Loan Approval Prediction Project

Prepared by:

Karina Sanchez-Duran
& Guillaume Lachapelle

March 29, 2025

Table of Contents

Abstract **3**
Introduction **3**
Data Collection **4**
Statistical Analyses **5**
Data Wrangling 5
Random Forest Implementation 8
Decision Tree Implementation 10
Gaussian Naive Bayes Implementation 13
Discussion of Results and Interpretation **15**
Conclusion **16**
References **17**

Abstract

The growing need among Canadians to apply for a loan in the current economy and the intricacy of the loan approval process presents a serious issue in Canada. Machine learning is a viable solution to this problem. In this report, an analysis of three different machine learning models was done in order to determine which would perform best and, thus, which should be utilized as a possible solution to the current loan application problem. The three models that were implemented and evaluated were a random forest model, a decision tree model, and a Gaussian Naive Bayes model. All three models were trained and tested with the *Loan Approval Classification Dataset* [4] on Kaggle which contains 14 features and 45000 rows of data.

All three models were evaluated based on accuracy, precision, recall and F1 score. A k-fold cross validation was also done on all three models using accuracy and the results indicate that the data is balanced and no model overfits. The results further indicate that the random forest model had the best overall performance with an accuracy of 92.63%, a precision of 88.73%, a recall of 76.77%, and an F1- score of 82.32% (as seen in *Figure 6* below). Lastly, both the random forest model and the decision tree model reveal that the top five most important features in determining whether a loan gets approved or rejected are previous loan defaults, loan percent income, loan interest rate, annual income and home ownership status. Thus, financial institutions focus on a loan applicant's ability to pay back a loan and their credibility to come through on payments when determining whether to accept or reject a loan application.

Introduction

The assessment of a potential loan candidate's financial situation is crucial for financial institutions to determine if a loan should be approved. There are many complex factors that financial institutions need to take into account when performing an evaluation of a potential loan candidate and a growing demand for loans [1]. According to *Statistics Canada*, there has been a steady increase in Canadian debt levels since 2022 which can be attributed to inflation, higher costs of living, and lower savings rates [2]-[3]. As such, there has been an increase in non-mortgage loans, and vehicle loans since 2022 [2]. In fact, non-mortgage loans increased by 13.5% between 2020 and 2023, vehicle loans increased by 16.3% between 2020 and 2023 [2].

The growing demand for loans in Canada and the complexity of the loan approval process presents a serious problem in the Canadian economy. Machine learning algorithms are an efficient solution to streamline the loan approval process. Financial institutions can expedite the process and applicants can receive news regarding their application in a timely manner. In this project, various machine learning classification algorithms will be implemented using a dataset containing relevant information about over 45 000 potential loan candidate information in order

to determine if a loan application should be approved or denied. The performance of these various implementations will be compared and analyzed in order to maximize results.

Data Collection

The dataset used for this project was collected from the *Loan Approval Classification Dataset* [4] on Kaggle. As stated on the Kaggle page, "This dataset is a synthetic version inspired by the original *Credit Risk dataset on Kaggle* [5] and enriched with additional variables based on *Financial Risk for Loan Approval data*" [6]. It contains 45,000 records and 14 variables, each described below:

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 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

The dataset was created for the purposes of Exploratory Data Analysis (EDA), Classification, and Regression. In the scope of this project, the main focus was EDA, such as data wrangling and analyzing key features, as well as Classification by building predictive models to classify the *loan_status* variable (approved/not approved) for potential applicants.

Statistical Analyses

Data Wrangling

Before implementing any machine learning algorithm, the data needed to undergo pre-processing. Thus, the data was analyzed to check for missing values, and duplicate values. Any row with missing values was dropped from the dataset and any duplicates were also dropped from the dataset. Additionally, columns were renamed to be more descriptive and data in columns were unified. For instance, the data was all brought to lowercase. The data was also visualized in order to better understand the relationships between variables.

