### **CHAPTER ONE: INTRODUCTION**

#### 

#### **1.1 BACKGROUND OF THE STUDY**

Credit risk assessment describes techniques and methodologies employed in the analysis of induced risk from providing credit services or facilities, the ability to properly assess credit risk in order to arrive at a fair decision is valuable to financial assistance providers as it can be the discerning factor between a successful and growing lender and an unprofitable credit provider.

Importance of risk assessment includes keeping track of the (credit risk of the) economy from a macroeconomic perspective, assessing financial market stability from a macroprudential perspective as well as assessing the credit quality of collateral in the context of monetary policy operations (Winkler, 2008)

“Traditional” credit risk evaluation models are easily operated, but these evaluation models heavily depend on estimators or the experts’ subjective judgement. So it may lead to an erroneous conclusion.

It is the target of scholars to discover more precise and objective credit risk evaluation techniques. Existing in enterprise financial data, Beaver (1966, pp. 71-111) discovered the single variable index useful for default standard judgement with comparative analysis. However, resulting predictions are not as accurate as required, leading many scholars to improve model addition variables in order to get a series of multivariable prediction models that are defined by single variables. The Z-scoring model developed by Altman & Haldeman (1977, pp. 29-54), one being an American finance scholar, is relatively well known. Later, Altman improved the initial model. After swapping and adding two new indices, Altman derived the new and improved Zeta-scoring model using seven indices.

There was also the linear discriminant model designed by Goldberger (1964) is the foremost multivariable risk evaluation model. With progression in the development of statistical mathematics, models like the Logistic (Ohlson, 1980) and Probit (Martin, 1977, pp. 249-276) are developed and widely implemented for predictions of default from loan and credit customers. The Logistic model also finds application in evaluating the probability of financial institutions’ default by (West, 1985, pp. 253-266).

In the absence of sample data or with a small volume of sample quantity available for use, the statistical method is no longer in application.

However, a method that is independent of data has been explored. Therefore, data envelope method (Porter et al., 1990), other non-statistical models as well as the tracing decision method (Huber, 1985) will surface at the appropriate moment. Accompanying the speedy development of artificial intelligence technologies, some methodologies and theories find use in loan & credit customer credit risk assessment, including genetic algo model (Varetto, 1998), support vector machine model (Wang, 2009), neural network model (Arzum & Karatepe, 2007), clustering model (Lundy, 1993) and rough set model (Pawlak, 1982). The corresponding evaluation model is then established (Cai & Qian, 2018, pp. 89-90)

#### **1.2 STATEMENT OF THE PROBLEM**

Big corporations are now becoming aware of the challenges associated with incorporating AI-driven credit risk evaluation solutions, a number of these issues to be considered being regulatory compliance, concern of ethics as well as privacy of data.

There is little to no transparency, brewing some distrust in these implementations, especially by the stakeholders. This absence of transparency also removes the accountability factor in credit risk assessment, with the assumptions that AI is always correct; this is not necessarily true!

Data privacy which is a significantly major concern is focused on with the amount of sensitive information of consumers that these financial service providers handle, there’s a need to ensure that any AI-driven solution that is implemented works with full consideration of data privacy policies, California Consumer Privacy Act (CCPA) and General Data Protection Regulation (GDPR) inclusive.

Concerns of ethics are also paramount to be factored. AI-powered credit risk evaluation models have the ability to automate processes involving decision-making, which could turn out to provide surprising biases which could be the proper results of the models or erroneous.

Even with the auditory provisions made by some enterprises to put their AI-driven solutions in check to remain compliant and effective, the opacity does not allow for total credibility in its decisions, considering another important factor; regulatory compliance, addressing that every enterprise in use of AI-powered solutions must ensure their implementations are compliant with relevant policies such as the Equal Credit Opportunity Act (ECOA) and theFair Credit Reporting Act (FCRA)

Additionally, there’s so much room for technical challenges to arise when incorporating AI-driven credit risk assessment solutions. There might be the need to invest in new infrastructure as well as systems, in order to provide support to these solutions, whilst ensuring that these same corporations have the required skills and in-house expertise to maintain and implement them. (Johnson, 2023)

#### **1.3 AIM AND OBJECTIVES**

The primary aim of this project is to develop an XAI model for credit risk evaluation that’d enhance interpretability as well as transparency in decision-making processes.

The following objectives will be used to achieve this goal:

i. Extensively review existing (X)AI techniques involved in credit risk.

ii. To design and develop a user-friendly and efficient XAI framework tailored for CRA to incorporate interpretable machine learning methods and algorithms.

iii. To identify and merge relevant features and variables from historically proven credit data to enhance comprehensibility.

iv. To evaluate and test the model’s performance against traditional AI models and benchmark results against credit risk assessment industry standard metrics.

iii. To validate and analyze the explainability of the model.

iv. To implement the XAI model in a simulated environment to measure practicability.

#### **1.4 SIGNIFICANCE OF THE STUDY**

Implementing explainable artificial intelligence in credit risk assessment is aimed at providing clear explanations of credit risk conclusions, with the XAI model aiming to aid financial corporations comply with regulatory policies that require transparency in borrowing practices, with this, stakeholder trust is significantly increased; this trust would be mutually existent with customers, investors as well as regulatory bodies, with the provision of comprehensive and decipherable insights into credit decisions.

The improvement in risk management cannot be overemphasised, with the clarity of risk factors offered by an XAI model, better informed and accurate risk management decisions are made.

The overall loan approval process can be improved, with financial incorporations having an in depth understanding of the underlying factors affecting the evaluation of credit risk, the tendency for default is reduced, still addressing the processes; fairness and equality in lending is encouraged, with the ability of XAI to point out and mitigate potential biases in existing credit risk evaluation models, this then leads to the accountability factor, with interpretability comes with responsibility, helping with holding the financial institutions accountable for their credit decisions whilst upholding ethical standards.

Another outlook of additional benefits will include the identification and correction of errors in risk evaluation and facilitation of smoother training procedures, this study will proceed to contribute to the advancement of AI technologies with models that are not merely correct but also comprehensive and explainable, setting a benchmark for future developments in AI applications within the finance sector, fostering further improvements as well as innovations; a ripple effect of this would be a considerably notable stability in financial markets seen as a contribution from implementing XAI to reduce the tendency of financial crises which could arise from as little as loan defaults.

The discoveries and methodologies developed in this project can find application as educational content for future training of data scientists, AI researchers and financial analysts; providing a solid foundation for further academic research in the field of XAI and its application in finance.

#### **1.5 SCOPE AND LIMITATIONS**

The scope of this research includes traditional AI models used in risk assessment breaking down their methodologies and how they utilise historical loan data, macroeconomic indicators, borrower financial profiles and credit-related data, this project will also include real-world case studies to demonstrate the practical application as well as pros of XAI in the world of credit risk.

The primary context is financial institutions, particularly the ones involved in lending as well as managing credit risk; corporations like credit unions, banks and other financial aiding agencies.

Addressing data preprocessing techniques such as cleaning and filtering; feature selection and engineering, model training and validation processes that are important for the development of detailed and explainable AI models. Comparative analysis between the existent traditional AI models and XAI-enhanced models will be conducted to assess correctness, interpretability as well as feasibility in credit risk assessment, using key evaluation metrics like model transparency, prediction accuracy as well as user dependence in AI-driven decisions.

XAI models tailored for specific financial corporations or credit risk scenarios, posing limitations in global applicability for example other contexts or sectors. Assumptions and biases made or hard coded into the model in its development stages could impact the feasibility of these models in a wide range, creating a gap or posing a tradeoff where explainability is traded for usability.

Advanced XAI techniques may require demanding computational resources and professional expertise. Adoption of XAI models by financial experts and credit stakeholders might require heavy modifications in existing workflows, personnel training and end user familiarity are other limitations of the system.

#### 

#### 

#### 

### **CHAPTER TWO**

#### **LITERATURE REVIEW**

##### **2.1 INTRODUCTION**

Credit risk describes the tendency for a lender to lose money when they provide funds to a borrower upon request, from this definition it can be further noted that, credit risk assessment then is a fundamental constituent of financial decision-making, influencing lending terms and conditions, interest rates, and general financial stability.

Credit risk assessment performs a crucial function in the finance sector, by weighing a financial institution's lending choices against the credit risk posed by the counterparty, in order to ensure the borrower’s competence and willingness to fulfil credit obligations that arise initially or at any point during the credit term. (“What is Credit Risk?”)

Traditionally, credit risk assessment depended on expert judgement combined with statistical models. Though efficient to some extent, these techniques oft lack transparency, leaving stakeholders uninformed regarding the requisites impacting assessments (Büschken & Villas-Boas, 2019).

Furthermore, inadvertent sustenance of biases or errors may arise, inducing flawed upshots (Zhang & Wu, 2021).

However, with the innovative intervention of artificial intelligence and machine learning, there has been a diversion towards approaches that are more sophisticated, Explainable AI (XAI) inclusive.

Lately, the incorporation of artificial intelligence (AI) methods into diverse industries has radically changed decision-making procedures, offering exceptional precision and competence (Huang et al., 2020). However, the ubiquitous implementation of AI is accompanied by its own array of difficulties, specifically in sectors where accountability and prevalence are preeminent, such as this project, credit risk assessment.

What is Explainable Artificial Intelligence?

Explainable artificial intelligence (XAI) is an array of procedures and techniques that permits human users to understand and trust the conclusions and results generated by machine learning algorithms.

Explainable AI is used to portray an AI model, its envisaged impact and possible prejudices. It helps depict model precision, evenhandedness, accuracy, transparency and results in AI-enhanced decision making. Explainable AI is indispensable for an institution in developing credence and assurance when involving AI models with operations. AI explainability also assists an incorporation in adopting a responsible strategy to AI development and utilisation.

As AI becomes more sophisticated, humans are challenged to backtrack and comprehend how the algorithm arrived at a result.

The entire determination process is turned into what is generally alluded to as a “black box" that is impracticable to decipher. These black box models are directly crafted from data. And, not even the data scientists or engineers who create the algorithm can comprehend or explain exactly how the AI algorithm arrived at a particular result.

There are numerous advantages to the comprehension of how an AI-enhanced system leads to a specific output. Explainability being one of them helps developers ensure that the system is functioning as expected, meeting necessary regulatory standards, or allowing the subjects of a decision to counter or alter that outcome. (“What is explainable AI?”)

Explainable AI refers to AI systems design concept, where the system’s decisions and behavioural patterns are concise to humans. This is of particular importance in critical fields, where AI is employed, such as defence and security, finance, healthcare, autonomous vehicles, criminal justice or customer service.

