**An Analysis of Logistic Regression vs. Artificial Neural Network for Predicting Loan Default**

Katherine Reed

Department of Computer Science

University of South Carolina Upstate

Spartanburg, South Carolina, USA

October 26, 2020

kr16@email.uscupstate.edu

**ABSTRACT**

A significant risk associated with the granting of a monetary loan is the possibility that the full amount of the loan will not be repaid to the lender. In the event that this occurs, the loan goes into default, and this can wreak havoc on the finances of the lender, the borrower, and in extreme situations, the economy as a whole. It is important for the lender to be able to evaluate relevant characteristics about the potential borrower and determine if the borrower is more likely to repay or default on the loan. The purpose of this study is to evaluate the respective ability of two machine learning algorithms, logistic regression and artificial neural networks, to predict the credit risk of a loan applicant. In this study, the algorithms are implemented using various Python machine learning libraries, and they are applied to a Kaggle credit risk dataset that contains information about 39,718 loans that were issued between 2007 and 2011 from Lending Club. The accuracy, precision, recall, and specificity of each algorithm are evaluated and compared to each other with the purpose of determining which algorithm is better suited for creating a model that can predict loan default. The results of this study indicate that the neural network model was superior in terms of accuracy and specificity, while the logistic regression model was better at precision and recall/sensitivity. Therefore, the results of this study do not provide a conclusive answer to the query as each algorithm excelled in different areas of evaluation. A financial institution that uses these results to choose the superior model for making a credit risk decision must decide which evaluation metric is most important in making this decision. However, the results of this study would likely be improved and more conclusive if performed on a larger data set.

**Keywords**

Machine learning, logistic regression, artificial neural networks, credit risk, loan default, peer-to-peer lending

1. **INTRODUCTION**

The United States loan market is a multi-trillion-dollar industry. Its size and significance are bolstered by the fact that a majority of Americans take out a loan from a financial institution at least once in their lives, whether they use it to make a major purchase, like a home or a car, finance their education, pay expensive medical bills, or help them refinance other types of debt, like credit card debt. A 2015 study by the Pew Research Center discovered that 80 percent of Americans possess some level of debt as a result of acquiring a loan, the majority of which is mortgage debt [17]. The statistics on the amount of debt held by Americans are staggering: as of Q3 of 2019, outstanding personal loans totaled $156 billion, $1.2 trillion was owed for auto loans, and mortgage loans amounted to $10.5 trillion [6, 7, 10].

The amount of a loan, in addition to interest, must be repaid by the borrower to the lender within a certain timeframe agreed upon by both parties. If the loan is not repaid in full, then it goes into default, which can have serious negative consequences for both the lender and the borrower. The lending institution suffers financially when they don’t recover the money from a defaulted loan. The borrower of a defaulted loan receives a derogatory mark on their credit report that stays there for at least seven years, which significantly lowers their credit score and adversely affects their chances of receiving loans in the future. In certain extreme situations, defaulted loans affect the American economy as a whole. For example, the Great Recession of 2008 was a direct result of lenders granting many high-risk home loans to borrowers with poor credit histories who were ultimately unable to repay the loan.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Conference’04*, Month 1–2, 2004, City, State, Country.

Copyright 2004 ACM 1-58113-000-0/00/0004…$5.00.

While the risk of default is ever-present, it is in the interest of financial institutions to loan money to qualified borrowers because they make a profit from the interest payments, which can be substantial depending upon the size of the loan and the credit history of the borrower. It is clear though that lenders must exercise extreme caution in determining who is eligible to receive a loan. They must be able to approve the loan application with the confidence that the borrower is capable of repaying the loan and the interest within a reasonable period of time.

Lenders typically determine a person’s loan eligibility by evaluating factors such as the loan amount and purpose, as well as the borrower’s current debt-to-income ratio, employment history, and credit history. They use this information to determine the applicant’s credit risk, which is the potential that the borrower will fail to repay the loan in accordance with terms agreed upon by the borrower and the lender [12]. Their goal is to classify the loan as either a “good” loan, which is one that is likely to be repaid, or a “bad” loan, which is likely to go into default. The loan application review process, known as loan underwriting, can be either of a manual or an automated form. In a manual loan review, the underwriter personally reviews all the details of the loan application and uses their employer’s rules to guide them in rendering a decision to approve or deny the loan request. In contrast, automated loan underwriting involves the use of a computer program to review these factors and generate a decision based on an algorithm [18].

