A project report on

EMPOWERING RURAL INDIA THROUGH EFFECTIVE MICROFINANCE BY IMPLEMENTING MACHINE LEARNING & BLOCKCHAIN TECHNOLOGY

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

ARASADA EKAVEERA ANEEL KUMAR (19BCE1535)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023

EMPOWERING RURAL INDIA THROUGH EFFECTIVE MICROFINANCE BY IMPLEMENTING MACHINE LEARNING & BLOCKCHAIN TECHNOLOGY

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

ARASADA EKAVEERA ANEEL KUMAR (19BCE1535)

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I hereby declare that the thesis entitled "**EMPOWERING RURAL INDIA THROUGH EFFECTIVE MICROFINANCE BY IMPLEMENTING MACHINE LEARNING & BLOCKCHAIN TECHNOLOGY**" submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai, is a record of bonafide work carried out by me under the supervision of Dr. B V A N S S Prabhakar Rao.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: Signature of the Candidate

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **“EMPOWERING RURAL INDIA THROUGH EFFECTIVE MICROFINANCE BY IMPLEMENTING MACHINE LEARNING & BLOCKCHAIN TECHNOLOGY”** is prepared and submitted by **ARASADA EKAVEERA ANEEL KUMAR (19BCE1535)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** programme is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr./Prof.: Dr. B V A N S S Prabhakar Rao

Date:

Signature of the Examiner 1	Signature of the Examiner 2
Name:	Name:
Date:	Date:

Approved by the Head of Department
B. Tech. CSE

Name: Dr. Nithyanandam P

Date: 24 – 04 – 2023

(Seal of SCOPE)

ABSTRACT

The prime focus of financial inclusion is centered on the necessity to provide financial services to the underprivileged sections of the community. Nevertheless, the conventional financial market is still unavailable to them, owing to the individual’s low income and lack of collateral. Consequently, many turn to neighborhood moneylenders, commonly referred to as "loan sharks," who demand excessive interest rates. The advent of microfinance has provided these individuals with a new ray of light, as it provides microcredit (small valued loans) to support their microbusinesses and engage in production activities, increasing financial inclusion and economic development. Some of the most persistent challenges underlying microfinance have been identified as high overhead costs, a lack of financial sustainability and scalability, the prevalence of significant information asymmetries, a low level of transparency and bad governance, all of which can be a conundrum at the same time and that may lead in serious jeopardy. As developing technology began to permeate every element of life, microfinance had to be linked with the technology as well to mitigate the risks and prevent the problems outlined in microfinance. The transaction history of loans may be impacted since databases are susceptible to data tampering. Incorporating blockchain technology into microfinance is the contemporary solution to this. Accordingly, in this article, we develop a decentralized application based on blockchain technology that enhances data integrity while making microcredit more accessible.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. B V A N S S Prabhakar Rao, Professor, SCOPE, Vellore Institute of Technology, Chennai, for his/her constant guidance, continual encouragement, understanding; more than all, he/she taught me patience in my endeavor. My association with him / her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of computer science.

It is with gratitude that I would like to extend thanks to our honorable Chancellor, Dr. G. Viswanathan, Vice Presidents, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan and Mr. G V Selvam, Assistant Vice-President, Ms. Kadhambari S. Viswanathan, Vice-Chancellor, Dr. Rambabu Kodali, Pro-Vice Chancellor, Dr. V. S. Kanchana Bhaaskaran and Additional Registrar, Dr. P.K.Manoharan for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dean, Dr. Ganesan R, Associate Dean Academics, Dr. Parvathi R and Associate Dean Research, Dr. Geetha S, SCOPE, Vellore Institute of Technology, Chennai, for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Nithyanandam P, Head of the Department, Project Coordinators, Dr. Abdul Quadir Md, Dr. Priyadarshini R and Dr. Padmavathy T V, B. Tech. Computer Science and Engineering, SCOPE, Vellore Institute of Technology, Chennai, for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staff at Vellore Institute of Technology, Chennai, who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: **Arasada Ekaveera Aneel Kumar**

CONTENTS

Chapter 1

INTRODUCTION

1.1 MICROFINANCE

Financial inclusion is the means of ensuring that everyone, regardless of economic level, has equitable and affordable access to financial services. It pertains to offering services to both individuals and firms. Low-income individuals generally lack collateral and hence have no opportunity to get a loan, save money, or invest for the future. Banks frequently consider women, in particular, as uncreditworthy.

Microfinance is the provision of financial services to low-income individuals or organizations that are traditionally excluded from regular banking. The terms of the repayment are typically flexible and tailored to the borrower's needs, and generally they are collateral free. The majority of microfinance institutions specialize on providing credit in the form of modest working capital loans, often known as microloans or microcredit.

1.1.1 OVERVIEW OF MICROFINANCE

Microfinance is a vital gear in the wheel that strives to deliver financial inclusion in the form of inexpensive financial goods. It is one of the methods for achieving all-around economic development. The ultimate purpose of microfinance is to provide impoverished individuals with the chance to become self-sufficient. As previously said, microfinance seems to have been a means for lifting a substantial section of the world's population out of poverty, but it was not successful on a broad scale.

However, microfinance has also faced criticism, particularly for charging high-interest rates and for perpetuating poverty by encouraging dependence on debt. There are several case studies of people who used these loans to advance in their life, but most of them were mishandled and they ended up getting into even more debt.

Despite these challenges, microfinance has continued to play a vital role in promoting financial inclusion and reducing poverty worldwide. The sector has evolved over the years, with new models and technologies emerging to improve the efficiency and reach of microfinance institutions. The future of microfinance looks promising, with many opportunities to scale up and expand access to financial services for the underserved and excluded populations.

1.1.2 MOTIVATION

The lockdowns and restrictions on movement have led to a loss of livelihoods for many rural workers, particularly those in the informal sector. Small businesses such as local shops and restaurants have also been affected. Many rural households and small businesses rely on credit to meet their financial needs. However, due to the economic uncertainty caused by the pandemic, banks and other financial institutions have become more cautious about lending, which has made it more difficult for rural communities to access credit. After careful consideration of how I could contribute to the financial sustainability of rural communities, I became enthusiastic about selecting this topic, which involves utilizing credit services to empower people.

1.2 IMPACT OF MICROFINANCE IN INDIA

Microfinance has been playing a significant role in India for the past few decades in providing access to credit and other financial services to people living in poverty, particularly in rural areas. The Indian microfinance industry started in the 1970s with the establishment of Self-Help Groups (SHGs) and has since grown into a formal sector with over 200 microfinance institutions (MFIs) operating across the country.

The microfinance industry in India has evolved over the years, from providing microcredit to small businesses to offering a range of financial services, including microinsurance, microsavings, and remittance services. The sector has also become more regulated, with the Reserve Bank of India (RBI) issuing guidelines for microfinance institutions in 2011 to ensure responsible lending practices and protect consumers.

However, the industry has faced challenges, particularly after the microfinance crisis in Andhra Pradesh in 2010, where over indebtedness and aggressive collection practices led to a wave of defaults and suicides. This crisis led to greater scrutiny of the industry and resulted in the enactment of the Microfinance Institutions (Development and Regulation) Bill, 2012, which aimed to regulate the sector and ensure consumer protection.

Despite the challenges, microfinance in India has helped millions of people to access credit and improve their livelihoods. According to the Microfinance Institutions Network (MFIN), the Indian microfinance industry had a gross loan portfolio of over Rs. 2.3 lakh crore ($30 billion) as of March 2021, with over 10 crore (100 million) active borrowers. Microfinance has also been instrumental in empowering women, as around 90% of microfinance borrowers in India are women.

1.3 CHALLENGES FACED BY MICROFINANCE IN INDIA

Microfinance in India has faced several challenges, some of which are unique to the country. Here are some of the key challenges faced by microfinance in India:

Overindebtedness has been a major issue in the Indian microfinance sector, particularly after the microfinance crisis in Andhra Pradesh in 2010. Aggressive lending practices, coupled with a lack of borrower education and awareness, led to high levels of debt and defaults. Microfinance institutions in India are subject to a complex regulatory environment, with multiple agencies involved in the oversight of the sector. This can create confusion and increase compliance costs for microfinance institutions, particularly for smaller players.

The microfinance industry in India has been subject to political interference, particularly at the state level. This has led to the imposition of interest rate caps and other restrictions, which can make it difficult for microfinance institutions to operate sustainably. Microfinance institutions in India may struggle to access funding from traditional sources, particularly in rural areas where the need for financial services is the greatest. This can limit their ability to scale up and expand their services.

The adoption of new technologies, particularly digital platforms, has been slow in the Indian microfinance sector. This can limit the reach of microfinance institutions and make it difficult to improve their operational efficiency. Low levels of financial literacy among microfinance borrowers can make it difficult for them to understand the terms of their loans and make informed financial decisions. This can contribute to over indebtedness and defaults.

Addressing these challenges requires a coordinated effort from microfinance institutions, regulators, policymakers, and other stakeholders. This may involve improving borrower education and financial literacy, promoting responsible lending practices, streamlining regulatory oversight, expanding funding sources, and leveraging new technologies to improve the reach and efficiency of microfinance institutions.

1.4 STATISTICS ON MICROFINANCE

6.2 crore unique borrowers with 12 crore loan accounts were served by the nation's microfinance loan portfolio at the end of September 2022, according to the MFIN Micrometer Q2 FY2022-23 report. As of September 30, 2022, the total gross loan portfolio (GLP) for the whole microfinance sector was Rs 3,00,974 crore. "A 23.5 percent year-on-year gain above Rs 2,43,737 crore as of September 30, 2021," according to the research.

With a loan amount outstanding of Rs 1,10,418 crore, NBFC-MFIs are the second largest source of microcredit, accounting for 36.7 percent of the entire sector portfolio. In comparison to the same quarter in FY21, the average loan disbursed per account during the quarter was Rs 40,571, an increase of around 12%. A total of Rs. 50,029 crores in outstanding loans, or 16.6% of the total, are held by small financing banks (SFBs).Other microfinance institutions (MFIs), which make up 1.1% of the microfinance industry, and the remaining 7.9% is constituted by non-banking financing firms (NBFCs).

According to the report, the number of active microfinance loan accounts climbed by 14.2% over the previous year, reaching 12 crores as of September 30, 2022. Thirteen banks hold the lion's share of the total micro loans outstanding, accounting for 37.7 percent of Rs 1,13,565 crore. Considering the stats, by 2025 the credit demand estimated is projected to be between Rs 17 and Rs 20 lakh crore, the growth pace is likely to ramp up even further.

In recent years, there has been a shift towards digital lending platforms in India, with the emergence of online peer-to-peer (P2P) lending platforms and mobile-based lending apps. The Reserve Bank of India (RBI) has played a key role in regulating the microfinance industry in India, implementing policies to ensure responsible lending practices and prevent overindebtedness.

1.5 PROJECT STATEMENT

The project statement of microfinance using blockchain and machine learning is to leverage the strengths of these two technologies to address the challenges faced by the microfinance industry, including high transaction costs, limited access to financial services, and the risk of overindebtedness.

Blockchain technology can provide a decentralized and secure platform for microfinance institutions to improve their operations and expand their reach, while machine learning can enable more accurate credit risk assessment and borrower profiling, improving loan underwriting and reducing the risk of defaults.

By combining blockchain and machine learning, microfinance institutions can:

Reduce transaction costs: Blockchain technology eliminates the need for intermediaries, reducing transaction costs and increasing efficiency, while machine learning can automate loan underwriting and reduce operational costs.

Increase transparency and trust: Blockchain technology provides a transparent ledger of all transactions, increasing trust and accountability among stakeholders, while machine learning can provide more accurate borrower profiling and credit risk assessment, reducing the risk of overindebtedness and defaults.

Expand access to financial services: Blockchain technology can facilitate the creation of decentralized financial services, while machine learning can enable more accurate credit risk assessment, enabling microfinance institutions to reach underserved communities.

Streamline regulatory oversight: Blockchain technology can provide regulators with real-time access to financial data, while machine learning can enable more effective risk monitoring and regulatory compliance.

Improve operational efficiency: Machine learning can automate loan underwriting, reducing the time and costs associated with manual processes, while blockchain can provide a secure platform for storing and transmitting financial data.

Overall, the use of blockchain and machine learning in microfinance has the potential to transform the industry, enabling microfinance institutions to reach more borrowers, reduce costs, and improve their operations. However, implementing these technologies requires overcoming technical, regulatory, and cultural challenges, including the need for standardization, interoperability, and stakeholder buy-in.

In order to verify rural people's credit history online and design a machine learning strategy, there is no reliable dataset accessible regarding their yearly credit expenditures and transactions.
Blockchain technology is still in its early stages. It requires thorough knowledge from the business to go through the whole process.
The move from conventional techniques to blockchain may pose operational hazards. Many applications may require some system redundancy, such as running blockchain with conventional systems functioning as backup until sufficient stability is achieved to allow for a complete transfer.

1.6 SCOPE OF THE PROJECT

The scope of the project could be to explore how machine learning algorithms can be trained to make more accurate loan decisions, reducing the risk of default and improving the overall efficiency of the lending process. The project can delve into investigating the potential applications of blockchain technology in developing a secure and transparent microfinance platform that enhances accountability, minimizes corruption, and fosters trust between lenders and borrowers.

