

MACHINE LEARNING APPROACHES FOR PREDICTING LOAN APPROVAL USING CHAT BOTS

**Mr. D.Ravi Kumar¹, Rohith Maddikunta², Abhinav Chella³, Rohith Myadaraboina⁴,
Sahitya Kanchanapalli⁵**

¹Assoc.Professor CSE De Pt Ace Engineering College Hyderabad, India

^{2,3,4,5}CSE Ace Engineering College Hyderabad, India

ABSTRACT

The project's objective is to develop a chatbot that uses machine learning to predict loan approval. The subject of this discussion is chat bots, which are smart frameworks that can comprehend a user's everyday dialect queries and respond accordingly in a conversation. at client care centers, and in areas handling inquiries, humans are deficient and often take a long time to handle a single query, which results in wastage of time and reduces customer service quality.

The main objective of this chatbot is that clients can connect by saying their inquiries in plain English, and the chatbot can resolve their questions with a fitting response in return. The suggested framework would provide assistance and mimic the client's advantage of involvement; however, the customer would be connected to a bot rather than a real person and would still receive the questions answered and the issues fixed. This paper explains the dataset that we have prepared from FAQs(Frequently Asked Questions) on bank websites, and how machine learning has been used to develop a trained model that can predict whether the client is eligible for loan or not.

1. INTRODUCTION

Nowadays Banks play a crucial role in the economy by providing financial resources to individuals, businesses, and various other entities. The distribution of loans is one of their main sources of income and asset management. Their ability to lend money to borrowers and receive interest as payment is a key component of their commercial activities. However, it can be a complicated and difficult task for banks to safeguard their cash and make informed loan approval decisions. Extensive documentation, manual evaluations, and arbitrary judgement are frequently part of the typical loan approval procedure.

It may take a while requiring substantial human effort and expertise. Furthermore, even with all of the bank employees' best efforts, the process might not always identify the worthiest candidates or correctly anticipate the potential hazards connected with each loan application. Banks have started using machine learning methods to address these issues and improve the effectiveness of the loan approval process. The goal of machine learning, a branch of artificial intelligence, is to create models and algorithms that can infer conclusions from data without explicit programming. Exploratory Data Analysis (EDA) is a crucial step in the machine learning pipeline. It involves examining and visualizing the loan application data to gain insights and understand patterns, relationships, and distributions within the dataset.

EDA helps in identifying outliers, missing values, and other data quality issues that may impact the accuracy of the loan approval decisions. Another crucial component of machine learning in the loan approval process is feature engineering. It entails choosing, modifying, and producing pertinent information from the dataset of loan applications. The machine learning algorithms use these features as inputs to help them learn and anticipate the future. Feature Engineering requires domain expertise and an understanding of the loan approval process to extract meaningful information from the raw data.

Logistic Regression is a commonly used machine learning algorithm in the context of loan approval. It is a statistical method that predicts the probability of a binary outcome, such as whether a loan application is safe or not. Logistic Regression considers various features and their weights to estimate a loan default or other risks. By training the model on historical loan data, the algorithm can learn patterns and relationships that can be used to assess the safety of new loan applications.

The use of machine learning techniques in the loan approval process can benefit financial institutions in a number of ways. First off, it makes manual assessment less time- and labor-intensive, enabling banks to handle loan applications more swiftly and effectively. By responding to loan requests more quickly, this efficiency can raise client satisfaction. Second, machine learning algorithms can quickly and accurately analyze huge amounts of loan application data, finding patterns and connections that human analysts might miss. Automating feature validation allows machine learning models to evaluate each loan application's risk objectively, resulting in more reliable and consistent loan approval decisions.

2. OBJECTIVES

This paper aims to comprehensively explore and evaluate the concept of a chatbot that uses machine learning to predict loan approval. The loan approval process is a crucial part of how banks operate, and implementing machine learning strategies, like chatbot-based methods, can make it more effective.

Through the loan approval process, chatbots that are powered by machine learning algorithms can greatly simplify and automate consumer interactions. By incorporating chatbot technology into the loan approval procedure, banks may offer 24/7 customer support while quickly resolving their questions and concerns.

Inquiries from customers about loan eligibility, documentation needs, interest rates, and repayment conditions can all be handled by chatbots. These virtual assistants can make use of machine learning and natural language processing to comprehend consumer intent, deliver pertinent information, and assist customers with the loan application process.

