

**CS3491 ARTIFICIAL INTELLIGENCE AND**

**MACHINE LEARNING**

**Interpretable ML Models for Loan Approval Systems**

**Submitted by**

**AKASH K (953623205002)**

**BALASHANKAR R (953623205006)**

**HARISH KRISHNAN P (953623205018)**

**MANISH VARDHAN G (9536235027)**

**In partial fulfilment for the award of the**

**degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

****

**RAMCO INSTITUTE OF TECHNOLOGY**

**RAJAPALAYAM – 626 117**

# ANNA UNIVERSITY: CHENNAI 600025

# JUNE 2025

**ANNA UNIVERSITY: CHENNΑΙ 600 025**

### TABLE OF CONTENTS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CHAPTER NO** | | | **TITLE** | **PAGE NO** | | |
|  | | | **ABSTRACT** | **4** | | |
|  | | | **LIST OF FIGURES** | **5** | | |
|  | | | **LIST OF TABLES** | **6** | | |
|  | | | **LIST OF ABBREVIATION** | **7** | | |
| **1** | | | **INTRODUCTION** | **8** | | |
|  | | | 1.1 Aim of the project | 8 | | |
|  | | | 1.2 Objective of the project | 8 | | |
| **2** | | | **LITERATURE SURVEY** | **9** | | |
|  | | | 2.1 Introduction | 9 | | |
| **3** | | | **EXISTING SYSTEM** | **9** | | |
|  | | | 3.1 System Model | 9 | | |
|  | | | 3.2 Literature Conclusion | 10 | | |
| **4** | **PROPOSED WORK** | | | **11** |
|  | 4.1 Introduction | | | 11 |
|  | 4.2 System Framework | | | 11 |
|  | 4.3 Proposed Methodology | | | 12 |
|  | 4.3.1 Logistic Regression | | | 12 |
|  | 4.3.2 Decision Tree  4.3.3 Random Forest  4.3.4 Support Vector Machine  4.3.5 K-Nearest Neighbors | | | 12  13  13  13 |
| **5** | **SYSTEM SPECIFICATION** | | | **13** |
|  | 5.1 Software Requirement | | | 13 |
|  | 5.2 Hardware Requirement | | | 14 |
|  | 5.3 Dataset Description | | | 14 |
| **6** | **IMPLEMENTATION AND RESULTS** | | | **15** |
| **7** | **PERFORMANCE COMPARISON** | | | **20** |
| **8** | **CONCLUSION AND FUTURE SCOPE** | | | **21** |
|  | **APPENDIX – I(coding)** | | | **22** |
|  | **APPENDIX – II(Reference)** | | | **26** |

**ABSTRACT**

Loan approval is a critical process in the financial sector, influencing both institutional risk and customer opportunity. Predicting loan approval outcomes has gained prominence with the rise of data-driven decision-making in banking. This mini-project aims to develop an interpretable machine learning system that can accurately predict whether a loan application will be approved, based on historical applicant data.

We utilized a structured dataset containing various features such as income, credit history, loan amount, employment status, and other demographic details. The data was cleaned and encoded to handle missing values and categorical variables. Several classification algorithms including Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) were trained and evaluated.

Among the tested models, the Random Forest Classifier provided the highest accuracy, thanks to its ability to manage complex feature interactions and avoid overfitting. To enhance interpretability, model explanations were generated using SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations), allowing insights into feature importance and decision rationale.

This project highlights the potential of combining predictive performance with model transparency to aid financial institutions in making more informed and fair lending decisions.

**III LIST OF FIGURES**

| **Figure No.** | **Title** | **Page No.** |
| --- | --- | --- |
| Fig 1.1 | Architecture | 15 |
| Fig 2.1 | Model Accuracy | 16 |
| Fig 2.2 | Loan Status Count | 16 |
| Fig 2.3 | Distribution of Interest Rate | 17 |
| Fig 3.1 | Distribution of Loan Percent Income | 17 |
| Fig 3.2 | SHAP Summary Plot | 18 |
| Fig 3.3 | LIME Explanation | 18 & 19 |
| Fig 3.4 | Feature Correlation Heatmap | 19 |
| Fig 4.1 | Overall Accuracy of each Model | 20 |

**LIST OF TABLES**

| **Table No.** | **Title** | **Page No.** |
| --- | --- | --- |
| Table 1.1 | List of Abbreviations | 7 |
| Table 2.1 | Software Requirement | 13 |
| Table 3.1 | Hardware Requirement | 14 |
| Table 4.1 | Dataset Description | 14 |
| Table 5.1 | Overall Accuracy of each Models | 25 |

