Model Development Phase Template

Date	3 August 2024
Team ID	740293
Project Title	Loan Sanction Amount Prediction Data With ML
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

This initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots. **Initial Model Training Code**

```
import time
import threading


#Speak function
def speak(audio):
    engine.say(audio)
    engine.runAndWait()


#Take Command function
def takeCommand():
    r=sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold=1
        audio=r.listen(source)

        try:
            print("Recognizing...")
            query=r.recognize_google(audio,language='en-in')
            print(f"User said: {query}")

        except Exception as e:
            print("Say that again please...")
            return "None"

        return query


def answer(g , m , d ,e ,s ,i ,c , l , lt ,cr ,p):
    data=pd.read_csv("preproceed.csv")
    data=data.drop("Loan_ID", axis=1)

    X = data.drop("Loan_Status", axis=1)
    y = data["Loan_Status"]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)


    # Fit the logistic regression model on the training data
    model = sm.Logit(y_train, X_train)
    result = model.fit()

    conf = result.conf_int()
    conf['Odds Ratio'] = np.exp(result.params)
    conf.columns = ['Lower CI', 'Upper CI', 'Odds Ratio']


    # Make predictions on the test data
    y_prob = result.predict([g , m , d ,e ,s ,i ,c , l , lt ,cr ,p])
    return y_prob

#Getting the available voices
engine=pyttsx3.init('sapi5')
voices=engine.getProperty('voices')
#print(voices)

#Setting the voice
engine.setProperty('voice',voices[0].id)


#Testing
if __name__=="__main__":
    counter1=1
    counter2=1
    counter3=1
    counter4=1
    counter5=1
    counter6=1
    counter7=1
```

```
while True:
    speak("Enter the Gender :")
    query=takeCommand().lower()
    if counter1>2:
        speak("Since your voice is not recognized for more than 2 times please enter the gender")
        print("Enter the gender :")
        user_input = [None]
        def get_input():
            user_input[0] = input()
        input_thread = threading.Thread(target=get_input)
        input_thread.start()
        input_thread.join(timeout=5)
        if user_input[0] is None:
            speak("Your time is out")
            print("Time out")
            time.sleep(3)
            speak("Restarting the session")
            time.sleep(2)
            continue
        else:
            print("User entered name:", user_input[0])
            g=user_input[0]
            break
    if query=="male"or query=="mail" or query=="m a l e" or query=="masculin":
        g=1
        break
    elif query=="female":
        g=0
        break
    elif query!="male" or query!="mail" or query!="female" or query!="masculin":
        counter1+=1
        speak("Could not understand please say again")
        counter1+=1
        continue

while True:
    speak("Whether you are married  :")
    query=takeCommand().lower()
    if counter2>2:
        speak("Since your voice is not recognized for more than 2 times please enter whether you are married")
        print("Enter whether you are married :")
        user_input = [None]
        def get_input():
            user_input[0] = input()
        input_thread = threading.Thread(target=get_input)
        input_thread.start()
        input_thread.join(timeout=5)
        if user_input[0] is None:
            speak("Your time is out")
            print("Time out")
            time.sleep(3)
            speak("Restarting the session")
            time.sleep(2)
            continue
        else:
            print("User entered name:", user_input[0])
            m=user_input[0]
            break
    if query=="married" or query=="not married":
        m=1
        break
    elif query=="not married":
        m=0
        break
    elif query!="married" or query!="not married":
        counter2+=1
        speak("Could not understand please say again")
        counter2=counter2+1
        continue
while True:
    speak("Whether you are dependent  :")
    query=takeCommand().lower()
    if counter3>2:
        speak("Since your voice is not recognized for more than 2 times please enter whether you are dependent")
        print("Enter whether you are dependent :")
        user_input = [None]
        def get_input():
```

```
                    user_input[0] = input()
                input_thread = threading.Thread(target=get_input)
                input_thread.start()
                input_thread.join(timeout=5)
                if user_input[0] is None:
                    speak("Your time is out")
                    print("Time out")
                    time.sleep(3)
                    speak("Restarting the session")
                    time.sleep(2)
                    continue
                else:
                    print("User entered name:", user_input[0])
                    d=user_input[0]
                    break
            if query=="dependent":
                d=1
                break
            elif query=="not dependent":
                d=0
                break
            elif query!="dependent" or query!="not dependent":
                counter3+=1
                speak("Could not understand please say again")
                counter3=counter3+1
                continue
        while True:
            speak("Whether you are educated  :")
            query=takeCommand().lower()
            if counter4>2:
                speak("Since your voice is not recognized for more than 2 times please enter whether you are educated")
                print("Enter whether you are educated :")
                user_input = [None]
                def get_input():
                    user_input[0] = input()
                input_thread = threading.Thread(target=get_input)
                input_thread.start()
                input_thread.join(timeout=5)
                if user_input[0] is None:
                    speak("Your time is out")
                    print("Time out")
                    time.sleep(3)
                    speak("Restarting the session")
                    time.sleep(2)
                    continue
                else:
                    print("User entered name:", user_input[0])
                    e=user_input[0]
                    break
            if query=="educated":
                e=1
                break
            elif query=="not educated":
                e=0
                break
            elif query!="educated" or query!="not educated":
                counter4+=1
                speak("Could not understand please say again")
                counter4=counter4+1
                continue
        while True:
            speak("Whether you are self-employed  :")
            query=takeCommand().lower()
            if counter5>2:
                speak("Since your voice is not recognized for more than 2 times please enter whether you are self-employed")
                print("Enter whether you are self-employed :")
                user_input = [None]
                def get_input():
                    user_input[0] = input()
                input_thread = threading.Thread(target=get_input)
                input_thread.start()
                input_thread.join(timeout=5)
                if user_input[0] is None:
                    speak("Your time is out")
                    print("Time out")
                    time.sleep(3)
                    speak("Restarting the session")
                    time.sleep(2)
```