An applicant can benefit from manual underwriting if they, for instance, have complicated income or a poor credit history because they are able to state their case for approval to the underwriter. However, to the detriment of the applicant, the manual process can take several weeks, and the ultimate decision is subject to potential biases held by the underwriter. In addition, the ability of humans to recognize complex relationships or hidden knowledge in historical data is limited in comparison to that of a computer algorithm that is trained to recognize such patterns [3]. Conversely, the automated process is easier to regulate and standardize, and it can render a decision instantly, but the applicant is unable to discuss their application with a person and convince them that they will repay the loan.

While underwriting was originally a manual process, more companies are transitioning to the automated form because it provides a better customer experience by streamlining the application process, it requires less overhead, and it simplifies the decision-making for even the most complicated loan applications [10]. Thus, it is imperative that the model for approving or denying the loan is robust and considers all of the necessary factors in a fair and impartial manner. The model must be able to efficiently and accurately analyze a vast amount of historical data and compare it with the applicant’s data to predict if the applicant is more likely to repay or default on the loan. In doing so, the lender is able to assume the risk of granting the loan with greater confidence that they will be repaid. In addition, the borrower would not be denied a loan that they are capable of repaying.

The increase of computing power, the amount of available data, and the successful use of artificial intelligence as a tool in a variety of industries in recent years has motivated researchers to study the value of applying machine learning methods to financial analysis, including in determining the likelihood of loan default. The strength of machine learning algorithms lies in their exceptional capacity to recognize complex patterns in historical data and use this knowledge to make predictions with new information. The primary purpose of this study is to experiment with the application of two popular machine learning algorithms, logistic regression and artificial neural networks, for their respective abilities to predict the credit risk of a potential borrower. Specifically, the goal is to compare the accuracy, precision, recall, and specificity of the two algorithms for classifying a loan application as a “good” loan or a “bad” loan in order to determine which algorithm provides a better model for making this prediction.

1. **LITERATURE REVIEW**

The use of artificial intelligence in the financial world is not a novel idea. Many studies have been conducted that demonstrate the value of applying machine learning algorithms in financial analysis and prediction.

Serrano-Cinca et al. investigated the various factors that could explain loan default in peer-to-peer (P2P) lending, which involves the lending of money from investors to borrowers through online services without the use of an official financial institution. For these types of loans, the online service provides relevant information about the borrower to the lender, and it assigns a grade to the loan based on the information provided in the loan application. They also developed a logistic regression model to predict the likelihood of loan defaults. From their analysis of 24,449 loans facilitated by the online P2P service Lending Tree, they discovered that the grade assigned to the loan is the best predictor of default and that the logistic regression model’s accuracy increases when it considers other variables, like the purpose of the loan and the borrower’s income, current level of debt, housing situation (rent or own), and credit history [14].

Jepkemei conducted a study on loan default prediction by applying a feed-forward neural network with multiple layers and backpropagation to data of 1000 customers of multiple financial institutions in Kenya. She evaluated the following loan risk factors: applicant age, consumer income, home ownership, number of existing loans, credit rating, the existence of a guarantor, gender, loan size, marital status, number of dependents, and loan purpose. Her experimental model predicted loan default with 99.3% accuracy [6]. Eletter et al. built a similar artificial neural network model to evaluate loan applications using a banking data set of 140 loan applications. Their model was able accurately classify 95% of the test cases as either “good” applications or “bad” applications [6].

Handzic et al. performed a more in-depth analysis of the ability of neural networks to evaluate loan applications. They did so by comparing the accuracy and efficiency of three different types of neural network models, Multi-Layer Perceptron (MLP) and two Committee Machine models called Ensemble Averaging and Boosting by Filtering. The results of their experiment showed that the Committee Machine models outperformed the MLP and that between the two Committee Machine models the Boosting by Filtering model was superior [4].

