ECS171 Group Project: Loan Prediction Problem

https://github.com/jeahernandez/Loan-Prediction-Project.git

Xiao-Bao Bao, Jennifer Hernandez, Travis Li, Han Wang

July 27, 2021

1 Introduction

The empirical modeling of finance and credit underwriting have historically been items of interest to both economists and researchers. Today, commercial lending towards consumer credit and loan analysis is a multi-billion dollar industry. Consequently, large investment firms and securities organizations have expended millions of dollars towards developing high accuracy scoring models in order to optimize profits as well as mitigate risk [6]. These financial institutions have often relied upon a combination of subject matter experts, statistical models, and logistic regression [9].

The history of credit scoring for loan default prediction goes back to the 1950s. Legacy models for early credit decision-making was largely judgment based; relying upon broad heuristics such as character, level of capital requisitioned, quantity of collateral offered, and market conditions. However, models suffered from being unable to process large volumes of applications. In addition, a scarcity of available financial records would make the task difficult [11]. Thus, underwriting for most loans required several months of training and expertise from senior underwriters since the task required the accurate analysis of affordability, repayment history, collateral, regulatory compliance standards, and investor requirements [10].

In modern financing, data requirements vary across institutions; but typically consists of demographic information, bureau information, and finally target flag variables. For instance, if a borrower has missed 90 days on missed payments, they are automatically flagged as a "bad" borrower. A statistical model would aggregate these heuristics into a singular credit score that is then applied to a predictive process known as reject inference. Typical recent models might utilize logistic regression for its simplicity and transparency [4].

Today, machine learning has disrupted and accelerated the efficiency with which these institutions are able to process data on their customers [1]. These advancements in data analytics, computer processing power, and data science research have enabled more accurate models to become employed at enterprise scale. That being said, the most common challenge in leveraging machine learning is the balance between interpretability and accuracy [7]. While Deep Neural Network (DNN) models are capable of producing highly accurate predictions, since regulatory bodies require that borrowers have a right to understand the circumstances of their loan review. As a result, these "black box" algorithms are prohibitively expensive to decipher [6].

In this paper we will investigate two different methods of performing loan default classification and attempt to reproduce findings from literature. Data from a small existing sample of categorical and numerical features is subject to rudimentary preprocessing and classification from the sci-kit learn library. The two classifiers we have chosen to implement are Decision Trees and Multi-layer Perceptron Neural Networks (MLPs). We analyze the performance of these classifiers on such a small sample set in order to characterize the veracity of implementing MLPs and Decision Trees with our dataset. Additionally, we investigate different sampling techniques and observe variance and correlation in the dataset through Pairplots. Finally, we evaluate and present metrics that verify the performances of both classifier types.

2 Literature Review

Dastile et al. [4] surveyed and found that most statistical techniques involved linear regression and naive bayes while common machine learning classifiers included k-Nearest Neighbor, Deep Multi-Layer Perceptron, Convolutional Neural Networks (CNN), Random Forests, Artificial Neural Networks (ANN), and Decision Trees, to say the least [5]. Ultimately, credit scoring is generally regarded as a supervised learning problem in which binary classification produces a delineation between good and bad borrowers.

Financial service corporations have invested significant sums of funding into machine learning products and services. According to Lee et al. [7], large financing firms such as Allstate, ING, and Mizuho have already been active in deploying models with high interpretability. In 2017, Allstate deployed Amelia, a customer-service chatbot developed by IPsoft, that could generate solutions and quotes to customer inquiries through a combination of Natural Language Processing NLP, machine learning models, and data analytics. Meanwhile, ING, the Dutch multi-national banking institution, teamed up with Google and PwC in 2018 to develop a machine learning early warning system EWS. This architecture scans and analyzes large financial data and non-financial data, such as news, to detect potential customer exposure to credit-risk.

