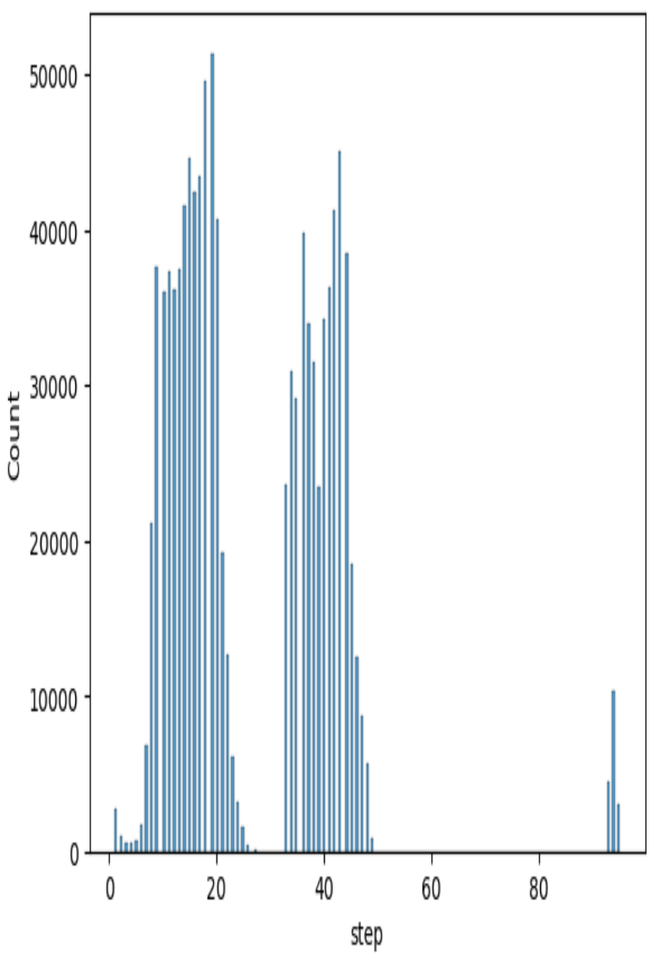
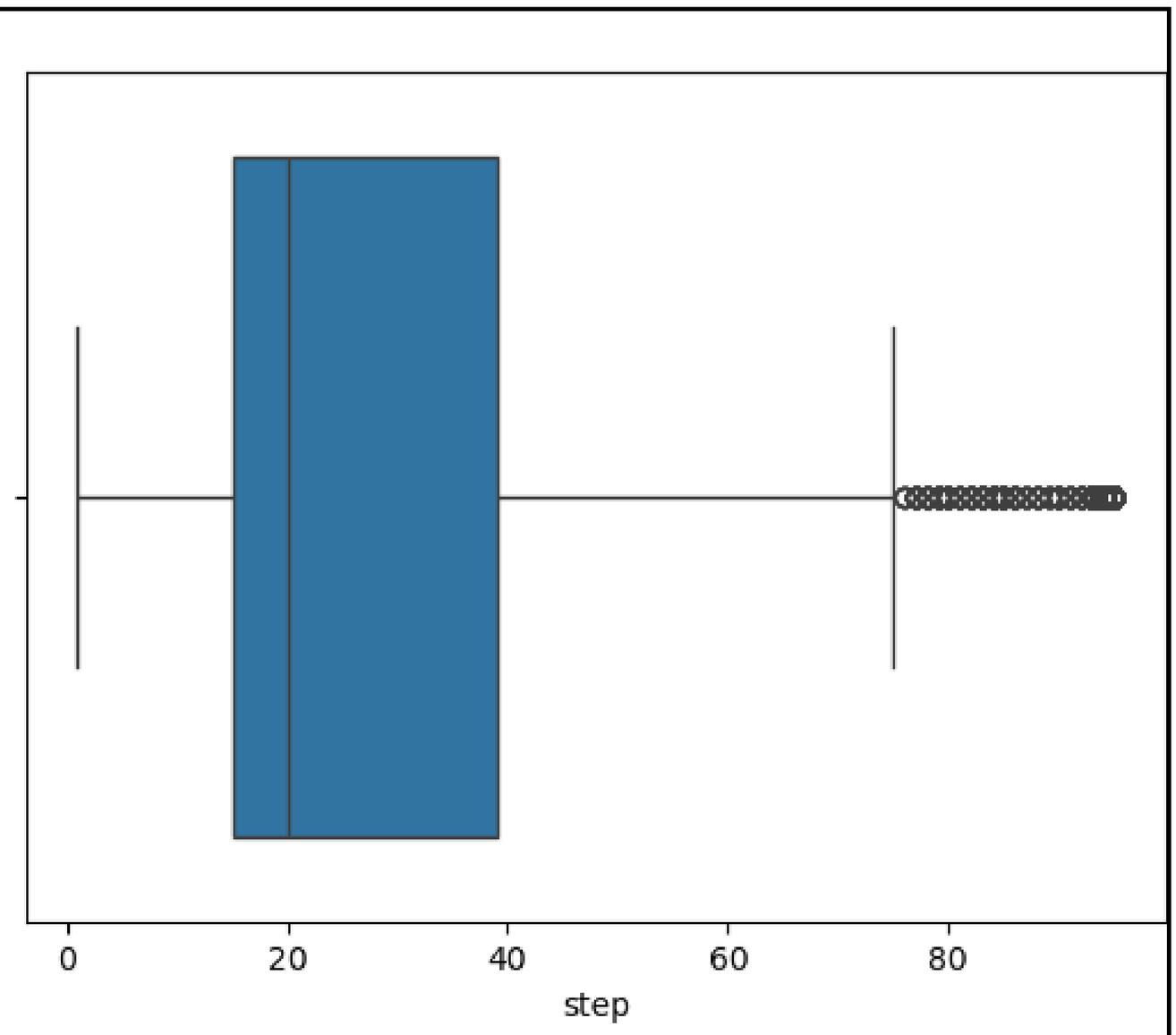
Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	xxxxxx
Project Title	xxxxxx
Maximum Marks	6 Marks

