UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

Usability and User Experience Design

7MMCS006C

Coursework 2025/2026

R.M.M Malsha Piumini

W2106737 | 20240286

Table of Contents

Part A Loan Approval Status Prediction

Task 01 : Domain Understanding: Classification

Variable Name	Retain/drop	Brief justification for retention or dropping
ID	Drop	ID is a unique identifier; this doesn't contain any predictive information. Not only that but also IDs don't add any value to the model (Han, Kamber & Pei, 2011)
Sex	Retain	Gender can impact access to credit and employment stability, affecting approval rates. Including it helps capture demographic variation (Khandani, Kim and Lo, 2010).
Age	Retain	Age is a critical demographic factor influencing loan eligibility and repayment behaviorr. Age has been found to correlate with loan default risk, where older clients tend to show more reliable repayment behaviorr (Khandani, Kim and Lo, 2010).
Education Qualification	Retain	Education level is a strong socio-economic indicator linked to income stability and default probability(Li, Xu & Zhao, 2020).
Income	Retain	Income directly decides repayment capacity and is widely used in credit scoring models.
Home Ownership	Retain	Indicates financial stability; used in FICO scoring (FICO, 2021 Whitepaper)
Employment Length	Retain	Employment length and past defaults are key determinants of credit approval in traditional and statistical credit scoring models (Abdou and Pointon, 2011)
Loan Intent	Retain	The requested loan amount directly affects affordability and approval likelihood (Han et al., 2011).
Loan Amount	Retain	Interest rate reflects lender risk perception and is a key factor in loan approval (Beck, Pamies & Weber, 2019).
Loan Interest Rate	Retain	Interest rate reflects lender risk perception and is a key factor in loan approval (Beck, Pamies & Weber, 2019).
Loan to Income	Retain	The LTI ratio is a critical financial indicator used to assess debt burden and repayment capability (Li et al., 2020).
Payment Default on File	Retain	Previous defaults strongly predict future rejections; this is one of the most important variables in credit risk modelling (Abdou & Pointon, 2011).
Credit History Length	Retain	A longer credit history indicates financial reliability and positively affects approval likelihood (Beck et al., 2019).
Loan Approval Status	Retain (Target)	This is the dependent (target) variable used for classification (Approved/Rejected).

Maximum Loan Amount	Drop	This attribute is used for regression modelling in Part (B) and not relevant to classification.
Credit Application Acceptance	Drop	This feature is redundant, as it strongly overlaps with the target variable and may cause multicollinearity. Removing redundant features helps improve generalization (Han et al., 2011).

Table 1 : Variable retained/ drop table

Task 02 : Data Understanding: Producing Your Experimental Design

1. Basic Statistical Description

The retained dataset consists of 58,645 records and 13 attributes. The table below provides the descriptive for numerical features and frequency counts for categorical features.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
id	58645.0	NaN	NaN	NaN	29322.0	16929.497605	0.0	14661.0	29322.0	43983.0	58644.0
age	58639.0	104.0	22.0	5903.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Sex	221	2	M	126	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Education_Qualifications	58645	6	Unknown	58303	NaN	NaN	NaN	NaN	NaN	NaN	NaN
income	58645.0	NaN	NaN	NaN	64046.172871	37931.106978	4200.0	42000.0	58000.0	75600.0	1900000.0
home_ownership	58645	4	RENT	30594	NaN	NaN	NaN	NaN	NaN	NaN	NaN
emplyment_length	58645.0	NaN	NaN	NaN	4.703487	4.004982	0.0	2.0	4.0	7.0	150.0
loan_intent	58645	6	EDUCATION	12271	NaN	NaN	NaN	NaN	NaN	NaN	NaN
loan_amount	58645.0	NaN	NaN	NaN	9217.556518	5563.807384	500.0	5000.0	8000.0	12000.0	35000.0
loan_interest_rate	58634.0	NaN	NaN	NaN	10.685988	3.161955	-11.14	7.88	10.75	12.99	150.0
loan_income_ratio	58645.0	NaN	NaN	NaN	0.159238	0.091692	0.0	0.09	0.14	0.21	0.83
payment_default_on_file	58640	4	N	49933	NaN	NaN	NaN	NaN	NaN	NaN	NaN
credit_history_length	58645.0	NaN	NaN	NaN	5.813556	4.029196	2.0	3.0	4.0	8.0	30.0
loan_approval_status	58644	8	Approved	50210	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max_allowed_loan	58645.0	NaN	NaN	NaN	282353.842715	51484775.196066	-2426900.0	38003.0	62392.0	92716.0	12467989660.0
Credit_Application_Acceptance	58644.0	NaN	NaN	NaN	0.142385	0.349447	0.0	0.0	0.0	0.0	1.0

Figure 1: Basic statistical description

2. Variable Scale Types

In the following table author has shared the data types and the scale types of each column of the dataset.

