**Loan Eligibility Prediction**

Ronald Bodnar

ID #011194327

WGU Email: rbodna1@wgu.edu

12/30/2023

C964 – Computer Science Capstone

**Table of Contents**

A. Letter of Transmittal …………………………………………………………………..…… 3

Letter of Transmittal …………………………………………………………..………. 3

B. Project Proposal Plan ……………………………………………………………..………... 5

Project Summary …………………………………………………………..………….. 5

Data Summary …………………………………………………………..…………….. 5

Implementation …………………………………………………………..……………. 6

Timeline …………………………………………………………………………..…… 7

Evaluation Plan …………………………………………………………………..……. 7

Resources and Costs ………………………………………………………………..…. 8

C. Application ……………………………………………………………………………..…... 9

D. Documentation ………………………………………………………………………….…. 10

Solution Summary ……………………………………………………………………. 10

Data Summary …………………………………………………………………….….. 11

Machine Learning ………………………………………………………………….…. 18

Validation …………………………………………………………………………….. 19

Visualizations ………………………………………………………………………… 20

User Guide ……………………………………………………………………….…… 28

E. References …………………………………………………………………………….…… 29

**Letter of Transmittal**

12/30/2023

John Doe

Operations Manager

Northern Bank

2400 W Main St, Turlock, CA 95380

Subject: Proposal for Implementing a Loan Eligibility Prediction System

Dear John,

I hope this letter finds you well. I am writing you to propose the implementation of a loan eligibility prediction system that aims to address Northern Bank's need for an efficient method of determining loan eligibility. Currently, all loan eligibility is manually processed, which is time-consuming, prone to inconsistent loan approvals, and causes delays for applicants.

This proposed loan eligibility prediction system utilizes machine learning algorithms and historical loan data to predict loan eligibility in real-time. Various demographic and financial factors, such as marital status, number of dependents, income, and credit history, will be considered to ensure consistent and accurate predictions.

This proposal aims to implement an automated loan eligibility prediction system to virtually eliminate inconsistencies and processing time for loan applications, facilitating a smooth loan application process for both Northern Bank and its applicants. The estimated budget for successful implementation of this data product is $24,520 and has a projected completion time of eight weeks.

Our team is comprised of accredited college graduates in the fields of Computer Science and Software Engineering, with experience in data science and machine learning projects. We have a vast portfolio of machine learning projects that have been implemented with great success, and we are confident that this loan eligibility prediction system aligns Northern Bank’s needs and will result in great benefits.

I appreciate your consideration and look forward to discussing this proposal further with you.

Sincerely,

Ronald Bodnar

**Project Proposal Plan**

**Project Summary**

This project aims to address operational inefficiencies in the loan application process by developing an automated loan eligibility prediction data product. The existing manual system leads to delays and inconsistencies in eligibility results, which costs Northern Bank to lose potential customers and waste resources that are best used performing other tasks.

By adopting this automated loan eligibility prediction system, Northern Bank can offer a more streamlined and transparent application process to its customers, ensure consistency across loan eligibility predictions, and conserve costly resources.

The machine learning prediction model will be delivered in the form of an intuitive, user-friendly application that benefit both Northern Bank and its potential customers by offering a streamlined loan approval workflow. In addition, a comprehensive user guide will be included that will provide a detailed breakdown of the application’s workflow and ensure easy adoption for all users.

**Data Summary**

The raw data for this project will be sourced from a comprehensive historical loan dataset on Kaggle.com, called Eligibility Prediction for Loan, and incorporates a broad spectrum of demographic and financial variables to ensure relevance to real-world scenarios [[1]](https://www.kaggle.com/datasets/devzohaib/eligibility-prediction-for-loan).

The development life cycle for the loan eligibility prediction system will take place in phases, to ensure effective data processing and management. The initial phase in the life cycle is the data collection phase, where the dataset will be loaded into the application, and brief analysis is done to gain familiarity with the dataset. Then comes the design phase, where exploratory data analysis (EDA) will be conducted to gain insights into dataset, such as relationships, potential patterns, and the distribution of features. This phase lays the foundational understanding of the data and informs decisions to be made in the development phase. Moving into the development phase, we will begin preprocessing of the dataset, including handling missing values, encoding categorical variables, and normalizing features of the dataset. By performing these steps, we will ensure quality data is input to the machine learning model, allowing it to provide reliable predictions.

