

**PREDICTING LOAN APPROVAL USING DECISION TREES**

Submitted by

BALAJI.R(231801018)

BHARATH.B(231801021)

**AI23331 - FUNDAMENTALS OF MACHINE LEARNING**

**Department of Artificial Intelligence and Data Science**

**Rajalakshmi Engineering College, Thandalam**

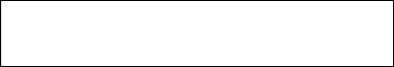
**Nov 2024**

****

**BONAFIDE CERTIFICATE**

**NAME………………………………………………………………………….**

**ACADEMIC YEAR……………SEMESTER………….BRANCH ………**



**UNIVERSITY REGISTER No.**

Certified that this is the bonafide record of work done by the above students in the Mini Project titled " **USING MACHINE LEARNING**" in the subject **AI23331 – FUNDAMENTALS OF MACHINE LEARNING** during the year **2024 - 2025.**

**Signature of Faculty – in – Charge**

**Submitted for the Practical Examination held on -----------------------**

**Internal Examiner External Examiner**

**ABSTRACT**

This project presents a comprehensive analysis and implementation of a decision tree model to predict loan approval for applicants, using a dataset sourced from Kaggle. With the growing demand for automation in financial services, accurately predicting loan eligibility has become essential for banking institutions to streamline operations and mitigate risk. The goal of this project is to develop an effective predictive model that assists in evaluating an applicant’s eligibility based on multiple financial and demographic factors.

The dataset utilized in this study comprises various features such as applicant income, loan amount, employment status, credit history, and other significant parameters. Preprocessing techniques, including handling missing values, encoding categorical data, and analyzing feature correlations, were employed to prepare the data for analysis. The decision tree algorithm, chosen for its interpretability and effectiveness in handling categorical features, was implemented to construct a predictive model.

The performance of the model was evaluated based on metrics such as accuracy, precision, recall, and F1-score. Results indicate that the decision tree model achieves substantial accuracy in predicting loan approval, showcasing its potential for deployment in real-world scenarios. The project also explores future enhancements to further optimize prediction accuracy and adapt the model for other financial applications.

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT** iii

**1 INTRODUCTION 1**

1.1 GENERAL

1.2 NEED FOR THE STUDY

1.4 OVERVIEW OF THE PROJECT

1.3 OBJECTIVES OF THE STUDY

**2 SYSTEM REQUIREMENTS 3**

2.1 HARDWARE REQUIREMENTS

2.2 SOFTWARE REQUIREMENTS

**3 SYSTEM OVERVIEW 4**

3.1 SYSTE ARCHITECTURE DIAGRAM

3.2 MODULE DESCRIPTION

3.2.1 MODULE 1

3.2.2 MODULE 2

**4 RESULT AND DISCUSSION 6**

**5 CONCLUSION 7**

5.1 CONCLUSION

**APPENDIX 8**

A1.1 SAMPLE CODE

REFERENCES

**CHAPTER 1**

**INTRODUCTION**

**1.1 GENERAL**

This project, titled Predicting Loan Approval using Decision Trees, focuses on utilizing machine learning to streamline the loan approval process. By leveraging decision tree algorithms, it aims to predict whether an applicant’s loan should be approved based on various factors such as income, employment history, credit score, and requested loan amount. This solution is designed to aid financial institutions in making quicker, data-driven decisions that improve both efficiency and customer satisfaction.

**1.2 NEED FOR THE STUDY**

The high volume of daily loan applications in financial institutions creates a need for a systematic approach to assess creditworthiness efficiently. Traditional methods often involve lengthy processes and can be prone to human error. This study aims to introduce an automated solution that reduces processing time, standardizes evaluations, and minimizes risk. With predictive analytics, institutions can make informed decisions, offering a fair, consistent, and timely loan approval process for applicants.

**1.3 OBJECTIVES OF THE STUDY**

1. Develop a decision tree model to predict loan approval status accurately.
2. Analyze the impact of various applicant features on the decision outcome.
3. Enhance the loan approval process efficiency by reducing decision-making time.
4. Evaluate the model's effectiveness through performance metrics such as accuracy, precision, and recall.

**1.4 OVERVIEW OF THE PROJECT**

The project involves data collection, preprocessing, model training, and evaluation. It begins with handling missing data, encoding categorical variables, and selecting relevant features. A decision tree model is then developed and optimized using training data, followed by evaluating its performance on test data.