Figure 1 below shows the loan approval status distribution of the dataset. It demonstrates that about 35000 loans were rejected, while about 10000 were approved. This gives a good summary of the proportion of approved loans within the dataset.

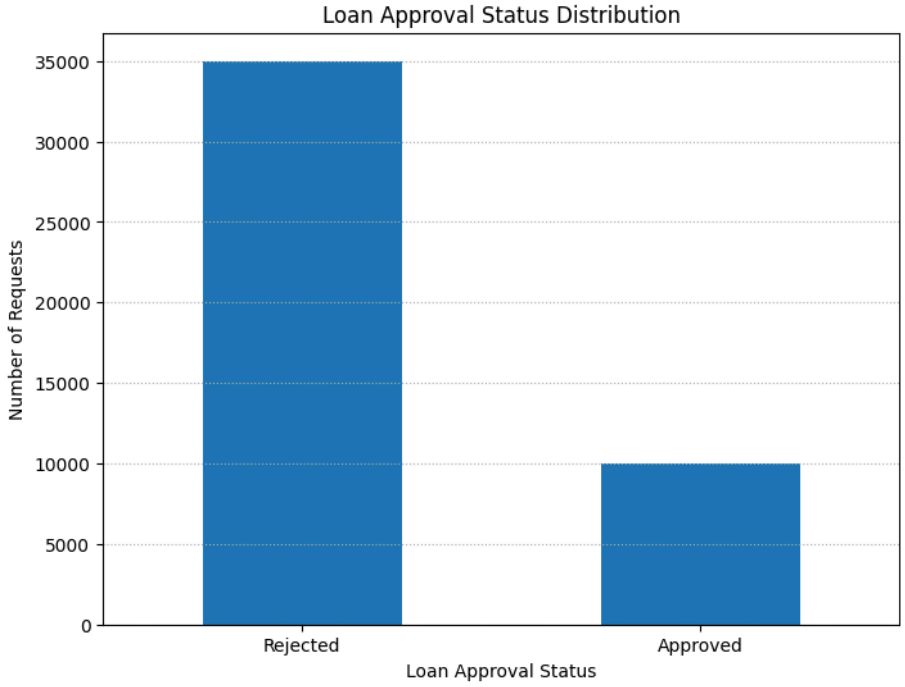

Figure 1: Loan Approval Status Distribution

Figure 2 below shows the loan approval status by loan intent. It helps determine which loan is most likely to be approved based solely on loan intent.