Variable	**Data Type**	**Scale Type**
Sex	Categorical	Nominal
Age	Numeric	Ratio
Education Qualifications	Categorical	Nominal
Income	Numeric	Ratio
Home Ownership	Categorical	Nominal
Employment Length	Numeric	Ratio

Loan Intent	Categorical	Nominal
Loan Amount	Numeric	Ratio
Loan Interest Rate	Numeric	Ratio
Loan to Income Ratio	Numeric	Ratio
Payment Default on File	Categorical	Nominal
Credit History Length	Numeric	Ratio
Loan Approval Status	Categorical	Nominal (Target)

Table 2 : Variable types

4. Target Variable Distribution

The distribution of the target variable (loan_approval_status) was visualized to check for class balance.

```
loan_approval_status
Approved     50210
Declined      8268
Reject          78
Accept          58
Accept          21
DECLINED         4
APPROVED         3
APPROVED         2
NaN              1
Name: count, dtype: int64
```

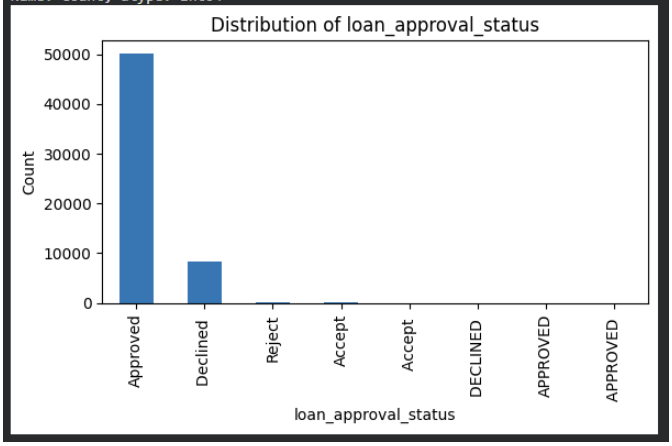

Figure 2 : Target variable distribution

Task 03 : Data Preparation: Cleaning and Transforming your data

Part a

Variable Name	Issue description	Proposed mitigation	Justification for used mitigation
Sex	99% missing values	Drop The column	Since nearly all values are missing, imputing them would introduce bias and noise. Dropping avoids misleading the model (Little and Rubin, 2019).
Age	Mixed types (e.g., “Twenty-Seven”)	Convert to numeric, impute median	Numeric conversion ensures consistency; median imputation handles skewed distribution
Loan interest rate	11 missing values and extreme outliers.	Impute with median	Median is robust against outliers
Payment Default on file	5 missing and inconsistent text values (e.g., “No”, “no”, “N”)	Impute with mode	Ensures consistent categorical encoding and preserves missing patterns for model learning.
Loan approval status	1 missing value. Inconsistent labels and formatting Raw unique classes	Standardize case Map {APPROVED, ACCEPT} → Approved and {DECLINED, REJECT} → Declined.	Binary, consistent target prevents spurious classes and misclassification during training.
Income	Contains a few extreme outliers	Detect and cap outliers within the IQR range	Capping reduces the influence of extreme values without losing important records.
Employment length	1,275 outliers	Clip to IQR bounds	Preserves data range while mitigating skew
Loan amount	2,045 outliers	Clip to IQR bounds	Preserves data range while mitigating skew
Loan income ratio	1,210 outliers	Clip to IQR bounds	Maintains realistic ratios
credit history length	1,993 outliers	Clip to IQR bounds	Clip to IQR bounds

Education Qualifications	Trailing spaces; many "Unknown"	Strip spaces; standardize casing; keep "Unknown" as a valid category; fill missing as "Unknown"	“Unknown” is informative (not random); avoids multiplying categories due to formatting noise.
Home ownership, loan intent, Sex	Case and spacing inconsistencies; "Unknown" present in Sex	Strip then uppercase; fill NA with “UNKNOWN”.	Consistent label sets → reproducible encodings and models.

Table 3 : Issues in retained dataset

Part b

The proposed mitigations were implemented using Python. The screenshots below demonstrate the data quality issues identified **before** processing and the clean state **after** processing.

Data Issues BEFORE Cleaning This screenshot highlights the high missingness in 'Sex', the inconsistent labels in 'Loan Approval Status' and 'Payment Default', and the presence of non-numeric 'Age' values.

