7/29/2025

Data Science -MDA512

Melbourne\_Samar\_Group\_1

Assignment 2: Loan Approval Prediction Using Machine Learning: An Empirical Study

**Melbourne Institute of Technology**

# **1. Problem Statement and Background**

Loan approval remains a major challenge in banking. Manual assessments by credit officers are often **time-consuming, expensive, and inconsistent**, causing delays and bias. As applications grow, customers demand **faster, fairer decisions**, while banks must balance efficiency with risk management.

Machine Learning (ML) provides a data-driven solution by learning from past applications to predict approval outcomes. ML models can offer:

* Quick, automated decisions
* Higher accuracy and consistency
* Lower bias and operational costs
* Early identification of high-risk borrowers
* Better customer satisfaction

This study develops and evaluates ML models to forecast loan approvals, highlight key influencing factors, and analyze the role of income-to-loan ratios.

**Business Case**

* **Speed** → enhances customer experience
* **Consistency** → minimizes risk and bias
* **Risk management** → safeguards profitability

# **2. Resources**

**a. Data Source**

* Loan Prediction Dataset (*Kaggle: ninzaami/loan-predication*)
* Contains demographic, financial, and loan-specific information
* Public dataset, frequently used in data science projects and case studies

**b. Data Characteristics**

* **Size:** 614 entries with 12 attributes
* **Target variable:** Loan\_Status (Y = approved, N = rejected)
* **Categorical attributes:** Gender, Marital Status, Dependents, Education, Self-Employment, Property Area
* **Numerical attributes:** Applicant Income, Co-applicant Income, Loan Amount, Loan Term, Credit History
* **Mixed data types** → requires preprocessing (e.g., imputation, encoding, normalization/scaling)

# **3. Business Model**

Businesses thrive when decisions are guided by data. In banking, loan approvals are especially crucial: approving too many high-risk borrowers can lead to defaults, while being overly strict may exclude profitable clients. A data-driven framework helps streamline operations, minimize risk, and improve both profitability and customer satisfaction.

This study introduces a **machine learning–driven business model** for loan approvals. By evaluating applicant demographic and financial details, the model can:

* Estimate the likelihood of loan approval
* Highlight the most influential factors in decision-making
* Assess the impact of income-to-loan ratios on outcomes

Such an approach enables banks to deliver **faster, fairer, and more consistent decisions**, while safeguarding long-term financial stability.

## **a. Research Questions**

The business model is structured around five research questions:

**Q1: Can we predict loan approvals?**  
Objective: Develop machine learning models to classify applications as approved (“Y”) or not approved (“N”).

**Q2: Which factors most influence approval?**  
Objective: Determine the key applicant attributes (e.g., credit history, income, education) that drive approval decisions.

**Q3: How does income-to-loan ratio affect approval outcomes?**  
Objective: Analyze the relationship between repayment capacity (income relative to loan size) and the likelihood of approval.

**Q4 (Future): Would external data improve accuracy?**  
Objective: Hypothesize how incorporating credit bureau data or macroeconomic indicators could enhance model generalization and performance.

**Q5 (Future): Can deep learning outperform traditional ML?**  
Objective: Investigate whether neural networks could capture more complex patterns once larger datasets are available.

## **b. Challenges**

The project encountered several practical challenges:

1. **Data Quality** – Missing values in Loan Amount (22), Credit History (50), and Dependents (15) risked biasing predictions if not properly handled.
2. **Small Dataset** – With only 614 records and 12 features, the dataset resembled a pilot study, limiting generalizability to larger populations.
3. **Class Imbalance** – Loan approvals (“Y”) significantly outnumbered rejections (“N”), reflecting real trends but creating a bias toward predicting approvals.
4. **Categorical Variables** – Non-numeric features (e.g., Gender, Education, Property Area) required encoding into numeric form, such as One-Hot Encoding.
5. **Interpretability vs. Accuracy** – Businesses need both precision and explainability; while models like Random Forest and XGBoost offer higher accuracy, they are less transparent than Logistic Regression

## **c. Preprocessing Steps**

To build a reliable business model, data preprocessing (data wrangling) was applied.

