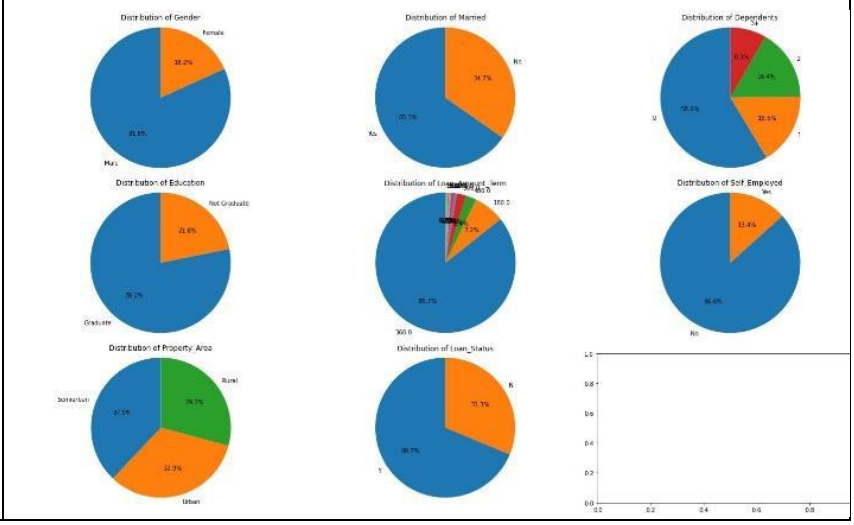
Data Collection and Preprocessing Phase

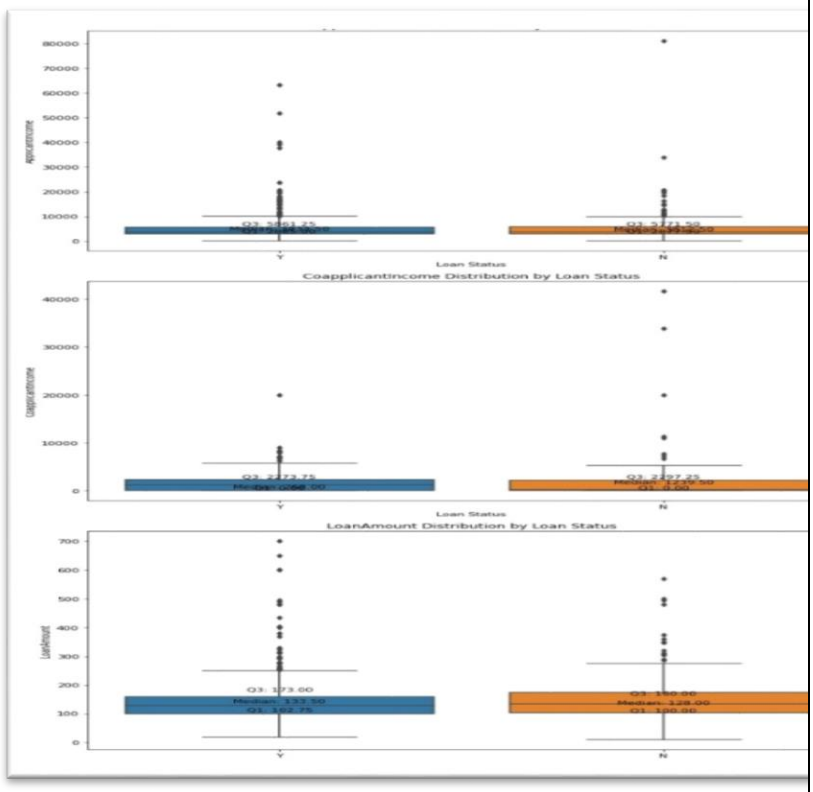
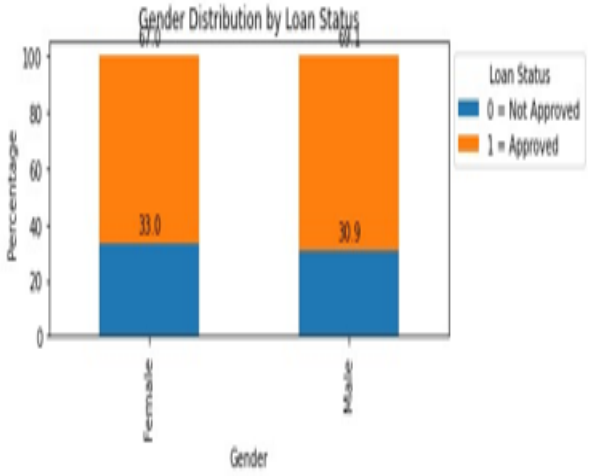
Date	3rd August 2024
Team ID	740293
Project Title	Loan Sanction Prediction Data with ML
Maximum Marks	6 Marks