Chapter 2

LITERATURE SURVEY

2.1 RESEARCH OBJECTIVES

To examine the current state of microfinance in rural India and identify the challenges faced by microfinance institutions (MFIs) in providing financial services to underserved populations. In addition, exploring the potential of machine learning (ML) and blockchain technology (BT) in addressing the challenges faced by MFIs and improving the efficiency and effectiveness of microfinance services in rural India.
Also, to identify the key success factors and challenges of implementing machine learning and blockchain technology in microfinance in rural India and provide recommendations for scaling up the intervention.
To contribute to the existing literature on microfinance, machine learning, and blockchain technology and provide insights for policymakers, practitioners, and researchers interested in using technology to enhance financial inclusion and rural development. design and develop a machine learning and blockchain-based microfinance platform tailored for the needs of rural communities in India.

2.2 LITREATURE SURVEY OF EXISTING WORK

[1] The paper proposes a decentralized microfinance model that utilizes blockchain technology to address the limitations of traditional microfinance models. The proposed model has the potential to increase transparency, reduce transaction costs, and enhance security in microfinance. However, the authors acknowledge that the proposed model may face challenges in implementation and regulatory compliance.

In this study [2], they introduced a blockchain-based conceptual model for sustainable microfinance outreach to farmers, where the microfinance institutions may dynamically change the loans they issue in response to the farmer's activities, reducing severe indebtedness and vulnerabilities and facilitating crop cultivation, improving farm outputs and revenue.

According to their research [3], with its paradigm of distributed, time-stamped ledgers, blockchain technology may successfully assist banks and financial institutions enhance their KYC process by permitting rapid and accurate real-time data interchange across many stakeholders for faster and more effective validation. The adoption of blockchain technology in microfinance makes credit assessment in the loan sanctioning process very simple and effective.

In this article [4], a distributed ledger system built on the blockchain that serves the demands of farmers is implemented. Through their system, microfinance institutions are not involved in the process of providing microcredits to the lower socioeconomic strata. As a result, it encourages both small and big investors to provide microcredits to the lower socioeconomic strata without sacrificing openness among all stakeholders, which is advantageous to both the investors and the farmers.

This study [5] explores how blockchain technology may be used to solve some of the major problems that the Social Business (SB) industry is now experiencing. A semi-formal modelling method using Blockchain technology was employed to simulate a modest sample of a micro-credit use-case from SB's microfinance activity.

This paper [6] first examines the operational difficulties with microcredit loans before looking at the possible applications of blockchain and mobile money. Due to the paucity of research in this field, a qualitative method is used, based on interviews with people with expertise in microfinance, mobile money, and blockchain technology.

[7] This research study's main goal is to persuade farmers to use microfinance in order to invest in their crops and to persuade farmers to make use of the funds provided by them without taking on debt. An investment offer will be made for the farmers' crop by confirming all preceding production.
The purpose of this conceptual paper [8] is to investigate the following research question: Can blockchain technology aid in the resolution of challenges usually connected with rural government and development?

The advantages and drawbacks of blockchain-based fintechs and cryptocurrencies aimed towards low-income households in developing nations are briefly discussed in this article [9].

They [10] suggested a distributed edge+cloud system with community-specific machine learning to help with microlending services. Microfinancing services are offered to support local companies and enable the community to develop into a healthy economy using a mix of technologies like microservices-based architecture, blockchain technology, and machine learning.

In the context of microfinance markets in developing nation, the goal of this study [11] is to examine how technology improvements reduce inefficiencies in marketing channels. The authors connect the research on marketing channel inefficiencies and technological innovation relevant to developing markets by closely investigating particular market inefficiencies that impede the attempts of micro and small businesses to access microfinance in emerging economies and the use of technology to ameliorate these shortcomings.

Investigating the behaviour of default prediction models based on credit scoring approaches and computational methods with machine learning algorithms is the major goal of this study [12].

[13] The paper provides a detailed description of the proposed credit rating system and its components, including the data collection process, machine learning algorithms, and blockchain implementation. The authors also evaluate the system's performance using simulated data and compare it to traditional credit rating systems. The results show that the proposed system outperforms traditional credit rating systems in terms of accuracy and fairness.

This paper [14] examines the early uses of AI in the emerging market financial services industry and discusses obstacles to developing market FSPs (financial service provider) using AI in a responsible and sustainable way. It also highlights what activities investors and development finance organisations like the IFC (International Finance Corporation) may take to guarantee that AI is used to maximise financial inclusion.

[15] The essay focuses on the local financial strategies used by small company owners in underrepresented neighbourhoods to help such companies become economically independent, thrive, and be resilient while also benefiting the neighbourhood to which they belong. They presented a microbanking-based concept, in which small company owners are transformed into managers of extremely tiny banks that give microcredit to their communities.

[16] This paper intends to investigate company finance trends and blockchain technology adoption in the agriculture sector. Numerous agricultural issues, including finance for agricultural businesses, can be resolved by the implementation of blockchain technology in terms of capturing, storing, verifying, and safeguarding data. The banking and insurance sectors might improve their credit ratings and profile models if they had real-time access to activity data from the agriculture sector.

They examined some of these issues in their paper [17], paying special attention to small-scale farming in Africa, and they highlighted some of the things that these farmers may need but are unable to obtain at the time of writing. The primary focus is on financial products that farmers cannot access, owing to a lack of eligible assets that banks recognise as collateral to reduce risk. Nonetheless, these farmers have claims on assets such as cattle, land, and harvests, which cannot be easily accepted as collateral. They scrutinized the possibilities for using blockchain technology to enable value transfers based on the assets that farmers cannot fully benefit from in the present banking system. With a focus on the regional circumstances in the African markets under discussion, they addressed some of the possible drawbacks of implementing this technology.

The absence of a verified customer credit history is a typical issue in emerging economies, thus this article [19] solves it by using non-traditional data from a Micro Finance Institution (MFI) in a Credit Scoring loan categorization problem. Using a real-world dataset, they conducted a series of experiments to construct a baseline model and demonstrate the use of node embedding features in credit scoring models.

With the help of alternative data from a mobile phone application for reporting agricultural operations, this study [20] suggests a credit determination tool. Over a five-year period, 41613 reports of agricultural activities from 11336 farmers were provided by 213 users. Based on personal information, connections to farms, and reports, users were classified into groups. Multiple logistic regression, a support vector machine with a linear kernel, and a support vector machine with a radial basis function kernel are used as the three analytical techniques.

The methodology for creating alternative scoring variables based on the demands of stakeholders is proposed in this study [21]. Data gathered from farmers in rural Cambodia via surveys and a mobile application is used as an example of how the suggested technique may be implemented. On the basis of data gathered and stakeholder requirements, alternative scoring variables are devised. These data are used to train and test multiple logistic regression and support vector machine models, which evaluate the chosen variables. The accuracy and values of the area under the receiver operating characteristics curve are used to compare models.

In order to combat fraud in financial institutions, this study [22] suggests a novel blockchain system design that enables the secure exchange of both current and future customer data. The National Microfinance Bank (NMB) of Jordan's clients' real-world data was used to develop and assess the suggested system architecture. The findings demonstrate that the system design has high performance in terms of reduced time needed to verify, upload, and append transactions and blocks to the blockchain system, as well as a good security level that makes it challenging for attackers to impersonate a genuine validator.

The purpose of this study [23] is to give a discourse on how blockchain technology may be used to facilitate crowdfunding. In this qualitative study, information from the relevant literature is analysed, and conclusions are drawn. The findings of this study showed that combining blockchain technology with crowdfunding is feasible and advantageous for all parties involved. This integration lowers transaction costs while also providing predictability and trust to the system and mitigating risk for the parties from a governance standpoint.

In addition to outlining the design and operational procedures of the smart agreement in detail, this article [25] suggests a private loaning platform that is entirely based on the blockchain era. Research shows that the flawless completion of the borrowing, compensation, and payback phases is made possible by the smart settlement implementation for personal loans with the blockchain.

By introducing a blockchain-based capital transfer system that intends to reduce obstacles to financial inclusion and offer financial services to the unbanked, this study [26] closes a gap in the state of the art. They outlined the system's benefits, its needs and objectives, and the design of the Everex financial eco-system.

The considerable difficulties and crucial flaws that the Indian government's current DBT cash transfer programme experienced have been explored. In order to enhance the current government's fund delivery system, the solution relies on blockchain-based smart contracts. In order to lessen the difficulties and significant problems experienced by Indian farmers, they have [27] developed and created a prototype model known as a single-window method based on blockchain technology. With the automated model, security and privacy are guaranteed, along with auditability, immutability, and transparency. So, with the help of Blockchain-based technology, the direct cash distribution prototype may be retooled and put into practise with a number of tangible advantages. Finally, they discussed the features of the robust system built with Blockchain smart contracts.

2.3 RESEARCH CHALLENGES

In order to verify rural people's credit history online and design a machine learning strategy, there is no reliable dataset accessible regarding their yearly credit expenditures and transactions.

Blockchain technology can potentially enable faster and cheaper transactions, but the scalability of Blockchain networks is a major concern. Microfinance institutions often handle large volumes of transactions, and the Blockchain network must be able to handle these volumes without compromising the system's performance.

Blockchain technology is still in its early stages. A comprehensive and thorough understanding of the blockchain is necessary to develop an application of this type.

entire process. The move from conventional techniques to blockchain may pose operational hazards. Many applications may require some system redundancy, such as running blockchain with conventional systems functioning as backup until sufficient stability is achieved to allow for a complete transfer.

In comparison to Polygon, which lacks a polygon testable network, Ethereum costs a higher transaction fee, according to the research. Before the technology is widely implemented, changes to existing regulations in areas such as asset transfer will be necessary. This might entail significant modifications to economic laws.

2.4 RESEARCH OBJECTIVES

To examine the current state of microfinance in rural India and identify the challenges faced by microfinance institutions (MFIs) in providing financial services to underserved populations. In addition, exploring the potential of machine learning (ML) and blockchain technology (BT) in addressing the challenges faced by MFIs and improving the efficiency and effectiveness of microfinance services in rural India.

Also, to identify the key success factors and challenges of implementing machine learning and blockchain technology in microfinance in rural India and provide recommendations for scaling up the intervention.

To contribute to the existing literature on microfinance, machine learning, and blockchain technology and provide insights for policymakers, practitioners, and researchers interested in using technology to enhance financial inclusion and rural development. design and develop a machine learning and blockchain-based microfinance platform tailored for the needs of rural communities in India.

Chapter 3

PROPOSED METHODOLOGY

3.1 EXPLANATION OF WORKFLOW

The first subproblem involves using different machine learning models to evaluate rural credit scores and selecting the best model among them. This can help in assessing the creditworthiness of rural inhabitants and determining their eligibility for financial services.

The second subproblem involves offering financial services using blockchain technology and crypto tokens to rural inhabitants in underdeveloped areas of society. The aim of this is to motivate them to act on their ideas and acquire the necessary financial resources to achieve self-sufficiency. This approach can help in addressing the lack of financial infrastructure and services in rural areas, enabling more people to access the resources they need to improve their economic conditions.

Overall, the proposed methodology combines advanced technology with financial services to create an innovative solution for promoting rural development and self-sufficiency.

The first module of this project utilizes a dataset sourced from open-source platform GitHub. Technical specifications for this module involve the use of Jupyter Notebook, although it can also be run on Google Colab. The module focuses on the features such as loan amount, tenure, and purpose of loan, among others, to predict the accuracy of loan approvals for micro credit users, treated as a regression problem.

Initially, the data undergoes visualization and feature engineering as primary steps. Following this, various classifier models such as Linear Regression, Decision Tree, and Random Forest and others are trained on the data, with their accuracy compared to identify the best machine learning model for predicting loan approvals which can help in streamlining the loan application process of rural people more easily.

The second subproblem of the project involves creating an application over a blockchain that facilitates crypto token transactions for rural populations, allowing them to buy, trade the tokens and use as digital assets like other securities. This application is built using the Motoko language and user authentication is carried out using Internet Identity, which has secured access makes our application to be most secured.

