**BPP Coursework Cover Sheet**

|  |  |
| --- | --- |
| Programme |  |
| Module name |  |
| Student Reference Number (SRN) |  |
| Report/Assignment Title |  |
| Date of Submission  *(Please attach the confirmation of any extension received)* |  |
|  | |
| Declaration of Original Work:   * I hereby declare that I have read and understood BPP’s regulations on plagiarism and that this is my original work, and that I have researched, undertaken, completed and submitted in accordance with the requirements of BPP School of Business. * I declare that where I have used any AI tools, it was for the following reasons (highlight as appropriate):   + To review a draft   + To correct language errors   + Other (please describe) * I have copied & retained for BPP University’s reference, all AI prompts used in the creation of AI content and all AI-generated responses in support of my assignment and attached relevant evidence in the appendices. * I understand that I may be required to participate in a *viva voce,* where I will be questioned on any aspect of my assignment, including key concepts, theories, examples used, & any sources included. * The word count, excluding contents table, bibliography and appendices, is 2490 words.   **Student Reference Number**: **Date**: | |



By submitting this coursework you agree to all rules and regulations of BPP regarding assessments and awards for programmes. **Please note, submission is your declaration you are fit to sit.**

BPP University reserves the right to use all submitted work for educational purposes and may request that work be published for a wider audience.

**BPP School of Business**

Table of Contents

[1 Introduction 4](#_Toc201606713)

[1.1 Current Business Environment 4](#_Toc201606714)

[1.2 Problems Addressed by the Programming Solution 4](#_Toc201606715)

[1.3 Impact and Benefits of Proposed Development 4](#_Toc201606716)

[1.4 Implications of Non-Proceeding 4](#_Toc201606717)

[1.5 User Requirements and Tools 4](#_Toc201606718)

[1.6 Data Supplied 5](#_Toc201606719)

[1.7 Development and Maintenance Challenges 5](#_Toc201606720)

[1.8 Regulatory and Ethical Implications 5](#_Toc201606721)

[2 Approach 5](#_Toc201606722)

[2.1 Technical Implementation Framework 5](#_Toc201606723)

[2.2 Loading and Preparing the Data for Analysis 6](#_Toc201606724)

[2.3 Exploratory Data Analysis 7](#_Toc201606725)

[2.3.1 Summary Statistics of the Data 7](#_Toc201606726)

[2.3.2 Loan Portfolio Analysis 8](#_Toc201606727)

[2.3.3 Loan Approval Analysis by Gender 8](#_Toc201606728)

[2.3.4 Loan Amount Distribution Analysis 9](#_Toc201606729)

[2.3.5 Self-Employment Approval Pattern 9](#_Toc201606730)

[2.3.6 Applicant Income Distribution 10](#_Toc201606731)

[2.3.7 Top Ten Loan Applications Analysis 11](#_Toc201606732)

[2.3.8 Property Area Distribution Analysis 12](#_Toc201606733)

[3 Recommendations for Future Predictive Model Development 12](#_Toc201606734)

[4 Conclusion 13](#_Toc201606735)

[5 References 14](#_Toc201606736)

[6 Appendices 15](#_Toc201606737)

[6.1 Appendix 1: Guthub Link 15](#_Toc201606738)

[6.2 Appendix 1: Pseudocode 15](#_Toc201606739)

[6.3 Appendix 2: Code Used 16](#_Toc201606740)

[6.4 Appendix 3: Code Testing in Excel 25](#_Toc201606741)

# Introduction

This report presents a comprehensive data-driven solution to the operational challenge that SUCCESS Financial Services (SFS) seek to respond to with regards to the exponentially increasing volumes of loan application. The purpose of the report is to outline the development and implementation of partial automation processes that will enhance operational efficiency while maintaining robust control environments. The report is structured in a way that initially it will focus on studying the existing business situation and then followed by analysis of the proposed programming solution, implementation and the continuous maintenance needs.

## Current Business Environment

Small business lending institutions like SFS operate in an increasingly competitive digital landscape where customer expectations for rapid service delivery continue to escalate (Haron et al., 2013). The increases in loans requests at SFS by 200% present the general trends related to the demand of more businesses to be able to access available financing opportunities (Mills and McCarthy, 2016). However, such an increase has revealed severe operational bottlenecks of conventional manual processing systems. Firms just like SFS are under increased pressure to strike the balance growth ambitions with regulatory compliance requirements, risk management protocols as well as operational efficiency demands (Haron et al., 2013).

## Problems Addressed by the Programming Solution

Existing manual loan application process has resulted in a high level of operational stress that is translated to shortages in skills, long approval periods, and high risk in operations (Nowak et al., 2018). Human scalability is restrained by the dependence on manual loan decision categorisation, with the aspect of human error being probable as well (Mills and McCarthy, 2016). Unless there is any intervention to these constraints, they will keep blocking growth, and at the same time, they may affect the quality of service.

## Impact and Benefits of Proposed Development

The main benefits are the shortened processing processes, the increased uniformity in the decision-making process, the optimized resource allocations that allow maximizing contact with the customers and operational controls enhanced (Hassan et al., 2024). The solution will open the capacity of loan teams to carry out tasks that add value including personalised customer interaction and multi-faceted case handling (Haron et al., 2013). The automated process will produce important insights about data that can be used to determine future business strategy and risk management choices in the business.

## Implications of Non-Proceeding

Failure of implementation of this solution would perpetuate existing operational constraints, which may culminate to declining service levels to customers, high levels of operational risks, and missed growth opportunities (Mills and McCarthy, 2016). Ineffective operation scaling may precondition a competitive drawback and regulatory issues over operation resilience.

## User Requirements and Tools

The requirements of the user have been taken into consideration on a wide variety of stakeholders such as marketing, internal audit, compliance, sales, customer service and support staff (Haron et al., 2013). The solution will utilise accessible programming tools and interfaces that accommodate both groups of users in terms of their technical competence ensuring broad organisational adoption and utility.

## Data Supplied

The data provided consists of two files, a PDF extract of business loan records which include historic results of loan applicants being approved as well as an excel file held by the sales team.

## Development and Maintenance Challenges

Some of the challenges presented to the team are how to deal with the programmers that hold unequal experience, how to make the code to be accessible to the diverse groups of users, and how to assure the quality of the data (Purificato et al., 2023). Implementing reusable code architecture will bring about a long-term advantage because it takes shorter period to develop, easier maintainable and consistency will be improved across the application (Purificato et al., 2023).

## Regulatory and Ethical Implications

The solution should be zealous of regulatory aspects around data privacy and algorithmic fairness, along with discrimination in lending (Challoumis, 2024). Ethical consideration involves adoption of transparent decision-making process and maintaining human oversight to avoid biasness of the algorithms, since the information on which demographic data is being gathered is sensitive (Challoumis, 2024).

# Approach

This project follows the PPDAC (Problem, Plan, Data, Analysis, Conclusion) framework, which offers a well-organized path towards delivering the solution of automating the loan process. The methodology includes five main steps, including problem identification concerning the SFS operational challenges and stakeholder needs within marketing, compliance, and operations departments, planning, which is related to the identification of the data analysis method and tools to use to conduct an exploratory data analysis, data phase where the quality of the data given in PDF and Excel files is examined, analysis, which is associated with exploratory data analysis, including the identification of patterns, trends, and insights in the loan application data and conclusion, which reflects the findings in actionable recommendations (Fielding-Wells, 2010).

