**LOAN APPROVAL PREDICTION USING GPU PROGRAMMING**

**PROJECT REPORT**

**INDEX**

Table of Contents

[1.Introduction: 3](#_Toc152935872)

[2. Design: 3](#_Toc152935873)

[3. Functions 5](#_Toc152935875)

[4. Workflow 7](#_Toc152935877)

[5. How to Execute 8](#_Toc152935878)

[6. Dependencies 8](#_Toc152935879)

[7. Conclusion 9](#_Toc152935880)

[8. References 10](#_Toc152935881)

# INTRODUCTION:

Loans are the primary necessity of the modern day world. Banks receive the majority of the entire profit only from this. Those who purchase any type of luxury, such as homes, vehicles, etc., and students who balance their living and educational costs will benefit from it.

But when it comes to assessing whether the applicant’s profile is suitable to be granted the loan or not. Banks have a lot of things to manage.

Therefore, to make the process easier and determine whether or not the candidate's profile is relevant, we will be employing GPU Programming using CUDA with Deep Learning and Data Analytics in Python. We will be using necessary data like Marital Status, Education, Applicant Income, Credit History, etc. A machine learning model can automate this process by analyzing historical loan data to identify patterns that distinguish between approved and rejected loans.

# 2. DESIGN:

There are two primary steps to this project:

* The goal of the exploratory data analysis (EDA) stage is to comprehend the data by looking at its features, distributions, and correlations between different variables.
* Model Building: Using the prepared data, a machine learning model is trained at this stage to predict whether fresh applicants will be approved for loans.

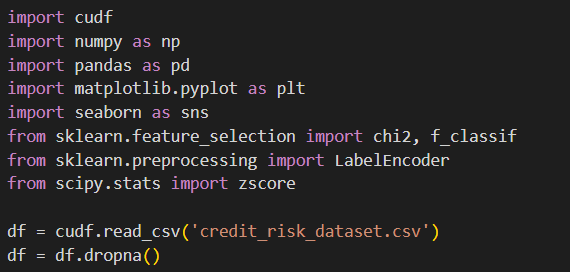
The typical methodology used by a loan approval prediction model is supervised machine learning. Below is a broad design overview:

* Data collection: Compile historical loan data that includes details about the following: loan characteristics (amount, purpose, term), applicant demographics, income, and employment, and loan outcomes (approved, rejected).
* Data preprocessing involves handling outliers, missing values, and encoding categorical variables to clean up and prepare the data. In order to generate new features from existing ones, feature engineering can also be used.
* Model Selection: Based on the goals of the project and the properties of the data, select the best machine learning algorithms. Gradient Boosting, Decision Trees, Random Forests, Support Vector Machines (SVM), and Logistic Regression are popular options.
* Model Training: Divide the data into training and testing sets for the model. To discover the correlation between features and loan outcomes, train the model using the training set.
* Model Evaluation: Use metrics such as accuracy, precision, recall, and F1-score to assess the model's performance on the unseen testing data.
* Model Refinement: Adjust hyperparameters, experiment with different algorithms, or apply feature engineering techniques to improve the model based on the evaluation results.

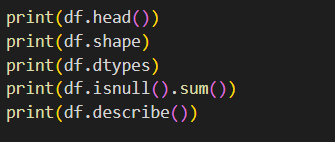
# 

# 3. FUNCTIONS:

Data Loading and Cleaning:



Basic Exploration:



Visualization:

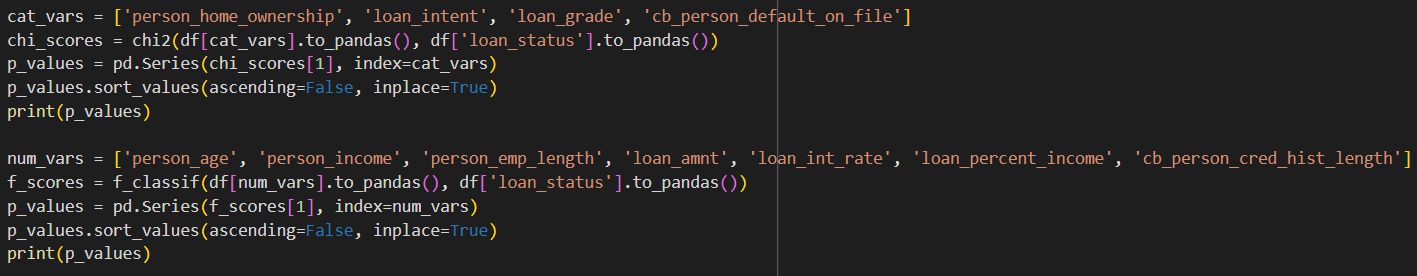
The code uses libraries like matplotlib and seaborn to create various visualizations to understand the data distribution:

\* Loan status counts (bar chart)

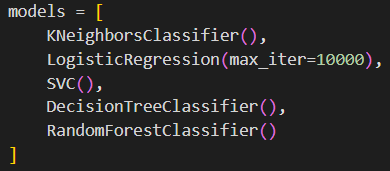
\* Distributions of numerical features (histograms)

\* Value counts of categorical features (bar charts)

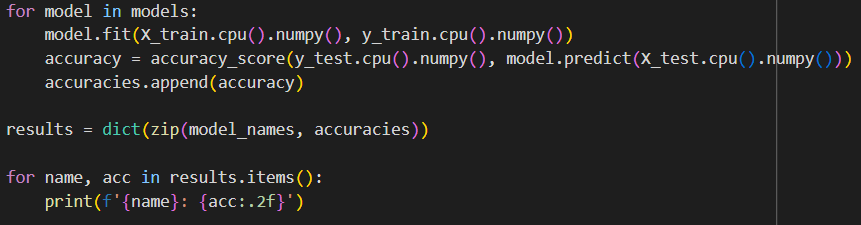
Feature Selection:



Model Selection:



Model Prediction:



# 4. WORKFLOW:

1. **Data Collection and Preprocessing:**
   * Gather loan data.
   * Clean and prepare the data.
2. **Exploratory Data Analysis (EDA):**
   * Analyze the data to understand its characteristics.
3. **Feature Engineering :**
   * Create new features if needed.
4. **Feature Preprocessing:**
   * Scale numerical features.
   * Encode categorical features.
5. **Model Training and Evaluation:**
   * Split data into training and testing sets.
   * Train machine learning models on the training data.
   * Evaluate model performance on the testing data.
   * Refine models based on evaluation results (optional).
6. **Model Selection:**
   * Choose the best performing model for prediction.

# 5. HOW TO EXECUTE:

* Choose the IDE that you want to work on such as Visual Studio Code, Jupyter Notebook, Anaconda, or Google CoLab
* Install required libraries (e.g., pandas, scikit-learn).
* Load and prepare the loan data.
* Perform exploratory data analysis.
* Preprocess features (scaling, encoding).
* Split data into training and testing sets.
* Train and evaluate different machine learning models.
* Select the best performing model.

# 6. DEPENDENCIES:

# Programming languages: Python

# Libraries: CuDf and Torch for GPU programming using deep learning frameworks and dataframe library, Pandas (data manipulation), NumPy (numerical computations), scikit-learn (machine learning), matplotlib/seaborn (data visualization)

* **Tools:** IDE such as Visual Studio Code, Jupyter Notebook, Anaconda or Google CoLab
* **Hardware:** NVIDIA GPU -> CUDA is specifically designed for NVIDIA graphics processing units (GPUs). Ensure you have a compatible NVIDIA GPU with sufficient compute capability. You can check NVIDIA's website for a list of supported GPUs: https://developer.nvidia.com/cuda-gpus
* **CUDA Toolkit:** This is the core toolkit that provides the development environment for creating GPU-accelerated applications with CUDA. It includes a compiler, libraries, debugging and optimization tools, and a runtime library. You can download the CUDA Toolkit from the NVIDIA developer website: https://developer.nvidia.com/cuda-toolkit
* **CUDA Libraries:** NVIDIA provides additional libraries like cuDNN (deep neural networks) and cuBLAS (basic linear algebra) that can accelerate specific tasks on GPUs.
* Operating System: While CUDA supports various operating systems, compatibility details and installation procedures might vary. Ensure you have a compatible OS version for the chosen CUDA Toolkit.
* **GPU Drivers:** Make sure you have the latest NVIDIA drivers installed for your specific GPU to ensure optimal performance and compatibility with the CUDA Toolkit.

# 7. CONCLUSION:

Financial institutions can benefit from using loan approval prediction models to automate loan decisions, increase productivity, and possibly even manage risk. Data scientists can create efficient models to assist loan approval procedures by employing a systematic process that includes data preparation, model selection, training, and evaluation.

# 8. REFERENCES:

Python:

<https://www.w3schools.com/python/python_reference.asp>

<https://www.python.org/>

<https://realpython.com/python-application-layouts/>

<https://scikit-learn.org/stable/>

<https://pandas.pydata.org>

<https://www.geeksforgeeks.org/getting-started-with-pytorch/>

https://www.geeksforgeeks.org/loan-approval-prediction-using-machine-learning/

CUDA:  
https://docs.nvidia.com/cuda/pdf/CUDA\_C\_Programming\_Guide.pdf