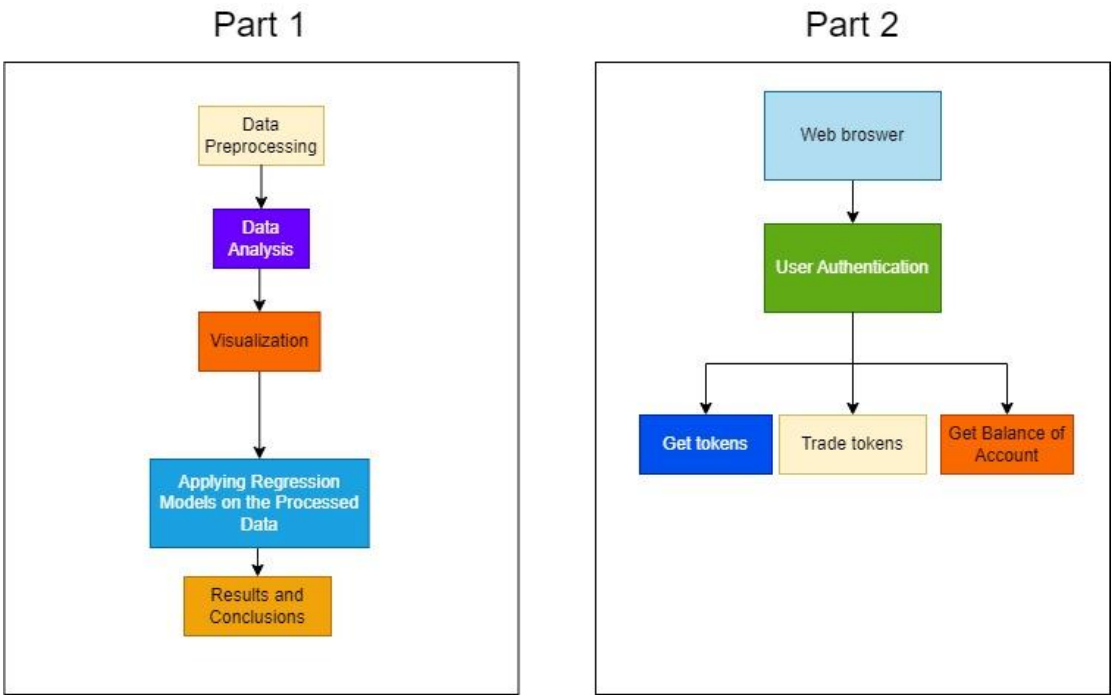


FIGURE 1 – Proposed methodology workflow

3.2 DATASET EXPLANATION

The dataset comprises 19 primary features, with 10 being numerical data such as age, annual income, monthly expenses, old dependents, young dependents, and the remaining 9 being categorical data such as city, sex, social class, primary business, secondary business, type of house, etc.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
Id	city	age	sex	social_clas	primary_bı	secondary	annual_inc	monthly_e	old_depen	young_dep	home_ow	type_of_h	occupants	house_are	sanitary_a	water_ava	loan_purp	loan_tenu	loan_insta	loan_amount	
1	Dhanbad	22	F	Mochi	Tailoring	Others	36000	5000	0	2	1	R	4	70	1	0.5	Apparels	12	12	5000	
2	Manjapra	21	F	OBC	Tailoring	none	94000	3600	1	1	1	T1	4	80	1	0.5	Apparels	12	50	7500	
3	Dhanbad	24	M	Nai	Beauty sal	Others	48000	4000	0	2	1	T1	4	50	1	0.5	Beauty Sal	12	12	5000	
4		26	F	OBC	Tailoring	none	7000	5000	0	2	1	T1	5	50	1	0.5	Apparels	12	50	7500	
5	Nuapada	23	F	OBC	General st	Agriculture	36000	3500	0	0	1	T1	1	112	1	0.5	Retail Stor	12	12	5000	
6	Nuapada	23	F	OBC	General st	none	36000	3500	0	0	1	T1	1	112	1	0.5	Retail Stor	12	12	5000	
7	Dhanbad	22	F	Muchi	Tailoring	Others	36000	3000	0	1	1	T1	3	60	1	0.5	Apparels	12	12	5000	
8		28	F	OBC	Tailoring	none	7000	5000	0	2	1	T1	5	40	1	0.5	Apparels	12	50	7500	
9	Dhanbad	38	F	Muchi	Puffed rice	Others	36000	5000	0	2	1	T1	5	50	1	0.5	Eateries	12	12	5000	
10	Dhanbad	25	F	Muchi	General st	Others	36000	5000	0	3	1	R	5	50	1	0.5	Retail Stor	12	12	5000	
11		25	F	OBC	Poultry far	Daily wage	40000	3000	0	0	1	T1	1	110	1	0.5	Meat Busi	12	12	5000	

FIGURE 2 – Attributes of Dataset

The attribute 'city' indicates the location where the user lives, 'primary_business' represents the user's main activity for generating income, and 'secondary_business' indicates their part-time work. The attribute 'home_ownership' takes the value 1 if the home is owned by the user and 0 if it is not. The 'type of house' attribute has three categories: R, T1, and T2, where R represents a single room, T1 represents a two-room house, and T2 represents a three-room house. 'Sanitary_availability' is a binary attribute that takes the value 1 if the facility is available and 0 if it is not. 'Water_availability' is also a categorical attribute with values such as 0, 0.5, and 1. Lastly, the attribute 'loan_purpose' represents the purpose for which the loan is being applied.

3.3 DATA PREPROCESSING AND RESULTS

3.3.1 DATA PREPROCESSING

Data preprocessing is a crucial step in this process, and it begins with loading the dataset into the Jupyter notebook for manipulation. As shown in Figure 3, the dataset is printed to ensure that the correct data has been loaded.

```
ruralDataset
```

	Id	city	age	sex	social_class	primary_business	secondary_business	annual_income	monthly_expenses	old_dependents	...	home_ownershi
0	1	Dhanbad	22	F	Mochi	Tailoring	Others	36000.0	5000.0	0	...	1.
1	2	Manjapra	21	F	OBC	Tailoring	none	94000.0	3600.0	1	...	1.
2	3	Dhanbad	24	M	Nai	Beauty salon	Others	48000.0	4000.0	0	...	1.
3	4	NaN	26	F	OBC	Tailoring	none	7000.0	5000.0	0	...	1.
4	5	Nuapada	23	F	OBC	General store	Agriculture	36000.0	3500.0	0	...	1.
...	...	...	...	...	...	...	...	...	...	...	...	.
39995	39996	Pusad	45	F	Muslim	Buffalo rearing	none	78000.0	4000.0	0	...	1.
39996	39997	Pusad	35	F	ST	Tailoring	none	48000.0	4000.0	0	...	1.
39997	39998	Pusad	35	F	Sc	Goat rearing	none	48000.0	3500.0	0	...	1.
39998	39999	Pusad	28	F	Sc	Goat rearing	none	48000.0	4000.0	0	...	1.
39999	40000	Pusad	32	F	Sc	Goat rearing	none	72000.0	4500.0	0	...	1.

40000 rows × 21 columns

FIGURE 3.1 – Loading Dataset

```
ruralDataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40000 entries, 0 to 39999
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Id                    40000 non-null  int64
 1   city                  38136 non-null  object
 2   age                   40000 non-null  int64
 3   sex                   40000 non-null  object
 4   social_class          34745 non-null  object
 5   primary_business      39974 non-null  object
 6   secondary_business    34759 non-null  object
 7   annual_income         40000 non-null  float64
 8   monthly_expenses      39880 non-null  float64
 9   old_dependents        40000 non-null  int64
 10  young_dependents      40000 non-null  int64
 11  home_ownership        39621 non-null  float64
 12  type_of_house         39306 non-null  object
 13  occupants_count       40000 non-null  int64
 14  house_area            40000 non-null  float64
 15  sanitary_availability 39792 non-null  float64
 16  water_availabity      34747 non-null  float64
 17  loan_purpose          39974 non-null  object
 18  loan_tenure           40000 non-null  int64
 19  loan_installments     40000 non-null  int64
 20  loan_amount           40000 non-null  float64
dtypes: float64(7), int64(7), object(7)
memory usage: 6.4+ MB
```

FIGURE 3.2 – Dataset info

The subsequent step involved determining the data types of the dataset, identifying any null values within the entire dataset, and examining the unique values of features like age. During this process, we discovered inconsistencies in the age column of the dataset, and as depicted in the figure 4, we proceeded to remove the erroneous age values.

As you can see we have found few irregular age values as 2,205,766105,288.Hence droping the age values and replacing them with mean

```
ageoutlier1=ruralDataset.loc[ruralDataset['age']==2]
ageoutlier2=ruralDataset.loc[ruralDataset['age']==205]
ageoutlier3=ruralDataset.loc[ruralDataset['age']==288]
ageoutlier4=ruralDataset.loc[ruralDataset['age']==766105]

drop2 = ruralDataset[(ruralDataset.age ==2)].index
drop205 = ruralDataset[(ruralDataset.age ==205)].index
drop288 =ruralDataset[(ruralDataset.age==288)].index
drop766105=ruralDataset[(ruralDataset.age==766105)].index

#deleting three records
updatedDataset=ruralDataset.drop(drop2)
updatedDataset=updatedDataset.drop(drop205)
updatedDataset=updatedDataset.drop(drop288)
updatedDataset=updatedDataset.drop(drop766105)


checkage = updatedDataset["age"].unique()
print(checkage)
# avg_age = updatedDataset["age"].astype('float').mean(axis=0)
# print("Average of Age:", avg_age)
```

```
[22 21 24 26 23 28 38 25 34 37 50 54 57 48 32 35 43 51 39 41 46 45 44 47
 40 42 33 53 30 29 36 19 27 20 55 49 31 52 58 56 18 59 60 61 63 64 62 65
 67 70 74 69 66 72 68 82 88 75]
```

FIGURE 4 – Updating dataset without erroneous age values

3.3.2 RESULTS

The first visualization is a bar chart created using the Matplotlib libraries, categorizing the loan requested based on the gender which was illustrated in figure 5.

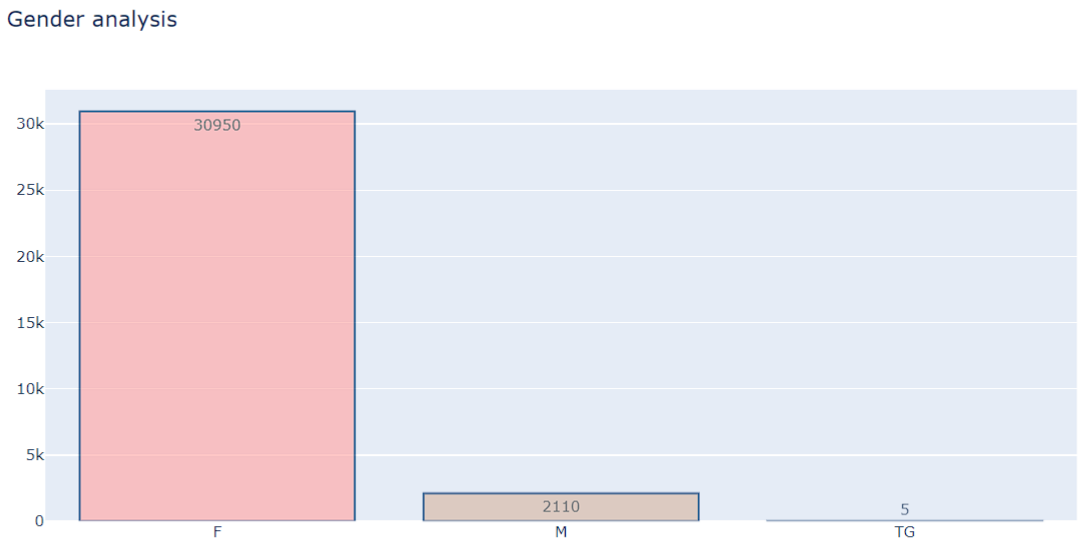


FIGURE 5 – Bar Chart on Loan Requested based on their gender

Generating a bar plot using Seaborn library with the x-axis representing the loan tenure, the y-axis representing the loan installments, and the data being taken from the "graphdf1" dataframe. Then, the sns.barplot() function is called to generate the bar plot. The plt.xticks() function is used to rotate the x-axis labels by 90 degrees. The plt.xlabel() and plt.ylabel() functions are used to set the labels for the x-axis and y-axis respectively.

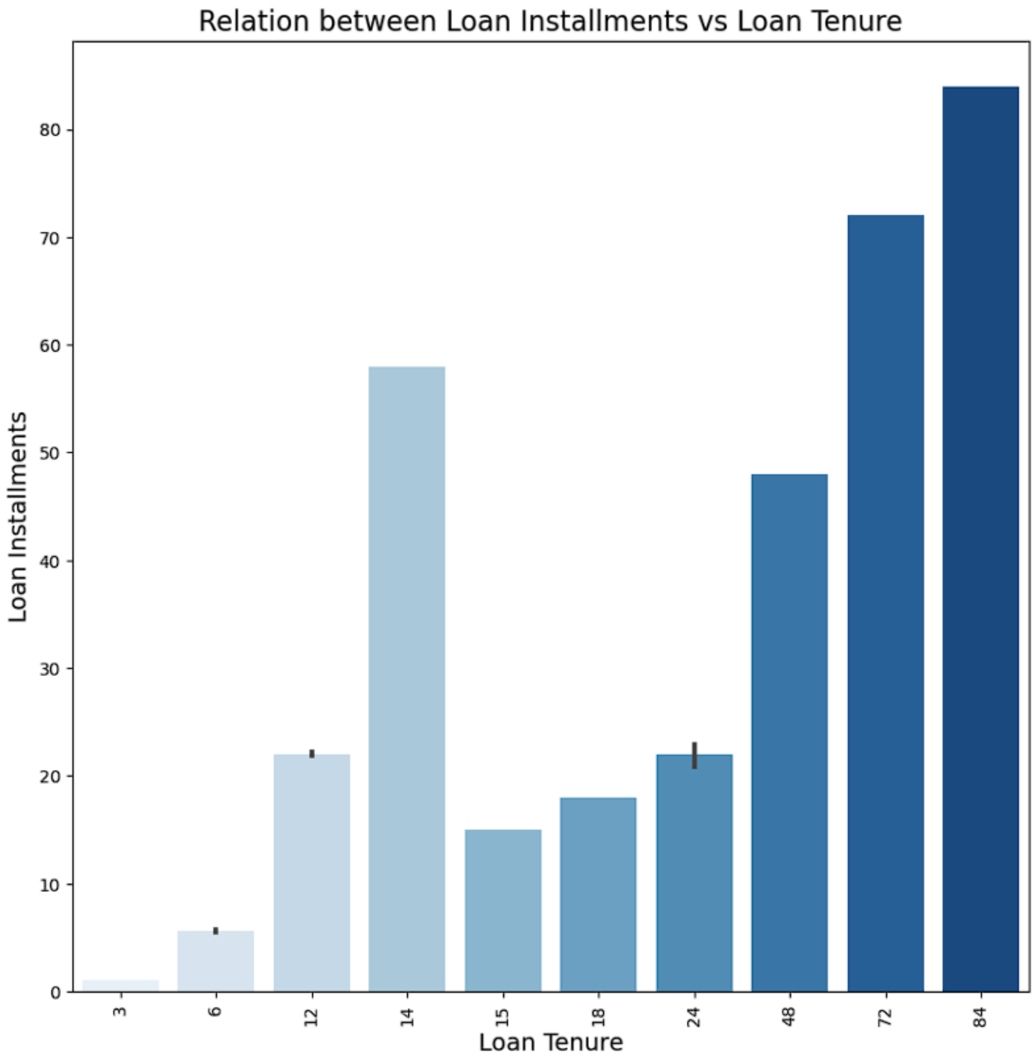


FIGURE 6 – Loan Installments vs loan tenure

Figure 7 defines two scatter plot objects, t1 and t2, using go.Scatter(). Both scatter plots plot the same x-axis data, which is the "age" column of df. The y-axis data for t1 is the "annual_income" column of df, while the y-axis data for t2 is the "monthly_expenses" column of df.