Zhu et al. applied various machine learning methods, including random forest, decision tree, support vector machine, and logistic regression, to the analysis of 115,000 Q1 2019 loans from the P2P lending service Lending Club. Their research showed that the performance of the random forest model was superior in predicting the likelihood of loan default. Specifically, the random forest model had an accuracy of 98%, compared to 95% accuracy for decision tree, 75% accuracy for support vector machine, and 73% for logistic regression [20].

Studies have also been conducted to evaluate the predictive power of humans. Handzic and Aurum conducted an experiment in which thirty-two subjects acted as the manager of a fake ice cream company. Their task was to forecast the amount of ice cream sold each day for thirty consecutive days based on historical transaction data available in the company’s database. This study found that humans have a reasonably good ability to discover relationships and patterns in data, and they can even improve their ability to make predictions over time. However, the study showed that this predictive power is less than what is theoretically possible based on the actual patterns and relationships present in the data, and the ability of humans to make predictions based on data decreases as the associative patterns among the data points weakens [3]. The results of this study further support the need for a technology-driven approach to credit risk evaluation.

Many models have been proposed for classifying loan applications based on credit risk, but there does not exist a single perfect model for this purpose. The various models must be evaluated and compared to each other in order to determine which model is the best at determining the likelihood of loan default. While studies have been conducted that separately demonstrate the merits of using logistic regression and neural networks in financial analysis and prediction, the strength of this study lies in using multiple metrics to compare these two algorithms in order to determine which algorithm provides a better model for determining credit risk. In addition, while the data set used in this study isn’t the exact data set used in the studies mentioned in this section, the data is derived from the same source that Zhu et al. used for their study, with the difference being that the data in this study covers a different span of time.

1. **METHODOLOGY**
   1. **Logistic Regression**

Logistic regression functions as a binary classifier by using a linear function of the independent predictor variables in the data to estimate the probability that a specific outcome will occur [1]. In this case, that outcome is that a person defaults on a loan. For this model, the independent variables are the details of the loan and the personal financial data that the person provided when they applied for the loan. The dependent variable is the final status of the loan. This variable is binary in nature as it assumes a value of ‘0’ if the loan was paid back in full, and it assumes a value of ‘1’ if the loan went into default.

Because the result of this algorithm is a probability value, the number generated, by definition, must be bounded by 0 and 1, regardless of the values of the intercepts and the predictor variables and their coefficients in the linear function. This makes this algorithm well-suited for solving binary classification problems. If the estimated probability, p, generated by the algorithm is , the probability is resolved to a value of ‘0’. If the estimated probability is , the probability is resolved to a value of ‘1’.

The linear function of the predictor variables is mapped to probability values through the *logit function*, which is also referred to as the *log-odds function* because it sets the linear function of the predictor variables equal to the natural logarithm of the odds [1]. The *logit function* states:

where *p* is the probability, *Xk*are the independent predictor variables, *bk*are the coefficients, and *b0*is the y-intercept. The *logit function* produces values that vary from as the value of the probability varies from [0, 1]:

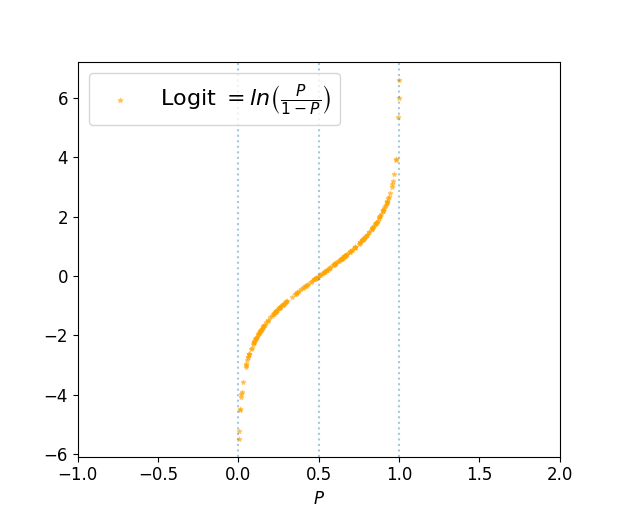


Figure . Example of logit curve

As the graph of an example of a logit function shows, the result of the logit function, and by extension, the linear function of the predictor variables, is plotted as dependent upon the value of *p.* However, the goal of logistic regression is the reverse: to generate values of *p* that are dependent upon the linear function of the predictor variables. In order to use the predictor variables to estimate *p*, the inverse of the logit function, also known as the sigmoid function, must be calculated [1]. The sigmoid function states:

\*

The *sigmoid function* produces probability values that vary from [0, 1] as the linear function of the predictor variables results in values that vary from :

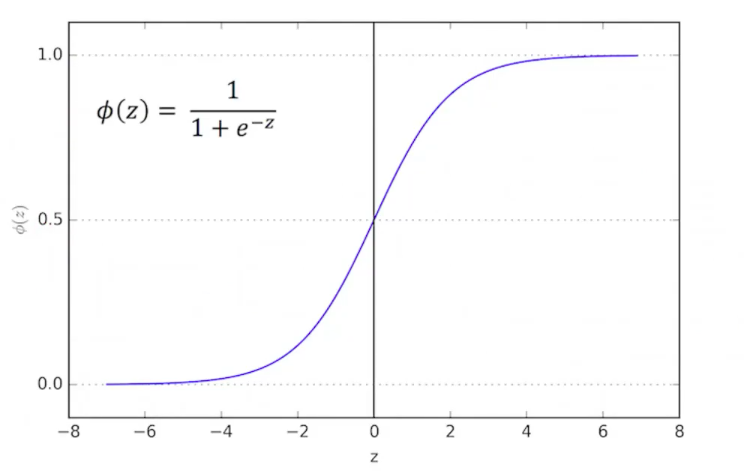


Figure . Example of sigmoid curve

Logistic regression calculates the coefficients and the intercept for the linear function of the independent variables that can be used to produce a curve that best fits the relationship between the independent predictor variables (the financial data) and the binary dependent variable (the loan status) [13].

* 1. **Artificial Neural Network**

Artificial neural networks are computational models that are designed to mimic the structure of biological neural networks found in the brain [7]. The structure of a neural network consists of nodes, commonly referred to as neurons, that are arranged into multiple sequential layers, as shown below:

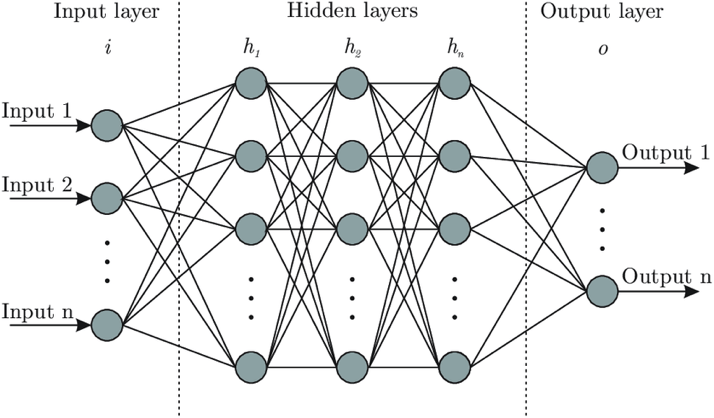


Figure . Neural network structure

The first layer of the network is referred to as the input layer, as it provides the initial data about the independent predictor variables to the network for processing [7]. After the input layer, there are one or more intermediate hidden layers. These layers of neurons are responsible for performing the computations on the data from the input layer and transferring the results of these computations to the final output layer [7]. The output layer, in turn, produces the final prediction generated by the neural network [7].

The neurons in each hidden layer have weighted connections to each neuron in the previous and next layers [7]. The weights associated with these connections are initially set to random values, which are later optimized when the network undergoes training on the data [8]. Each neuron also has an associated bias term that is initially set to zero and is also adjusted during training [5].

For each layer of neurons, a weighted sum of the inputs to the neurons multiplied by the values of the weights associated with the connections between each neuron and all the neurons of the previous layer, plus an additional bias term, is calculated [7]. This weighted sum is computed using the following formula:

where *wm*are the weights, *xm*are the independent predictor variables, *bias* is the bias term, and *m* represents the number of neurons in the layer.

The weighted sum formula clearly demonstrates the necessity of the bias term. This formula represents a linear function, so if a bias term was not included in the calculations, then the function would necessarily be tied to the origin (0, 0). The bias term serves to offset the result and allows the function to have greater flexibility and movement.