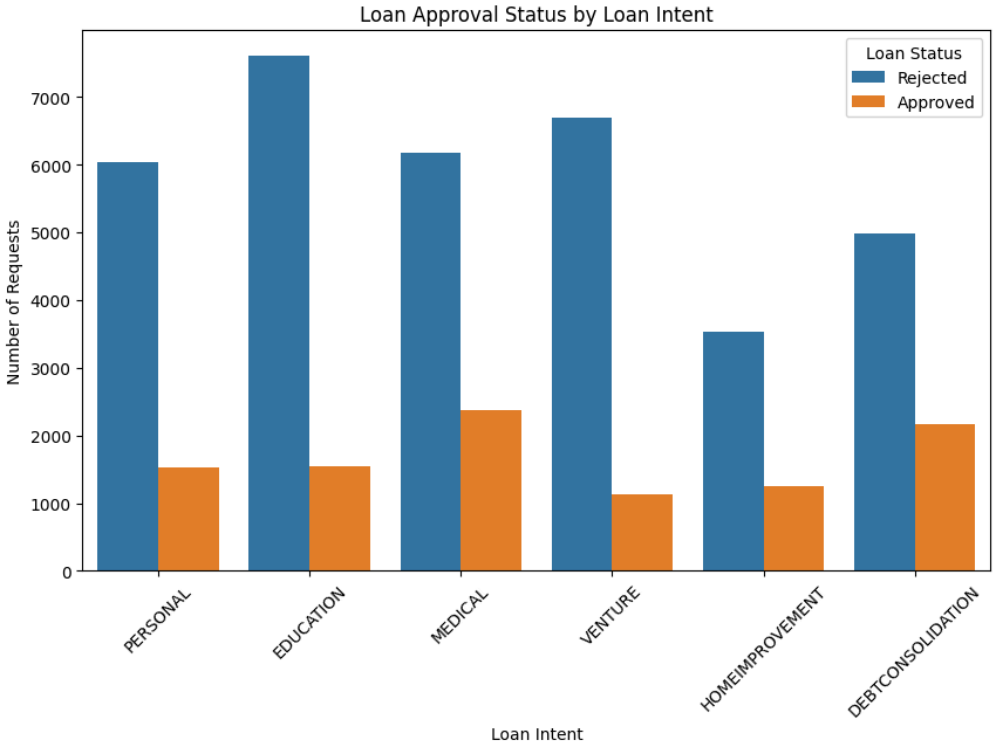

Figure 2: Loan Approval Status by Loan Intent

Figure 3 below shows the loan approval status by loan percentage of income. It helps determine which loan is most likely to be approved based solely on the percentage of a person's income that the loan represents.

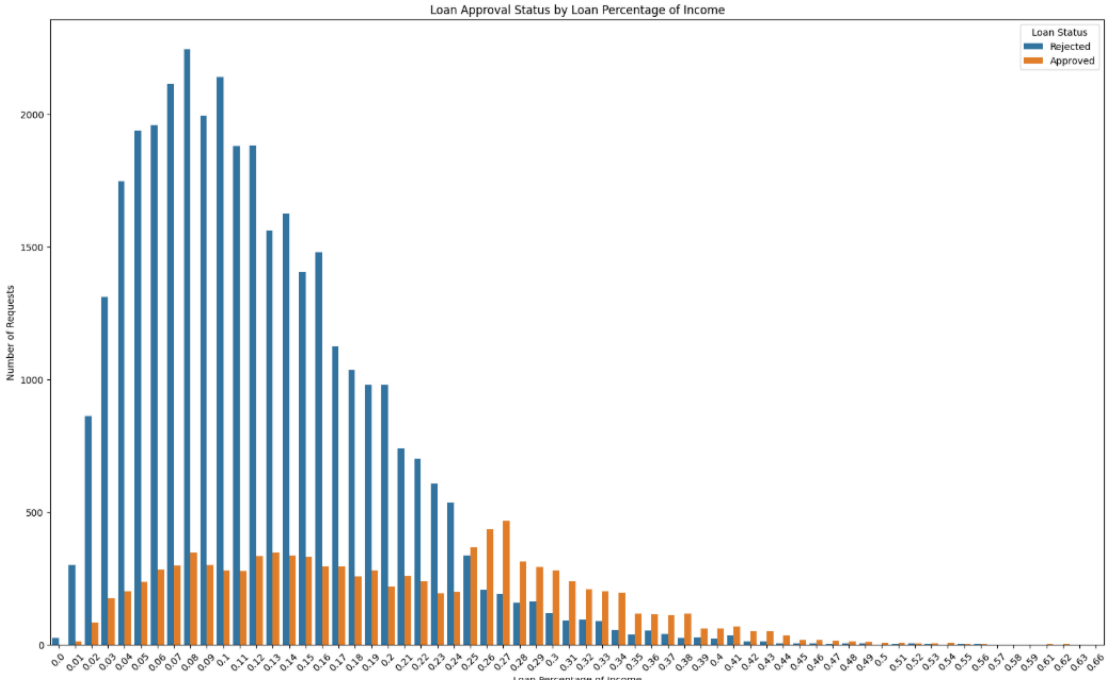

Figure 3: Loan Approval Status by Loan Percentage of Income

Figure 4 below shows the loan approval status by loan defaults. It helps determine if missing a payment on a previous loan has a big impact on if a new loan will be approved or rejected.