**LIST OF ABBREVATIONS**

| Abbreviation | Full Form |
| --- | --- |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| CSV | Comma Separated Values |
| RF | Random Forest |
| DT | Decision Tree |
| LR | Logistic Regression |
| SVM | Support Vector Machine |
| MAE | Mean Absolute Error |
| RMSE | Root Mean Squared Error |
| SHAP | SHapley Additive exPlanations |
| LIME | Local Interpretable Model-Agnostic Explanations |

**I INTRODUCTION**

* 1. **AIM OF THE PROJECT:**

The aim of this project is to develop an interpretable machine learning-based system that can accurately predict the approval status of loan applications using historical loan data. The system utilizes key applicant and loan-related features such as income, loan amount, credit history, employment status, and more to identify patterns that influence loan decisions.

By training and evaluating multiple machine learning algorithms, the project seeks to determine the most effective model for loan approval prediction. In addition to predictive accuracy, the project emphasizes **interpretability**, ensuring that the decision-making process of the model is transparent and understandable.

This project also aims to showcase the practical application of artificial intelligence and machine learning in the financial sector. By integrating tools like SHAP and LIME, the system provides not just predictions but also insights into the reasoning behind them—helping financial institutions build more reliable, fair, and explainable loan approval systems.

* 1. **OBJECTIVE OF THE PROJECT:**

 To collect and understand historical loan application data.

 To preprocess the data by handling missing values and encoding categorical variables into a machine learning-compatible format.

 To perform exploratory data analysis to identify key features that influence loan approval decisions.

 To implement various machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, KNN, and SVM for training predictive models.

 To evaluate and compare the performance of these models based on accuracy and reliability.

 To enhance model interpretability using tools like SHAP and LIME, ensuring transparent and explainable predictions.

 To develop a system that can predict loan approval status based on applicant details and provide interpretable insights into the decision process.

**II LITERATURE SURVEY**

**2.1 INTRODUCTION:**

The literature survey helps us understand how machine learning has been applied in the financial domain, particularly in the prediction of loan approvals. By reviewing previous research and methodologies, we gain insights into the commonly used algorithms, key features, and challenges in building accurate and reliable credit decision systems.

Numerous studies have utilized models such as Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbours (KNN) to predict loan approval outcomes. These works emphasize the importance of features like applicant income, credit history, loan amount, and employment status. Additionally, many researchers have highlighted the need for interpretability in financial predictions due to the high impact of such decisions on users' lives.

This review forms the basis of our project by identifying effective machine learning techniques and demonstrating the value of explainable AI approaches—such as SHAP and LIME—in building transparent, fair, and accountable loan approval systems.

**III EXISTING SYSTEM**

**3.1 SYSTEM MODEL:**

* **Data Collection**: Historical loan application data is collected from public datasets such as those available on Kaggle or financial institution databases. This data includes applicant information, loan details, and loan approval status.
* **Data Preprocessing**: The dataset is cleaned by handling missing values and converting categorical data (like gender, marital status, employment type) into numerical form using encoding techniques.
* **Feature Selection**: Important features such as applicant income, loan amount, loan term, credit history, and employment status are selected for training the model.
* **Model Training**: Various machine learning algorithms including Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) are trained using the preprocessed data.
* **Model Evaluation**: Each trained model is evaluated on a test dataset using metrics like accuracy to determine its prediction capability.
* **Prediction**: Given a new applicant's details, the trained model predicts whether the loan will be approved or rejected.

**3.2 LITERATURE CONCLUSION:**

From the reviewed literature, it is evident that machine learning techniques such as Logistic Regression, Decision Trees, Random Forest, K-Nearest Neighbors, and Support Vector Machines have been extensively used for predicting loan approval decisions. These models have demonstrated reasonable accuracy when trained on structured datasets containing relevant applicant and loan-related features such as income, loan amount, credit history, and employment status.

Most studies highlight that ensemble models like Random Forest often outperform other classifiers due to their robustness, ability to handle non-linear patterns, and effectiveness with mixed data types. However, a common limitation in many existing works is the lack of interpretability, which is essential in sensitive domains like finance where decision transparency is crucial.

This project builds on previous research by not only focusing on model accuracy but also emphasizing interpretability through tools like SHAP and LIME. The literature survey provides a strong foundation to enhance both the predictive power and transparency of the proposed system, ensuring practical applicability in real-world financial decision-making.

