MSc in Business Administration and Data Science
Final Project Machine Learning and Deep Learning

Home Mortgage Approval Prediction

Alvaro Diaz-Davila Alvarez; 167299

Jan Portius; 167605

Julian Irigoyen; 167279

Lecturer: Somnath Mazumdar
Date: 17.05.2024
Pages: 11
Characters: 28541
Link to GitHub Repository:
`https://github.com/Bartimaeus25/ML_DL_Final_Project`

Abstract

The topic of this study is the prediction of U.S. home mortgage approvals using machine learning models. The problem formulation involves integrating previous loan approval decisions into the current process to enhance consistency and accuracy. The research question addresses which machine learning models can best predict loan approval decisions based on historical data. The concepts explored include feature importance, model interpretability, and the balance between model complexity and performance. The dataset used is the 2022 Home Mortgage Disclosure Act (HMDA) data, and the main data analytics methods and tools are Logistic Regression, Random Forest, and Multi-Layer Perceptron (MLP). The most important results include Random Forest achieving the highest performance with an accuracy of 96.14% and valuable insights on feature importance for credit risk assessment. The conclusions and recommendations suggest that Random Forest is the most suitable model for this application due to its balance of performance, interpretability, and computational efficiency.

Keywords: Credit Risk Assessment, Loan Approval, Machine Learning, Random Forest, Logistic Regression, Multi-Layer Perceptron, Feature Importance, Home Mortgage Disclosure Act

1 Introduction

Credit risk assessment is a core process for banks, savings associations, and credit unions. These assessments are crucial for maintaining competitiveness and profitability by reducing loan defaults. Typically, credit scoring procedures are applied to make approval decisions. Assuming that lenders achieve high rates of correct approval decisions, this study proposes incorporating previous decisions into the process to further minimize risk of loan default. Such an additional layer of validation might be beneficial in the following ways: First, it provides a consistency check by assessing whether the decision is aligned with the previous decision boundary, helping maintain fairness and transparency. Second, it can detect anomalies in decisions, potentially identifying errors or unique situations and consequently improve accuracy.

In particular, this study aims to predict the approval of U.S. home mortgages by applying three machine learning models: Logistic Regression, Random Forest, and Multi-Layer Perceptron. Effective models could then be integrated into the decision-making process to enhance its overall reliability. Consequently, the research question of this study is: Which machine learning model can best predict loan approval decisions based on previous decisions?

The remainder of this paper is organized as follows: Section 2 reviews the relevant literature. Section 3 describes the methodology, including the dataset description, data analysis process, and modeling methods. Sections 4 and 5 discuss the hyperparameter tuning of the models and present the results, which are discussed in Section 6, including ethical considerations. Section 7 outlines the limitations of the analysis. Finally, Section 8 concludes the paper and suggests directions for future work.

2 Related Work

The field of loan approval has been extensively studied, with numerous methodologies and models proposed to improve the accuracy and reliability of predictions. This section reviews the significant contributions in this domain, focusing on the application of machine learning models to predict loan approval.

Kandula et al. (2024) compare several machine learning methods for predicting loan approval. The study incorporates Naïve Bayes, XGBoost, K-NN, Decision Tree, Random Forest, and Support Vector Machines (SVM). Among those, the authors report the best performance for SVM with an accuracy of 80.83 percent and F1-score of 0.84. However, the dataset used for this study contains only 615 instances and 13 features, limiting the ability to generalize. Additionally, the time of data collection remains unknown.

In contrast, Nalawade et al. (2022) also predict loan approval using the same dataset employed Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, as well as K-NN and come to the conclusion that Logistic Regression performs best with an accuracy of 88.7 percent. However, the F1-score or similar metric is not provided, making a comparison to Kandula et al. (2024) difficult. Due to the small dataset, the limitations of that study remain the same.

Furthermore, Ndayisenga (2021) incorporates other methodologies, including Gradient Boosting, and Gaussian Gradient Boosting. However, its scope is limited to a single banking institution. Additionally, while the work primarily provides theoretical explanations for model variances, it lacks in-depth exploration of any of its individual models.

A more thorough research considering loan default was conducted by Uddin et al. (2023). In their study, 12 models including selected neural networks were tested. Additionally, ensembles of all of them as well as only the best three were used by employing majority voting. The voting ensemble consisting of the top three models showed the best performance, followed by Random Forest. The neural networks, including a dense neural network, showed worse performance.

