Loan Approval Prediction Using Classification Algorithms

Mohammed Aman(2020A7PS0050U)

Department of Computer Science,

Birla Institute of Technology and Science,

Pilani, Dubai International Academic City,

Dubai, United Arab Emirates

Abstract

For financial organizations and lenders, predicting loan approval is a critical responsibility in risk management and loss minimization. Machine learning algorithms have become very popular in recent years and are frequently used to forecast loan approvals based on historical data. This task's objective is to correctly categorize requests for loans as approved or rejected based on a variety of factors, including income, credit score, loan amount, employment history, and more.

Pre-processing of data, feature engineering, model selection, and evaluation are typically steps in the method of loan approval prediction. For predicting loan acceptance, several machine learning methods have been utilized, including logistic regression, decision trees, random forests, and neural networks. The calibre and volume of the data utilized for training and testing these algorithms affects how well they work.

Accurate loan approval prediction can give lenders possibilities to acquire financing while also assisting them in identifying dangerous applicants and preventing losses. By offering a more unbiased and based on information method, it can also aid in lowering prejudice in financing choices.

Keywords

- Regularization - refers to methods for calibrating machine learning models that aim to reduce the adjusted loss function and avoid either over or under fitting.
- Multicollinearity - a statistical idea in which a model's several independent variables are correlated.
- Variable selection - To pick acceptable variables from a comprehensive list of variables by deleting those that are unnecessary or redundant, or to choose among numerous variables which to include in a certain model.
- High Dimensional data- High-dimensional data are those where the number of features (observed variables), p, is near to or greater than the number of observations n.
- Shrinkage Method-Some of the parameters are intended to be shrunk to zero.
- Computational Algorithms-A computational algorithm comprises several control parameters that are deterministic in the first part of the algorithm, constitute a fixed component of the program, completely determine the computational process, and guarantee the first part of the algorithm is adaptable to a specific machine.
- Data Visualization-Data visualization is the graphical representation of information and data.
- Decision Tree- It is a hierarchical model in machine learning that uses a tree- like structure to represent a sequence of decisions and their possible consequences, leading to a final prediction or decision.

- Random Forest- It is an ensemble machine learning algorithm that groups multiple decision trees to better the accuracy and strength of predictions.

Introduction

Applying machine learning, loan approval prediction entails creating a classification model that can correctly categorize loan applications as accepted or refused based on a variety of input features. These attributes can be pre- processed and tailored to extract the data that is most closely related to judgments about loan acceptance.

The selection of algorithm depends on the needs of the task, such as accuracy, interpretability, and speed. Each algorithm has strengths and disadvantages of its own.

Several performance metrics, including precision, accuracy, recall, and F1-score, can be used to evaluate how well the machine learning model is performing. The task-specific needs, such as reducing false positives or false negatives, must be taken into consideration while selecting an assessment metric. To check that the algorithm can generalize to new data, it can also be tested using a holdout dataset, a portion of the data that was not utilized during training.

Machine learning has several benefits over conventional methods for predicting loan acceptance, including increased accuracy and less bias. Machine learning models can discover patterns and connections in the data that conventional statistical methods might miss. They can also deal with complicated interactions between parameters and enormous amounts of data. This can lower the risk of failure or scam and help lenders make more informed decisions about loan approvals.

Accurate loan approval prediction can give lenders possibilities to acquire financing while also assisting them in identifying dangerous applicants and preventing losses. By offering a more unbiased and based on information method, it can also aid in lowering prejudice in financing choices.

Literature Survey

The paper[1] presents a loan prediction system using machine learning. They collected data on various parameters including income, credit history and employment history to predict loan eligibility. Although linear regression did not perform well, GBRT and BTC achieved 93% and 95% accuracy, respectively. Despite their effectiveness, the complexity of GBRT-based regression trees makes them difficult to interpret because they involve 50 decision trees. The system successfully reduces loan officer workload by accurately predicting loan eligibility.

The authors of the paper [2] created a system for predicting loan eligibility based on the Random Forest model. To predict loan eligibility, they gathered a variety of data elements, including income, credit history, and employment history. The Random Forest model provided encouraging loan eligibility predictions, making it a useful tool for evaluating loan applications. The paper, however, made no mention of the model's complexity or particular accuracy rates. Nonetheless, the study demonstrated the potential of machine learning for facilitating loan eligibility decisions.

In paper [3], an exploratory analysis was conducted on loan prediction. The study examined various factors affecting loan approval, including income, credit history, and employment details. Different machine learning techniques were employed, showcasing the significance of algorithms in predicting loan eligibility. The paper emphasized the need for accurate predictions to streamline the loan approval process and reduce the burden on financial institutions.

In paper [4], the authors sought to create predictive models for figuring out who qualifies for bank loans. They acquired and examined information on numerous factors that were important for loan approval. They compared the effectiveness of various machine learning models through their research. According to the study, some models, such Gradient Boosted Regression Trees (GBRT) and Boosted Trees Classifiers (BTC), were highly successful and accurate at predicting loan eligibility. However, they acknowledged that the intricacy of the GBRT-based regression trees, which include several decision trees, made it difficult to interpret the results. Despite this complexity, the generated models effectively shown their ability to expedite the banking industry's process for determining loan eligibility.

In [5], the authors developed a loan approval prediction model using machine learning. They collected and analyzed data on various features such as income, credit history, and loan amount to determine the likelihood of loan approval. Through the application of classification algorithms, the model achieved a high accuracy rate, effectively predicting loan approval outcomes. They emphasized the significance of feature engineering in improving the model's performance. Despite potential challenges in interpretability, their system showcased the practical viability of machine learning in streamlining the loan approval process.

The paper [6] aims to predict loan approvals using machine learning techniques. It entails gathering pertinent information on things like earnings, credit history, and employment specifics. A combination of Gradient Boosted Regression Trees (GBRT) and Boosted Trees Classifiers (BTC) produced a high accuracy rate of 92% for loan acceptance prediction after the authors tested with a number of machine learning methods. However, they pointed out that the complexity added by the ensemble of decision trees hindered the interpretability of the GBRT-based regression trees. Nevertheless, the system successfully automated the loan approval procedure, illuminating the usefulness of machine learning in this field.

The paper [7] focuses on predicting loan approvals using supervised machine learning algorithms. To determine the results of loan approval, the authors gathered relevant data and used a variety of supervised learning approaches. The research highlighted the accurate loan approval predictions, demonstrating the relevance of machine learning in this field despite the lack of particular algorithm details.

The paper [8] focuses on predicting customer loan eligibility in the banking industry using machine learning algorithms. It places a focus on gathering pertinent consumer information, including information about income, credit history, and employment. Gradient Boosted Regression Trees (GBRT) and Boosted Trees Classifiers (BTC) are successful machine learning models that achieve high accuracy rates of 92% and 94%, respectively, according to the study, which emphasizes the performance of several machine learning models. Given that GBRT-based regression trees are made up of several decision trees, however, it is still difficult to evaluate them due to their complexity, which is consistent with other findings. But the algorithm does a good job of forecasting a customer's loan eligibility, which might lessen the manual work that bankers must do.

In paper [9], the authors developed a machine learning-based model for predicting loan approval. To determine loan eligibility, they used a variety of factors, including financial background and employment history. Their model was highly effective in forecasting loan approval rates properly, illuminating the usefulness of machine learning in the field of loan approval prediction.