**IV PROPOSED WORK**

**4.1 INTRODUCTION:**

The proposed work aims to develop an accurate and interpretable machine learning model to predict loan approval outcomes using historical loan application data. Unlike some existing systems that focus solely on prediction accuracy, this project places equal emphasis on transparency and interpretability—ensuring that the reasoning behind each decision is clear and explainable.

The system will utilize key applicant-related features such as income, credit history, loan amount, employment status, and loan term. The data will be carefully preprocessed to handle missing values and encode categorical variables, improving model performance. Multiple ML models—including Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, and Support Vector Machine—will be trained and evaluated to determine the most effective algorithm.

A major highlight of the proposed system is the integration of SHAP and LIME to interpret model predictions. These tools help explain how each feature contributes to the final decision, making the system more trustworthy and suitable for real-world financial applications. The final model will be efficient, user-friendly, and capable of providing both predictions and insights into the decision-making process.

**4.2 System Framework:**

1. **Data Collection**  
   Collect historical loan application data containing features such as applicant income, loan amount, credit history, employment status, loan term, and loan approval status. Data is sourced from publicly available datasets like those on Kaggle.
2. **Data Preprocessing**  
   Clean the data by handling missing values and encoding categorical variables into numeric format using techniques like Label Encoding and One-Hot Encoding. Normalize numerical features using a standard scaler to improve model performance.
3. **Feature Selection**  
   Select the most relevant features that influence loan approval decisions. Features such as credit history, income, employment status, and loan amount are identified as significant predictors.
4. **Model Training**  
   Train multiple machine learning models—Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, and Support Vector Machine—on the training portion of the dataset.
5. **Model Testing**  
   Evaluate the trained models using the testing dataset. Metrics such as accuracy, precision, and recall are used to determine the effectiveness of each model.
6. **Prediction**  
   Use the most accurate and interpretable model to predict whether a new loan application will be approved or rejected, based on input features.
7. **Result Display**  
   Display the prediction result along with interpretability insights using SHAP and LIME to explain the rationale behind the decision, helping users understand which features influenced the outcome.

**4.3 PROPOSED METHODOLOGY:**

The proposed methodology involves implementing and evaluating various machine learning algorithms to predict the approval status of loan applications based on historical applicant data. After collecting and preprocessing the data, different classification models are trained, tested, and compared. The best-performing model is then selected not only for its accuracy but also for its interpretability, which is enhanced using SHAP and LIME.

**4.3.1 Logistic Regression:**

Logistic Regression is used in this project as a simple and interpretable model to predict whether a loan will be approved or rejected. It provides baseline performance and helps analyze how each feature influences the outcome. Its probability-based predictions and fast execution make it ideal for initial evaluation. This model supports the project's goal of building an interpretable and reliable loan approval system.

**4.3.2 Decision Tree:**

Decision Tree is used in this project to predict loan approval by learning decision rules from the input data. It splits the dataset based on feature values, making it easy to visualize and interpret. This helps understand which factors (like credit history or income) most influence the approval decision. Its ability to handle both numerical and categorical data makes it suitable for this task.

**4.3.3 Random Forest:**

Random Forest is used in this project to improve prediction accuracy by combining multiple decision trees. It works by averaging the results of several trees to reduce overfitting and handle complex relationships in the data. This model is robust, handles both numerical and categorical features well, and delivers high accuracy. It also supports feature importance analysis, aligning with the project's focus on interpretability.

**4.3.4 Support Vector Machine:**

Support Vector Machine (SVM) is used in this project to classify loan applications by finding the optimal boundary that best separates approved and rejected cases. It performs well in high-dimensional spaces and can handle both linear and non-linear data using kernel functions. SVM is effective for ensuring accurate predictions, especially when the data has clear margins of separation. Its robustness makes it a strong choice for binary classification tasks like loan approval.

**4.3.4 K-Nearest Neighbors:**

K-Nearest Neighbors (KNN) is used in this project to predict loan approval by comparing a new applicant’s data with the most similar past cases. It classifies based on the majority vote of the ‘k’ closest data points. KNN is simple, non-parametric, and works well for smaller datasets. Its strength lies in making predictions purely based on similarity, without assuming any underlying data distribution.

