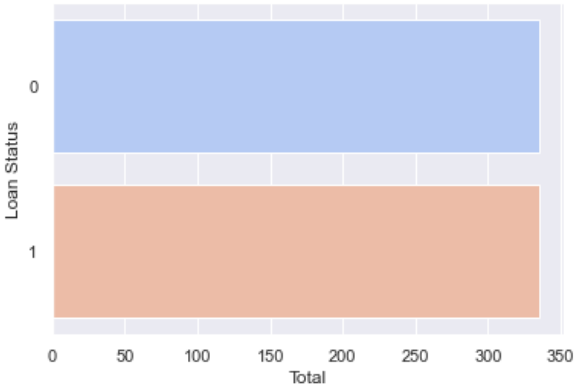

Fig2: Training data after applying SMOTE

3 Methodology

First step was to find the missing values present in the dataset and the decision of the right imputation technique. The column loan_id does not provide any valuable information to the analysis and model building. Hence it was dropped. On further analyzing data, the presence of missing values was found in many features. The target variable did not have any missing values. Imputation of missing values in each categorical variable was done by using the mode of the respective variable. For numerical variables, instead of using traditional imputation techniques such as mean and median, MICE (multiple

imputations by chained technique) was used. Traditional imputation techniques can cause data bias in-case of a high number of missing values. To overcome this drawback MICE is used. MICE treats each variable with missing values as the dependent variable in a regression, with some or all of the remaining variables as predictors. MICE technique cycles through these models, fitting each in turn using a procedure called procedure mean matching (PMM) to generate random draws from the predictive distribution determined by fitted models. These random draws become the imputed values for one imputed dataset [11].

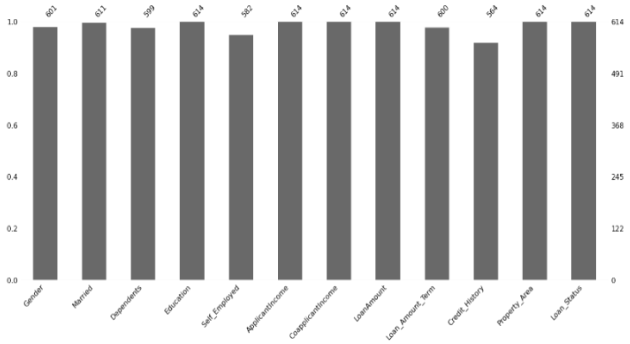

Fig 3: Number of observations present in each variable

After imputation further exploratory data analysis was conducted. Large skewness was observed in the variables applicant income and co-applicant income. Presence of outliers were checked in these columns and one of the observations was removed that seemed irrelevant. Visualizations based on demographic information of customers (Gender, Education and Property area) with loan amount was conducted. It was observed that the gender ratio of male is getting graduated is more compared to females. When it comes to property area customers having property in semiurban have more chances of getting a loan than the rest of the customers. The number of males, graduated, and married people were significantly higher. The number of self-employed people was less. Also, since the data is imbalanced, the proportion of people with good credit history is also high, as the people with loan acceptance is higher.

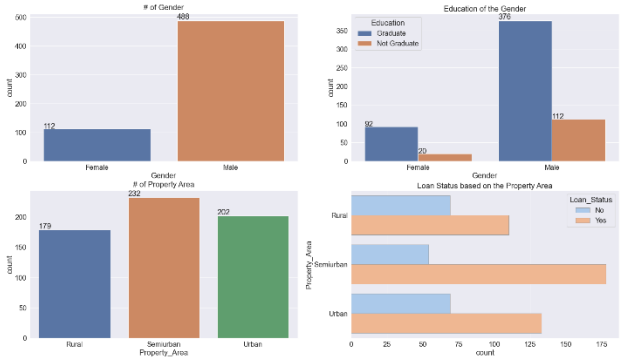

Fig4: Exploratory data analysis

Due to large skewness observed, square root transformation was done on the features - applicant income, co-applicant income and loan amount. This has reduced the skewness to an extend