In paper [10], the authors concentrated on utilizing machine learning to forecast a modernized loan approval procedure. To build a predictive model, they gathered pertinent data and used machine learning algorithms. The summary made no explicit mention of the exact methods used. However, the article showed how machine learning may be used to update the loan approval procedure.

In paper [11], the authors evaluated the performance of classification algorithms for predicting loan approval. They concentrated on applying several computational strategies to improve the precision of loan approval predictions. Although specifics were withheld, it can be assumed that they sought to improve the loan acceptance prediction process, perhaps by evaluating and contrasting the effectiveness of various categorization systems. Further insights into the paper's conclusions, however, are constrained given the absence of precise information on the algorithms or their outcomes.

In paper [12], the authors developed a customer loan eligibility prediction system using machine learning. They collected relevant data and applied various ML algorithms to create a predictive model. Their work focused on accurately assessing the eligibility of customers for loans. The models employed showcased promising results, demonstrating the practical viability of ML in determining loan eligibility efficiently.

In paper [13], the study focuses on using several machine learning models to predict bank loan eligibility. It compares and contrasts these models thoroughly in an effort to determine which strategy is the most successful. The study underlines the significance of precise predictions in cutting down on the time and labor required for loan processing. The predictive algorithms take into account a number of variables, including credit history and income. The study emphasizes how important it is to use the right model for precise loan eligibility forecasts, which will eventually help to speed up the loan approval process.

In paper [14], the authors conducted exploratory data analysis for loan prediction. They likely analyzed various parameters relevant to loan approval, possibly including income, credit history, and employment history. While the specifics of the analysis were not provided, it can be inferred that they aimed to understand the relationships and patterns within the dataset. This process likely involved visualizations, statistical summaries, and data cleaning techniques to gain insights into the data's characteristics and distributions.

In paper [15], the author focuses on predicting credit card approvals using machine learning techniques. The study involves the use of various machine learning algorithms to assess the likelihood of credit card approval for applicants. While specific models were not explicitly mentioned, the paper emphasized the effectiveness of the chosen techniques in accurately predicting credit card approvals. The study's primary goal was to develop a reliable system for assessing credit card approval, potentially streamlining the application process and enhancing decision-making for financial institutions.

Algorithm

MODEL.PY

sbajshhas PROCEDURE train_and_save_models():

    SET data_df = READ_CSV_FILE('C:/Users/mailm/Downloads/loan_predicition-master/loan_predicition-master/LoanApprovalPrediction.csv')

    CALL data_df.drop(['Loan_ID'], axis=1, inplace=True)

```
label_encoder = CREATE_LabelEncoder()
categorical_cols = GET_CategoricalColumns(data_df)
FOR EACH col IN categorical_cols:
    ENCODE_CategoricalColumn(data_df, label_encoder, col)
FOR EACH col IN data_df.columns:
    FILL_MISSING_VALUES(data_df, col)
features = GET_Features(data_df)
target = GET_Target(data_df)
features_train, features_test, target_train, target_test = SPLIT_DATA(features, target, test_size=0.3,
random_state=7)
rfc_model = CREATE_RandomForestClassifier()
FIT_MODEL(rfc_model, features_train, target_train)
SAVE_MODEL(rfc_model, 'trained_model_rfc.pkl')
svm_model = CREATE_SVC()
FIT_MODEL(svm_model, features_train, target_train)
SAVE_MODEL(svm_model, 'trained_model_svm.pkl')
PROCEDURE READ_CSV_FILE(file_path):
    RETURN READ_DATA_FROM_CSV(file_path)
PROCEDURE GET_CategoricalColumns(data):
    RETURN LIST_OF_CATEGORICAL_COLUMNS(data)
PROCEDURE ENCODE_CategoricalColumn(data, label_encoder, column):
    FIT_TRANSFORM_LABEL_ENCODER(label_encoder, data[column])
PROCEDURE FILL_MISSING_VALUES(data, column):
    FILL_MISSING_VALUES_WITH_MEAN(data, column)
PROCEDURE GET_Features(data):
    RETURN EXTRACT_FEATURES(data)
PROCEDURE GET_Target(data):
    RETURN EXTRACT_TARGET(data)
PROCEDURE SPLIT_DATA(features, target, test_size, random_state):
    RETURN TRAIN_TEST_SPLIT(features, target, test_size, random_state)
PROCEDURE CREATE_LabelEncoder():
    RETURN NEW_LabelEncoder()
PROCEDURE CREATE_RandomForestClassifier():
    RETURN NEW_RandomForestClassifier()
```

```
PROCEDURE CREATE_SVC():
    RETURN NEW_SVC()
PROCEDURE FIT_MODEL(model, features, target):
    FIT_MODEL_ON_TRAINING_DATA(model, features, target)
PROCEDURE SAVE_MODEL(model, file_path):
    SAVE_MODEL_TO_FILE(model, file_path)
train_and_save_models()
```

APP.PY

```
PROCEDURE main():
    SET background_style = "<div style='background-color:black; padding:13px'>
                      <h1 style='color:white'>Loan Approval Prediction GUI</h1>
                      <h2 style='color:grey'>Mohammed Aman</h2>
                      <h3 style='color:grey'>2020A7PS0050U</h3>
                   </div>"
    CALL st.markdown(background_style, unsafe_allow_html=True)
    SET col_left, col_right = st.columns((2,2))
    SET gender_input = col_left.selectbox('Gender', ('Male', 'Female'))
    SET married_input = col_right.selectbox('Married', ('Yes', 'No'))
    SET dependents_input = col_left.selectbox('Dependents', ('None', 'One', 'Two', 'Three'))
    SET education_input = col_right.selectbox('Education', ('Graduate', 'Not Graduate'))
    SET self_employed_input = col_left.selectbox('Self-Employed', ('Yes', 'No'))
    SET applicant_income_input = col_right.number_input('Applicant Income')
    SET coapplicant_income_input = col_left.number_input('Coapplicant Income')
    SET loan_amount_input = col_right.number_input('Loan Amount')
    SET loan_term_input = col_left.number_input('Loan Tenor')
    SET credit_history_input = col_right.number_input('Credit History', 0.0, 1.0)
    SET property_area_input = st.selectbox('Property Area', ('Semiurban', 'Urban', 'Rural'))
    SET classifier_selection = st.selectbox('Classifier Type',('Random Forest','SVM'))
    SET predict_button = st.button('Predict')
    IF predict_button:
        SET result = predict(gender_input, married_input, dependents_input, education_input,
self_employed_input, applicant_income_input,
```

coapplicant_income_input, loan_amount_input, loan_term_input, credit_history_input, property_area_input, classifier_selection)

CALL st.success(f'Prediction Result: {result} for the loan')

PROCEDURE predict(gender, married, dependents, education, self_employed, applicant_income,

coapplicant_income, loan_amount, loan_term, credit_history, property_area, classifier_selection):

SET selected_classifier = 1 IF classifier_selection == 'Random Forest' ELSE 2

SET gender_code = 0 IF gender == 'Male' ELSE 1

SET married_code = 0 IF married == 'Yes' ELSE 1

SET dependents_code = 0.0 IF dependents == 'None' ELSE 1.0 IF dependents == 'One' ELSE 2.0 IF dependents == 'Two' ELSE 3.0