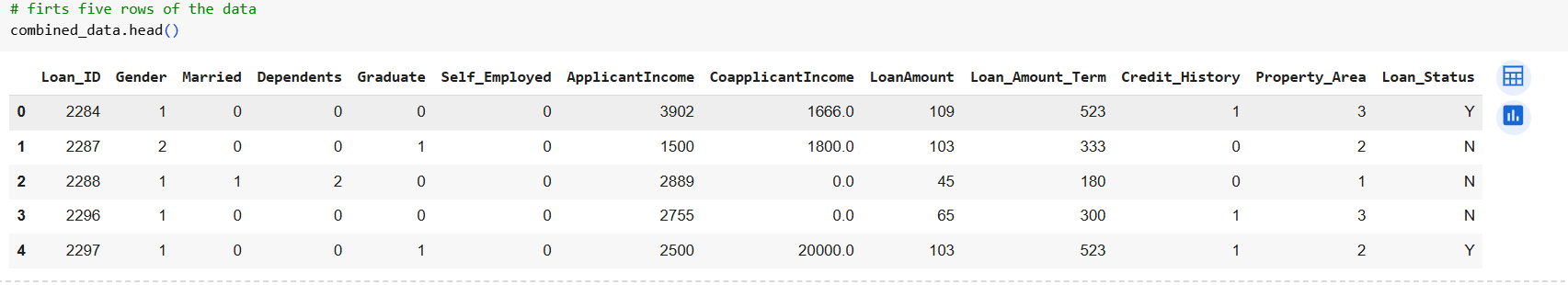
## Technical Implementation Framework

The programming solution is developed in Python based on its wide variety of data analysis tools, such as pandas to perform data operations and cleaning, matplotlib and seaborn to plot graphs, and numpy to solve numerical problems (Saabith et al., 2021). PyPDF2 or pdfplumber libraries are used to extract PDF and openpyxl used to process excel files (Sorvisto, 2023). The libraries are essential for handling data in a variety of formats, exploratory data analysis, and to produce required visualizations of the data that stakeholders could easily understand despite having different levels of expertise on data analytics (Purificato et al., 2023). Jupiter notebook in Google Colab is chosen as the platform for code development. The choice of the platform guarantees compatibility with other operating systems and the ability to integrate with existing business tools (Saabith et al., 2021).

The design is based on modular, code-agnostic architecture built upon structured pseudocode whose data processing workflows and validation rules and output generation processes are well-defined. This can be used later to implement with other programming languages with continued logical consistency and easier maintenance, and improvement by different technical departments.

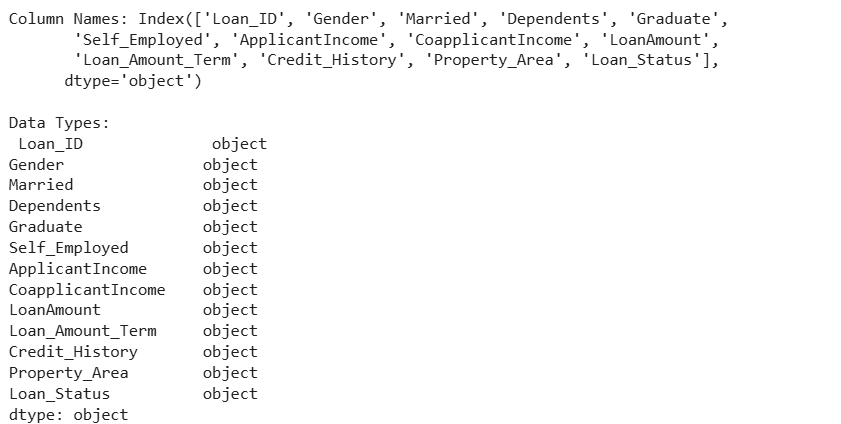
## Loading and Preparing the Data for Analysis

The loading process of the data commenced with the import of the required python libraries that included the pandas library to manipulate the data and the pdfplumber for PDF extraction capabilities (Sorvisto, 2023). Two separate datasets were loaded. These datasets were an excel sheet that the sales team was using and a pdf database extract that included historical loan records with the results of approvals. The data loading into the excel was easy with the use of the read\_excel in pandas that effectively loaded the structured excel spreadsheets data (Bercowsky Rama, 2022). The PDF extraction however needed a more advanced method because it was in a multi-page tabular form (Bercowsky Rama, 2022). The solution was to read the first page, extract the top row to determine column structure, and then read every other page, each one of which provided a row. Quality data was captured systematically and structural integrity across several pages of PDFs were maintained.

******

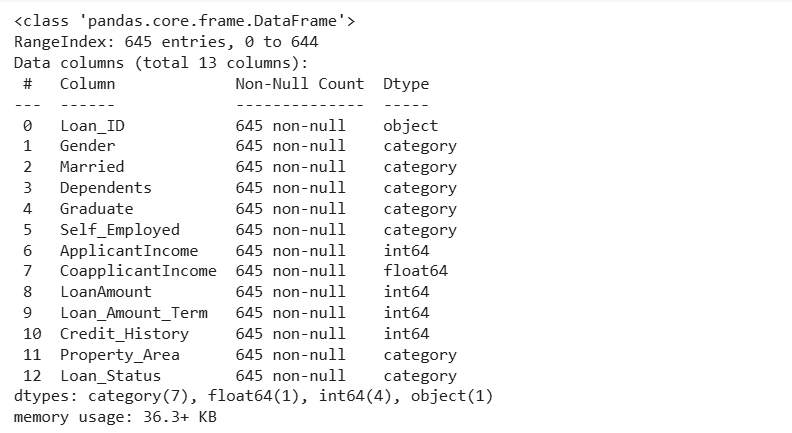
***Figure: first five rows of the data***

A critical challenge emerged during initial inspection which was that all the columns were in the object data type which means that they were treated as text strings when they should be in their actual data format (Zervou, 2024).

******

***Figure: columns and their data type***

To solve this problem systematically, the conversion of data types was introduced. Numeric columns, such as ApplicantIncome, CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, and Credit\_History were converted using pd.to\_numeric() along with error handling that takes care of an invalid entry. Gender, Married, Dependents, Graduate, Self\_Employed, Property\_Area, and Loan\_Status were coded as categorical variables so that they achieve better memory and analysis capacity (Zervou, 2024).

******

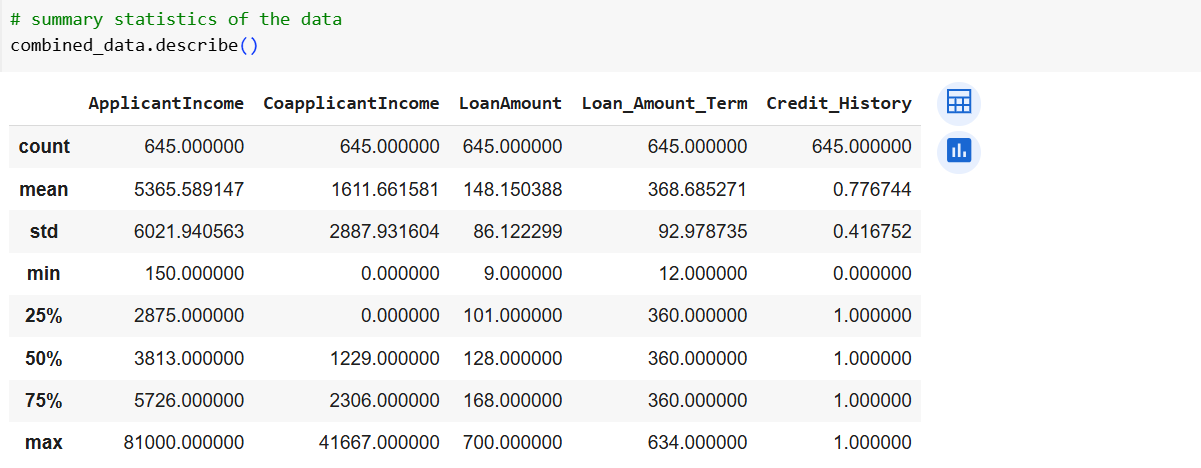
***Figure: final data with the correct data types***

The final data had no missing values and thus was ready for the analysis in the following phase of exploratory data analysis and visualization.