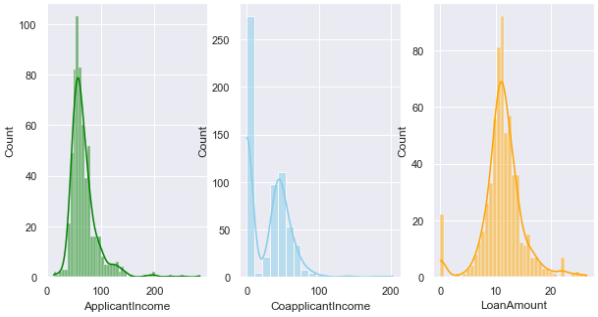

Fig 5: Numerical columns after transformation

Categorical data has to be encoded into numerical data to feed the data into a machine learning algorithm. Label encoder was used to encode categorical variables with 2 categories. Thus, it will be encoded as 0 and 1. In multiclass categorical variables get_dummies() in pandas was used to convert single categorical columns into many dummy or indicator columns. This technique is very similar to one hot encoding. Thus, if a variable has 3 categories, it will be converted into 3 separate dummy variables. As different numerical variables have data in different scales it is important to scale the data before implementing a machine learning model to achieve good performance. Scaling is used to make the data points more generalized or else the distance between the data points will be very far from each other which is not good for a model. Scaling can be done either by using standardization (using z-score) or by normalization. Normalization is the rescaling of features into the range between 0 and 1. Minmaxscaler was used to normalize all numerical variables. In the initial model building stage, Logistic Regression, Support Vector Machine, K- Nearest Neighbor, Decision Tree and Random Forest classifiers were used. These are the

most widely used classical machine learning algorithms for a classification task. The 5 models were trained using the training data with cross validation. Cross validation is a technique to measure the effectiveness of the model created. It is a re-sampling technique of the data to evaluate the performance of the model. Fivefold cross validation was used to validate the results. Out of all the models Support Vector Machine (SVM) and Random Forest models gave the highest validation accuracy. Hence these two models were selected for further hyper parameter tuning.

GridsearchCv was used to conduct hyper parameter tuning. GridsearchCv loops through all the predefined hyper parameters to find the best fit of the model. For SVM the parameters tuned were C, gamma and kernel. C is the penalty parameter of the error term. I.e. It adds a penalty to misclassified points. Gamma determines the curvature weight of decision boundary and kernel is a function in SVM algorithm [12]. Random Forest model the parameters tuned were n_estimators: number of trees in random forest, max_features: number of features to be considered at every split, min_samples_leaf: minimum number of samples required at each leaf node, bootstrap: method of selecting samples for training in each tree, min_saamples_split: minimum number of samples required to split a node, min_samples_leaf: minimum number of samples required at each leaf node [13].

4 Evaluation and Results

Initial evaluation of the models produced the results shown in fig6. It was found that Random Forest and SVM models produced the best validation accuracy.

Model	Accuracy (%)
Logistic Regression	74
Support Vector Machine	77
K- Nearest Neighbours	76
Decision Tree	76
Random Forest	82

Fig6: Validation accuracy of models

Hence these models were selected for hyper parameter tuning. After tuning SVM, best C and gamma parameters were found to be 1. Best kernel was the radial basis function. Fig7 shows the optimum parameters obtained after tuning the Random Forest model.

```
{'bootstrap': True,
 'max_depth': 15,
 'max_features': 'sqrt',
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 200}
```

Fig7: Optimum hyper parameters Random Forest model

In any classification model there would be a positive class and a negative class. Here positive class is defined as 1 or 'Yes' i.e., the customer is eligible for a loan and negative class is defined as 0 or 'No' i.e., the customer is not eligible for the loan.

True positive (TP) → Truly predicted as positive class. That is both predicted and actual classes were positive class. Here it is class 1.

True negative (TN) → Truly predicted as negative class. That is both predicted and actual classes were negative class. Here it is class 0. False positive (FP) → Falsely predicted as positive class. That is the actual labels in training data was negative class 0, but the model

Fig8: Test data statistics on two final models

predictions were positive class 1.

False negative (FN) → Falsely predicted as negative class. That is the actual labels were positive class 1, but the model predicted labels were negative class 0.

The overall test accuracy of the model is given by $(TP+TN) / (TP+TN+FP+FN)$

Accuracy metric depicts how many were correctly predicted by the model out of the total predictions.

Recall/Sensitivity= $TP /(TP+FN)$

Recall tells us how many actual positive classes were there in the entire testing data and out of that how many were predicted correctly. That is, how many class 1 were predicted correctly out of all the class 1 available.

Precision= $TP /(TP+FP)$

Precision tells how good the model is in predictions. Precision talks only about predicted classes.

F1 score is obtained by combining both precision and recall by taking their harmonic mean.