SET education_code = 0 IF education == 'Graduate' ELSE 1

SET self_employed_code = 0 IF self_employed == 'Yes' ELSE 1

SET property_area_code = 0 IF property_area == 'Semiurban' ELSE 1 IF property_area == 'Urban' ELSE 2

SET loan_amount_in_thousands = loan_amount / 1000

SET coapplicant_income_in_thousands = coapplicant_income / 1000

IF selected_classifier == 1:

WITH OPEN('trained_model_rfc.pkl', 'rb') AS pkl:

SET model_rfc = LOAD_MODEL_FROM_FILE(pkl)

SET prediction = PREDICT_WITH_MODEL(model_rfc, [[gender_code, married_code, dependents_code, education_code, self_employed_code, applicant_income,

coapplicant_income_in_thousands, loan_amount_in_thousands, loan_term, credit_history, property_area_code]])

ELSE IF selected_classifier == 2:

WITH OPEN('trained_model_svm.pkl', 'rb') AS pkl:

SET model_svm = LOAD_MODEL_FROM_FILE(pkl)

SET prediction = PREDICT_WITH_MODEL(model_svm, [[gender_code, married_code, dependents_code, education_code, self_employed_code, applicant_income,

coapplicant_income_in_thousands, loan_amount_in_thousands, loan_term, credit_history, property_area_code]])

SET prediction_result = 'Not Eligible' IF prediction == 0 ELSE 'Eligible'

RETURN prediction_result

IF __NAME__ == '__MAIN__':

CALL main()

Implementation(GUI):

MODEL.PY

```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import RidgeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
import pickle
import pandas as pd

data_df = pd.read_csv('C:/Users/mailm/Downloads/loan_predicition-
master/loan_predicition-master/LoanApprovalPrediction.csv')
data_df.drop(['Loan_ID'], axis=1, inplace=True)

label_encoder = LabelEncoder()
categorical_cols = (data_df.dtypes == 'object')
for col in list(categorical_cols[categorical_cols].index):
    data_df[col] = label_encoder.fit_transform(data_df[col])

for col in data_df.columns:
    data_df[col] = data_df[col].fillna(data_df[col].mean())

features = data_df.drop(['Loan_Status'], axis=1)
target = data_df['Loan_Status']

features_train, features_test, target_train, target_test =
train_test_split(features, target, test_size=0.3, random_state=7)

rfc_model = RandomForestClassifier()
rfc_model.fit(features_train, target_train)

with open('trained_model_rfc.pkl', mode='wb') as pkl:
    pickle.dump(rfc_model, pkl)

svm_model = SVC()
svm_model.fit(features_train, target_train)

with open('trained_model_svm.pkl', mode='wb') as pkl:
    pickle.dump(svm_model, pkl)
```

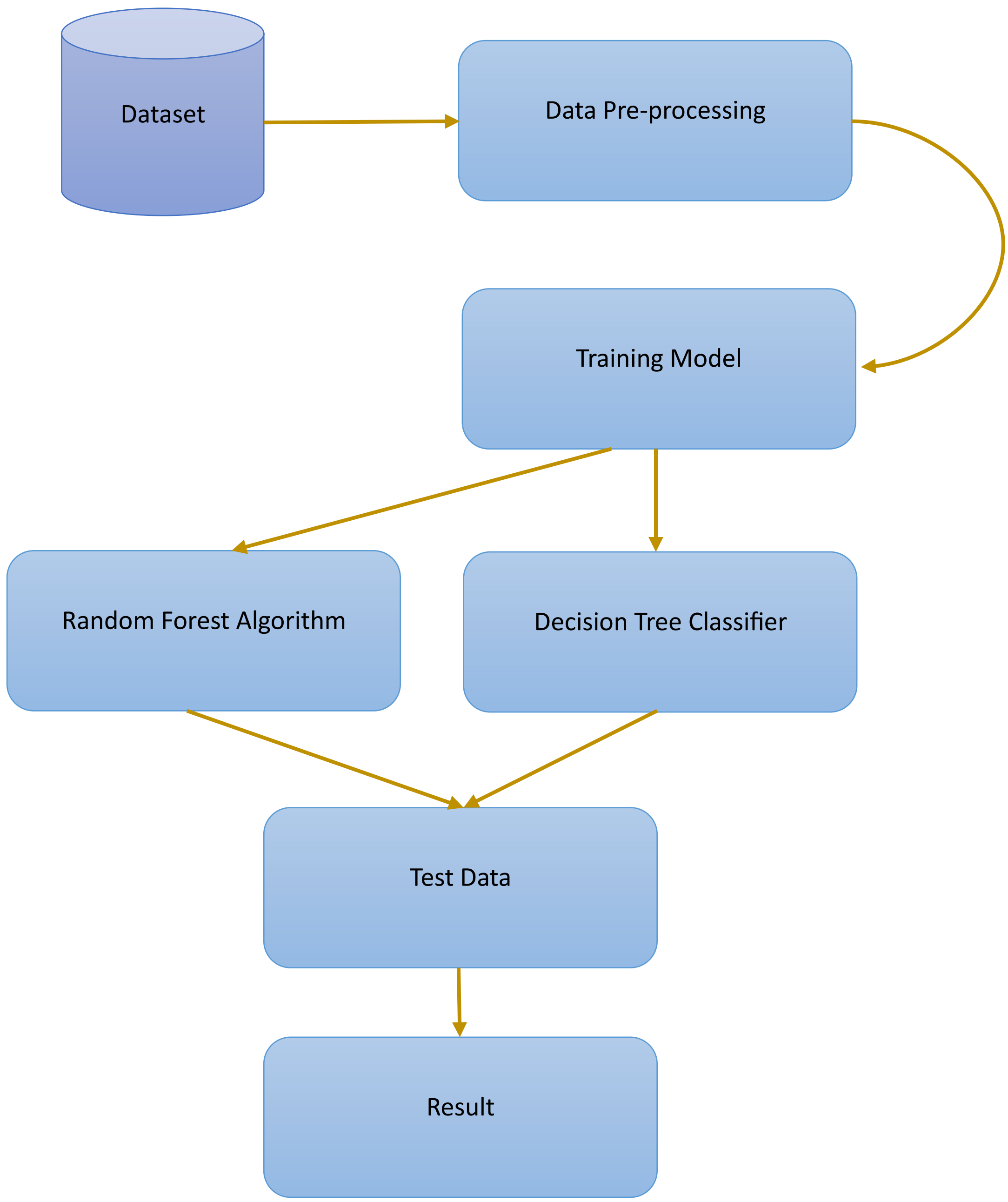
APP.PY

```
import streamlit as st
import pickle
```

```
def main():
    bg = """<div style='background-color:black; padding:13px'>
              <h1 style='color:white'>Loan Approval Prediction GUI</h1>
              <h2 style='color:grey'>Mohammed Aman</h2>
              <h3 style='color:grey'>2020A7PS0050U</h3>
       </div>"""
    st.markdown(background_style, unsafe_allow_html=True)

    col_left, col_right = st.columns((2,2))
    gender_input = col_left.selectbox('Gender', ('Male', 'Female'))
    married_input = col_right.selectbox('Married', ('Yes', 'No'))
    dependents_input = col_left.selectbox('Dependents', ('None', 'One', 'Two',
'Three'))
    education_input = col_right.selectbox('Education', ('Graduate', 'Not
Graduate'))
    self_employed_input = col_left.selectbox('Self-Employed', ('Yes', 'No'))
    applicant_income_input = col_right.number_input('Applicant Income')
    coapplicant_income_input = col_left.number_input('Coapplicant Income')
    loan_amount_input = col_right.number_input('Loan Amount')
    loan_term_input = col_left.number_input('Loan Tenor')
    credit_history_input = col_right.number_input('Credit History', 0.0, 1.0)
    property_area_input = st.selectbox('Property Area', ('Semiurban', 'Urban',
'Rural'))
    classifier_selection = st.selectbox('Classifier Type',('Random
Forest','SVM'))
    predict_button = st.button('Predict')

    if predict_button:
        result = predict(gender_input, married_input, dependents_input,
education_input, self_employed_input, applicant_income_input,
                         coapplicant_income_input, loan_amount_input,
loan_term_input, credit_history_input, property_area_input,
classifier_selection)
        st.success(f'Prediction Result: {result} for the loan')

def predict(gender, married, dependents, education, self_employed,
applicant_income,
            coapplicant_income, loan_amount, loan_term, credit_history,
property_area, classifier_selection):
    selected_classifier = 1 if classifier_selection == 'Random Forest' else 2
    gender_code = 0 if gender == 'Male' else 1
    married_code = 0 if married == 'Yes' else 1
    dependents_code = float(0 if dependents == 'None' else 1 if dependents ==
'One' else 2 if dependents == 'Two' else 3)
    education_code = 0 if education == 'Graduate' else 1
    self_employed_code = 0 if self_employed == 'Yes' else 1
    property_area_code = 0 if property_area == 'Semiurban' else 1 if
property_area == 'Urban' else 2
```

```python
    loan_amount_in_thousands = loan_amount / 1000
    coapplicant_income_in_thousands = coapplicant_income / 1000

    if selected_classifier == 1:
        with open('trained_model_rfc.pkl', 'rb') as pkl:
            model_rfc = pickle.load(pkl)
        prediction = model_rfc.predict([[gender_code, married_code,
dependents_code, education_code, self_employed_code, applicant_income,
                                         coapplicant_income_in_thousands,
loan_amount_in_thousands, loan_term, credit_history, property_area_code]])
    elif selected_classifier == 2:
        with open('trained_model_svm.pkl', 'rb') as pkl:
            model_svm = pickle.load(pkl)
        prediction = model_svm.predict([[gender_code, married_code,
dependents_code, education_code, self_employed_code, applicant_income,
                                         coapplicant_income_in_thousands,
loan_amount_in_thousands, loan_term, credit_history, property_area_code]])
    prediction_result = 'Not Eligible' if prediction == 0 else 'Eligible'
    return prediction_result

if __name__ == '__main__':
    main()
```