To tackle these obstacles, the notion of Explainable AI (XAI) has arisen as an encouraging approach. XAI strives to enhance the pellucidity and comprehensibility of AI prototypes, enabling interested parties to comprehend and rely on the process of making decisions. By offering perspectives on how AI algorithms arrive at their results, XAI not only facilitates the recognition and alleviation of prejudices or inaccuracies but also promotes accountability (Lipton, 2016).

This introductory chapter establishes the context for examining the incorporation of XAI in credit risk assessment. Revealing the optimism behind harnessing XAI methods whilst highlighting the importance of accountability and transparency in making decisions as well as identifying crucial challenges facing primitive credit risk assessment techniques.

Moreover, it dispenses a synopsis of the goals and structure of this project, facilitating an exhaustive evaluation of the function of XAI in transforming credit risk assessment methodologies.

XAI techniques aim to provide pellucidity and interpretability, allowing users to trust AI systems to perform effectively whilst diagnosing and rectifying any biases or errors that might occur.

##### **2.2 THE RISE OF EXPLAINABLE AI IN CREDIT RISK ASSESSMENT**

Explainable AI (XAI) has picked up steam in credit risk assessment because regulatory compliance, stakeholder trust, and transparency are required for longevity.

XAI techniques strive for understandable explanations for abstruse machine learning prototypes, allowing stakeholders to be cognizant of the decision-making process. XAI is used in expressing the strength of machine learning algorithms as to oppose the black-box nature that is noticeably the most significant roadblock preventing the banks from performing their operations and business efficiently. (Nallakaruppan et al., 2024)

**2.3 GENERAL STATEMENTS ABOUT THE PROBLEM OF IMPLEMENTING EXPLAINABLE AI IN CREDIT RISK ASSESSMENT**

Incorporating Explainable AI in assessing credit risk aims to improve transparency in the decision-making process by providing explanations as to how AI algorithms arrive at their conclusions. Invested individuals can better comprehend and rely on the recommendations and conclusions that these systems make.

One of the main aims of Explainable AI in assessing credit risk is the identification and mitigation of biases intrinsic to the traditional models. By prioritising the transparency of every step of the decision-making process, XAI techniques detect biases that may result in partial or erroneous credit risk assessments.

XAI techniques raise the bar, especially for accountability in credit risk assessment. Done by empowering stakeholders to trace and understand the factors that influence the final decisions made by the model. This transparency holds both XAI models along with their developers, accountable for the outcomes of credit risk assessments based on said models.

Incorporating Explainable AI encourages a tradition of continuous refinement in credit risk assessment processes. By streamlining the recognition of inaccuracies and leanings, XAI techniques empower organisations to improve their models and techniques in the long run, leading to more equitable and precise credit risk assessments.

Implementing Explainable AI can culminate into insightful and decisive decision-making in credit risk assessment. By providing succinct explanations for decisions, XAI techniques equip stakeholders well enough to make better-informed judgments regarding lending and associated risk management.

The utilisation of Explainable AI in credit risk assessment assists corporations with compliance to regulatory requirements associated with transparency and fairness. Regulatory bodies continuously count on financial institutions to adopt AI techniques that can provide apt explanations for their choice of actions, making the implementation of XAI crucial for compliance.

Transparent and understandable credit risk assessment techniques can foster customer satisfaction and reliance. When borrowers are in the know of the rationale behind financing choices, they are prone to trust the financial institutions to handle the situation professionally and in their best interest, assuring customers of the best possible financing.

##### **2.4 OVERVIEW OF RESEARCHES ON IMPLEMENTABLE EXPLAINABLE AI IN CREDIT RISK ASSESSMENT**

Here is an overview of the research coverage in XAI’s implementation in Credit Risk Assessment currently:

1. **Interpretability Methods**:

Various interpretability techniques to enhance transparency such as feature usefulness analysis and local explanations have been explored to make AI models employed, more understandable in credit risk assessment (Lundberg & Lee, 2017).

In addition, model-neutral techniques such as SHAP and LIME present insights into unit predictions and system model behavior (Ribeiro et al., 2016).

2. **Proposal of Hybrid Models**:

Hybrid models which combine the comprehensibility of primitive statistical models with the predictive abilities of sophisticated machine learning algorithms, have been theoretically prototyped so as to establish symmetry between transparency and precision (Lakkaraju et al., 2016; Wang et al., 2015).

3. **Ensuring Regulatory Compliance**:

Research activities in XAI directed at better assessment of credit risk also places priority on ensuring and enforcing compliance with set policies and regulations such as GDPR (Wachter et al., 2017) and handling the right to detailed description in decision-making processes that are automated (Goodman & Flaxman, 2017).

4. **Tools for Visualization**:

User-friendly tools for visualisation have been designed to induce decisions that are better informed by stakeholders including regulators as well as loan officers, by representing complicated information in a visual format that is intuitive, these tools enable users to make informed decisions with confidence. (Strobelt et al., 2017; Liu et al., 2018).

5. **Equality and Bias Mitigation of Assessment Procedures**:

XAI techniques are employed to detect and put in check, instances of biases in credit risk assessment models, increasing fairness whilst encouraging inclusivity in lending practices.

With explanations for model predictions clearly provided, stakeholders can detect and rectify biases associated with aspects such as gender, race, exposure, or socioeconomic status. This would ensure that XAI enhanced lending practices have equity as a priority whilst fostering trust in the financial system.

(Hardt et al., 2016; Zemel et al., 2013).

6. **Case Studies and Real-world Instances**:

Real-world scenarios and case studies are now popularly being used for the demonstration of how effective XAI can be, when employed in the assessment of credit risk whilst providing insights into challenges posed by implementation and optimal practices. By exhibiting successful implementation cases, researchers in this field can speed up the integration of straightforward and accountable AI systems in the finance sector. (Kamiran & Calders, 2012; Pleiss et al., 2017).

In General, cross-disciplinary efforts among ethicists, computer scientists, statisticians, and domain experts has proved crucial for advancement in developing AI systems that are accountable and dependable for use in the financial industry.

##### **2.5 COMPARISON OF EXISTING METHODS IN THE IMPLEMENTATION OF XAI IN CREDIT RISK ASSESSMENT**

| METHOD | DESCRIPTION | STRENGTHS | LIMITATIONS | USE CASES |
| --- | --- | --- | --- | --- |
| LIME (Local Interpretable Model-agnostic Explanations) | Provides explanations for individual predictions by approximating the model on a local scale with an interpretable one | - Compatible with textual, tabular and image data.  - Fidelity features give an overall idea on the reliability of the interpretable model in explaining black-box predictions in the region of interest. (Saha, 2019) | - Provides explanations for individual predictions, hence, not reflecting the model's overall behaviour.  - Reliability is affected by randomness in the sampling process outputting varying explanations for the same input. (IITian, 2024) | Explains the scores around a specific query point or observation with a simple local model, such as a decision tree or a linear regression model. (*MATLAB & Simulink*) |
| Feature Usefulness Analysis | Evaluates the usefulness of features in the model to enhance transparency | - Identifies features of significance  - Provides insights into model behaviour | - Irrelevant feature selections lead to overfitting in the learning process with no significant contribution to the performance of the model. (Jemai, 2023) | - Misleading features could be correlated  - Limited ability in providing detailed local explanations |
| Visualisation Tools | User-friendly interfaces designed to visually describe complex information in a comprehensive manner for stakeholders. | - Shows details that might otherwise remain unseen to an unprofessional eye  - Enhances better informed decision making | - Requires good and explanatory design to be effective  - These tools may overly simplify data | Interactive dashboards in use by credit officers |
| SHAP (SHapley Additive eXplanations) | Its value calculates the impact a feature X has on the model’s prediction Y, done by computing the output in absence and presence of every X over every possible permutation of the selected features. (*SHAP for Credit Risk: Interpreting Machine Learning Black Box*, 2021) | - Provides both local and global explanations  - Usually consistent and theoretically sound | - Computationally demanding  - Increasing complexity with increase in datasets | SHAP values for feature engineering in evaluating credit risk |
| Regulatory Compliance Models | Holds compliance with regulations such as GDPR with highest priority | - Provides legally compliant conclusions  - Maximises feature engineering, by ignoring unaligned features | - Limits flexibility  - Might not be suitable over various jurisdictions | GDPR certified models used in automated lending processes |
| Bias Mitigation Techniques | Techniques to uncover and mitigate biases in the results of models | - Promoting inclusive lending practices  - Drastically reduces unfairness in the credit sector | - Difficulty in identifying every occurrence of bias  - Constant monitoring | Fairness constraints existing in lending models |
| Hybrid Models | Technical combination of statistical models with one or more machine learning algorithms | - Strikes the balance between accuracy and transparency  - Leverages the strengths of multiple models | -Requires extensive tuning  - It is complex to implement | Combination of neural networks with logistic regression for risk assessment |
| Case Studies | Demonstrates how effective XAI has been through real-world scenarios | - It provides practical insights  - Highlights challenges and notes best practices | - May exempt some possibilities  - Implementation might be unsuitable in some contexts | Research or published papers on XAI in credit risk assessment |
| Counterfactual Explanations | Provides explanations by showing how to convert inputs to desired outputs | - Easily understood by stakeholders  - It creates actionable insights | - Feasibility is not certain  - Can be computationally demanding | Counterfactual explanations for declined loans |