In summary, the related work shows inconsistent results. There is not a single model that always performs best. Thus, further research is valuable to get a better understanding which models should be applied in which situations.

3 Methodology

3.1 Dataset Description

The dataset used in this study is obtained from the Home Mortgage Disclosure Act (HMDA) database, which is publicly available and maintained by the Federal Financial Institutions Examination Council (FFIEC). That is, the analysis is based on publicly available home mortgages in the United States. The data was accessed via the FFIEC's online platform. The study uses the most recent publication, which is 2022. The dataset comprises approximately 16.1 million rows and 99 features. The different data types of the features can be summarized in figure 1.

The target variable for this study is the binary decision of the home mortgage application, indicating whether the application was approved or denied. The different denial reasons or acceptance

Feature Type	Count	Percentage
Integer	40	40%
Float	16	16%
Categorical	43	44%

Figure 1: Feature Types and their Counts and Percentages

of the original dataset are consolidated into a binary feature, stating that the application was either accepted or rejected. This leads to a distribution of 81.05 percent approved cases, while 18.95 percent of cases are rejected. Thus, the classes show to be moderately imbalanced. As no strong class imbalance can be observed, the decision was made against oversampling techniques, such as SMOTE, as well as undersampling techniques. However, due to the skewed distribution of the target variable, accuracy scores alone may not provide a comprehensive evaluation of model performance and must be interpreted with caution.

3.2 EDA and Data Preprocessing

To provide the algorithms with accurate data inputs, it was first necessary to perform an Exploratory Data Analysis (EDA). This process allowed for a comprehensive understanding of the data's composition, structure, patterns, and outliers.

To begin with, it was verified that the values in our data correctly matched the website's dictionary description for each feature. Even though most of the 16 million data points are consistent with the data dictionary, rows with invalid entries were removed in order to work with accurate data.

Each of the 99 features is explored to understand their value distribution, data types, and address any missing values. Some features, initially loaded as object types, are converted to categorical, integer, and float types to accurately represent their values. Most relevant features have less than 1% missing values, except for *debt_to_income_ratio* and *combined_loan_to_value_ratio*, which have approximately 30% missing data. Since the missing values are not correlated with the target variable, instances with missing values for these two features are dropped to maintain an accurate and clean dataset. Additionally, features like *applicant_race_5* are not mandatory in loan application forms, explaining their missing values. These features are excluded from the analysis as they are not relevant.

To ensure data integrity, numerical values are examined for outliers. Although most data is accurate, anomalies are found in the variable *tract_median_age_of_housing_unit*, which records the median age of housing units in an area. This feature contains 152 negative values, which are removed from the dataset to address this issue.

Furthermore, the EDA reveals an almost perfect correlation between the feature *action_taken* and the outcome variable. This feature represents the status of the loan application as determined by the banking institution. To avoid introducing bias into our model, this feature is dropped. Additionally, if any feature contains the value ’1111’, it is likely that other features, including the outcome variable, also contain this value. Due to a lack of contextual understanding of these exemptions and because the models do not aim to predict them, instances with this value are excluded from the analysis.

Ultimately, the final dataset contains 57 features and 9.6 million rows, with no missing values.

In alignment with the analysis objectives, the target variable *denial_reason_1* is mapped to binary values: 0 for instances where the loan was denied and 1 for approved cases. This transformed variable serves as the target for the analysis.

To prepare the categorical variables for analysis, different encoding methods are applied. While most features are already numerically encoded, certain variables, such as *state_code*, lack numerical representation. To maintain consistency, unique numerical labels are assigned to each category within these variables through label encoding. This encoding ensures compatibility with the analytical models.

Additionally, one-hot encoding is considered for categorical variables. One-hot encoding creates binary columns for each category of a feature, with a value of 1 indicating the presence of the category and 0 otherwise. This method is effective for models that interpret binary inputs well, although it can increase dimensionality, which may be computationally intensive for large datasets with many categories.

Depending on the model, either label encoding or one-hot encoding is used to ensure optimal performance. This choice was, when applicable, determined during the hyperparameter tuning process to find the best encoding method for each specific model.

