REFLECTIVE REPORT

BAA1027: Machine Learning & Advanced Python

Abstract

This is a reflective report on machine learning algorithms that predict whether a prospective borrower will be approved or denied a loan based on a dataset from Kaagle.

Rory James Mulhern

Student No. 21311696

Course: Business Studies

Course Code: BSI4

Table of Contents

Introduction

There are two sides to the loan approval process: the lender and the borrower. The lender's primary goal is to approve suitable borrowers swiftly and accurately, while borrowers seek a fair and efficient approval process. Machine learning models can enhance the decision-making process by allowing lenders to assess creditworthiness more promptly and improving the speed and consistency of loan approvals. A well-trained model enables lenders to approve eligible applicants rapidly while reserving manual reviews for more complex cases.

As online personal loans gain popularity, banks must explore how they can approve candidates without in-person meetings and lengthy application processes. This necessitates examining the accuracy of predicting loan-worthy applicants quickly and determining how to expedite loan approvals [1]. Many different variables will be considered when getting a loan: age, education, loan intent, income, loan amount, previous defaults, current job, etc.; this is because there are also qualitative factors when it comes to approval loans that usually come into life when it is in the face-to-face meetings, that are becoming less due to online loan applications [2].

While advanced machine learning models such as XGBoost offer high predictive accuracy, their lack of interpretability poses challenges to regulatory compliance and effective communication with clients. Banks must strike a balance between accuracy and transparency, ensuring that models improve efficiency and adhere to legal and ethical standards [3]. This research examines the effectiveness of various machine learning models in predicting loan approvals, focusing on their applicability in real-world banking scenarios. Evaluation metrics such as accuracy, precision, recall, F1-Score and AUC-ROC will be used to compare how effective models are at predicting the creditworthiness of borrowers, using Decision Trees, Random Forest, Logistic Regression and XGBoost.

Identifying the most effective machine learning model for loan approvals can assist banks in streamlining lending decisions, mitigating risk, and improving customer experience, all while ensuring compliance with regulatory standards.

Methodology

Data Description

The example dataset contains records of past loan applications with the final status of approval or denial. Each record has several features relating to the applicant's personal and financial profile [4]. *(See data table of values and explanation in Appendix Table 1)*

Data Prepping

When it came to cleaning the dataset, there were a few outliers in terms of age, with an age of 144 years old. Given this, if an individual is above 70, it is very unlikely that an individual will get a get [5]; I removed anyone whose age is 70 or above from the dataset.

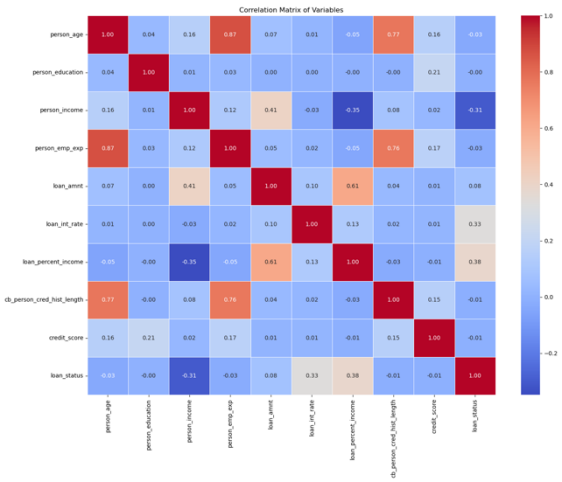
Looking at the data, there were plenty of outliers in the numerical features. From a box-and-whiskers plot, we can see that there are discrepancies between the IQR and the outliers. To first look at this, I used the Windsor method and then a robust scaler to help with the skewness of the data. However, this didn't fully fix the issue; I used a log transformation, which reduced and fixed the skewness of the data.

```
person_age skewness: 1.18
person_income skewness: 1.27
person_emp_exp skewness: 1.23
loan_amnt skewness: 0.94
```

```
person_age skewness after log1p: -0.22
person_income skewness after log1p: -0.72
person_emp_exp skewness after log1p: 0.22
loan_amnt skewness after log1p: -0.67
```

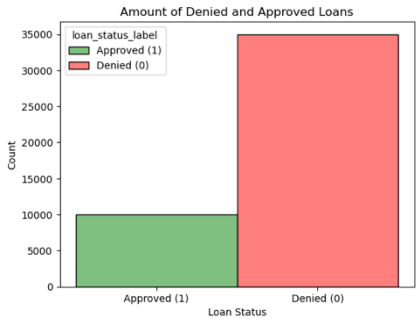
Machine learning models, such as Logistic Regression, struggle with categorical features; they cannot use them to produce outcomes. To help with this and allow this feature to be used, we need to use one-hot encoding, which changes categorical features into binary vectors. This will allow us to use these features to help our model be more accurate [6].