###### **Table 2.1.1**: Comparison Of Existing Methods In The Implementation Of XAI In Credit Risk Assessment

##### **2.6 REVIEW OF RELATED WORKS**

This section focuses on the comparison and reviewing of related works done by several academicians, authors, researchers and writers. Providing an analysis of their work done, key points as well as relevance to xai in credit risk assessment

| Title | Author(s) | Year | Key Points | Relevance to XAI in Credit Risk Assessment |
| --- | --- | --- | --- | --- |
| Equality of Opportunity in Supervised Learning | Hardt, M., Price, E., & Srebro, N. | 2016 | Introduces methods for achieving fairness in machine learning. | Paramount for ensuring fairness and mitigating bias in credit risk models. |
| European Union regulations on algorithmic decision-making and a "right to explanation" | Goodman, B., & Flaxman, S. | 2017 | Discusses EU regulations and the right to explanation in AI. | Essential for understanding regulatory compliance in credit risk assessment. |
| Interpretable Decision Sets: A Joint Framework for Description and Prediction | Lakkaraju, H., Bach, S. H., & Leskovec, J. | 2016 | Proposes interpretable decision sets for machine learning models. | Useful for creating interpretable models in credit risk assessment. |
| Data preprocessing techniques for classification without discrimination | Kamiran, F., & Calders, T. | 2012 | Discusses some data preprocessing techniques for discrimination avoidance | Important for credit data preprocessing to ensure fairness. |
| A Unified Approach to Interpreting Model Predictions | Lundberg, S. M., & Lee, S. I. | 2017 | Introduces SHAP, a unified framework for interpreting model predictions. | Key for applying SHAP values to explain credit risk models. |
| Feature Selection Engineering for Credit Risk Assessment in Retail Banking | Jemai, J. | 2023 | Discusses feature selection techniques for improving credit risk assessment models. | Relevant for selecting the most important features in credit risk models |
| "Why Should I Trust You?" Explaining the Predictions of Any Classifier | Ribeiro, M. T., Singh, S., & Guestrin, C. | 2016 | Introduces LIME, a known technique for explaining individual model predictions | Useful for explaining individual credit risk decisions transparently |
| On Fairness and Calibration | Pleiss, G.,  Raghavan, M.,  Wu, F., Kleinberg, J., & Weinberger, K. Q. | 2017 | Discusses fairness and calibration in machine learning models. | Essential for ensuring fair and calibrated credit risk assessment models |
| Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation | Wachter, S., Mittelstadt, B., & Floridi, L. | 2017 | Analyses the GDPR and the lack of a right to explanation. | Important for understanding the legal landscape of explainability in credit risk assessment. |
| Learning Fair Representations | Zemel, R., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. | 2013 | Proposes methods for identifying fair representations in machine learning. | Techniques for enhancing fairness in credit risk assessment models. |
| Understanding LIME (Local Interpretable Model Agnostic Explanations) | IITian | 2024 | Provides an overview of LIME and its applications. | Useful for understanding how to apply LIME to credit risk models. |
| SHAP for Credit Risk: Interpreting Machine Learning Black Box | Valoores in'Analytics | 2021 | Addresses the application of SHAP in interpreting credit risk models. | Crucial for implementing SHAP in credit risk assessment models. |
| Local Interpretable Model-Agnostic Explanations (LIME) — the ELI5 way | Saha, S. | 2019 | Breaks down LIME in a comprehensive manner. | For applying LIME in an accessible way to credit risk assessment |

###### **Table 2.1.2**: Review of Related Works In The Implementation Of XAI In Credit Risk Assessment

#### 

#### 

#### 

#### 

### **CHAPTER THREE: RESEARCH METHODOLOGY**

#### **3.1 ANALYSIS OF THE DATASET**

The process of analysing the demographics dataset for credit risk assessment required evaluations, of different manners of approach, shown below:

1. **Checking out the dataset**: In this analysing step, important libraries are imported and utilised for the purpose of reading the required dataset from an excel file into memory, the library in question here is called pandas, known for its use in analysing, exploring, cleaning, and manipulating data. The name "Pandas'' references both "Python Data Analysis", and "Panel Data" created in 2008 by Wes McKinney. Also provided is a display of the first 10 rows, providing an overview of its structure.

import pandas as pd

# read the dataset into a dataframe

df = pd.read\_excel("../data/dataset\_v0.xlsx")

# show the first 10 rows

df.head(10)

**2. Checking the dataset information:** This part of the analysis examines missing values, it addresses the dimensions of the dataset, checks unique values in the 'Gender' column as well as their distribution, providing insights into the dataset's characteristics.

# Check for missing values in each column to assess data completeness.

missing\_values = df.isna().sum()

# Retrieve the dimensions (rows and columns) of the dataset to understand its size.

dataset\_shape = df.shape

# Identify unique values in the 'Gender' column to understand categorical data distribution.

gender\_unique\_values = df['Gender'].unique()

# Count occurrences of each gender category to analyze data imbalance or uniformity.

gender\_value\_counts = df['Gender'].value\_counts()

missing\_values, dataset\_shape, gender\_unique\_values, gender\_value\_counts

#### 

#### **3.2 DATA PREPROCESSING**

Before proceeding with the process of generating a model, preprocessing the data is of high importance, part of which is to ensure it's ready for modeling, the steps this project uses to approach the data preprocessing includes:

1. Loading and standardising the dataset

2. Handling categorical columns

3. Encoding categorical features

4. Standardizing numerical features

5. Combining processed data

6. Optional dimensionality reduction

7. Splitting the dataset

8. Encoding categorical variables

#### **3.2.1 LOADING AND STANDARDIZING THE DATASET**

This preprocessing step involves processes such as loading the cleaned dataset and separating features from the target variable (Loan\_Status) followed by the identification of numerical and categorical columns for advanced preprocessing.

import pandas as pd

from sklearn.preprocessing import StandardScaler

# Load cleaned dataset

df\_cleaned = pd.read\_excel('../data/dataset\_v1.xlsx', index\_col='Loan\_ID')

df\_cleaned.head()

# Separate features and target

X = df\_cleaned.drop('Loan\_Status', axis=1)

y = df\_cleaned['Loan\_Status'].copy()

# Select numerical and categorical columns

num\_values = X.select\_dtypes(include=np.number)

cat\_values = X.select\_dtypes(exclude=np.number)

num\_values.columns, cat\_values.columns

#### **3.2.2 HANDLING CATEGORICAL COLUMNS**

Categorical columns are specifically converted to binary representations of ('0' or '1') to make handling them easier during their encoding and modeling processes.

# Converting categorical columns to binary

def convert\_to\_0\_1(row):

return 1 if (row == 'Male') or (row == 'Yes') else 0

cat\_values['Married'] = cat\_values['Married'].apply(convert\_to\_0\_1)

cat\_values['Self\_Employed'] = cat\_values['Self\_Employed'].apply(convert\_to\_0\_1)

cat\_values['Gender'] = cat\_values['Gender'].apply(convert\_to\_0\_1)

cat\_values.head()

#### **3.2.3 ENCODING CATEGORICAL FEATURES**

So as to perform the encoding of the categorical features OneHotEncoder is applied to these features to transform them into a format suitable for machine learning algorithms. This step expands categorical variables into binary columns for every present category.

# Encode categorical features using OneHotEncoder

from sklearn.preprocessing import OneHotEncoder

cat\_values\_to\_one\_hot = cat\_values.drop(['Gender', 'Married', 'Self\_Employed'], axis=1)

one\_hot = OneHotEncoder()

one\_hot.fit(cat\_values\_to\_one\_hot.astype(str))

cat\_values\_encoded = one\_hot.transform(cat\_values\_to\_one\_hot.astype(str)).todense()

cat\_values\_encoded = pd.DataFrame(cat\_values\_encoded, columns=one\_hot.get\_feature\_names\_out(), index=cat\_values.index)

cat\_values\_encoded.head()

#### 

#### **3.2.4 STANDARDIZING NUMERICAL FEATURES**

The present numerical features are standardized with the use of StandardScaler, ensuring all features have unit variance as well as zero mean. The sole purpose of standardizing these numerical features is how it aids in the prevention of biases due to varying scales of numerical data.

# Standardize numerical features

std = StandardScaler()

std.fit(num\_values)

pickle.dump(std, open("models\_and\_encoders/credit\_risk\_scaler.pkl", "wb"))

num\_values\_standardized = scaler.transform(num\_values)

num\_values\_standardized = pd.DataFrame(num\_values\_standardized, columns=num\_values.columns, index=num\_values.index)

num\_values\_standardized.head()

#### **3.2.5 COMBINING PROCESSED DATA**

The processed numerical and categorical features are combined into X\_processed, thereby representing the final dataset which is now ready for model training. The resulting consolidated dataset maintains the original data integrity while simultaneously preparing it for predictive modeling tasks.

# Combine standardized numerical and encoded categorical features

perfect\_cat\_values = pd.concat([cat\_values\_ordinal, cat\_values\_to\_one\_hot], axis = 1)

perfect\_cat\_values = pd.concat([perfect\_cat\_values, num\_values["Credit\_History"]], axis = 1)

num\_values = num\_values.drop("Credit\_History", axis = 1)

restructured\_x = pd.concat([standardized\_num\_values, perfect\_cat\_values], axis = 1)

restructured\_x.head()

full\_df = pd.concat([restructured\_x, y], axis =1)

full\_df.head()

#### 

#### **3.3 MODEL GENERATION**

In order to handle the requirement of a model, the steps listed below were followed in order to build a suitable model

#### **3.3.1 IMPORTING NECESSARY LIBRARIES**

Varying requirements from the models to be sampled necessitates the importing of different libraries, these includes:

1. **Pandas (pd) and NumPy (np)**: Essential for numerical operations data manipulation and numerical operations respectively.

2. **Pickle and Joblib**: Used for saving and loading trained models and encoders.

3. **Scikit-learn Modules**: Proper library used for machine learning tasks such as splitting data (train\_test\_split), model evaluation (confusion\_matrix, accuracy\_score, f1\_score, recall\_score), preprocessing (OneHotEncoder, StandardScaler), and algorithms for classification such as (LogisticRegression, RandomForestClassifier, GradientBoostingClassifier, SVC, KNeighborsClassifier and XGBClassifier).

4. **Seaborn and Matplotlib**: For data visualization, including plotting heatmaps and barplots.

import pandas as pd

import numpy as np

import pickle

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import OneHotEncoder, StandardScaler

import joblib

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix, f1\_score, recall\_score, accuracy\_score

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

from xgboost import XGBClassifier

#### 

#### **3.3.2 STANDARDIZING AND ENCODING DATASET**

In order to efficiently encode and standardize the dataset, we continue by loading the cleaned dataset (df\_cleaned) and separating features (x) from the target (y), these steps cover the preparation required.

With the goal of generating the model we proceed with categorical handling, one-hot encoding and then standardization but these procedures have been handled earlier under data preprocessing, hence the importance of preprocessing includes accuracy in model selection.

