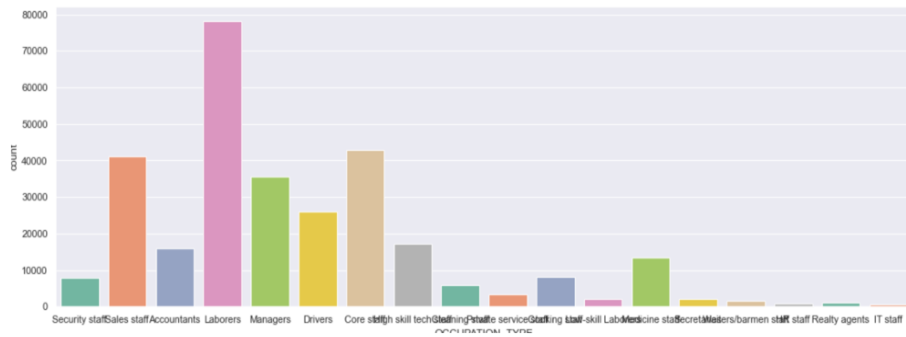
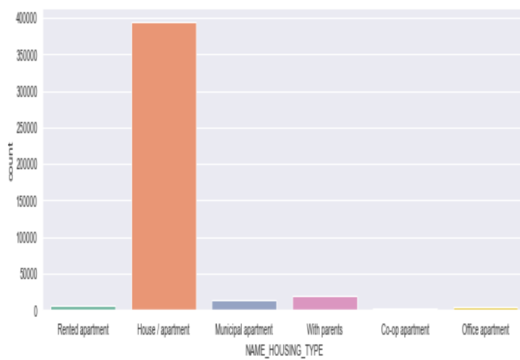
Data Collection and Preprocessing Phase

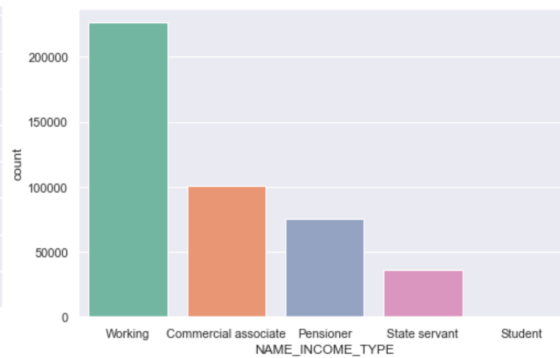
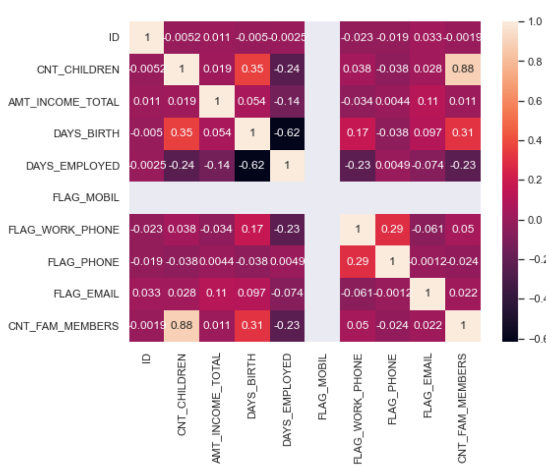
Date	10 July 2024
Team ID	739722
Project Title	Credit card approval prediction using ML
Maximum Marks	6 Marks

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.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>
<u>Dimension:</u>
614 rows × 13 columns
<u>Descriptive statistics:</u>
<table>
<tr><th>Feature</th><th>Count</th><th>Mean</th><th>Std</th><th>Min</th><th>25%</th><th>50%</th><th>75%</th><th>Max</th></tr>
<tr><td>ApplicantIncome</td><td>614</td><td>5403.46</td><td>6109.04</td><td>150</td><td>2877.50</td><td>3812.50</td><td>5795.00</td><td>81000</td></tr>
<tr><td>CoapplicantIncome</td><td>614</td><td>1621.25</td><td>2926.25</td><td>0</td><td>0.00</td><td>1186.50</td><td>2297.25</td><td>41667</td></tr>
<tr><td>LoanAmount</td><td>592</td><td>146.41</td><td>85.59</td><td>9</td><td>100.00</td><td>128.00</td><td>168.00</td><td>700</td></tr>
<tr><td>Loan_Amount_Term</td><td>600</td><td>342.00</td><td>65.12</td><td>12</td><td>360.00</td><td>360.00</td><td>360.00</td><td>480</td></tr>
<tr><td>Credit_History</td><td>564</td><td>0.842</td><td>0.365</td><td>0</td><td>1.00</td><td>1.00</td><td>1.00</td><td>1</td></tr>
<tr><td>Age</td><td>614</td><td>35.5</td><td>8.7</td><td>18</td><td>28.0</td><td>35.0</td><td>43.0</td><td>60</td></tr>
<tr><td>Dependents</td><td>614</td><td>0.5</td><td>0.7</td><td>0</td><td>0.0</td><td>0.0</td><td>1.0</td><td>3</td></tr>
<tr><td>Approval_Status</td><td>614</td><td>0.69</td><td>0.46</td><td>0</td><td>0.00</td><td>1.00</td><td>1.00</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Univariate Analysis</td><td></td></tr>
</table>

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


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

Outliers and Anomalies

-

Data Preprocessing Code Screenshots

Loading Data

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

Handling Missing Data

```
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])
```

Data Transformation

```
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)
```

Feature Engineering

Attached the codes in final submission.

Save Processed Data

-