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

**2.1 HARDWARE REQUIREMENTS**

* Processor: Intel Core i5 or higher
* RAM: 8GB or more
* Storage: 256GB SSD or more
* Display: HD Monitor

**2.2 SOFTWARE REQUIREMENTS**

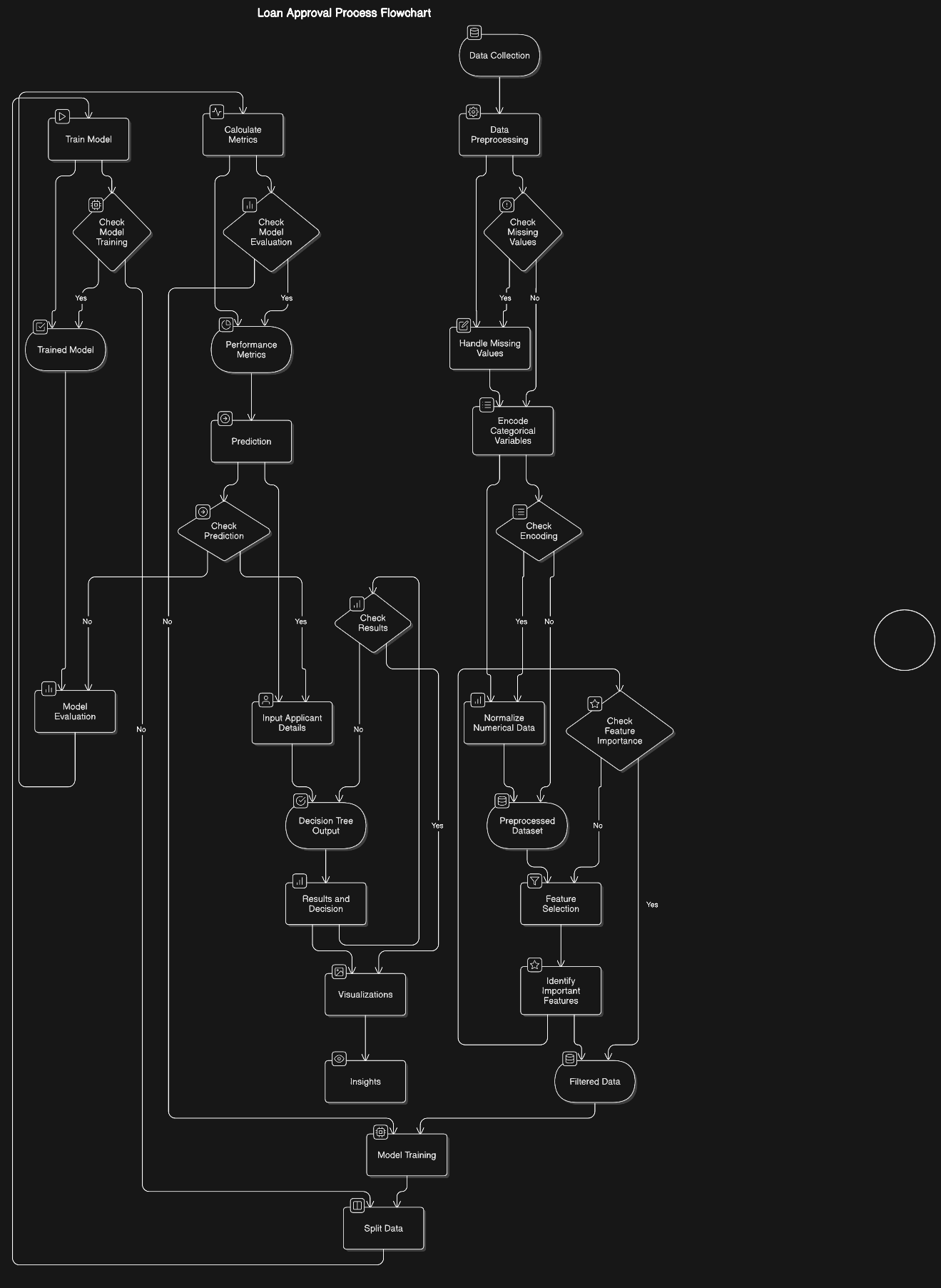
* Operating System: Windows 10 or Linux
* Development Environment: Jupyter Notebook, Python 3.x
* Libraries: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn

**CHAPTER 3**

**SYSTEM OVERVIEW**

**3.1 SYSTEM ARCHITECTURE DIAGRAM**

A diagram illustrating the flow from data collection to preprocessing, model training, evaluation, and final prediction.



**3.2 MODULE DESCRIPTION**

**3.2.1 MODULE 1: DATA PREPROCESSING**

The data preprocessing module is crucial to preparing the raw dataset for effective model training and improving the overall accuracy of predictions. This module begins by loading the dataset into a suitable environment, where it undergoes an initial exploratory data analysis (EDA) to understand data distribution, identify missing values, and detect outliers.