In details, the procedures followed are shown below:

# Checkout the cleaned dataset

df\_cleaned = pd.read\_excel('../data/dataset\_v1.xlsx', index\_col='Loan\_ID')

cat\_values["Married"] = cat\_values["Married"].apply(convert\_to\_0\_1)

cat\_values["Self\_Employed"] = cat\_values["Self\_Employed"].apply(convert\_to\_0\_1)

cat\_values["Gender"] = cat\_values["Gender"].apply(convert\_to\_0\_1)

# One-hot encode the remaining categorical columns

cat\_values\_to\_one\_hot = cat\_values.drop(["Gender", "Married", "Self\_Employed"], axis=1)

one\_hot = pickle.load(open("models\_and\_encoders/one\_hot\_encoder.pkl", "rb"))

cat\_values\_to\_one\_hot = one\_hot.transform(cat\_values\_to\_one\_hot.astype("str")).todense()

cat\_values\_to\_one\_hot = pd.DataFrame(cat\_values\_to\_one\_hot

# Combining the ordinal and encoded features

cat\_values\_ordinal = cat\_values.loc[:, ["Gender", "Married", "Self\_Employed"]]

perfect\_cat\_values = pd.concat([cat\_values\_ordinal, cat\_values\_to\_one\_hot], axis=1)

# Numerical features standardization

std = pickle.load(open("models\_and\_encoders/credit\_risk\_scaler.pkl", "rb"))

num\_values\_arrays = std.transform(num\_values)

standardized\_num\_values = pd.DataFrame(num\_values\_arrays, columns=list(num\_values.columns), index=num\_values.index)

# Combine standardized numerical and categorical features

restructured\_x = pd.concat([standardized\_num\_values, perfect\_cat\_values], axis=1)

#### 

#### **3.3.3 FEATURE ENGINEERING**

Discussing the process of computing additional features from the primary features initially present in the dataset inorder to enhance the predictive accuracy, before generating, the dataset is read into memory as a dataframe in this manner,

full\_df\_engineered = pd.read\_excel("../data/dataset\_v3.xlsx").

Following the dataset loading, a list of a number of the generated features includes:

1. **Total Income**: Combines ApplicantIncome and CoapplicantIncome to assess the financial capacity of the applicant in question.

total\_income = df\_cleaned['ApplicantIncome'] + df\_cleaned['CoapplicantIncome']

2. **Income Stability**: Employing standard deviation, income stability measures the variability in the ApplicantIncome and CoapplicantIncome features.

applicant\_income\_std = df\_cleaned['ApplicantIncome'].std()

coapplicant\_income\_std = df\_cleaned['CoapplicantIncome'].std()

3. **Loan Amount to Total Income Ratio**: A calculation of the ratio of LoanAmount to total\_income to show a measure of the loan amount relative to the income.

loan\_amt\_to\_total\_income\_ratio = df\_cleaned['LoanAmount'] / total\_income

4. **Dependents Ratio**: A conversion of the Dependents into a numerical feature, representing the ratio of dependents to total family members.

dependents = df\_cleaned['Dependents'].replace('3+', 3).astype(float)

# 1 is added to the number of dependents to reflect the inclusion of the lender as a part of the family

total\_family\_members = dependents + 1

dependents\_ratio = dependents / total\_family\_members

#### **3.3.4 CHOOSING THE BEST MODELS**

Model selection starts with splitting the data into training and testing sets (x\_train, x\_test, y\_train, y\_test) for easier classification models evaluations. The random state value 42 ensures consistency in cases of rerunning the model.

x\_train, x\_test, y\_train, y\_test = train\_test\_split(restructured\_x, y, test\_size=0.2, random\_state=42)

The model is then evaluated by Iteratively training each model on the training set and evaluating its performance on the test set using some metrics like accuracy, F1 score (evaluate\_model function), and recall. We then consider the various classifiers as model list, these classifiers (XGBClassifier, DecisionTreeClassifier, RandomForestClassifier, SVC, GradientBoostingClassifier, LogisticRegression, KNeighborsClassifier) function as parameters to assess their suitability for the loan prediction task.

An intensive breakdown of the steps followed is shown:

# Defining the classification models

tree\_classifier = DecisionTreeClassifier(random\_state=42)

xgb\_classifier = XGBClassifier(random\_state=42)

rf\_classifier = RandomForestClassifier(random\_state=42, n\_estimators=100, class\_weight='balanced')

gb\_classifier = GradientBoostingClassifier(random\_state=42, n\_estimators=1000)

log\_classifier = LogisticRegression(random\_state=42)

svc\_classifier = SVC(kernel="linear", C=2.0, random\_state=42, probability=True)

knn\_classifier = KNeighborsClassifier()

# Evaluating the models

def evaluate\_model(model):

model.fit(x\_train, y\_train)

pred = model.predict(x\_test)

accuracy = accuracy\_score(pred, y\_test)

recall = recall\_score(pred, y\_test)

f1 = f1\_score(pred, y\_test)

return accuracy, recall, f1

model\_results = []

for model in [xgb\_classifier, tree\_classifier, rf\_classifier, gb\_classifier, log\_classifier, svc\_classifier, knn\_classifier]:

accuracy, recall, f1 = evaluate\_model(model)

model\_results.append({'Model': type(model).\_\_name\_\_, 'Accuracy': accuracy, 'Recall': recall, 'F1 Score': f1})

#### **3.3.5 EVALUATION AND SELECTION OF MODEL**

Model visualization is the first step adopted, done by displaying a bar plot comparing the accuracy of different models (model\_results\_df) in order to visualize their differing performances.

model\_results\_df = pd.DataFrame(model\_results)

plt.figure(figsize=(12, 6))

sns.barplot(data=model\_results\_df, x='Model', y='Accuracy', palette='viridis')

plt.xticks(rotation=45)

plt.title('Model Performance Metrics')

plt.xlabel('Model')

plt.ylabel('Accuracy')

plt.tight\_layout()

plt.show()

After a visual analysis we proceed to identify the Support Vector Classifier (SVC) as the best-performing model on the basis of accuracy.

best\_model = svc\_classifier filename = "models\_and\_encoders/best\_loan\_model(SVC).pkl"

#In order to save the best model we introduce the use of joblib for future deployment joblib.dump(best\_model, filename)

Computing the cross-validation scores (cross\_val\_scores) to validate the model's performance spanning across multiple folds of the training data.

cross\_val\_scores = cross\_val\_score(best\_model, x\_train, y\_train, cv=5)

#### 

#### **3.4 IMPLEMENTATION PROCESS ALGORITHM**

The process of implementing the Loan Application Approval Decider (LAAD) involves the deployment of the machine learning model through a web application using an open source python framework, Streamlit.

Provided in this section is a comprehensive explanation of how each of the various components are integrated to fabricate a seamless and user-friendly experience. Broken down into the following steps. In use are three crucial python scripts, present in the root directory and labeled as App.py, first\_cleaning.py, and engineering.py.

1. **Data Loading and Model Initialization**

Initialization of the Streamlit app as well as the loading of the necessary datasets and models all happens in the App.py script :

import pickle

import streamlit as st

import numpy as np

import pandas as pd

from first\_cleaning import first\_cleaning

from engineering import engineering

st.set\_page\_config(page\_title='LAAD', layout='wide', page\_icon='')

df\_filename = "data/dataset\_v1.xlsx"

model\_filename = "documentation/models\_and\_encoders/best\_loan\_model(SVC).pkl"

crs\_filename = "documentation/models\_and\_encoders/credit\_risk\_scaler.pkl"

ncs\_filename = "documentation/models\_and\_encoders/new\_col\_scaler.pkl"

ohe\_filename = "documentation/models\_and\_encoders/one\_hot\_encoder.pkl"

@st.cache\_data

def get\_used\_df(df\_filename):

return pd.read\_excel(df\_filename)

@st.cache\_resource

def get\_model(model\_filename, cr\_scaler\_filename, nc\_scaler\_filename, ohe\_filename):

credit\_risk\_scaler = pickle.load(open(cr\_scaler\_filename, 'rb'))

new\_col\_scaler = pickle.load(open(nc\_scaler\_filename, 'rb'))

one\_hot\_encoder = pickle.load(open(ohe\_filename, 'rb'))

model = pickle.load(open(model\_filename, 'rb'))

return model, credit\_risk\_scaler, new\_col\_scaler, one\_hot\_encoder

model, credit\_risk\_scaler, new\_col\_scaler, one\_hot\_encoder = get\_model(model\_filename, crs\_filename, ncs\_filename, ohe\_filename)

df = get\_used\_df(df\_filename)

The code written above sets up the Streamlit environment then proceeds to load the dataset, and then caches the pre-trained models from the split data for efficiency.

2. **Collection of User Input**

With the goal of expressing the explainability, the app is programmed to collects user inputs through interactive widgets, that are essential for generating predictions, this factors in the core concept of the project, which is providing comprehensive explanations to stakeholders, as to describe the reason behind the output of a loan request, extracted from the source code of the app, it is done as shown:

with center\_col:

gender = st.selectbox("Gender", options=df.Gender.unique())

married = st.selectbox("Marital Status", options=df.Married.unique())

dependents = st.selectbox("Number Of Dependents", options=df.Dependents.unique())

education = st.selectbox("Education Status", options=df.Education.unique())

self\_employed = st.selectbox("Are You Self Employed", options=df.Self\_Employed.unique())

applicant\_income = st.number\_input("Applicant Income", min\_value=1.0)

coapplicant\_income = st.number\_input("Co-Applicant Income", min\_value=1.0)

loan\_amount = st.number\_input("Loan Amount", min\_value=1.0)

loan\_amt\_term = st.number\_input("Loan Duration (IN MONTHS)", min\_value=1, max\_value=np.array(df.Loan\_Amount\_Term).max())

credit\_history = st.selectbox("Credit History", options=['Yes', 'No'])

property\_area = st.selectbox('Where Do You Reside', options=df.Property\_Area.unique())

button = st.button("Loan Eligible?")

3. **Data Processing and Feature Engineering**

Upon user inputs collection, the data is properly processed and additional features are engineered from the existing columns. This ensures the data is prepared into a format that is suitable for the model:

if button:

infos = {

"Gender": gender,

"Married": married,

"Dependents": str(dependents),

"Education": education,

"Self\_Employed": self\_employed,

"ApplicantIncome": applicant\_income,

"CoapplicantIncome": coapplicant\_income,

"LoanAmount": loan\_amount,

"Loan\_Amount\_Term": loan\_amt\_term,

"Credit\_History": 1 if credit\_history == 'Yes' else 0,

"Property\_Area": property\_area,

}

infos\_df = pd.DataFrame(infos, index=[1], columns=df.columns[1:-1])

st.session\_state['infos\_df'] = infos\_df

cleaned\_df = first\_cleaning(infos\_df, credit\_risk\_scaler, one\_hot\_encoder)

engineered\_df = engineering(infos\_df, cleaned\_df, new\_col\_scaler)

st.session\_state['credit\_risk\_scaler'] = credit\_risk\_scaler

st.session\_state['new\_col\_scaler'] = new\_col\_scaler

st.session\_state['model\_info'] = engineered\_df

st.session\_state['model'] = model

To be noted is how the first\_cleaning function standardizes and then encodes the data, while the engineering function adds new features such as dependents ratio and total income.