Once one-hot encoding was done on the dataset, I ran a pair plot to see the correlation of the dataset; this should show correlations between different values used to predict our model. However, due to the size of the pair plot, it was hard to see each of the features. To help with this, I ran a correlation matrix to be able to see the information more easily and at a better size to read.



The correlation matrix shows that a few variables are very highly correlated. Due to this, I removed age and employment history from the dataset. I think that credit history shows the individual's previous loan worthiness, while someone may be older, have had no job, or have never taken out a loan. As there are no

credit histories that are older than the age in the same row, I decided to use this feature for the dataset.



With only 22% of loans approved, we need to oversample to allow for a more accurate representation of the model to train on. The Synthetic Minority Oversampling Technique (SMOT) generates synthetic examples of minority classes to balance data [7]. This will be done after cleaning the data and after the training dataset has been split from the test dataset. This will allow for the models' predictions to be more accurate as they will be able to see the same amount of denied and approved loans.

Model Selection

This report will look at different models to determine the effectiveness of classifying loan applications. To achieve this, we need to identify the models we will consider, focusing on classification models such as Logistic Regression, Decision Tree, XGBoost, and Random Forest.

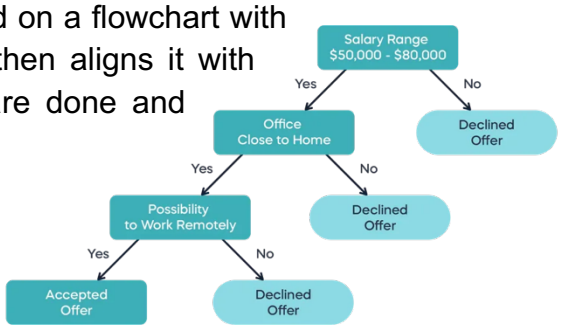
Logistic Regression

Logistic regression is a machine learning algorithm widely used for binary classification, where outcomes have two possible values, such as a borrower getting approved (1) or denied (0) [8].

$$\sigma(z) = \frac{1}{1+e^{-z}}$$, $z=w^{T}x+b$ (linear combination of weights w, input features x, and bias b

Decision Tree

A decision tree model predicts the outcome based on a flowchart with rules. It checks if the criteria are matched and then aligns it with different outcomes until all the final outcomes are done and there are no more options for it to be matched to.



Random Forrest

Random Forest is a model that creates multiple decision trees and combines them to produce a mean of how all of these decision trees perform, improving the model's predictive performance and reducing overfitting [9].

XGBoost

XGBoost is a model that uses gradient boosting to create multiple decision trees, with each tree correcting the previous tree's mistakes. It handles classification quickly and accurately and is commonly used in loan prediction models [9].

K-Nearest Neighbour (KNN)

K-Nearest Neighbours (KNN) is a non-parametric, instance-based learning algorithm utilised for both classification and regression tasks. In the context of predicting loan approvals, KNN classifies a new loan application by comparing it to similar cases in the training data using a distance metric.

Model Training

To train the models, I used a stratified cross-validation using 10 k-folds, which helps reduce overfitting and allows for the model to be trained on the same amount of approved and denied loans. This trains the model on nine folds of data then tests on one fold that is the same size of each of the nine training folds, and repeats it till all of the folds have been used as a training dataset nine times and a testing dataset once.

Results

The evaluation of my loan approval prediction models relies on a set of performance metrics, including accuracy, precision, recall, F1-Score, and AUC-ROC. These metrics were calculated for various models: Logistic Regression, Decision Tree, Random Forest (with both untuned and tuned configurations), XGBoost (untuned and tuned), and K-Nearest Neighbour (KNN). Additionally, the evaluation benefited from oversampling the minority class using SMOTE, which alleviated the inherent class imbalance in our dataset, where only 22% of loans were approved, along with a stratified K-Fold training method.

Overview of Model Performance

Logistic Regression performs reliably, achieving an accuracy close to 90% while balancing precision (78.04%) and recall (75.70%). Its AUC-ROC of 0.955 is notably robust for a relatively simple linear model, demonstrating a strong capability to rank applicants effectively across various thresholds.