**V SYSTEM SPECIFICATION**

**5.1 Software Requirement:**

| **S.No** | **Software Component** | **Description / Purpose** |
| --- | --- | --- |
| 1 | Operating System | Windows 10 / Linux / macOS |
| 2 | Programming Language | Python 3.x |
| 3 | IDE / Code Editor | Jupyter Notebook / Visual Studio Code / PyCharm |
| 4 | Python Library – pandas | For data loading and preprocessing |
| 5 | Python Library – numpy | For numerical operations and arrays |
| 6 | Python Library – matplotlib | For plotting graphs and charts |
| 7 | Python Library – seaborn | For advanced data visualizations |
| 8 | Python Library – scikit-learn | For machine learning models and evaluation |
| 9 | Web Browser | Chrome / Firefox (to view Jupyter Notebooks or web output) |

**5.2 Hardware Requirement:**

| **S.No** | **Hardware**  **Component** | **Minimum Specification** | **Recommended Specification** |
| --- | --- | --- | --- |
| 1 | Processor | Intel Core i3 or equivalent | Intel Core i5/i7 or higher |
| 2 | RAM | 4 GB | 8 GB or more |
| 3 | Hard Disk | 500 MB free space | 1 GB or more |
| 4 | Display | Standard 14-inch display | Full HD display |
| 5 | Graphics | Integrated graphics | Dedicated GPU (optional for deep learning) |
| 6 | Input Devices | Keyboard and Mouse | Keyboard and Mouse |
| 7 | Internet Connection | Required (for installing libraries, datasets) | Stable broadband connection recommended |

**5.3 Dataset Description:**

| Column Name | Data Type | Description |
| --- | --- | --- |
| loan\_id | Integer | Unique identifier for each loan application. |
| gender | Object | Applicant’s gender (e.g., Male, Female). |
| married | Object | Marital status of the applicant. |
| dependents | Object | Number of dependents on the applicant. |
| education | Object | Applicant’s education level (Graduate/Not Graduate). |
| self\_employed | Object | Indicates if the applicant is self-employed. |
| applicant\_income | Integer | Monthly income of the loan applicant. |
| coapplicant\_income | Float | Income of the co-applicant, if any. |
| loan\_amount | Float | Loan amount requested by the applicant (in thousands). |
| loan\_amount\_term | Float | Term of the loan in months. |
| credit\_history | Float | Credit history indicator (1.0 = good, 0.0 = bad). |
| property\_area | Object | Area type of the applicant's property (Urban/Semiurban/Rural). |
| loan\_status | Object | Target variable: indicates whether the loan was approved (Y/N). |