3.3 Data Analysis Process

Figure 2 provides an overview of the underlying workflow:

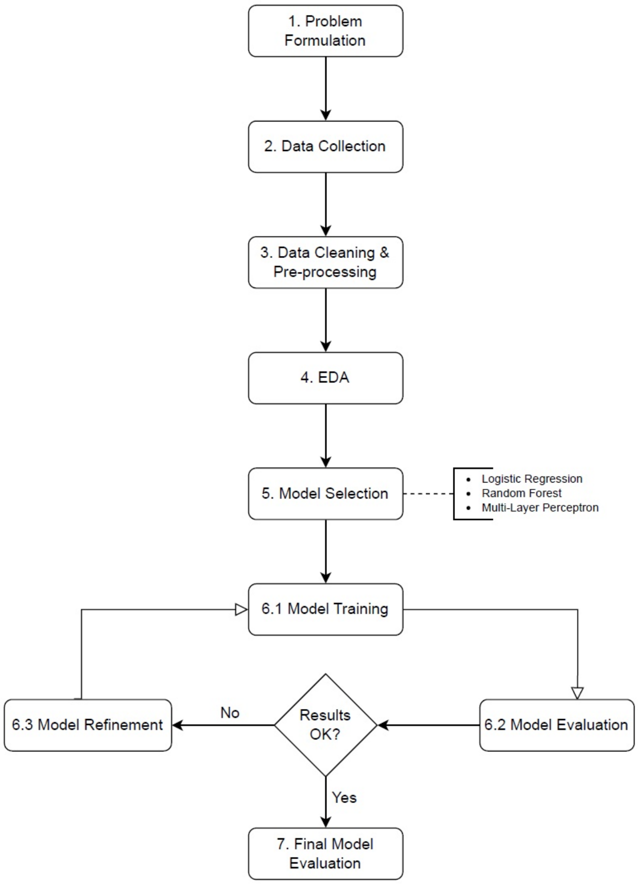

Figure 2: Data Analysis Pipeline

3.4 Feature Importance

After preprocessing the dataset, the number of features was reduced to 57. Given the large number of rows (9.6 million), further dimensionality reduction was performed to reduce the risk of overfitting and enhance the computational speed of the models. To rank the most relevant features for the classification task, a random forest with 1000 trees was executed (Annex 1). Using this ranking, another random forest was progressively trained by increasing the number of features and monitoring the resulting accuracy (Figure 3). This analysis reveals that using 25 features provides an optimal balance between running time and dimensionality, as the accuracy decreases by only 0.1% compared to using all features.

Number of Features	Accuracy
1	0.907
6	0.953
11	0.957
16	0.958
21	0.960
26	0.962
31	0.963
36	0.963
41	0.963

Figure 3: Number of Features vs Accuracy

For feature selection, Logistic Regression with L1-regularization, PCA, and Random Forest methods were considered amongst others. In the end, Random Forest was the chosen method for two main reasons. First, our dataset mainly consists of categorical variables encoded as labels. Random Forests handle these well without needing one-hot encoding, which would increase dimensionality. Second, interpretability is crucial. Given that PCA transforms features into components making it difficult to interpret individual feature contributions, the decision is made to use Random Forest to conduct feature importance. This allows the identification and analysis of the impact of each feature, including potential biases, which is particularly important in a loan dataset, where sensitive features need careful consideration.

3.5 Technical Considerations and Data Subsampling

For the running time and performance analysis, the results will be based on the performance of the models on 64 vCPU-core machines with 384 GB of Memory in "Ucloud," a cloud service that offers remote computing power. The models will be trained and developed in Python using the Scikit-learn and TensorFlow packages.

Given the vast amount of data and limited computational resources, the effect of training with varying fractions of the dataset on model accuracy is studied (Figure 4). It can be observed that the difference in accuracy between training with more than 30% of the dataset and using the entire dataset is marginal, with only a 0.3% difference. Consequently, a smaller sample can be used for hyperparameter tuning, depending on the model's runtime. For the final model 50% of the dataset, accounting for 4.8 million rows, is used for the train-test split.

Fraction of the Data	Accuracy
0.01	0.874
0.1	0.885
0.2	0.888
0.4	0.889
0.5	0.889
0.75	0.891
1	0.892

Figure 4: Accuracy with different fractions of the data

3.6 Training Pipeline

The dataset was split with less than 30% used for hyperparameter tuning and 50% for model evaluation. Only the top 25 features were used for the models.

For model evaluation, the best hyperparameters of each different type of model were determined by training on the same 50% of the dataset to enable a fair comparison across the models.