Decision Tree has an accuracy of about 90% and balanced precision and recall. It stands out for its interpretability, providing simpler explanations for why certain loans were approved or denied. However, its AUC-ROC of 0.865 is relatively lower than that of some ensemble methods, suggesting that while it performs adequately at a fixed threshold, it is less adaptable when threshold adjustments are needed.

Even before tuning, the **Random Forest (Untuned)** model achieves high accuracy of 92.96%, demonstrating a robust balance between precision at 89.72% and recall at 76.80%. Its AUC-ROC of 0.975 is exceptionally high, reflecting outstanding overall discriminative power. This underscores how ensemble methods can more effectively capture complex, nonlinear relationships in loan data compared to single-tree models.

Random Forest (Tuned v1) slightly lowers its overall accuracy to 92.14%, gaining a higher precision of 91.75%. This trade-off results in reduced recall at 72.30%. An AUC-ROC of 0.968 remains impressive, indicating that this model continues to rank applicants

effectively across various threshold settings even though its approvals are more conservative.

Random Forest (Tuned v2) yields nearly the same accuracy as v1, but increases precision to 92.18% while recall decreases slightly to 71.90%. With an AUC-ROC of 0.968, it remains highly proficient at distinguishing good loans from bad ones. Both tuned Random Forests (v1 and v2) illustrate the typical precision-recall trade-off found in imbalanced classification scenarios.

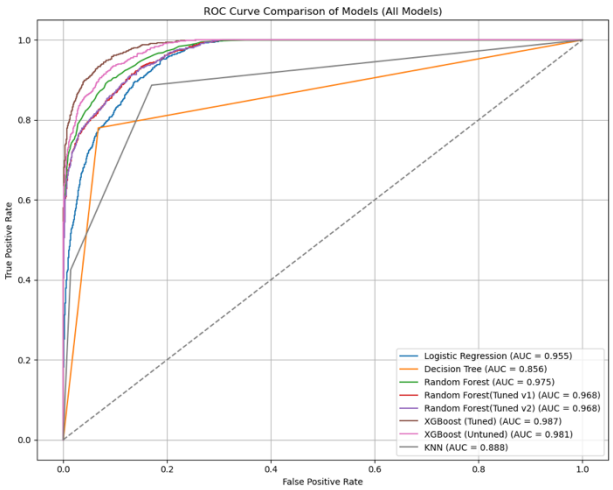
XGBoost (Untuned) already excels with over 93% accuracy and a strong precision-recall balance, resulting in an 84.40% F1-Score. Its AUC-ROC of 0.981 illustrates a remarkable ability to rank applicants effectively and highlights how gradient boosting can capture complex interactions without extensive parameter tuning.

After hyperparameter optimisation, **XGBoost (Tuned)** stands out as the leading performer: accuracy exceeds 94%, precision reaches 93.32%, and recall is at 82.40%. The F1-Score of 87.52% is the highest among all models tested, and its AUC-ROC of 0.987 demonstrates exceptional discriminative power. This indicates that XGBoost (Tuned) effectively ranks applicants at nearly any threshold, making it highly adaptable to changing risk policies.

K-Nearest Neighbour (KNN) shows mixed performance. Its precision is relatively high at 89.47%, but recall drops significantly to 42.50%, indicating that many qualified borrowers are missed. The AUC-ROC score of 0.888 is lower than those from ensemble and boosting methods, highlighting KNN's difficulty in consistently ranking loan applicants in a high-dimensional feature space.

Impact of AUC-ROC

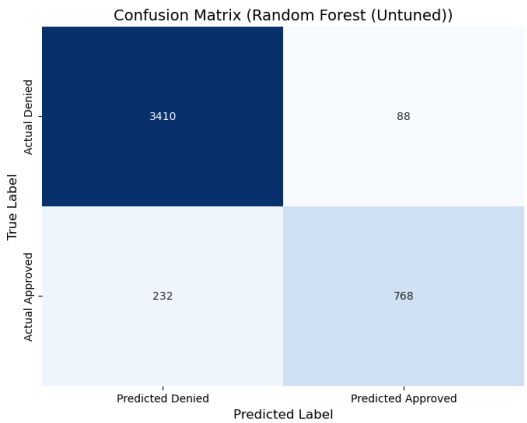
The AUC-ROC metric is very important and requires special attention, as it encapsulates the overall ranking ability of a model independent of any particular threshold. The high AUC-ROC values observed in ensemble methods, particularly in the tuned XGBoost