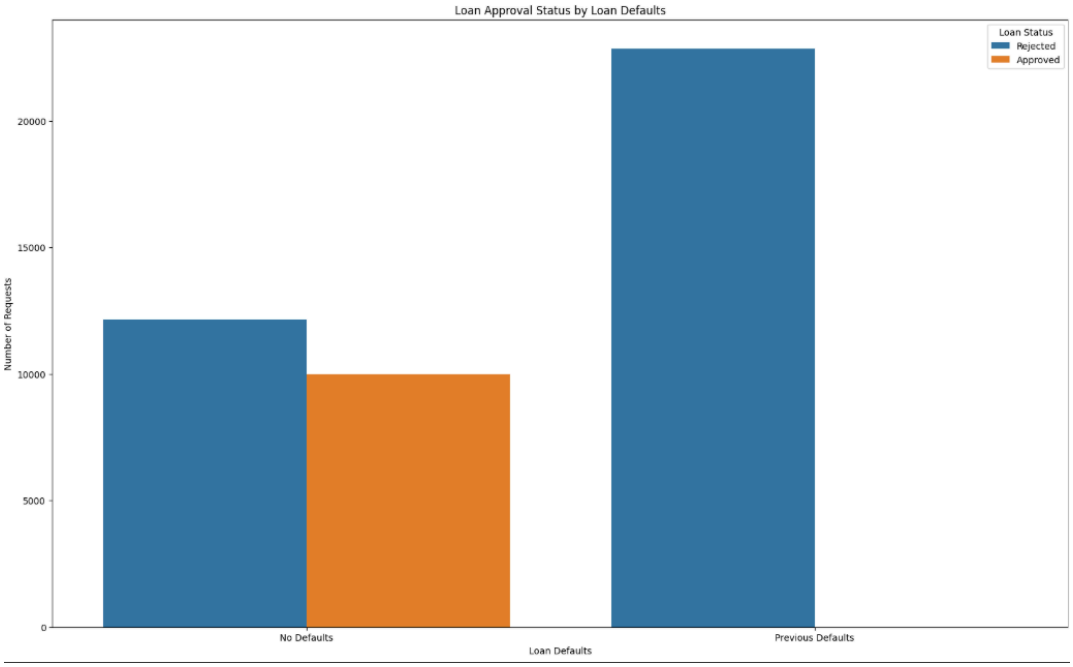

Figure 4: Loan Approval Status by Loan Defaults

Random Forest Implementation

In the random forest implementation, there were several hyperparameters used to improve performance of the model. Namely, the parameters *n_estimators*, *max_depth*, *min_sample_split*, and *min_samples_leaf*. The parameter *n_estimators* refers to the number of trees used in the random forest model [7]-[8]. The *max_depth* parameter refers to the maximum depth of the trees in the random forest. The *min_sample_split* parameter specifies the minimum number of samples required to split a node and the, finally, the *min_samples_leaf* refers to the minimum number of samples in a leaf [7]-[8]. The GridSearchCV algorithm which evaluates the model using the hyperparameters then selects the model with the hyperparameters that had the best accuracy.

The best performing model is then visualized in the form of a confusion matrix (seen in *Figure 5*) and further evaluated based on accuracy, precision, recall and F1-score (as seen in *Figure 6*). A k-fold cross-validation is done on the model as well in order to ensure the model is not overfitting (as seen in *Figure 7*) [9]-[10]. Lastly, a feature importance, seen in *Figure 8*, is done on the model in order to determine which features have a greater impact on the outcome of the model (i.e. which features are most important in determining if a loan is accepted or rejected).