Lessmann et al. [8] found from a comprehensive study that a large number of classifiers outperformed logistic regression, the industry standard. They also found that both random forests and ANNs achieved large cost reductions when the cost of classification errors was analyzed. In addition, Chen and Guestrin proposed utilizing extreme gradient boosting or XGBoost for training an ensemble of decision trees [3]. As mentioned earlier, Lessmann et al. [8] found that ANNs performed quite high with broad capabilities of accurately fitting any problem can resolve within compute-space. Gunnarsson et al. [6] built upon the applicability of deep learning by expanding upon the current state-of-the-art by comparing deep learning architectures with conventional and ensemble classifiers. They also evaluated the performance of these classifiers across several indicators and found that ensemble XGBoost significantly outperformed all other classifiers. Interestingly, Gunnarsson et al. [6] also found that DNNs do not always outperform shallower networks with one hidden layer due to the high computational costs when complexity is increased. However, if interpretability of a model's prediction is more important than predictive performance, resorting to conventional Bayesian statistical models and logistic regression have proven to be equally as suitable.

Attribute	Values	Description
Loan ID	LP[6 digit #]	The applicant's loan ID number
Gender	Male, Female, [blank]	The applicant's sex. Blank values could refer to individuals that do not identify as male or female.
Married	Yes, No	The applicant's marital status. Yes if they are in a legally formed union or no if they are not.
Dependents	0, 1,2, 3+	Number of dependents that applicant has.
Education	Graduate, Not Graduate	Whether or not the applicant is a high school graduate.
Self Employed	No, Yes, [blank]	Whether or not the applicant is self-employed. Blank values refers to individuals that do not identify as working for an employer or for themself.
Applicant Income	numerical	Annual income of the applicant in thousands.
Co Applicant Income	numerical	Co-applicant's annual income in thousands.
Loan Amount	numerical	Money, in thousands, the applicant is seeking to loan.
Loan Amount Term	12, 36, 60, 84, 120, 180, 360, 480, [blank]	The repayment period for the loan in months. Blank values refer to applicant's that were unsure of which term they wished to get
Credit History	0, 1, [blank]	Whether or not the applicant has a credit history. 0 represents lack of a history, 1 is an established record of credit, and blank is unsure.
Property Area	Urban, Rural, Semiurban	Description of the housing area the applicant currently inhabits.
Loan_Status	Y, N	Whether or not the applicant was approved for the loan based on the information provided

Table 1: Attributes along with all their possible values

3 Dataset Description and exploratory data analysis of the dataset

This public dataset comes from a compilation of different datasets. It was made available for machine learning applications by the GeeksforGeeks data repository. Aside from the owner of the dataset, Debdatta Chatterjee [2] an associate analyst at Clarivate Analytics, we unfortunately do not possess any information on how the data was aggregated or how its numerical and categorical attributes were originally defined. Therefore, several assumptions had to be made when interpreting the data; particularly due to the fact that several of these attributes contained empty sample cells. Table 1 describes these attributes, along with interpretations of their possible values. As mentioned before, there are several rows in this dataset that contain empty values. In order to remedy the limited nature of the data, we attempted to ascertain correlations amongst attributes by using Pairplot in order to analyze the data and possibly combine categorical attributes such as Gender and Married. Unfortunately, from Figure 3, we determined that the pairplots we created could not reveal substantial or significant correlation between any of the attributes; a point of concern which we will explain in further sections. Ultimately, we decided to omit all of the rows that possessed missing cell data as the percentage of missing data was still reasonably small. Imputation was not deemed completely necessary since the deletion still left roughly 78.18%