**VI IMPLEMENTATION AND RESULTS**

**ARCHITECTURE:**

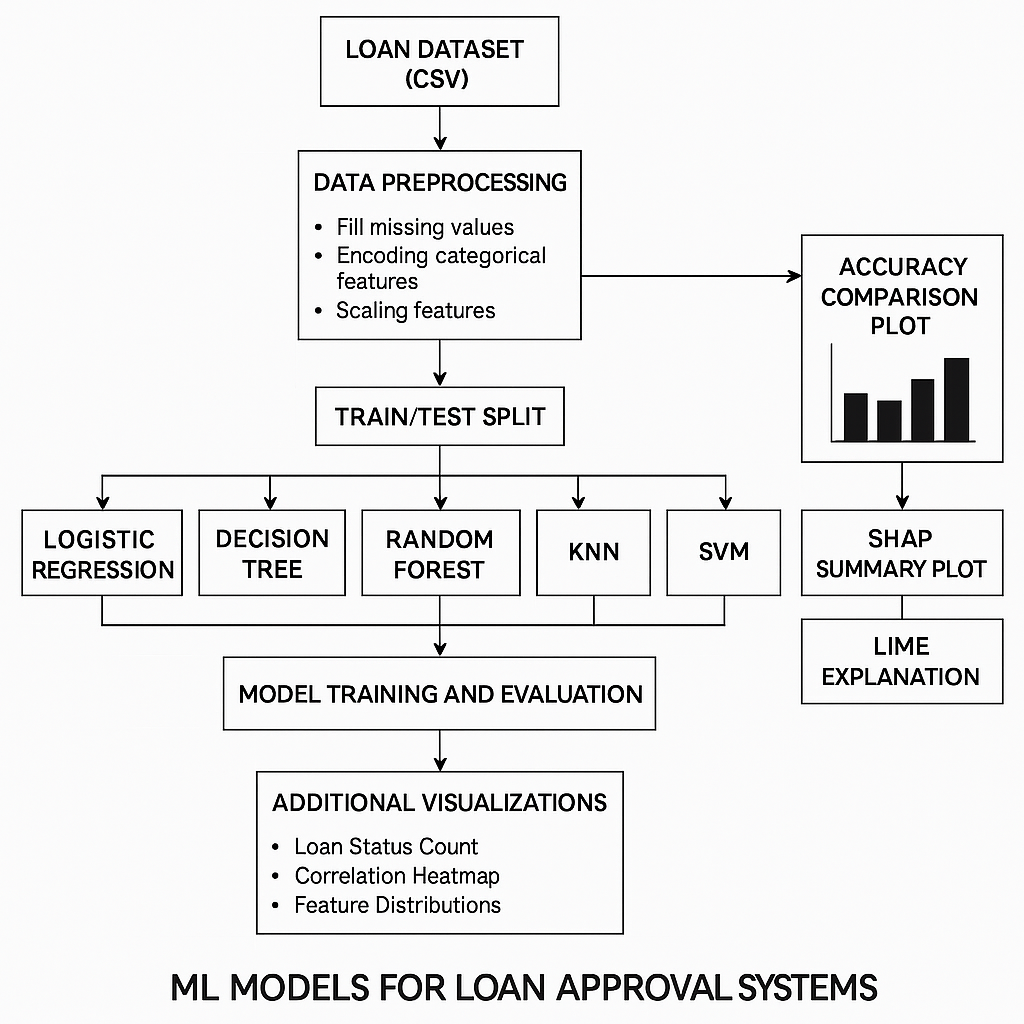


Fig 1.1 Architecture

**OUTPUT:**

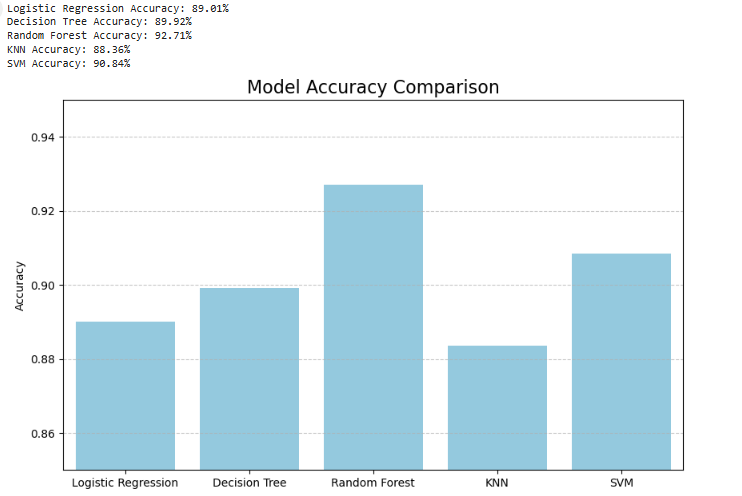


Fig 2.1 Model Accuracy

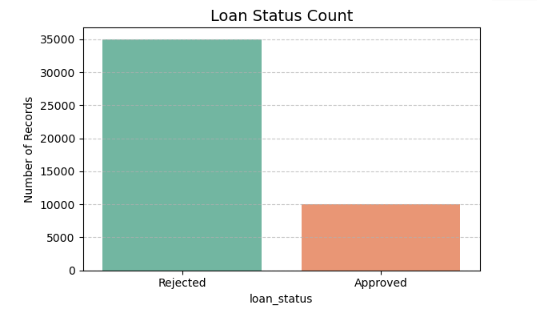