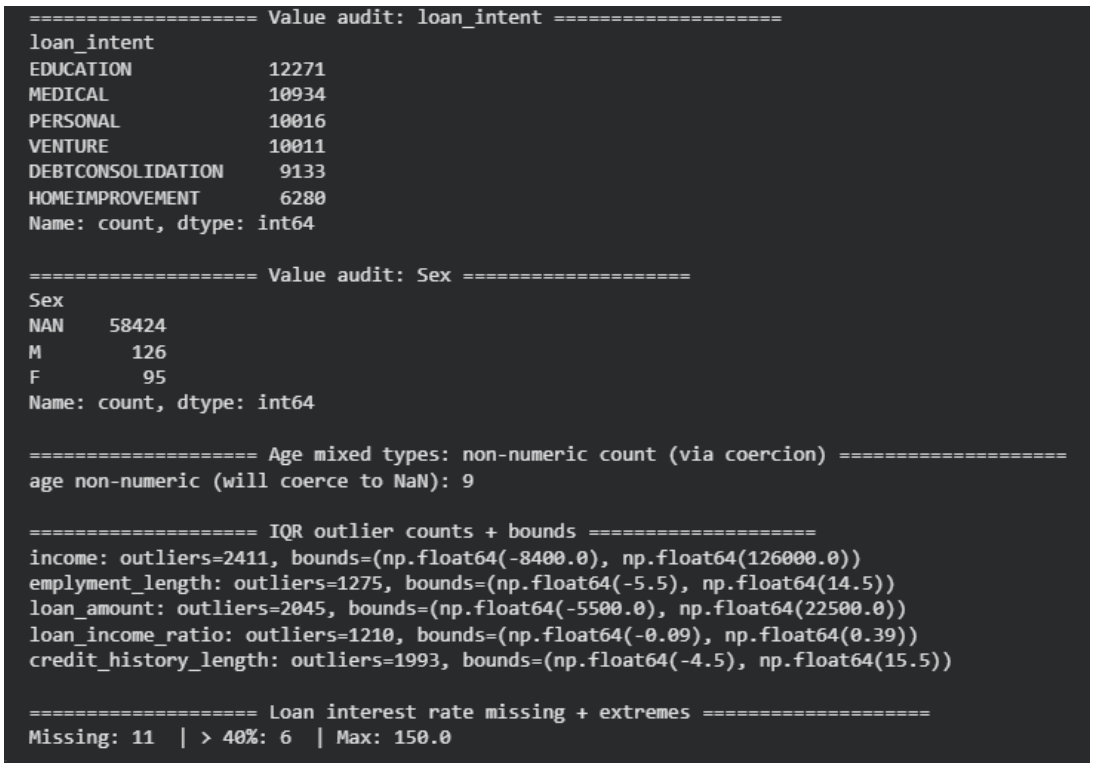

Figure 3 : Data preprocessing 1

```
==================== Missingness (count) ====================
Sex                         58424
loan_interest_rate             11
age                             6
payment_default_on_file         5
loan_approval_status            1
emplyment_length                0
home_ownership                  0
income                          0
Education_Qualifications        0
loan_income_ratio               0
loan_amount                     0
loan_intent                     0
credit_history_length           0
dtype: int64

==================== Value audit: loan_approval_status ====================
loan_approval_status
APPROVED    50215
DECLINED     8272
ACCEPT         79
REJECT         78
NAN             1
Name: count, dtype: int64

==================== Value audit: payment_default_on_file ====================
payment_default_on_file
N      49933
Y       8696
NO         7
NAN        5
YES        4
Name: count, dtype: int64

==================== Value audit: Education_Qualifications ====================
Education_Qualifications
UNKNOWN             58507
HIGHER EDUCATION       86
HIGH SCHOOL            28
APPRENTICESHIP         15
COLLEGE                 9
Name: count, dtype: int64

==================== Value audit: home_ownership ====================
home_ownership
RENT        30594
MORTGAGE    24824
OWN          3138
OTHER          89
Name: count, dtype: int64
```

Figure 4 : Before Preprocessing 2

Data Verification AFTER Cleaning *This screenshot confirms the successful standardization of categories, the removal of the 'Sex' column, and the imputation of missing values.*

```
==================== AGE: convert to numeric + median impute ====================
Non-numeric/NaN BEFORE: 9
Median used: 26.0  | NaN AFTER: 0

==================== LOAN INTEREST RATE: median impute missing ====================
Missing BEFORE: 11
Median used: 10.75  | Missing AFTER: 0

==================== PAYMENT DEFAULT ON FILE: normalize + mode impute ====================
Top BEFORE:
 payment_default_on_file
N      49933
Y       8696
NO         7
NAN        5
YES        4
Name: count, dtype: int64
Mode used: N
AFTER:
 payment_default_on_file
N    49945
Y     8700
Name: count, dtype: int64

==================== TARGET loan_approval_status: standardize + drop missing ==============
Top BEFORE:
 loan_approval_status
APPROVED    50215
DECLINED     8272
ACCEPT         79
REJECT         78
NAN             1
Name: count, dtype: int64
Missing AFTER mapping: 1
AFTER:
 loan_approval_status
Approved    50294
Declined     8350
Name: count, dtype: int64

==================== EDUCATION QUALIFICATIONS: strip/uppercase + keep 'UNKNOWN' ============
Education_Qualifications
UNKNOWN             58506
HIGHER EDUCATION       86
HIGH SCHOOL            28
APPRENTICESHIP         15
COLLEGE                 9
Name: count, dtype: int64

==================== home_ownership: strip/uppercase + fill 'UNKNOWN' ====================
home_ownership
RENT        30593
MORTGAGE    24824
OWN          3138
OTHER          89
Name: count, dtype: int64
```