Each scatter plot object has a different name, color, and plot mode. The name and color are used to differentiate between the scatter plots in the plot legend. The plot mode is set to "lines + markers" to plot the data points as markers and connect them using lines.

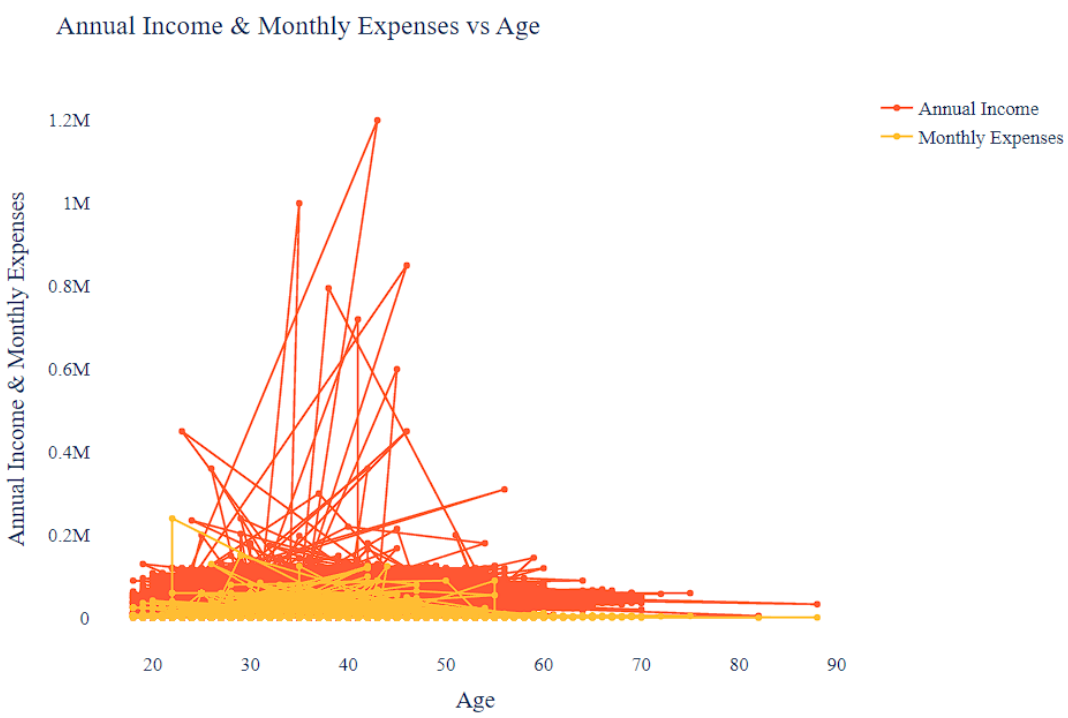


FIGURE 7 – Annual Income & Monthly expenses vs Age

Creating a scatter plot with a regression line between the age and annual income columns of the dataset using the seaborn library. The sns.regplot() function is used to create the scatter plot with a regression line, where x is set to the age column and y is set to the annual income column. Finally, sns.despine() is used to remove the top and right spines of the plot and plt.show() is used to display the plot as shown in fig 8.

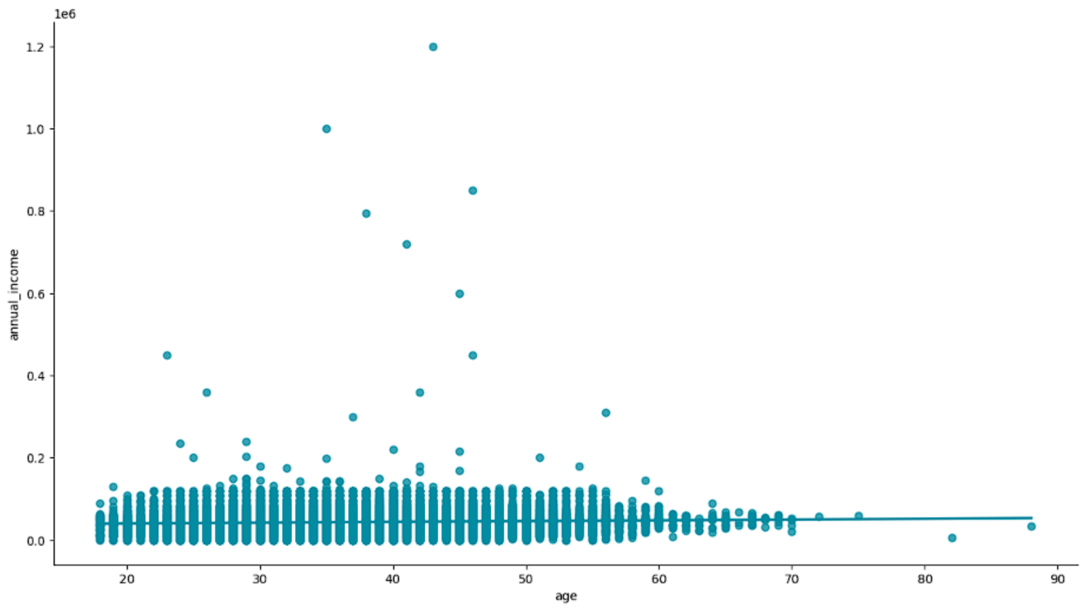


FIGURE 8 – Regression line with scatter plot between age and annual income

From Plotly Express library a scatter plot is visualized in Figure 9 using the px.scatter() function with the x-axis representing the loan tenure and y-axis representing the loan amount. The color and size of the dots in the plot are determined by the loan amount as well.

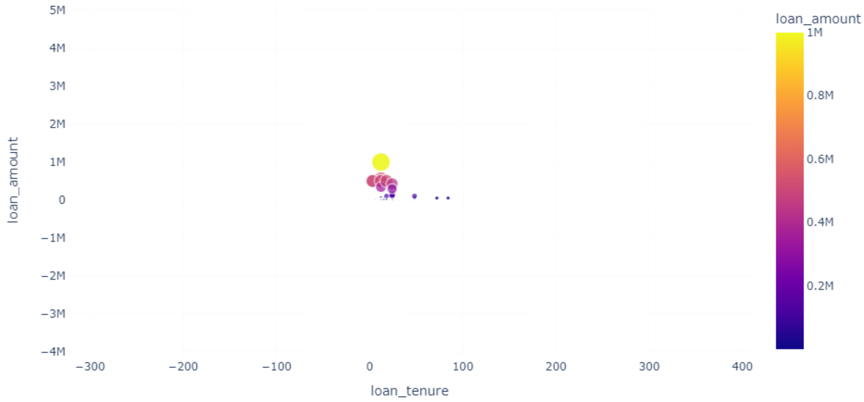


FIGURE 9 – Scatter plot of loan tenure vs loan amount

Contrived a joint plot using the seaborn library. The kind="reg" parameter specifies that a linear regression line should be included in the plot. The resulting plot will have the annual income on the y-axis and loan amount on the x-axis.

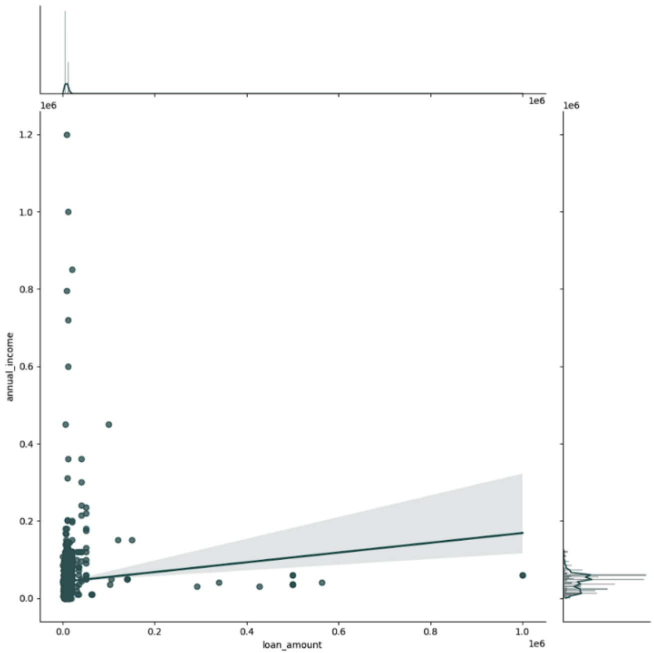


FIGURE 10 – Joint plot of Annual Income and Loan amount

The plot in Figure 11 includes a regression line to show the direction and strength of the linear relationship between these variables.

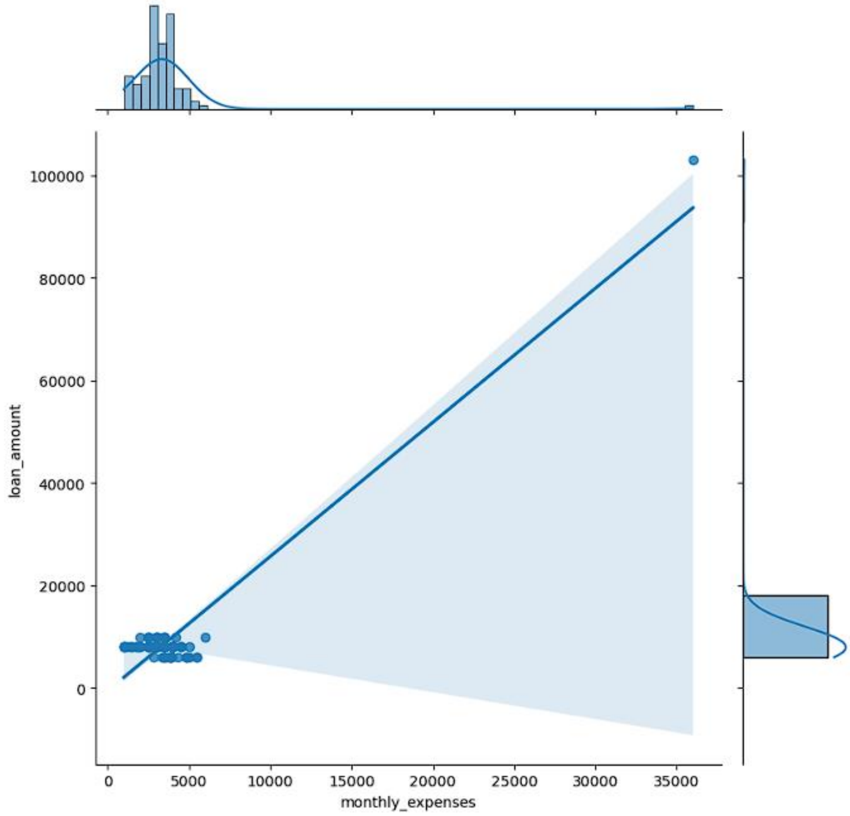


FIGURE 11 – Joint plot of Linear relationship between loan amount and monthly expenses

The first section uses the Counter function from the collections module to count the number of occurrences of each unique loan purpose in the Series. The unique loan purposes are then used as labels for a pie chart, with the count of each loan purpose represented as a segment of the chart. This pie chart in fig 12 a) is created using the pie function from Matplotlib.

The second section uses the px.pie function from Plotly Express to create another pie chart showing the distribution of loan purposes as shown in 12 b) for transparent and clear depiction of ratios in pie chart.