## Exploratory Data Analysis

### Summary Statistics of the Data

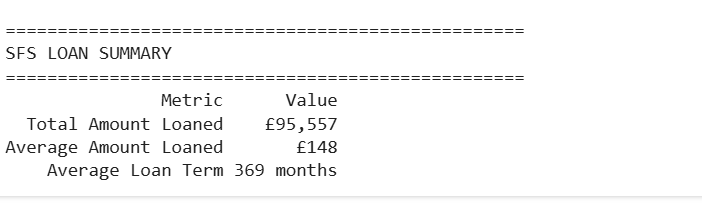
The summary statistics reveal significant income variability among applicants with mean applicant income of £5,366 and there is a wide standard deviation in applicant income (£9622) which means varied economic background. Credit history reveals applicants with positive credit report to stand at 77.7%. Earnings vary between £150 to £81,000 which proves that SFS offers services to small customers and larger corporations.

******

***Figure: summary statistics of the data***

### Loan Portfolio Analysis

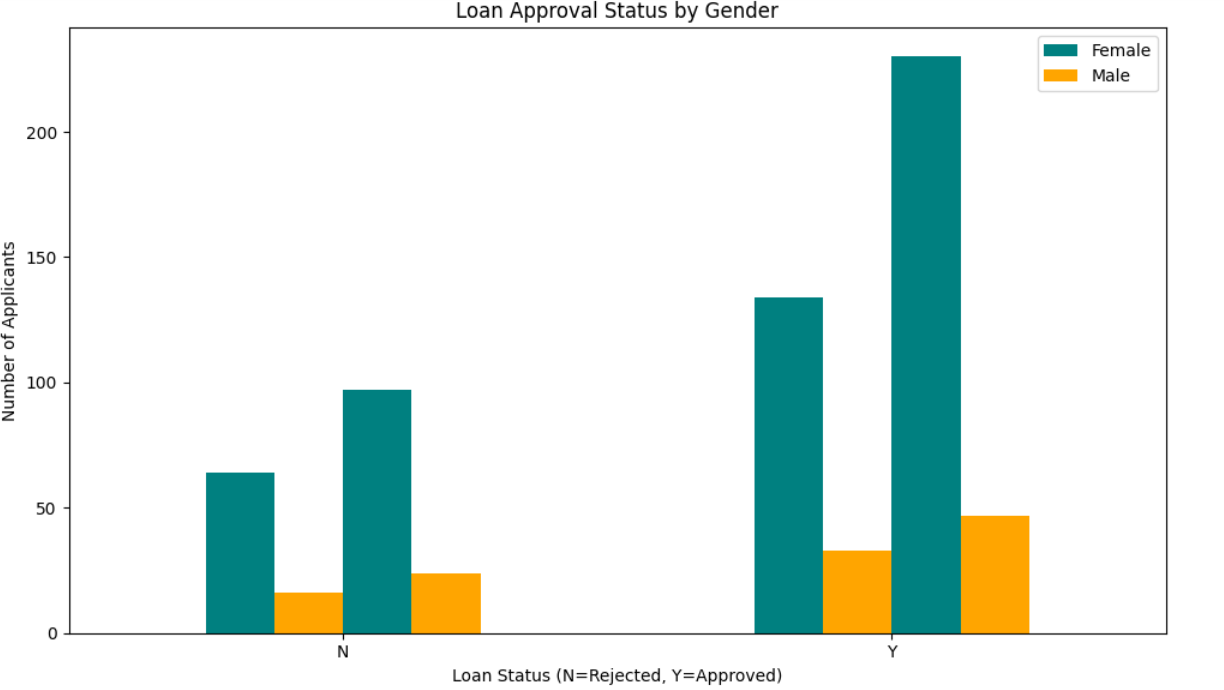
SFS loan summary provides some important information about the lending activities and positioning of the company. The portfolio size of the total amount loaned at £95,557 is fairly modest indicating that SFS lent to smaller businesses rather than to large financing corporations (Arora et al., 2022). However, the mean value of loan size is £148 which is very small suggesting that SFS is specialized in giving out loans to small businesses.

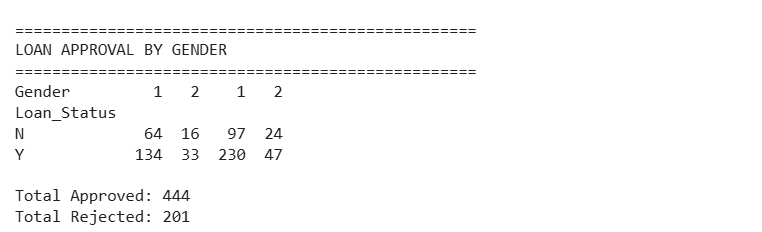
******

***Figure: SFS loan summary analysis***

Average loan length of 369 months (about 30.7 years) is very long as far as ordinary business lending is concerned, and thus may mean SFS specialises in asset-based lending or real estate related business lending. This long repayment period implies that the firm is a provider to the businesses that need to have large capital investments in which the guarantee of repayment is long (Agarwal et al., 2024). The loan portfolio characteristics demonstrate SFS's strategic positioning in serving small to medium enterprises with patient capital requirements.