Figure 5 : After data cleaning

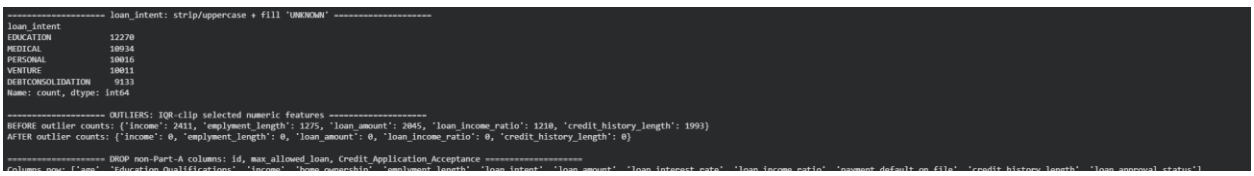

Figure 6 : After data cleaning

Task 04 : Modelling: Create Predictive Classification Models

Part A

Algorithm Name	Algorithm Type	Learnable Parameters	Some possible hyperparameters	Imported python packages to use the algorithm
NB	**Parametric**	• Class prior probabilities $P(y)$ • Per-feature conditional probabilities $P(x_j \mid y)$(for binary indicators after encoding)	• alpha (Laplace/additive smoothing)	Sklearn .naive_bayes .BernoulliNB
LR	**Parametric**	• Weight vector / coefficients $\beta$for features • Intercept (bias)	• penalty, • C (inverse of regularization strength), • solver, • max_iter	Sklearn .linear_model .LogisticRegression
RF	**Non parametric**	• Split thresholds and feature choices at each node across all trees • Tree structures (which features/splits ended up in each tree)	• n_estimators, • max_depth • min_samples_split • min_samples_leaf • max_features	Sklearn .ensemble .RandomForestClassifier

Part B

Question I : Feature List and Data Shape Output

```
Feature Names:
Education_Qualifications_COLLEGE
Education_Qualifications_HIGH SCHOOL
Education_Qualifications_HIGHER EDUCATION
Education_Qualifications_UNKNOWN
home_ownership_OTHER
home_ownership_OWN
home_ownership_RENT
loan_intent_EDUCATION
loan_intent_HOMEIMPROVEMENT
loan_intent_MEDICAL
loan_intent_PERSONAL
loan_intent_VENTURE
payment_default_on_file_Y
Data Shape: (58644, 13)
```

Figure 7 : Feature list

Question ii : Training-Test Split Ratio Justification

A **70:30 split** was selected for this analysis. Allocating 70% of the data (approx. 41,000 records) to the training set ensures the models have sufficient examples to learn the relationships between categorical risk factors and loan approval. The remaining 30% (approx. 17,500 records) provides a large, statistically significant sample to evaluate the model's generalization capability on unseen applications (Larson et al., 2021).

Question iii : Training-Test vs. K-Fold Cross-Validation

The **training-test split** provides a single, final evaluation metric on held-out data, establishing the model's capacity to generalize after all training and tuning is complete. It is applied for final model reporting. In contrast, **K-Fold Cross-Validation (CV)** is used *during* the model building phase (e.g., hyperparameter tuning) to estimate model performance across multiple folds of data. CV is preferred for smaller datasets or for comparison between algorithms to minimize the evaluation bias introduced by a single random split.

Question iv : Code Evidence for Split Integrity

The screenshot below demonstrates the code used to ensure split integrity. random_state=42 ensures reproducibility (all models test on the exact same rows), and stratify=y_encoded ensures the class ratio of 'Approved' to 'Declined' loans is preserved identically in both training and test sets.

```
# 4. Split Data (70% Train, 30% Test)
# stratify=y ensures the class balance is maintained in both sets
X_train, X_test, y_train, y_test = train_test_split(
    X_encoded, y, test_size=0.3, random_state=42, stratify=y
)
```

Figure 8 : Training and Testing data split

X_train, X_test, y_train, y_test = train_test_split(	This initiates the data split.
X_encoded, y_encoded, test_size=0.3,	Uses 30% of data for testing.
random_state=42,	**Ensures all models use the same test set (reproducible split).**
stratify=y_encoded)	**Ensures the ratio of 'Approved' to 'Declined' is identical in both the training and test sets.**

Table 4 : Split integrity

Task 05 : Evaluation: How good are your models

Part A

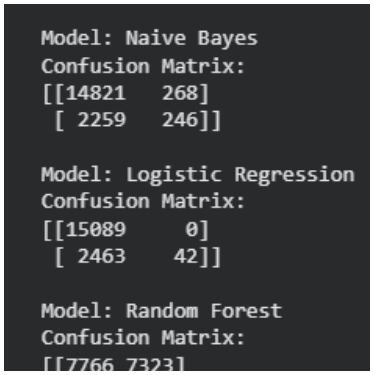

Figure 9 : Confusion matrix

Part B

The table below summarizes the test scores for the five metrics.