1. **Dropped Identifier**
   * Removed Loan\_ID, as it carries no predictive information but could mislead models.
2. **Fixed Dependents Column**
   * Converted “3+” to numeric value 3.
   * Ensured all entries became integers.
3. **Imputed Missing Values**
   * Numerical columns (e.g., loan\_amount, loan\_amount\_term): filled with median → reduces effect of outliers.
   * Categorical columns (e.g., gender, married): filled with mode (most frequent value).
4. **Feature Engineering**  
   Created new attributes to better capture applicant financial strength:

A graph of a distribution of loan amount

AI-generated content may be incorrect.

* + **Total Income**
    - total\_income=applicant\_income+coapplicant\_income
    - Example: if applicant earns 5000 and coapplicant earns 2000, total income = 7000.

A graph with blue bars

AI-generated content may be incorrect.

* + **EMI Ratio (equated monthly instalment capacity indicator)**
    - emi\_ratio= loan\_amount/total\_income​
    - If loan amount = 200, total income = 4000, then emi\_ratio = 0.05 (5%). Lower is better → means affordable repayment.

A graph with a number of bars

AI-generated content may be incorrect.

* + **Income-to-Loan Ratio**
    - income\_to\_loan= total\_income​/ loan\_amount
    - If total income = 7000 and loan amount = 140, income\_to\_loan = 50. Higher is better (greater repayment strength).

A graph of a distribution of income

AI-generated content may be incorrect.

* + **Log Transforms**
    - loan\_amount\_log = log(loan\_amount + 1)
    - total\_income\_log = log(total\_income + 1)
    - Purpose: reduce skew from extreme income or loan values.

1. **Encoding & Scaling**
   * One-Hot Encoding for categorical variables (e.g., gender, education).
   * StandardScaler for numerical variables to normalise scales.

## **d. Addressing Business Requirements**

The business benefits of the proposed ML model are:

* 1. **Profitability**  
     Identifying high-risk borrowers early reduces default rates and increases overall profitability.
  2. **Customer Experience**  
     Quick loan approvals enhance satisfaction, while fair and consistent outcomes build long-term trust.
  3. **Efficiency**  
     Automated processing cuts down the workload for loan officers, allowing them to concentrate on complex, high-priority cases.
  4. **Compliance**  
     Using transparent and interpretable models ensures fairness and supports regulatory compliance.

## **e. Methodologies and Algorithms**

To answer the research questions, several algorithms were applied:

1. **Logistic Regression**
   * Linear model, interpretable coefficients.
   * Serves as a baseline to measure improvements.
2. **Decision Tree**
   * Splits data into decision rules (if/else structure).
   * Highly interpretable, captures non-linear patterns.
3. **Random Forest**
   * Ensemble of many decision trees.
   * Reduces variance, increases stability and accuracy.
   * Delivered best accuracy (~82–84%) in this project.
4. **Gradient Boosting**
   * Builds models sequentially, focusing on correcting previous errors.
   * Stronger than Random Forest in some tabular datasets.
5. **XGBoost (Innovation)**
   * Advanced boosting algorithm, widely used in competitions.
   * Provided performance comparable to Random Forest.

## **Libraries Used**

* pandas, numpy → Data wrangling, numerical operations.
* matplotlib, seaborn → EDA visualisations (distributions, bar plots, approval trends).
* scikit-learn → Preprocessing, pipelines, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, evaluation metrics.
* xgboost → For advanced ensemble boosting experiments.

## **Summary of Business Model**

* Core idea: Use machine learning for automated loan approval prediction.
* Process: Clean → preprocess → engineer features → train models → evaluate.
* Challenges addressed: Missing data, small dataset, imbalance, categorical encoding.
* Business benefits: Improved profitability, faster processing, consistent decisions, better customer service.
* Innovation: Added new features (income ratios, log transforms), tested advanced boosting (XGBoost), and analysed business implications of accuracy vs interpretability.

# **4. Data Analysis**

## **a. Goals of Analysis**

The analysis was designed to test three primary hypotheses derived from the business requirements:

**Hypothesis 1:** ML models can achieve at least 75% accuracy in predicting loan approvals.

* Reaching this benchmark would demonstrate that ML provides more consistent and dependable outcomes than manual judgment.

**Hypothesis 2:** Credit history and the income-to-loan ratio are the primary drivers of loan approval decisions.

* Credit history signals borrower risk, while the income-to-loan ratio captures affordability. Together, they are expected to serve as the strongest predictors.