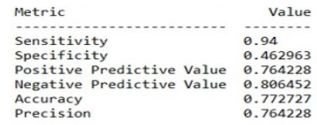
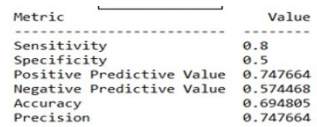
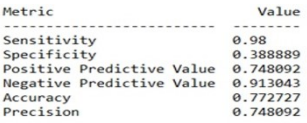
```
            continue
        else:
            print("User entered name:", user_input[0])
            s=user_input[0]
            break
    if query=="self employed":
        s=1
        break
    elif query=="not self employed":
        s=0
        break
    elif query!="self employed" or query!="not self employed":
        counter5+=1
        speak("Could not understand please say again")
        counter5=counter5+1
        continue

speak("Tell your income  :")
query=takeCommand().lower()
i=int(query)
speak("Tell your coapplicant income  :")
query=takeCommand().lower()
c=int(query)
speak("Tell your loan amount  :")
query=takeCommand().lower()
l=int(query)
speak("Tell your loan amount term  :")
query=takeCommand().lower()
lt=int(query)
while True:
    speak("Do you have credit history  :")
    query=takeCommand().lower()
    if counter6>2:
        speak("Since your voice is not recognized for more than 2 times please enter whether you have credit history")
        print("Enter whether you have credit history :")
        user_input = [None]
        def get_input():
            user_input[0] = input()
        input_thread = threading.Thread(target=get_input)
        input_thread.start()
        input_thread.join(timeout=5)
        if user_input[0] is None:
            speak("Your time is out")
            print("Time out")
            time.sleep(3)
            speak("Restarting the session")
            time.sleep(2)
            continue
        else:
            print("User entered name:", user_input[0])
            cr=user_input[0]
            break
    if query=="true":
        cr=1
        break
    elif query=="false":
        cr=0
        break
    elif query!="true" or query!="false":
        counter6+=1
        speak("Could not understand please say again")
        counter6=counter6+1
        continue
while True:
    speak("Enter number of property you have  :")
    query=takeCommand().lower()
    if counter7>2:
        speak("Since your voice is not recognized for more than 2 times please enter number of properties you have")
        print("Enter number of properties you have :")
        user_input = [None]
        def get_input():
            user_input[0] = input()
        input_thread = threading.Thread(target=get_input)
        input_thread.start()
        input_thread.join(timeout=5)
        if user_input[0] is None:
            speak("Your time is out")
```

```
            print("Time out")
            time.sleep(3)
            speak("Restarting the session")
            time.sleep(2)
            continue
        else:
            print("User entered name:", user_input[0])
            cr=user_input[0]
            break
    if query=="two" or "to":
        p=2
        break
    elif query=="one":
        p=1
        break
    elif query=="false":
        p=0
        break
    elif query!="two" or query!="to" or query!="one" or query!="false":
        counter7+=1
        speak("Could not understand please say again")
        counter7=counter7+1
        continue

print(g , m , d ,e ,s ,i ,c , l , lt ,cr ,p)
validate=answer(g , m , d ,e ,s ,i ,c , l , lt ,cr ,p)
if(validate >= 0.5).astype(int):
    speak(" congratulation You are Eligible for Loan")
    print("Eligible for Loan")
else:
    speak(" sorry You are Not Eligible for loan")
    print("Not Eligible for loan")
```

Model Valuation And Evalution Report

Model	Classification Report	F1 Score
Random Forest		77%
Decision Tree		69%
Logistic Regression		77%