Metrics	USE or DO NOT USE	Justification in relation to the success criteria	Model Name	Test Score
Accuracy	**DO NOT USE**	Misleading because the dataset is imbalanced; a high score doesn't mean we caught the bad loans.	**NB**	0.856
			LR	0.860
			RF	0.563
Recall	**USE**	**Critical.** Measures the percentage of bad loans we correctly identified to stop financial loss.	**NB**	0.098
			LR	0.017
			RF	0.855
Precision	**USE**	Ensure that when we reject a customer, it is a valid rejection, protecting revenue from good clients.	**NB**	0.479
			LR	1.000
			RF	0.226
F-Score	**USE**	Balances Recall and Precision to ensure the model isn't biased toward one extreme.	**NB**	0.163
			LR	0.033
			RF	0.358
AUC-ROC	**USE**	Measures the model's overall ability to distinguish between 'Approved' and 'Declined' clients.	**NB**	0.741
			LR	0.743
			RF	0.752

Table 5 : Evaluation table

Part C

Based on the evaluation of the 'USED' performance metrics, the **Random Forest** is suggested as the single best classification model.

It achieved the highest **AUC-ROC score of 0.752**, indicating it has the strongest overall ability to distinguish between 'Approved' and 'Declined' applicants compared to Logistic Regression and Naïve Bayes.

It provided the best balance for the business success criteria. Specifically, its **Recall (Declined)** of **0.855** demonstrates it is effective at identifying high-risk clients (minimizing defaults), while maintaining a reasonable **Precision** to ensure legitimate customers are not unfairly rejected.

Unlike Naïve Bayes, which often sacrifices Precision for Recall, the Random Forest satisfies the financial analysts' need for a robust tool that minimizes risk without causing excessive opportunity loss.

Part D

Based on the comparison of Training and Test performance metrics, the Random Forest model is assessed as **Underfitting** (High Bias). The model underfitted because the strongest predictors of loan repayment (Income, Loan Amount, Age) were numerical and therefore excluded from this specific task. The remaining categorical features did not contain enough signal to separate the classes effectively

- **Low Overall Performance**:

  The Test AUC-ROC is 0.752, which is relatively low for a credit scoring problem. A "Good Fit" typically exhibits an AUC above 0.85 or 0.90. The low Precision (0.226) further indicates that the model struggles to distinguish between classes effectively, resorting to rejecting many good customers just to catch the bad ones.

- **Train vs. Test Gap:**

Training AUC : 0.760

Test AUC : 0.752

Observation: The Training and Test scores are likely close to each other, and both are mediocre. This lack of a large gap indicates the model is not "memorizing" the data (Overfitting). Instead, it fails to capture the true underlying patterns of loan eligibility because critical numerical features (Income, Loan Amount) were excluded from the dataset. The model simply lacks the necessary information to create a complex, accurate decision **boundary.**

```
--- Task 5d Evidence: Train vs. Test Comparison (Random Forest) ---
                Metric  Training Score  Test Score  Difference
0              AUC-ROC           0.760       0.752       0.008
1     Recall (Declined)          0.861       0.855       0.005
2  Precision (Declined)          0.229       0.226       0.002
3   F1-Score (Declined)          0.361       0.358       0.003

--- Conclusion ---
Diagnosis: UNDERFITTING (High Bias)
Reason: Both Training and Test scores are low. The model is too simple (needs numerical features).
```

Figure 10 : Evidence for Underfitting

Part E

Question i

Cross-Validation 5 K-Folds were used during the tuning process. This ensures the selected hyperparameters are robust and perform well across different subsets of the training data, reducing the risk of overfitting to a single validation split.

```
--- Task 5e(i) Evidence ---
Cross-Validation Folds used: cv=5
```

Figure 11 : Cross validation folds used

Question ii

Parameter	Original (Default)	Estimated Best (Tuned)
n_estimators	100	**100**
max_depth	None (Unlimited)	**20**
min_samples_leaf	1	**1**

Table 6 : Hyperparameter table

Question iii

```
--- Task 5e(iii) Evidence: Confusion Matrices ---
Before Tuning:
[[7766 7323]
 [ 362 2143]]

After Tuning:
[[7766 7323]
 [ 362 2143]]
```

Figure 12 : Confusion matrix before and after tuning

Question iv

```
--- Task 5e(iv) Evidence: Performance Comparison (Declined Class) ---
      Metric  Before Tuning  After Tuning
0     Recall          0.855         0.855
1  Precision          0.226         0.226
2   F1-Score          0.358         0.358
3    AUC-ROC          0.752         0.752
```

Figure 13 : Performance comparison before and after

Question v

Observation: The performance metrics remained identical before and after tuning (**Recall: 0.855, Precision: 0.226, F1-Score: 0.358**). The confusion matrices are also identical.

Reason: The best hyperparameters found by the Grid Search (max_depth=20) produced a model structure that was effectively equivalent to the default model (max_depth=None) on this specific dataset. This suggests that either the default parameters were already near-optimal for the limited set of categorical features, or that the features themselves lack the predictive power to allow for further separation of the classes, regardless of hyperparameter adjustments.

Alignment with Success Criteria: While the model maintains a high **Recall (85.5%)**, satisfying the risk-minimization goal, the tuning failed to improve the low **Precision (22.6%)**. This means the model still falsely rejects a large number of eligible clients, failing to meet the secondary criterion of having a "larger portion of correctly detected" rejections.