3. PROBLEM STATEMENT

Nowadays Banks play a crucial role in the economy by providing financial resources to individuals, businesses, and various other entities. The distribution of loans is one of their main sources of income and asset management.

Their ability to lend money to borrowers and receive interest as payment is a key component of their commercial activities. However, it can be a complicated and difficult task for banks to safeguard their cash and make informed loan approval decisions.

Extensive documentation, manual evaluations, and arbitrary judgement are frequently part of the typical loan approval procedure. It may take a while requiring substantial human effort and expertise. Furthermore, even with all of the bank employees' best efforts, the process might not always identify the worthiest candidates or correctly anticipate the potential hazards connected with each loan application.

4. PROPOSED SYSYTEM

In the financial sector, predicting whether a loan application will be approved is a crucial task that can be successfully solved as a classification issue.

An automatic loan prediction system can be created in this regard using a variety of machine learning approaches, such as Exploratory Data Analysis (EDA), Random Forest, K-Nearest Neighbors (KNN), Learning Vector Quantization (LVQ), and K-Fold cross validation. A wide range of categorization techniques are used to address this issue. By examining the data's features, spotting patterns, and spotting abnormalities, EDA plays a significant part in comprehending the data.

This preliminary study offers important insights into the loan application information, allowing us to make wise choices while building the model.

The automatic loan prediction system's foundation is built on a combination of various categorization algorithms and methods.

This system can create precise predictions for future loan applications by analyzing historical loan data, identifying trends, and learning from previous loan decisions.

Advantages of Proposed System:

- **Improved Efficiency:** By automating the loan approval process, the system reduces the time and effort required for manual review and decision-making
- **Improved Customer Experience:** The automatic loan prediction system offers a seamless and efficient loan application process for customers. They receive prompt responses on their loan eligibility and can make informed decisions based on the prediction outcome.
- **Consistent Decision-Making:** The loan prediction system follows predefined rules and algorithms consistently for evaluating loan applications.
- **Cost Savings:** Automating the loan approval process reduces the need for extensive manual review and evaluation, thereby reducing operational costs.

OBJECTIVES:

- It helps in increase of loan application submissions
- It helps to reduce loan processing time
- It helps in providing accurate loan information
- It also helps to provide better security and authentication methods

5. HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

- Processor – Intel core i5
- RAM – 8 GB (min)
- Hard Disk – 1 TB
- Key Board – Standard Windows Keyboard
- Mouse – Two or Three Button Mouse
- Monitor – SVGA

SOFTWARE REQUIREMENTS:

- Operating system – Windows 7, 8, 10, 11, mac os
- Coding Language –Python, HTML, NodeJS, CSS
- Back-End – Server side Database, VS Code, python flask
- Designing – HTML, CSS.

6. TECHNOLOGY DESCRIPTION PYTHON FLASK

A lightweight and adaptable web framework called Python Flask enables programmers to create web apps fast and effortlessly. Based on the Python programming language, Flask adheres to a minimalistic design aesthetic that emphasizes clarity and extensibility. It is a popular option for building small to medium- sized web applications because it offers the required tools and packages for building web APIs (application programming interfaces), managing routing, rendering templates, and managing sessions.

The Python package installer, pip, can be used to complete the simple process of installing Python Flask. Make sure Python is installed on your system before installing Flask. Both Python 2.x and Python 3.x versions work with Flask. Python 2.x has reached the end of its support life, hence Python 3.x is advised for new projects.

7. ALGORITHM

The steps involved in building the model are:

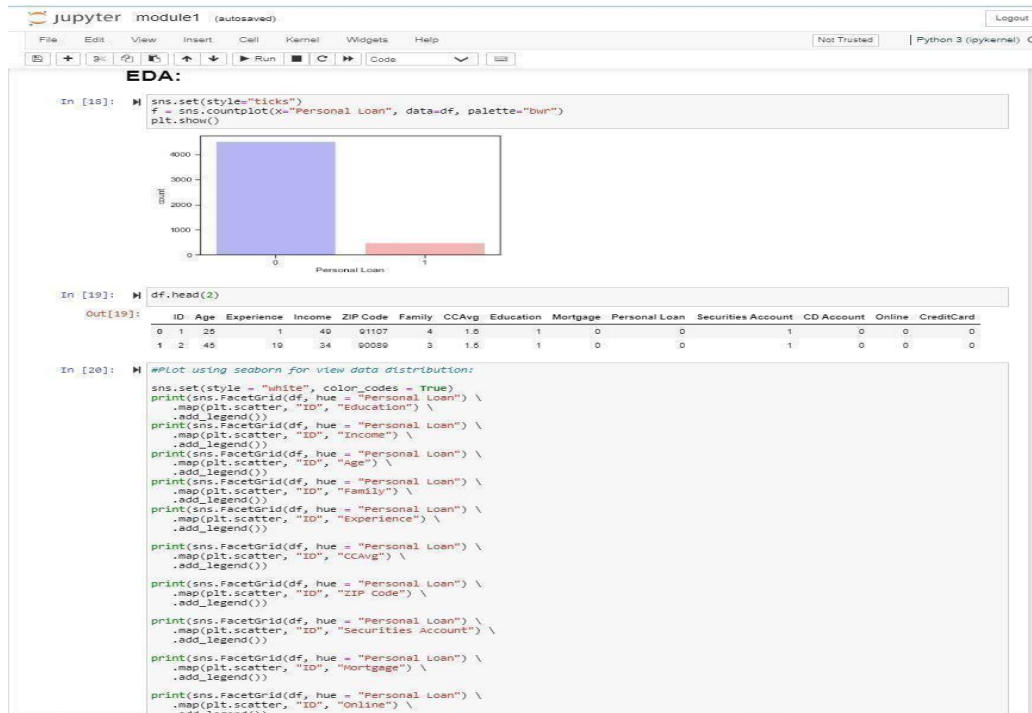
Generally, to apply for loan, people have to wait in long queue just to get their turn. Usage of chatbots helps in increase of loan application submissions in fastest and easiest way as the people can clarify their doubts and apply for loan through online.

This allows customers to complete the entire process without waiting on the phone. It helps to reduce loan processing time and provide accurate information by training the chatbot. Chatbots provide users with automated checklists of all the papers they need to provide and reduce defaults. Users can scan and submit documents while the lenders can verify these documents with AI in KYC.

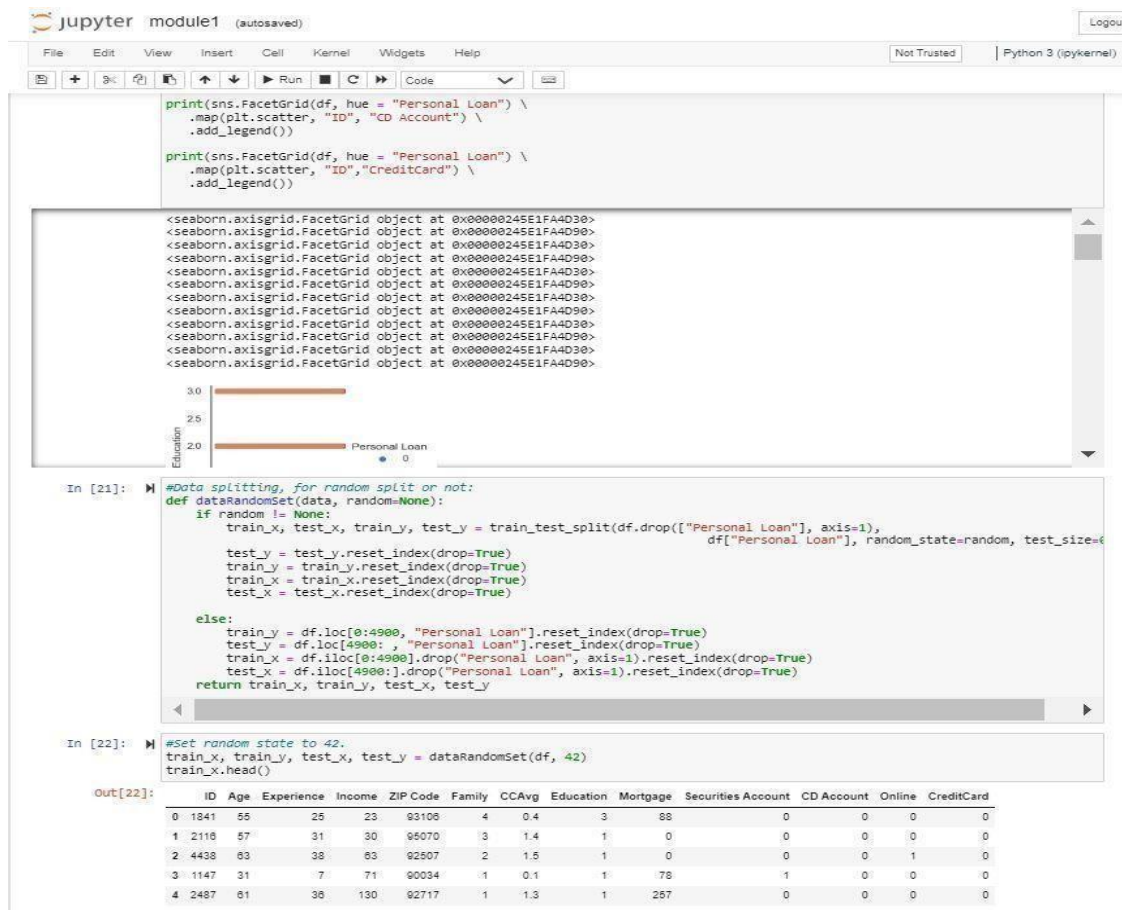
If any documents fall short, the lenders can follow up with the user and get the required documents. Chatbots helps us to communicate with customers more efficiently and effectively than traditional methods.