3.7 Models

To predict the approval of loan requests, the following baseline model and three more progressively sophisticated models are used:

1. **Baseline Model:** This simple model always assigns the most common class as the predicted label. With 81% of loans being approved, this baseline model always predicts "approved" and, therefore, achieves an accuracy of around 81%. If no model could surpass the accuracy of this baseline model, the features would have no predictive power in the context of the chosen models and hyperparameters.

2. **Logistic Regression:** A linear model, suitable for large-scale classification, using L1 and L2 regularization (Yuan et al., 2012). Hyperparameter tuning will be conducted for the regularization strength (C) and the encoding of the categorical variables. The solver used for training the model is SAGA (Stochastic Average Gradient Augmented), chosen for its faster performance on larger datasets and support for simultaneous multi-core training.

3. **Random Forest Classifier:** An ensemble method that utilizes multiple decision trees, first introduced by Ho (1995). Each tree is built using a random subset of features and training data, which helps reduce correlation among trees. The model's final prediction is determined through majority voting based on individual tree predictions. This algorithm is particularly good at handling large datasets with many features and noisy data, while also exhibiting robustness against outliers. Furthermore, it facilitates feature selection and can be further enhanced by hyperparameter optimization.

4. **Multi-Layer Perceptron (MLP):** An extension of the single perceptron, solving the XOR classification problem by introducing hidden layers. This fully connected feed-forward network allows to model complex decision boundaries. In order to optimize model performance, the best combination among the number of layers, neurons (TLUs) in each layer, dropout rate to introduce regularization, as well as the learning rate, based on validation accuracy is used. In addition, early stopping is introduced to ensure better convergence.

3.8 Evaluation Metrics

This study uses four evaluation metrics: Accuracy, Precision, Recall, and F1-score. Subsequently, the evaluation metrics are described:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\text{Precision} = \frac{\text{Number of true positive predictions}}{\text{Total number of predicted positives}}$$

$$\text{Recall} = \frac{\text{Number of true positive predictions}}{\text{Total number of actual positives}}$$

$$F_1\ \text{Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$F_{1\text{-macro}} = \frac{1}{N} \sum_{i=1}^{N} F_{1,i}$$

$$F_{1,i} = 2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

Accuracy is a popular evaluation metric. It has a very straightforward interpretation as it gives the percentage of correct predictions. However, as the target shows moderate class imbalance, the accuracy can give a distorted picture of the performance. A model can have high accuracy by predominantly predicting the majority class correctly, while performing poorly on the minority class. Precision and Recall metrics allow for a more detailed evaluation of a model's performance. Precision is the ratio of correctly predicted positive instances to the total predicted positives. Recall is the ratio of correctly predicted positive instances to the total actual positives in the data. The closer both metrics are to 1, the better the model's performance. However, there is a trade-off between them: increasing precision usually decreases recall, and vice versa.

A metric that summarizes both Precision and Recall is the F_β-score This metric provides a quick impression of the overall performance of the model. Since both Precision and Recall are equally important in this study, F1-score is used, which is the harmonic mean of Precision and Recall. This means that the F1-score favors results where Precision and Recall are similar.

Finally, F1-macro-score is used, which combines the average of the F1 scores for each class. This allows for a comprehensive overview of the overall performance.

4 Model Tuning

Logistic Regression: During the hyperparameter tuning process, the performance of both label encoding and one-hot encoding is explored for categorical variables. The models trained with one-hot encoding consistently perform better. As a result, one-hot encoding is selected for the final model.

The hyperparameter tuning, based on Grid Search, indicates that the best-performing model utilizes Lasso Regularization (L1) with a regularization strength of C=1. This is expected because the feature dimensionality increased from 25 to 135 due to the one-hot encoding of categorical variables, which heightened the risk of overfitting. Lasso Regularization effectively mitigates this risk by shrinking the less important feature coefficients to zero, thus enhancing the model's generalizability.

Random Forest: This algorithm is very practical as it does not require normalization or scaling of the data. Moreover, no further encoding of categorical features was needed beyond what was conducted during the preprocessing phase. Moreover, to optimize its performance Randomized Search was conducted by iterating over various parameter distributions to identify the most suitable settings for our model. The best performing model is achieved with 1000 estimators (trees). Each tree is assembled with a max_depth (or decision points) of 30, and a min_sample_split of 6. Entropy was the best performing measure used to evaluate the quality of a split (criterion). These hyperparameters improve the model's balance between complexity and generalization.