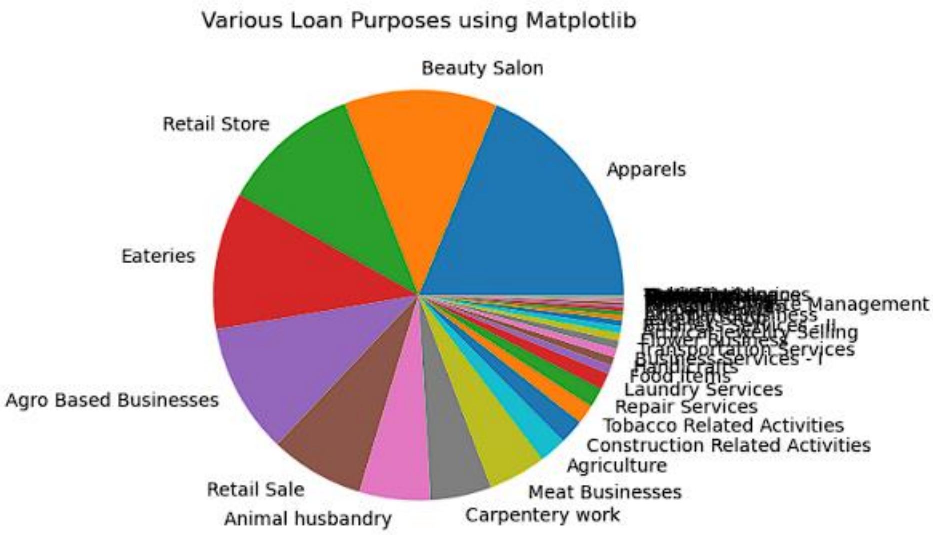


FIGURE 12 a) – Pie Chart of Loan Purposes using Matplotlib Library

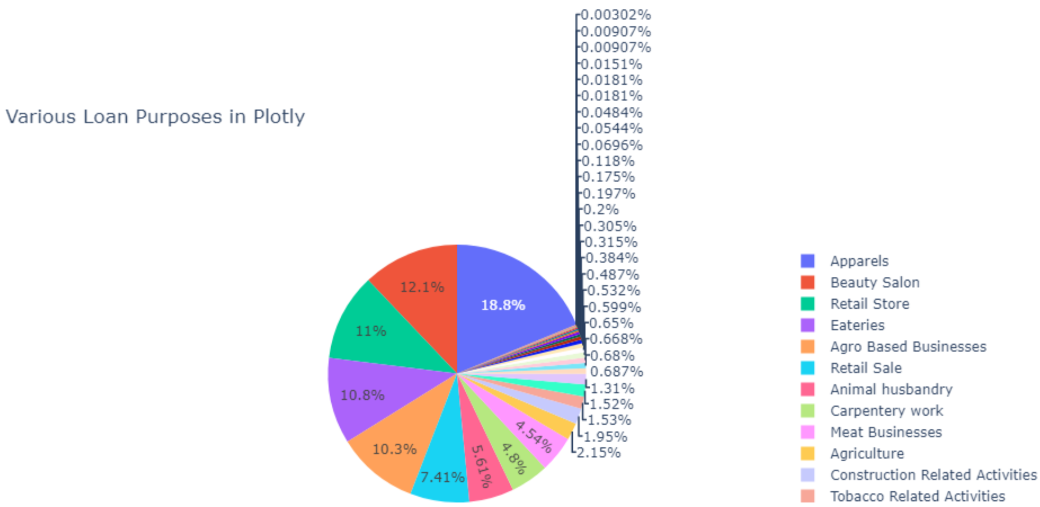


FIGURE 12 b) – Pie Chart of Loan Purposes using Plotly library

The columns monthly expenses and age from dataset are first chosen, and then, in Python, the Seaborn module is used to produce a visualization of the violin plot as shown in figure 13.

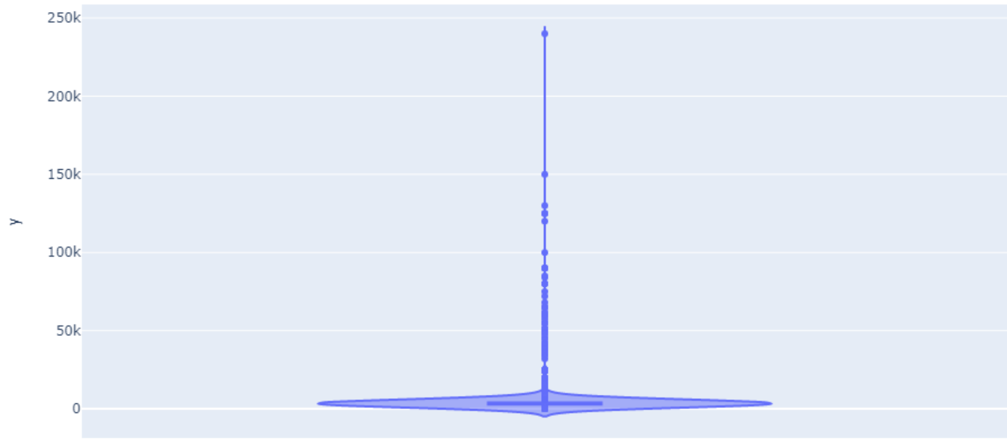


FIGURE 13 – Violin Plot of Monthly expenses and Age

Using the plotly library, a box plot is presented as shown in fig 14. The box plot object with the x property set to the monthly expenses column, to get the distributions of numeric values of monthly expenses. The plot is displayed using plotly's iplot function.

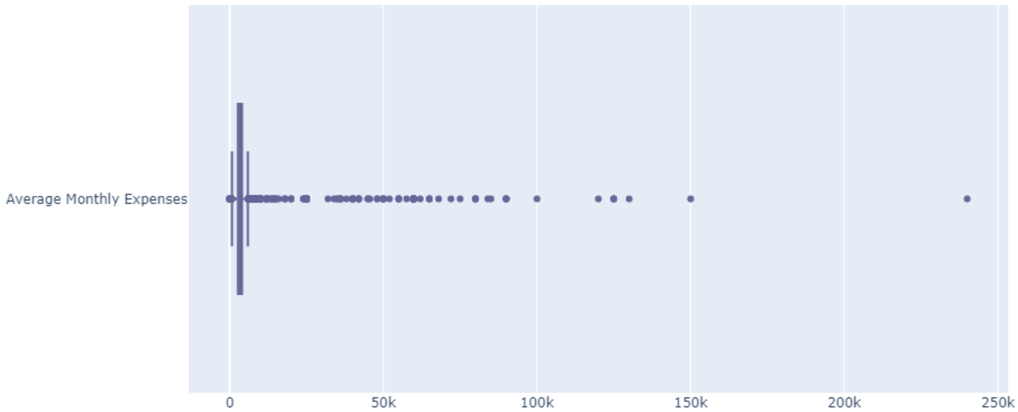


FIGURE 14 – Box plot of monthly expenses

The sns.heatmap function from seaborn is imported to make a heat map that shows the correlation values between the top correlated features in dataset as shown in Figure 15.

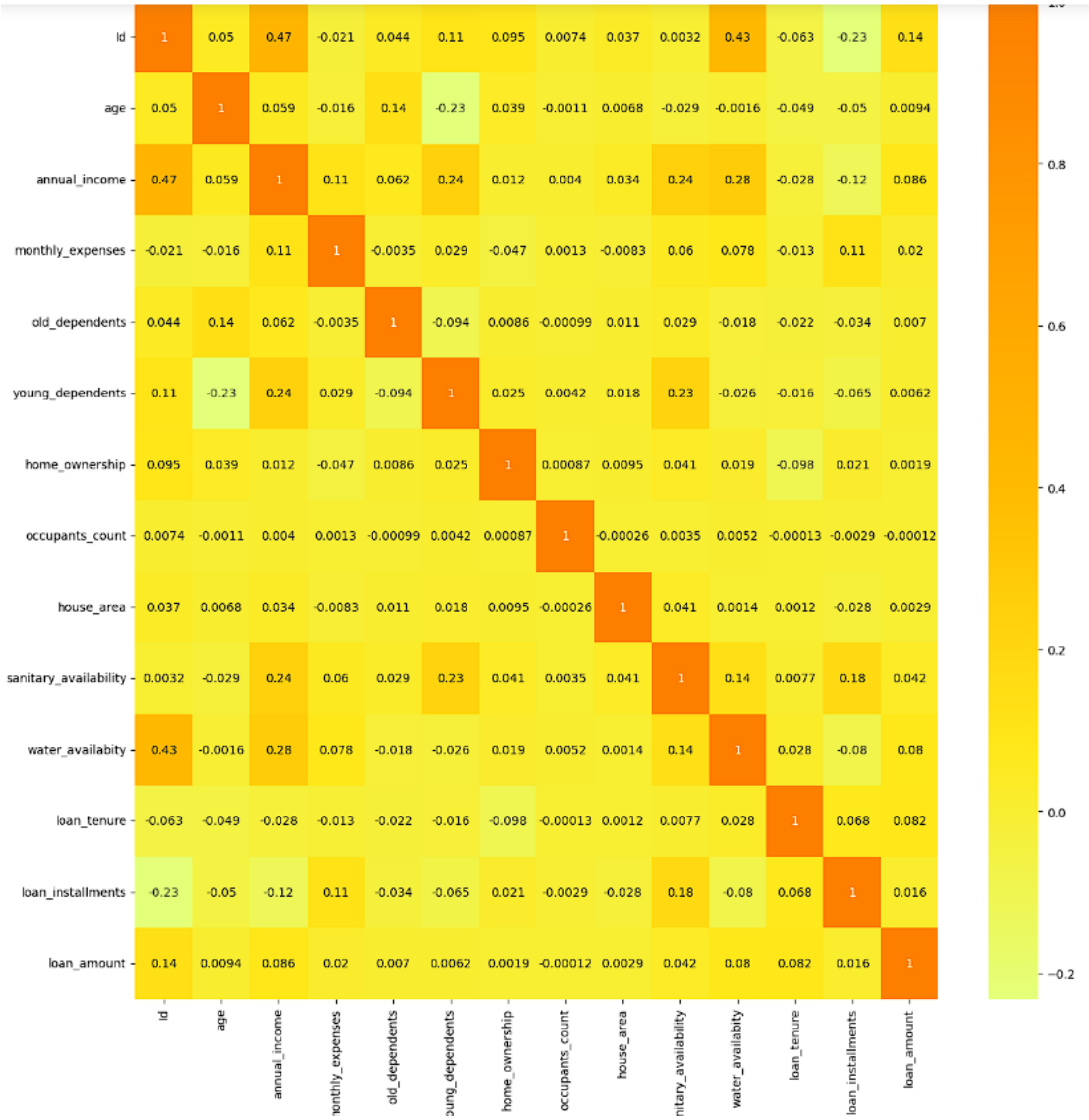


FIGURE 15 – Heatmap of correlated values

Chapter 4

EXPERIMENTAL RESULTS EXPLANATION

4.1 HOT ENCODING

To enhance the predictive capabilities of the machine learning models applied to the dataset and make them best suited, we are making the data more manageable. The code snippet in figure 16 is used to preprocess data in a Pandas DataFrame named 'preprocessData'. It performs data imputation by filling the missing values with the mode (for categorical variables) or mean (for numerical variables) of the respective columns.

```
preprocessData['social_class']=preprocessData['social_class'].fillna(preprocessData['social_class'].mode()[0])
preprocessData['city']=preprocessData['city'].fillna(preprocessData['city'].mode()[0])
preprocessData['primary_business']=preprocessData['primary_business'].fillna(preprocessData['primary_business'].mode()[0])
preprocessData['secondary_business']=preprocessData['secondary_business'].fillna(preprocessData['secondary_business'].mode()[0])
preprocessData['type_of_house']=preprocessData['type_of_house'].fillna(preprocessData['type_of_house'].mode()[0])
preprocessData['sanitary_availability']=preprocessData['sanitary_availability'].fillna(preprocessData['sanitary_availability'].mc
preprocessData['water_availabity']=preprocessData['water_availabity'].fillna(preprocessData['water_availabity'].mode()[0])
preprocessData['loan_purpose']=preprocessData['loan_purpose'].fillna(preprocessData['loan_purpose'].mode()[0])
preprocessData['monthly_expenses']=preprocessData['monthly_expenses'].fillna(preprocessData['monthly_expenses'].mean())
preprocessData['home_ownership']=preprocessData['home_ownership'].fillna(preprocessData['home_ownership'].mode()[0])
```

FIGURE 16 – Data imputation of dataset

The code fills missing values in the following columns:

'social_class'

'city'

'primary_business'

'secondary_business'

'type_of_house'

'sanitary_availability'

'water_availabity'

'loan_purpose'

'monthly_expenses'

'home_ownership'

For categorical variables, the mode (most common value) of each column is used to fill the missing values using the 'fillna()' method. For numerical variables, the mean of each column is used to fill the missing values.

Overall, this code is now properly handled for modeling by ensuring that missing values are handled appropriately. Thereafter, we created a new Pandas DataFrame called 'dataforpred' by selecting a subset of columns from the 'preprocessData' DataFrame. The columns that are selected are specified within the 'columns' parameter as a set of column names.

The selected columns are: 'age', 'sex', 'annual_income', 'monthly_expenses', 'old_dependents', 'young_dependents', 'home_ownership', 'type_of_house', 'occupants_count', 'house_area', 'loan_tenure', 'loan_installments', 'loan_amount'.

Further we transformed and combined the data in the 'dataforpred' DataFrame by creating new columns that encode categorical variables as binary indicator variables as shown in Figure 17. This process is known as one-hot encoding and is often used to convert categorical variables to a format that can be used in machine learning models.

In [14]:
```
dataforpred=pd.DataFrame(preprocessData,columns={"age","sex","annual_income",
                                                  "monthly_expenses","old_dependents","young_dependents","home_ownership","type_of_house",
                                                  "occupants_count","house_area","loan_tenure","loan_installments","loan_amount"})
dataforpred.head()
```

Out[14]:

	house_area	loan_amount	monthly_expenses	occupants_count	type_of_house	old_dependents	sex	loan_tenure	loan_installments	home_ownership	annua
0	70.0	5000.0	5000.0	4	R	0	F	12	12	1.0	
1	80.0	7500.0	3600.0	4	T1	1	F	12	50	1.0	
2	50.0	5000.0	4000.0	4	T1	0	M	12	12	1.0	
3	50.0	7500.0	5000.0	5	T1	0	F	12	50	1.0	
4	112.0	5000.0	3500.0	1	T1	0	F	12	12	1.0	

In [15]:
```
data1=pd.DataFrame(dataforpred,columns={"type_of_house","sex"})
data2= pd.get_dummies(data1.type_of_house,prefix='type_of_house',drop_first=True)
```

In [16]:
```
data3 = pd.get_dummies(data1.sex, prefix='sex',drop_first=True)
```

In [17]:
```
data4= pd.concat([data2, data3],axis=1)
data4.head()
```

Out[17]:

	type_of_house_T1	type_of_house_T2	sex_M	sex_TG
0	0	0	0	0
1	1	0	0	0
2	1	0	1	0
3	1	0	0	0
4	1	0	0	0

FIGURE 17 – Performing hot encoding on selected

Further, performing feature selection using the ExtraTreesRegressor model from the scikit-learn library. Calculating the feature importance scores for each of the features using model.feature_importances_. The resulting feature importance scores are stored in a pandas Series. Thus, selecting the top 15 most important features by calling the nlargest() method on the important_features Series and passing in a parameter of 15. Finally, it plots these top 15 most important features in a horizontal bar graph as shown in figure 18.