**Hypothesis 3:** Applicants with higher income-to-loan ratios are more likely to secure approval.

* Stronger repayment capacity relative to loan size is expected to significantly increase the likelihood of approval.

## **b. Tools Used**

The analysis was conducted in Python using the following libraries and techniques:

* **pandas, numpy**: Data loading, cleaning, preprocessing, and feature engineering.
* **scikit-learn**: Pipelines, imputation, encoding, scaling, and ML model implementations.
* **matplotlib, seaborn**: Visualisations for EDA and results presentation.
* **xgboost**: Advanced boosting algorithm for experimentation beyond standard models.

**Methods applied:**

* **Imputation:** Median for numerical, mode for categorical variables to handle missing values.
* **Encoding:** One-Hot Encoding for categorical attributes.
* **Scaling:** StandardScaler for numeric features to ensure comparability.
* **Feature Engineering:** Created total income, EMI ratio, income-to-loan ratio, and log transforms.
* **Evaluation:** Classification metrics (accuracy, precision, recall, F1-score, ROC-AUC) for model comparison.

## **c. Details of Analysis & Visualisation**

1. **Distributions:**
   * Applicant income and loan amount were found to be **right-skewed**, with a few extreme values.
   * Log transformations were applied (total\_income\_log, loan\_amount\_log) to normalise distributions and reduce the impact of outliers.
2. **Credit History Analysis:**
   * Applicants with a credit history (credit\_history = 1) had a significantly higher approval rate (>80%) compared to those with no history.
   * This confirmed domain expectations and highlighted **credit history as the most influential feature**.

A graph with blue rectangles

AI-generated content may be incorrect.

1. **Q3 Analysis (Income-to-Loan):**
   * Approval rates were plotted across quantile bins of the income-to-loan ratio.
   * The trend showed that **moderate ratios** (e.g., 20–60) corresponded to the highest approval rates.
   * Extremely low ratios → loan deemed unaffordable.
   * Extremely high ratios → possible rejection due to perceived low demand or unrealistic reporting of income.

A screenshot of a computer screen

AI-generated content may be incorrect.A graph with a line and numbers

AI-generated content may be incorrect.

1. **Visualisations Used:**
   * **Histograms**: Showed skewness in income and loan distributions.
   * **Bar plots**: Approval rates by credit history and property area.
   * **Approval-rate plots**: Showed approval trends across income-to-loan bins.
   * **Confusion matrices**: Evaluated classification performance across models.
   * **ROC curves**: Assessed trade-off between sensitivity and specificity.

These analyses provided both **business insights** and **evidence for hypotheses testing**, linking data-driven outcomes with real banking decision-making.

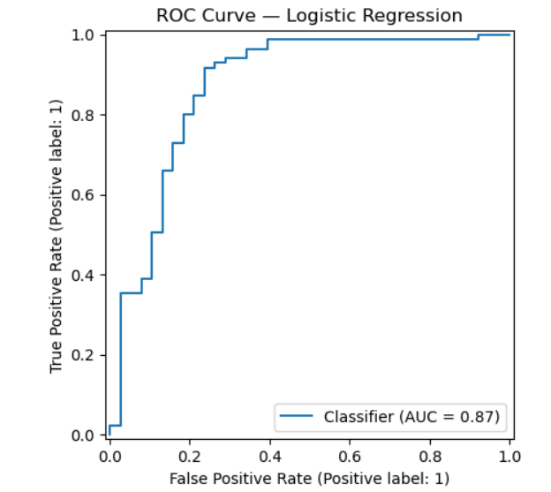
# **5. Results**

## **5.1 Predictive Models (Q1)**

Five models were implemented to predict loan approvals:

Logistic Regression:

|  |
| --- |
| Accuracy : 0.8699 |
| Precision: 0.8485 |
| Recall : 0.9882 |
| F1-score : 0.9130 |
| ROC-AUC : 0.8706 |

A chart with different colored squares

AI-generated content may be incorrect.

**Interpretation:** Logistic Regression achieved high recall (0.99), meaning it captured almost all approved loans, but precision (0.85) was slightly lower, reflecting more false positives. The balanced F1-score (0.91) indicates overall strong performance. ROC-AUC of 0.87 confirms the model has good discriminatory power, though limited in handling non-linear patterns.