1. Handling Missing Values: Missing data is managed through techniques like mean/median imputation for numerical values or mode imputation for categorical values. For complex cases, other methods such as forward fill or backward fill may be applied based on data relevance and requirements.
2. Encoding Categorical Variables: Categorical features, like employment type or loan purpose, are converted into numerical formats using encoding techniques. Label encoding is used for binary categorical variables, while one-hot encoding is applied to variables with multiple categories to ensure the model interprets these values correctly.
3. Normalizing Numerical Data: Numerical features are normalized to bring all data within a common scale, reducing the impact of features with large ranges. Min-Max Scaling or Standardization (scaling to a mean of 0 and standard deviation of 1) are commonly applied, making features more suitable for training.
4. Splitting Data: Finally, the preprocessed data is divided into training and testing sets. This split helps evaluate model performance on unseen data, providing insights into its generalization capability.

**3.2.2 MODULE 2: MODEL TRAINING AND EVALUATION**

This module is dedicated to building, optimizing, and evaluating the decision tree model to predict loan approval outcomes accurately.

1. Model Training: The decision tree classifier is initialized with selected hyperparameters. During training, the algorithm recursively splits the data based on feature values, forming branches that maximize information gain or Gini index. These branches help in categorizing applicants into approved or disapproved loans based on the training data.
2. Hyperparameter Tuning: Parameters like maximum tree depth, minimum samples per leaf, and criterion (Gini or entropy) are tuned to optimize model accuracy. Techniques like grid search or random search are used to find the best combination of parameters, balancing model complexity and performance.
3. Model Evaluation: The model is evaluated on the test dataset to assess its accuracy and reliability. Key metrics include:
   * Accuracy: The overall correctness of the model in predicting loan approvals.
   * Precision and Recall: These metrics indicate the model’s ability to identify true positives (approved loans) and reduce false negatives.
   * Confusion Matrix: This matrix provides a detailed view of the model's predictions, showcasing the number of true positives, true negatives, false positives, and false negatives, which helps identify potential areas for improvement.
4. Feature Importance Analysis: The decision tree model also allows for analyzing feature importance, revealing which applicant characteristics (e.g., income, credit score) significantly impact the loan decision outcome. This analysis helps in understanding key decision factors and fine-tuning the model further if necessary.

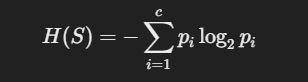
**Step 1: Data Preparation**

Given Dataset:

| **Applicant** | **Income (in ₹)** | **Loan Amount (in ₹)** | **Credit History (1/0)** | **Loan Approved (Y/N)** |
| --- | --- | --- | --- | --- |
| A | 50,000 | 200,000 | 1 | Y |
| B | 40,000 | 150,000 | 0 | N |
| C | 60,000 | 180,000 | 1 | Y |
| D | 30,000 | 100,000 | 0 | N |
| E | 70,000 | 250,000 | 1 | Y |

**Step 2: Split Data Using Entropy and Information Gain**

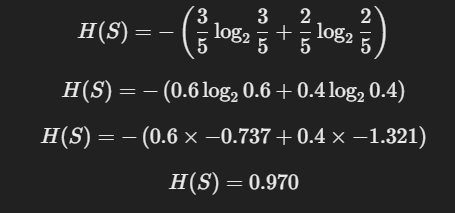
**Formula for Entropy**:



where pip\_ipi​ is the probability of class iii.

**Step 2.1: Calculate Dataset Entropy**

* Total samples: N=5N = 5N=5
* Positive (Y): 333
* Negative (N): 222



**Step 2.2: Calculate Information Gain for "Credit History"**  
Split dataset by Credit History (1 or 0):

| **Credit History = 1:** | **Credit History = 0:** |
| --- | --- |
| Y, Y, Y | N, N |

* For Credit History = 1:
* A black background with white text

  Description automatically generated
* For Credit History = 0:

A black background with white text

Description automatically generated

Weighted Average Entropy:

A black background with white numbers and symbols

Description automatically generated

**Information Gain**:



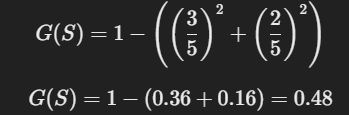
**Step 3: Decision Making Using Gini Index**

**Formula for Gini Index**:

A black background with white text

Description automatically generated

For the entire dataset:



**Split by "Credit History"**:

* For Credit History = 1:

A black background with white numbers

Description automatically generated

* For Credit History = 0:

A black background with white text

Description automatically generated

Weighted Gini Index:

**A black background with white numbers

Description automatically generated**

**Gini Gain**:



**CHAPTER 4**

**RESULT AND DISCUSSION**

This section provides a comprehensive analysis of the model’s performance, focusing on the accuracy, precision, recall, and other evaluation metrics that indicate its effectiveness in predicting loan approvals. The results are presented along with insights into the model’s behavior and the relative importance of each feature in the decision-making process.