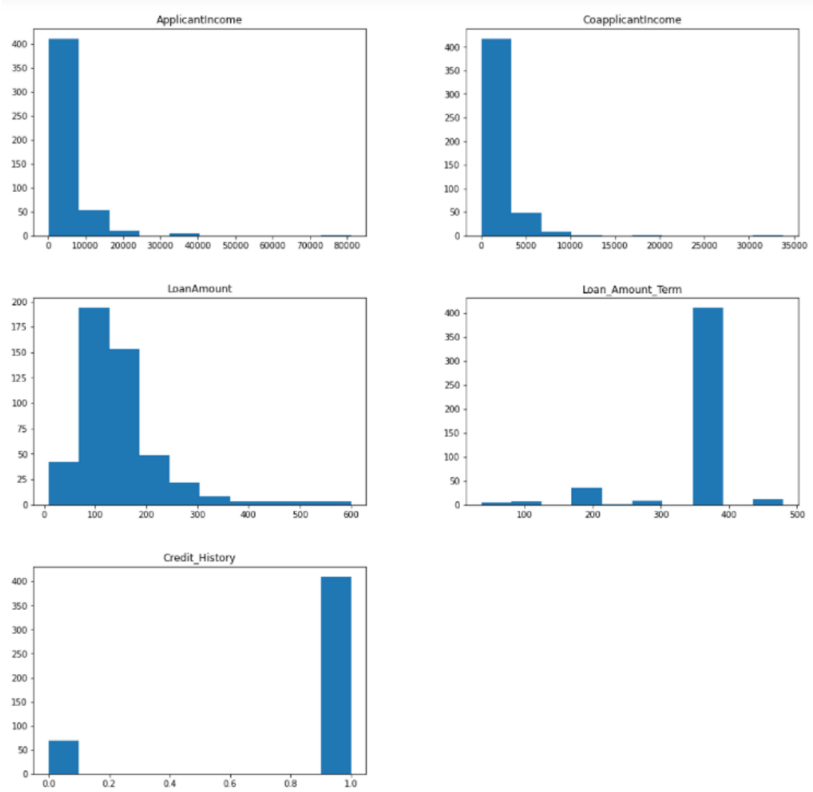


Figure 1: Distribution of data attributes

	Gender	Married	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
Gender	1.000	0.327	0.026	-0.038	0.093	0.128	0.139	-0.087	-0.008
Married	0.327	1.000	0.000	-0.036	0.054	0.111	0.170	-0.109	0.008
Education	0.026	0.000	1.000	0.010	-0.131	-0.066	-0.171	-0.085	-0.057
Self_Employed	-0.038	-0.036	0.010	1.000	0.109	-0.020	0.064	-0.038	0.043
ApplicantIncome	0.093	0.054	-0.131	0.109	1.000	-0.124	0.570	-0.063	-0.029
CoapplicantIncome	0.128	0.111	-0.066	-0.020	-0.124	1.000	0.157	0.002	-0.012
LoanAmount	0.139	0.170	-0.171	0.064	0.570	0.157	1.000	0.022	-0.021
Loan_Amount_Term	-0.087	-0.109	-0.085	-0.038	-0.063	0.002	0.022	1.000	0.024
Credit_History	-0.008	0.008	-0.057	0.043	-0.029	-0.012	-0.021	0.024	1.000
Property_Area	-0.012	0.027	-0.076	-0.068	-0.003	0.002	-0.072	-0.074	-0.001
Loan_Status	0.034	0.098	-0.067	0.041	-0.008	-0.045	-0.038	-0.020	0.535

Figure 2: Possibilities of correlations

of the data. More importantly, undersampling our data would reinforce our qualitative investigation of the performance of classification models with smaller datasets since we are focused purely on the capabilities of such models in this scenario. The dimensions of the data changed from (614, 13) to (480,13). The attributes that are numerical tend to be skewed to certain ranges of values; as can seen in Figure 1, where the x-axis represents our range of possible values and the y-axis shows the number of applicants. Unfortunately, observations of possible correlations between various attributes and their loan status target feature was already shown have been too negligibly small to warrant combining attributes. However, Figure 2 shows that applicants have a 1/2 chance of being accepted for the loan if they already have a credit history and people typically only loan as much as their income. A