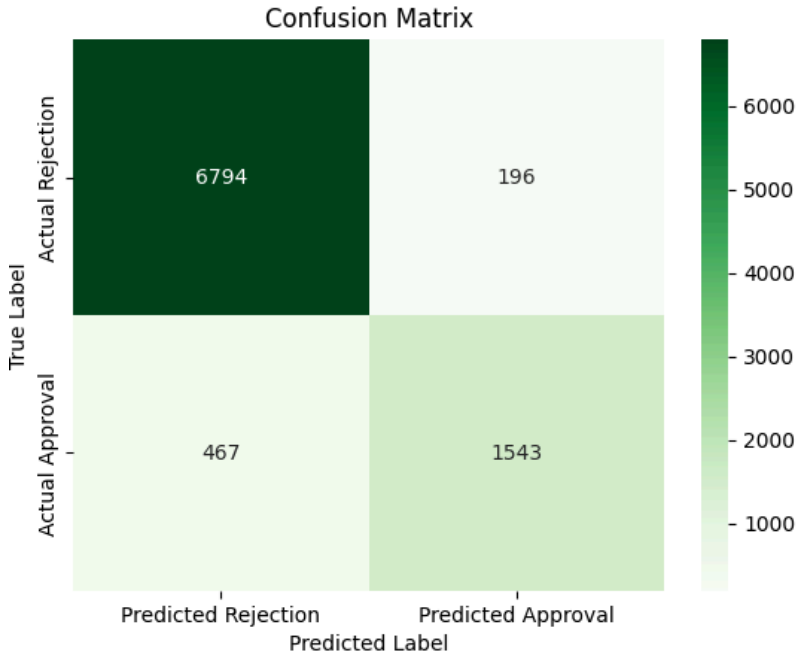

Figure 5: Confusion Matrix for Random Forest

Accuracy	Precision	Recall	F1 Score
92.63%	88.73%	76.77%	82.32%

Figure 6: Metrics table for Random Forest

Fold	Accuracy
1	0.9258
2	0.9262
3	0.9289
4	0.9280
5	0.9327
6	0.9218
7	0.9267
8	0.9307
9	0.9258
10	0.9258
Average	0.9272

Figure 7: k-fold Evaluation Table for Random Forest

Feature	Importance
previous_loan_defaults	0.3161
loan_percent_income	0.1496
loan_int_rate	0.1472
annual_income	0.1139
home_ownership_status	0.0694

loan_amnt	0.0533
credit_score	0.0445
loan_intent	0.0317
age	0.0215
years_of_employment	0.0197
years_of_credit_hist	0.0180
education	0.0108
gender	0.0044

Figure 8: Feature Importance for Random Forest

Decision Tree Implementation

In the decision tree implementation, we use multiple hyperparameters such as *max_depth*, *min_samples_split*, and *min_samples_leaf* to create multiple trees and find the best model [12]. To be able to find the best one using a combination of the hyperparameters mentioned above, we use GridSearchCV, which then gives a *best_estimator_* as a result of its search [7]. This best estimator found by GridSearchCV is the one that was selected for this project, and was therefore trained on the cleaned loan data and then tested. The decision tree classifier was created with the 'entropy' criterion and we decided to use 'accuracy' as the scoring parameter for the GridSearchCV algorithm, which therefore selects the best model based on accuracy.

During its evaluation phase, a confusion matrix was generated, which you can see in *Figure 9*, as well as multiple metrics such as accuracy, precision, recall, and F1-Score, which are displayed in *Figure 10*. Finally, a k-fold cross-validation was performed on the best model, which you can see in *Figure 11*, and a list of the feature importances were extracted from the model and displayed in *Figure 12*.