Ethical considerations for this project will include encryption of personal identifiers to protect privacy and attempting to mitigate biases in the machine learning model. By adhering to data protection regulations and obtaining applicant consent for the use of data, we will ensure the ethical development of this loan eligibility prediction model.

**Implementation**

Implementation of the Loan Eligibility Prediction system, we will adhere to the industry-standard Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology.

**Business Understanding**

The initial phase involves collaborating with stakeholders to define objectives and requirements for the project and establish key performance indicators (KPIs) for the Loan Eligibility Prediction system, ensuring a clear understanding of the desired outcomes.

**Data Understanding**

This phase focuses on the collection of data, exploration of the historical loan dataset to gain insights into feature distributions and relationships, data quality and anomaly issues, and sets the foundation for preprocessing.

**Data Preparation**

The data preparation phase focuses on the preprocessing of the dataset and is where we will address missing values, handle outliers, and perform any necessary encoding for categorical variables. With a clean dataset we will then split the dataset into two sets for use in training and testing the model.

**Modeling**

In the modeling phase, we will select a suitable machine learning algorithm for predicting loan eligibility, train the model with the training dataset, and aim for optimal accuracy.

**Evaluation**

Evaluate model comparisons and perform cross-validation to obtain evaluation metrics for each algorithm that is tested. Confusion matrices will be utilized to visualize model performance.

**Deployment**

The final phase is the deployment of the model, which will involve integrating the loan eligibility prediction model into the loan application. User training will be conducted, and the model will be deployed for public use.

**Timeline**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone / Deliverable** | **Duration** | **Start Date** | **End Date** |
| Proof of Concept | 1 week | December 15, 2023 | December 22, 2023 |
| Data Collection and Preparation | 2 weeks | December 25, 2023 | January 5, 2024 |
| Model Development and Training | 2 weeks | January 8, 2024 | January 19, 2024 |
| Model Evaluation and Testing, Integration | 2 weeks | January 22, 2024 | February 2, 2024 |
| Application Delivery and Deployment | 1 week | February 5, 2023 | February 9, 2024 |

**Evaluation Plan**

Throughout the phases of development, various verification methods will be applied to ensure the deliverable meets the specified requirements. Data profiling will be used to validate the consistency of the collected data and its alignment with the objectives. Insights gained from Exploratory Data Analysis (EDA) will be verified using visualizations and unit tests will be performed during data preprocessing, ensuring proper encoding and handling of missing values. In training the model, cross-validation and performance metric evaluation will take place to verify the effectiveness of the machine learning model.

Upon completion of the project, user acceptance testing (UAT) will be utilized to ensure that the application is meeting end-users’ expectations. End-to-End testing will be used to verify the seamless workflow of the loan eligibility application, ensuring accurate and reliable results.

**Resources and Costs**

|  |  |  |
| --- | --- | --- |
| **Resource** | **Description** | **Cost** |
| Hardware | Workstations are already owned.  Network is already provisioned. | $0 |
| Software | Jupyter Notebook, Python 3.11, and all libraries necessary for this project are free and open source. | $0 |
| Cloud Infrastructure | Necessary cloud infrastructure for model deployment and hosting, yearly cost. | $9,000 |
| Data Scientist | Engineer to perform data analysis and processing.  $63/hr x 80 hours | $5,040 |
| Machine Learning Engineer | Engineer to develop and fine-tune the machine learning model.  $60/hr x 80 hours | $4,800 |
| Software Engineer | Engineer to perform model integration and testing.  $71/hr x 80 hours | $5,680 |
| User Training | No specific user training is required. | $0 |
|  | **Total** | $24,520 |

**Application**

The application’s necessary files are included with this proposal with the following file names:

* C964 CS Capstone.ipynb
* loan\_data.csv

**Post-Implementation Report**

**Solution Summary**

This project was developed to address operational inefficiencies in the loan application process at Northern Bank, where manual determination methods resulted in delays, inconsistencies, and wasted resources. The solution involved developing an automated loan eligibility prediction system using a machine learning model that was trained on historical loan data. This model was integrated into an intuitive user-friendly interface designed to streamline the application process and eliminate manual eligibility determination. The system ensures consistent loan eligibility determinations with minimal errors and conserves valuable resources for Northern Bank.

The application offers a comprehensive solution to the operational inefficiencies that Northern Bank faces in its loan application process by utilizing machine learning and an intuitive user interface. The application addresses delays and inconsistencies in loan applications due to manual processing limitations and thorough preprocessing ensures the reliability and consistency of the input data. Incorporating the Random Forest Classifier model, which was trained and fine-tuned on the historical loan dataset, provides accurate and consistent determinations, and mitigates errors associated with manual decision-making.