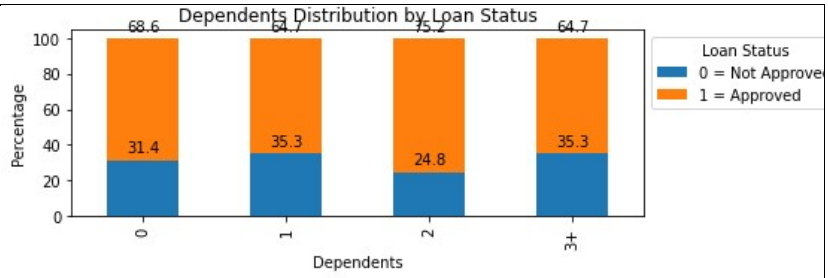
Data Exploration and Preprocessing Template

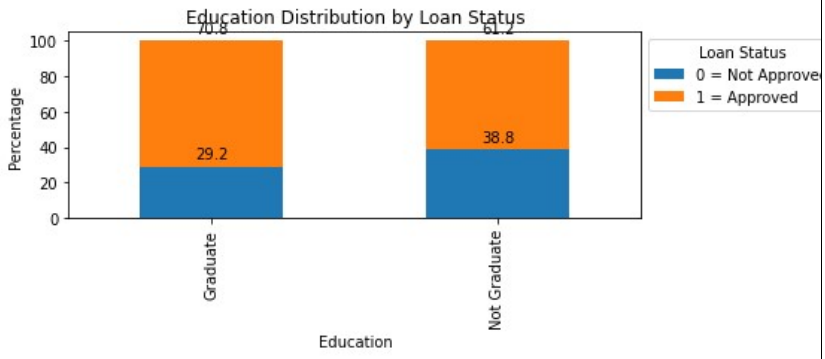
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

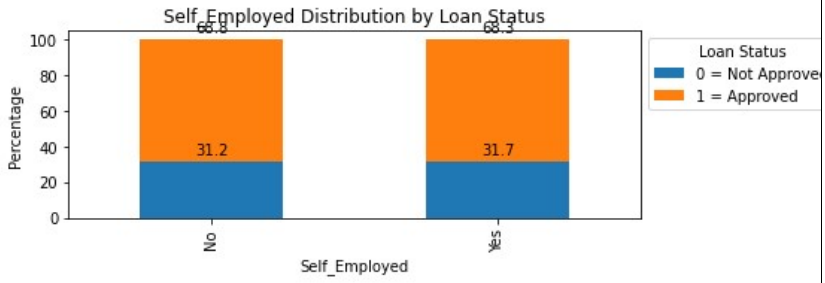
Section	Description
Data Overview	Descriptive Analysis: `data.describe()` (output below)
Univariate Analysis	