Multi-Layer Perceptron: The best hyperparameters for the MLP reveal an architecture of an input layer with 135 neurons, one for each predictor, followed by a Batch Normalization layer to scale the inputs and ensure easier convergence. This was followed by a series of hidden layers, all using the ReLU activation function. The architecture can be seen in Figure 5. The number of layers (2 to 5), the number of units in each layer (32 to 256), and the dropout layers (from 0% to 50%) were tuned through hyperparameter tuning using RandomSearch. Throughout the entire neural network, the Adam optimizer was used with varying learning rates, and binary cross-entropy was used as the loss function. The output layer consisted of a single neuron with a sigmoid activation function to produce a binary result.

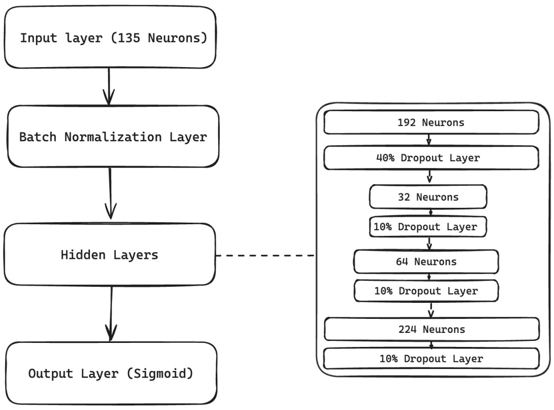

Figure 5: MLP Architecture

5 Results

After selecting the best-performing hyperparameters for each model, they are retrained on the same 50% random sample of the dataset and tested on the remaining fraction. The following metrics were obtained from the results on the testing dataset:

Metric	Baseline	LR	RF	MLP
Accuracy	0.805	0.956	**0.961**	0.959
F1-Macro	0.446	0.934	**0.941**	0.937
Precision (1)	0.805	0.988	0.990	0.988
Recall (1)	1.000	0.957	0.962	0.961
F1-score (1)	0.892	0.972	0.976	0.974
Precision (0)	0.000	0.844	0.858	0.856
Recall (0)	0.000	0.952	0.960	0.949
F1-score (0)	0.000	0.895	0.906	0.901
Run Time (s.)	0.000	11,207	322	18,120

Figure 6: Performance of Logistic Regression (LR), Random Forest (RF) and Multi-layer Perceptron (MLP)

The Random Forest model shows the best performance with an accuracy of 96.14%, followed closely by the Multi-Layer Perceptron (95.92%) and Logistic Regression (95.64%). In terms of F1-score, Random Forest performs slightly better than the other two models with a score of 0.94 for both target labels, indicating that it balanced precision and recall effectively without sacrificing accuracy. Regarding running time, Random Forest is the fastest, completing the task in 5.5 minutes, compared to 3 hours for Logistic Regression and over 5 hours for the MLP.

6 Discussion

6.1 Model Comparison

Since all models show strong performance, the runtime of the algorithms, interpretability, as well as the complexity are also important when selecting the best model. Figure 7 summarizes the different dimensions for each of the models:

Criterion	LR	RF	MLP
Performance	Worst	Best	Mid
Complexity	Low	Medium	High
Interpretability	High	Medium	Low
Runtime	Medium	Short	Long

Figure 7: Model Comparison

As the table shows, Random Forest not only demonstrates the best performance among the models but also scores with medium complexity and interpretability. First of all, Random Forest can be considered medium complexity as it provides a good trade-off between performance and computational efficiency. The ability to derive feature importance and visualize decision paths for individual trees offers a moderate level of interpretability for Random Forests. However, as Géron (2019) points out, Random Forests can be considered black box models due to the difficulty in understanding predictions from a vast ensemble of trees. Logistic Regression provides the best interpretability among the models, as it allows to obtain probabilities.

Further, the runtime was by far the shortest. Random Forest trained 56 times faster than the Multi-Layer Perceptron and 35 times faster than Logistic regression. This is particularly important, as the models might need to be retrained when there is a shift in the decision boundary, for example, when new credit scoring procedures are introduced. These results are contrary to the expectations that Logistic Regression trains the fastest. A possible explanation for the longer training time of the Logistic Regression model could be the use of L1-regularization, which might slow down convergence compared to L2-regularization (Yuan et al., 2012). However, L1-regularization shows the best performance. The Multi-Layer Perceptron takes the longest training time, which is further extended by the use of dropout layers. Srivastava et al. (2014) concluded that a dropout network typically increases training time by a factor of two to three compared to a standard neural network of the same architecture.