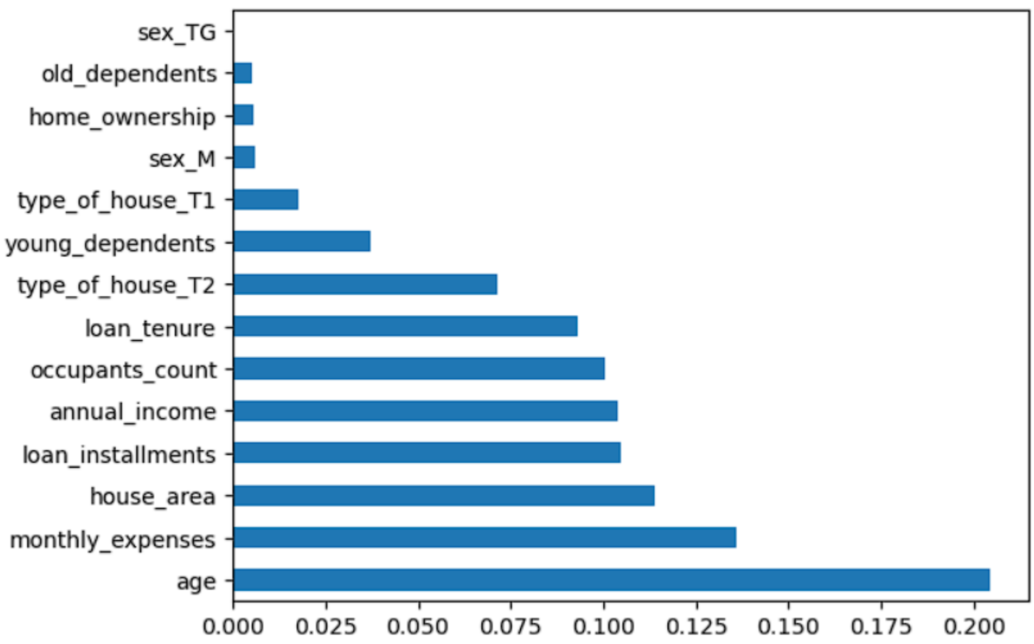


FIGURE 18 – Visualizing the top 15 features from the dataset

4.2 LINEAR REGRESSION ALGORITHM RESULT

Build a linear regression model to predict loan amounts based on input features. First, it imports the Linear Regression model from the sklearn.linear_model module. Then, it creates an instance of the model. The code then fits this linear regression model to the training data X_train and y_train using the fit() method.

After that, it predicts the loan amounts for the test data using the predict() method on the trained model, and the resulting predictions are stored in the variable.

```
df_linearregressor_results = pd.concat([linearregressor_data_prediction,y_samples],axis=1)
df_linearregressor_results.head(10)
```

	Predicted Loan Amount	loan_amount
0	7939.000459	8000.0
1	9633.556556	15000.0
2	6530.560403	3000.0
3	7942.395120	10000.0
4	7295.899493	8000.0
5	10252.634928	8000.0
6	11758.390860	8000.0
7	7141.211232	8000.0
8	9595.567272	12000.0
9	7163.242060	4000.0

FIGURE 18 – Linear Regression

4.3 DECISION TREE ALGORITHM RESULT

The model will use mean squared error as the measure of quality to evaluate splits during the construction of the decision tree. The model is then fit to the training data, and used to make predictions on the test data. The resulting predicted values are stored in a pandas Dataframe for further analysis.

```
df_decisiontreeregressor_results = pd.concat([decisiontreeregressor_data_prediction,y_samples],axis=1)
df_decisiontreeregressor_results.head(10)
```

	Predicted Loan Amount	loan_amount
0	12000.000000	8000.0
1	10000.000000	15000.0
2	3666.666667	3000.0
3	8000.000000	10000.0
4	5000.000000	8000.0
5	8000.000000	8000.0
6	8000.000000	8000.0
7	8000.000000	8000.0
8	15000.000000	12000.0
9	7500.000000	4000.0

FIGURE 19 – Decision Tree

4.4 RANDOM FOREST ALGORITHM RESULT

Random forest regression is a type of ensemble learning method that combines multiple decision tree models to improve the prediction performance.

In this code, an instance of the RandomForestRegressor class is created with the default hyperparameters, and then fit to the training data (X_train, y_train). After that, the model is used to generate predictions on the test data (X_test), and the predicted values are stored in a pandas DataFrame. The resulting DataFrame is used to analyze the model's prediction.

```
df_randomforestregressor_results = pd.concat([randomforest_data_prediction,y_samples],axis=1)
df_randomforestregressor_results.head(10)
```

	Predicted Loan Amount	loan_amount
0	10802.500000	8000.0
1	11385.000000	15000.0
2	3635.510823	3000.0
3	7790.000000	10000.0
4	7962.642857	8000.0
5	7832.000000	8000.0
6	11993.333333	8000.0
7	7342.500000	8000.0
8	12531.000000	12000.0
9	7350.000000	4000.0

FIGURE 20 – Random Forest Regression

4.5 KNN ALGORITHM RESULT

In regression tasks, KNN predicts the output value of a new observation based on the output values of its k nearest neighbors in the training data.

Then, the fit method is called with X_train and y_train as arguments, which trains the model on the training data.

Next, the predict method is called on the created instance of K- Neighbors Regressor with X_test as an argument, which makes predictions on the test data.

```
df_knnregressor_results = pd.concat([knnregressor_data_prediction,y_samples],axis=1)
df_knnregressor_results.head(10)
```

	Predicted Loan Amount	loan_amount
0	9857.142857	8000.0
1	10000.000000	15000.0
2	3428.571429	3000.0
3	7428.571429	10000.0
4	6857.142857	8000.0
5	8071.428571	8000.0
6	15000.000000	8000.0
7	6000.000000	8000.0
8	12857.142857	12000.0
9	6071.428571	4000.0

FIGURE 21 – K-Neighbors Regression

4.6 GRADIENT BOOSTING ALGORITHM RESULT

Import the GradientBoostingRegressor class from the Scikit-Learn library and the ensemble module. It then defines a dictionary params that specifies the hyperparameters to be used for the Gradient Boosting model, including the number of estimators, the maximum depth of the trees, the learning rate, and the loss function (squared error). The values of number of estimators is set to 300, the maximum depth of the trees is set to 20, and the learning rate is set to 0.02 respectively.

```
df_gradientboostregressor_results = pd.concat([gradientboostregressor_data_prediction,y_samples],axis=1)
df_gradientboostregressor_results.head(10)
```

	Predicted Loan Amount	loan_amount
0	10846.161195	8000.0
1	11171.687648	15000.0
2	3677.073527	3000.0
3	8094.770700	10000.0
4	8121.743035	8000.0
5	7738.696814	8000.0
6	11733.878937	8000.0
7	7833.414568	8000.0
8	11464.295743	12000.0
9	7499.512846	4000.0

FIGURE 22 – Gradient Boosting Regression

4.7 COMPARISON OF REGRESSION MODEL PERFORMANCE METRICS ACROSS ALGORITHMS

MSE, MAE, and RMSE are common metrics used to evaluate the performance of regression models.

Mean Squared Error (MSE): MSE measures the average squared difference between the predicted and true values. It is calculated by taking the average of the squared differences between the predicted and true values.

Mean Absolute Error (MAE): MAE measures the average absolute difference between the predicted and true values. It is calculated by taking the average of the absolute differences between the predicted and true values.

Root Mean Squared Error (RMSE): RMSE measures the square root of the average squared difference between the predicted and true values. It is calculated by taking the square root of the MSE.

ALGORITHM	ACCURACY	MAE	MSE	RMSE
Linear Regression	53.324	3463.824	9881007431.738	99403.256
Decision Tree	73.511	2054.410	187566905.810	13695.506
Random Forest	66.244	1856.346	118393108.234	10880.859
KNN	73.852	2098.395	126577202.198	11250.653
Gradient Boost	75.343	1860.653	185654585.004	13625.512

Chapter 5

DECENTRALIZED APPLICATION OVER BLOCKCHAIN NETWORK

Using Motoko, we can create a decentralized application that enables the transfer of crypto tokens between users. With the right design, this application can be extended to provide other digital assets and services that are useful for rural communities has the potential to transform the way people interact with digital assets and services. Motoko is designed to be highly efficient than solidity, vyper and optimized for resource utilization on the Internet Computer.

The transactional costs of Ethereum and other similar blockchain providers can be relatively high due to their consensus mechanisms and the associated gas fees. Gas fees are the fees paid by users to execute transactions on the blockchain, and they are determined by the amount of computing resources required to complete the transaction. As the demand for transactions on the blockchain increases, so do the gas fees, making it more expensive to use the blockchain.

Dfinity Internet Computer is a decentralized blockchain platform that aims to address some of the scalability and transactional cost issues associated with other blockchains. The Internet Computer uses a unique consensus mechanism called the Chain Key Technology (CKT) that enables it to scale more efficiently and handle more transactions per second than other blockchains. Additionally, the Internet Computer's token, called the ICP token, is used to pay for computation and storage on the platform, rather than gas fees, which can help to reduce transactional costs for users.

By using Dfinity Internet Computer, you may be able to take advantage of its higher scalability and more efficient transaction processing, as well as potentially lower transactional costs compared to other blockchain providers. For the aforementioned reasons, we have chosen to work with Dfinity Internet Computer over other blockchain providers.

One approach is to create a token that is specific to the needs of rural communities. For example, you could create a token that can be used to purchase agricultural products or services, such as seeds, fertilizers, or land preparation services. The token could be traded between farmers and suppliers, enabling a more efficient and transparent marketplace.

To create this application, we need to define the token contract and implement the transfer function using Motoko. We would also need to create a user interface that enables users to interact with the contract and transfer tokens between accounts.

In this approach we are employing Internet Identity authentication and avoiding regular password sign-ins to reduce the risk of data breaches and unauthorized access.

In the application development for the frontend ReactJS is used, as it provides a fast and efficient way to build complex user interfaces.

Chapter 6

IMPLEMENTATION

6.1 DATA ANALYSIS AND VISUALIZATION

Empowering Rural India through effective Microfinance by implementing Machine Learning & Blockchain Technology.

Importing required Libraries

In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing
import seaborn as sns
import re
import matplotlib.pyplot as plt
%matplotlib inline

import chart_studio.plotly as py
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go

import os
```

Loading Dataset

In [2]:
```
ruralDataset = pd.read_csv("trainingData.csv")
```

Analysis of the dataset

Dropping columns that have atleast 1 null or missing value

In [17]:
```
datasetforanalysis = ruralDataset.dropna(axis = 0, how ='any')
datasetforanalysis.shape
```

Out[17]:
```
(33065, 21)
```

In [18]:
```
from matplotlib import pyplot as plt

graphdf1 = datasetforanalysis.copy()
personalinfo = pd.DataFrame(graphdf1,columns={"city","age","sex","social_class"})
print(personalinfo["sex"].value_counts())
x= ['F','M','TG']
y=[30950,2110,5]
graphcolors = ['#FFACAC', '#D8BCAB','#F8F988']
fig = go.Figure(data=[go.Bar(
            x=x, y=y,
            text=y,
            textposition='auto',
        )])

fig.update_layout(dict(title="Gender analysis"), {'plot_bgcolor': 'rgba(0, 0, 0, 0)','paper_bgcolor': 'rgba(0, 0, 0, 0)'})
fig.update_traces(marker_color=graphcolors, marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5, opacity=0.7)
fig.show()
```

```
F     30950
M      2110
TG        5
Name: sex, dtype: int64
```

```python
import plotly.graph_objs as go

df = datasetforanalysis.copy()

t1 =go.Scatter(
    x =df.age,
    y = df.annual_income,
    mode ="lines + markers",
    name = "Annual Income",
    marker = dict(color = "rgb(255, 87, 51,0.5)")
)
t2 = go.Scatter(
    x = df.age,
    y = df.monthly_expenses,
    mode = "lines + markers",
    name = "Monthly Expenses",
    marker = dict( color = "rgb(255, 189, 51,0.7)")
)


graph3=[t1,t2]
layout = dict(autosize=False,width=1000, height=700,
              title = " Annual Income & Monthly Expenses vs Age",
              hovermode = "x",xaxis = {'showgrid' : False},yaxis = {'showgrid' : False},
             paper_bgcolor='rgba(0,0,0,0)',plot_bgcolor='rgba(0,0,0,0)',
             xaxis_title="Age",
             yaxis_title="Annual Income & Monthly Expenses",
        font=dict(
        family="Monserrat, Serif",
        size=18,