Architecture Diagram



Components:

- **Dataset(Dataset Acquisition):**
  - Involves finding a dataset related to loan applications containing features such as income, credit history, employment details, etc. The dataset used here is obtained from Kaggle.
- **Data Pre-processing:**
  - Involves filling missing values(null values), ensuring data quality, data cleaning tasks, etc.
- **Training model:**
  - Consists of training the model using various classification algorithms. We have used Random Forest Classifier, Decision Tree Classifier, k-nearest Neighbour Classifier & Support Vector Classifier here and as for the GUI, we have given the user an option between Random Forest Classifier and Decision Tree Classifier in order to predict their approval status.

- **Test Data/Deployment:**
  - Deploys the trained models for real-time predictions in the loan approval system.

Results:

- **Random Forest Classifier:**
  - Accuracy: 76.39%
  - Precision: 91.58%
  - Recall: 76.99%
  - F1-Score: 83.65%

- **Decision Tree Classifier:**
  - Accuracy: 61.11%
  - Precision: 68.42%
  - Recall: 71.43%
  - F1-Score: 69.89%

- **K-Nearest Neighbour Classifier:**
  - Accuracy: 65.28%
  - Precision: 86.32%
  - Recall: 76.99%
  - F1-Score: 76.64%

- **Support Vector Classifier:**
  - Accuracy: 65.97%
  - Precision: 100.00%
  - Recall: 65.97%
  - F1-Score: 79.50%

Discussion:

- **Random Forest Classifier vs Decision Tree Classifier:**
  - The Random Forest Classifier has proved to be better than the decision tree classifier amongst all evaluation metrics.
  - We can conclude that Random Forest tends to provide more robust and accurate predictions by aggregating multiple decision trees.
- **K-Nearest Neighbour:**
  - This classifier achieved good precision, indicating a low false positive rate.
- **Support Vector Classifier:**
  - Support Vector Classifier achieved perfect precision but has a lower recall.
- **Overall Considerations:**
  - The choice of best model depends on what the user requires between accuracy, precision & recall.

- In the GUI, we have integrated an option for the user to choose between Random Forest Classifier & Support Vector Classifier.

Screenshots of Output:

Importing Dataset

Click here to ask Blackbox to help you code faster

```
df = pd.read_csv("C:/Users/mailm/Downloads/LAP_Dataset.csv")
df.head(5)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y

Pre-processing

Click here to ask Blackbox to help you code faster

```
from sklearn.preprocessing import LabelEncoder
Labelencoder_x=LabelEncoder()
```

✓ 2.9s

Click here to ask Blackbox to help you code faster

```
for i in range(0,5):
    x[:,i]=Labelencoder_x.fit_transform(x[:,i])
x[:,7]=Labelencoder_x.fit_transform(x[:,7])

x
```

```
array([[1, 1, 1, ..., 1.0, 4.852030263919617, 261],
       [1, 1, 0, ..., 1.0, 4.189654742026425, 34],
       [1, 1, 0, ..., 1.0, 4.787491742782046, 185],
       ...,
       [1, 1, 1, ..., 1.0, 5.53338948872752, 338],
       [1, 1, 2, ..., 1.0, 5.231108616854587, 322],
       [0, 0, 0, ..., 0.0, 4.890349128221754, 144]], dtype=object)
```

Splitting into training and testing datasets

Click here to ask Blackbox to help you code faster

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(x,y,test_size=0.3,random_state=0)
```

Click here to ask Blackbox to help you code faster

```
X_train
```

```
array([[1, 0, 0, ..., 1.0, 4.430816798843313, 304],
       [0, 0, 0, ..., 1.0, 4.709530201312334, 91],
       [0, 0, 0, ..., 1.0, 4.382026634673881, 47],
       ...,
       [1, 0, 0, ..., 1.0, 5.497168225293202, 353],
       [1, 1, 0, ..., 1.0, 4.787491742782046, 273],
       [1, 1, 0, ..., 0.0, 4.0943445622221, 7]], dtype=object)
```