1. Data Collection: Gathering data on loan applications is the initial phase, which includes information on things like credit score, income, loan amount, and other pertinent aspects. The information can be gathered from a variety of places, including databases maintained by financial organizations or online loan application forms.
2. Data Cleaning and Preprocessing: Once the data is collected, it needs to be cleaned and preprocessed to remove any missing values, duplicates, or outliers. This step ensures that the data is of high quality and ready for analysis.
3. Feature Selection: The next step is to select relevant features that will be used to predict loan approval. This step involves analysing the correlation between unique features and loan approval and selecting the most relevant ones.
4. Model Training: Using classification methods like Logistic Regression, Decision Tree, Random Forest Classifier, and Support Vector Machine, the chosen features are used to train the loan prediction model. These algorithms create a model from the training data that accurately forecasts loan approval based on the chosen features.
5. Model Evaluation: To make sure the model is accurate and dependable after it has been trained, it must be assessed. In this step, the model's performance is assessed using testing data, and the projected outcomes are contrasted with the actual loan approval outcomes.
6. Model Deployment: The last step is to deploy the loan prediction model in a chatbot or other application, allowing users to input their loan application information and receive personalized loan approval predictions.

8. EXPLORATORY DATA ANALYSIS



EDA is a crucial phase in the data analysis process that aids in understanding and condensing the key elements of a dataset. It is an essential method for statisticians and data scientists since it enables them to investigate the data, spot trends, and develop hypotheses. It helps to find outliers, missing numbers, and data problems that might have a big impact on the outcomes of further analysis. In order to find underlying links, patterns, and trends, EDA entails studying the data using various graphical and statistical tools. Box plots, correlation matrices, histograms, scatter plots, and histograms are a few of the common methods used in EDA.



K-nearest neighbors:

A supervised machine learning (ML) technique known as K-nearest neighbors (KNN) can be applied to classification and regression predicting issues. Its foundation is the idea that items or data points near to one another in a feature space are more likely to belong to the same class or have the same goal value. However, it is utilized in industry for classification and forecasting issues.