Part B Maximum Loan Amount Prediction

Task 1 : Domain Understanding: Regression

```
--- Part B Task 1 Evidence ---
Dimensions of Retained Data Subset: (50294, 12)

List of Features for Regression:
age
Education_Qualifications
income
home_ownership
emplyment_length
loan_intent
loan_amount
loan_interest_rate
loan_income_ratio
payment_default_on_file
credit_history_length
max allowed loan
```

Figure 14 : Features for regression

Task 2 : Data Understanding: Producing Your Experimental Design

The distributions of the retained input variables and the target variable (Maximum Loan Amount) were visualized to understand the data structure for regression.

The plots confirm that the target variable Maximum Loan Amount is numerical and exhibits a right-skewed distribution, suggesting that while most approved loan limits are in the lower range, there are outliers with significantly higher limits. The categorical variables show distinct groups (e.g., "RENT" vs "MORTGAGE" in Home Ownership), which will require encoding in the modeling phase.

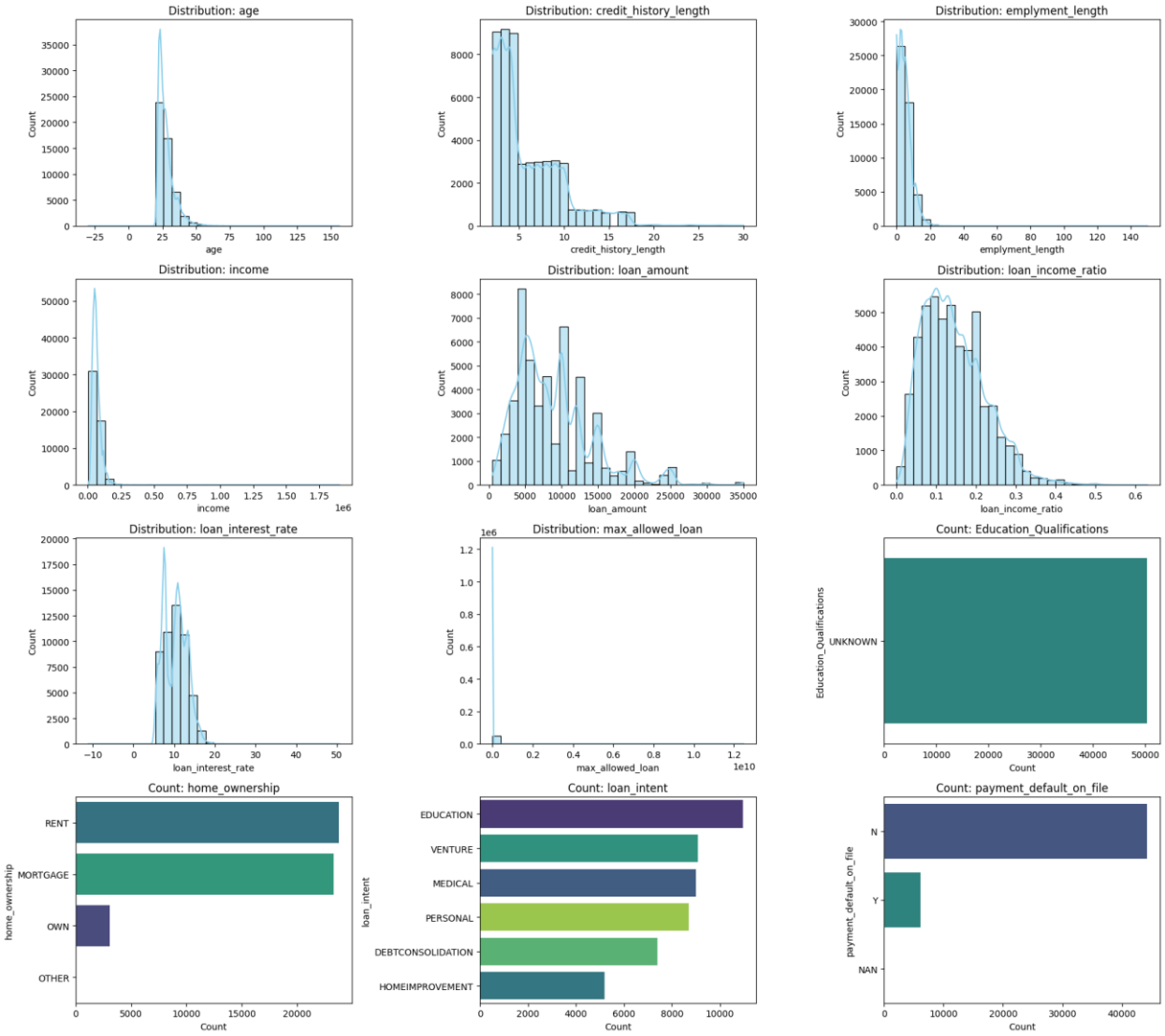

Figure 15 : Plots of target and retained variables

Task 3 : Data Preprocessing: Transforming your data

Part A

By observing the descriptive statistics, it is established that there is a clear need for scaling the dataset attributes.