### Loan Approval Analysis by Gender

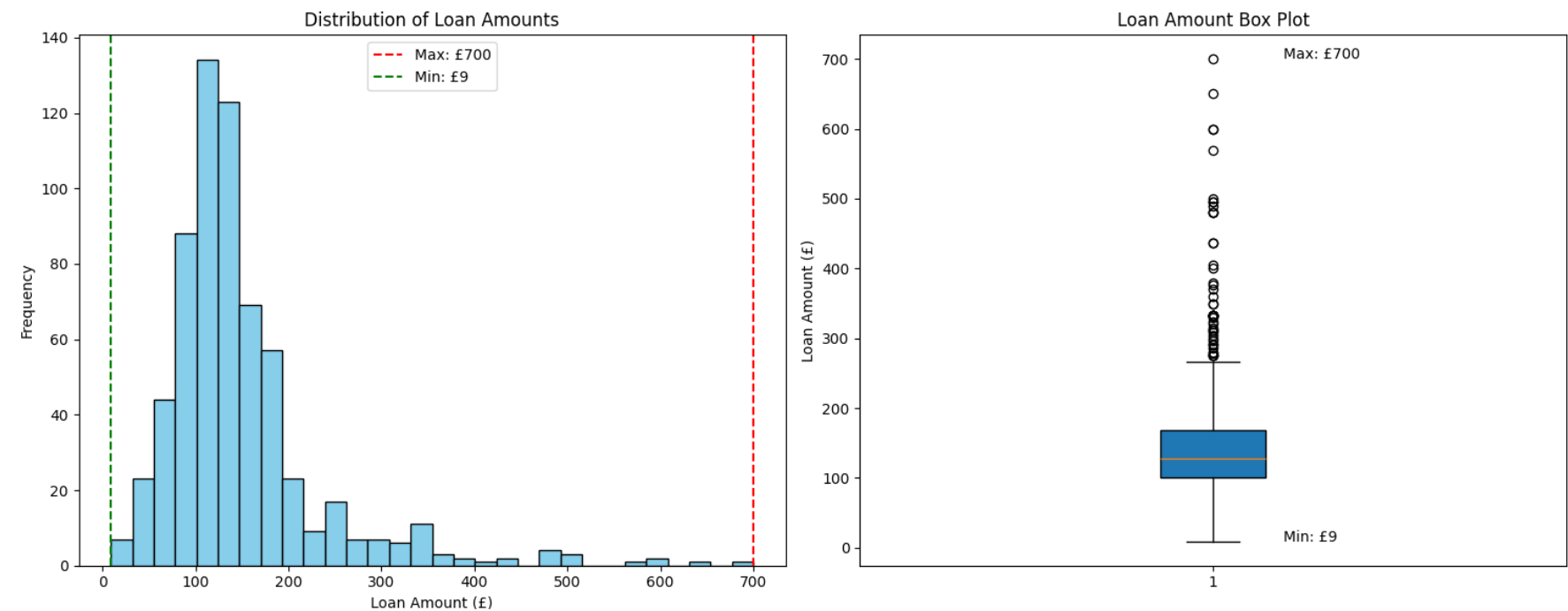
******

******

***Figure: loan approval analysis by gender***

This analysis confirms that there exist vast gender differences on applications and approval rates of SFS loans. The number of male applicants (Gender 1) is larger (364 applications) in comparison with 80 females (Gender 2) applications. Male applicants experience 63.2% approval (230 approved, 134 rejected) and females demonstrate more successful approval of 66.2% (47 approved, 33 rejected). On the whole, 444 applications (68.8%) were approved and 201 (31.2%) were rejected in SFS. This trend indicates possible differences in the quality of applications, riskiness, or business attributes of women and men applicants (Agarwal et al., 2024).

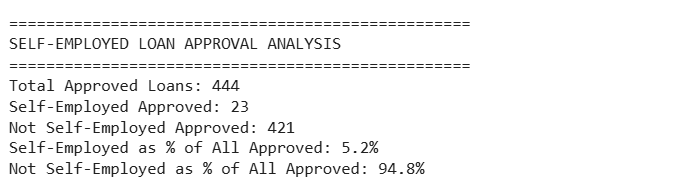
### Loan Amount Distribution Analysis

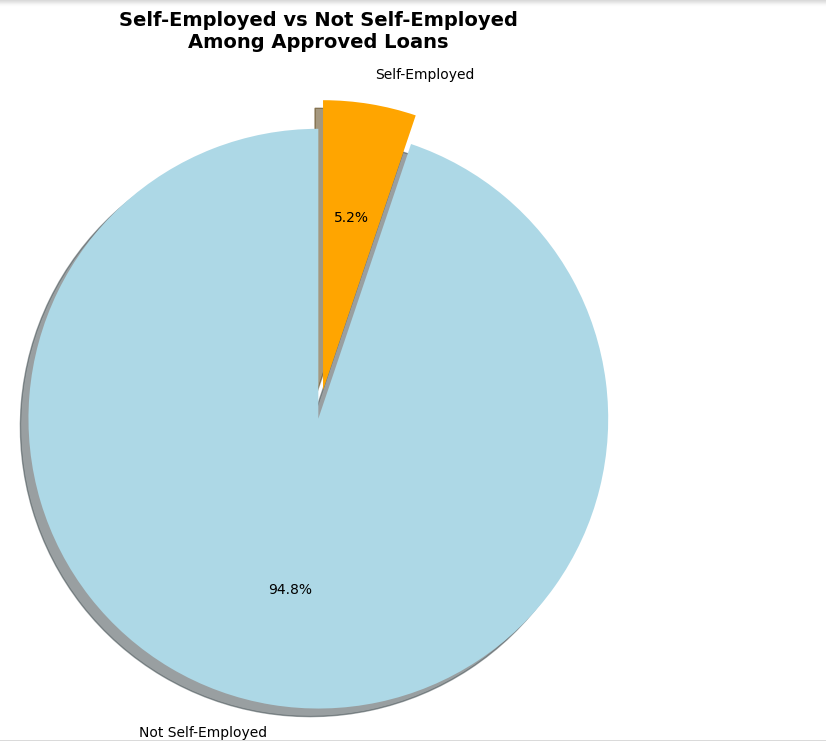
******

***Figure: distribution of loan amount***

SFS depicts excellent flexibility in lending that enables loans between £9 and £700 which leaves a gap of 691 pound in a way that facilitates financing of various businesses. The low threshold of £9 represent possibilities of micro-lending to the smallest business or to the needs of a particular business and the high limit of £700 indicates an orientation toward financing to small businesses as opposed to large corporate lending (Thandapani and Karthika, 2025).

### Self-Employment Approval Pattern

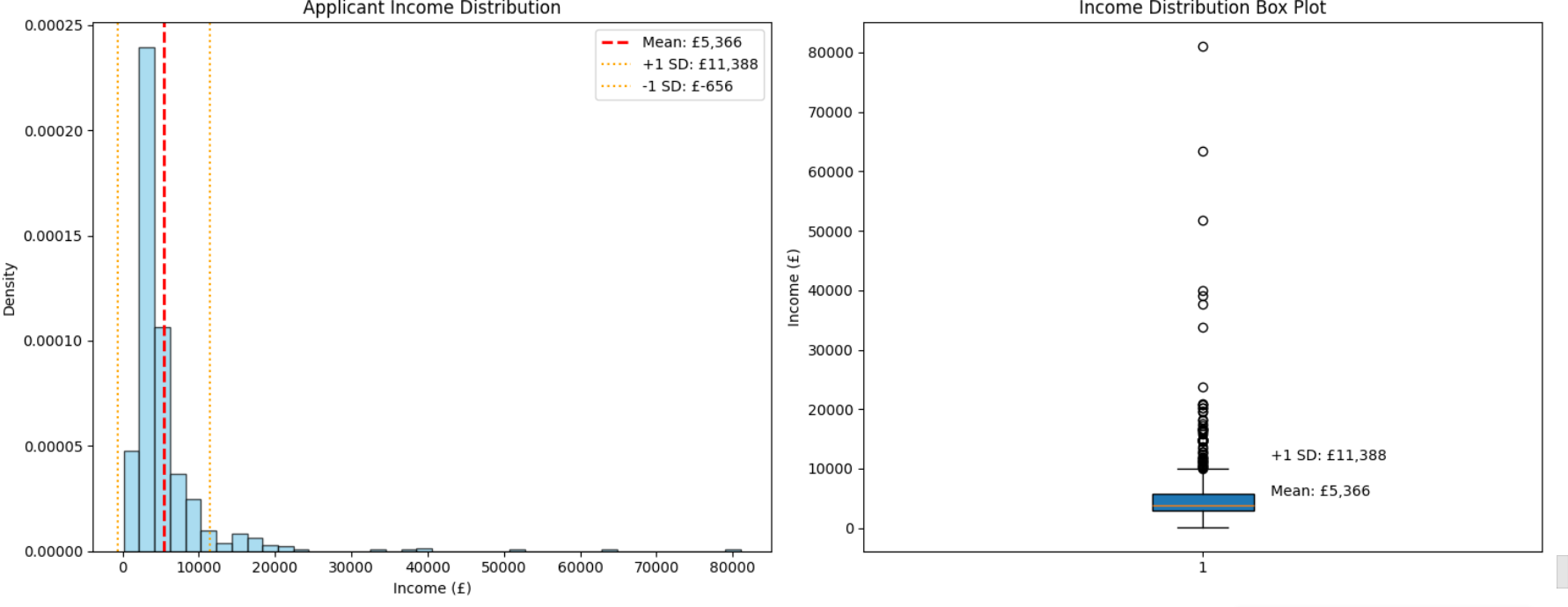


******

***Figure: self employed approval analysis***

Self-employed applicants represent a minimal 5.2% (23 individuals) of SFS's 444 approved loans, while not self-employed dominate at 94.8% (421 approvals). This huge imbalance indicates that there is not much number of self-employed applicants or that the applicants in this group may be facing approval problems. The approval trend at SFS is that salaried applicants whose earnings are stable and verifiable have an advantage over self-employed who maybe regarded as risky entrepreneurships that are unpredictable in terms of earnings (Thandapani and Karthika, 2025).