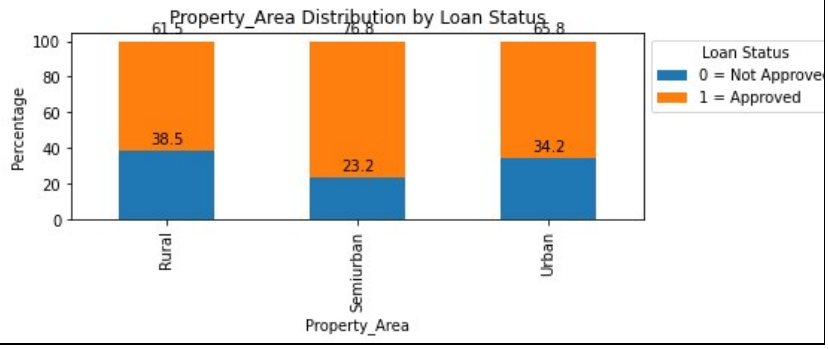
	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount
count	614.000000	614.000000	614.000000	614.000000	614.000000	614.000000	614.000000	614.000000
mean	0.817590	0.653094	0.744300	0.218241	0.133550	5403.459283	1621.245798	146.412162
std	0.386497	0.476373	1.009623	0.413389	0.340446	6109.041673	2926.248369	84.037468
min	0.000000	0.000000	0.000000	0.000000	0.000000	150.000000	0.000000	9.000000
25%	1.000000	0.000000	0.000000	0.000000	0.000000	2877.500000	0.000000	100.250000
50%	1.000000	1.000000	0.000000	0.000000	0.000000	3812.500000	1188.500000	129.000000
75%	1.000000	1.000000	1.000000	0.000000	0.000000	5795.000000	2297.250000	164.750000
max	1.000000	1.000000	3.000000	1.000000	1.000000	81000.000000	41667.000000	700.000000

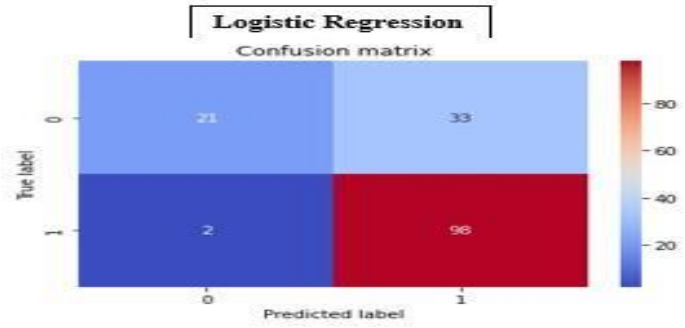
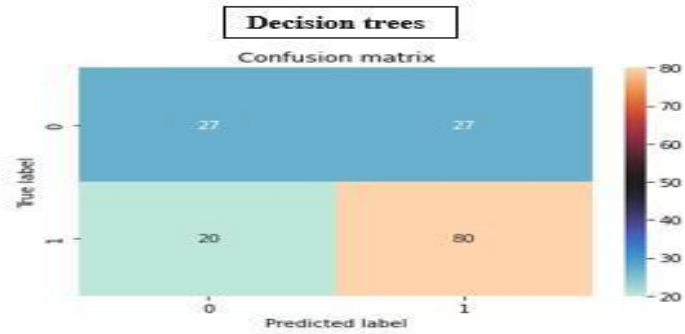
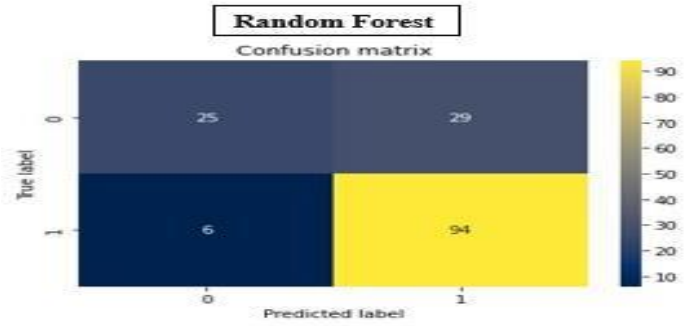
Bivariate Analysis	
Multivariate Analysis	


Dependents Distribution by Loan Status
Percentage
Dependents
0
1
2
3+
68.6
64.7
75.2
64.7
31.4
35.3
24.8
35.3
Loan Status
0 = Not Approve
1 = Approved