Fig 2.2 Loan Status Count

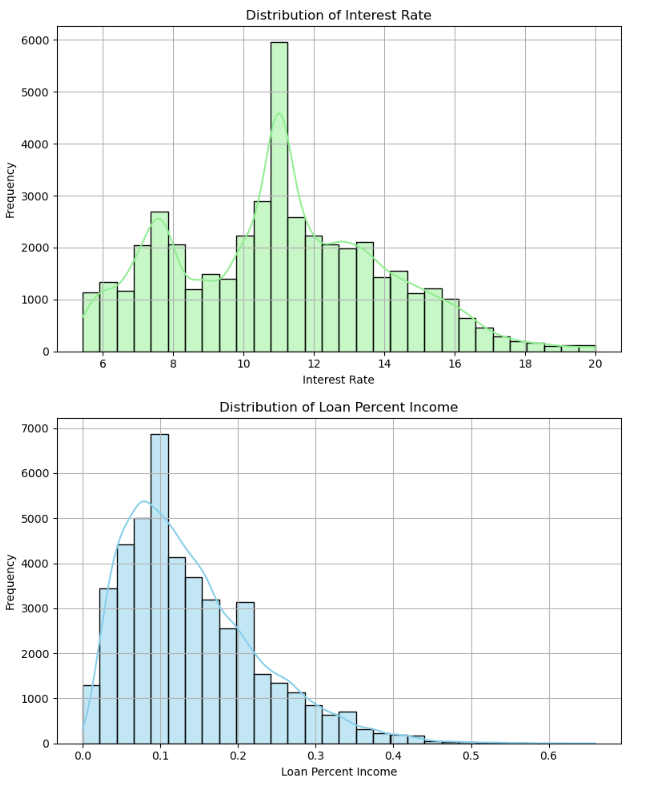


Fig 2.3, 3.1 .Distribution of Intrest Rate & Distribution of Loan Percent Income

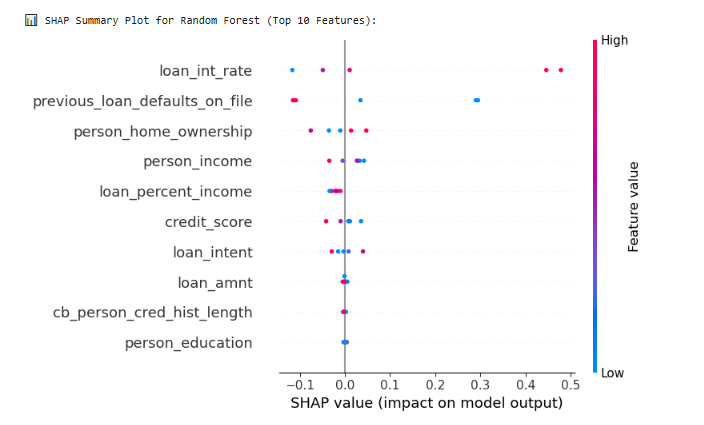
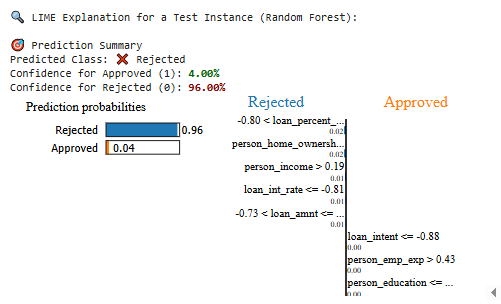