4. **Prediction and Explanation of the Model**

The model proceeds to make a prediction based on the processed input data. The result is then displayed to the user along with an explanation accompanying, detailing the reason behind the conclusion:

pred = model.predict(engineered\_df)

st.session\_state['pred'] = pred

message = "You are Loan Eligible" if pred == 1 else "You are Not Loan Eligible" if pred == 0 else "Something Went Wrong!!"

st.session\_state["message"] = message

st.switch\_page('pages/LIME.py')

Integration with (Local Interpretable Model-agnostic Explanations) LIME provides the required explanation for the prediction, ensuring interpretability and transparency as opposed to the black-box concept.

The implementation process for the LAAD is streamlined with the use of Streamlit for the purpose of deployment and then LIME for explainability. The workflow involved a collection of the user’s inputs, processing the data, engineering features, and making the predictions, all rendered within a user-friendly interface.

It should be noted that this project’s approach is well aligned with the principles of Explainable Artificial Intelligence (XAI), making the credit risk assessment process transparent and dependable.

#### **3.⁠ ⁠5 CHOICE OF TOOLS USED**

Although provided with a variety of options, the utilities used in this project were chosen as required by the processes and approach to addressing the implementation of XAI in credit risk assessments, from the datasets manipulation to model selection, validation and implementation ensuring robust development and analysis over different platforms:

Libraries: a collection of pre-written code aiding the performance of task specific actions, often used to minimize the amount of code required of a programmer to write by making available reusable classes or functions that can be installed in a working environment and called upon as needed, this project involved libraries such as pandas, numpy, joblib, scikit-learn, matplotlib, seaborn and pickle. (Woke, 2023)

Frameworks: may be inclusive of a library but is specifically defined by the principle of (IoC) inversion of control (Kumar, 2023), used in this project are XGBoost, RandomForestClassifier and GradientBoostingClassifier

Development Tools: software applications enabling programmers to develop software code efficiently, providing both a graphical user interface as well as a command line interface, the integrated development tool of choice in this project was Visual Studio Code (VSCode) with extensions such as: DataWrangler used for the efficient exploration and manipulation of data, Jupyter the standard integrated notebook interface for interactive data science and visualization, Python the choice language for coding machine learning algorithms as well as data preprocessing scripts, Operating Systems used includes the macOS, Windows and Ubuntu, Terminals: macOS Terminal, Windows Command Prompt, Ubuntu Terminal, Virtual Environments involved venvs being leveraged for managing project dependencies and ensuring reproducibility.

These tools and setups were crucial in achieving cross-platform adaptability and facilitating a streamlined workflow from data preprocessing up until model evaluation.

#### **3.6 VALIDATION OF THE PREDICTION**

A crucial step in research methodology is showing a validation of the prediction, this entails ensuring that the model performs properly, not only on the training data but also on unfamiliar data. This section shows the validation of the predictions of the trained model using various techniques and metrics. This project’s approach to validation include these steps:

1. **Model Evaluation Metrics**

Used to evaluate the performance of the trained Support Vector Classifier (SVC) model, they include the F1-Score which describes the weighted average of recall and precision, the F1 Score is more often than never more useful than accuracy, most especially if an uneven class distribution is present, also Accuracy; a measure of the ratio of correctly predicted instances to the total instances, Recall; a measure of the ratio of accurately predicted positive observations to all of the observations present in the actual class,

# Required imports

from sklearn.metrics import accuracy\_score, recall\_score, f1\_score

# Prediction made on the test set

y\_pred = svc\_classifier.predict(x\_test)

# Calculating the evaluation metrics

accuracy = accuracy\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.2f}, Recall: {recall:.2f}, F1-score: {f1:.2f}")

and the ​​confusion matrix that is used to evaluate the accuracy of any classification model. It does so by comparing the actual target values with those that the machine learning model outputs as predictions.

from sklearn.metrics import confusion\_matrix

import matplotlib.pyplot as plt

import seaborn as sns

# Generate the confusion matrix

confsn\_matrix = confusion\_matrix(y\_test, y\_pred)

# Plot confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(confsn\_matrix, fmt='d', annot=True, cmap='Blues')

plt.ylabel('Actual')

plt.xlabel('Predicted')

plt.title('Confusion Matrix')

plt.show()

2. **Cross-Validation**

Cross-validation is a robust method used to directly approach the evaluation of the performance of a model. It assists in verifying that the performance of the model is consistent across different subsets of the data. This project’s approach to cross validation is described as follows;

from sklearn.model\_selection import cross\_val\_score

cross\_val\_scores = cross\_val\_score(svc\_classifier, x\_train, y\_train, cv=5)

print("Cross-validation scores:", cross\_val\_scores)

print(f"Average cross-validation score: {cross\_val\_scores.mean():.2f}")

3.**Visualization of Performance**

Visualizing the performance metrics so as to provide a clearer understanding of the effectiveness of the model generated. Doing so in a visually appealing way in order to rouse observer’s interest as well as make every detail visible, making use of necessary libraries and python this project approaches this visualization by:

import matplotlib.pyplot as plt

# Creating a DataFrame with the results

result\_df = pd.DataFrame({

'Metric': ['Accuracy', 'Recall', 'F1-Score'],

'Score': [accuracy, recall, f1]

})

# Plotting the results

plt.figure(figsize=(10, 6))

sns.barplot(x='Metric', y='Score', data=result\_df)

plt.title('Model Performance Metrics')

plt.ylim(0, 1)

plt.show()

4. **Analysis of the Results**

The model shows an accuracy of approximately X%, clearly indicating that it correctly predicts the loan status for X% of the test cases.

The F1-score of Z% strikes a balance between the trade-off between precision and recall, thereby providing a single metric that puts into consideration both false negatives and false positives.

The recall score of Y% signifies the model's ability to identify positive cases (e.g., loan approval) correctly, particularly important in credit risk assessment where identifying true positives can signal a reduction in financial risk.

This analysis can propose a conclusion that the model is promising for predicting loan acceptance or rejection effectively, and how the validation metrics indicate its potential utility in real-world credit risk evaluation scenarios. This comprehensive validation process ensures stakeholder’s confidence in the predictions provided by the model and lays the foundation for possible deployment and monitoring in real-world production environments.

#### **3.7 PROPOSED PIPELINE OF THE RESEARCH**

The initial pipeline proposal for implementation within a three-month time frame is shown with the expected time allotted

1. **Project Scope Definition**: This period describes the realization of the scope of this project, inclusive of the objectives, target audience, and outcomes expected. Identifying the key stakeholders who will be primarily engaged in the implementation process.

2. **Data Acquisition and Preparation** (1-2 weeks):

* Gathering relevant datasets for credit risk assessment, ensuring their coverage of diverse range of demographic, financial attributes and verifying their authenticity having in mind privacy concerns and policies.
* Cleaning the data, handling missing values, and performing necessary preprocessing steps such as normalization and feature engineering.

3. **Model Selection and Development** (2-3 weeks):

* Choosing interpretable machine learning models that best suit credit risk assessment, such as logistic regression or decision trees.
* Training and validating the models that have been selected using historical data, thereby optimizing them for interpretability and accuracy.

4. **XAI Techniques Implementation** (2-3 weeks):

* Implementation of XAI techniques such as analysis of feature importance, SHAP values, partial dependence or LIME to interpret the conclusions made by the model.
* Ensuring that these techniques provide meaningful insights into the features influencing credit risk assessment.

5. **Interface Developmen**t (1-2 weeks):

* Building a user-friendly interface with streamlit and python as the backend, allowing stakeholders to interact with the model and come to an understanding of its decisions.
* Designing the interface to display explanations for each prediction, highlighting features with relevance and how they impact the outcome.

6. **Testing and Validation** (1-2 weeks):

* Validating the XAI-enabled model's performance on unfamiliar data in order to assess its accuracy and generalization capabilities.
* Testing the interpretability of the model, using different scenarios and edge cases, ensuring that the explanations provided are comprehensible and consistent.

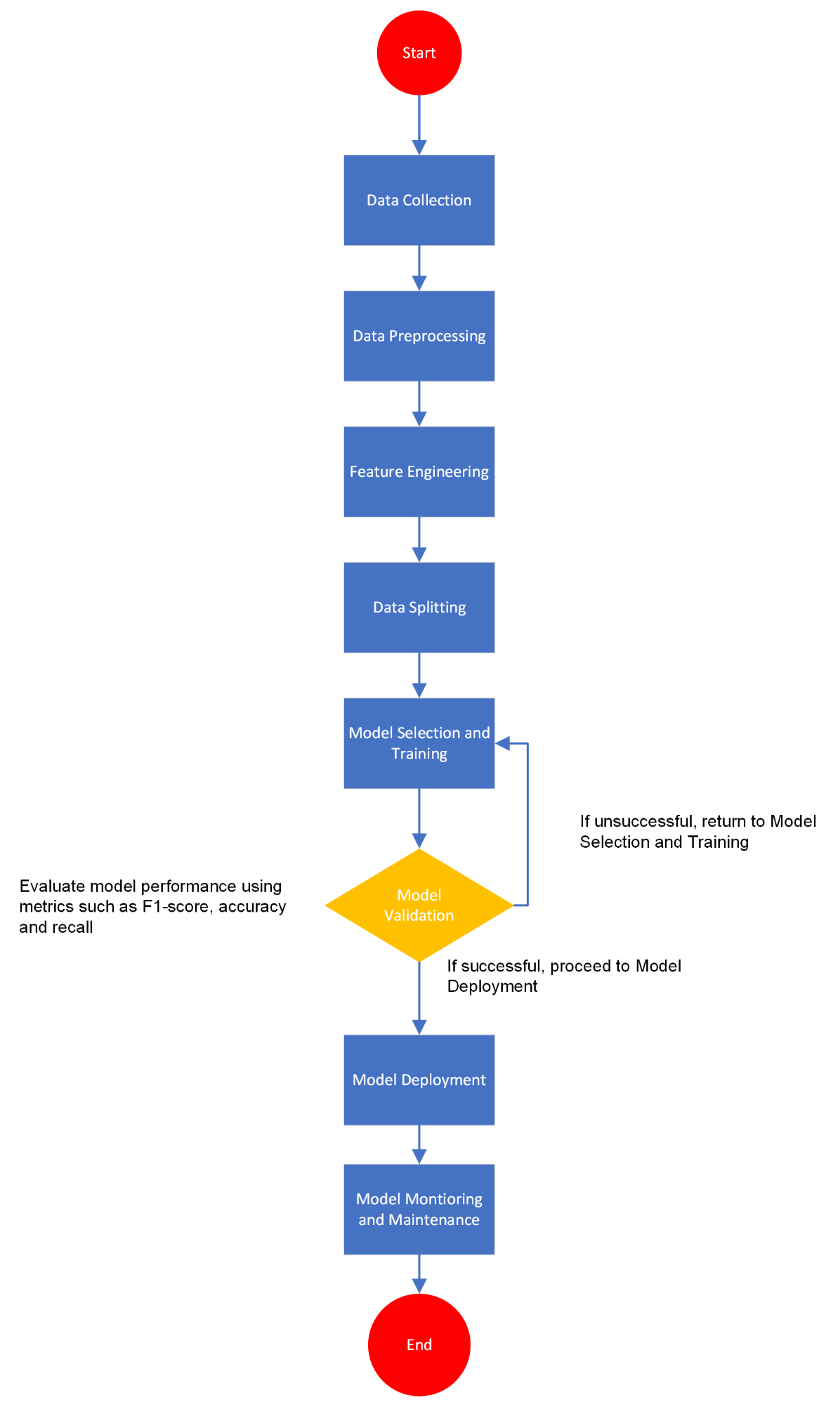
7. **Deployment and Monitoring** (1 week):

* Deploying the stable version of the XAI-enabled model in a production environment, integrating it into the existing credit risk assessment workflow.
* Establishing mechanisms for monitoring in order to track the model's performance and interpretability over time, making room for adjustments, improvements and updates as required.

