CSE 422: Artificial Intelligence

Loan approval prediction using machine learning

Afsan Haque (20301145)

Antu Dutta (20101282)

Asif Rahman (20101287)

Aditta Barua (20101023)

Section : 07

Table of Contents

Introduction

This project focuses on building a machine-learning model for loan approval based on individual-specific factors. We gather data on critical indicators related to loan eligibility, conduct data preprocessing and feature engineering, and then train and assess machine learning models. The most accurate model will be integrated into a user-friendly platform. This platform will offer personalized loan assessments and recommendations, empowering individuals to make well-informed choices about their financial decisions

Motivation

Loan approval is a significant financial challenge, impacting millions of individuals and institutions globally and placing a substantial burden on financial systems. Early determination and processing are critical to enhancing the efficiency and effectiveness of the loan approval process and reducing associated costs. Machine learning (ML) has the potential to revolutionize loan approval by utilizing extensive datasets and advanced algorithms to recognize patterns and predict loan approval risk.

The motivation for developing a loan approval ML project originates from several key factors. First, loan eligibility criteria are often complex, making it difficult for applicants to ascertain their approval chances. ML algorithms can analyze various financial data, credit history, personal information, and economic indicators to identify subtle patterns and indicators of loan approval risk that may not be readily apparent to loan officers.

Second, prompt identification of loan approval risks can enable proactive measures and prevent financial complications. ML models can be trained on extensive datasets to predict loan approval risk with high accuracy, allowing financial institutions to identify high-risk applicants and implement tailored solutions, such as credit counseling, loan terms adjustments, or referral to financial advisors, to enhance the chances of loan approval and minimize financial issues. Third, loan approval processes often entail substantial administrative and operational costs, including paperwork, compliance checks, and manual verifications. Early detection of loan approval risks can lead to cost savings by streamlining approval procedures, reducing default rates, optimizing credit terms, and improving loan approval outcomes.

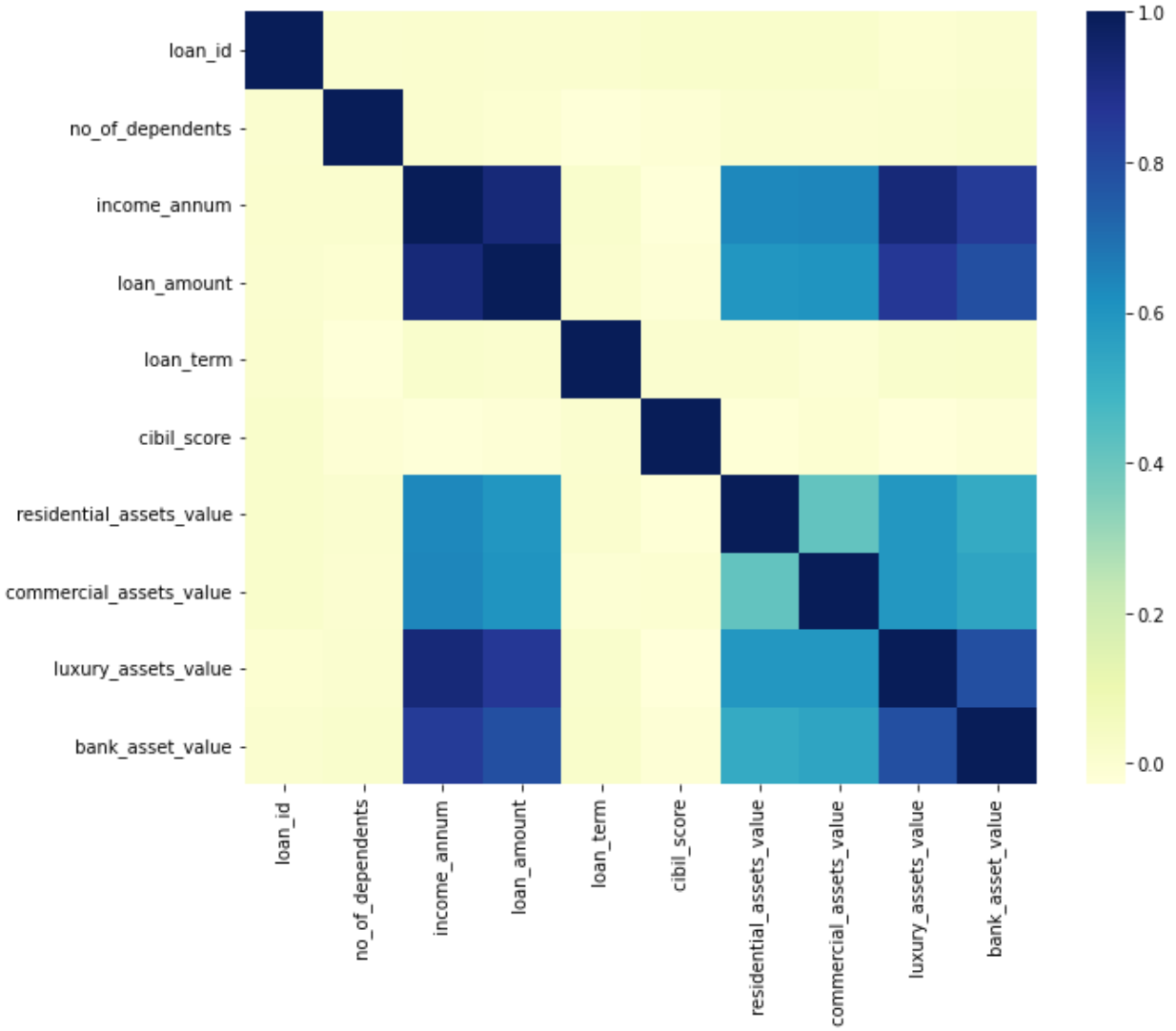
In summary, developing a loan approval ML project can revolutionize the financial sector by enabling early risk detection and intervention, enhancing loan approval outcomes, and reducing associated costs. By harnessing the capabilities of ML, we can elevate the accuracy and efficiency of loan approval processes, ultimately leading to better financial outcomes for both lenders and borrowers.