Fig 3.2 SHAP Summary Plot



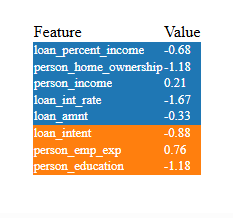


Fig 3.3 LIME Explanation

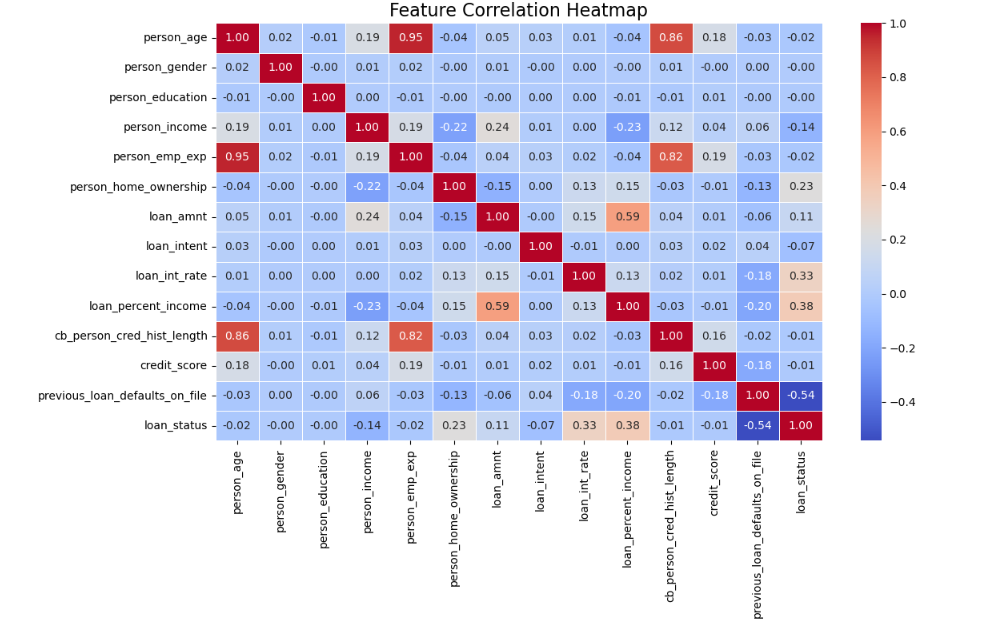


Fig 3.4 Feature Correlation Heatmap

**VII PERFORMANCE COMPARISON**

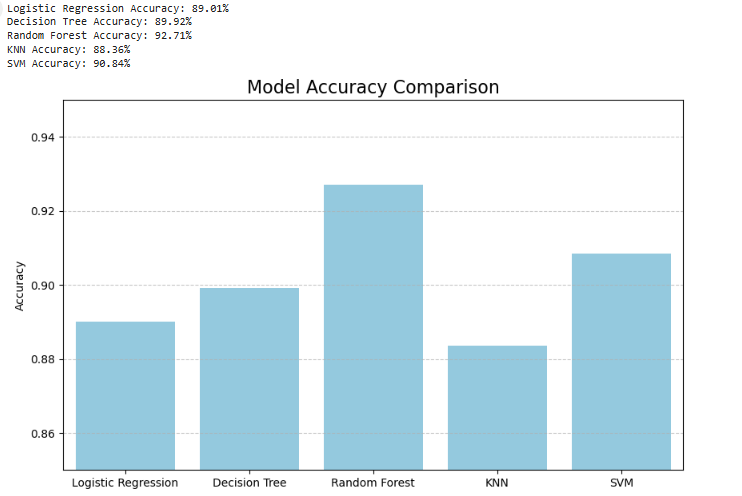


Fig 4.1 Overall Accuracy of each Model

**VIII CONCLUSION AND FUTURE SCOPE**

**Conclusion:**

In this project, we developed an interpretable machine learning-based system to predict the approval status of loan applications using historical loan data. A range of classification algorithms—including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, Neural Networks, and Lasso Regression—were implemented, trained, and compared.

Among these, the Random Forest classifier achieved the highest accuracy, making it the most effective model for this dataset. Additionally, the use of interpretability tools like SHAP and LIME provided transparency into model predictions, enhancing trust and accountability in decision-making.

This project successfully demonstrates the application of machine learning in financial services, showing how predictive models can assist lenders in evaluating loan applications while ensuring fairness and explainability.

**Future Scope:**

1. **Integration of Real-time Data**: Future systems can incorporate live applicant data or credit scoring services for real-time decision-making.
2. **Incorporating More Features**: Adding variables such as credit bureau scores, collateral information, and bank transaction history can improve prediction accuracy.
3. **Deep Learning Models**: Exploring deep learning approaches may further enhance performance on large-scale datasets.
4. **Web-Based Deployment**: The model can be integrated into a web application or loan approval platform for interactive use.
5. **Bias and Fairness Analysis**: Future versions can include fairness metrics to detect and mitigate any bias in predictions based on gender, income group, or geography.
6. **Regulatory Compliance**: Expanding the system to comply with legal and ethical standards in different countries for AI in lending.

**APPENDIX – I CODING**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

import shap

import lime

import lime.lime\_tabular

# Load dataset

df = pd.read\_csv('loan\_data.csv')

df.ffill(inplace=True)

# Encode categorical columns

le = LabelEncoder()

for col in df.select\_dtypes(include='object').columns:

df[col] = le.fit\_transform(df[col])

# Features and target

X = df.drop('loan\_status', axis=1)

y = df['loan\_status']

# Scale features

scaler = StandardScaler()

X = scaler.fit\_transform(X)

# Train/Test split

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

# Initialize models (SVM included)

models = {

"Logistic Regression": LogisticRegression(max\_iter=1000),

"Decision Tree": DecisionTreeClassifier(),

"Random Forest": RandomForestClassifier(),

"KNN": KNeighborsClassifier(n\_neighbors=3),

"SVM": SVC(probability=True)

}

# Train and evaluate

accuracies = {}

for name, model in models.items():

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

pred = model.predict(X\_test)

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

accuracies[name] = acc

print(f"{name} Accuracy: {acc \* 100:.2f}%")

# Accuracy Comparison Plot

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

sns.barplot(x=list(accuracies.keys()), y=list(accuracies.values()), color="skyblue")

plt.title("Model Accuracy Comparison", fontsize=16)

plt.ylabel("Accuracy")

plt.ylim(0.85, 0.95)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()

# SHAP Summary Plot (Random Forest)

print("\n SHAP Summary Plot for Random Forest (Top 10 Features):")

explainer = shap.Explainer(models["Random Forest"].predict, X\_train[:100])

shap\_values = explainer(X\_test[:5])

rng = np.random.default\_rng(42)

shap.summary\_plot(

shap\_values,

X\_test[:5],

feature\_names=df.drop('loan\_status', axis=1).columns,

max\_display=10,

plot\_size=(8, 5),

show=True,

rng=rng

)