DECISION TREE CLASSIFIER

💡 Click here to ask Blackbox to help you code faster

```
from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
dt.fit(X_train,Y_train)
y_pred_dt = dt.predict(X_test)
print("Accuracy of Decision Tree Classifier = ",metrics.accuracy_score(y_pred_dt,Y_test))
print("Precision of Decision Tree  Classifier = ",metrics.precision_score(y_pred_dt,Y_test))
print("Recall of Decision Tree Classifier = ",metrics.recall_score(y_pred_dt,Y_test))
print("F1-Score of Decision Tree  Classifier = ",metrics.f1_score(y_pred_dt,Y_test))
y_pred_dt
```

```
Accuracy of Decision Tree Classifier =  0.6111111111111112
Precision of Decision Tree  Classifier =  0.6842105263157895
Recall of Decision Tree Classifier =  0.7142857142857143
F1-Score of Decision Tree  Classifier =  0.6989247311827957

array([1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1,
       1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1])
```

CONFUSION MATRIX

💡 Click here to ask Blackbox to help you code faster

```
pd.crosstab(Y_test, y_pred_dt, rownames=[''], colnames=[''], margins=True)
```

	0	1	All
0	23	26	49
1	30	65	95
All	53	91	144

RANDOM FOREST CLASSIFIER

💡 Click here to ask Blackbox to help you code faster

```
import joblib
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=80)
rfc.fit(X_train,Y_train)
y_pred_rc = rfc.predict(X_test)
joblib.dump(rfc, 'loan_classifier.pkl')
print("Accuracy of Random Forest Classifier = ",metrics.accuracy_score(y_pred_rc,Y_test))
print("Precision of Random Forest Classifier = ",metrics.precision_score(y_pred_rc,Y_test))
print("Recall of Random Forest Classifier = ",metrics.recall_score(y_pred_rc,Y_test))
print("F1-Score of Random Forest Classifier = ",metrics.f1_score(y_pred_rc,Y_test))
y_pred_rc
```

```
Accuracy of Random Forest Classifier =  0.7638888888888888
Precision of Random Forest Classifier =  0.9157894736842105
Recall of Random Forest Classifier =  0.7699115044247787
F1-Score of Random Forest Classifier =  0.8365384615384616

array([1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1])
```

CONFUSION MATRIX

💡 Click here to ask Blackbox to help you code faster

```
pd.crosstab(Y_test, y_pred_rc, rownames=[''], colnames=[''], margins=True)
```

	0	1	All
0	23	26	49
1	8	87	95
All	31	113	144

K-NEAREST NEIGHBORS CLASSIFIER

💡 Click here to ask Blackbox to help you code faster

```
from sklearn.neighbors import KNeighborsClassifier
kn=KNeighborsClassifier(n_neighbors=10)
kn.fit(X_train,Y_train)
y_pred_kn = kn.predict(X_test)
print("Accuracy of K-Neighbors Classifier = ",metrics.accuracy_score(y_pred_kn,Y_test))
print("Precision of K-Neighbors Classifier = ",metrics.precision_score(y_pred_kn,Y_test))
print("Recall of K-Neighbors Classifier = ",metrics.recall_score(y_pred_rc,Y_test))
print("F1-Score of K-Neighbors Classifier = ",metrics.f1_score(y_pred_kn,Y_test))
y_pred_kn
```

```
Accuracy of K-Neighbors Classifier =  0.6527777777777778
Precision of K-Neighbors Classifier =  0.8631578947368421
Recall of K-Neighbors Classifier =  0.7699115044247787
F1-Score of K-Neighbors Classifier =  0.7663551401869159

array([1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0])
```

CONFUSION MATRIX

💡 Click here to ask Blackbox to help you code faster

```
pd.crosstab(Y_test, y_pred_kn, rownames=[''], colnames=[''], margins=True)
```