Data Description

Link:https://drive.google.com/file/d/1N85wK8dbjfdQ7SnWjx_g3BX68nlh0ul5/view?usp=sharing

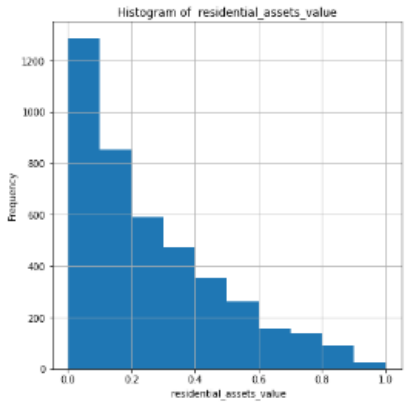
Number of Features: 13
Type of class/label: Categorical and Continuous
Number of data points: 4269
Types of features: 13

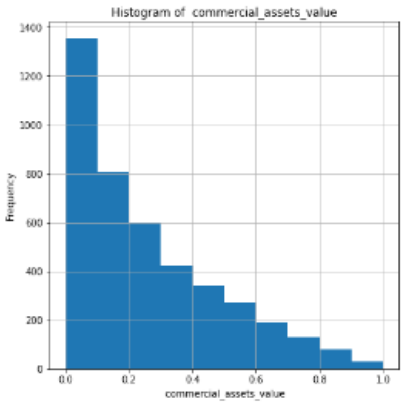
Correlation of the features along with the label/class:

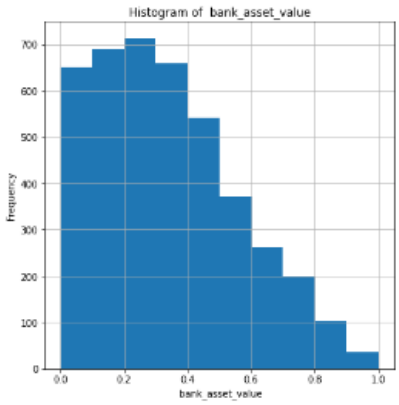


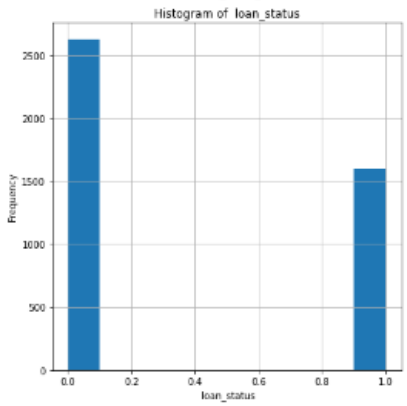
From the sns heatmap of correlation we can see that income amount have high correlation with loan amount,loan amount have high correlation with income annum and luxury asset value has high correlation with income annum.So ' income_annum', ' loan_amount', ' luxury_assets_value' need to be dropped from the dataset for better redundancy reduction ,avoiding multicollinearity

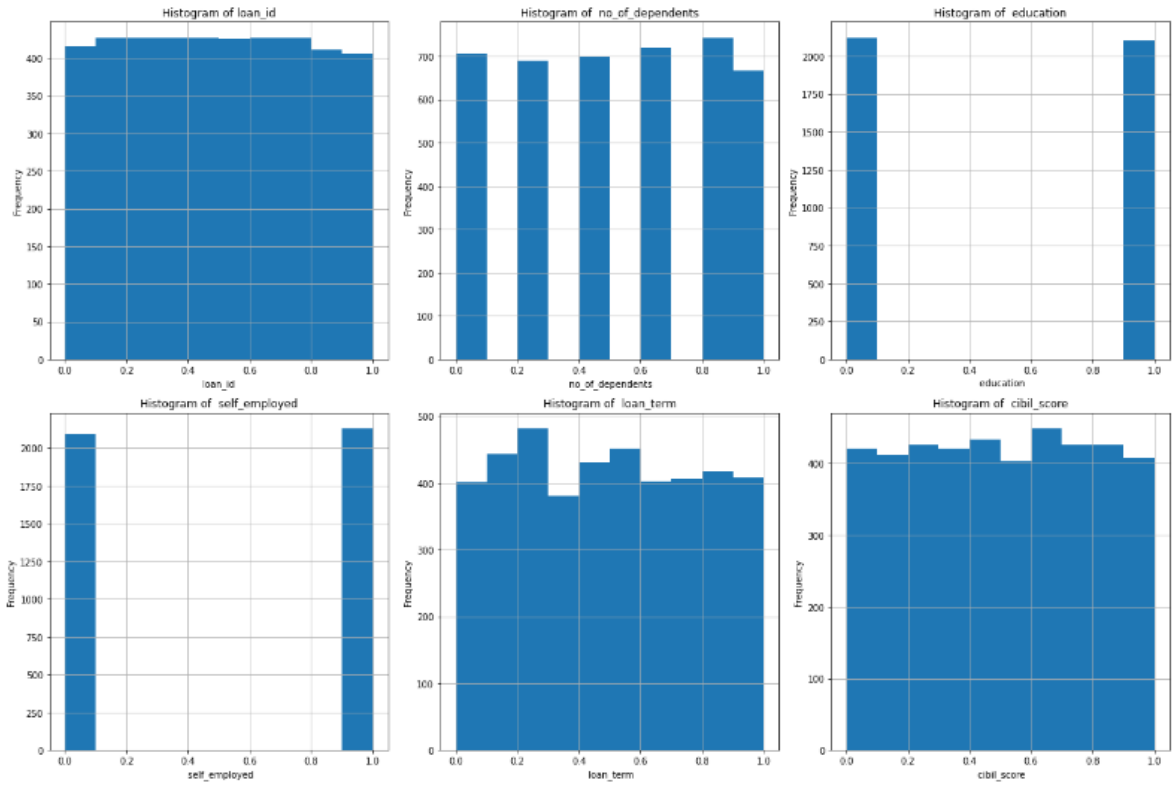
Biasness/Balanced











Like this bar chart, all the classes' bias/ balance parts were checked and the result was the database was biased in residential_assets_value,c, and ommercial_assets_value.

Solution: Replace the data greater than 75 percentile and less than 25 percentile with their mean values

Dataset pre-processing

Problem 1: 13 features had a total of 83 null values.