It is important to mention that the Multi-Layer Perceptron yields high performance after one epoch already and quickly overfits. This is likely due to the fact that the final training was conducted with 4.8 million instances. Thus, the model is able to adapt to a complex decision boundary in one epoch.

In summary, incorporating all aspects of model choice, Random Forest might be the best model to choose, as it combines a very good performance with medium complexity and interpretability while having a short training time. In addition, Random Forest is an established algorithm, which is beneficial in stakeholder communication.

6.2 Ethical Considerations

Ethical considerations have a high priority in the development of the models. This was enforced by ensuring no ethically questionable features such as race or ethnicity are included in the models.

Ensuring transparency in our modeling process is crucial, as it allows stakeholders to understand the decision-making mechanisms, fostering accountability and trust in the outcomes. The report achieves this by offering a comprehensive description of our procedures and providing access to the code.

Moreover, the utilization of data inputs from

2022 reduces the potential of historical data bias, which may otherwise lead to systemic biases related to race, gender, or socioeconomic status. Our model is designed to complement existing procedures, adding an additional layer of reliability and accuracy to the already existing approval decisions. However, it must be kept in mind that potential biases of previous decisions get reinforced, as a potentially biased decision boundary is used to assist in the decision-making.

This study adheres to the four key principles of Robustness, Interpretability, Controllability, and Ethicality, collectively known as the RICE Framework (Ji et al., 2023). Ensuring ethical considerations in our machine learning models, the RICE framework is applied. Robustness is achieved by training on diverse data and regular updates. Interpretability is ensured through feature importance analysis and clear documentation. Controllability allows for expert review and feedback mechanisms. Ethicality is maintained by not using discriminatory features, as well as complying with data privacy regulations. This approach fosters trust and accountability in the models.

7 Limitations

There are some limitations to the results of this study. First of all, the dataset used comprises a high number of banks, savings associations, and credit unions. That means, when deploying the proposed model, this would lead to a comparison of the application decision versus the market. However, as companies have individual risk profiles, different loan applications get accepted, and consequently the decision boundary is likely to be different among them. Thus, to employ such an additional validation layer, company-specific data should be used.

Another limitation is that the models were trained on the application decisions in 2022. If the approval procedure changes, for example when new credit scoring procedures are introduced, the decision boundary will shift, and the models likely need to be retrained with new data.

In addition, the dataset used contains U.S. home mortgages applications. This limits the generalizability in two ways: First, this only considers the U.S. market and the models might not be applicable outside of this market. Second, it remains unclear whether the results only apply to home mortgages, or also other types of loans.

8 Conclusion and Future Work

This study applied Logistic Regression, Random Forest, as well as a Multi-Layer Perceptron to predict the approval of home mortgages. The finding shows the best performing algorithm is the Random Forest with an F1-macro-score of 0.941, closely followed by the Multi-Layer Perceptron with 0.937. Logistic Regression, although the least performant of the models, still achieved a respectable score of 0.934. Considering the complexity and training time of the models, the most promising model is the Random Forest. It balances the trade-off between performance, training time, and complexity the best. This makes it suitable as a validation layer in the approval process, while being moderately complex, which is crucial for communication with stakeholders.

The findings of this study show that all chosen models perform well when predicting the approval decision. Additional research is needed to understand if other models might perform even better. For example, XGBoost, Support Vector Machines, or K-Nearest Neighbors could be included in future research. Further, future research could extend the prediction to a multiclass classification, not only predicting whether an application gets accepted, but also the reason for a denial. Also, company specific data should be used to further understand the applicability on a company level. Finally, future research should focus on other types of loans as well as other markets.

References

Géron, A.
(2019). *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.). O'Reilly Media.

Ho, Tin Kam (Ed.)
(1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*. `https://doi.org/10.1109/ICDAR.1995.598994`.

Ji, Jiaming; Qiu, Tianyi; Chen, Boyuan; Zhang, Borong; Lou, Hantao; Wang, Kaile et al.
(2023). AI Alignment: A Comprehensive Survey. Available online at `https://arxiv.org/pdf/2310.19852`, checked on 5/16/2024.