**Data Summary**

The dataset used for the development of the loan eligibility prediction system was obtained from Kaggle.com [[1]](https://www.kaggle.com/datasets/devzohaib/eligibility-prediction-for-loan).

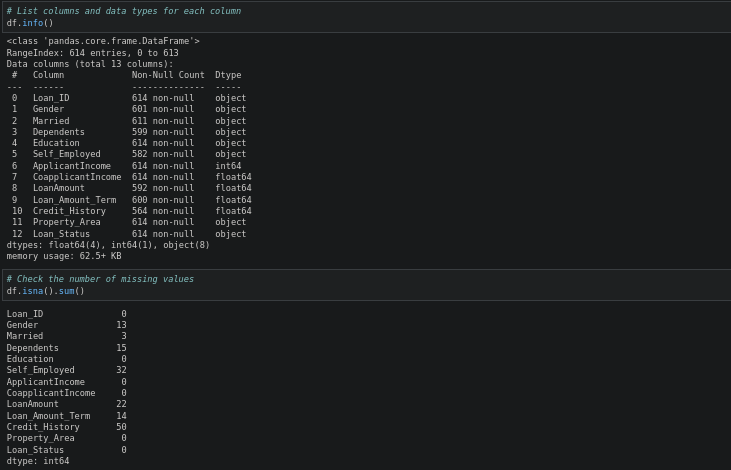
The initial dataset had a total of 13 columns, as shown here:

A screenshot of a computer

Description automatically generated

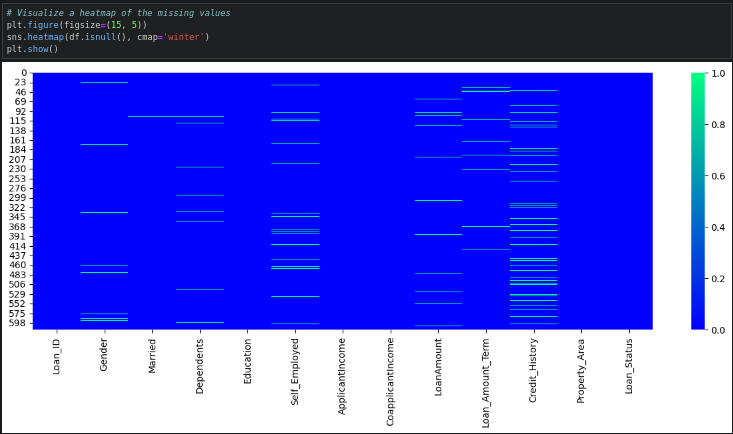
A screenshot of a computer

Description automatically generated

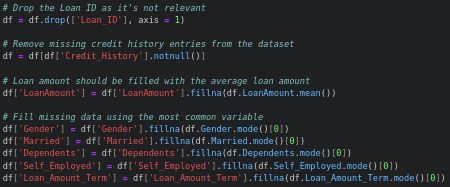
Upon initial data analysis, I determined that label encoding would be necessary and missing values were present and needed handling. The following were used for initial data analysis:

A screen shot of a computer

Description automatically generatedWith a pie chart, I visualized the loan approval balance within the dataset and determined there to be an imbalance, as seen here:

Then, I determined how to handle the missing values by visualizing the distribution of missing values. I determined the data to be missing at random (MAR) with this heat map:

Credit history was the most prevalent feature with missing values, and due to the high correlation observed between credit history and loan eligibility, those values were dropped instead of assumed and filled. Loan ID was also dropped due to its irrelevance. All other missing values were handled by filling them with the most common value (mode) and the average for loan amount, seen below:



Once all missing values were filled, I continued with label encoding of categorical columns and printed them to confirm the changes with the following code:

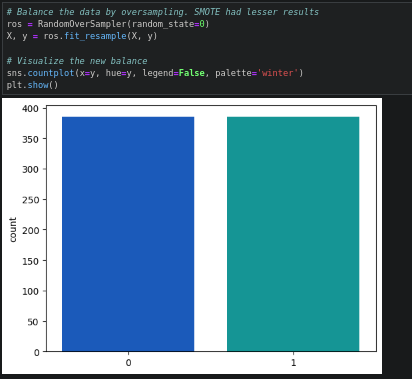
A black screen with white text

Description automatically generated

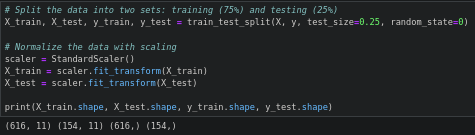
Then, I defined two datasets, where the independent features in X are a one-to-one mapping of the dependent variables (targets) in y, before balancing and scaling the testing dataset.