Solutions: Delete rows

```
loan_id                     0
 no_of_dependents           2
 education                  4
 self_employed              5
 income_annum               5
 loan_amount               16
 loan_term                  2
 cibil_score                2
 residential_assets_value  13
 commercial_assets_value   13
 luxury_assets_value        6
 bank_asset_value           2
 loan_status                0
dtype: int64
```

Before Preprocessing

```
loan_id                     0
 no_of_dependents           0
 education                  0
 self_employed              0
 income_annum               0
 loan_amount                0
 loan_term                  0
 cibil_score                0
 residential_assets_value   0
 commercial_assets_value    0
 luxury_assets_value        0
 bank_asset_value           0
 loan_status                0
dtype: int64
```

After Preprocessing

Dataset splitting

To train the model data splitting was done as like 70% for training model and 30% for testing. On this basis- Training data set - 2955 and Testing data set - 1267

Model training

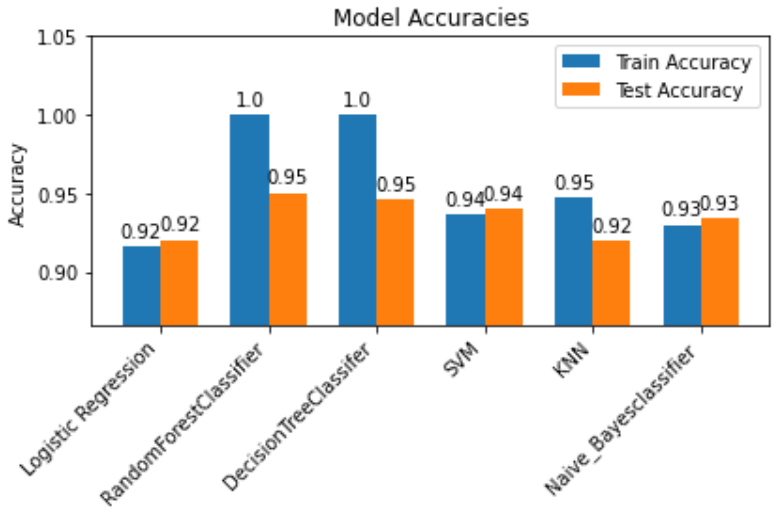
Model Name	Accuracy(Train) (%)	Accuracy(Test) (%)
Logistic Regression	92	92
Decision tree	100	95
Kth Nearest Neighbor	95	92
SVM Model	94	94
Naive Bayes Classifier	93	93
RandomForestClassifier	100	95

The table summarizes the accuracy scores (in percentage) of different machine learning models on both the training and testing datasets. Here's a brief summary:

-Logistic Regression achieves an accuracy of 92% on both the training and testing datasets.
-Decision Tree achieves perfect accuracy (100%) on the training data and 95% accuracy on the testing data, indicating potential overfitting.
-Kth Nearest Neighbor** achieves 95% accuracy on the training data and 92% accuracy on the testing data.
-SVM Model achieves 94% accuracy on both the training and testing datasets.
-Naive Bayes Classifier** achieves 93% accuracy on both the training and testing datasets.
-RandomForestClassifier achieves perfect accuracy (100%) on the training data and 95% accuracy on the testing data, also suggesting potential overfitting.

In summary, the Decision Tree and RandomForestClassifier models show signs of overfitting as they have significantly higher accuracy on the training data compared to the testing data. The other models have similar or consistent performance on both datasets.

Model selection/Comparison analysis



Also, from the Bar comparison we can see that Logistic Regression has the best performance than other models.

Model testing

To test the model, the data was used

First five instances of the Loan_status are:

```
0     Approved
1     Rejected
2     Rejected
3     Rejected
4     Rejected
5     Rejected
Name:  loan_status, dtype: object
```

The predicted results of all the algorithms are:

```
Model: Logistic Regression, Prediction: [0 1 1 1 1]
Model: RandomForestClassifier, Prediction: [0 1 1 1 1]
Model: DecisionTreeClassifer, Prediction: [0 1 1 1 1]
Model: SVM, Prediction: [0 1 1 1 1]
Model: KNN, Prediction: [0 1 1 1 1]
Model: Naive_Bayesclassifier, Prediction: [0 1 1 1 1]
```

Result: From the results we can observe that all the models give the correct predictions according to the given dataset.