Education Distribution by Loan Status
Percentage
Education
Graduate
Not Graduate
70.8
61.2
29.2
38.8
Loan Status
0 = Not Approve
1 = Approved


Self_Employed Distribution by Loan Status
Percentage
Self_Employed
No
Yes
68.8
68.3
31.2
31.7
Loan Status
0 = Not Approve
1 = Approved


Property_Area Distribution by Loan Status
Percentage
Property_Area
Rural
Semiurban
Urban
61.5
76.8
65.8
38.5
23.2
34.2
Loan Status
0 = Not Approve
1 = Approved

Confusion matrix for models	  
Data Preprocessing Code Screenshots	
Loading Data	

Handling Missing Data

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
```



```
 0   Loan_ID            614 non-null    object
 1   Gender             614 non-null    float64
 2   Married            614 non-null    float64
 3   Dependents         614 non-null    int64
 4   Education          614 non-null    int64
 5   Self_Employed      614 non-null    float64
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         614 non-null    float64
 9   Loan_Amount_Term   614 non-null    float64
 10  Credit_History     614 non-null    float64
 11  Property_Area      614 non-null    int64
 12  Loan_Status        614 non-null    int64
dtypes: float64(7), int64(5), object(1)
memory usage: 62.5+ KB
```

```
data = data.drop('Loan_ID', axis=1)
```

```
data
```

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	1,0	0,0	0	0	0,0	5849	0,0	146,412162	360,0	1,0	2	1
1	1,0	1,0	1	0	0,0	4583	1508,0	128,000000	360,0	1,0	0	0
2	1,0	1,0	0	0	1,0	3000	0,0	66,000000	360,0	1,0	2	1
3	1,0	1,0	0	1	0,0	2583	2358,0	120,000000	360,0	1,0	2	1
4	1,0	0,0	0	0	0,0	6000	0,0	141,000000	360,0	1,0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...
609	0,0	0,0	0	0	0,0	2900	0,0	71,000000	360,0	1,0	0	1
610	1,0	1,0	3	0	0,0	4106	0,0	40,000000	180,0	1,0	0	1
611	1,0	1,0	1	0	0,0	8072	240,0	253,000000	360,0	1,0	2	1
612	1,0	1,0	2	0	0,0	7583	0,0	187,000000	360,0	1,0	2	1
613	0,0	0,0	0	0	1,0	4583	0,0	133,000000	360,0	0,0	1	0

614 rows × 12 columns

Next steps: Generate code with data View recommended plots New interactive sheet

```
Start coding or generate with AI.
```

```
data.isnull().sum()
```


	0
Gender	0
Married	0
Dependents	0
Education	0
Self_Employed	0
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	0
Loan_Amount_Term	0
Credit_History	0
Property_Area	0
Loan_Status	0

```
data
```

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	1,0	0,0	0	0	0,0	5849	0,0	146,412162	360,0	1,0	2	1
1	1,0	1,0	1	0	0,0	4583	1508,0	128,000000	360,0	1,0	0	0
2	1,0	1,0	0	0	1,0	3000	0,0	66,000000	360,0	1,0	2	1
3	1,0	1,0	0	1	0,0	2583	2358,0	120,000000	360,0	1,0	2	1
4	1,0	0,0	0	0	0,0	6000	0,0	141,000000	360,0	1,0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...
609	0,0	0,0	0	0	0,0	2900	0,0	71,000000	360,0	1,0	0	1
610	1,0	1,0	3	0	0,0	4106	0,0	40,000000	180,0	1,0	0	1
611	1,0	1,0	1	0	0,0	8072	240,0	253,000000	360,0	1,0	2	1
612	1,0	1,0	2	0	0,0	7583	0,0	187,000000	360,0	1,0	2	1
613	0,0	0,0	0	0	1,0	4583	0,0	133,000000	360,0	0,0	1	0

614 rows × 12 columns

Next steps: Generate code with data View recommended plots New interactive sheet

Testing

```
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
```

	
Data Transformation	-
Feature Engineering	-
Save Processed Data	-

```
        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")
```