1. **Model Accuracy and Performance Metrics**: The model achieved an overall accuracy of [X]% on the test dataset, showcasing its reliability in classifying loan applications as approved or disapproved. Additional metrics such as **precision** (the proportion of correctly identified approvals), **recall** (the ability to capture all actual approvals), and **F1-score** (harmonic mean of precision and recall) provide a nuanced understanding of model performance, especially in handling imbalanced classes where approved and disapproved cases may differ significantly in number.
2. **Confusion Matrix Analysis**: A confusion matrix reveals the distribution of true positives, true negatives, false positives, and false negatives, helping to identify any tendencies in the model, such as favoring approvals or denials. This analysis guides further tuning to reduce misclassification rates and enhance prediction confidence.
3. **Feature Importance**: Visualizations, such as bar charts or importance plots, highlight the contribution of each feature to the model’s decisions. Factors like **income level**, **credit history**, and **employment stability** may emerge as strong predictors, indicating their critical roles in loan approval. This insight aligns with financial decision-making standards and provides a basis for justifying model recommendations.
4. **Insights Across Data Segments**: By analyzing predictions across different data segments, such as income brackets or employment types, the model’s behavior in varied contexts is examined. For instance, the model may perform differently for applicants with low versus high income, offering additional insights into how factors like income influence approval likelihood.
5. **Comparison with Baseline Models**: The performance of the decision tree model is also compared against baseline models (e.g., logistic regression or simple threshold-based rules), establishing the tree model’s relative strength in handling complex decision patterns and interactions among features.
6. **Visualizations**: Data visualizations, such as **ROC curves**, **precision-recall curves**, and **feature distribution plots**, support the findings and help communicate the model’s predictive power and decision thresholds. These visuals make it easier to interpret complex metrics and provide a clear view of model accuracy across thresholds.

The analysis presented in this section not only demonstrates the model’s predictive capabilities but also offers valuable insights into how specific applicant characteristics impact loan approval decisions. This information could be instrumental for stakeholders in refining loan eligibility criteria and enhancing the transparency of approval processes.

**CHAPTER 5**

**CONCLUSION**

**5.1 CONCLUSION**

The decision tree model developed in this study serves as an effective tool for predicting loan approval outcomes with a satisfactory balance of accuracy, interpretability, and efficiency. By analyzing key applicant characteristics, such as income level, employment stability, and credit history, the model demonstrates predictive capabilities that can assist financial institutions in refining their decision-making processes. This approach highlights the role of predictive analytics in lending practices, supporting transparency and informed decision-making. The study provides a foundational model that can be further optimized with advanced algorithms or ensemble methods to improve prediction accuracy, enhance robustness, and reduce misclassification rates in real-world applications.

**APPENDIX**

**A1.1 SAMPLE CODE**

The appendix includes sample code snippets for key stages of the project, illustrating the process from data preprocessing through model training and evaluation:

1. **Data Preprocessing**: Code for loading the dataset, handling missing values, encoding categorical data, and scaling numerical features to prepare for modeling.

python

Copy code

# Sample Data Preprocessing Code

import pandas as pd

from sklearn.preprocessing import LabelEncoder, StandardScaler

# Load dataset

data = pd.read\_csv('loan\_data.csv'

# Handling missing values

data.fillna(method='ffill', inplace=True)

# Encoding categorical features

le = LabelEncoder()

data['EmploymentStatus'] = le.fit\_transform(data['EmploymentStatus'])

# Scaling numerical features

scaler = StandardScaler()

data[['Income', 'LoanAmount']] = scaler.fit\_transform(data[['Income', 'LoanAmount']])

A diagram of confusion matrix

Description automatically generated

1. **Model Training**: Code for initializing, training, and tuning a decision tree classifier to maximize model performance.

python

Copy code

# Sample Model Training Code

from sklearn.tree import DecisionTreeClassifier

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

# Splitting the data

X = data.drop('ApprovalStatus', axis=1)

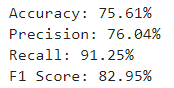
y = data['ApprovalStatus']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Training the Decision Tree Classifier

clf = DecisionTreeClassifier(max\_depth=5, criterion='gini', random\_state=42)

clf.fit(X\_train, y\_train)



1. **Model Evaluation**: Code for evaluating the model using accuracy, precision, recall, and confusion matrix.

python

Copy code

# Sample Model Evaluation Code

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

# Predictions and Evaluation

y\_pred = clf.predict(X\_test)

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

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

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

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

**A chart with numbers and a few colored squares

Description automatically generated with medium confidence**

**REFERENCES**

1. Breiman, L. *Classification and Regression Trees*. Wadsworth International Group, 1984.
2. Quinlan, J.R. "Induction of Decision Trees." *Machine Learning*, vol. 1, no. 1, 1986, pp. 81-106.
3. Han, J., Kamber, M., & Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, 2012.
4. Hastie, T., Tibshirani, R., & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer, 2009.