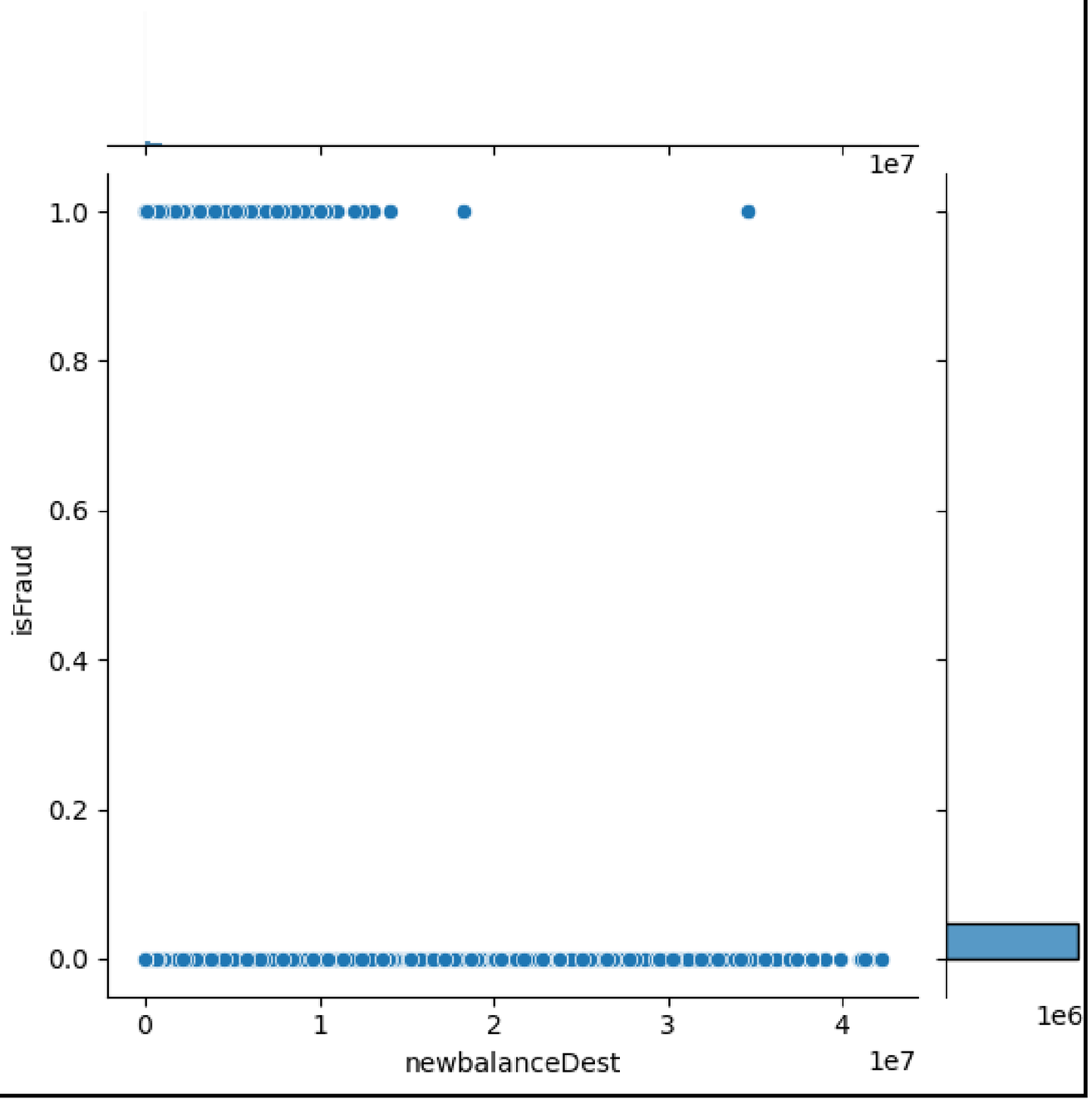
Data Exploration and Preprocessing Template