(0.987) and Random Forest (approximately 0.975), indicate that these models perform well at the current moment and maintain performance across several thresholds [10]. This characteristic is critical for dynamic risk environments. As market conditions or regulatory frameworks shift, lenders may need to adjust approval criteria. A model with a high AUC-ROC ensures that such threshold adjustments won't lead to a drastic change in performance, as the performance among the different thresholds will remain high. This flexibility is particularly valuable in volatile economic conditions, where the cost of approving a risky loan is high, while missing out on a profitable opportunity can also be costly.

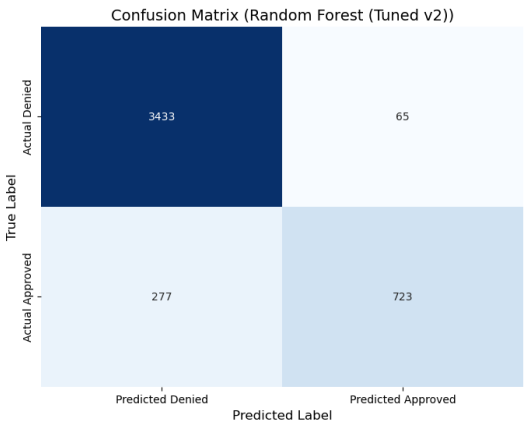
Impact of Precision

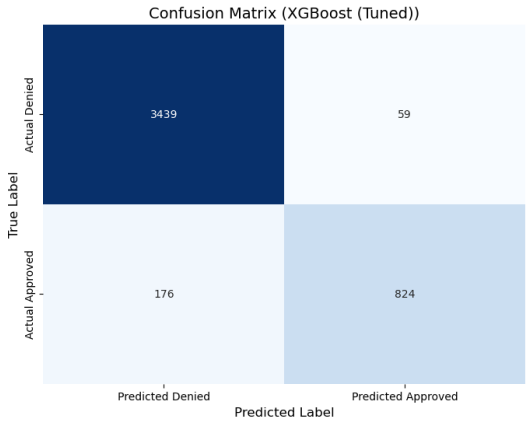
Loan approval models mean that precision is very important, as we don't want risky loans to be approved when there is a high likelihood of default. Although accuracy and recall are important for getting the most correct loans denied and improved, we don't want to lose potentially profitable loans. Letting in loans that should be denied is more problematic due to the cost to the business.



Looking at the random forest models, it may get tricky for a bank to pick which model to use. The untuned model has the highest accuracy and AUC-ROC, but the tuned v2 model has much higher precision while maintaining a similar AUC-ROC value. The two confusion matrices show that the tuned v2 allows for 23 fewer falsely approved loans but misses 45 loans that should have been approved. A bank may choose the less optimal model at this point, as the cost of

taking on borrowers who default on a loan will be worse for the business than missing loans that would be approved.





Overall, the XGBoost model is the best choice for all metrics and accurately predicts approved and denied loans. It will be the most profitable model in the code.

Conclusion

Comparing the five machine-learning classifiers for predicting loan approvals and balancing different evaluation metrics. After using SMOTE and stratified cross-validation, the tuned XGBoost is the best model, achieving the strongest performance across all metrics.

It will translate into fewer risky loans, more correctly approved loans, and faster, fairer decisions for borrowers. Although it is less interpretable compared to models such as decision trees or logistic regression, the trade-off is well worth it due to the much-improved performance of the models.

Future works could incorporate additional metrics, such as live information on what an applicant spends money on or include more economic data related to the current situation with loan approvals and central bank interest rates, which may assist with the volatile credit cycle. However, future research needs to be careful when it comes to minority applicants, as they may be unfairly denied due to previous data where there was bias based on someone's characteristics that have nothing to do with the loan.

Reflective

The dataset I've chosen to work on genuinely interests me. During my internship at KKR, an alternative asset manager, I focused on leveraged credit, specifically corporate bonds and leveraged loans. While these areas differ from personal loans, they have a general connection, as they both involve approving and denying potential borrowers.

I looked at the data and removed those that looked off from the applicants' ages. I then looked at the model in a box-and-whisker chart, used Windsor at 5%, which I learnt about studying for the CFA, and used robust scaler and log transformation, which I was taught about in the lectures. I used other aspects from the lecture, such as correlation, to see which aspects of the model were highly correlated and could be removed.