There is a trade-off between precision and recall metrics in machine learning model. When precision increases, recall decreases. And if cut offs were changed to increase recall, precision goes down. So, the amount of precision and recall

required depends on the use case of the model. To understand this, we need to look at the number of false positive and false negative cases in the model predictions and which is more dangerous to the use case.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	75.6	69	71	70
Support Vector Machine	72.3	67	67	67

Fig8: Performance metrics on test data

Here loan prediction classification, false positives are more dangerous. Because in false positive, the actual label was negative or class 0. But the predicted labels by the model were class 1. That is, a person who is not eligible for loan was predicted as eligible by the model. If a loan was issued to him, that will most probably result in a loss for the company. Hence, FP has to be reduced and precision has to be higher.

From Fig8, the accuracy of the final random forest model (75.6%) was found to be higher than SVM model (72.3%). Also, precision, recall and F1 score was found to be higher for the Random Forest model. For further analysis, classification report for each model can be used. It will generate metrics of each model broken down by class.

Classification Report of SVM Model

	precision	recall	f1-score	support
0	0.54	0.54	0.54	37
1	0.80	0.80	0.80	86
accuracy			0.72	123
macro avg	0.67	0.67	0.67	123
weighted avg	0.72	0.72	0.72	123

Classification Report of Random Forest Model

	precision	recall	f1-score	support
0	0.51	0.61	0.56	31
1	0.86	0.80	0.83	92
accuracy			0.76	123
macro avg	0.69	0.71	0.70	123
weighted avg	0.77	0.76	0.76	123

Fig9: Classification report

From fig9 it is observed that although the overall accuracy, precision and recall of both models are fairly good, it is mostly because of the majority class 1. Precision, recall and f1-score of minority class 0 is still not good. Hence classification report tells that although over sampling of minority class was done using SMOTE technique it has not been effective enough to classify them in unseen test data. Also, the overall test accuracy is much less than the training accuracy which indicates potential overfitting of the model. This overfitting might have been due to oversampling or hyper parameter tuning as well.

On comparing the two models, Random Forest model have done well except on the precision of minority class 0.

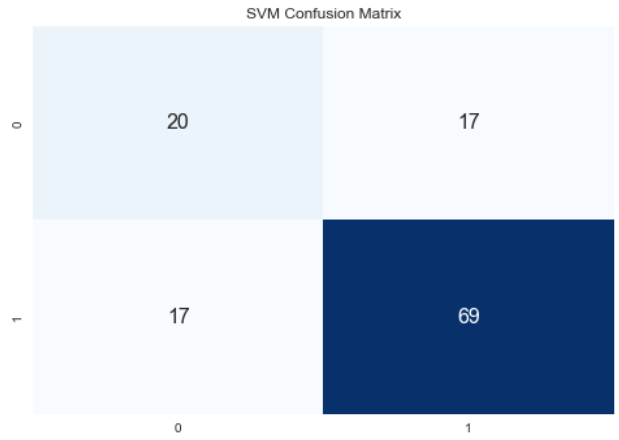

Fig10: Confusion matrix of SVM model

Fig10 can be used for the calculation of all metrics of the SVM model. True Positive-69, True Negative- 20, False Positive- 17, False Negative- 17.

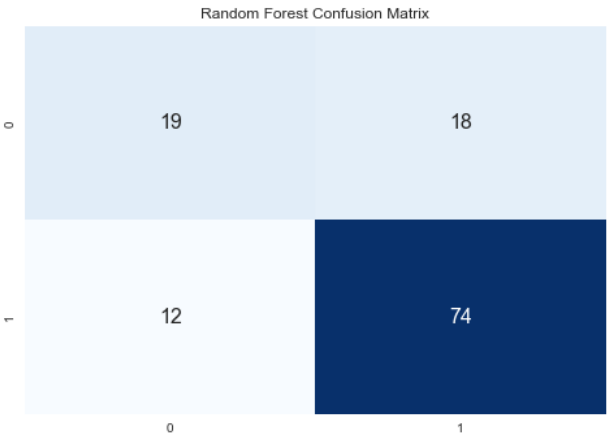

Fig11: Confusion matrix of Random Forest model

From fig11 True Positive-74, True Negative- 19, False Positive- 12, False Negative- 18. For example, calculation of precision for class 1 of Random Forest model will be $TP/(TP+FP) = 74/(74+12) = 0.86$ similarly all other calculations can be completed.

ROC- Receiver operating characteristic and AUC- Area under the curve is another major metric used in the evaluation of a model. ROC curve shows true positive (y axis) vs false positive (x axis) rate

for the model. If AUC is more the model is better.