	0	1	All
0	12	37	49
1	13	82	95
All	25	119	144

SUPPORT VECTOR CLASSIFIER

💡 Click here to ask Blackbox to help you code faster

```
from sklearn.svm import SVC
svc = SVC()
svc.fit(X_train,Y_train)
y_pred_svc = svc.predict(X_test)
print("Accuracy of Support Vector Classifier = ",metrics.accuracy_score(y_pred_svc,Y_test))
print("Precision of Support Vector Classifier = ",metrics.precision_score(y_pred_svc,Y_test))
print("Recall of Support Vector Classifier = ",metrics.recall_score(y_pred_svc,Y_test))
print("F1-Score of Support Vector Classifier = ",metrics.f1_score(y_pred_svc,Y_test))
y_pred_svc
```

```
Accuracy of Support Vector Classifier =  0.6597222222222222
Precision of Support Vector Classifier =  1.0
Recall of Support Vector Classifier =  0.6597222222222222
F1-Score of Support Vector Classifier =  0.7949790794979079

array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

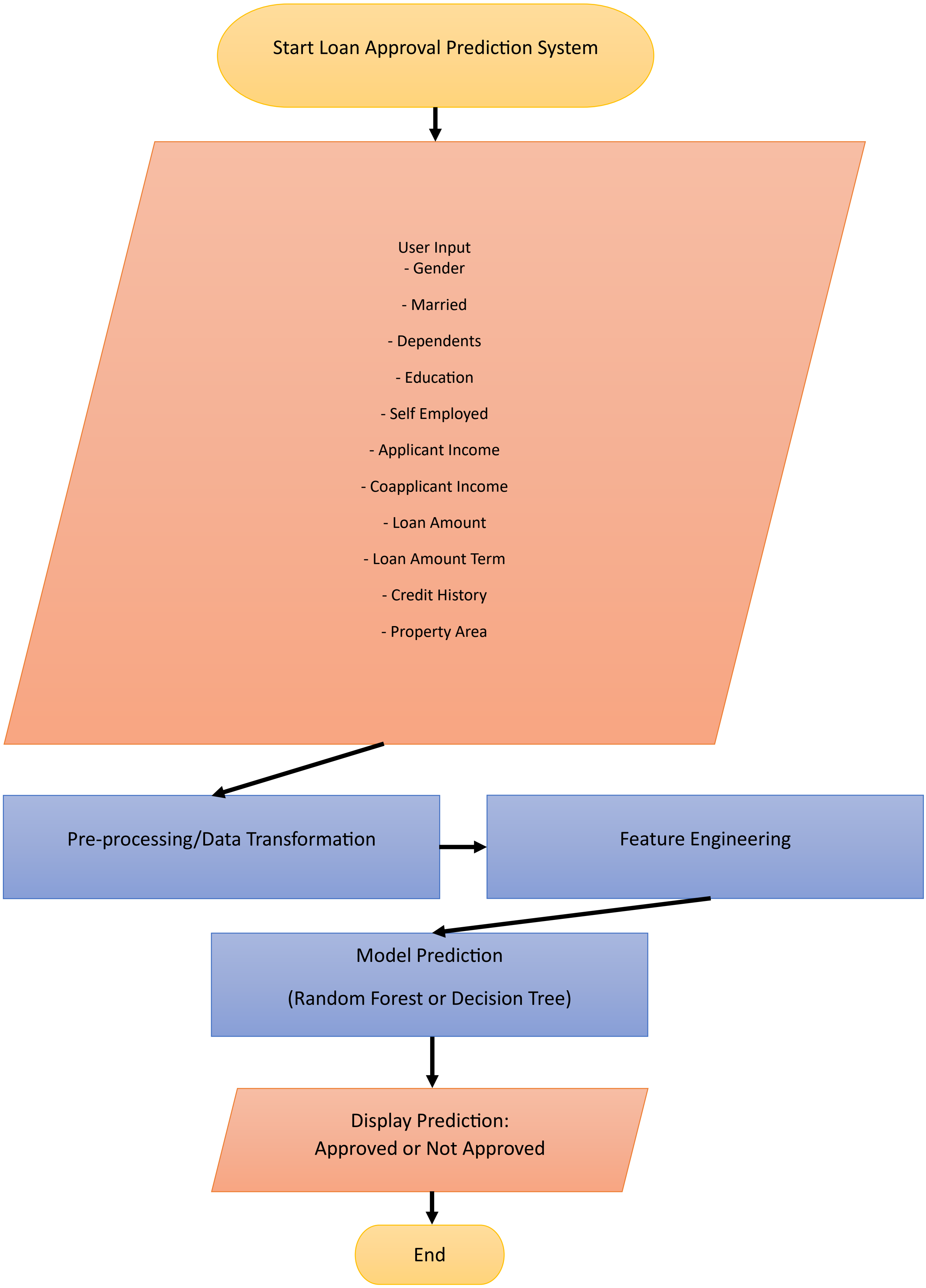
CONFUSION MATRIX

💡 Click here to ask Blackbox to help you code faster

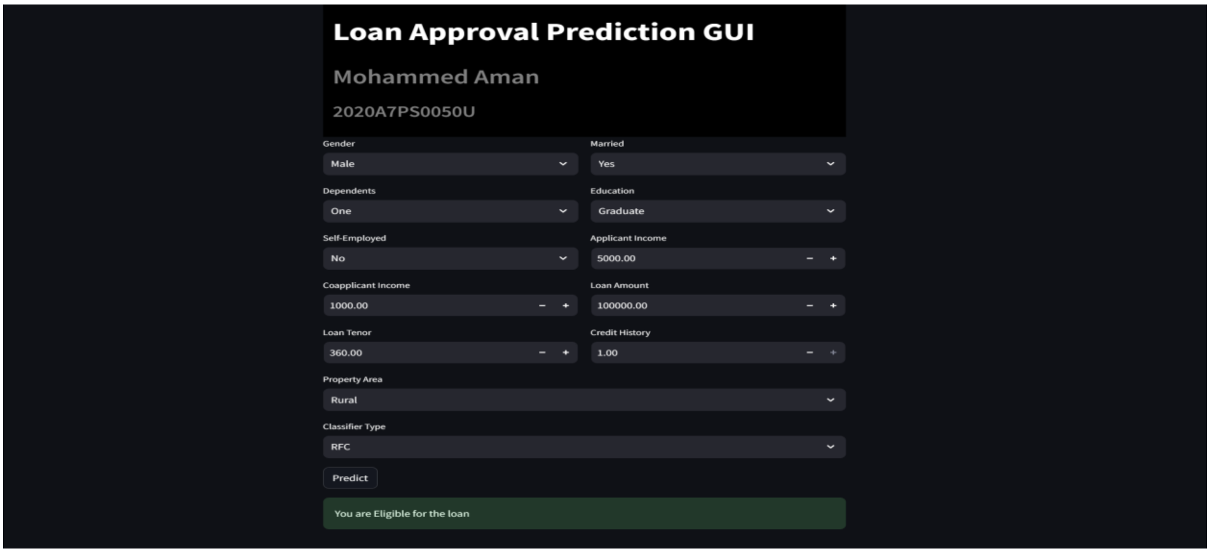
```
pd.crosstab(Y_test, y_pred_svc, rownames=[''], colnames=[''], margins=True)
```

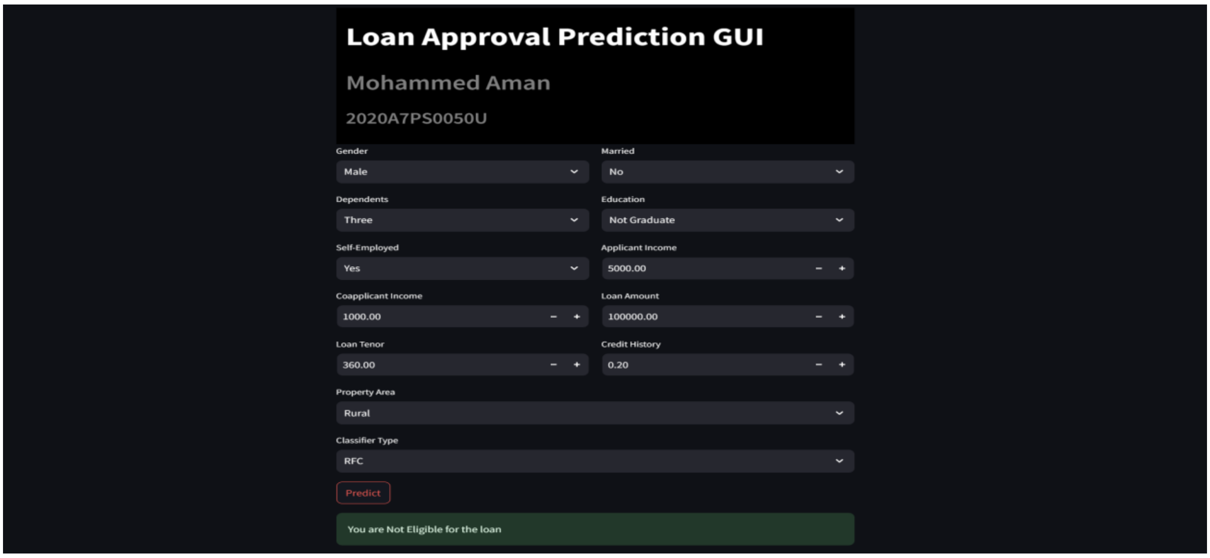
	1	All
0	49	49
1	95	95
All	144	144

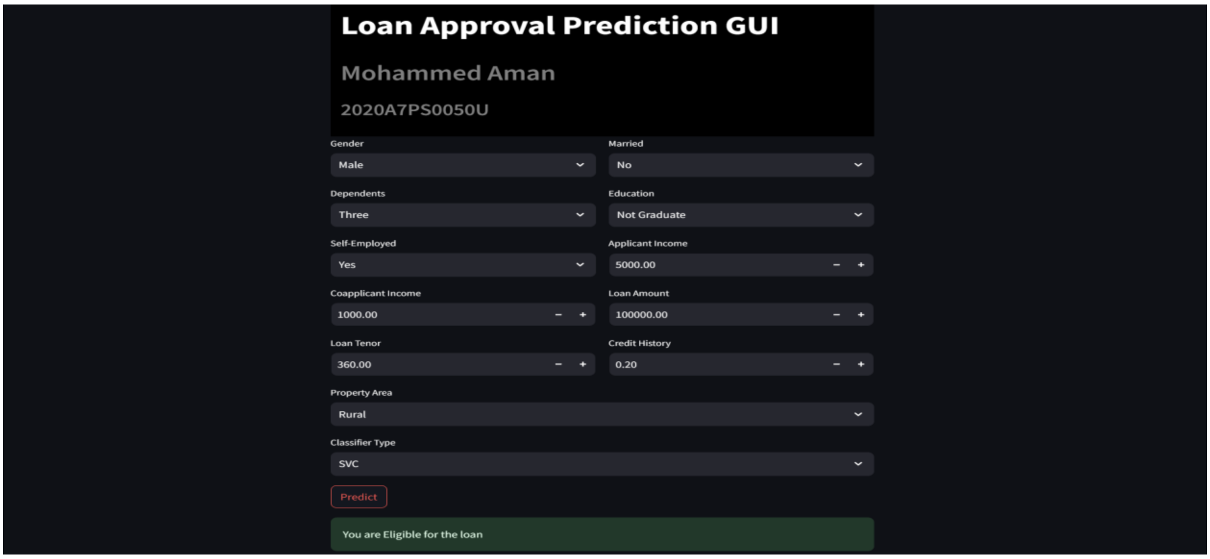
GUI Implementation Flowchart



GUI


Loan Approval Prediction GUI
Mohammed Aman
2020A7PS0050U
Gender
Male
Married
Yes
Dependents
One
Education
Graduate
Self-Employed
No
Applicant Income
5000.00
Coapplicant Income
1000.00
Loan Amount
100000.00
Loan Tenor
360.00
Credit History
1.00
Property Area
Rural
Classifier Type
RFC
Predict
You are Eligible for the loan


Loan Approval Prediction GUI
Mohammed Aman
2020A7PS0050U
Gender
Male
Married
No
Dependents
Three
Education
Not Graduate
Self-Employed
Yes
Applicant Income
5000.00
Coapplicant Income
1000.00
Loan Amount
100000.00
Loan Tenor
360.00
Credit History
0.20
Property Area
Rural
Classifier Type
RFC
Predict
You are Not Eligible for the loan


Loan Approval Prediction GUI
Mohammed Aman
2020A7PS0050U
Gender
Male
Married
No
Dependents
Three
Education
Not Graduate
Self-Employed
Yes
Applicant Income
5000.00
Coapplicant Income
1000.00
Loan Amount
100000.00
Loan Tenor
360.00
Credit History
0.20
Property Area
Rural
Classifier Type
SVC
Predict
You are Eligible for the loan

Conclusion

Applying machine learning to loan approval prediction has shown tremendous potential to improve accuracy, automate systems, and reduce bias. Although there are challenges such as model interpretation, ongoing research and advancements in the field continue to enhance the effectiveness of predictive models. However, I can confidently say that the model created(as displayed above) is a step into making the whole process easier and automating it.

Modifications that could be made in order to improve the project:

- We could use a larger dataset to train the model so that we can accommodate a higher variety of changes.
- We could accommodate more classifier models in order to check higher evaluation metrics to find out one true best classifier method in order to evaluate loan approval status.

References

[1]. *Loan Prediction System Using Machine Learning Anant Shinde , Yash Patil , Ishan Kotian , Abhinav Shinde and Reshma Gulwani.*

[2]. *Machine Learning based Loan Eligibility Prediction using Random Forest Model Chenchireddygari Sudharshan Reddy, Adoni Salauddin Siddiq, N. Jayapandian.*

[3]. *AN EXPLORATORY STUDY BASED ANALYSIS ON LOAN PREDICTION, ANKIT SHARMA*

[4]. *Machine Learning Models for Predicting Bank Loan Eligibility, Ugochukwu .E. Orji, Chikodili .H. Ugwuishiwu, Joseph. C. N. Nguemaleu, Peace. N. Ugwuanyi*

[5]. *LOAN APPROVAL PREDICTION Naga V Vara Prasad Mella , R . Rishitha Sai*

[6]. *AN APPROACH TO LOAN APPROVAL PREDICTION USING MACHINE LEARNING, B. Yamuna, B. Yamuna, D. Sai Nithin, E. Sri Ramya.*

[7]. *LOAN APPROVAL PREDICTION USING SUPERVISED MACHINE LEARNING ALGORITHM Dr.E.Neelima , Venkata Ayyappa Reddy Maruprolu , Sai Prasad Penta , Karthikeya Mallareddy*

[8]. *Customer Loan Eligibility Prediction using Machine Learning Algorithms in Banking Sector Ch. Naveen kumar , D. keerthana , M Kavitha , M Kalyani*

[9]. *Machine Learning Based Model for Prediction of Loan Approval, Bhanu Prakash Lohani, Mayank Trivedi, Ridhik Jeet Singh*

[10]. *PREDICTION OF MODERNIZED LOAN APPROVAL SYSTEM BASED ON MACHINE LEARNING APPROACH, Soumya Ranjan Jena, Vasantha S*

[11]. *AN IMPROVED PERFORMANCE EVALUATION OF CLASSIFICATION ALGORITHM FOR PREDICTION OF LOAN APPROVAL 1A.U. Farouk, A. 2Abdulkadir, A.Y. Abarshi*

[12]. *Customer Loan Eligibility Prediction using Machine Learning, Amjan Shaik, Kunduru Sai Asritha, Neelam Lahre*

[13]. *Predicting Bank Loan Eligibility Using Machine Learning Models and Comparison Analysis ,Miraz Al Mamun*

[14]. *EXPLORATORY DATA ANALYSIS FOR LOAN PREDICTION Vinuthna Ramavarapu, Bali Satya Durga, Basu Sindhu Devi, Chippada Satya Pravallika, Ms. D. Lalitha Kumari*

[15]. *Credit Card Approval Prediction by Using Machine Learning Techniques, M. P. C. Peiris*

No.	Reference	Objective	Problem Statement	Dataset	Algorithm	Performance Measure Value