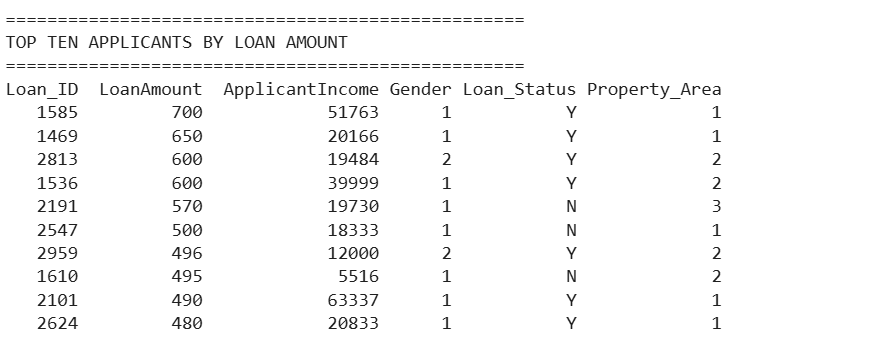
### Applicant Income Distribution

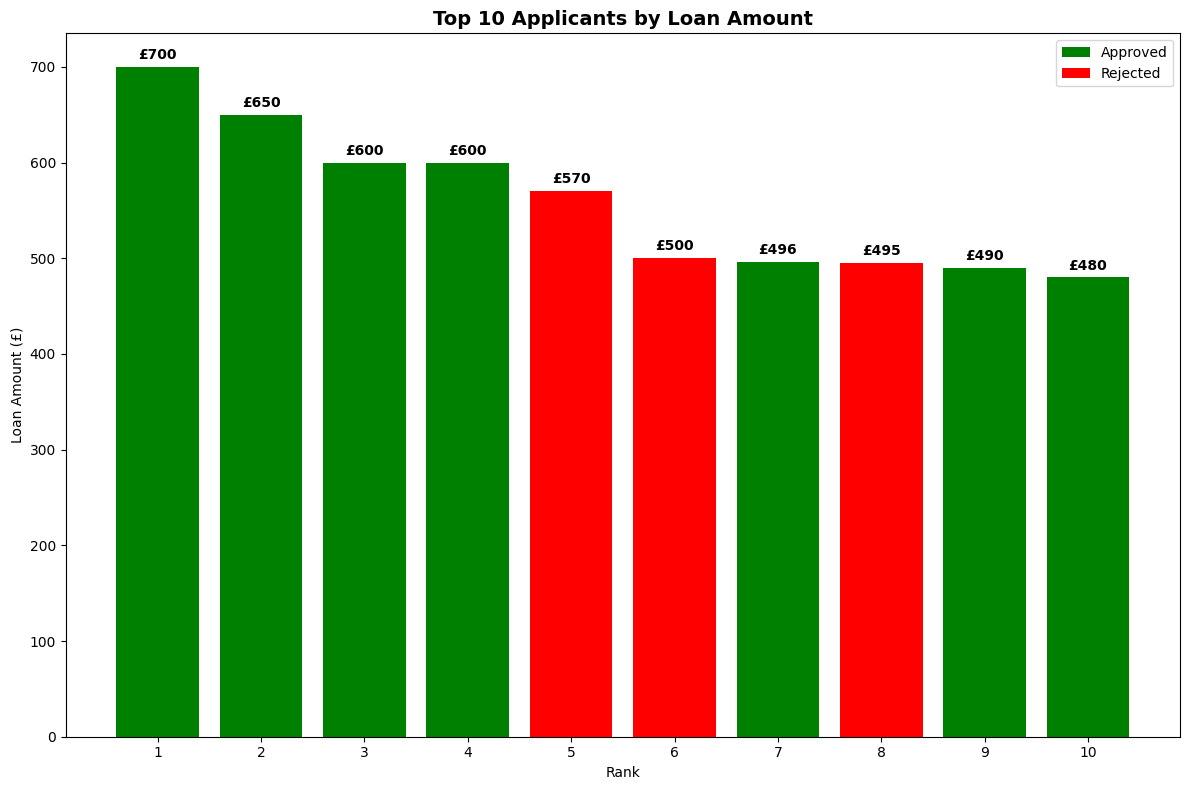
******

***Figure: applicant income distribution***

The average income level of SFS applicants is high at £5,366 and the standard deviation of £6,022, shows a wide range of the applicants in terms of economic backgrounds. The median of the incomes, £3 813 falls below the average, suggesting positive skewness with higher-income (Arora et al., 2022). There is widespread income disparity of between £150 to £81000.

### Top Ten Loan Applications Analysis

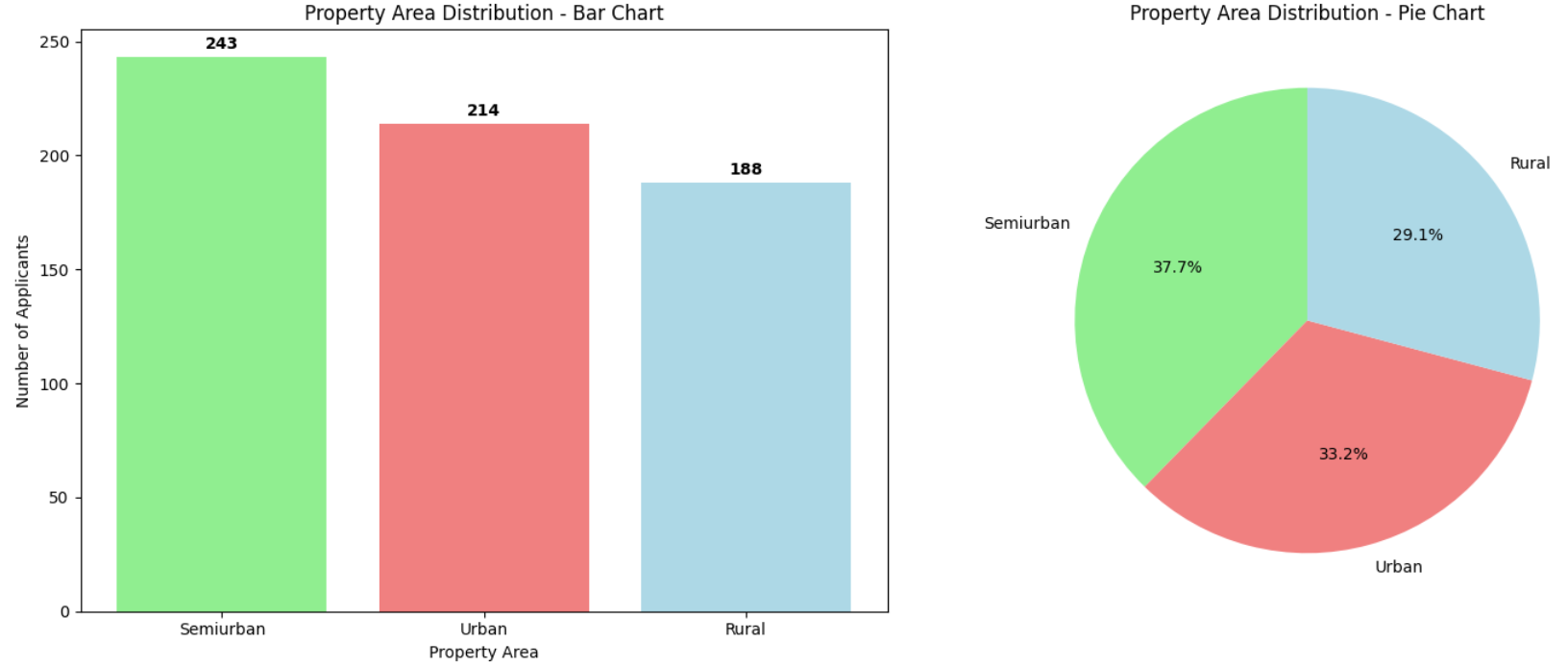


******

***Figure: top ten loan applications analysis***

The largest values of loans are within the range of 480 and 700 with six loans being approved out of the top ten being rejected. The applicant income is not directly linked with the loan amount in that the highest amount (700) of loan was given to the highest income (51,763), and rejections were given to some high-income cases. This approval pattern indicates that there are additional factors to income that play into approving lending such as area of property and perhaps credit history in making such premium loan applications (Thandapani and Karthika, 2025).