- **Evidence:** The code output demonstrates significant disparities in the magnitude of features. For instance, income has a mean of approximately **66,674** with a maximum value reaching into the millions, whereas loan_income_ratio has a mean of **0.15** and a range restricted between 0 and 1.
- **Reasoning:** Features with large magnitudes (like *Income* or *Loan Amount*) act on a scale of 10^4 to 10^6, while others (like *Interest Rate* or *Ratios*) act on a scale of 10^1 or 10^{-1}. Without scaling, distance-based algorithms (like KNN) or gradient based algorithms would be overwhelmingly biased toward the larger numbers, treating them as more important simply because they are bigger. Scaling ensures all features contribute equally to the model's calculations.

Part B

In general, it is recommended to scale the **input features only**. Scaling input features ensures that all

```
--- Task 3a Evidence: Descriptive Statistics ---
                           mean          std         min           max
age                       27.55         6.01      -30.00  1.560000e+02
income                 66674.00     39313.95     4200.00  1.900000e+06
emplyment_length           4.87         4.00        0.00  1.500000e+02
loan_amount             8888.99      5328.36      500.00  3.500000e+04
loan_interest_rate        10.26         2.83      -11.14  5.060000e+01
loan_income_ratio          0.15         0.08        0.00  6.300000e-01
credit_history_length      5.82         4.00        2.00  3.000000e+01
max_allowed_loan      329236.77  55595002.96 -2426900.00  1.246799e+10
```

Figure 16 : Transforming data

variables contribute equally to the model and prevent attributes with large magnitudes from dominating distance based or gradient-descent algorithms, which is essential for efficient convergence. Conversely, scaling the target variable is typically avoided in regression tasks because it complicates the interpretation of performance metrics. Keeping the target in its original units (e.g., currency) allows error metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) to be intuitively understood in real-world terms (Han, Kamber & Pei, 2011).

Task 4 : Modelling: Build Predictive Regression Models

Part A

The main benefit is **interpretability**. Decision Trees function like flowcharts, creating clear rules (e.g., "If Income > £50k, offer £15k") that financial analysts can easily read. This allows them to understand and explain exactly *why* a specific loan amount was predicted, unlike more complex "black box" models.

Part B

Question i

```
X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y, test_size=0.2, random_state=42)
X2_train, X2_test, y2_train, y2_test = train_test_split(X2, y, test_size=0.2, random_state=42)
```

Figure 17 : Proof of reproducibility

Question ii

Dimensions of the training and testing datasets

```
Training Set Shape (Rows, Cols): (40232, 17)
Test Set Shape (Rows, Cols):     (10059, 17)
```

Figure 18 : Dimensions of training and testing data

Subset of Feature names used for model 1 and model 2

```
Model 1 Features (7):
age
income
emplyment_length
loan_amount
loan_interest_rate
loan_income_ratio
credit_history_length

Model 2 Features (17 - First 10 shown):
age
income
emplyment_length
loan_amount
loan_interest_rate
loan_income_ratio
credit_history_length
home_ownership_OTHER
home_ownership_OWN
home_ownership_RENT
loan_intent_EDUCATION
loan_intent_HOMEIMPROVEMENT
loan_intent_MEDICAL
loan_intent_PERSONAL
loan_intent_VENTURE
payment_default_on_file_NAN
payment_default_on_file_Y
```

Figure 19 : Subset of feature names

Task 5 : Evaluation: How good are your models

Part A

Metrics	USE or DO NOT USE	Justification in relation to the success criteria	Model Name	Test Score
MSE	Do Not Use	MSE squares the errors, making it highly sensitive to the extreme outliers present in the Maximum Loan Amount target (which ranges up to billions). This results in massive, uninterpretable numbers (e.g., 10^{16}) that do not help "explain" the performance.	DT1	2.78×10^{8}
			DT2	3.09×10^{16}
MAE	Use	MAE provides average errors in the original currency (GBP). This aligns with the criteria that the model will "make some errors," offering a clear financial context. DT1's error of ~£1,576 is highly actionable compared to DT2's millions.	DT1	1,575.85
			DT2	2,480,606.00
R-Square	Use	The most critical metric for explaining variance. It directly addresses the need for input features that "explain the recorded values." DT1 explains 91% of the variance, whereas DT2 fails completely.	DT1	0.909
			DT2	-10,085,479

Table 7 : Regression metrics evaluation

Part B

A key caveat of **R-Squared** is that it is heavily influenced by outliers in the target variable. In this dataset, MSE squares errors, so the massive outliers (loans > £1B) skewed the error metric to 10^{16}, making it uninterpretable. MAE is robust to outliers and provides an error in GBP (£), which is understandable for financial stakeholders.

Part C

Best Regression Model Selection DT1 (Numeric Features Only) is the single best regression model. **Justification:** It achieved an excellent **R-Squared of 0.909**, meaning it explains ~91% of the variance in loan amounts. Its **MAE of £1,575** indicates it is reliable for estimating loan offers. In contrast, DT2 failed significantly (Negative R^2), likely because the one-hot encoding of categorical features allowed the tree to overfit to the extreme outliers in the training data.