    ))

figure3 = dict (data = graph3 , layout = layout)

iplot(figure3)
```

6.2 REGRESSION MODELS

In [70]:
```python
data5= pd.concat([dataforpred, data4],axis=1)
data5.head()
```

Out[70]:

	loan_installments	loan_amount	sex	monthly_expenses	age	loan_tenure	house_area	home_ownership	occupants_count	old_dependents	young_depender
0	12	5000.0	F	5000.0	22	12	70.0	1.0	4	0	
1	50	7500.0	F	3600.0	21	12	80.0	1.0	4	1	
2	12	5000.0	M	4000.0	24	12	50.0	1.0	4	0	
3	50	7500.0	F	5000.0	26	12	50.0	1.0	5	0	
4	12	5000.0	F	3500.0	23	12	112.0	1.0	1	0	

In [71]:
```python
newDataforpred=pd.DataFrame(data5,columns={"old_dependents","loan_installments","loan_tenure","home_ownership",
                    "young_dependents","monthly_expenses","annual_income","age","house_area",
                    "occupants_count","type_of_house_T1","type_of_house_T2","sex_M","sex_TG",})
newDataforpred.head()
```

Out[71]:

	loan_installments	sex_TG	annual_income	sex_M	monthly_expenses	loan_tenure	age	house_area	type_of_house_T2	occupants_count	old_dependents	y
0	12	0	36000.0	0	5000.0	12	22	70.0	0	4	0	
1	50	0	94000.0	0	3600.0	12	21	80.0	0	4	1	
2	12	0	48000.0	1	4000.0	12	24	50.0	0	4	0	
3	50	0	7000.0	0	5000.0	12	26	50.0	0	5	0	
4	12	0	36000.0	0	3500.0	12	23	112.0	0	1	0	

In [87]:
```python
df_linearregressor_results = pd.concat([linearregressor_data_prediction,y_samples],axis=1)
df_linearregressor_results.head(10)
```

Out[87]:

	Predicted Loan Amount	loan_amount
0	7939.000459	8000.0
1	9633.556556	15000.0
2	6530.560403	3000.0
3	7942.395120	10000.0
4	7295.899493	8000.0
5	10252.634928	8000.0
6	11758.390860	8000.0
7	7141.211232	8000.0
8	9595.567272	12000.0
9	7163.242060	4000.0

In [88]:
```python
df_linearregressor_results['error_rate']=(abs(df_linearregressor_results['loan_amount']-df_linearregressor_results['Predicted Loa

dt_mean_error=df_linearregressor_results.error_rate.mean()
dt_mean_accuracy=100-dt_mean_error
dt_mean_accuracy
```

Out[88]: 53.32494611383746

In [90]:
```
df_decisiontreeregressor_results = pd.concat([decisiontreeregressor_data_prediction,y_samples],axis=1)
df_decisiontreeregressor_results.head(10)
```

Out[90]:

	Predicted Loan Amount	loan_amount
0	12000.000000	8000.0
1	10000.000000	15000.0
2	3666.666667	3000.0
3	8000.000000	10000.0
4	5000.000000	8000.0
5	8000.000000	8000.0
6	12000.000000	8000.0
7	8000.000000	8000.0
8	15000.000000	12000.0
9	7500.000000	4000.0

In [91]:
```
df_decisiontreeregressor_results['error_rate']=(abs(df_decisiontreeregressor_results['loan_amount']-df_decisiontreeregressor_resu

dt_mean_error=df_decisiontreeregressor_results.error_rate.mean()
dt_mean_accuracy=100-dt_mean_error
dt_mean_accuracy
```

Out[91]: 73.51194263593953

In [93]:
```
df_randomforestregressor_results = pd.concat([randomforest_data_prediction,y_samples],axis=1)
df_randomforestregressor_results.head(10)
```

Out[93]:

	Predicted Loan Amount	loan_amount
0	10585.000000	8000.0
1	10776.666667	15000.0
2	3583.698413	3000.0
3	7950.000000	10000.0
4	8142.023810	8000.0
5	7915.000000	8000.0
6	11387.333333	8000.0
7	6805.000000	8000.0
8	13051.666667	12000.0
9	7325.000000	4000.0

In [94]:
```
df_randomforestregressor_results['error_rate']=(abs(df_randomforestregressor_results['loan_amount']-df_randomforestregressor_resu

dt_mean_error=df_randomforestregressor_results.error_rate.mean()
dt_mean_accuracy=100-dt_mean_error
dt_mean_accuracy
```

Out[94]: 66.24417729322025

In [96]:
```
df_knnregressor_results = pd.concat([knnregressor_data_prediction,y_samples],axis=1)
df_knnregressor_results.head(10)
```

Out[96]:

	Predicted Loan Amount	loan_amount
0	9857.142857	8000.0
1	10000.000000	15000.0
2	3428.571429	3000.0
3	7428.571429	10000.0
4	6857.142857	8000.0
5	8071.428571	8000.0
6	15000.000000	8000.0
7	6000.000000	8000.0
8	12857.142857	12000.0
9	6071.428571	4000.0

In [97]:
```
df_knnregressor_results['error_rate']=(abs(df_knnregressor_results['loan_amount']-df_knnregressor_results['Predicted Loan Amount'

dt_mean_error=df_knnregressor_results.error_rate.mean()
dt_mean_accuracy=100-dt_mean_error
dt_mean_accuracy
```

Out[97]: 73.85280683764307

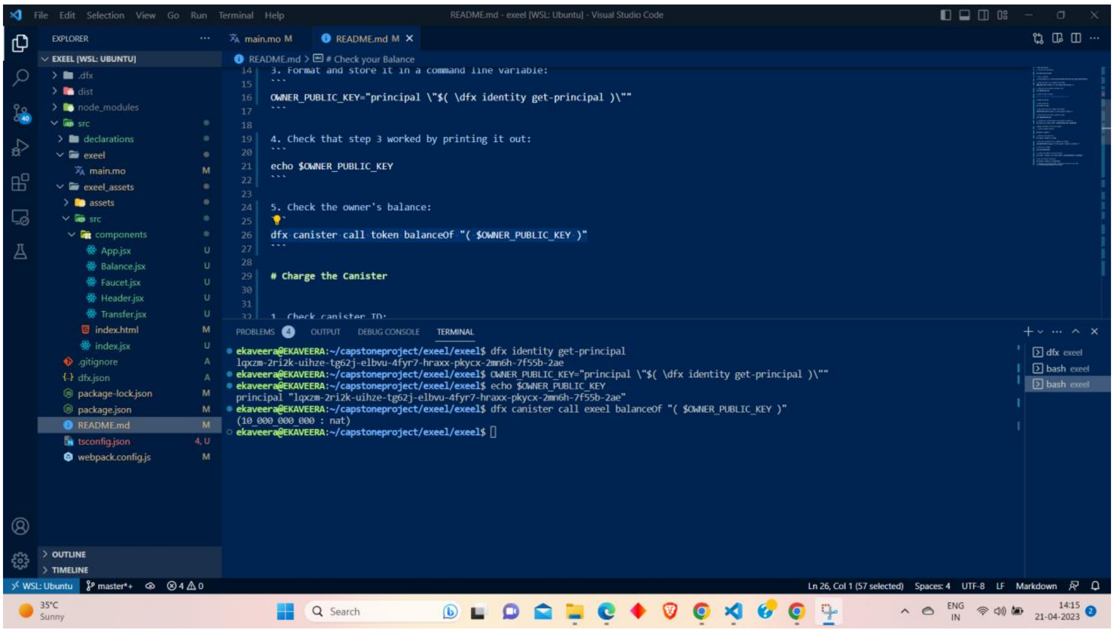
```
In [99]: df_gradientboostregressor_results = pd.concat([gradientboostregressor_data_prediction,y_samples],axis=1)
         df_gradientboostregressor_results.head(10)
```

Out[99]:

	Predicted Loan Amount	loan_amount
0	10517.276309	8000.0
1	10944.921808	15000.0
2	3676.497204	3000.0
3	8158.387384	10000.0
4	6628.288908	8000.0
5	7864.329667	8000.0
6	13190.304783	8000.0
7	8053.894408	8000.0
8	11546.687478	12000.0
9	7500.581100	4000.0

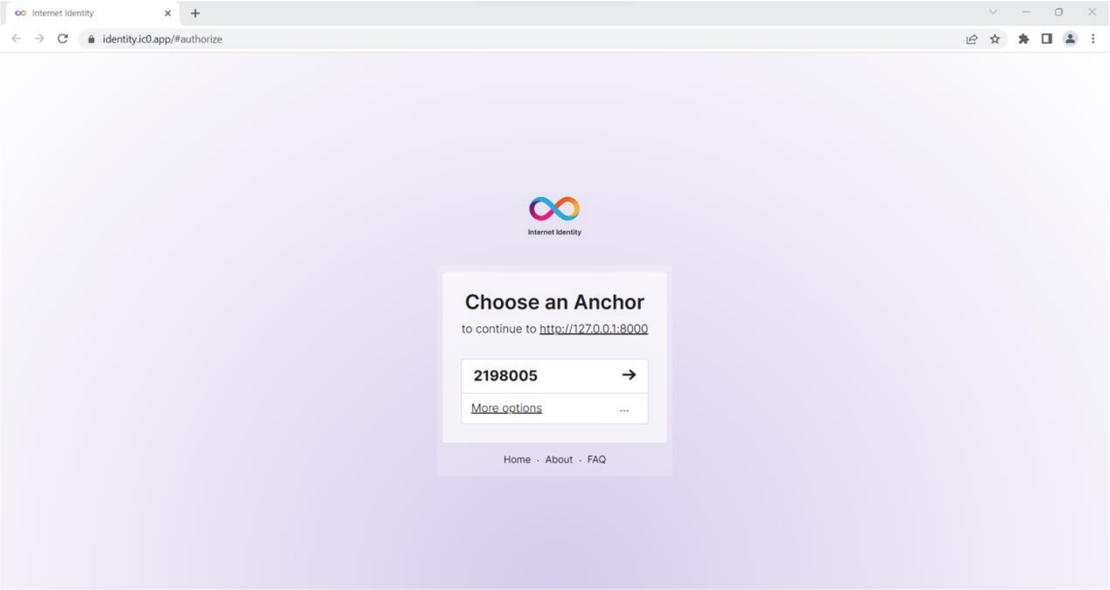
```
In [100]: df_gradientboostregressor_results['error_rate']=(abs(df_gradientboostregressor_results['loan_amount']-df_gradientboostregressor_r

          dt_mean_error=df_gradientboostregressor_results.error_rate.mean()
          dt_mean_accuracy=100-dt_mean_error
          dt_mean_accuracy

Out[100]: 75.34319536923034
```

6.3 DECENTRALIZED APPLICATION OVER BLOCKCHAIN

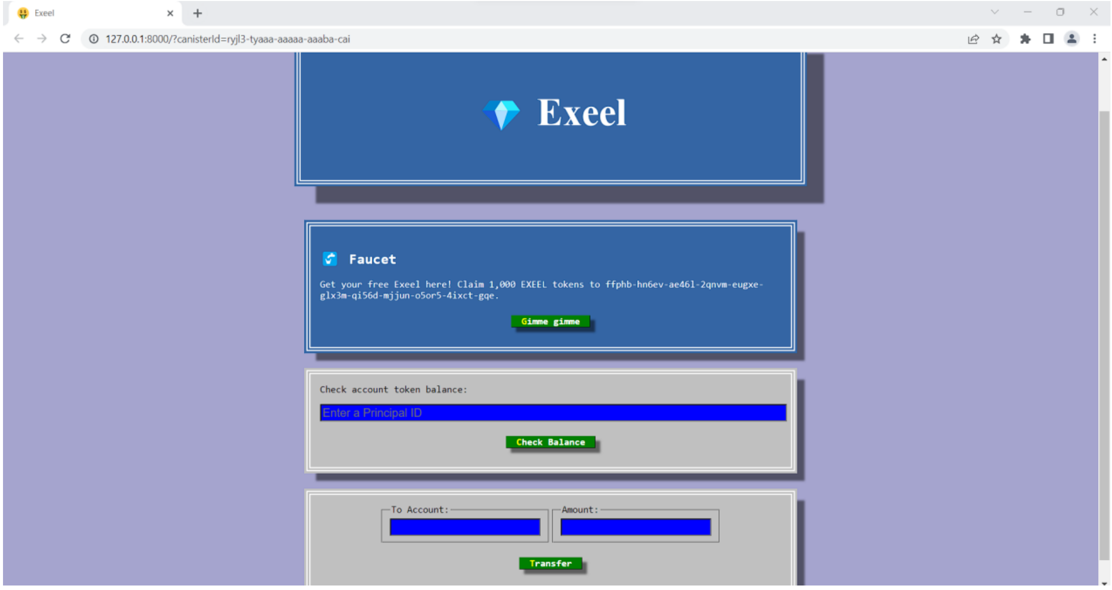