When it came to what models I was going to use, I used the classification algorithms taught in the lecture, which ran without being hugely computationally expensive. I used information from my econometrics lectures, ChatGPT and Kaagle repositories. I originally was using ChatGPT, which wasn't great for what I wanted to do, so I removed the vast majority of the code and decided to use what I did in my econometrics lectures, Kaagle repositories and the scikit-learn website to look at what the different hyperparameters meant. I used k-folds and stratification to train the model, to remove under- or overfitting on the dataset, and

The evaluation metrics that were used to see how the models performed were all metrics that were taught in lectures and were shown how to code in the tutorial classes. They are very easily understandable, except for the AUC-ROC, which is a bit more complex to understand, but is easily interpretable when understood.

Some code used in this work was adapted from ChatGPT suggestions and publicly available sources on Kaagle, developed by other contributors [11] [12].

Link to code and data – ***Click Here***

References

[1] X. Zhu, Q. Chu, X. Song, P. Hu, and L. Peng, "Explainable prediction of loan default based on machine learning models," *Data Sci. Manag.*, vol. 6, no. 3, pp. 123–133, Sep. 2023, doi: 10.1016/j.dsm.2023.04.003.

[2] M. A. Sheikh, A. K. Goel, and T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm | IEEE Conference Publication | IEEE Xplore." Accessed: Feb. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9155614

[3] R. Hlongwane, K. Ramabao, and W. Mongwe, "A novel framework for enhancing transparency in credit scoring: Leveraging Shapley values for interpretable credit scorecards," *PLOS ONE*, vol. 19, no. 8, p. e0308718, Aug. 2024, doi: 10.1371/journal.pone.0308718.

[4] "Loan Approval Classification Dataset." Accessed: Mar. 09, 2025. [Online]. Available: https://www.kaggle.com/datasets/taweilo/loan-approval-classification-data

[5] R. Goodman and C. Shackleton, "Can I Get a Loan if I'm Retired or Over 60? | MoneySuperMarket," moneysupermarket.com. Accessed: Mar. 10, 2025. [Online]. Available: https://www.moneysupermarket.com/loans/loans-for-pensioners/

[6] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding | IEEE Journals & Magazine | IEEE Xplore." Accessed: Mar. 10, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9512057

[7] E. Hussein Sayed, A. Alabrah, K. Hussein Rahouma, M. Zohaib, and R. M. Badry, "Machine Learning and Deep Learning for Loan Prediction in Banking: Exploring Ensemble Methods and Data Balancing," *IEEE Access*, vol. 12, pp. 193997–194019, 2024, doi: 10.1109/ACCESS.2024.3509774.

[8] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3446–3453, Feb. 2012, doi: 10.1016/j.eswa.2011.09.033.

[9] H. Ashtari, "XGBoost vs. Random Forest vs. Gradient Boosting: Differences | Spiceworks - Spiceworks," Spiceworks Inc. Accessed: Apr. 15, 2025. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/xgboost-vs-random-forest-vs-gradient-boosting/

[10] "Classification: ROC and AUC | Machine Learning," Google for Developers. Accessed: Apr. 16, 2025. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[11] "Loan Approval Insights: Data to Decision-Making 📊." Accessed: Apr. 16, 2025. [Online]. Available: https://kaggle.com/code/sulaniishara/loan-approval-insights-data-to-decision-making

[12] "Loan Approval Classification | EDA & ML." Accessed: Apr. 16, 2025. [Online]. Available: https://kaggle.com/code/youssefelbadry10/loan-approval-classification-eda-ml

Appendix

Data Description

This is a detailed overview of the dataset used in this study. It includes a data dictionary table explaining the dataset's variables, their meaning, and the data type.