1	*Loan Prediction System Using Machine Learning Anant Shinde , Yash Patil , Ishan Kotian , Abhinav Shinde and Reshma Gulwani.*	The most reason of this plan is to guess which clients will be reimbursed with an advance because the moneylender expect the issue that the borrower won't be appropriate to reimburse the danger.	A calculated relapse demonstrate has been executed The most elevated gotten with unique dataset is 0.811. Models are compared based on execution estimations such as affectability and specificity. As a result of dissecting, the taking after conclusions	Fannie Mae Single-Family Loan Performance Data	Logistic Regression using stratified k-fold cross validation and Random Forest	75.08
2	*Machine Learning based Loan Eligibility Prediction using Random Forest Model Chenc hireddygari Sudharshan Reddy, Adoni Salauddin Siddiq, N. Jayapandian.*	The proposed model introduces random forest machine learning model, it is really wanting to transfigure every one of the straight out factors into figures. That involving the LabelEncoder in presenting the wholeness of the factors have figures that our models can comprehend.	A credit isn't limited to a single sort since it covers all sorts of scenarios empowering by and large scope for the expectation of the securing of advances. Each method has numerous steps and a diagram for the endorsement of advances. The primary step begins with the client. In order for the advance to urge prepared, the client must apply for the advance sort based on the necessity.	Personal dataset with 1000 entries	Random Forest Algorithm	Accuracy rate ranging from 76% to much more above 80%
3	*An Exploratory Study Based Analysis On Loan*	The interest rate, together with other factors (such as the payback period), assesses the	The consumer first qualifies for a house loan, and the firm then verifies the customer's loan	In our feed forward Neural Network Model, the dataset that we used consist of 5000 clients or consumers(personal dataset)	Logistic Regression, SVM, J48, KNN and tree model	The neural network model's accuracy rate within the

	Prediction, Ankit Sharma	borrower's riskiness; the higher the rate of interest, the riskier the consumer. Depending upon the interest rate, we will determine if the applicant is eligible for the loan.	eligibility. The firm seeks to automate (in real-time) the loan qualifying procedure based on information supplied by customers while completing out online application forms.			normal cutoff hit 98 percent
4	*Machine Learning Models for Predicting Bank Loan Eligibility, Ugochukwu .E. Orji, Chikodili .H. Ugwuishiwu, Joseph. C. N. Nguemaleu, Peace. N. Ugwuanyi*	The authors aimed to show the various ML algorithms utilized by researchers for credit assessment of rural borrowers, especially those with inadequate loan history. Their finding showed that ML algorithms we utilized in this research were widely used and showed great results.	In their research, deployed various ensemble ML techniques such as AdaBoost, LogitBoost, Bagging, and Random Forest Model to predict loan approval of bank direct marketing data.	‘Loan Eligible Dataset’, available on Kaggle	Random Forest, Gradient Boost, Decision tree, Support Vector Machine, K-Nearest Neighbour, & Logistic Regression	Accuracy ranged from 80% to 76% respectively
5	*Loan Approval Prediction Naga V Vara Prasad Mella , R . Rishitha Sai*	This paper aims to detect users generating spam reviews or review spammers. We identify several characteristic behaviors of review spammers and model	In the proposed system, each review goes through a tokenization process first. Then, unnecessary words are removed and candidate feature words are generated. Each candidate feature words are checked against the dictionary and if	Amazon Review Dataset	TF-IDF, SVM Algorithm, Naïve Bayes Algorithm, Decision Tree Algorithm.	Prediction using Logistic Regression accuracy is 82.7%

		these behaviors so as to detect the spammers.	its entry is available in dictionary.			
6	*An Approach To Loan Approval Prediction Using Machine Learning, B. Yamuna, B. Yamuna, D. Sai Nithin, E. Sri Ramya.*	Build an ML application which can reduce the time required to approve a loan using ML based prediction model to approve the loan with minimal human intervention by filtering huge number of applications and forward very few applications for human verification.	By calculating the possibility of loan escape, the ideal customers to target for loan giving will be easily identified using a XGBoost model approach.	Kaggle dataset with 1010 entries.	Logistic Regression, Decision tree, Support Vector Machine, Random Forest Tree, and XGB	XGBoost Model – 0.82
7	*Loan Approval Prediction Using Supervised Machine Learning Algorithm Dr.E.Neelima , Venkata Ayyappa Reddy Maruprolu , Sai Prasad Penta , Karthikeya Mallareddy*	We strive to limit the uncertainty in the back of opting the authentic individual so that we can minimize the Bank's Human Resource, can tightly closed Banks assets, can decrease the length of mortgage get sanction.	This proposed method is highly preferred among the image with dynamic variations. The technique used in the paper is evaluated using 4500 instance of the MRI and 3000 instance of CT.	Google’s KAGGLE	Classification, Logic Regression, Decision Tree and Gradient Boosting.	All models can achieve upto 80% accuracy. The highest accuracy is 93%