```
ekaveera@EKAVEERA:~/exeel$ dfx deploy
Creating a wallet canister on the local network.
The wallet canister on the "local" network for user "default" is "rwlgt-iiaaa-aaaaa-aaaaa-cai"
Deploying all canisters.
Creating canisters...
Creating canister "exeel"...
"exeel" canister created with canister id: "rrkah-fqaaa-aaaaa-aaaaq-cai"
Creating canister "exeel_assets"...
"exeel_assets" canister created with canister id: "ryjl3-tyaaa-aaaaa-aaaba-cai"
Building canisters...
Building frontend...
Installing canisters...
Creating UI canister on the local network.
The UI canister on the "local" network is "r7inp-6aaaa-aaaaa-aaabq-cai"
Installing code for canister exeel, with canister_id rrkah-fqaaa-aaaaa-aaaaq-cai
Installing code for canister exeel_assets, with canister_id ryjl3-tyaaa-aaaaa-aaaba-cai
Uploading assets to asset canister...
Starting batch.
Staging contents of new and changed assets:
  /index.js (gzip) 1/1 (401574 bytes)
  /index.html 1/1 (606 bytes)
  /index.html (gzip) 1/1 (352 bytes)
  /index.js.map 2/2 (51609 bytes)
  /index.js 1/1 (1733377 bytes)
  /index.js.map 1/2 (1900000 bytes)
  /main.css 1/1 (3022 bytes)
  /index.js.map (gzip) 1/1 (420521 bytes)
  /favicon.ico 1/1 (15406 bytes)
  /main.css (gzip) 1/1 (941 bytes)
Committing batch.
Deployed canisters.
URLs:
  Frontend:
    exeel_assets: http://127.0.0.1:8000/?canisterId=ryjl3-tyaaa-aaaaa-aaaba-cai
  Candid:
    exeel: http://127.0.0.1:8000/?canisterId=r7inp-6aaaa-aaaaa-aaabq-cai&id=rrkah-fqaaa-aaaaa-aaaaq-cai
ekaveera@EKAVEERA:~/exeel$
```

AUTHENTICATION



APPLICATION



Chapter 7

RESULTS AND CONCLUSION

After comparing the performance metrics of various regression algorithms, it can be concluded that the Gradient Boosting Algorithm outperformed other algorithms, including Linear Regression, Decision Tree, Random Forest, and KNN. The evaluation of model performance was based on different metrics such as Accuracy, mean squared error (MSE), and root mean squared error (RMSE).

In comparison, the Linear Regression model had a higher MSE and RMSE values, indicating that it is less accurate than the Gradient Boosting Algorithm. The Decision Tree and Random Forest models performed better than Linear Regression but not as well as Gradient Boosting.

Overall, the Gradient Boosting Algorithm is the most suitable regression algorithm for this project due to its high accuracy and ability to explain a significant amount of variance in the dependent variable.

This approach enables rural people to trade tokens in a more efficient and transparent marketplace, which can help to reduce the costs associated with traditional payment methods.

Furthermore, the use of a token that is specific to the needs of rural communities can also provide an opportunity for these communities to become more self-sufficient. By using the token to purchase agricultural inputs and services, farmers can potentially improve their crop yields and increase their income. This can lead to greater economic independence and a more sustainable agricultural industry.

The token-based payment system can be used to purchase a wide range of products and services beyond just agricultural inputs and services. In addition, the token-based payment system can also potentially provide new opportunities for local businesses and entrepreneurs to participate in the local economy. By accepting the token as payment for their goods and services, they can expand their customer base and potentially increase their revenue. This can help to stimulate economic growth in rural areas and provide new opportunities for local residents.

With further development and testing, this application could have a positive impact on rural communities and finance development of them as a whole.

REFERENCES

[1]. Sai, B. D. S., Nikhil, R., Prasad, S., & Naik, N. S. (2023). A decentralised KYC based approach for microfinance using blockchain technology. ***Cyber Security and Applications*****,** ***1*****, 100009.**

[2]. Kumarathunga, M., Neves Calheiros, R., & Ginige, A. (2022). Sustainable microfinance outreach for farmers with blockchain cryptocurrency and smart contracts. International Journal of Computer Theory and Engineering, 9-14.

[3]. Lalitha, N., & Soujanya, D. (2019, November). Financial sector Innovations: Empowering Microfinance through the application of KYC Blockchain technology. In 2019 International Conference on Digitization (ICD) (pp. 237-243). IEEE.

[4]. Khara, R., Pomendkar, D., Gupta, R., Hingorani, I., & Kalbande, D. (2020, July). Micro loans for farmers. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[5]. Mukkamala, R. R., Vatrapu, R., Ray, P. K., Sengupta, G., & Halder, S. (2018, December). Converging blockchain and social business for socio-economic development. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 3039-3048). IEEE.

[6]. Hoque, M. M. (2022). Microfinance challenges and the potential benefits of blockchain technology and mobile money (Doctoral dissertation, Queensland University of Technology).

[7]. Sridharan, R., Joseph, L., Jeyaprakash, T., Kumar, S. L., & Sanjana, M. (2022, May). Supply Chain Dapp using

Blockchain and Microfinance. In 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 1343-1347). IEEE.

[8]. Merrell, I. (2022). Blockchain for decentralised rural development and governance. Blockchain: Research and Applications, 3(3), 100086.

[9]. Kshetri, N. (2020). Blockchain-based financial technologies and cryptocurrencies for low-income people: Technical potential versus practical reality. Computer, 53(1), 18-29.

[10]. Mahajan, Y., Krishnaswamy, D., & Chelliah, P. R. (2020, April). MiSA-A System for a Microlending Service to Assist Edge Communities. In 2020 IEEE Conference on Technologies for Sustainability (SusTech) (pp. 1-8). IEEE.

[11]. Elliot, E. A., Ngugi, B., & Malgwi, C. A. (2018). Mitigating microfinance marketing channels inefficiencies with customerization of mobile technology. International Marketing Review.

[12]. López-Vargas, A., Ledezma-Espino, A., & Sanchis-de-Miguel, A. (2022). Methods, data sources and applications of the Artificial Intelligence in the Energy Poverty context: A review. Energy and Buildings, 112233.

[13]. Mahajan, Y., & Srivastava, S. (2019). Holistic credit rating system for online microlending platforms with blockchain technology. In Security in Computing and Communications: 6th International Symposium, SSCC 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers 6 (pp. 605-616). Springer Singapore.

[14]. Biallas, M., & O'Neill, F. (2020). Artificial intelligence innovation in financial services.

[15]. Jain, P., Blandin, A. J., O'Neill, J., Perry, M., Ibtasam, S., Shahid, S., ... & Jones, M. (2022, April). Platformisation of Digital Financial Services (DFS): The Journey of DFS in

the Global North and Global South. In CHI Conference on Human Factors in Computing Systems Extended Abstracts (pp. 1-5).

[16]. Rijanto, A. (2021). Business financing and blockchain technology adoption in agroindustry. Journal of Science and Technology Policy Management, 12(2), 215-235.

[17]. Du, R., Ma, C., & Li, M. (2022). Privacy-Preserving Searchable Encryption Scheme Based on Public and Private Blockchains. Tsinghua Science and Technology, 28(1), 13-26.

[18]. Kumar, N., & Veerachary, M. (2020). Stability region based robust controller design for high-gain boost DC–DC converter. IEEE Transactions on Industrial Electronics, 68(3), 2246-2256.

[19]. Paraíso, P., Ruiz, S., Gomes, P., Rodrigues, L., & Gama, J. (2021). Using network features for credit scoring in microfinance. International Journal of Data Science and Analytics, 12, 121-134.

[20]. Simumba, N., Okami, S., & Kohtake, N. (2017, December). Credit decision tool using mobile application data for microfinance in agriculture. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 4714-4721). IEEE.

[21]. Naik, N. (2018, October). LPWAN technologies for IoT systems: choice between ultra narrow band and spread spectrum. In 2018 IEEE international systems engineering symposium (ISSE) (pp. 1-8). IEEE.

[22]. QURAN, A., & QATAWNEH, M. (2021). A BLOCKCHAIN-BASED SYSTEM TOWARDS OVERCOMING FRAUD IN FINANCIAL INSTITUTIONS. Journal of Theoretical and Applied Information Technology, 99(24).

[23]. Muneeza, A., & Mustapha, Z. (2020). Application of blockchain technology in crowdfunding to fuel the rise of the rest globally. A Journal of Interest Free Microfinance, 7(1), 9-26.

[24]. Schuetz, S., & Venkatesh, V. (2020). Blockchain, adoption, and financial inclusion in India: Research opportunities. International journal of information management, 52, 101936.

[25]. Gembali, S. Layout of Decentralized Non-Public Loaning Platform Primarily Based on Block Chain.

[26]. Norta, A., Leiding, B., & Lane, A. (2019, April). Lowering financial inclusion barriers with a blockchain-based capital transfer system. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 319-324). IEEE.

[27]. Bakare, S., Shinde, S. C., Hubballi, R., Hebbale, G., & Joshi, V. (2021, March). A Blockchain-based framework for Agriculture subsidy disbursement. In IOP Conference Series: Materials Science and Engineering (Vol. 1110, No. 1, p. 012008). IOP Publishing.

APPENDIX

SOURCE CODE

Models

Linear Regression

```
from sklearn.linear_model import LinearRegression

linearregressor=LinearRegression()
linearregressor.fit(X_train,y_train)

linearregressor_prediction=linearregressor.predict(X_test)

linearregressor_data_prediction=pd.DataFrame(linearregressor_prediction,columns={
"Predicted Loan Amount"})
linearregressor_data_prediction.head()
```

Decision Tree

```
from sklearn.tree import DecisionTreeRegressor

decisiontreeregressor=DecisionTreeRegressor(criterion='squared_error')
decisiontreeregressor.fit(X_train,y_train)

decisiontreeregressor_prediction=decisiontreeregressor.predict(X_test)

decisiontreeregressor_data_prediction=pd.DataFrame(decisiontreeregressor_predictio
n,columns={"Predicted Loan Amount"})
decisiontreeregressor_data_prediction.head()
from sklearn.ensemble import RandomForestRegressor
```

Random Forest

```
randomforestregressor=RandomForestRegressor()
randomforestregressor.fit(X_train,y_train)

randomforestregressor_prediction=randomforestregressor.predict(X_test)

randomforest_data_prediction=pd.DataFrame(randomforestregressor_prediction,columns={"Predicted Loan Amount"})
randomforest_data_prediction.head()
```

KNN

```
from sklearn.neighbors import KNeighborsRegressor

knnregressor = KNeighborsRegressor(n_neighbors=7)
knnregressor.fit(X_train, y_train)

knnregressor_prediction = knnregressor.predict(X_test)

knnregressor_data_prediction=pd.DataFrame(knnregressor_prediction,columns={"Predicted Loan Amount"})
knnregressor_data_prediction.head()
```

Gradient Boost

```
from sklearn.ensemble import GradientBoostingRegressor
from sklearn import ensemble

params =
{'n_estimators':300,'max_depth':20,'learning_rate':0.02,'criterion':'squared_error'}

gradientboostregressor = ensemble.GradientBoostingRegressor(**params)
gradientboostregressor.fit(X_train, y_train)

gradientboostregressor_prediction = gradientboostregressor.predict(X_test)
```

```
gradientboostregressor_data_prediction=pd.DataFrame(gradientboostregressor_predic
tion,columns={"Predicted Loan Amount"})
gradientboostregressor_data_prediction.head()
```

Main.mo

```
import Principal "mo:base/Principal";
import HashMap "mo:base/HashMap";
import Debug "mo:base/Debug";
import Iter "mo:base/Iter";

actor Exeel {

  let owner : Principal = Principal.fromText("lqxzm-2ri2k-uihze-tg62j-elbvu-4fyr7-
hraxx-pkycx-2mn6h-7f55b-2ae");
  let totalSupply : Nat = 1000000000000000;
  let symbol : Text = "EXEEL";

  private stable var balanceEntries : [(Principal, Nat)] = [];
  private var balances = HashMap.HashMap<Principal, Nat>(1, Principal.equal,
Principal.hash);
  if (balances.size() < 1) {
      balances.put(owner, totalSupply);
    };

  public query func balanceOf(who: Principal) : async Nat {

    let balance : Nat = switch (balances.get(who)) {
      case null 0;
      case (?result) result;
    };

    return balance;
```

```
};

public query func getSymbol() : async Text {
  return symbol;
};

public shared(msg) func payOut() : async Text {
  Debug.print(debug_show(msg.caller));
  if (balances.get(msg.caller) == null) {
    let amount = 1000;
    let result = await transfer(msg.caller, amount);
    return result;
  } else {
    return "Tokens already claimed";
  }
};

public shared(msg) func transfer(to: Principal, amount: Nat) : async Text {
  let fromBalance = await balanceOf(msg.caller);
  if (fromBalance > amount) {
    let newFromBalance : Nat = fromBalance - amount;
    balances.put(msg.caller, newFromBalance);

    let toBalance = await balanceOf(to);
    let newToBalance = toBalance + amount;
    balances.put(to, newToBalance);

    return "Transaction Successful";
  } else {
    return "Insufficient Funds in your account";
  }

};
```

```
  system func preupgrade() {
    balanceEntries := Iter.toArray(balances.entries());
  };

  system func postupgrade() {
    balances := HashMap.fromIter<Principal, Nat>(balanceEntries.vals(), 1,
Principal.equal, Principal.hash);
    if (balances.size() < 1) {
      balances.put(owner, totalSupply);
    };
  };

};
```

Index.js

```
import ReactDOM from "react-dom";
import React from "react";
import App from "./components/App";
import { AuthClient } from "@dfinity/auth-client";
import { Principal } from "@dfinity/principal";

const init = async () => {
  const authClient = await AuthClient.create();

  if (await authClient.isAuthenticated()) {
    handleAuthenticated(authClient);
  } else {
    await authClient.login({
      identityProvider: "https://identity.ic0.app/#authorize",
      onSuccess: () => {
        handleAuthenticated(authClient);
      },
    });
  }
};
```

```
async function handleAuthenticated(authClient) {
  const identity = await authClient.getIdentity();
  const userPrincipal = identity._principal.toString();
  console.log(userPrincipal);
  ReactDOM.render(
    <App loggedInPrincipal={userPrincipal} />,
    document.getElementById("root")
  );
}

init();
```