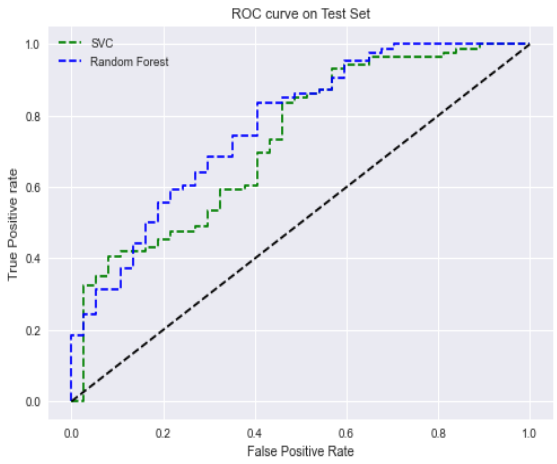

Fig12: ROC curve on two models on test data

Fig12 shows that the ROC curve is more for the Random Forest model. Also, the AUC score for Random Forest model was found to be 77% while that for SVM model was found to be 73%
Overall better metrics values were obtained for Random Forest model and hence it is selected as the best model in this classification problem.

5 Conclusion and Future Work

In this work automation of loan eligibility of customers was done using machine learning algorithms. The dataset obtained showed several challenges. Imputation of missing values was done by MICE technique. The data also had several outliers and skewness. Square root transformation method was applied to reduce skewness of the data and outliers were carefully observed and treated. Encoding of categorical variables and scaling of numerical variables were conducted before building the model. Finally, as there was a class imbalance in the data, SMOTE oversampling was performed. Different models were initially considered in the training phase. Two of the best models were selected and hyper parameter tuning was performed. Several metrics such as accuracy, precision, recall, F1-score, roc-auc on both classes were considered. Finally the Random Forest model with a test data accuracy of 75.6% was selected as the best model by looking into all the metrics.

Although the final model gave a good level of accuracy. It was observed that the test accuracy was lower than the validation accuracy. In future, potential overfitting of the models has to be checked. Further hyper parameter tuning may yield better results. Also, instead of performing over sampling using SMOTE, under sampling techniques like Tomek links can be used to see if it generates better results. Finally, only basic classification models were used in the current analysis. More advanced classification models such as Gradient Boost, Ada Boost etc. can be used to create better performance.

6 References

[1] "Loan Prediction." https://www.kaggle.com/ninzaami/loan-predication (accessed May 11, 2022).
[2] R. B. Avery, R. W. Bostic, P. S. Calem, and G. B. Canner, "Credit Risk, Credit Scoring, and the Performance of Home Mortgages," *Fed. Res. Bull.*, vol. 82, p. 621, 1996.
[3] K. Arun, G. Ishan, and K. Sanmeet, "Loan Approval Prediction based on Machine Learning Approach," p. 4.
[4] A. Goyal, "A survey on Ensemble Model for Loan Prediction," vol. 3, no. 1, p. 6, 2016.
[5] M. A. Sheikh, A. K. Goel, and T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Jul. 2020, pp. 490–494. doi: 10.1109/ICESC48915.2020.9155614.
[6] P. Dutta, "A STUDY ON MACHINE LEARNING ALGORITHM FOR ENHANCEMENT OF LOAN PREDICTION," vol. 03, no. 01, p. 6.
[7] R. Longadge and S. Dongre, "Class Imbalance Problem in Data Mining Review," *arXiv:1305.1707 [cs]*, May 2013, Accessed: May 11, 2022. [Online]. Available: http://arxiv.org/abs/1305.1707
[8] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Information Sciences*, vol. 513, pp. 429–441, Mar. 2020, doi: 10.1016/j.ins.2019.11.004.
[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
[10] J. Wang, M. Xu, H. Wang, and J. Zhang, "Classification of Imbalanced Data by Using the SMOTE Algorithm and Locally Linear Embedding," in *2006 8th international Conference on Signal Processing*, Nov. 2006, vol. 3. doi: 10.1109/ICOSP.2006.345752.
[11] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?," *International Journal of Methods in Psychiatric*

Research, vol. 20, no. 1, pp. 40–49, 2011, doi: 10.1002/mpr.329.
[12] C. Gold and P. Sollich, "Model selection for support vector machine classification," *Neurocomputing*, vol. 55, no. 1, pp. 221–249, Sep. 2003, doi: 10.1016/S0925-2312(03)00375-8.
[13] W. Koehrsen, "Hyperparameter Tuning the Random Forest in Python," *Medium*, Jan. 10, 2018. https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74 (accessed May 09, 2022).