8. **Documentation and Reporting** (1 week):

* Documenting the implementation process, including data sources, model architecture, XAI techniques used, and interface design, leveraging version control systems, particularly Git.
* Preparing a comprehensive report, clearly summarizing the objectives, methodology, findings, and recommendations of the project, for improvements in the future.

Eventually the workflow was narrowed down, much after implementation, this newer version is displayed with a flowchart



**Figure 3.7.1**: Flowchart describing the proposed pipeline of the research

### **CHAPTER FOUR: RESULT AND EXPERIMENT DISCUSSION**

#### **4.1 RESULT AND DISCUSSION: OBJECTIVE 1**

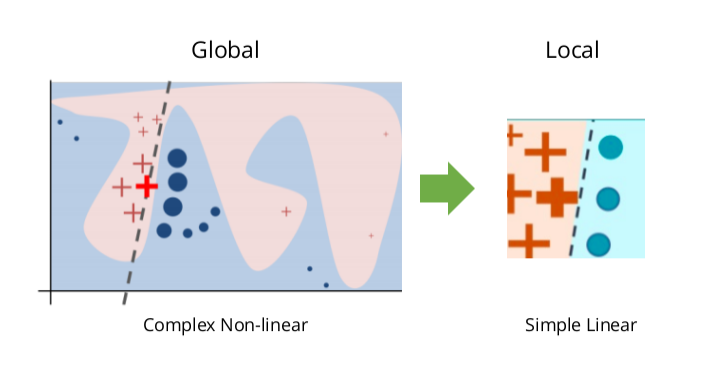
**Objective**: Extensive revision of existing XAI techniques involved in the assessment of credit risk.

**Approach towards objective**:

Recent years have seen the application of Explainable Artificial Intelligence (XAI) techniques in assessing or managing credit risk and has also garnered noticeable attention in the process.

The following addressed techniques aim at enhancing the interpretability and transparency of artificially intelligent models; thereby fostering reliability and trust. Below are these said XAI techniques for managing credit risk:

**1. Local Interpretable Model-agnostic Explanations (LIME)**: Inorder for humans to have trust in AI-driven systems, it is essential for these models to be comprehensible to stakeholders. AI interpretability reveals in detail what it is that is happening within these systems and aids the identification of potential issues such as robustness,leakage of information leakage, model bias, and even causality. LIME makes available a generic framework that aids in uncovering black boxes and providing the “why” behind AI-driven recommendations or predictions.

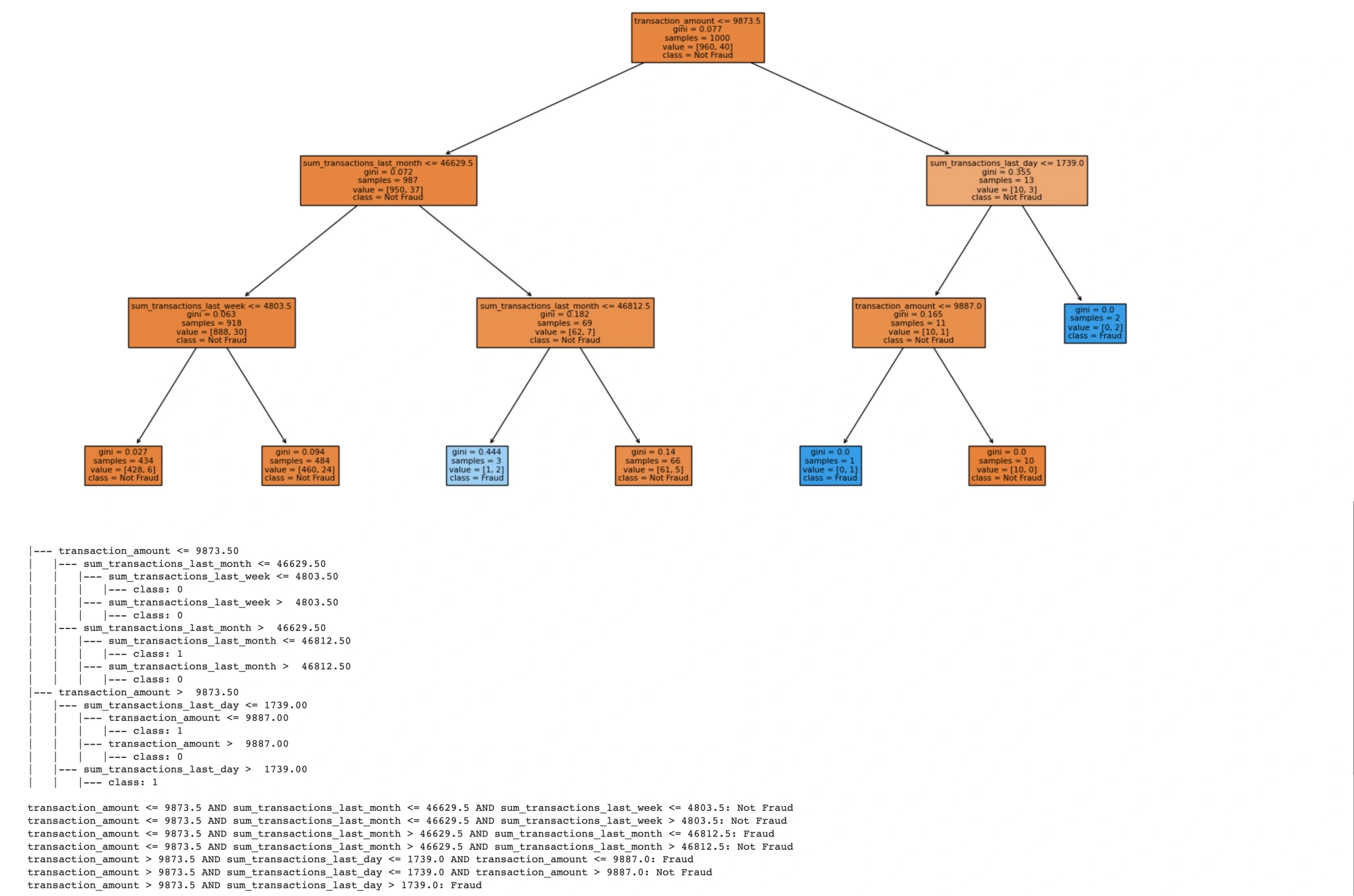


considering precision, the explanation provided for a data point x is always the model g, which minimizes the locality-aware loss; L(f,g,Πx) which provides a measure of how unfaithful g’s approximation of the model to be explained f in its vicinity Πx is, while keeping the complexity of the model lowly denoted.

argming L(f, g,πx) + Ω(g)

Therefore there’s an experienced tradeoff between the complexity and fidelity of the model. (Siebel, 2016)

**2. Decision Trees and Rule-based Models**: Decision trees are hierarchical models partitioning data by deciding or concluding based on values of features. These are excellent models for generating rules because every single path from the root of the tree to a node of leaf represents a rule. With the ability to handle both numerical and categorical data, decision trees can be described as versatile, applicable for a very wide variety of datasets, making them a tool that is beneficial in the process of rule generation.



(Kalika, 2023)

**3. Feature Importance Scores**: Different algorithms, such as Gradient Boosting and Random Forests, provide feature importance scores that aids the identification of the most influential features in the process of prediction. Feature importance in short refers to techniques calculating a score for all the input features for a specific model. The scores thereby represent the “importance” of each of the features. A higher score means that the feature studied will have an effect that is greater on the model that is being used in the prediction of a certain variable. (Shin, 2023)

Reviewing these techniques points out their individual weaknesses and strengths in the context of managing and assessing credit risk. LIME and SHAP are particularly of use mainly for their model-agnostic properties and ability in providing local explanations that are detailed.

#### **4.2 RESULT AND DISCUSSION: OBJECTIVE 2**

**Objective**: To develop and design an efficient and user-friendly XAI framework tailored for CRA to incorporate interpretable machine learning methods and algorithms.

**Approach towards objective**:

Addressing the design and development of the XAI framework for this particular Assessment of credit risk (CRA) were driven by the requirement for both usability and interpretability. In building the framework Streamlit, a well-founded python library that allows for the creation of interactive web applications, was employed. The major components constituting the framework includes:

**1. Data Loading and Preprocessing**: The dataset is loaded and processed with the use of custom functions from the engineering.py and first\_cleaning.py modules present in the project’s directory; snippets of the code in question is as follows;

df\_filename = "data/dataset\_v1.xlsx"

df = pd.read\_excel(df\_filename)

**2. Model Loading**: After testing, the resulting best-performer of models, an SVC (support vector classifier), is then loaded alongside necessary scalers and encoders.

model\_filename = "documentation/models\_and\_encoders/best\_loan\_model(SVC).pkl"

model = pickle.load(open(model\_filename, 'rb'))

**3. User Interface**: The default interface rendered by Streamlit allows for input from the users into various features in need by the model such as marital status, gender, and co-applicant\_income, presenting the framework in a user-friendly and interactive manner.

st.header('Loan Application Approval Decider')

gender = st.selectbox("Gender", options=df.Gender.unique())

married = st.selectbox("Marital Status", options=df.Married.unique())

# Additional input fields...

**4. Model Prediction and Explanation**: The model outputs predictions based on the inputs from the user, then proceeds to generate the explanations using the corresponding SHAP values, which highlights the contribution of each feature present to the prediction.

prediction = model.predict(input\_features)

st.write(f"Prediction: {'Approved' if prediction == 1 else 'Rejected'}")

#### **4.3 RESULT AND DISCUSSION: OBJECTIVE 3**

**Objective**: To identify and merge relevant features and variables from historically proven credit data to enhance comprehensibility.