Kandula, Ashok Reddy; Divya, K.; Movva, Sri Rama Krishna Datta; Motineni, Mohan Sai; Pappala, Harika; Jakkireddy, Karthika Venkata Satish Reddy (Eds.)
(2024). Comparative Analysis for Loan Approval Prediction System Using Machine Learning Algorithms. *Proceedings of Fifth International Conference on Computer and Communication Technologies*. IC3T 2023 (897).

Nalawade, Shubham; Andhe, Suraj; Parab, Siddhesh; Sankhe, Amruta
(2022). Loan Approval Prediction. In *International Research Journal of Engineering and Technology (IRJET)* 09, Article 04, pp. 669–673. `https://doi.org/10.1109/ICESC48915.2020.9155614`.

Ndayisenga, Theoneste.
(2021). Predicting loan approval using machine learning techniques. *Journal of Financial Technology and Data Science*. Available online at `https://example.com/doi/10.1234/jftds.2021.0123`.

Srivastava, Nitish; Hinton, Geoffrey; Krizhevsky, Alex; Sutskever, Ilya; Salakhutdinov, Ruslan.
(2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 15. 1929-1958.

Uddin, Nazim; Uddin Ahamed, Md. Khabir; Uddin, Md Ashraf; Islam, Md. Manwarul; Talukder, Md. Alamin; Aryal, Sunil
(2023). An ensemble machine learning based bank loan approval predictions system with a smart application. In *International Journal of Cognitive Computing in Engineering* 4, pp. 327–339. `https://doi.org/10.1016/j.ijcce.2023.09.001`.

Yuan, G.-X.; Ho, C.-H.; Lin, C.-J.
(2012). Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9), 2584-2603. Available online at `http://www.csie.ntu.edu.tw/~cjlin/papers/survey-linear.pdf`.

9 Annexes

Rank	Feature
1	Hoepa Status
2	Purchaser Type
3	Debt to Income Ratio
4	Initially Payable to Institution
5	Occupancy Type
6	Income
7	Combined Loan to Value Ratio
8	Loan Purpose
9	Aus 1
10	Tract Minority Population Percent
11	Loan Amount
12	Property Value
13	Tract to MSA Income Percentage
14	Tract Population
15	Tract One to Four Family Homes
16	Tract Owner Occupied Units
17	FFIEC MSA MD Median Family Income
18	Tract Median Age of Housing Units
19	State Code
20	Derived Loan Product Type
21	Applicant Credit Score Type
22	Loan Term
23	Applicant Age
24	Business or Commercial Purpose
25	Loan Type

Figure Annex 1: Feature Importance

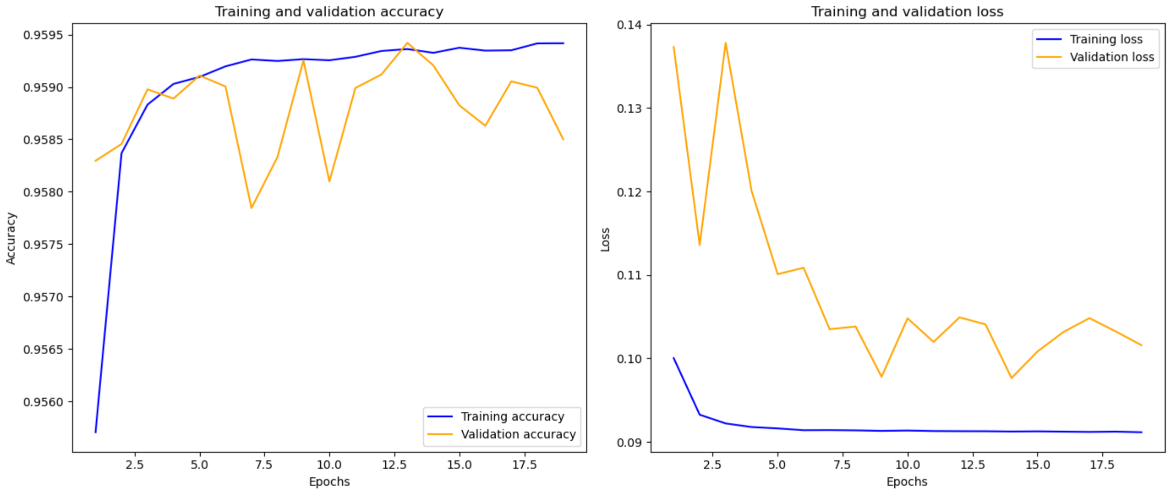

Figure Annex 2: MLP Training Evaluation