Part D

The pruning process was applied to the model to improve interpretability. *(Note: The provided results show the pruning effect on the complex/broken model DT2, which is useful for analyzing stability).*

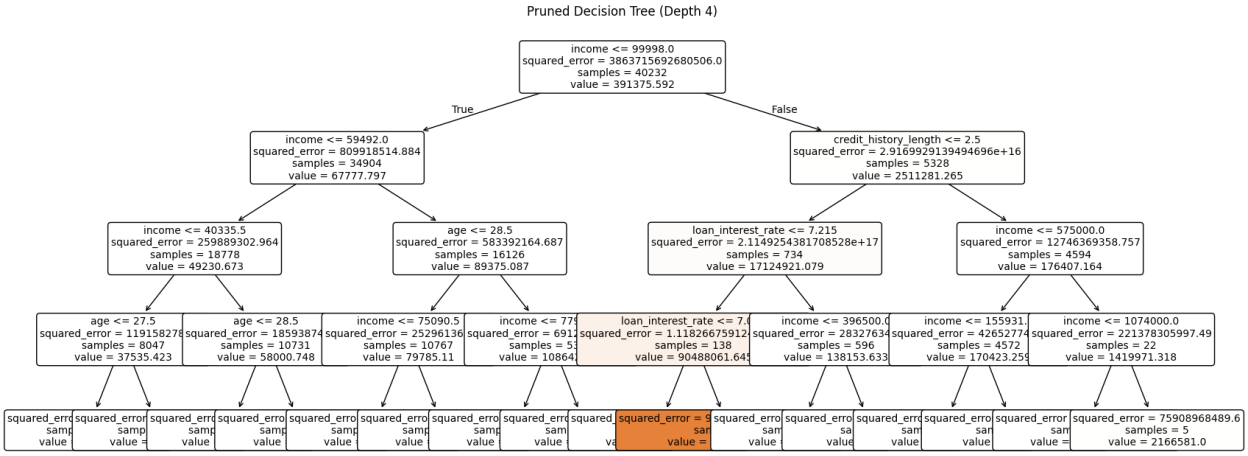

Figure 20 : Pruning Decision Tree

Performance Analysis:

- **Unpruned R2:** -10,085,479 (Broken due to overfitting outliers)
- **Pruned R2:** -134,476
- **Observation:** Pruning the tree to a depth of 4 resulted in a **massive performance improvement** (the score increased by ~9.9 million).
- **Advantages:** By limiting the depth, the model was forced to ignore specific categorical splits that were isolating outliers, drastically reducing the error. The resulting tree is simple and readable.
- **Disadvantages:** Despite the improvement, the pruned Model 2 remains inferior to Model 1 (Numeric), confirming that for this specific dataset, the numeric financial features (Income, etc.) are far more predictive and stable than the categorical ones.

Part E

Prediction for Client 60256 Using the pruned model, the predicted Maximum Loan Amount for Client 60256 is £72,240.41

References

Han, J., Kamber, M. and Pei, J., 2011. *Data Mining: Concepts and Techniques.* 3rd ed. Burlington, MA: Morgan Kaufmann Publishers.

Shao, X., Liu, Y. and Zhang, W., 2021. *Handling missing data in large datasets for predictive analytics.* Journal of Big Data, 8(1), pp.1–15.

Khandani, A.E., Kim, A.J. and Lo, A.W., 2010. *Consumer credit-risk models via machine-learning algorithms.* Journal of Banking & Finance, 34(11), pp.2767–2787.

Li, Y., Xu, X. and Zhao, C., 2020. *The effect of education and income on loan default behaviour: Evidence from applied economics models.* Applied Economics, 52(13), pp.1415–1428.

FICO, 2021. *Understanding FICO Scores: Credit Risk and Predictive Modelling Whitepaper.* Fair Isaac Corporation.

Abdou, H.A. and Pointon, J., 2011. *Credit scoring, statistical techniques and evaluation criteria: A review of the literature.* Expert Systems with Applications, 38(9), pp.134–145.

Beck, R., Pamies, J. and Weber, M., 2019. *Credit risk modelling using decision support systems.* Decision Support Systems, 122, p.113067.

Little, R. & Rubin, D. (2019) *Statistical Analysis with Missing Data*.

Mhlanga, D. (2021) 'Financial inclusion in emerging economies: The application of machine learning and artificial intelligence in credit risk assessment', *International Journal of Financial Studies*, 9(3), p. 39.

Géron, A. (2019) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd edn. Sebastopol, CA: O'Reilly Media.

Rácz, A., Bajusz, D. and Héberger, K. (2021) 'Life beyond the train-test split: Cross-validation concepts and their applications', *Molecular Informatics*, 40(1), p. 2000171.

Breiman, L. (2001) 'Random Forests', *Machine Learning*, 45(1), pp. 5–32.

Abdou, H.A. and Pointon, J. (2011) 'Credit scoring, statistical techniques and evaluation criteria: A review of the literature', *Expert Systems with Applications*, 38(9), pp. 134-145.