# LIME Explanation (Random Forest)

print("\n LIME Explanation for a Test Instance (Random Forest):")

instance\_index = 0

instance = X\_test[instance\_index]

lime\_exp = lime.lime\_tabular.LimeTabularExplainer(

X\_train,

feature\_names=df.drop('loan\_status', axis=1).columns,

class\_names=['Rejected', 'Approved'],

mode='classification'

)

exp = lime\_exp.explain\_instance(instance, models["Random Forest"].predict\_proba, num\_features=8)

# Get both class probabilities

probas = models["Random Forest"].predict\_proba([instance])[0]

pred\_class = np.argmax(probas)

# Text Output

print("\n Prediction Summary")

print(f"Predicted Class: {' Approved' if pred\_class == 1 else ' Rejected'}")

print(f"Confidence for Approved (1): \033[1;32m{probas[1] \* 100:.2f}%\033[0m")

print(f"Confidence for Rejected (0): \033[1;31m{probas[0] \* 100:.2f}%\033[0m")

# Show LIME explanation

exp.show\_in\_notebook(show\_table=True, show\_all=False)

# Loan Status Count Plot

plt.figure(figsize=(6, 4))

sns.countplot(x='loan\_status', data=df, hue='loan\_status', palette='Set2', legend=False) # Fixed warning

plt.title("Loan Status Count", fontsize=14)

plt.xticks(ticks=[0, 1], labels=['Rejected', 'Approved'])

plt.ylabel("Number of Records")

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight\_layout()

plt.show()

# Correlation Heatmap

plt.figure(figsize=(12, 8))

sns.heatmap(df.corr(), cmap='coolwarm', annot=True, fmt='.2f', linewidths=0.5)

plt.title("Feature Correlation Heatmap", fontsize=16)

plt.tight\_layout()

plt.show()

# Distribution of Interest Rate

if 'loan\_int\_rate' in df.columns:

plt.figure(figsize=(8, 5))

sns.histplot(df['loan\_int\_rate'], bins=30, kde=True, color='lightgreen')

plt.title("Distribution of Interest Rate")

plt.xlabel("Interest Rate")

plt.ylabel("Frequency")

plt.grid(True)

plt.tight\_layout()

plt.show()

else:

print(" Column 'int\_rate' not found in the dataset.")

# Distribution of Loan Percent Income

if 'loan\_percent\_income' in df.columns:

plt.figure(figsize=(8, 5))

sns.histplot(df['loan\_percent\_income'], bins=30, kde=True, color='skyblue')

plt.title("Distribution of Loan Percent Income")

plt.xlabel("Loan Percent Income")

plt.ylabel("Frequency")

plt.grid(True)

plt.tight\_layout()

plt.show()

else:

print(" Column 'loan\_percent\_income' not found in the dataset.")

Overall Accuracy of each Models:

| S.No |  | Model | Accuracy (%) |
| --- | --- | --- | --- |
| 1 |  | Logistic Regression | 89.01% |
| 2 |  | Decision Tree | 89.92% |
| 3 |  | Random Forest | 92.72% |
| 4 |  | K-Nearest Neighbors (KNN) | 88.36% |
| 5 |  | Support Vector Machine (SVM) | 90.84% |

**APPENDIX – II REFERENCES**

1. U. Soni, N. Gudadhe, and S. Sharma (2020). *Machine Learning Approach for Loan Approval Prediction*. International Journal of Scientific & Engineering Research (IJSER), Vol. 11, Issue 3.  
   https://www.ijser.org/researchpaper/Machine-Learning-Approach-for-Loan-Approval-Prediction.pdf
2. R. Agarwal, A. Rai (2021). *Loan Approval Prediction using Random Forest and SVM*. International Research Journal of Engineering and Technology (IRJET), Volume 8, Issue 6.  
   https://www.irjet.net/archives/V8/i6/IRJET-V8I6789.pdf
3. R. Lundberg, S. Lee (2017). *A Unified Approach to Interpreting Model Predictions*. In *Advances in Neural Information Processing Systems (NeurIPS)*.  
   https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
4. Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016). *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.  
   <https://arxiv.org/abs/1602.04938>
5. SHAP Documentation – Interpretable Machine Learning  
   <https://shap.readthedocs.io/en/latest/>
6. LIME Documentation – Local Interpretable Model-Agnostic Explanations  
   <https://lime-ml.readthedocs.io/en/latest/>
7. Kaggle Loan Prediction Dataset  
   https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset
8. GitHub – Loan Prediction using ML (Community Codebase)  
   <https://github.com/krishnaik06/Loan-Approval-Prediction>