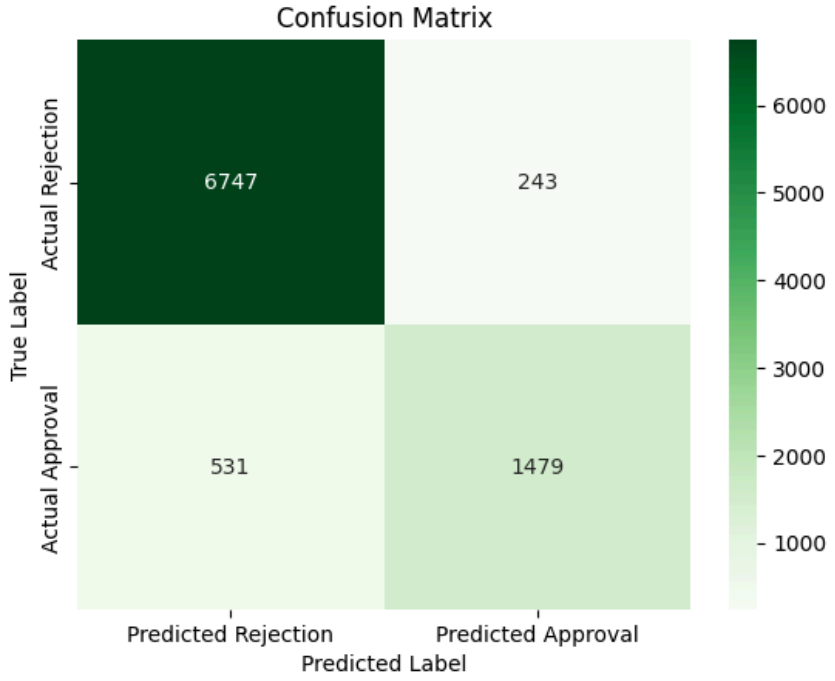

Figure 9: Confusion Matrix for Decision Tree

Accuracy	Precision	Recall	F1 Score
91.40%	85.89%	73.58%	79.26%

Figure 10: Metrics table for Decision Tree

Fold	Accuracy
1	0.9149
2	0.9207
3	0.9180
4	0.9187
5	0.9193
6	0.9129

7	0.9153
8	0.9233
9	0.9202
10	0.9153
Average	0.9179

Figure 11: k-fold Evaluation Table for Decision Tree

Feature	Importance
previous_loan_defaults	0.5098
loan_percent_income	0.1470
loan_int_rate	0.1418
annual_income	0.0825
home_ownership_status	0.0620
loan_intent	0.0258
credit_score	0.0207
loan_amnt	0.0041
age	0.0026
years_of_credit_hist	0.0013
years_of_employment	0.0012
education	0.0012
gender	0.0000

Figure 12: Feature Importance for Decision Tree

Gaussian Naive Bayes Implementation

In the Gaussian Naive Bayes implementation, the basic model using *scikit-learn* was implemented [11]-[13]. Only the basic model was used because there are no real hyperparameters to use for the Gaussian Naive Bayes model in order to improve performance.

After the Gaussian Naive Bayes model was implemented, it was evaluated based on accuracy, precision, recall and F1-score. It was also visualized in the form of a confusion matrix, as seen in *Figure 13*. The main metrics such as accuracy, precision, recall, and F1-Score are displayed in *Figure 14*. Lastly, a k-fold cross validation was performed on the model in order to test for possible overfitting. The results of the k-fold cross validation can be found in *Figure 15*.