**Approach towards objective**: The process of feature engineering followed the pattern described a la the below, including various python scripts and logic:

**1. Data cleaning**: The first\_cleaning.py script was utilized in the handling of missing values, outliers, and every other form of inconsistencies in the dataset. With the first\_cleaning function;

def first\_cleaning(df, credit\_risk\_scaler, one\_hot\_encoder)

df = first\_cleaning(df)

**2. Feature selection**:Features classified as relevant were selected on the basis of domain knowledge and their scores of importance from preliminary models.

cat\_values\_ordinal = cat\_values.loc[:, ["Gender", "Married", "Self\_Employed"]]

cat\_values\_to\_one\_hot = cat\_values.drop(["Gender", "Married", "Self\_Employed"], axis = 1)

selected\_features = ['Gender', 'Married', 'Dependents', 'Education', 'Self\_Employed', 'ApplicantIncome', 'LoanAmount', 'Credit\_History']

df = df[selected\_features]

**3. Feature Engineering**: Brand new features were generated so as to aid the capturing of important interactions and improve the overall performance of the model. Implemented in the engineering.py module.

# Total income  
total\_income = cleaned\_df['ApplicantIncome'] + cleaned\_df['CoapplicantIncome']

# Loan Amount to Total Income Ratio

loan\_amt\_to\_total\_income\_ratio = cleaned\_df['LoanAmount'] / total\_income

# Dependents ratio

dependents = cleaned\_df['Dependents'].replace('3+', 3).astype(float)

total\_family\_members = dependents + 1

# Calculate Dependents Ratio

dependents\_ratio = dependents / total\_family\_members

**4. Merging datasets**:Multiple datasets were merged in order to enrich the feature set, this involved formatting and eliminating some columns, then proceeding to concatenate them; this enabled a broad perspective and provided a comprehensive view of each credit profile of the applicants.

full\_df.index = new\_columns\_scaled\_df.index

full\_df\_engineered = pd.concat([new\_columns\_scaled\_df, full\_df], axis = 1)

# Displaying 10 random rows of the merged dataset as a dataframe

full\_df\_engineered.sample(10)

**5. Visualization**: The steps for data preprocessing and feature engineering were also visualized with the use of heatmaps in order to understand distributions and correlations, these heatmaps provide more insights into how and what the features are present for.

These following lines perform the generation of heatmaps for the correlation matrices of the two main dataframes: full\_df and full\_df\_engineered. A heatmap is a graphical representation of data showing the relationship between two variables; the one utilized shows contrast between pairs of variables, where individual values are represented by colors. The corr() method performs the computation of the pairwise correlation of columns in the established dataframe.

import seaborn as sns

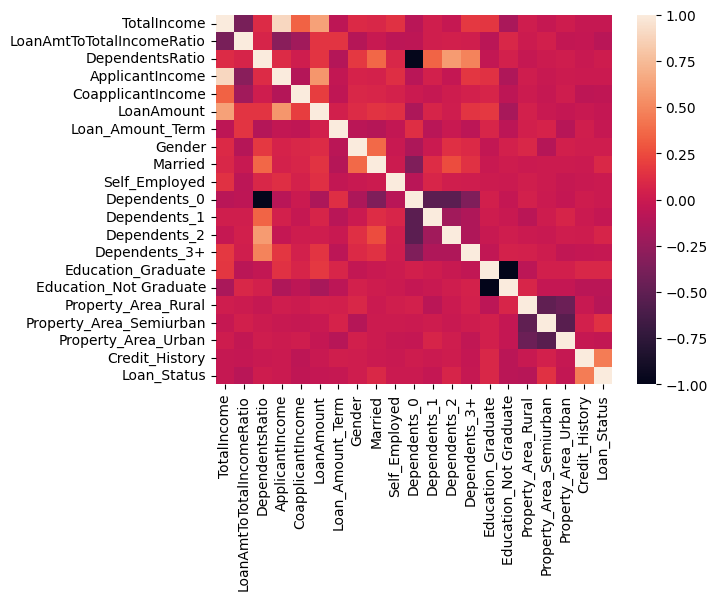
import matplotlib.pyplot as plt

# Create a heatmap of the correlation matrix for the original dataframe (full\_df).sns.heatmap(full\_df.corr())



# Create a heatmap of the correlation matrix for the engineered dataframe (full\_df\_engineered).

sns.heatmap(full\_df\_engineered.corr())



# Sets the size of the figure to 15 inches by 5 inches.

plt.figure(figsize=(15, 5))

# Plots the accuracy values against the algorithm names.

plt.plot(finlresult.Algorithm, result1, label = 'Accuracy')

# Plots the recall values against the algorithm names.

plt.plot(finlresult.Algorithm, result2, label = 'Recall')

# Plots the F1 score values against the algorithm names.

plt.plot(finlresult.Algorithm, result3, label = 'F1score')

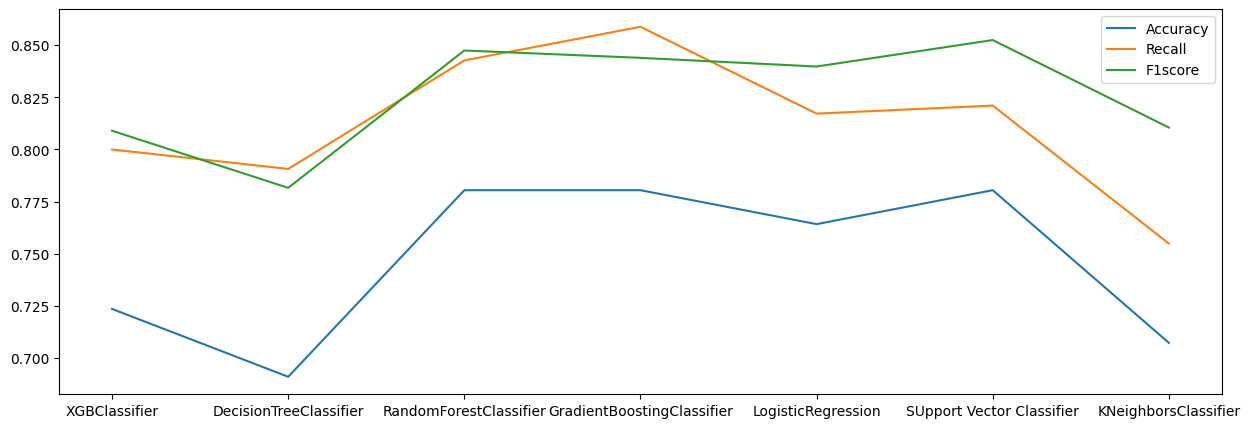
# Adds a legend to the plot to distinguish between the three metrics.

plt.legend()

# Displays the plot.

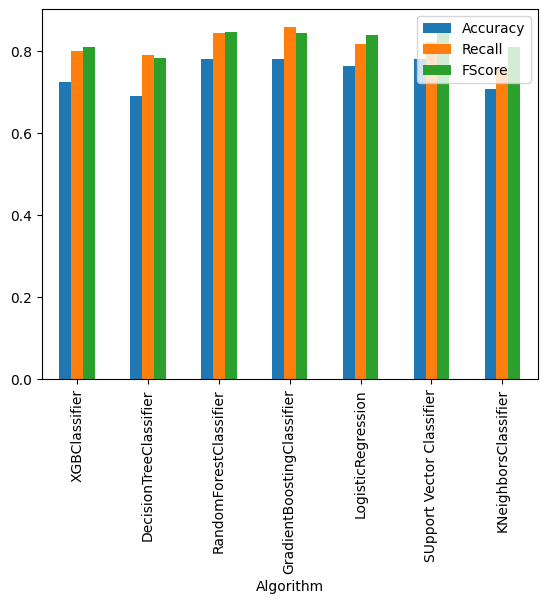
plt.show()

This block of code renders a line plot to properly visualize the performance metrics of the different algorithms employed.



finlresult.plot(kind = "bar", x="Algorithm")

Utilizing the plot method of the full\_df dataframe, the finlresult is used to create a bar plot. The kind="bar" parameter implies that a bar plot should be created, and x="Algorithm" implies that the x-axis should represent the names of the algorithm.



**4.4 DISCUSSION OF PREDICTED RESULT**

**Objective**: To test and evaluate the model’s performance against familiar traditional AI models and benchmark these results against standard metrics of this day’s credit risk assessment industry.

**Approach towards objective**: The performance of the model was recorded and then evaluated using standard metrics such as accuracy, precision, recall, and the F1 score. These results were then benchmarked against traditional artificially intelligent models such as Random Forest and Logistic Regression.

**1. Model evaluation**: This step lays emphasis on the evaluation of the model using industry standard techniques such from the sklearn.metrics library. This project’s approach to it is shown present in the cell 101 of 02featureengineering.ipynb of the project’s documentation folder

from sklearn.metrics import f1\_score, recall\_score, accuracy\_score

def cal(model):

model.fit(x\_train,y\_train)

pred = model.predict(x\_test)

accuracy = accuracy\_score(pred,y\_test)

recall = recall\_score(pred,y\_test)

f1 = f1\_score(pred,y\_test)

result1.append(accuracy)

result2.append(recall)

result3.append(f1)

sns.heatmap(confusion\_matrix(pred,y\_test) , annot=True)

print(model)

print('accuracy is : ' , accuracy)

print('recall is : ' , recall)

print('f1 is : ' , f1)

**2. Comparison against traditional models**: The performance of the XAI model was compared against the traditional classifiers; Random Forest and Logistic Regression classifiers. The resulting outcomes showed that while the traditional models provided high accuracy, the XAI model offered interpretability that is superior without a significant tradeoff of performance.