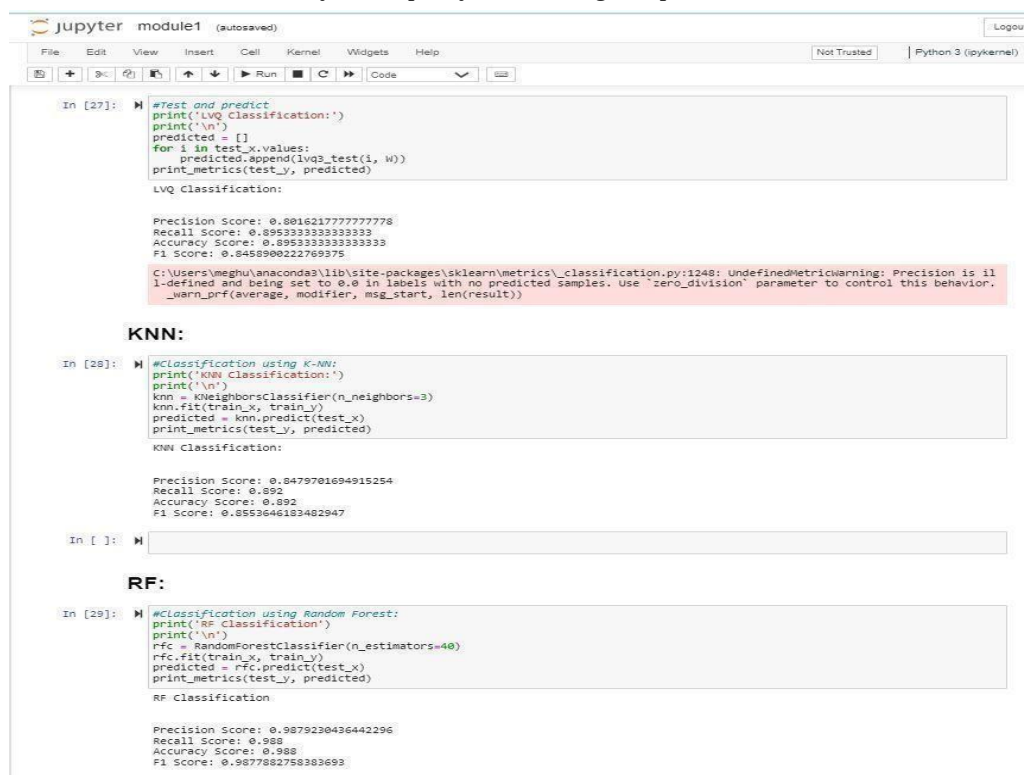
An essential KNN parameter is K selection. The algorithm may be susceptible to noise and outliers if K is too low, and it may overlook significant patterns in the data if K is too high. KNN has the potential to be computationally expensive, particularly for large datasets. This is due to the fact that KNN must determine the distance between the new data point and each of the previously stored data points

Random Forest

Random forest is a type of supervised machine learning algorithm which combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest".

The random forest algorithm can be used for both regression and classification tasks. An approach for machine learning called Random Forest is used to solve classification and regression issues. Decision trees serve as its foundation, and numerous decision trees are combined to increase accuracy.

Because of its great accuracy and resistance to overfitting, Random Forest is well-known. This is due to the fact that each decision tree is created using a random subset of characteristics and samples, which lowers correlation between the trees and enhances generalization. Overall, Random Forest is a strong and flexible method that may be applied to a variety of Python machine learning issues. It is a preferred option for many data scientists and machine learning professionals due to its excellent accuracy and capacity for handling complex data.



```
jupyter module1 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted Python 3 (ipykernel)

In [27]: #Test and predict
print('LVQ Classification:')
print('\n')
predicted = []
for i in test_x.values:
    predicted.append(lvq3_test(i, W))
print_metrics(test_y, predicted)

LVQ Classification:

Precision Score: 0.8016217777777778
Recall Score: 0.8953333333333333
Accuracy Score: 0.8962333333333333
F1 Score: 0.8458900222769375

C:\Users\meghu\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

KNN:

In [28]: #Classification using K-NN:
print('KNN Classification:')
print('\n')
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(train_x, train_y)
predicted = knn.predict(test_x)
print_metrics(test_y, predicted)

KNN Classification:

Precision Score: 0.8479701694915254
Recall Score: 0.892
Accuracy Score: 0.892
F1 Score: 0.8553646183482947

In [ ]:

RF:

In [29]: #Classification using Random Forest:
print('RF Classification:')
print('\n')
rfc = RandomForestClassifier(n_estimators=40)
rfc.fit(train_x, train_y)
predicted = rfc.predict(test_x)
print_metrics(test_y, predicted)

RF Classification:

Precision Score: 0.9879230436442296
Recall Score: 0.989
Accuracy Score: 0.988
F1 Score: 0.9877882758383693
```