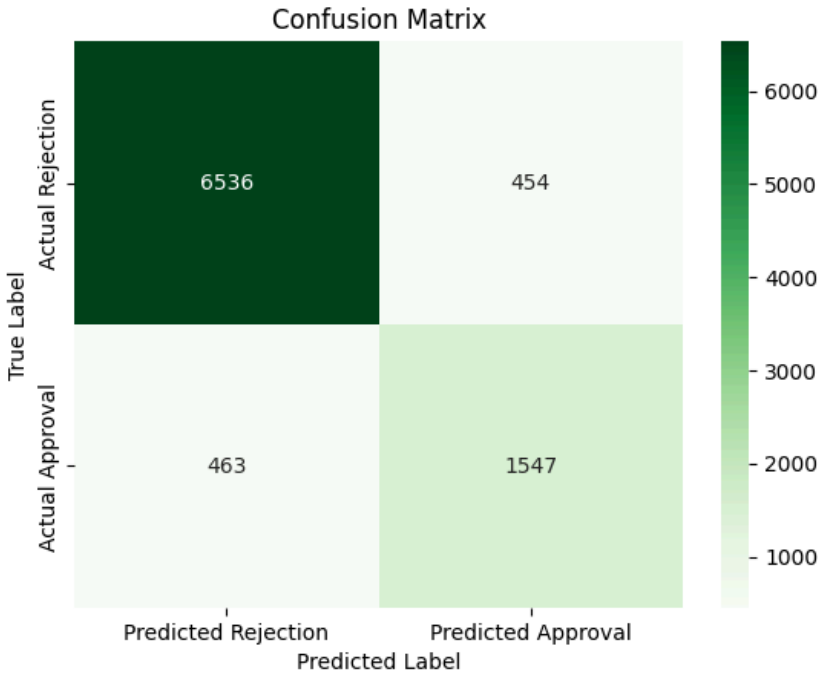

Figure 13: Confusion Matrix for Gaussian Naive Bayes

Accuracy	Precision	Recall	F1 Score
81.06%	66.49%	30.60%	41.91%

Figure 14: Metrics table for Gaussian Naive Bayes

Fold	Accuracy
1	0.7998
2	0.8198
3	0.8151
4	0.8222
5	0.8053
6	0.8171
7	0.8129
8	0.8196
9	0.8180
10	0.8151
Average	0.8145

Figure 15: k-fold Evaluation Table for Gaussian Naive Bayes

Discussion of Results and Interpretation

The model with the best overall metrics was the random forest classifier. This model had the best values for all four metrics, which were 92.63% accuracy, 88.73% precision, 76.77% recall, and 82.32% F1-Score (as seen in *Figure 6*). This shows that the model was able to correctly classify most of the features, resulting in a high number of true positives and true negatives, as well as a low number of false positives and false negatives. These high metrics demonstrate that the random forest classifier is a good candidate for classifying whether a loan gets approved or rejected. The decision tree model had the second best overall performance with an accuracy of 91.40%, a precision of 85.89%, a recall of 73.58% and an F1 score of 79.26% (as seen in *Figure 10*). The Gaussian Naive Bayes model had the worst performance of all three with an accuracy of 81.06%, a precision score of 66.49%, a recall of 30.60% and an F1 score of 41.91% (as seen in *Figure 14*).

K-fold cross validation was performed on all three models. The purpose of k-fold cross validation is to split the testing data into k groups and test the model with that subset of data. This ensures that the model does not overfit and can therefore generalize to an independent dataset. Looking at the results from *Figure 7*, *Figure 11*, and *Figure 15*, we can see that the accuracy doesn't change significantly between folds. This proves that the models do not overfit and that it can generalize to independent datasets. This cross validation was used to ensure this previous requirement, though even if it shows accuracy, it does not necessarily demonstrate which model is the best. For that purpose, as mentioned above, the combination of all evaluated metrics, which are accuracy, precision, recall, and F1-Score, were used to determine which classification model was the best fit for our use case.

A list of feature importance was determined for both the random forest implementation (as seen in *Figure 8*) and the decision tree implementation (as seen in *Figure 12*). The random forest model and the decision tree model both revealed that the top five most important features in determining whether a loan gets approved or rejected are previous loan defaults, loan percent income, loan interest rate, annual income and home ownership status. Meaning that, in general, financial institutions prioritize features that help determine an applicant's ability and trustworthiness to pay back a loan. The top 5 least important features are age, years of employment, years of credit history, education and gender. Meaning that, in general, financial institutions do not focus on personal background and attributes when determining if a loan should be accepted or rejected.

With the growing trend of Canadians applying for a loan and the complexity of the loan approval process, a machine learning algorithm is a viable solution to streamline the entire process. In this report, the random forest model was seen to produce the best results and can thus be utilized to expedite the loan application process efficiently. Banking institutions can also modify the

random forest's hyperparameters if time permits, given that they have access to good enough hardware that allows for heavier and longer processes. Parameters such as the number of estimators and the maximum depth of each tree could enable them to find a better performing model. This requires a thorough evaluation and k-fold cross validation process to ensure that the new model doesn't overfit. However, it can be a good idea to test out different hyperparameters to try to find an even more optimal model.