### Property Area Distribution Analysis

******

***Figure: property area distribution analysis***

SFS shows quite an equal distribution of geographical areas of property, with semiurban locations in the lead with 243 applicants (37.7 percent), urban areas with 214 applicants (33.2 percent), and rural locations with 188 applicants (29.1 percent). Such a distribution denotes strategic targeting of the market in various geographical areas, with some preference to a semiurban market (Agarwal et al., 2024).

# Recommendations for Future Predictive Model Development

**Model Development Approach:** a supervised machine learning model could predict the loan results with the help of scikit-learn. Random forest or gradient boosting algorithms would handle mixed data types effectively, with cross-validation ensuring robust performance (Uddin et al., 2023). The major libraries are pandas to preprocess data, scikit-learn for ML models, and matplotlib/seaborn to interpret a model (Haque and Hassan, 2024). Measures such as precision, recall, F1-score, and AUC-ROC will be used to evaluate the model.

**Data Quality Improvements**: Automate data validation pipelines using Great Expectations library to identify anomalies and missing values. Data inconsistency between PDF and Excel resources should be tackled regularly with the aid of data audit (Haque and Hassan, 2024).

**Enhanced Model Accuracy:** More sophisticated libraries such as XGBoost, LightGBM which use ensembles of models might help gain higher scores in prediction (Uddin et al., 2023). They would optimize performance with feature engineering technique and automated hyperparameter tuning with Optuna (Haque and Hassan, 2024).

**Staff Training Requirements:** Loan officers should be trained about how to interpret models, reading probability scores, and how to have human oversight. Machine learning basics and model monitoring functions are necessary to technical staff members (Uddin et al., 2023).

**Regulatory Compliance**: Implement explainable AI techniques using SHAP or LIME for transparency. Implement bias detecting mechanisms that track demographic fairness (Challoumis, 2024). Frequent auditing of the models, documentation and well-defined escalation process are a few cursory processes to ensure ethical lending is done whilst still upholding the regulatory act of fair lending laws (Agu et al., 2024).

# Conclusion

This report successfully demonstrated the formulation of an end-to-end data analytics solution to the problem of the loan application handling faced by SUCCESS Financial Services. The evaluation provided some important insights on lending portfolio of SFS, such as strong fluctuations of income levels of applicants (£150-81,000), sparse presence of self-employed borrowers (5.2%), and geographic clustering in semiurban locations. Technical implementation successfully combined different data sets based on Python pandas and visualization libraries, building up the base of automatization of the operational processes.

The exploratory data analysis showed that there is a severe problem with the quality of the data that needs to be addressed, especially in relation to inconsistent data type and possible variability in the amount of loan. The recommended transition to the predictive modelling approach based on machine learning frameworks would observably boost the efficiency of SFS processes without infringing the regulations. Nevertheless, good data governance, training of the staff, and effective ethical protection are the keys to success.

# References

Agarwal, A., Prabha, S. and Yadav, R., 2024. Exploratory Data Analysis for Banking and Finance: Unveiling Insights and Patterns. *arXiv preprint arXiv:2407.11976*.

Agu, E.E., Abhulimen, A.O., Obiki-Osafiele, A.N., Osundare, O.S., Adeniran, I.A. and Efunniyi, C.P., 2024. Discussing ethical considerations and solutions for ensuring fairness in AI-driven financial services. *International Journal of Frontier Research in Science*, *3*(2), pp.001-009.

Arora, P., Gautam, S., Kalra, A., Negi, A. and Tyagi, N., 2022. Credit Risk Analysis Using EDA. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 2* (pp. 563-574). Singapore: Springer Nature Singapore.

Bercowsky Rama, A., 2022. Python: Data handling, analysis and plotting. In *Bioimage Data Analysis Workflows‒Advanced Components and Methods* (pp. 29-57). Cham: Springer International Publishing.

Challoumis, C., 2024, November. What are the ethical implications of AI in FINANCIAL systems. In *XVII International Scientific Conference* (pp. 41-75).

Fielding-Wells, J., 2010, July. Linking problems, conclusions and evidence: Primary students’ early experiences of planning statistical investigations. In *Proceedings of the 8th international conference on teaching statistics. Voorburg, The Netherlands: International Statistical Institute*.

Haque, F.M. and Hassan, M.M., 2024. Bank Loan Prediction Using Machine Learning Techniques. *arXiv preprint arXiv:2410.08886*.

Haron, H., Said, S.B., Jayaraman, K. and Ismail, I., 2013. Factors influencing small medium enterprises (SMES) in obtaining loan. *International Journal of Business and Social Science*, *4*(15).

Hassan, M.M., Abrar, M.F., Kakon, S., Hossen, A. and Arafat, Y., 2024, December. Improving Loan Approval Decisions: The Impact of Data Balancing on the Classifier’s Performance in Predicting Borrower Reliability. In *2024 27th International Conference on Computer and Information Technology (ICCIT)* (pp. 1690-1695). IEEE.

Mills, K. and McCarthy, B., 2016. The state of small business lending: Innovation and technology and the implications for regulation. *Harvard Business School Entrepreneurial Management Working Paper*, (17-042), pp.17-042.

Nowak, A., Ross, A. and Yencha, C., 2018. Small business borrowing and peer‐to‐peer lending: Evidence from lending club. *Contemporary Economic Policy*, *36*(2), pp.318-336.

Purificato, E., Lorenzo, F., Fallucchi, F. and De Luca, E.W., 2023. The use of responsible artificial intelligence techniques in the context of loan approval processes. *International Journal of Human–Computer Interaction*, *39*(7), pp.1543-1562.

Saabith, S., Vinothraj, T. and Fareez, M., 2021. A review on Python libraries and Ides for Data Science. *Int. J. Res. Eng. Sci*, *9*(11), pp.36-53.

Sorvisto, D., 2023. Tools for Data Science Developers. In *MLOps Lifecycle Toolkit: A Software Engineering Roadmap for Designing, Deploying, and Scaling Stochastic Systems* (pp. 67-102). Berkeley, CA: Apress.

Thandapani, G. and Karthika, S., 2025. A data-driven approach to loan approvals prediction and profitability: Leveraging exploratory data analysis for improved decision-making. In *Advances in Electrical and Computer Technologies* (pp. 261-266). CRC Press.

Uddin, N., Ahamed, M.K.U., Uddin, M.A., Islam, M.M., Talukder, M.A. and Aryal, S., 2023. An ensemble machine learning based bank loan approval predictions system with a smart application. *International Journal of Cognitive Computing in Engineering*, *4*, pp.327-339.

Zervou, M., 2024. *Python Data Cleaning and Preparation Best Practices: A practical guide to organizing and handling data from various sources and formats using Python*. Packt Publishing Ltd.

