Data Collection and Preprocessing Phase

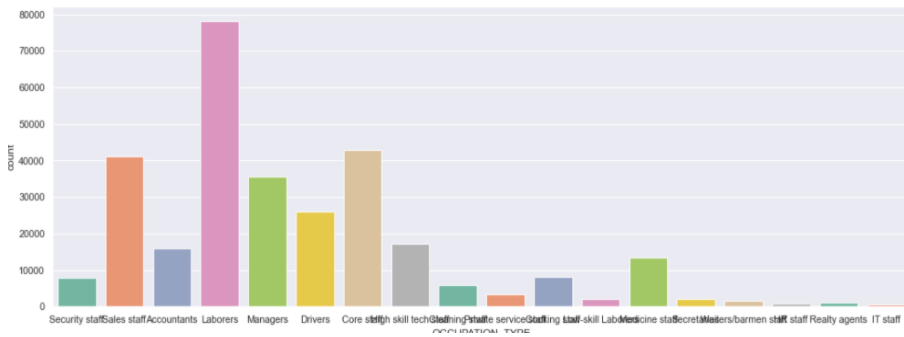
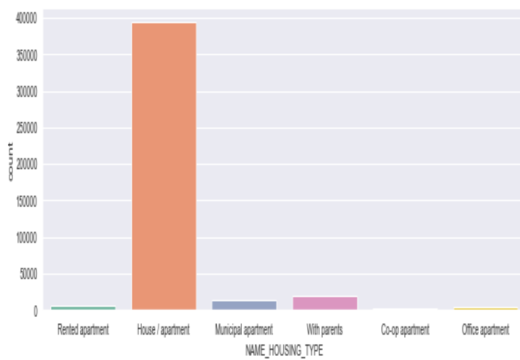
Date	10 July 2024
Team ID	739835
Project Title	Credit card approval prediction using ML
Maximum Marks	6 Marks

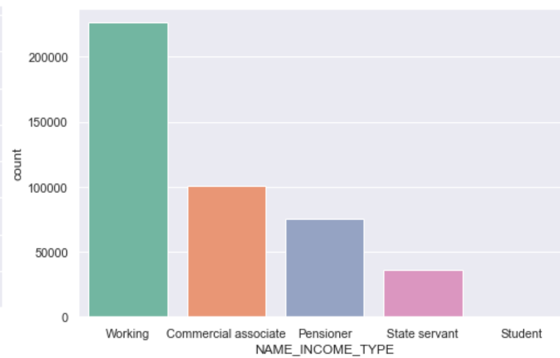
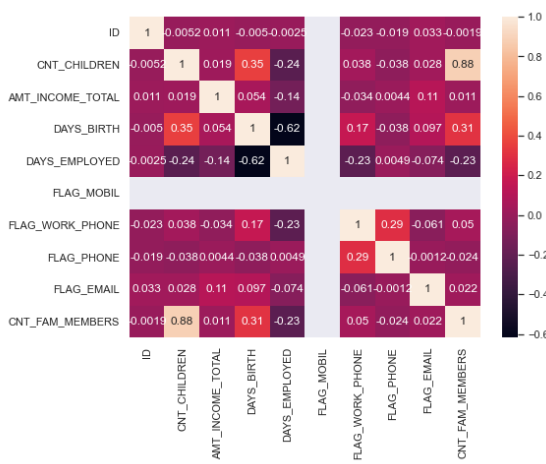
Data Exploration and Preprocessing Report

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 modelling, and forming a strong foundation for insights and predictions.

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

Feature	Count	Mean	Std	Min	25%	50%	75%	Max
ApplicantIncome	614	5403.46	6109.04	150	2877.50	3812.50	5795.00	81000
CoapplicantIncome	614	1621.25	2926.25	0	0.00	1186.50	2297.25	41667
LoanAmount	592	146.41	85.59	9	100.00	128.00	168.00	700
Loan_Amount_Term	600	342.00	65.12	12	360.00	360.00	360.00	480
Credit_History	564	0.842	0.365	0	1.00	1.00	1.00	1
Age	614	35.5	8.7	18	28.0	35.0	43.0	60
Dependents	614	0.5	0.7	0	0.0	0.0	1.0	3
Approval_Status	614	0.69	0.46	0	0.00	1.00	1.00	1

<table>
<tr><td></td><td>
</td></tr>
<tr><td>Bivariate Analysis</td><td>


</td></tr>
<tr><td>Multivariate Analysis</td><td>
</td></tr>
</table>

<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>
<table>
<tr><th></th><th>ID</th><th>CODE_GENDER</th><th>FLAG_OWN_CAR</th><th>FLAG_OWN_REALTY</th><th>CNT_CHILDREN</th><th>AMT_INCOME_TOTAL</th><th>NAME_INCOME_TYPE</th><th>NAME_EDUCATION_TYPE</th><th>NAME_FAMILY_STATUS</th><th>NAME_HOUSING_TYPE</th><th>DAYS_BIRTH</th></tr>
<tr><td>0</td><td>5008804</td><td>M</td><td>Y</td><td>Y</td><td>0</td><td>427500.0</td><td>Working</td><td>Higher education</td><td>Civil marriage</td><td>Rented apartment</td><td>-12005</td></tr>
<tr><td>1</td><td>5008805</td><td>M</td><td>Y</td><td>Y</td><td>0</td><td>427500.0</td><td>Working</td><td>Higher education</td><td>Civil marriage</td><td>Rented apartment</td><td>-12005</td></tr>
<tr><td>2</td><td>5008806</td><td>M</td><td>Y</td><td>Y</td><td>0</td><td>112500.0</td><td>Working</td><td>Secondary / secondary special</td><td>Married</td><td>House / apartment</td><td>-21474</td></tr>
<tr><td>3</td><td>5008808</td><td>F</td><td>N</td><td>Y</td><td>0</td><td>270000.0</td><td>Commercial associate</td><td>Secondary / secondary special</td><td>Single / not married</td><td>House / apartment</td><td>-19110</td></tr>
<tr><td>4</td><td>5008809</td><td>F</td><td>N</td><td>Y</td><td>0</td><td>270000.0</td><td>Commercial associate</td><td>Secondary / secondary special</td><td>Single / not married</td><td>House / apartment</td><td>-19110</td></tr>
</table>
</td></tr>
<tr><td>Handling Missing Data</td><td>

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

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

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

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

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

data['ApplicantIncome'] = data['ApplicantIncome'].fillna(data['ApplicantIncome'].mean())

data['LoanAmount'] = data['LoanAmount'].fillna(data['LoanAmount'].mean())

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['Married'] = data['Married'].map({'Yes': 1, 'No': 0})
data['Dependents'] = data['Dependents'].map({'0': 0, '1': 1, '2': 2, '3+': 3})
data['Education'] = data['Education'].map({'Graduate': 1, 'Not Graduate': 0})
data['Self_Employed'] = data['Self_Employed'].map({'Yes': 1, 'No': 0})
data['Property_Area'] = data['Property_Area'].map({'Urban': 2, 'Semiurban': 1, 'Rural': 0})
data['Loan_Status'] = data['Loan_Status'].map({'Y': 1, 'N': 0})

# Performing feature scaling using StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

</td></tr>
<tr><td>Feature Engineering</td><td>Attached the codes in final submission.</td></tr>
<tr><td>Save Processed Data</td><td>-</td></tr>
</table>