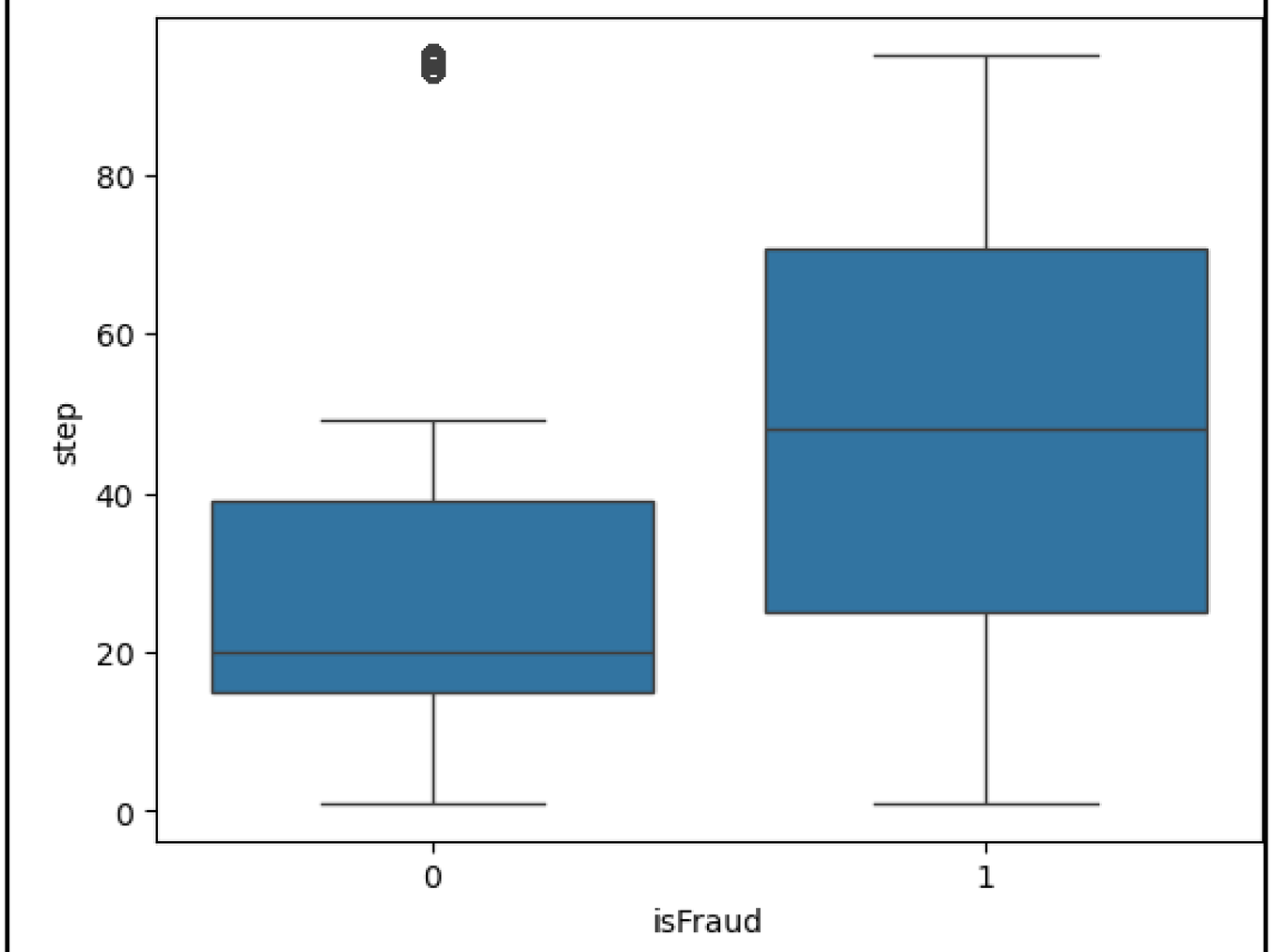
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	
Univariate Analysis	



Bivariate Analysis



Multivariate Analysis	
Outliers and Anomalies	-

Data Preprocessing Code Screenshots

Loading Data	

```
#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

Handling Missing Data	`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])`

Data Transformation	`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)`

Feature Engineering	Attached the codes in final submission
Save Processed Data	-