Coding:

```
from flask import Flask, redirect, url_for, render_template, request
from flask_restful import Resource, Api, reqparse
import pandas as pd
import ast
from flask_cors import CORS
import json
from googletrans import Translator
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer
```



```
import os
import time
import joblib
import warnings
import numpy as np
warnings.filterwarnings(action='ignore')
app = Flask(__name__)
app.secret_key = 'chatbot'
CORS(app)
api = Api(app)
app.static_folder = 'static'
def prediction(X):
    clf = joblib.load('Best_Model.pkl')
    label = clf.predict([X])
    out = "
    if label[0] == 1:
        out = 'customer is eligible for loan'
        print('customer is eligible for loan')
    else:
        out = 'customer is not eligible for loan'
        print(out)
    return out
translator = Translator()
d = {
    'hello': 'hello',
    'how are you': 'I am good',
    'I am good': 'how can I help you?',
    'I am fine': 'how can I help you?'
}
chatbot = ChatBot('loan bot')
trainer = ChatterBotCorpusTrainer(chatbot)
trainer.train("chatterbot.corpus.english.greetings", "chatterbot.corpus.english.conversations")
def index():
    return render_template('index.html')
l1 = [1502, 30, 4, 35, 92130, 2, 0.3, 2, 0, 1, 0, 0, 1]
rr = prediction(l1)
print('test', rr[0])
loan_flag = False
loan_qus = {
    'ID': 0,
    'Age': 0,
    'Experience': 0,
    'Income': 0,
    'ZIP Code': 0,
    'Family': 0,
    'CCAvg': 0,
    'Education': 0,
```

```
'Mortgage': 0,
'Securities Account': 0,
'CD Account': 0,
'Online': 0,
'CreditCard': 0
}
loan_count = 0
l = []
lang = ""
my_lang = 'en'
class process(Resource):
def get(self):
global loan_flag, loan_count, loan_qus, l, my_lang, lang, app
loan = []
parser = reqparse.RequestParser() # initialize parser
parser.add_argument('task', required=True) # add args
args = parser.parse_args() # parse arguments to dictionary
print('type', type(args))
print('task', args['task'])
task = str(args['task'])
print('loan_flag', loan_flag)
text = ""
res = ""
text = task
if loan_flag == False:
detection = translator.detect(task)
print("Language code:", detection.lang)
if detection.lang == 'kn':
lang = 'k'
result = translator.translate(task, src='kn', dest='en')
text = result.text
print('user', text)
response = chatbot.get_response(text)
print('bot', response.text)
result = translator.translate(response.text, src='en', dest='kn')
res = lang + str(response.text)
print(res)
my_lang = 'kn'
if 'loan' in text or 'cash' in text or 'money' in text or 'financial' in text:
loan_flag = True
else:
lang = 'e'
text = task
print('user', text)
response = chatbot.get_response(text)
print('bot', response)
print('>>', response.text)
```

