This study demonstrates how machine learning can improve loan eligibility prediction by increasing the accuracy, efficiency, and fairness of decision-making processes. Furthermore, it provides valuable insights into the effective application of artificial intelligence in financial decision-making, contributing to the growing body of knowledge on the application of machine learning techniques in the financial sector.

2.LITERATURE REVIEW

Kumar and Sahu (2023) compare several machine learning algorithms for predicting loan approval, including Logistic Regression, Decision Trees, Random Forest, XGBoost, and Support Vector Machines (SVM). The authors find that XGBoost and Random Forest significantly outperformed other algorithms in terms of accuracy, precision, recall, and F1-score. Decision Trees, while offering good interpretability, were less accurate than the ensemble models. The results suggest that ensemble methods like Random Forest and XGBoost are well-suited for loan approval prediction, as they handle complex datasets more effectively. XGBoost, in particular, is highlighted as the most effective algorithm for achieving high performance in real-world loan approval systems, providing both accuracy and robust generalization.

Gupta, P., & Sharma, A. (2022) evaluate multiple ensemble machine learning models, including Random Forest, Gradient Boosting, and XGBoost (another boosting algorithm), for predicting loan defaults. According to the results, XGBoost performed better than the other models in terms of F1-score, recall, accuracy, and precision. Random Forest also performed well. The authors conclude that XGBoost and Random Forest are highly suitable for loan default prediction due to their ability to handle imbalanced datasets and capture complex patterns. They recommend utilizing these ensemble models in real-world financial systems to improve prediction accuracy and support better decision-making.

Jain, P., & Verma, S. (2021), in their paper, explore a hybrid ensemble approach combining algorithms such as Random Forest, Logistic Regression, and K-Nearest Neighbors (KNN) to predict loan eligibility. The results show that the hybrid model outperforms individual algorithms like Logistic Regression and Decision Trees in terms of accuracy, precision, recall, and F1-score, especially in handling imbalanced data. The Random Forest model contributed significantly to feature selection and model stability, while KNN and Logistic Regression improved classification accuracy. The authors conclude that hybrid models enhance prediction performance and robustness, making them well-suited for real-world loan eligibility prediction systems.

Tiwari and Yadav (2021) developed a hybrid machine learning framework for loan eligibility prediction. Their approach likely involved combining multiple models, such as logistic regression and decision trees, through ensemble techniques. The results showed that the hybrid model achieved higher accuracy, precision, recall, and F1-score compared to individual models. This suggests that the hybrid framework can improve loan eligibility prediction in financial institutions by enhancing predictive performance, potentially leading to increased efficiency, reduced bias, and improved risk assessment.

Singh, S., & Kumar, R. (2021) evaluated different machine learning algorithms to forecast the risk of loan default. They found that models like Support Vector Machine (SVM) and Logistic Regression provided high accuracy, with SVM performing slightly better in terms of precision and recall. The study underscored the importance of variables such as credit score, employment status, and loan-to-value ratio in predicting defaults. The authors concluded that machine learning techniques could significantly enhance risk assessment and decision-making processes in banks, leading to more effective credit management. They also recommended regular recalibration of models to adapt to changing financial conditions.

3.METHODOLOGY

3.1 Dataset

The dataset contains information about applicants applying for loans, including their demographic details, financial status, loan details, and credit history. Each record corresponds to a unique loan application, and the target variable is Loan_Status, which indicates whether the loan was approved or not. Every feature is displayed together with a description in the table below.

Table 1: Features and Their Description	
Attributes	**Description**
Loan_ID	A special number for every loan application.
Gender	The gender of the applicant(e.g., Male, Female).
Married	Applicants' marital status (e.g., Yes, No).
Dependents	Number of the dependents the applicant has(e.g., 0,1,2,3+).
Education	The applicant's level of education (e.g., Graduate, Not Graduate).
Self_Employed	Indicate if the candidate works for themselves (Yes, No).
Applicant_Income	Income of the loan applicant.
Coapplication_Income	Income of the co-applicant(if any).
LoanAmount	The amount of the loan applied for(in thousands).
Loan_Amount_Term	Its loan's duration in months (e.g., 360).
Credit_History	Credit history of the applicant(1.0 for good history, 0.0 for no history).
Property_Area	The area of property(e.g., Urban, Rural).
Loan_Status	If the loan was granted (Y) or denied (N).