Column	Description	Type
person_age	Age of the person	Float
person_gender	Gender of the person	Categorical
person_education	Highest education level	Categorical
person_income	Annual income	Float
person_emp_exp	Years of employment experience	Integer
person_home_ownership	Home ownership status (e.g., rent, own, mortgage)	Categorical
loan_amnt	Loan amount requested	Float
loan_intent	Purpose of the loan	Categorical
loan_int_rate	Loan interest rate	Float
loan_percent_income	Loan amount as a percentage of annual income	Float
cb_person_cred_hist_length	Length of credit history in years	Float
credit_score	Credit score of the person	Integer
previous_loan_defaults_on_file	Indicator of previous loan defaults	Categorical
loan_status (target variable)	Loan approval status: 1 = approved 0 = rejected	Integer

Table 1

Logistic Regression Model Evaluation Metrics	
Average Accuracy	0.8994
Precision	0.7804
Recall	0.7570
F1-Score	0.7685
AUC-ROC	0.9552

Decision Tree Model Evaluation Metrics	
Average Accuracy	0.9017
Precision	0.7677
Recall	0.7800
F1-Score	0.7738
AUC-ROC	0.9552

Random Forest (Untuned) Model Evaluation Metrics	
Average Accuracy	0.9296
Precision	0.8972
Recall	0.7680
F1-Score	0.8276
AUC-ROC	0.8563

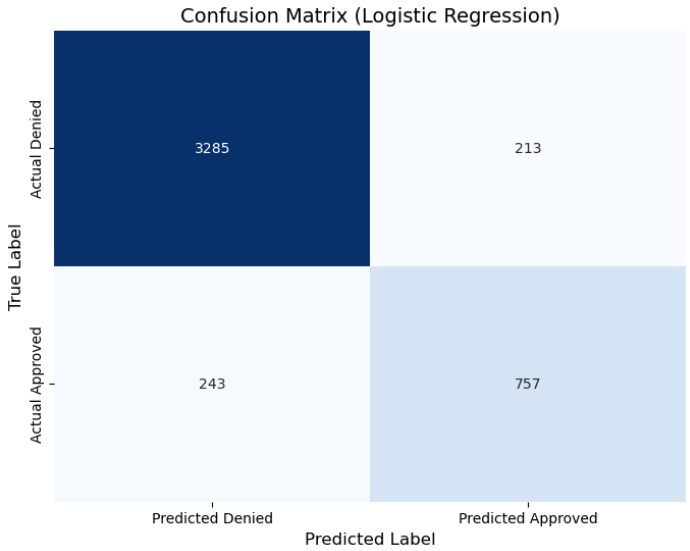
Random Forest (Tuned v1) Model Evaluation Metrics	
Average Accuracy	0.9214
Precision	0.9175
Recall	0.7230
F1-Score	0.8276
AUC-ROC	0.9747

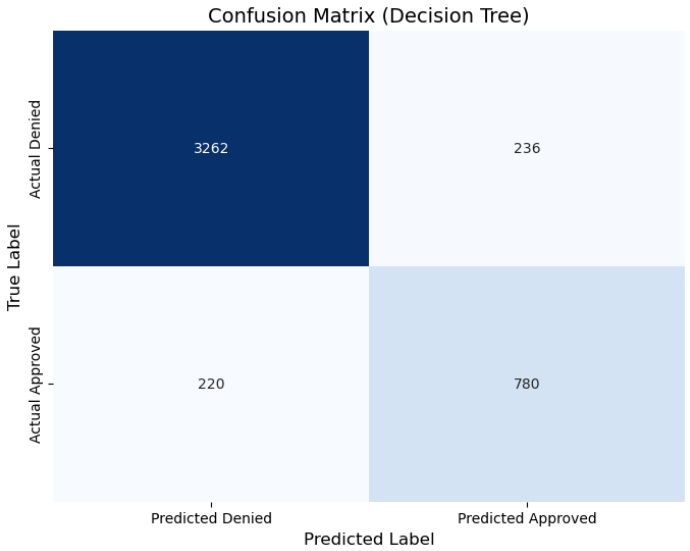
Random Forest (Tuned v2) Model Evaluation Metrics	
Average Accuracy	0.9214
Precision	0.9218
Recall	0.7190
F1-Score	0.8079
AUC-ROC	0.9684

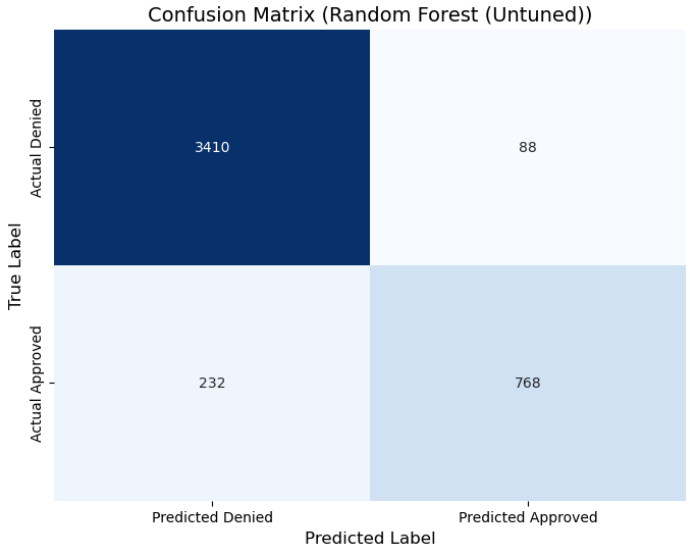
XGBoost (Untuned) Model Evaluation Metrics	
Average Accuracy	0.9375
Precision	0.9098
Recall	0.7870
F1-Score	0.8440
AUC-ROC	0.9810