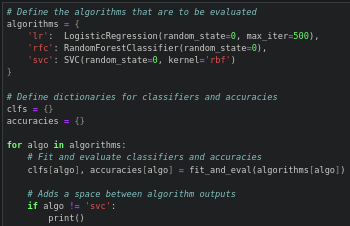
To balance the dataset, I used RandomOverSampler to over-sample the minority classes and ensured balance with a bar graph of target classes:

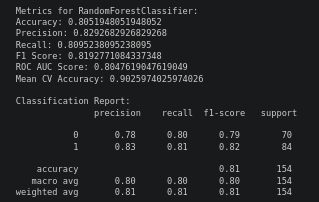


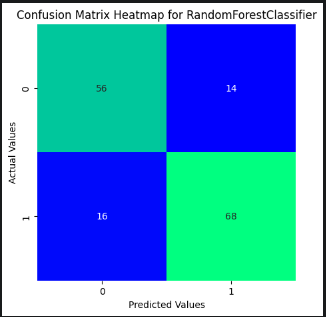
After the dataset was balanced, it was split into testing and training datasets that were used to train and evaluate the model’s performance and the feature datasets (X\_train and X\_test) were then scaled with StandardScaler:



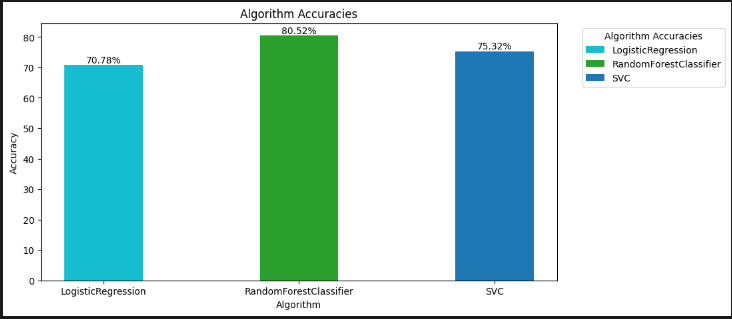
The datasets at this point have now been normalized and are ready to be used to fit and evaluate each model’s performance. This was done through a utility function I wrote, fit\_and\_eval, that was designed to fit and evaluate each model with the same process, providing consistent evaluation results across all models. This function fits the model, makes predictions on test data, captures metrics such as accuracy, recall, precision, f1 score, and ROC AUC score, performs cross-validation, and outputs the evaluation in text form as well as a confusion matrix heat map.



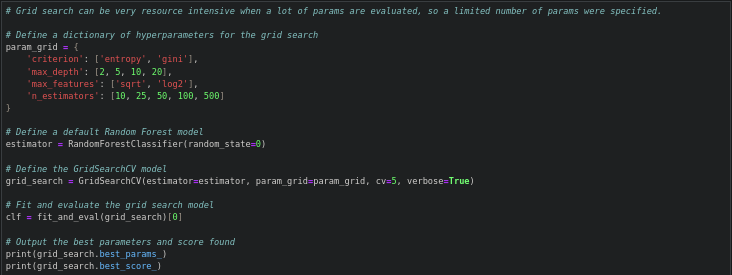
A sample output for Random Forest Classifier can be seen here, with the other two algorithms omitted:

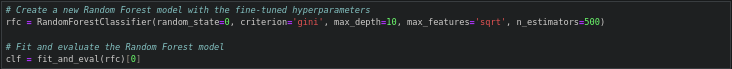


Since more than one algorithm was evaluated, I compare the accuracy results for each algorithm and pick one to proceed with hyperparameter tuning:

****

Random Forest Classifier results in the highest accuracy, so I set up a GridSearchCV parameter grid to begin hyperparameter tuning for best results:



The fine-tuned model resulted in negligible changes in accuracy, but I used the best parameters suggested by the GridSearchCV model and ran fit\_and\_eval on the fine-tuned Random Forest Classifier model to provide a final model to use for predictions.

**Machine Learning**

Three machine learning algorithms were trained and evaluated in the development of this model: Logistic Regression, Random Forest Classifier, and Support Vector Classifier.