Conclusion

In conclusion, the random forest classifier model was found to be the best choice for determining if a loan should be accepted or rejected. Using multiple evaluation tools such as a confusion matrix, as well as measuring the accuracy, precision, recall, and F1-Score of each model, it was found that the random forest classifier had the best metrics. To make sure that the model didn't overfit, k-fold cross validation was performed on the model, ensuring that the accuracy didn't have too much variation between folds, therefore demonstrating that the model was able to generalize to unseen datasets. The random forest classifier also offers multiple hyperparameters which can be modified to attempt to improve the model's performance.

In a world where artificial intelligence and machine learning are closely knitted with everyday use of technology, it is important to build models that can efficiently and correctly classify data. This loan classification model could not only greatly speed up the loan approval process for banking institutions, therefore saving time and money, but it would also be a useful tool for clients to have an easy and fast estimation of if their loan request would be approved or not. This model could be added as a tool on banking institutions' online platforms as a way to facilitate the loan approval process. In the future, financial institutions can modify the random forest model to improve its overall performance if they have access to more powerful hardware.

References

[1] F. M. A. Haque and Md. M. Hassan, “Bank loan prediction using Machine Learning Techniques,” SCIRP, https://doi.org/10.4236/ajibm.2024.1412085 (accessed Mar. 26, 2025).

[2] A. Fortier-Labonté and M. McGillivray, “This analysis is divided into three sections. section 1 looks at the principal components of non-mortgage loans, while section 2 looks at mortgage loans and the breakdown of insured and uninsured mortgages. section 3 looks at both mortgage and non-mortgage loans and highlights indicators related to household indebtedness and financial stability.,” The evolving landscape of Canadian lending: Key trends in mortgage and non-mortgage loans, https://www150.statcan.gc.ca/n1/pub/11-621-m/11-621-m2024009-eng.htm (accessed Mar. 26, 2025).

[3] T. Canada, “Canadian Consumer Debt Continues to Grow Despite Macroeconomic Relief,” *Canadian Consumer Debt Continues to Grow Despite Macroeconomic Relief* , Feb. 19, 2025. https://newsroom.transunion.ca/canadian-consumer-debt-continues-to-grow-despite-macroeconomic-relief/ (accessed Mar. 26, 2025).

[4] T. Lo, “Loan approval classification dataset,” Kaggle, https://www.kaggle.com/datasets/taweilo/loan-approval-classification-data (accessed Mar. 17, 2025).

[5] L. Tse, “Credit risk dataset,” Kaggle, https://www.kaggle.com/datasets/laotse/credit-risk-dataset (accessed Mar. 17, 2025).

[6] L. Zoppelletto, “Financial risk for loan approval,” Kaggle, https://www.kaggle.com/datasets/lorenzozoppelletto/financial-risk-for-loan-approval (accessed Mar. 17, 2025).

[7] “GRIDSEARCHCV,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed Mar. 21, 2025).

[8] “Randomforestclassifier,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed Mar. 21, 2025).

[9] “3.1. cross-validation: Evaluating estimator performance,” scikit, https://scikit-learn.org/stable/modules/cross_validation.html (accessed Mar. 21, 2025).

[10] “Kfold,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html (accessed Mar. 21, 2025).

[11] GeeksforGeeks, “Gaussian naive Bayes using Sklearn,” GeeksforGeeks, https://www.geeksforgeeks.org/gaussian-naive-bayes-using-sklearn/ (accessed Mar. 21, 2025).

[12] GeeksforGeeks, “Building and implementing decision tree classifiers with Scikit-Learn: A Comprehensive Guide,” GeeksforGeeks, https://www.geeksforgeeks.org/building-and-implementing-decision-tree-classifiers-with-scikit-learn-a-comprehensive-guide/ (accessed Mar. 22, 2025).

[13] “Gaussiannb,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html (accessed Mar. 21, 2025).