Due to the nature of linear functions, the output of the weighted sum formula could be any number in the range . The neural network must be able to determine if this result provides any meaningful value in its ability to make accurate predictions. To do this, the output of the neuron is first passed to a non-linear activation function [16]. There are a variety of non-linear activation functions that can be applied to the output of neuron in a neural network, but they all serve the same purpose. Activation functions rescale the output so that it falls somewhere in a narrow range, typically between (-1, 1) or (0, 1) [16]. They set a specific threshold, whereby if the rescaled output of the function exceeds that value, then that neurons fires a stimulus to the next layer of neurons [16]. In mathematical terms, if is the weighted sum formula of the neuron, and is the activation function, then the final output of the neuron is

These calculations are iteratively performed for each hidden layer of neurons and for the final output layer of neurons. The calculations performed by the final output layer generate the ultimate prediction made by the neural network. After the output layer generates a prediction, the network measures the error rate of the network by computing a loss function for that specific instance [11]. This loss function is calculated by comparing the value predicted by the neural network with the true value in the data and generating a numerical value that indicates the quality of the performance of the network in making that prediction [11]. Finally, a cost function is calculated by measuring the average of the loss function outputs over the entire set of training data [11].

During the training session, the neural network strives to minimize the error rate produced by the loss and cost functions because this signifies that the model has successfully detected patterns present in the data or “learned” how to make more accurate predictions [11]. It accomplishes this through a process called backpropagation. During this process, the weights and biases in the network are adjusted in attempts to achieve the global minimum of the cost function [9]. This process begins with the connections at the end of the network and propagates backwards until the beginning of the network is reached [9].

1. **IMPLEMENTATION**
   1. **Technology**

The code for the implementation of the logistic regression and neural networks was written in Python 3.8 in a Jupyter Notebook. Python has multiple libraries that are designed to help maximize the efficiency of the code in machine learning studies such as this one. For this study, the following Python libraries were used: Pandas for data analysis, NumPy for operations on arrays and matrices, Matplotlib for graphical visualization of the data, Scikit-learn for the logistic regression implementation and metrics evaluation, Imblearn for correcting the imbalance in the data set, and Keras for the neural network implementation.

* 1. **Data**

The data set used in this study was retrieved from Kaggle. It contains loan data for loans issued between 2007 and 2011 from Lending Club, an online peer-to-peer lending club company. The data set contains 39,718 rows/loans and 111 columns/features. The features include relevant details of the loan and information about the financial health of the applicant. The feature *loan\_status* is the dependent variable, so it will be the value that the two algorithms will be attempting to learn how to predict.

**5. EXPERIMENTAL SETUP**

**5.1 Data Pre-Processing**

*5.1.1 Irrelevant columns*

Data pre-processing began with the removal of columns that were considered irrelevant in predicting the probability of loan default. Out of 110 predictor features, 26 features were not considered relevant for this study, including features such as the ID assigned to the borrower, the most recent date a payment was made on the loan, and the total amount of money that has been repaid to-date. These columns were removed from the data set, reducing the number of predictor columns to 84.

*5.1.2 Null values*

Of the remaining 84 predictor columns in the data set, 54 columns contained only Null values. These columns were removed from the data set. In addition, greater than 50% of the values in 2 of the remaining columns were Null values. These 2 columns were also removed from the data set. Finally, of the remaining columns that contain Null values, less than 2% of these columns consist of Null values, so the rows containing these Null values were removed from the data set. At this point, the data set no longer contained any Null values.

*5.1.3 Columns with unique values*

After the Null values were removed from the data, the columns containing only one unique value were removed from the data set. 7 columns in the data contained only one unique value. These columns would not meaningfully contribute to the model, so they were removed from the data set.

*5.1.4 ‘loan\_status’ column*

The *loan\_status* column initially contained 3 distinct values: “Fully Paid”, “Charged Off” (default), and “Current”. Since the final status of the line items with a *loan\_status* of “Current” has not yet been determined, these line items were removed from the data. The logistic regression and neural network algorithms can only operate on numerical data, so *loan\_status* column values were converted to numerical values. The “Fully Paid” status was encoded as a ‘0’ and the “Charged Off” status was encoded as a ‘1’.