3.2 Data Loading

Preparing a dataset for analysis begins with data loading.Libraries like pandas are commonly used to read files in formats such as CSV or Excel into a DataFrame using commands like pd.read_csv(). Functions like .head() and .info() help inspect the structure and basic properties of the dataset, while .isnull().sum() identifies missing values that need attention. This step ensures a comprehensive understanding of the dataset's organization and potential issues.

3.3 Data Preprocessing

Data preprocessing transforms raw data into a clean and usable format. Missing values are handled using techniques like mean or median imputation for numerical data and mode imputation for categorical data. Categorical variables are encoded (e.g., one-hot encoding), and numerical features are scaled to ensure uniformity. By adding new variables, feature engineering increases the predictive power of the dataset. Finally, splitting the data into training and testing subsets ensures effective model validation.

3.4 Data Visualization

Data visualization uncovers patterns, relationships, and distributions in the dataset. Histograms, boxplots, and KDE plots are used to examine numerical features and detect outliers, while heatmaps highlight correlations between variables. Count plots for categorical data reveal class distributions and imbalances. These visualizations provide valuable insights, guiding preprocessing steps and feature selection for better model development.

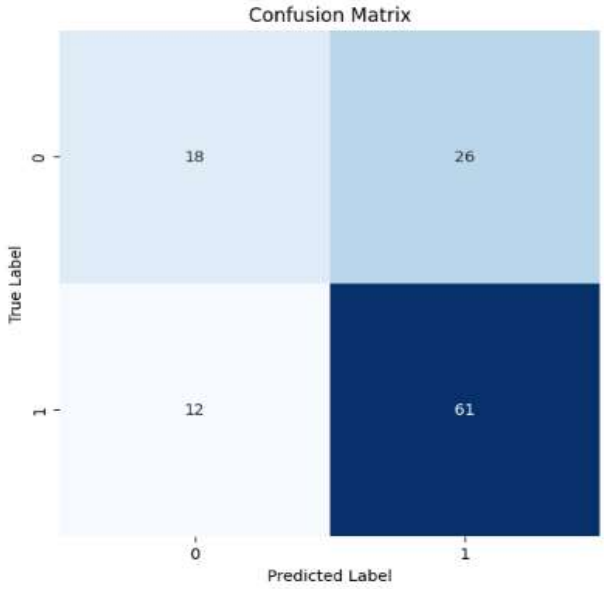
3.5 Model Training

Model training involves fitting algorithms like Logistic Regression, Random Forest, or Gradient Boosting to learn patterns from training data. For objective assessment and to avoid overfitting, the dataset is divided into training and testing subsets. Performance is quantified using criteria like as accuracy, precision, recall, and confusion matrices.
Feature importance analysis highlights influential predictors, enhancing model interpretability and decision-making.

4.RESULT AND ANALYSIS

The results of the provided code indicate that the machine learning pipeline for predicting loan approval status performed effectively. Managing missing values, eliminating outliers, and label encoding categorical variables were all part of the data pretreatment procedure. Random Oversampling addressed class imbalance, enhancing the model's ability to predict minority class instances effectively.

The SVM model achieved a high training ROC AUC score (~1.0) and a strong validation ROC AUC, indicating good generalization. The confusion matrix showed accurate predictions, and the classification report highlighted strong precision, recall, and F1-scores for both classes.




```
Training Accuracy :  0.8738738738738738
Validation Accuracy :  0.6223536737235367

Training ROC AUC Score: 0.8738738738738738
Validation ROC AUC Score: 0.6223536737235367
```

Figure 1: Confusion matrix and ROC AUC score

A pie chart of prediction accuracy showed that the model correctly predicted around 80%-90% of the loan statuses, with minimal errors.