# Appendices

## Appendix 1: Guthub Link

## Appendix 1: Pseudocode

BEGIN SFS\_Data\_Analysis

// Import Required Libraries

IMPORT pandas, pdfplumber, matplotlib, numpy

// Load Data Sources

SET excel\_file = "SUCCESS Loan Data (1).xlsx"

SET pdf\_file = "SUCCESS\_Loans\_Database\_Table (1).pdf"

LOAD excel\_data FROM excel\_file

OPEN pdf\_file

GET header FROM first\_page.table[0]

FOR each page IN pdf\_file

IF page = first\_page THEN

ADD table\_data[1:] TO all\_data (skip header)

ELSE

ADD table\_data TO all\_data

END IF

END FOR

CREATE pdf\_data FROM all\_data WITH header

CLOSE pdf\_file

// Combine Datasets

COMBINE excel\_data AND pdf\_data INTO combined\_data

// Data Type Conversion

SET numeric\_columns = [ApplicantIncome, CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, Credit\_History]

SET categorical\_columns = [Gender, Married, Dependents, Graduate, Self\_Employed, Property\_Area, Loan\_Status]

FOR each column IN numeric\_columns

CONVERT column TO numeric\_type

END FOR

FOR each column IN categorical\_columns

CONVERT column TO category\_type

END FOR

// Basic Data Inspection

DISPLAY dataset\_shape

DISPLAY column\_names

DISPLAY data\_types

DISPLAY first\_5\_rows

DISPLAY missing\_values\_count

DISPLAY basic\_statistics

// Loan Summary Analysis

CALCULATE total\_loaned = SUM(LoanAmount)

CALCULATE average\_loaned = MEAN(LoanAmount)

CALCULATE average\_term = MEAN(Loan\_Amount\_Term)

DISPLAY loan\_summary\_table

// Gender-Based Approval Analysis

CREATE cross\_table OF Loan\_Status BY Gender

DISPLAY approval\_counts\_by\_gender

CREATE bar\_chart OF approval\_status BY gender

// Loan Amount Range Analysis

CALCULATE max\_loan = MAX(LoanAmount)

CALCULATE min\_loan = MIN(LoanAmount)

DISPLAY loan\_range\_statistics

CREATE histogram AND boxplot OF loan\_amounts

// Self-Employment Analysis

FILTER approved\_loans WHERE Loan\_Status = 'Y'

COUNT self\_employed\_approved WHERE Self\_Employed = 1

CALCULATE percentage = (self\_employed\_approved / total\_approved) \* 100

DISPLAY self\_employment\_statistics

CREATE pie\_chart OF employment\_status IN approved\_loans

// Income Distribution Analysis

CALCULATE income\_statistics (mean, std\_dev, median, min, max)

DISPLAY income\_distribution\_summary

CREATE histogram AND boxplot OF applicant\_income

// Top Applicants Analysis

GET top\_10\_applicants BY LoanAmount (descending)

DISPLAY top\_10\_table

CREATE bar\_chart OF top\_10\_loan\_amounts WITH approval\_status\_colors

// Property Area Distribution

MAP property\_codes TO area\_names (1=Urban, 2=Semiurban, 3=Rural)

COUNT applicants BY property\_area

CALCULATE percentages BY property\_area

DISPLAY property\_distribution\_summary

CREATE bar\_chart AND pie\_chart OF property\_distribution

END SFS\_Data\_Analysis

## Appendix 2: Code Used

pip install pandas pdfplumber

import pandas as pd

import pdfplumber

# Load Excel data

excel\_data = pd.read\_excel("SUCCESS Loan Data (1).xlsx")

# Load PDF data

with pdfplumber.open("SUCCESS\_Loans\_Database\_Table (1).pdf") as pdf:

header = pdf.pages[0].extract\_table()[0] # Get header from first page

all\_data = []

for page in pdf.pages:

table = page.extract\_table()

if page == pdf.pages[0]:

all\_data.extend(table[1:]) # Skip header on first page

else:

all\_data.extend(table) # Add all rows from other pages

pdf\_data = pd.DataFrame(all\_data, columns=header)

# Combine datasets

combined\_data = pd.concat([excel\_data, pdf\_data], ignore\_index=True)

# firts five rows of the data

combined\_data.head()

# Basic data inspection

print("Dataset Shape:", combined\_data.shape)

print("\nColumn Names:", combined\_data.columns)

print("\nData Types:\n", combined\_data.dtypes)

print("\nFirst 5 rows:\n", combined\_data.head())

print("\nMissing Values:\n", combined\_data.isnull().sum())

# Combine datasets

combined\_data = pd.concat([excel\_data, pdf\_data], ignore\_index=True)

# Convert data types

# Numeric columns

numeric\_cols = ['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan\_Amount\_Term', 'Credit\_History']

for col in numeric\_cols:

combined\_data[col] = pd.to\_numeric(combined\_data[col], errors='coerce')

# Categorical columns

categorical\_cols = ['Gender', 'Married', 'Dependents', 'Graduate', 'Self\_Employed', 'Property\_Area', 'Loan\_Status']

for col in categorical\_cols:

combined\_data[col] = combined\_data[col].astype('category')

# final information about the datasets including data types

combined\_data.info()

"""# \*\*Exploratory Data Analysis\*\*

## \*\*Summary Statistics of the Data\*\*

"""

# summary statistics of the data

combined\_data.describe()

"""## \*\*Loan Portfolio Analysis\*\*"""

# SFS Loan Summary Table

print("\n" + "="\*50)

print("SFS LOAN SUMMARY")

print("="\*50)

total\_loaned = combined\_data['LoanAmount'].sum()

avg\_loaned = combined\_data['LoanAmount'].mean()

avg\_term = combined\_data['Loan\_Amount\_Term'].mean()

summary\_data = {

'Metric': ['Total Amount Loaned', 'Average Amount Loaned', 'Average Loan Term'],

'Value': [f'£{total\_loaned:,.0f}', f'£{avg\_loaned:,.0f}', f'{avg\_term:.0f} months']

}

summary\_df = pd.DataFrame(summary\_data)

print(summary\_df.to\_string(index=False))

"""## \*\*Loan Approval Analysis by Gender\*\*"""

# Loan Approval Analysis by Gender

import matplotlib.pyplot as plt

# Create cross-tabulation

approval\_gender = pd.crosstab(combined\_data['Loan\_Status'], combined\_data['Gender'])

print("\n" + "="\*50)

print("LOAN APPROVAL BY GENDER")

print("="\*50)

print(approval\_gender)

# Calculate totals

total\_approved = approval\_gender.loc['Y'].sum()

total\_rejected = approval\_gender.loc['N'].sum()

print(f"\nTotal Approved: {total\_approved}")

print(f"Total Rejected: {total\_rejected}")

# Create chart

fig, ax = plt.subplots(figsize=(10, 6))

# Change the bar colors here

approval\_gender.plot(kind='bar', ax=ax, color=['teal', 'orange'])

plt.title('Loan Approval Status by Gender')

plt.xlabel('Loan Status (N=Rejected, Y=Approved)')

plt.ylabel('Number of Applicants')

plt.legend(['Female', 'Male'])

plt.xticks(rotation=0)

plt.tight\_layout()

plt.show()

"""## \*\*Loan Amount Distribution Analysis\*\*"""

# Loan Amount Range Analysis

print("\n" + "="\*50)

print("LOAN AMOUNT RANGE ANALYSIS")

print("="\*50)

max\_loan = combined\_data['LoanAmount'].max()

min\_loan = combined\_data['LoanAmount'].min()

print(f"Maximum Loan Amount: £{max\_loan:,.0f}")

print(f"Minimum Loan Amount: £{min\_loan:,.0f}")

print(f"Range: £{max\_loan - min\_loan:,.0f}")

# Create chart for loan amount distribution

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

# Histogram

ax1.hist(combined\_data['LoanAmount'].dropna(), bins=30, color='skyblue', edgecolor='black')

ax1.axvline(max\_loan, color='red', linestyle='--', label=f'Max: £{max\_loan:,.0f}')

ax1.axvline(min\_loan, color='green', linestyle='--', label=f'Min: £{min\_loan:,.0f}')

ax1.set\_title('Distribution of Loan Amounts')

ax1.set\_xlabel('Loan Amount (£)')

ax1.set\_ylabel('Frequency')

ax1.legend()