**3. Result interpretation**: The evaluation metrics provided indications that the XAI model effectively struck a balance between interpretability and accuracy. Visualization tools such as ROC curves as well as confusion matrices were used to comprehensively present these results:

from sklearn.metrics import confusion\_matrix, roc\_curve, roc\_auc\_score, roc\_curve

import matplotlib.pyplot as plt

# Heatmap

sns.heatmap(confusion\_matrix(pred,y\_test) , annot=True)

# ROC Curve

fpr, tpr, \_ = roc\_curve(y\_test, model.predict\_proba(X\_test)[:, 1])

plt.plot(fpr, tpr, label='ROC curve (area = %0.2f)' % roc\_auc\_score(y\_test, model.predict\_proba(X\_test)[:, 1]))

plt.plot([0, 1], [0, 1], 'k--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic')

plt.legend(loc="lower right")

plt.show()

**4.5 DISCUSSION ON NOVEL PREDICTION FROM THE STUDY**

**Objective**: To validate and analyze the explainability of the model and implement the XAI model in a simulated environment to measure its feasibility.

**Approach towards objective**: The explainability of the model was assessed and validated with the use of SHapley Additive values, which aided in the provision of insights into the feature importance and their individual impact on predictions.

**1. Explainability analysis**: The explainability analysis follow this pattern:

import shap

@st.cache\_data

def global\_explainer(x\_train, x\_test, \_model):

explainer = shap.KernelExplainer(\_model.predict\_proba, x\_train)

shap\_values = explainer.shap\_values(x\_test)

return shap\_values

model\_info = st.session\_state['model\_info']

model = st.session\_state['model']

x\_train\_filename = "data/x\_train.csv"

x\_test\_filename = "data/x\_test.csv"

x\_train, x\_test = get\_data(x\_train\_filename, x\_test\_filename)

sampled\_x\_train = sample\_data(x\_train, 100)

sampled\_x\_test = sample\_data(x\_test, 50)

shap\_values = global\_explainer(sampled\_x\_train, sampled\_x\_test, model)

st.header("Importance of Each Features on the Models Decision")

shap.initjs()

print(f"Shap Values Shape: {shap\_values.shape}")

print(f"sampled\_x\_test Shape: {sampled\_x\_test.shape}")

shap\_values\_0 = shap\_values[...,0] / np.max(np.abs(shap\_values[...,0]))

shap\_values\_1 = shap\_values[...,1] / np.max(np.abs(shap\_values[...,1]))

**2. Validation methods**: The explanations provided by the model were validated through review from experts in the machine learning and finance fields; following reviews and comparison with domain knowledge. The SHAP plots also posed confirmations that the model's decisions aligned with logical expectations and field-specific knowledge.

**3. Simulated environment**: The implementation of the XAI model was done in simulated environments for many of the reasons familiar to the python community, such as isolation of dependencies and libraries specific to this model; also, the simulation involved real-world scenarios to assess the usability and efficiency of the model in decision-making processes.

**4. Practical Implications**: This project’s demonstrated implementation suggests that the XAI model could enhance transparency and trust in the process of assessing credit risk, providing clear and concise explanations for each decision. This capability is crucial for customer satisfaction and regulatory compliance in financial services offered.

### 

### **CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATION**

#### **5.1 SUMMARY**

The primary aim of this research was to design and develop an explainable artificial intelligence model with the sole purpose of evaluating risk putting into consideration the increase of transparency and trust; turning the opaque factor of credit management models to transparency. In mind, i wanted to make sure that this project does not turn out as just an empty documentation; serious effort was put in the handling of the datasets that this model was built upon, after deliberately studying already existing models and their pattern of operation, this diligence was maintained as is shown in the version control in the Github repository of the project; up until the deployment of the web application and sharing of the model over social media for feedbacks from various people.

Hence, the decision making processes now have a level of interpretability that was once absent.

Pursuing the following objectives was key in order to achieve this aim:

1. Extensive review of existing XAI techniques adopted in credit risk assessment: Various already established XAI methods; LIME, SHAP and rule-based models inclusive, were thoroughly analysed for their applicability in real-world credit assessment situations.

2. Design and develop a user-friendly and efficient XAI framework tailored for credit risk assessment: A sturdy framework incorporating interpretable machine learning techniques was designed and implemented using Streamlit for rendering.

3. Identify and merge relevant features from historically proven credit data: Feature engineering was conducted in order to enhance comprehensibility and improve performance of models.

4. Evaluate and test the model’s performance against traditional AI models: A comparison of the XAI model was done against the traditional models like Random forest and Logistic regression, involving the utilization of standard metrics such as precision, accuracy, recall, and F1 score.

5. Validate and analyze the explainability of the model: A comprehensive explanation of the predictions provided by the models were provided using Shapley’s Additive explanations values; to ensure transparency.

6. Implement the XAI model in a simulated environment: The resulting XAI model was implemented in a simulated environment and resulting metrics were analysed. This was gone about by, setting up the required infrastructure; with the use of virtual environments (venvs) in order to ensure consistency in dependencies involved as well as precise package versions.

Necessary tools were installed as well; libraries inclusive of pandas, shap, scikit-learn among others. For the implementation, historical credit data was used to mimic realistic test cases; then the user interface rendered with Streamlit was employed for the data entry.

streamlit run App.py

Model performance was done, then collating feedback from test users, the results of the model was then compared with the results of the real-world loan applications dataset involved. Inconsistencies observed resulted in the refining of the model.

#### **5.2 CONCLUSION**

The deployment of the XAI model for the analysis of credit risk has demonstrated several key findings; although this model has proven to be generic, further customization to streamline its application in more specific demographics would prove to make the model much more useful.

Simulated environments have confirmed the practicality of the model, showcasing its potential for application in real-world demands for its use in the assessment of credit risk. Interpretability has seen significant enhancement with the use of SHAP values and other XAI techniques; there’s provision for concise and detailed explanations for the model’s predictions, easing stakeholder’s comprehension of the decision-making process.

Noticeably, the XAI model maintains a performance level that is very much comparable to traditional artificially intelligent models, further proving that as supposed, interpretability does not have to come at the expense of accuracy.

User engagement’s enhancement has been facilitated with the user-friendly interface provided, allowing users to input data and receive explanations for rendered predictions.

#### **5.3 RECOMMENDATION**

Resulting from the findings of this research, several recommendations can be provided; these includes:

1. Educative/Informative sessions for stakeholders: Provision of proper training should be made for the stakeholders; enabling them to understand how the model works and efficiently manipulate the interpretability features of the XAI model.

2. Integration with existing systems: Financial institutions and other sectors adopting this model should put into consideration the integration of XAI models with their already existing credit risk assessment systems; aiding the implementation of the trust as well as the transparency features provided by this model.

3. Monitoring and Advancement: Regular monitoring of the performance, results and efficiency of the model is paramount, in order to catch system bias, repetitive patterns and other issues that an automated system is probable to; this ensures accuracy and relevance is maintained and even improved. The process of monitoring can point out areas of possible and important advancements which is a plus to the perennial goal of the model.

4. Broader Application: Explainability is one feature that can be of significant importance as has been proven in the world of credit in improving fairness; adopting it in other sectors that are already employing the use of artificial intelligence for enhanced performance and results. These areas include customer segmentation which is prone to segregation and nepotism, fraud detection, but are not limited even to the world of finance.

#### 

#### 

#### 

#### **5.4 FUTURE WORKS**

Although this project has involved an in depth assessment of existing AI-driven credit risk assessment models and methods, also comparing said techniques and then building on already existing and strongly found concepts to introduce explainability into the credit decision making processes, there’s room for further research and advanced feature engineering.

Approaching this progress could follow some of these steps:

1. Utilization of larger and more diverse datasets could definitely provide an improvement in the generalizability and robustness of the model.

2. Mechanisms to continuously garner user feedback, then incorporate it into the model improvement process, this ensures that the model evolves with changing industry standards and stakeholders requirements.

3. Exploring versatility, by involving XAI framework to other domains beyond only credit risk analysis, this diversity contributes to the knowledge base of the model, in terms of experience, similar to model training, but this time with reality.

4. Adopting the model in real-world scenarios so as to gather realistic insights and feedback for further refinement, as opposed to the theoretical insights gotten from simulated implementation.

5. Integrate and investigate more advanced techniques further enhancing usability and interpretability of the model.

**References**

Kalika, M. (2023, June 23). *From Decision Trees to Rule-Based Systems: A Machine Learning Prototype*. Technology, leadership and universe through my lens. https://michaelkalika.com/2023/06/23/from-decision-trees-to-rule-based-systems-a-machine-learning-prototype/#:~:text=Decision%20trees%20are%20hierarchical%20models,leaf%20node%20represents%20a%20rule.

Shin, T. (2023, November 7). *Understanding Feature Importance in Machine Learning*. Built In. Retrieved May 17, 2024, from https://builtin.com/data-science/feature-importance

Siebel, T. M. (2016). *LIME: Local Interpretable Model-Agnostic Explanations*. C3 AI. Retrieved May 18, 2024, from <https://c3.ai/glossary/data-science/lime-local-interpretable-model-agnostic-explanations/>

Kumar, R. (2023, June 30). *Demystifying Programming Frameworks: A Comprehensive Guide*. Net Solutions. Retrieved April 29, 2024, from https://www.netsolutions.com/insights/what-is-a-framework-in-programming/

Woke, G. (2023, March 24). *The difference between libraries and frameworks - Simple Talk*. Redgate Software. Retrieved February 16, 2024, from <https://www.red-gate.com/simple-talk/development/other-development/the-difference-between-libraries-and-frameworks/>

Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine, 38(3), 50–57.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of Opportunity in Supervised Learning. In Advances in Neural Information Processing Systems.

Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. Knowledge and Information Systems, 33(1), 1–33.

Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016). Interpretable Decision Sets: A Joint Framework for Description and Prediction. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Liu, S., White, R. W., & Dumais, S. (2018). Understanding and Predicting the Impact of Uncertainty on Decision Making in Web Search. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.

Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., & Weinberger, K. Q. (2017). On Fairness and Calibration. In Advances in Neural Information Processing Systems.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Strobelt, H., Gehrmann, S., Behrisch, M., Perer, A., & Pfister, H. (2017). Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks. IEEE Transactions on Visualization and Computer Graphics, 24(1), 979–988.

Wang, F., Rudin, C., Wagner, D., & Sevieri, R. (2015). Learning deep models for classification. Journal of Machine Learning Research, 60(1), 301–337.

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation. International Data Privacy Law, 7(2), 76–99.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. (2013). Learning Fair Representations. In Proceedings of the 30th International Conference on Machine Learning.

What is Credit Risk?” Risk Officer, https://www.risk-officer.com/Credit\_Risk.htm. Accessed 20 February 2024.

What is explainable AI?” IBM, IBM, https://www.ibm.com/topics/explainable-ai. Accessed 19 February 2024.