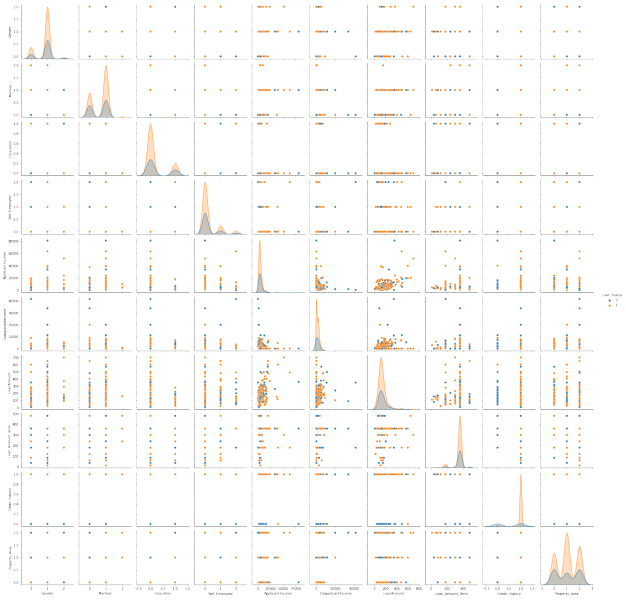


Figure 3: Pairplots

pairplot with the target class, "Loan Status", (Figure 3) reflected this correlation among the attributes and showed that there was virtually no visible correlation between any pair of attributes; regardless of whether the attribute contained numerical or categorical inputs.

4 Proposed Methodology

4.1 Multi-layer Perceptron

Additional preprocessing was performed by first splitting the dataset between it's categorical and numerical features. The categorical features that denoted discrete data such as the marital status, educational level, and employment status were subject to scaling. These categorical features were subject to one hot encoding in order to represent the feature set as numerical integers. This was performed by one hot encoding each individual categorical feature and then aggregating into a single DataFrame that represented the complete set of features. The numerical features were pulled into a separate matrix of values and normalized via Min-Max scaling. These independent feature matrices were then concatenated in a single dataset with fifteen columns. Finally, since the target dataset was a boolean-valued output, we one-hot encoded the target set through simple binarization of the Series frame. After cleaning up the dataset and removing all the applicants that left blank information, we have to do some data preprocessing. All the numerical data has to be scaled which we did with the default minmax scaler in the sklearn library. Then we one-hot encode all the categorical attributes (married, gender, dependents, education, self-employed, property area, and loan status). Then we combine and shuffle all the attributes to become X and make Y equal to the loan status since that is what we are trying to predict. The data then gets split 90:10 for the training and testing set. We then run a grid search in order to determine the optimal number of layers, learning rate, and number of epochs to avoid overfitting.

4.2 Decision Tree

Besides the Neural Network, we also leverage Decision Tree to conduct the loan status classification and prediction. Similar to Neural Network, preprocessing is conducted before the training and testing. To achieve this, we remove all the NA values and leverage e one-hot encode all the categorical attributes (married, gender, dependents, education, self-employed, and property area). The whole dataset is split into 80:20 for training and testing. To better understand the impact of Decision Tree' depth on the classification results, we set the depth to 2/3/4/nolimit as shown in following Figure 6 and Figure 7. The results are presented in confusion matrix format in heatmap. In this way, we will analyze whether over-fitting or under-fitting happens by comparing training and testing results.

5 Experiment Results

5.1 Multi-layer Perceptron

We combined and shuffled the input features and target values in order to avoid overfitting. The data was then split to a ratio of 80:20 for the training and testing sets, respectively.