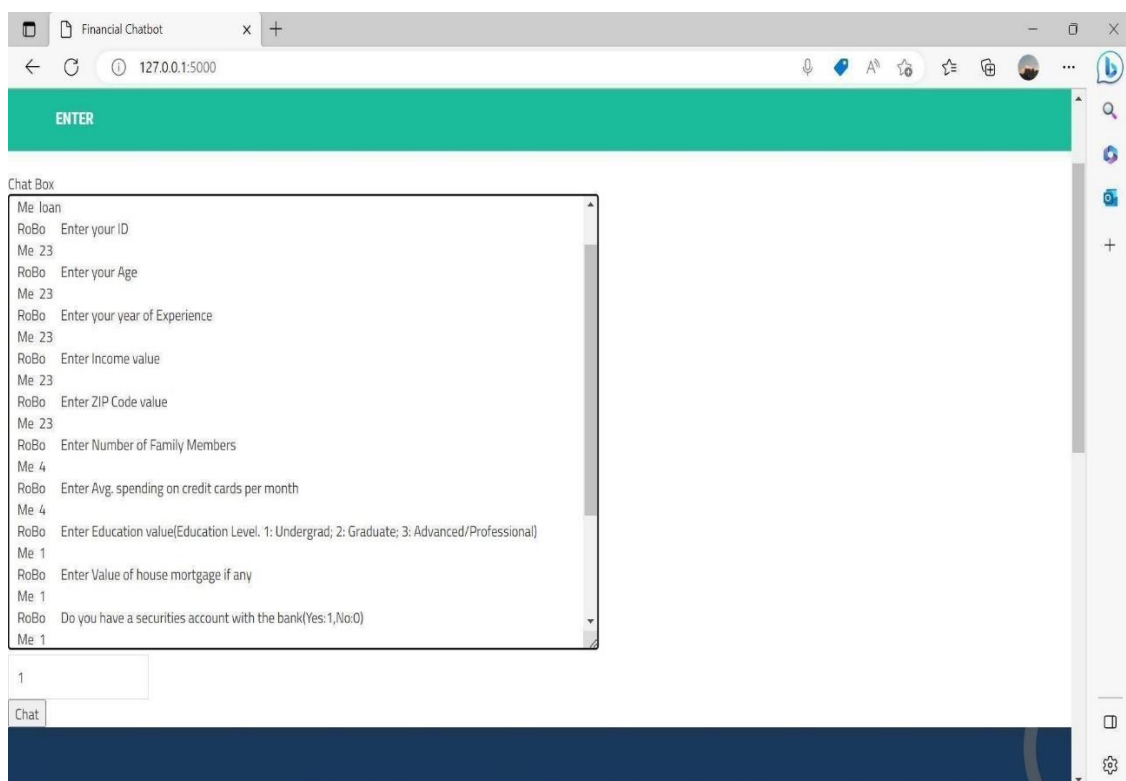
```
res = lang + str(response.text)
if 'loan' in text or 'cash' in text or 'money' in text or 'financial' in text:
    loan_flag = True
if 'loan' in text or 'cash' in text or 'money' in text or 'financial' in text or loan_flag == True:
    print('Ask loan parameter', loan_count)
    loan_flag = True
if loan_count == 0:
    res = lang + 'Enter your ID'
    loan_count = loan_count + 1
elif loan_count == 1:
    res = lang + 'Enter your Age'
    loan_count = loan_count + 1
elif loan_count == 2:
    res = lang + 'Enter your year of      Experience'
    loan_count = loan_count + 1
elif loan_count == 3:
    res = lang + 'Enter Income value'
    loan_count = loan_count + 1
elif loan_count == 4:
    res = lang + 'Enter ZIP Code value'
    loan_count = loan_count + 1
elif loan_count == 5:
    res = lang + 'Enter Number of Family Members'
    loan_count = loan_count + 1
elif loan_count == 6:
    res = lang + 'Enter Avg. spending on credit cards per month'
    loan_count = loan_count + 1
elif loan_count == 7:
    res = lang + 'Enter Education value(Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional)'
    loan_count = loan_count + 1
elif loan_count == 8:
    res = lang + 'Enter Value of house mortgage if any'
    loan_count = loan_count + 1
elif loan_count == 9:
    res = lang + 'Do you have a securities account with the bank(Yes:1,No:0)'
    loan_count = loan_count + 1
elif loan_count == 10:
    res = lang + 'Do you have a certificate of deposit (CD) account with the bank?(Yes:1,No:0)'
    loan_count = loan_count + 1
elif loan_count == 11:
    res = lang + 'Do you use internet banking facilities?(Yes:1,No:0)'
    loan_count = loan_count + 1
elif loan_count == 12:
    res = lang + 'Do you use a credit card issued by State Bank of India?'
    loan_count = loan_count + 1
elif loan_count == 13:
    loan_count = loan_count + 1
```