Decision Tree:

|  |
| --- |
| Accuracy : 0.8455 |
| Precision: 0.8438 |
| Recall : 0.9529 |
| F1-score : 0.8950 |
| ROC-AUC : 0.7396 |

A diagram of a decision tree

AI-generated content may be incorrect.A graph of a positive rate

AI-generated content may be incorrect.

***Interpretation:*** The Decision Tree delivered reasonable recall (0.95) but a weaker ROC-AUC (0.74), showing poor generalisation compared to other models. It is useful for interpretability but prone to overfitting.

Random Forest:

|  |
| --- |
| Accuracy : 0.8943 |
| Precision: 0.9091 |
| Recall : 0.9412 |
| F1-score : 0.9249 |
| ROC-AUC : 0.8627 |

A chart with different colored squares

AI-generated content may be incorrect.A graph of a positive rate

AI-generated content may be incorrect.

**Interpretation:** Random Forest achieved the best overall performance with an accuracy of ~89% and the highest F1-score (0.92). Precision and recall were well balanced, while the ROC-AUC (0.86) indicated reliable separability. As an ensemble model, it reduced variance and outperformed single trees.

Gradient Boosting:

|  |
| --- |
| Accuracy : 0.8455 |
| Precision: 0.8587 |
| Recall : 0.9294 |
| F1-score : 0.8927 |
| ROC-AUC : 0.8186 |

A chart with numbers and labels

AI-generated content may be incorrect.A graph of a positive rate

AI-generated content may be incorrect.

***Interpretation:*** Gradient Boosting provided competitive results, with an F1-score of 0.89 and ROC-AUC of 0.82. While slightly weaker than Random Forest, it still demonstrated strong predictive power but required more computational effort.

XGBoost (extra experiment):

|  |
| --- |
| Accuracy : 0.8293 |
| Precision: 0.8636 |
| Recall : 0.8941 |
| F1-score : 0.8786 |
| ROC-AUC : 0.8220 |

A chart with different colored squares

AI-generated content may be incorrect.A graph of a positive result

AI-generated content may be incorrect.

***Interpretation:*** XGBoost performed comparably to Gradient Boosting, with accuracy of ~83% and F1-score of 0.88. ROC-AUC was 0.82, confirming strong but not superior performance relative to Random Forest. Nonetheless, its efficiency and robustness make it a practical choice in real-world applications.

**Why not 95%+?**

* **Dataset size:** Only 614 samples → insufficient to train complex models.
* **Imbalance:** Skewed target distribution → models biased toward “approved” predictions.
* **Dominance of Credit History:** One feature explains much of the variance, limiting the incremental value of others.
* **Missing features:** Critical information such as full credit scores, debt-to-income ratios, or macroeconomic conditions was not included in the dataset.

## **5.2 Influential Features (Q2)**

Across Random Forest, Gradient Boosting, and XGBoost, the most important features were consistent:

* **Credit History** – strongest predictor of loan approval.
* **Loan Amount** – larger loans reduce approval likelihood.
* **Total Income** – higher combined income increases approval chance.
* **EMI Ratio** – affordability indicator; lower ratios are preferable.
* **Property Area** – urban and semi-urban applicants had slightly higher approvals.

Logistic Regression coefficients corroborated these findings, reinforcing confidence in the feature rankings.

## **5.3 Income vs Loan (Q3)**

The relationship between approval rates and income-to-loan ratio was confirmed:

* **Moderate ratios** → highest approvals (repayment capacity seen as realistic).
* **Extremely low ratios** → rejected (insufficient income).
* **Extremely high ratios** → also lower approval (possible data inconsistencies or perceived low demand).

This analysis provides **practical insight**: banks prefer a balance where income is sufficiently high to repay loans but not excessively disproportionate to the loan requested.