By invoking the MLPClassifier library, a simple multi-layer perceptron was used to fit and classify the small sample set. The number of nodes in the model's input layer was fifteen due to dimensioality of the numerical and categorical training feature set while the number of hidden layers used was 2; with initially fifty nodes in the first hidden layer and ten in the second layer. We set the classification modality to Stochastic Gradient Descent with a learning rate of 0.1 and batch size of 10. We also regularized the classifier with the L2 expression by setting the coefficient's alpha value to 0.003 in order to reduce the high degree of overfitting found in preliminary network models. The activation function chosen for the final MLP was with Rectified Linear Units (ReLU) due to ease of training and slightly better performance metric than the Tanh function. The Tanh activated MLP produced an accuracy of roughly 73.958% while the ReLU activated MLP produced an accuracy of 78.125%.

In order to optimize the number of layers, learning rate, and number of epochs, we performed an exhaustive hyper-parameter sweep in order to avoid overfitting. The method of brutal search chosen was a comprehensive Grid Search. The hyper-parameters that we swept through in the grid search included: the number of nodes in both hidden layers, the learning rate, and the maximum number of iterations. Afterwards, we ran the MLP classifier on the data using the optimal hyper-parameters found from this search. These final hyper-parameters include a learning rate of 0.05, as well as 60 nodes for the first hidden layer and 10 nodes for the second layer. The maximum number of iterations before convergence was found to be 10 iterations. Lastly, we checked the accuracy of the model by comparing the ReLU and tanh activation functions before determining that the final model would use ReLU activation.

We found that our classifier produced an accuracy of roughly 0.78125 and mean squared error of 0.21875. Our classification reports and confusion matrices of our prediction results also concluded that the model produced an average precision of 0.84, recall of 0.78, and an F1-score of 0.75. Both the tanh and ReLU activated classifiers produced low recall scores. Our

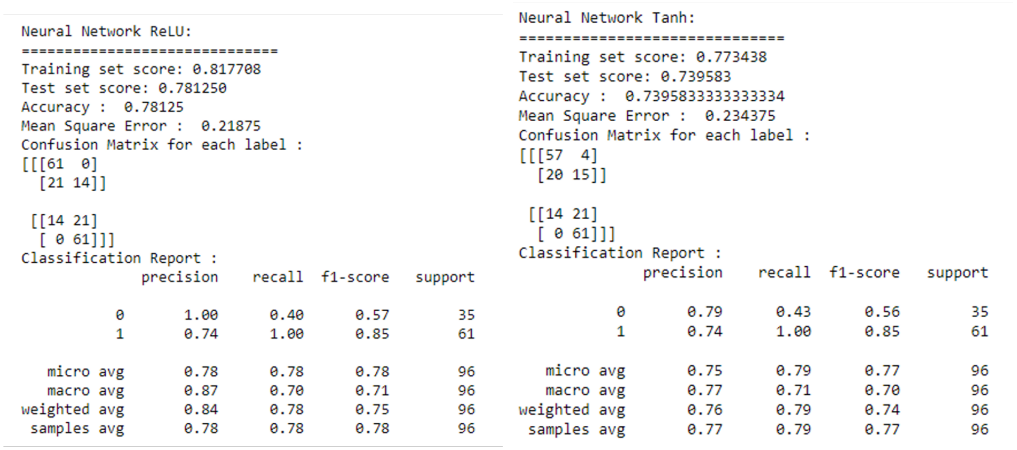
MLP accuracy score is consistent with experimental results from Moscota et. al., who's best performing MLP model produced an accuracy of 0.771. Of course, we skewed our classifier with an undersampled dataset; which is the likely explanation for the higher accuracy value.

ACTIV	ACC	MSE	PREC	RECALL	F1-SCORE	SUPPORT
ReLU	0.78125	0.21875	0.78	0.78	0.78	96
Tanh	0.73958	0.23437	0.77	0.79	0.77	96

Table 2: Result comparison between ReLu and Tanh

Classifier	AUC	TPR	TNR	FP-Rate	G-Mean	ACC
MLP	0.704	0.990	0.040	0.945	0.2060	0.771