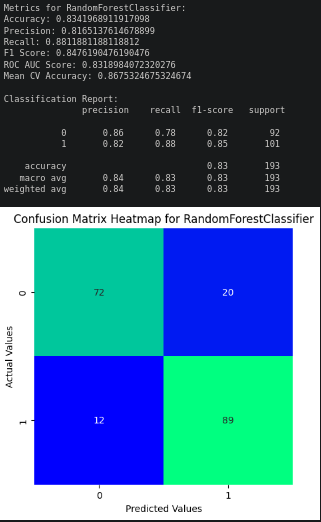
**Logistic Regression** is a supervised machine learning algorithm primary used for binary classification problems and uses probability to predict the target class. The method was developed by fitting the sigmoid function to the training data and optimizing coefficients through gradient descent. Logistic Regression was chosen as a possible algorithm due to its simplicity and effectiveness in binary classification problems.

**Random Forest Classifier** is a machine learning algorithm that is an extension of the bagging ensemble learning method. It builds a Random Forest of decision trees with random subsets of data during training and outputs the mode of the classes for classification. The method was developed by combining various decision trees that were each trained on different subsets of data and features. The Random Forest Classifier was chosen for its ability to handle complex relationships in the dataset, reduce overfitting, and provide feature importance metrics.

**Support Vector Classifier** is a supervised learning algorithm that performs classification by finding the hyperplane that best separates the feature classes. The method was developed by selecting the ‘linear’ kernel function, which is responsible for transforming the input data into the required form. The hyperplane is then optimized to minimize the classification error and determine optimal coefficients for the support vectors. The Support Vector Classifier was selected for its ability to generalize new, unseen data, and for its effectiveness in finding decision boundaries in complex relationships.

**Validation**

The accuracy of all three models was determined using k-fold cross-validation to provide a multitude of metrics for each fold, such as accuracy, precision, recall, and ROC AUC score. The average results of these metrics provided me with insights into each model’s overall performance and its effectiveness in correctly predicting the correct class. An evaluation report for each model was generated after fitting the model and was the basis for selecting a final model for fine-tuning. The final model’s evaluation report, coming in at 83.41% accuracy and 86.75% cross-validation mean accuracy, can be seen below:



**Visualizations**

A pie chart with text on it

Description automatically generated

Figure : Loan approval distribution

A group of blue and green bars

Description automatically generated

Figure 2.1: Distribution of independent features

A group of blue and green bars

Description automatically generated

Figure 2.2: Distribution of independent features (cont.)

A group of blue and green bars

Description automatically generated

Figure 3.1: Loan approval distribution for independent features

A group of blue and green bars

Description automatically generated

Figure 3.2: Loan approval distribution for independent features (cont.)

A blue screen with white text

Description automatically generated

Figure 4: Matrix heat map for missing values

A green and white chart

Description automatically generated

Figure 5: Feature correlation matrix heat map

A blue and green rectangular bars

Description automatically generated

Figure 6: Dependent variable distribution after oversampling

A graph of a bar chart

Description automatically generated with medium confidence

Figure 7: Algorithm accuracy comparison for trained models

A graph of a graph with text

Description automatically generated with medium confidence

Figure 8: Feature importances for fine-tuned model

**User Guide**

1. Open a web browser and navigate to the Google Colaboratory online notebook link [[2]](https://colab.research.google.com/drive/1idhWC1-8uCMqcpl1CTYpKDxWe-0ljfqB?usp=sharing).

2. In the menu at the top, select **Runtime** and then **Run all**, as seen below:

A screenshot of a computer

Description automatically generated

3. If you receive a popup stating this notebook was not authored by Google, click **Run anyway**:

A screenshot of a computer

Description automatically generated

5. Scroll down to the bottom of the page and wait for the user interface to begin. This will take some time, please be patient.

You should see the following under the last cell at the bottom of the page when it completes:

A black background with white text

Description automatically generated

6. Answer the prompt questions to receive a loan eligibility prediction.

**References**

[1] Devzohaib. (2022). Eligibility Prediction for Loan. Retrieved 12/18/2023 from <https://www.kaggle.com/datasets/devzohaib/eligibility-prediction-for-loan>.

[2] Google. (n.d.). Google Colaboratory. <https://colab.research.google.com/drive/1idhWC1-8uCMqcpl1CTYpKDxWe-0ljfqB?usp=sharing>