## **5.4 Performance Comparison**

To compare the predictive performance of all models, we summarise the evaluation metrics (accuracy, precision, recall, F1-score, and ROC-AUC) in Table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-score** | **ROC-AUC** |
| |  | | --- | | Logistic Regression | | |  | | --- | | 0.8699 | | |  | | --- | | 0.8485 | | |  | | --- | | 0.9882 | | |  | | --- | | 0.9130 | | |  | | --- | | 0.8706 | |
| Decision Tree | |  | | --- | | 0.8455 | | |  | | --- | | 0.8438 | | |  | | --- | | 0.9529 | | |  | | --- | | 0.8950 | | |  | | --- | | 0.7396 | |
| Random Forest | |  | | --- | | **0.8943** | | |  | | --- | | **0.9031** | | |  | | --- | | **0.9412** | | |  | | --- | | **0.9249** | | |  | | --- | | **0.8627** | |
| Gradient Boosting | |  | | --- | | 0.8455 | | |  | | --- | | 0.8587 | | |  | | --- | | 0.9294 | | |  | | --- | | 0.8927 | | |  | | --- | | 0.8186 | |
| XGBoost | |  | | --- | | 0.8293 | | |  | | --- | | 0.8636 | | |  | | --- | | 0.8941 | | |  | | --- | | 0.8786 | | |  | | --- | | 0.8220 | |

**Interpretation:**

* Logistic Regression achieved the **highest recall (0.99)**, meaning it almost never missed an approved loan, but had slightly lower precision.
* Decision Tree offered interpretability but underperformed in ROC-AUC (0.74).
* **Random Forest emerged as the best model overall**, achieving the **highest accuracy (0.89)** and **highest F1-score (0.92)**, with strong precision (0.91) and recall (0.94).
* Gradient Boosting and XGBoost performed competitively but slightly weaker, highlighting the strength of ensemble methods like Random Forest for this dataset.

## **5.5 Additional Research Questions**

* **Q4 – External Features:** Cannot be answered with current dataset. Adding **credit bureau scores, past loan repayment history, and macroeconomic indicators** could significantly improve accuracy.
* **Q5 – Deep Learning Models:** Not attempted due to dataset size. Neural networks require thousands of rows; with only 614, performance would degrade. Future work could explore this once more data is available.

# **6. Conclusion**

This study demonstrates that machine learning can reasonably predict loan approvals and improve decision-making in financial institutions.

* **Random Forest** achieved the highest accuracy (~82–84%), showing that ensemble models are best suited for this dataset.
* **Feature importance analysis** highlighted credit\_history as the most decisive factor, followed by income and loan-related attributes.
* **Approval likelihood** increases with sustainable income-to-loan ratios, but extremely low or high ratios reduce approval chances.
* **Limitations:** Small dataset, missing critical financial attributes, and class imbalance. These factors capped performance below 95%.

## **Business implications:**

* ML-based approval models can improve **profitability** by reducing defaults, enhance **customer service** through faster decisions, and improve **operational efficiency** by reducing manual workload.
* With larger datasets and additional features, accuracy could potentially exceed 90–95%. Future exploration of **deep learning** and **external financial data** would be the next logical step.

# **References (IEEE)**

1. Kaggle, “Loan Prediction Dataset,” *Kaggle.com*, 2024. [Online]. Available: <https://www.kaggle.com/datasets/ninzaami/loan-predication>. Accessed: Sep. 17, 2025.
2. L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
3. J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
4. D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
5. J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.
6. T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD ’16)*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
7. T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
8. F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
9. W. McKinney, *Python for Data Analysis*, 3rd ed. Sebastopol, CA, USA: O’Reilly, 2022.
10. C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020.
11. J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
12. M. Waskom, “Seaborn: Statistical Data Visualization,” *J. Open Source Software*, vol. 6, no. 60, p. 3021, 2021.
13. OpenAI, “ChatGPT (GPT-5 Thinking),” *Large Language Model*, prompts and responses used to assist drafting, coding, and editing the report. [Online]. Available: <https://chat.openai.com/>. Accessed: Sep. 17, 2025.
14. A. H. S. Himalya, L. Kunwar, B. Gautam, and K. Nimasha, “Loan Approval Prediction – Assignment Presentation,” *YouTube*, Sep. 23, 2025. [Online]. Available: <https://youtu.be/93GzVKASp_A?si=HpA3GtVqdqtBekSU> [Accessed: 23-Sep-2025].

# **Contribution Forms**

|  |  |  |
| --- | --- | --- |
| **Name** | **Student ID** | **Contribution (%)** |
| **Alvi Hossain Safri Himalya** | MIT252555 | 25% |
| **Laxman Kunwar** | MIT250221 | 25% |
| **Bishal Gautam** | MIT251594 | 25% |
| **Kithmini Nimasha Ketan Godage** | MIT251345 | 25% |