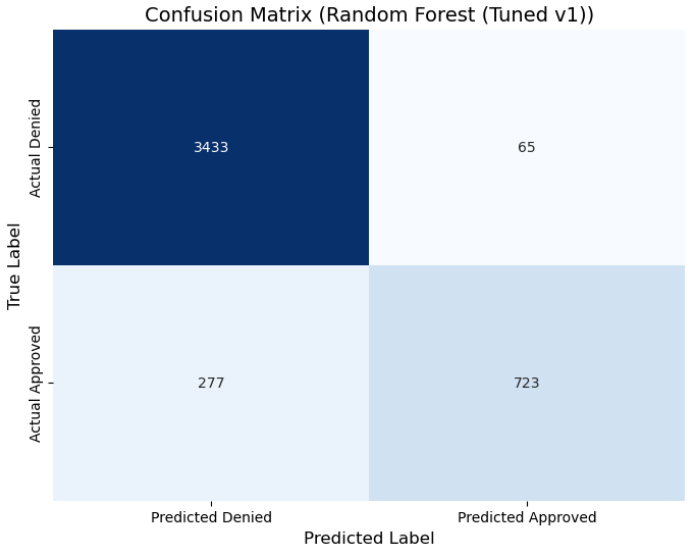
XGBoost (Tuned) Model Evaluation Metrics	
Average Accuracy	0.9486
Precision	0.9332
Recall	0.8240
F1-Score	0.8572
AUC-ROC	0.9868

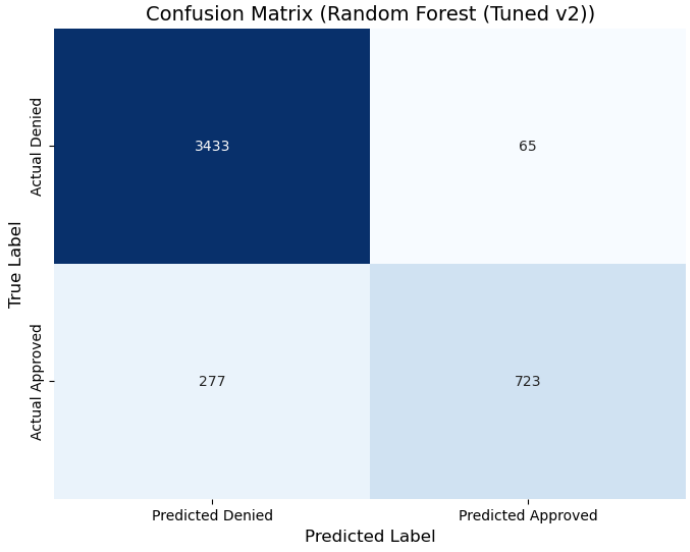
K-Nearest Neighbour Model Evaluation Metrics	
Average Accuracy	0.8678
Precision	0.8947
Recall	0.4250
F1-Score	0.5763
AUC-ROC	0.8882

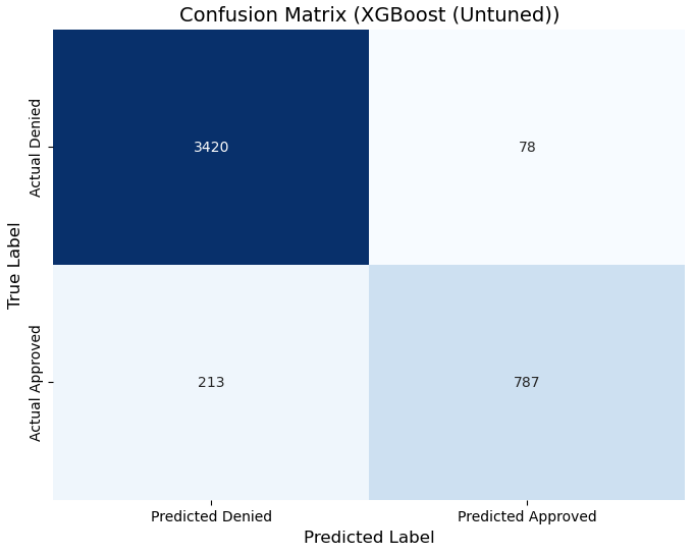

Confusion Matrix (Logistic Regression)
Actual Denied
Actual Approved
True Label
3285
213
243
757
Predicted Denied
Predicted Approved
Predicted Label

