

## Data Collection and Preprocessing Phase

Date	20 June 2024
Team ID	740018
Project Title	Determine: Loan from KIVA crowdfunding data
Maximum Marks	6 Marks

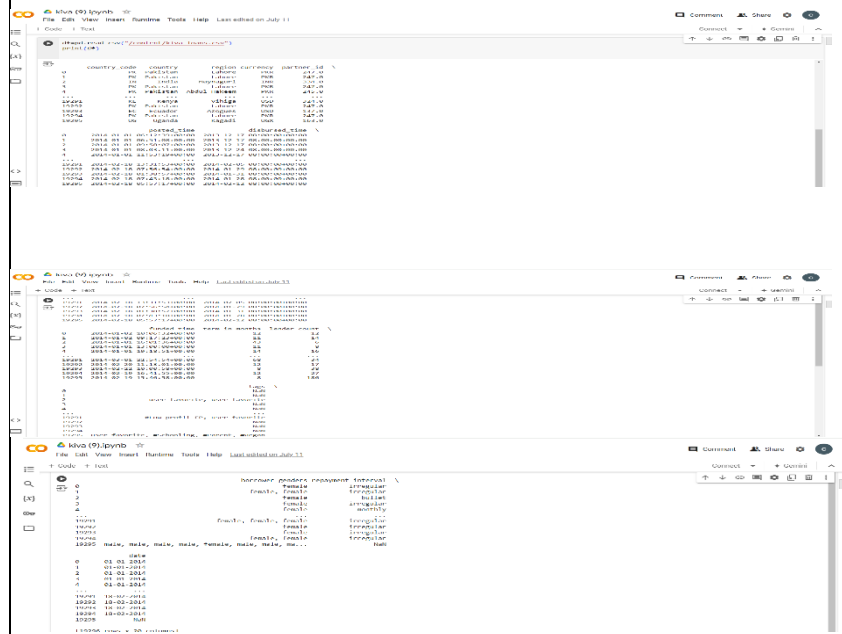
### Data Exploration and Preprocessing Template.

Data Exploration and Preprocessing Template for KIVA Crowdfunding: Load data, handle missing values, explore basic statistics, visualize distributions, encode categorical variables, normalize/scale features, identify outliers, and prepare for modeling.

Section	Description
Data Overview	Summary of the dataset, including number of rows and columns, data types of each column, and brief descriptions of each column.
Univariate Analysis	Distribution analysis of individual variables using histograms, bar charts, and descriptive statistics (mean, median, mode, standard deviation).
Bivariate Analysis	Examination of relationships between pairs of variables using scatter plots, correlation matrices, and pairwise plots to identify patterns and trends.
Multivariate Analysis	Investigation of interactions between multiple variables using heatmaps, PCA (Principal Component Analysis), and clustering to understand data structure.
Outliers and Anomalies	Identification and description of outliers and anomalies, summarized in a table with details on detection method, number of outliers, description, and potential impact.

## Data Preprocessing Code Screenshots

### Loading Data



```

# Read the CSV file into a DataFrame
df = pd.read_csv('data/borrower_data.csv')

# Display the first few rows of the DataFrame
df.head()

# Display the structure of the DataFrame
df.info()

# Display the dtypes of the DataFrame
df.dtypes

```

### Handling Missing Data



```

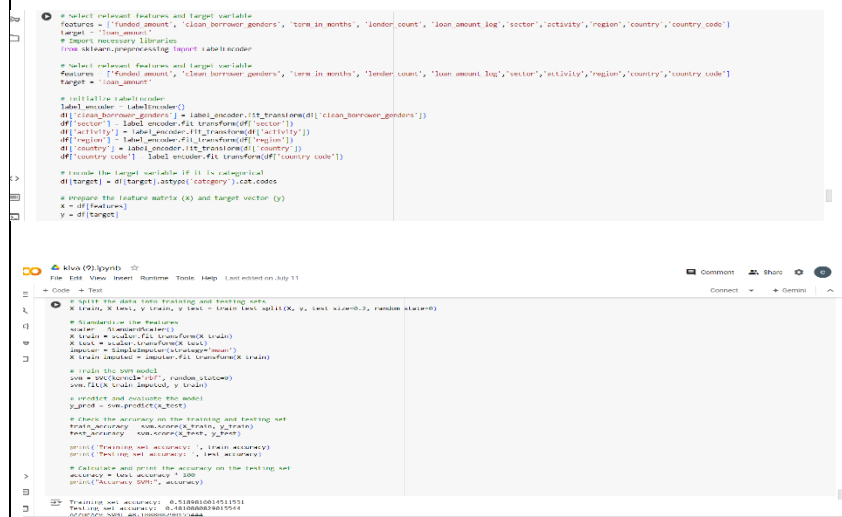
# Calculate the number of missing values in each column
missing_values = df.isnull().sum()

# Print the missing values
print(missing_values)

# Only 2.75814845734614 of the dataset is missing.

```

### Data Transformation



```

# Select relevant features and target variable
features = ['loan_amount', 'loan_borrower_genders', 'term_in_months', 'loan_count', 'loan_amount_log', 'sector', 'activity', 'region', 'country', 'country_code']
target = 'loan_amount'

# Import necessary libraries
from sklearn.preprocessing import LabelEncoder

# Create a LabelEncoder object
label_encoder = LabelEncoder()

# Fit the LabelEncoder on the target variable
label_encoder.fit(target)

# Transform the target variable into numerical values
df['loan_amount'] = label_encoder.transform(df['loan_amount'])

# Split the data into training and testing sets
X = df[features]
y = df['loan_amount']

# Create a Linear Regression model
model = LinearRegression()

# Fit the model on the training data
model.fit(X_train, y_train)

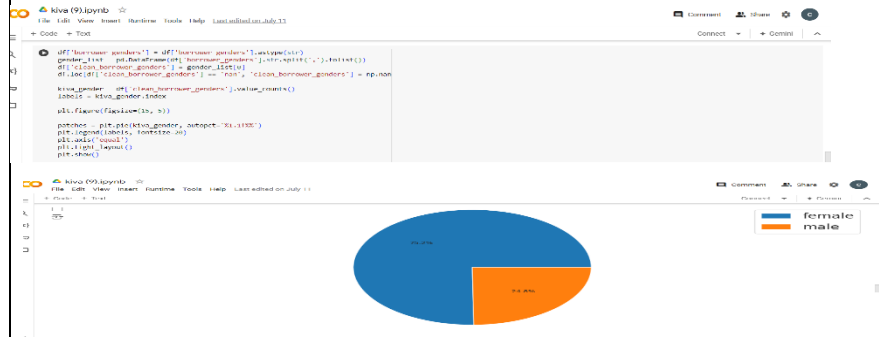
# Predict the loan amount for the testing data
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = r2_score(y_test, y_pred)

# Print the accuracy
print('Accuracy: ', accuracy)

```

## Feature Engineering



## Save Processed Data