*5.1.5 Data format*

There were several columns in the data set that contained data that needed to be converted to a numerical format. There were 4 columns that contained a mix of numbers and text. The unnecessary text was removed from these columns so that only the relevant numbers remained. There was 1 column that contained the date that the applicant first opened a credit line. These dates were converted to a duration as the number of months that have passed since that date.

*5.1.6 Outliers*

In order to view the presence of potential outliers in the data, histograms were created for each column in order to visualize the distribution of the data. From viewing the histograms, it appeared that 13 columns contained outliers towards the upper end of the range of values. The top 3% of values in these columns were removed from the data.

*5.1.7 Categorical variables*

5 of the columns in the data set contained textual categorical data that needed to be converted to numerical data. To accomplish this, dummy variable columns were created for each category present in the categorical columns.

*5.1.8 Train-test split*

The data set was then split into two sets: a training set and a test set. The training set was used to train the two models, and the test set was used to verify how well the two models performed on unseen data. The data was split into 80% for training and 20% for testing.

*5.1.9 Feature standardization*

The scale of the data varied considerably from column to column. This wide difference in scale need to be corrected so that a column with relatively large numbers didn’t have a greater impact on the model than a column with relatively small numbers. To normalize the features, the values were rescaled such that the mean was 0 and the standard deviation was 1 in each column.

*5.1.10 Imbalanced data*

In reality, loan repayment occurs much more frequently than loan default. Because of this, and because this data set was retrieved from a real lending company, the data was initially imbalanced in terms of the number of loans that were categorized as “Fully Paid” compared to the number of loans that were categorized as “Charged Off”. At this point in data pre-processing, there were 14,375 loans that were labeled as “Fully Paid” and only 2,014 loans that were labeled as “Charged Off”. This imbalance needed to be corrected so that the models would not be biased in favor of predicting that the loan will be repaid in full. The data for the training set was balanced through a technique called Synthetic Minority Oversampling Technique, or SMOTE. This technique uses a k-nearest neighbor algorithm to create more line items of the minority class that are similar to the instances of the minority class that are already present in the data [19]. By implementing SMOTE, new line items were created in the training set such that the number of loans that were labeled as “Fully Paid” were equal to the number of loans that were labeled as “Charged Off”.

**5.2 Algorithm Implementation**

*5.2.1 Logistic Regression*

The logistic regression algorithm was implemented using scikit-learn’s logistic regression classifier. An instance of this classifier was created, and then it was fitted to the data in the training set. During the training session, the model was fed instances of loans from the training set that were labeled as either “Fully Paid” (0) or “Charged Off” (1) so that it could try to extract the complex relationships between the independent predictor variables that indicate whether or not a person will default on a loan. After the model was trained, it was tested on the data in the test set. The model was tested by feeding it unlabeled and previously unseen instances of loans from the test set. For each loan instance in the test set, the model had to generate a prediction of either 0 or 1 based on the combination of values for the predictor variables.

*5.2.2 Artificial Neural Network*

The artificial neural network model was created using the Keras library. An instance of a sequential model was created, and the layers were subsequently added to the model. Two hidden layers, each containing 70 neurons, were added to the model. The Rectified Linear Unit (ReLU) function was set as the activation function for each of the hidden layers. The output layer contained one neuron, and the sigmoid function was set as the activation function for this layer. The model was then configured for training by setting the Adam algorithm as the model’s optimizer, binary cross-entropy as the model’s loss function, and accuracy as the model’s metrics. The model was then trained on the data in the training set by using batch processing. During batch processing, the data in the train set is split into batches of equal size. The neural network then makes predictions for each sample in the batch and an error rate is calculated for the entire batch. This process occurs for each batch of data. The data was split into batches of 10 samples per batch. The model was trained on the entire training set 100 times. After the model was trained, it was fed unlabeled and previously unseen instances of loans from the test set. Just like for the logistic regression algorithm, the model had to generate a prediction of either 0 or 1 for each loan instance in the test set based on the combination of values for the predictor variables.

1. **RESULTS**

After the logistic regression model and the artificial neural network model generated predictions for the test set, a confusion matrix was generated for each model in order to view the experimental results.

