Data Collection and Preprocessing Phase

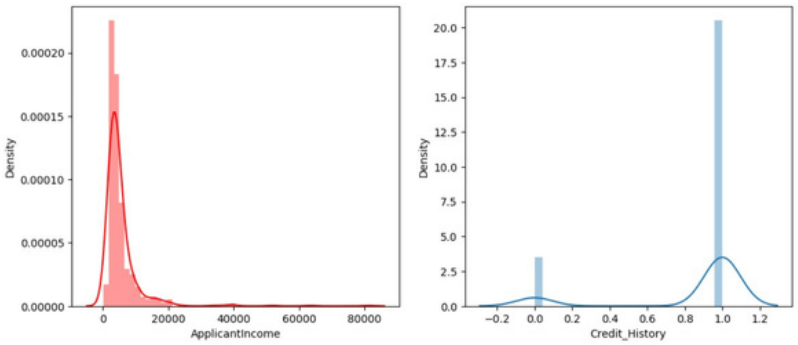
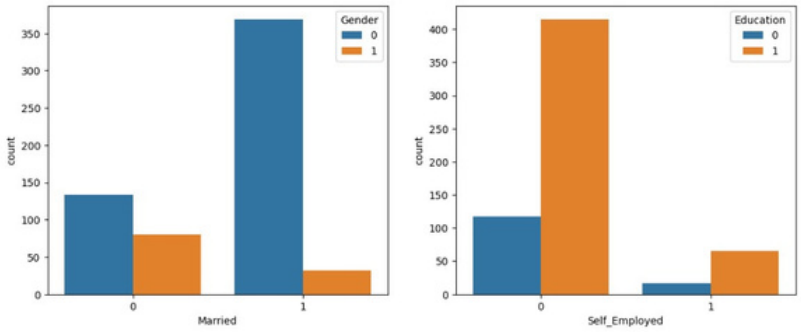
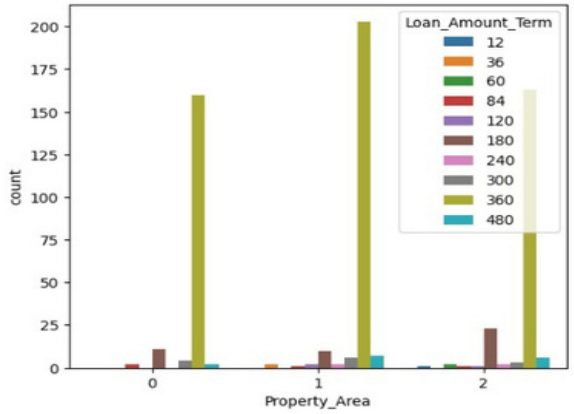
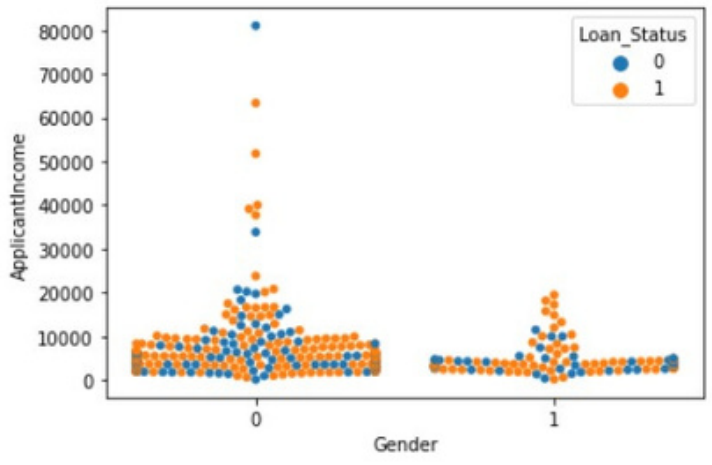
Date	27 January 2025
Team ID	SWUID20240011509
Project Title	Restaurant Recommendation System
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description
Data Overview	Dimension: 614 rows × 13 columns Descriptive statistics:
Univariate Analysis	

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

	
Bivariate Analysis	 
Multivariate Analysis	

<table>
<tr><td>Outliers and Anomalies</td><td>-</td></tr>
<tr><td colspan="2">Data Preprocessing Code Screenshots</td></tr>
<tr><td>Loading Data</td><td>

```
#importing the dataset which is in csv file
data = pd.read_csv('/content/Dataset/loan_prediction.csv')
data
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	LP001002	Male	No	0	Graduate	No	5849	0.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0

</td></tr>
<tr><td>Handling Missing Data</td><td>

```
data['Gender'] = data['Gender'].fillna(data['Gender'].mode()[0])
```

```
data['Married'] = data['Married'].fillna(data['Married'].mode()[0])
```

```
#replacing + with space for filling the nan values
data['Dependents']=data['Dependents'].str.replace('+','')
```

```
<ipython-input-71-6ac39c248773>:2: FutureWarning: The default value of regex will change from
  data['Dependents']=data['Dependents'].str.replace('+','')
```

```
data['Dependents'] = data['Dependents'].fillna(data['Dependents'].mode()[0])
```

```
data['Self_Employed'] = data['Self_Employed'].fillna(data['Self_Employed'].mode()[0])
```

```
data['LoanAmount'] = data['LoanAmount'].fillna(data['LoanAmount'].mode()[0])
```

```
data['Loan_Amount_Term'] = data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].mode()[0])
```

```
data['Credit_History'] = data['Credit_History'].fillna(data['Credit_History'].mode()[0])
```

</td></tr>
<tr><td>Data Transformation</td><td>

```
data['Gender']=data['Gender'].map({'Female':1,'Male':0})
data['Property_Area']=data['Property_Area'].map({'Urban':2,'Semiurban': 1,'Rural':0})
data['Married']=data['Married'].map({'Yes':1,'No':0})
data['Education']=data['Education'].map({'Graduate':1,'Not Graduate':0})
data['Loan_Status']=data['Loan_Status'].map({'Y':1,'N':0})
```

```
# perfroming feature Scaling op[eration using standard scaller on X part of the dataset because
# there different type of values in the columns
sc=StandardScaler()
x_bal=sc.fit_transform(x_bal)
```

</td></tr>
<tr><td>Feature Engineering</td><td>Attached the codes in final submission.</td></tr>
<tr><td></td><td></td></tr>
</table>