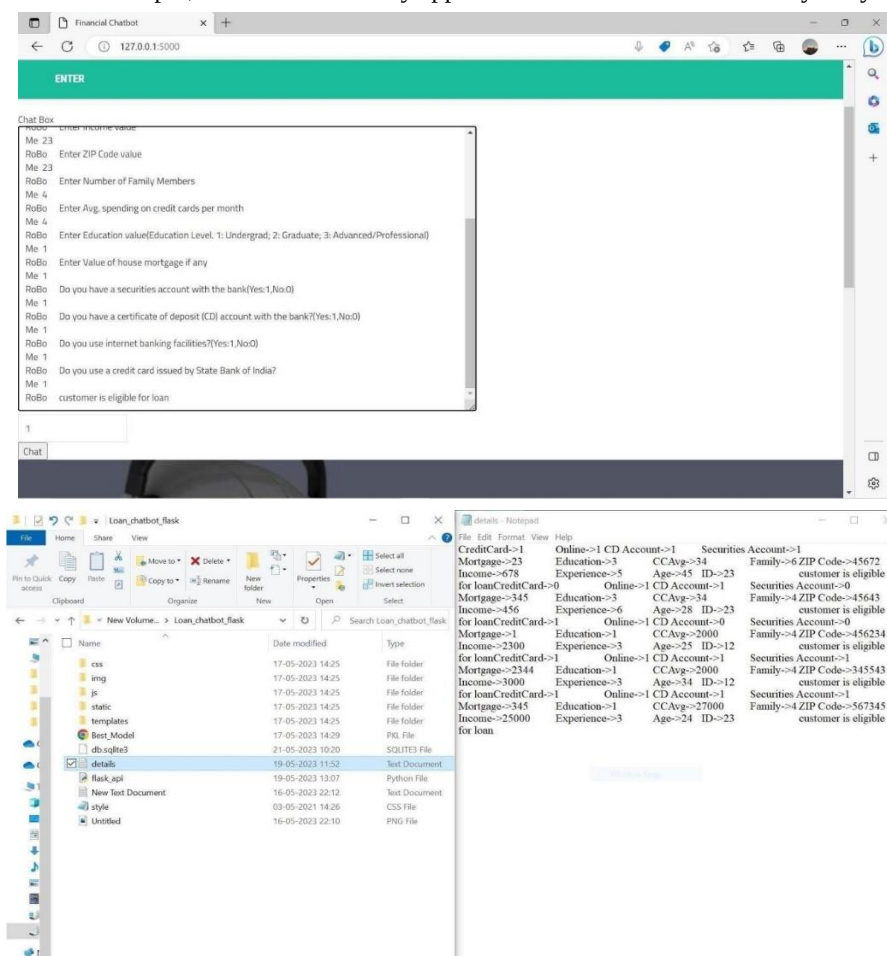
```
rint('final loan_qus', loan_qus)
print('final l=', l)
loan_pred = prediction(l)
print('loan_pred', loan_pred)
res = loan_pred
print(loan_qus)
ind = []
info = ""
for i, j in loan_qus.items():
    info = i + "->" + j + "\t" + info
    info = info + "\t" + loan_pred
f = open('details.txt', 'a')
f.write(info)
f.close()
if my_lang == 'en':
    res = 'e' + res
else:
    res = 'k' + res
print(res)
print('l=', l)
loan_count = 0
loan_flag = False
return res

# To switch on a particular URL, we use the add resource method and route it to the default slash.
api.add_resource(process, '/process')
if __name__ == '__main__':
    app.run() # run our Flask app
```

9. WORKING



Storing the date of customers input, it will automatically append the entered details after every entry from a customer.



10. CONCLUSION

Loan firms make loans after a thorough screening and validation process. They do not, however, know with absolute confidence if the applicant will be able to repay the loan without problem. The loan Prediction System will enable them to select the most deserving candidates quickly, effortlessly, and efficiently. It may provide the bank with advantages. In this assignment, we looked at how a Loan Approval Prediction System is created. Data collection, exploratory data analysis, data pre-processing, model creation, and model testing are the analytical processes used to build this system. We conducted a thorough evaluation of earlier research articles on this subject for our work.

The most widely used algorithms have been recognized as Random Forest Technique, Decision Tree, and Logistic Regression. We have analyzed earlier studies on this subject. The most well-known algorithms have all been carefully researched and analyzed, including Logistic Regression, Decision Trees, and Random Forest Technique.

11. REFERENCES

- [1] PhilHyo Jin Do ,Ho-Jin Choi, "Sentiment analysis of reallife situations using loca- tion, people and time as contextual features," International Conference on Big Data and Smart Computing (BIGCOMP), pp. 39–42. IEEE, 2015.
- [2] Bing Liu, "Sentiment Analysis and Opinion Mining," Morgan Claypool Publishers, May 2012.
- [3] Bing Liu, "Sentiment Analysis: Mining Opinions, Sentiments, and Emotions," Cambridge University Press, ISBN:978-1-107-01789-4.
- [4] Shiyang Liao, Junbo Wang, Ruiyun Yu, Koichi Sato, and Zixue Cheng, "CNN for situations understanding based on sentiment analysis of twitter data," Procedia computer science, 111:376–381,2017.CrossRef.
- [5] Aakanksha Saha, Tamara Denning, VivekSrikumar, Sneha Kumar Kasera. "Secrets inSource Code: Reducing False Positives usingMachine Learning," 2020 InternationalConference on Communication Systems Networks (COMSNETS), 2020.
- [6] Divya Madhu, Neeraj Jain C.J, Elmy Sebastain, Shinoy Shaji and Anandhu Ajayakumar, "A Novel Approach for Medical Assistance Using Trained Chatbot",International Conference on Inventive Communication and Computational Technologies(ICICCT 2017)