The confusion matrix for the logistic regression model was:

**Table 1. Confusion Matrix for Logistic Regression Model**

Predicted Values

Positive (1)

Negative (0)

Actual Values

Positive (1)

Negative (0)

|  |  |
| --- | --- |
| 2291 | 1297 |
| 188 | 322 |

According to this confusion matrix, the logistic regression model correctly predicted 2,291 negatives (0 values) that were truly negatives in the test set, and it correctly predicted 322 positives (1 values) that were truly positives in the test set. In addition, it incorrectly predicted 1,297 positives that were truly negatives in the test set, and it incorrectly predicted 188 negatives that were truly positives in the test set.

The confusion matrix for the artificial neural network model was:

**Table 2. Confusion Matrix for Neural Network Model**

Predicted Values

Positive (1)

Negative (0)

Actual Values

Positive (1)

Negative (0)

|  |  |
| --- | --- |
| 3115 | 473 |
| 402 | 108 |

According to this confusion matrix, the neural network model correctly predicted 3,115 negatives (0 values) that were truly negatives in the test set, and it correctly predicted 108 positives (1 values) that were truly positives in the test set. In addition, it incorrectly predicted 473 positives that were truly negatives in the test set, and it incorrectly predicted 402 negatives that were truly positives in the test set.

The results for the models were evaluated and compared to each other using four metrics that are commonly utilized to assess machine learning models:

*Accuracy*: the proportion of the total number of predictions that were correctly predicted. Calculated using the following formula:

*Precision:* the proportion of the positive predictions (model output of 1) that were correctly predicted. Calculated using the following formula:

*Recall/sensitivity:* the proportion of actual positive instances (target variable value of 1) that were correctly predicted. Calculated using the following formula:

*Specificity:* the proportion of actual negative instances (target variable value of 0) that were correctly predicted. Calculated using the following formula:

The results of the calculations for these four metrics for each model were the following:

**Table 3. Metrics Comparison of Logistic Regression vs. Neural Networks**

|  |  |  |
| --- | --- | --- |
|  | Logistic Regression | Neural Network |
| Accuracy | 63.76% | **78.55%** |
| Precision | **19.89%** | 18.78% |
| Recall/  Sensitivity | **63.14%** | 21.76% |
| Specificity | 63.85% | **86.62%** |

1. **DISCUSSION**

The purpose of this project was to determine whether logistic regression or artificial neural networks are better suited for creating a model that can predict the likelihood that a person will default on a loan from a financial institution. The results of this study do not provide a conclusive answer to the query as each algorithm excelled in different areas of evaluation. It can be seen from Table 3 that the neural network model was superior in terms of accuracy and specificity, and the logistic regression model was better at precision and recall/sensitivity.

The problem with a model with high sensitivity and low specificity is that it may incorrectly classify an applicant who would in fact repay the loan in full as someone who would default on the loan, resulting in a loss of potential profits for the institution. The problem with a model with high specificity and low sensitivity is that it may incorrectly classify an applicant who would default on the loan as someone who would repay the loan in full, resulting in a higher rate of default. Therefore, a financial institution that uses these results to choose the superior model for making a credit risk decision must decide which evaluation metric is most significant in making this decision.

Even though the data for this study included real world data from a legitimate lending company, the data set used in the experiment was relatively small in comparison to the large data sets normally used for machine learning studies. It is generally understood that the predictive power of supervised learning techniques such as neural networks and logistic regression improve with more data [2].Thus, a more definitive conclusion for this study could likely be made with a larger data set.

As discussed in the Literature Review, the logistic regression model for loan default prediction that was developed for the study conducted by Zhu et al. had an accuracy of 73%, which is better than the 63.76% accuracy for the logistic regression model developed for this study. Zhu et al. also used Lending Club loan data for their study, but they were able to acquire a much larger data set than the one used in this study. Specifically, their data set contained 115,000 loans, while the data set used in this study only contained 39,718 loans. Comparing the accuracy of these two studies while taking into consideration the size of the data set bolsters the notion that machine learning models perform better when they have more data to use for training.