8	*Customer Loan Eligibility Prediction Using Machine Learning Algorithms In Banking Sector Ch. Naveen Kumar , D. Keerthana , M Kavitha , M Kalyani*	Customer data is collected based on various banks and accessing the customer profiles to analyze the data based on the parameters which are essential to integrate with the machine learning techniques	Analyze the data and provide the results based on the customer profile to grant the loans using machine learning approach is more advanced than traditional loan approval-based systems	Dataset from public repository	Decision tree, Random Forest, Support Vector Machine, K – nearest neighbour, and decision tree with adaboost	78 to 82% on forest model
9	*Machine Learning Based Model For Prediction Of Loan Approval, Bhanu Prakash Lohani, Mayank Trivedi, Ridhik Jeet Singh*	In this study we have applied logistic regression as a tool to predict whether an applicant is eligible for the loan or not. The data is collected from the Kaggle for studying and prediction.	The logistic regression approach is used to forecast the loan's safety. Logistic regression is a type of supervised machine learning algorithm which uses labelled dataset	Kaggle and includes applicants of various ages and genders. Education, marital status, income, and other characteristics are included in the dataset.	Supervised Learning, unsupervised learning, reinforcement learning	75%
10	*Prediction Of Modernized Loan Approval System Based On Machine Learning Approach, Soumya*	The main objective of this paper is to predict whether a new applicant granted the loan or not using machine learning models trained on the historical data set.	The main objective of this paper is to predict whether a new applicant granted the loan or not using machine learning models trained on the historical data set.a The experimental results conclude that the accuracy of Decision Tree machine	Kaggle Dataset	SVM, LR, Random forest	85% for Random Forest (Highest)

	Ranjan Jena, Vasantha S		learning algorithm is better as compared to Logistic Regression and Random Forest machine learning approaches.			
11	*An Improved Performance Evaluation Of Classification Algorithm For Prediction Of Loan Approval 1a.U. Farouk, A. 2abdulkadir, A.Y. Abarshi*	The study aimed at evaluating the performance of five classification models used in predicting loan approval.	Study concluded that Naïve Bayes is the best algorithms for predicting loan approval.	The dataset used consist of 12 variables and 689 data instances	Random forest,naïve bayes, LR	Random Forest-83.2% and 79.2% Logistic-81.6% and 73.%
12	*Customer Loan Eligibility Prediction Using Machine Learning, Amjan Shaik, Kunduru Sai Asritha, Neelam Lahre*	Main goal is to determine which machine algorithm performs best at predicting whether b person is qualified for a loan.	Mining the Big Data belonging to the previously loan issued individuals, and based on this data, machine learning models were taught to produce the most accurate results.	Thirteen aspects make up the dataset we're using: Loan ID, Gender, Married, Dependents, Education,etc.	Adaboost classifier, Passive Aggressive Classifier, Random forest	73% to 78%
13	*Predicting Bank Loan Eligibility Using Machine Learning Models And Comparison Analysis*	(ML) algorithms are employed to extract patterns from a common loan-approved dataset and predict deserving loan applicants. Customers' previous data will be used to	The training data set was provided to the machine learning model, and the model was trained using that data set. Every new applicant's information entered on the application form serves as a test data set. In this paper, they used	Bank of Kigali	Random Forest, XGBoost, Adaboost, Lightgbm, Decision tree, aand K-nearest Neighbour	Highest accuracy of 92%

	,Miraz Al Mamun	undertake the study, including their age, income type, loan annuity, last credit bureau report, type of organization they work for, and length of employment.	three machine learning methods to predict client loan approval.			
14	*Exploratory Data Analysis For Loan Prediction Vinuthna Ramavarapu, Bali Satya Durga, Basu Sindhu Devi, Chippada Satya Pravallika, Ms. D. Lalitha Kumari*	The aim of this project is to provide a quick, immediate and easy way to choose the deserving applicants.	Loan Prediction System can automatically calculate the weight of each feature field taking part in loan processing and compares the new data with its associated weight	Personal Dataset	Random Forest	83%
15	*Credit Card Approval Prediction By Using Machine Learning Techniques, M. P. C. Peiris*	This research is focusing on application of machine learning (ML) techniques to predict customer eligibility for a credit card.	Many researchers have conducted machine learning applications on credit scoring and customer default predictions. Researchers' have concluded that SVM (support vector machine) and ANN (Artificial Neural Network) performed better than other classifiers	UCI Repository credit card defaulter	SVM, ANN	90% correctly