Conclusion

The results obtained from applying various machine learning algorithms to predict loan approval are highly promising. These models demonstrate strong performance in determining loan eligibility. Here is a summary of the key findings: Random ForestClassifiere and DecissionTreeClassifier achieved the highest accuracy on the training set, both scoring a perfect 100%. On the test set, they both exhibited exceptional performance with an accuracy of 95%, indicating their potential as robust models for loan approval. Logistic Regression and SVM Model also performed admirably, with an accuracy rate of 92% and 94%, respectively, on the test set. These models are promising candidates for loan approval predictions. Kth Nearest Neighbor displayed solid results with an accuracy of 91% on the test set, further underlining its suitability for loan eligibility prediction. Naive Bayes Classifier exhibited respectable performance with an accuracy of 93% on the test set. While it performed slightly below some other models, it still showcases potential in loan approval prediction.

In conclusion, the results from these machine learning algorithms provide a strong foundation for the prediction of loan approval. They demonstrate high accuracy, suggesting that these models can effectively aid in automating the loan approval process. However, further research is recommended, including the evaluation of models on larger and more diverse datasets and the consideration of additional performance metrics to ensure the reliability and generalizability of these models in real-world financial scenarios. These findings represent a promising step toward streamlining and improving the loan approval process, ultimately benefiting both lenders and borrowers.

Future work/Extension

The prediction of loan approval using personal and financial indicators with machine learning models has shown promise in improving the efficiency and accuracy of the lending process. However, there are several avenues for future research and extensions that can further enhance the performance and applicability of these models:

Feature Selection and Engineering:

Future research can delve deeper into identifying and selecting the most relevant personal and financial indicators that hold significant predictive power for loan approval. Advanced feature selection methods like Recursive Feature Elimination, Principal Component Analysis (PCA), or domain-specific feature engineering can be explored to enhance model accuracy.

Ensemble Methods:

Ensemble techniques, such as Random Forest, Gradient Boosting, or Stacking, can be applied to combine predictions from various base models like Logistic Regression, Decision Trees, K-Nearest Neighbors, Support Vector Machines, and Naive Bayes classifiers. This ensemble approach can create more robust and accurate models by leveraging the strengths of different algorithms.

Hyperparameter Tuning:

Hyperparameters play a crucial role in machine learning models' performance. Future work can focus on fine-tuning these hyperparameters to optimize model accuracy. Techniques like Grid Search, Random Search, or Bayesian Optimization can be employed for hyperparameter tuning.

Validation on Larger and Diverse Datasets:

The robustness and generalizability of machine learning models are often influenced by the size and diversity of the training dataset. Research efforts can involve validating models on larger and more diverse datasets to assess their performance in real-world lending scenarios. This can help ensure the models are effective across different borrower profiles.

Real-world Implementation and Regulatory Compliance:

Once models are optimized and validated, real-world implementation and regulatory compliance become paramount. Collaborations with financial institutions, data collection from actual loan applicants, and compliance with lending regulations can pave the way for assessing model applicability in practical lending environments.

Interpretability and Explainability:

The interpretability and explainability of machine learning models are crucial for loan approval systems. Future work can focus on developing models that are not only accurate but also interpretable and explainable. This can enhance trust in the system by allowing financial professionals to understand the reasoning behind approval or rejection decisions

In summary, while the current research has shown promise in predicting loan approval using personal and financial indicators with machine learning, ongoing exploration and validation of these extensions can further improve the accuracy, fairness, and practical utility of these models in the lending industry. Continued research efforts in this area have the potential to enhance decision-making, reduce risks, and streamline the loan approval process.

References

1. scikit-learn: machine learning in Python — scikit-learn 1.3.0 documentation. (n.d.). https://scikit-learn.org/stable/

2. pandas documentation — pandas 2.0.3 documentation. (n.d.). https://pandas.pydata.org/docs/index.html