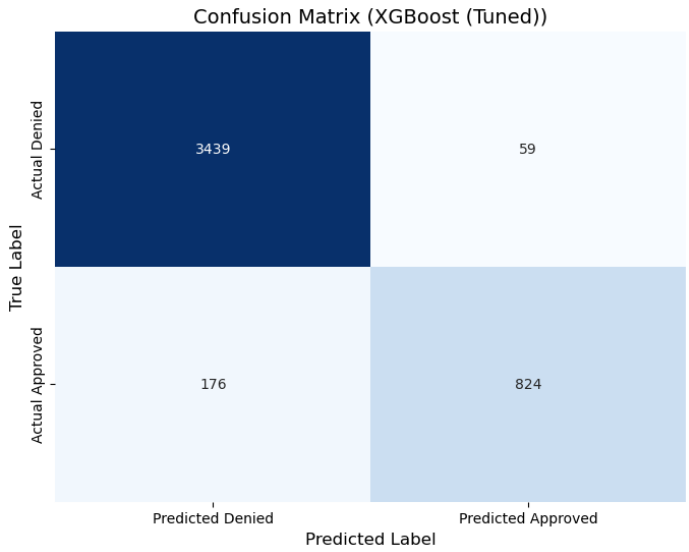

Confusion Matrix (Decision Tree)
Actual Denied
Actual Approved
True Label
3262
236
220
780
Predicted Denied
Predicted Approved
Predicted Label

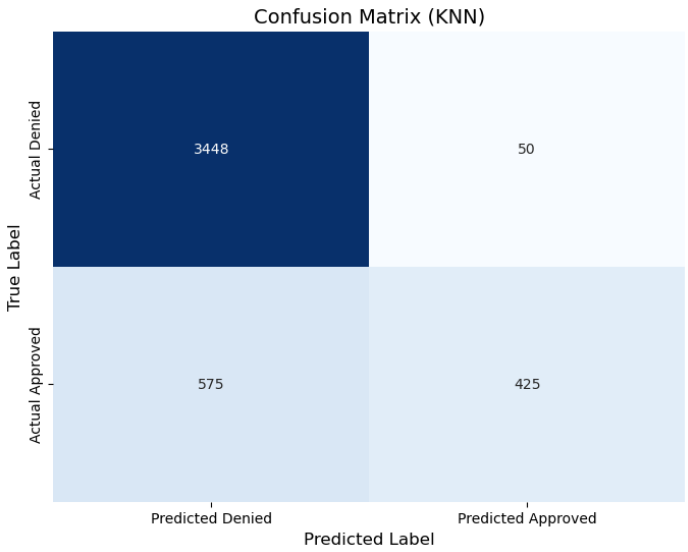

Confusion Matrix (Random Forest (Untuned))
Actual Denied
Actual Approved
True Label
3410
88
232
768
Predicted Denied
Predicted Approved
Predicted Label


Confusion Matrix (Random Forest (Tuned v1))
Actual Denied
Actual Approved
True Label
3433
65
277
723
Predicted Denied
Predicted Approved
Predicted Label


Confusion Matrix (Random Forest (Tuned v2))
Actual Denied
Actual Approved
True Label
3433
65
277
723
Predicted Denied
Predicted Approved
Predicted Label


Confusion Matrix (XGBoost (Untuned))
Actual Denied
Actual Approved
True Label
3420
78
213
787
Predicted Denied
Predicted Approved
Predicted Label


Confusion Matrix (XGBoost (Tuned))
True Label
Actual Denied
Actual Approved
3439
59
176
824
Predicted Denied
Predicted Approved
Predicted Label


Confusion Matrix (KNN)
True Label
Actual Denied
Actual Approved
3448
50
575
425
Predicted Denied
Predicted Approved
Predicted Label