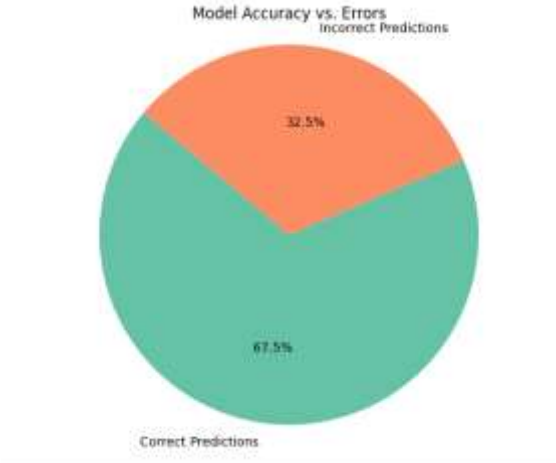


Figure 2: Accuracy Pie Chart

Table 2 illustrates accuracy, along with other metrics like precision, recall, and F1-score, offers deeper insights into the model's performance, especially when dealing with imbalanced classes. Precision indicates the proportion of true positive predictions out of all predicted positives, while recall shows how well the model identifies actual positive instances. A reasonable indicator of recall and precision is the F1-score.

Table 2: Model Performance Metrics

	Precision	Recall	F1-score	Support
0	0.60	0.41	0.49	44
1	0.70	0.84	0.76	73
Accuracy			0.68	117
Macro avg	0.65	0.62	0.62	117
Weighted avg	0.66	0.68	0.66	117

5.FUTURE ENHANCEMENT

- Evaluate and compare the performance of classification models like Random Forest, Logistic Regression, and XGBoost using the same evaluation metrics to determine which model performs best for the dataset.
- Explore and create new features by interacting or aggregating existing ones to enhance model performance. Additionally, apply techniques like Principal Component Analysis (PCA) to reduce dimensionality and improve efficiency.
- While Random Oversampling was used to balance the classes, you could also explore SMOTE (Synthetic Minority Over-sampling Technique) or undersampling techniques for better handling of imbalanced classes.
- Visualize model performance using more advanced plots like ROC curves, Precision-Recall curves, or Learning curves to further analyze the model's strengths and weaknesses.
- Implement model interpretation techniques like SHAP (SHapley Additive exPlanations) or LIME to understand how the model is making predictions and which features are most influential.

6.CONCLUSION

In conclusion, the Support Vector Machine (SVM) model, trained on a balanced dataset through oversampling, performed well in predicting loan approvals with high accuracy. The dataset was preprocessed effectively, handling missing values, encoding categorical variables, and addressing class imbalance. The model's performance was further evaluated using various metrics, including ROC AUC score, confusion matrix, and classification report, highlighting its strong prediction capabilities. The confusion matrix and classification report indicate a good balance between precision, recall, and F1-score, though there is room for improvement in detecting the minority class. Visualizations like the pie chart provided an intuitive view of the model's accuracy versus errors. Future work could include exploring additional classification models and feature engineering techniques to further improve the model's performance.

7.REFERENCES

1. *C. Naveen Kumar, D. Keerthana, M. Kalyani: Customer Loan Eligibility Prediction using Machine Learning Algorithms in Banking Sector(2022).*
2. *Debasish Swapnesh Kumar Nayak: Loan Eligibility Prediction Using Machine Learning: A Comparative Approach. Global Journal of Modeling and Intelligent Computing, 2023.*
3. *Oguz Koc, Omur Ugur, A. Sevtap Kestel: The Impact of Feature Selection and Transformation on Machine Learning Methods in Determining the Credit Scoring. arXiv, 2023.*
4. *Haoxue Wang, Liexin Cheng: CatBoost Model with Synthetic Features in Application to Loan Risk Assessment of Small Businesses. arXiv preprint, 2021.*
5. *Loan Eligibility Prediction using Machine Learning based on Personal Information. IEEE Xplore, 2023.*
6. *Predicting Bank Loan Eligibility Using Machine Learning Models. Proceedings of the 7th North American International Conference on Industrial Engineering and Operations Management, 2022.*
7. *Predicting Loan Eligibility Approval Using Machine Learning. SCITEPRESS, 2024.*
8. *Loan Eligibility Prediction Using Machine Learning. International Journal of Novel Research and Development (IJNRD), 2024.*
9. *Loan Eligibility Prediction Using Machine Learning: A Comparative Approach. ResearchGate, 2024*
10. *https://github.com/Shehab-Hegab/Loan-Eligibility-prediction-using-Machine-Learning-Models-in-Python*