1. **REFERENCES**
2. Brixius, Nathan. “The Logit and Sigmoid Functions.” *Nathan Brixius*, 30 Jan. 2019, nathanbrixius.wordpress.com/2016/06/04/functions-i-have-known-logit-and-sigmoid/.
3. Brownlee, J. (2020, August 25). Impact of Dataset Size on Deep Learning Model Skill And Performance Estimates. Retrieved November 18, 2020, from https://machinelearningmastery.com/impact-of-dataset-size-on-deep-learning-model-skill-and-performance-estimates/
4. Eletter, Shorouq & Yaseen, Saad & Elrefae, Ghaleb. 2010. Neuro-Based Artificial Intelligence Model for Loan Decisions. *American Journal of Economics and Business Administration* 2, 1 (Jan. 2010), 27-34. DOI: 10.3844/ajebasp.2010.27.34.
5. Handzic, Meliha & Aurum, Aybüke. 2001. Knowledge Discovery: Some Empirical Evidence and Directions for Future Research. 70. DOI: 10.1007/978-3-642-57547-1\_86.
6. Handzic, Meliha & Tj, Felix & Yeo, Julia. 2003. How Neural Networks Can Help Loan Officers to Make Better Informed Application Decisions. DOI: 10.28945/2602.
7. “Initializing and Accessing Bias with Keras.” *Deeplizard*, deeplizard.com/learn/video/zralyi2Ft20.
8. Jepkemei, Betty. 2019. Effectiveness of Artificial Neural Network in Credit Risk Analysis. *International Journal of Scientific & Engineering Research* 10, 7 (June 2019).
9. Jordan, Jeremy. “Neural Networks: Representation.” *Jeremy Jordan*, Jeremy Jordan, 26 Jan. 2018, www.jeremyjordan.me/intro-to-neural-networks/.
10. Keras Team. “Keras Documentation: Layer Weight Initializers.” *Keras*, keras.io/api/layers/initializers/.
11. Kostadinov, S. (2019, August 12). Understanding Backpropagation Algorithm. Retrieved October 25, 2020, from https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd
12. Manual to Automated Credit Underwriting.” *Medium*, Medium, 2 Jan. 2020, medium.com/@DigiFi/transitioning-from-manual-to-automated-credit-underwriting-ed56da6ab142.
13. Mulla, Zeeshan. “Cost, Activation, Loss Function|| Neural Network|| Deep Learning. What Are These?” *Medium*, Medium, 5 May 2020, medium.com/@zeeshanmulla/cost-activation-loss-function-neural-network-deep-learning-what-are-these-91167825a4de.
14. “Principles for the Management of Credit Risk.” *The Bank for International Settlements*, 27 Sept. 2000, www.bis.org/publ/bcbs75.htm.
15. Schoonjans, Frank. “Logistic Regression.” *MedCalc*, MedCalc Software, 1 July 2020, www.medcalc.org/manual/logistic\_regression.php
16. Serrano-Cinca C, Gutiérrez-Nieto B, López- Palacios L. 2015. Determinants of Default in P2P Lending. *PLoS ONE* 10(10): e0139427 (Oct. 2015). DOI:10.1371/ journal.pone.0139427.
17. Keras Team. “Keras Documentation: Layer Weight Initializers.” *Keras*, keras.io/api/layers/initializers/.
18. Sharma, Avinash V. “Understanding Activation Functions in Neural Networks.” *Medium*, The Theory Of Everything, 30 Mar. 2017, medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0.
19. “The Complex Story of American Debt.” Pew Research Center, Washington, D.C. (July 2015). https://www.pewtrusts.org/-/media/assets/2015/07/reach-of-debt-report\_artfinal.pdf.
20. “What Is Manual Underwriting And How Does It Work?” *Rocket Mortgage*, 16 July 2020, www.rocketmortgage.com/learn/what-is-manual-underwriting-and-how-does-it-work.
21. Wijaya, C. Y. (2020, September 14). 5 SMOTE Techniques for Oversampling your Imbalance Data. Retrieved October 25, 2020, from https://towardsdatascience.com/5-smote-techniques-for-oversampling-your-imbalance-data-b8155bdbe2b5
22. Zhu, Lin & Qiu, Dafeng & Ergu, Daji & Ying, Cai & Liu, Kuiyi. 2019. A study on predicting loan default based on the random forest algorithm. *Procedia Computer Science* 162 (Jan. 2019), 503-513. DOI: 10.1016/j.procs.2019.12.017.