Table 3: Classification results from Moscota et. al.



```
Neural Network ReLU:
==============================
Training set score: 0.817708
Test set score: 0.781250
Accuracy :  0.78125
Mean Square Error :  0.21875
Confusion Matrix for each label :
[[[61  0]
  [21 14]]

 [[14 21]
  [ 0 61]]]
Classification Report :
              precision    recall  f1-score   support

           0       1.00      0.40      0.57        35
           1       0.74      1.00      0.85        61

   micro avg       0.78      0.78      0.78        96
   macro avg       0.87      0.70      0.71        96
weighted avg       0.84      0.78      0.75        96
 samples avg       0.78      0.78      0.78        96
```

```
Neural Network Tanh:
==============================
Training set score: 0.773438
Test set score: 0.739583
Accuracy :  0.7395833333333334
Mean Square Error :  0.234375
Confusion Matrix for each label :
[[[57  4]
  [20 15]]

 [[14 21]
  [ 0 61]]]
Classification Report :
              precision    recall  f1-score   support

           0       0.79      0.43      0.56        35
           1       0.74      1.00      0.85        61

   micro avg       0.75      0.79      0.77        96
   macro avg       0.77      0.71      0.70        96
weighted avg       0.76      0.79      0.74        96
 samples avg       0.77      0.79      0.77        96
```

Figure 4: Comparison between ReLu and Tanh

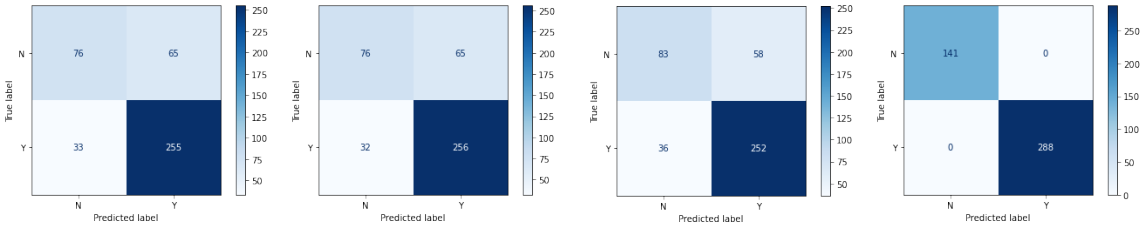
```
Optimal Hyper-parameters :  {'hidden_layer_sizes': (60, 10), 'learning_rate_ini
t': 0.05, 'max_iter': 10}
Optimal Accuracy :  0.8083333333333332
Optimal Hyper-parameters :  {'hidden_layer_sizes': (60, 10), 'learning_rate_ini
t': 0.05, 'max_iter': 10}
Optimal Accuracy :  0.8083333333333332
```

Figure 5: Hyper-parameters

5.2 DecisionTree

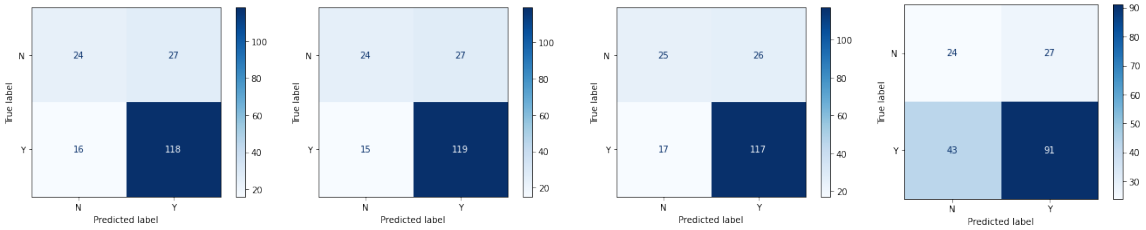
As introduced in Section 4, we also leveraged Decision Tree to conduct the loan status classification and prediction. To better evaluate the effectiveness and influence of depth of