# Box plot

ax2.boxplot(combined\_data['LoanAmount'].dropna(), patch\_artist=True)

ax2.set\_title('Loan Amount Box Plot')

ax2.set\_ylabel('Loan Amount (£)')

ax2.text(1.1, max\_loan, f'Max: £{max\_loan:,.0f}', ha='left')

ax2.text(1.1, min\_loan, f'Min: £{min\_loan:,.0f}', ha='left')

plt.tight\_layout()

plt.show()

"""## \*\*Self-Employment Approval Pattern\*\*"""

# Self-Employed Approval Analysis

print("\n" + "="\*50)

print("SELF-EMPLOYED LOAN APPROVAL ANALYSIS")

print("="\*50)

# Filter approved loans

approved\_loans = combined\_data[combined\_data['Loan\_Status'] == 'Y']

total\_approved = len(approved\_loans)

# Count self-employed in approved loans

self\_employed\_approved = len(approved\_loans[approved\_loans['Self\_Employed'] == 1])

not\_self\_employed\_approved = total\_approved - self\_employed\_approved

# Calculate percentages

self\_employed\_pct = (self\_employed\_approved / total\_approved) \* 100

not\_self\_employed\_pct = (not\_self\_employed\_approved / total\_approved) \* 100

print(f"Total Approved Loans: {total\_approved}")

print(f"Self-Employed Approved: {self\_employed\_approved}")

print(f"Not Self-Employed Approved: {not\_self\_employed\_approved}")

print(f"Self-Employed as % of All Approved: {self\_employed\_pct:.1f}%")

print(f"Not Self-Employed as % of All Approved: {not\_self\_employed\_pct:.1f}%")

# Create pie chart

fig, ax = plt.subplots(figsize=(10, 8))

labels = ['Not Self-Employed', 'Self-Employed']

sizes = [not\_self\_employed\_pct, self\_employed\_pct]

colors = ['lightblue', 'orange']

explode = (0, 0.1) # explode self-employed slice

ax.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%',

shadow=True, startangle=90)

ax.set\_title('Self-Employed vs Not Self-Employed\nAmong Approved Loans', fontsize=14, fontweight='bold')

plt.tight\_layout()

plt.show()

"""## \*\*Applicant Income Distribution\*\*"""

# Applicant Income Distribution Analysis

print("\n" + "="\*50)

print("APPLICANT INCOME DISTRIBUTION ANALYSIS")

print("="\*50)

# Calculate income statistics

mean\_income = combined\_data['ApplicantIncome'].mean()

std\_income = combined\_data['ApplicantIncome'].std()

median\_income = combined\_data['ApplicantIncome'].median()

min\_income = combined\_data['ApplicantIncome'].min()

max\_income = combined\_data['ApplicantIncome'].max()

print(f"Average Income: £{mean\_income:,.0f}")

print(f"Standard Deviation: £{std\_income:,.0f}")

print(f"Median Income: £{median\_income:,.0f}")

print(f"Minimum Income: £{min\_income:,.0f}")

print(f"Maximum Income: £{max\_income:,.0f}")

# Create income distribution chart

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

# Histogram with normal distribution overlay

ax1.hist(combined\_data['ApplicantIncome'].dropna(), bins=40, density=True,

alpha=0.7, color='skyblue', edgecolor='black')

ax1.axvline(mean\_income, color='red', linestyle='--', linewidth=2, label=f'Mean: £{mean\_income:,.0f}')

ax1.axvline(mean\_income + std\_income, color='orange', linestyle=':', label=f'+1 SD: £{mean\_income + std\_income:,.0f}')

ax1.axvline(mean\_income - std\_income, color='orange', linestyle=':', label=f'-1 SD: £{mean\_income - std\_income:,.0f}')

ax1.set\_title('Applicant Income Distribution')

ax1.set\_xlabel('Income (£)')

ax1.set\_ylabel('Density')

ax1.legend()

# Box plot

ax2.boxplot(combined\_data['ApplicantIncome'].dropna(), patch\_artist=True)

ax2.set\_title('Income Distribution Box Plot')

ax2.set\_ylabel('Income (£)')

ax2.text(1.1, mean\_income, f'Mean: £{mean\_income:,.0f}', ha='left')

ax2.text(1.1, mean\_income + std\_income, f'+1 SD: £{mean\_income + std\_income:,.0f}', ha='left')

plt.tight\_layout()

plt.show()

"""## \*\*Top Ten Loan Applications Analysis\*\*"""

# Top Ten Applicants by Loan Amount

print("\n" + "="\*50)

print("TOP TEN APPLICANTS BY LOAN AMOUNT")

print("="\*50)

# Get top 10 applicants by loan amount

top\_10\_loans = combined\_data.nlargest(10, 'LoanAmount')[['Loan\_ID', 'LoanAmount', 'ApplicantIncome',

'Gender', 'Loan\_Status', 'Property\_Area']]

# Display the table

print(top\_10\_loans.to\_string(index=False))

# Create bar chart for top 10 loan amounts

fig, ax = plt.subplots(figsize=(12, 8))

bars = ax.bar(range(1, 11), top\_10\_loans['LoanAmount'],

color=['green' if status == 'Y' else 'red' for status in top\_10\_loans['Loan\_Status']])

ax.set\_title('Top 10 Applicants by Loan Amount', fontsize=14, fontweight='bold')

ax.set\_xlabel('Rank')

ax.set\_ylabel('Loan Amount (£)')

ax.set\_xticks(range(1, 11))

# Add value labels on bars

for i, (bar, amount) in enumerate(zip(bars, top\_10\_loans['LoanAmount'])):

ax.text(bar.get\_x() + bar.get\_width()/2, bar.get\_height() + 5,

f'£{amount:,.0f}', ha='center', va='bottom', fontweight='bold')

# Add legend

from matplotlib.patches import Patch

legend\_elements = [Patch(facecolor='green', label='Approved'),

Patch(facecolor='red', label='Rejected')]

ax.legend(handles=legend\_elements)

plt.tight\_layout()

plt.show()

"""## \*\*Property Area Distribution Analysis\*\*"""

# Define mapping dictionary

property\_map = {1: 'Urban', 2: 'Semiurban', 3: 'Rural', '1': 'Urban', '2': 'Semiurban', '3': 'Rural'}

# Apply mapping using map and fillna to retain values not in map

combined\_data['Property\_Area'] = combined\_data['Property\_Area'].map(property\_map).fillna(combined\_data['Property\_Area'])

# Count and display the distribution

property\_counts = combined\_data['Property\_Area'].value\_counts()

print("Property Area Distribution:")

print(property\_counts)

# Create charts

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

# Bar chart

bar\_colors = ['lightgreen', 'lightcoral', 'lightblue']

bars = ax1.bar(property\_counts.index, property\_counts.values, color=bar\_colors)

ax1.set\_title('Property Area Distribution - Bar Chart')

ax1.set\_xlabel('Property Area')

ax1.set\_ylabel('Number of Applicants')

# Add value labels on bars

for bar, count in zip(bars, property\_counts.values):

ax1.text(bar.get\_x() + bar.get\_width()/2, bar.get\_height() + 2,

str(count), ha='center', va='bottom', fontweight='bold')

# Pie chart

wedges, texts, autotexts = ax2.pie(property\_counts.values,

labels=property\_counts.index,

colors=bar\_colors,

autopct='%1.1f%%',

startangle=90)

ax2.set\_title('Property Area Distribution - Pie Chart')

plt.tight\_layout()

plt.show()

## Appendix 3: Code Testing in Excel