the Decision Tree, we set the depth to 2/3/4/nolimit, where nolimit meant that the nodes were expanded until all of the leaves were pure or until all of the leaves contained less than min_samples_split samples. For training as shown in Figure 6, both false positives and false negatives droped significantly from 58 and 36 to 0 and 0 when we increased the depth of Decision Tree from 4 to nolimit. This highlights the notion that increasing the depth of a Decision Tree can greatly help to improve classification accuracy for the training dataset. Though improvement of the training dataset was given by increasing the depth of the Decision Tree, we were not able to observe similar results for testing the dataset. As shown in Figure 7, changing the depth from 2 to 3 and then 4 did not not impact the classification significantly; while depth without restriction yielded more false negatives with similar false positives. This observation indicates that the no limit of depth creates over-fitting problems with high accuracy for training and low accuracy for testing. Additionally, the optimal accuracy given by the Decision Tree was 76.7% calculated based on: (25+117)/(25+26+17+117) while the least accuracy given by the no-depth limit Decision Tree model, 62.2%, calculated based on (24+91)/(24+27+43+91). Lastly, we observed the 'N' class in load status yielded quite low accuracy no matter what the depth was, only 50% of 'N' samples classified correctly.



a)Depth=2 b)Depth=3 c)Depth=4 d)Nolimit

Figure 6: Training confusion matrix of DecisionTree with different depth setting



a)Depth=2 b)Depth=3 c)Depth=4 d)Nolimit

Figure 7: Testing confusion matrix of DecisionTree with different depth setting

6 Conclusion and Discussion

Utilizing machine learning to create models that accurately determines whether an applicant's loan is approved or not can provide useful information to all parties on what features can impact their application the most and reduce risk on the firm's part. However, with

uncertainty in the data, it is essential to use different models in order to analyze which ones are more effective and efficient with this problem.

We found that both of the models we created were consistent with the accuracy reports we explored in the literature review. Ultimately we determined that the decision tree model would be more compatible for loan prediction algorithms due to the challenges presented when creating a neural network. Neural network computation tends to be expensive and consumes significant quantities of processing power. Additionally, it would be difficult to scale our current MLP algorithm to include more features.

Future work on these models would seek to improve the pre-processing done on the dataset in order to oversample successfully. While we attempted to perform imputation on the dataset to account for the applicants that left some fields empty via the pairplots, the correlation was too weak to justify feature combination. A possible solution could involve finding the feature's mean value, using that to replace the blanks, and testing these datasets for accuracy until we arrive at an ideal model.

References

[1] Saurabh Arora, Sushant Bindra, Survesh Singh, and Vinay Kumar Nassa. Prediction of credit card defaults through data analysis and machine learning techniques. *Materials Today: Proceedings*, 2021.

[2] Debdatta Chatterjee. Loan prediction problem dataset. *https://www.kaggle.com/altruistdelhite04/loan-prediction-problem-dataset*, 2019.

[3] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 144:113100, 2020.

[4] Xolani Dastile, Turgay Celik, and Moshe Potsane. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, 91:106263, 2020.

[5] Jose Romulo de Castro Vieira, Flavio Barboza, Vinicius Amorim Sobreiro, and Herbert Kimura. Machine learning models for credit analysis improvements: predicting low-income families' default. *Applied Soft Computing*, 83:105640, 2019.

[6] Björn Rafn Gunnarsson, Seppe Vanden Broucke, Bart Baesens, María Óskarsdóttir, and Wilfried Lemahieu. Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, 295(1):292–305, 2021.

[7] In Lee and Yong Jae Shin. Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, 63(2):157–170, 2020.

[8] Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, and Lyn C Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015.

[9] Vincenzo Moscato, Antonio Picariello, and Giancarlo Sperlí. A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, 165:113986, 2021.

[10] Swati Sachan, Jian-Bo Yang, Dong-Ling Xu, David Eraso Benavides, and Yang Li. An explainable ai decision-support-system to automate loan underwriting. *Expert Systems with Applications*, 144:113100, 2020.

[11] Lyn Thomas, Jonathan Crook, and David Edelman. *Credit scoring and its applications*. SIAM, 2017.