LOYALIST COLLEGE IN TORONTO

AISC1006- Step Presentation

Loan Eligibility System

Group Name: D

Course Code: AISC1006

Instructor Name: Usman Ahmad

Problem Statement:
Loan Eligibility System

Overview: In the modern financial landscape, the influx of loan applications poses a significant challenge for institutions. manual evaluation of these applications is time-consuming and prone to errors. To streamline this process, the task is to develop an automated loan eligibility system. This system will leverage machine learning techniques to analyze applicant data and provide predictions on whether to approve or reject loan requests.

Objective: The primary goal is to design and implement a robust loan eligibility system that accurately assesses the creditworthiness of loan applicants. By doing so, financial institutions can expedite their decision-making process and mitigate risks associated with lending.

Key Features:

1. **Data Collection and Preparation:** Gather historical loan data and preprocess it to ensure quality and consistency.
2. **Feature Engineering:** Extract relevant features from the dataset and potentially create new ones to enhance predictive capabilities.
3. **Model Development:** Explore various machine learning algorithms to build predictive models capable of evaluating loan applications.
4. **Model Training and Validation:** Train the developed models using historical data and validate their performance to ensure reliability.
5. **Deployment:** Integrate the trained model into an operational system that can process loan applications in real-time.

Deliverables:

1. A fully functional loan eligibility system capable of handling real-world loan applications.
2. Documentation detailing the system architecture, algorithms used, and instructions for deployment.
3. A user-friendly interface for interacting with the system, ensuring accessibility for financial institution staff.
4. Comprehensive training materials and support to facilitate the adoption of the system within financial institutions.

Success Criteria: The success of the loan eligibility system will be evaluated based on its accuracy in predicting loan outcomes, efficiency in processing applications, and user satisfaction. Additionally, adherence to regulatory requirements and data privacy standards will be crucial.

Constraints: The development process must comply with legal regulations pertaining to lending practices and data privacy. Moreover, the system should be scalable to accommodate varying volumes of loan applications.

Conclusion: By developing an effective loan eligibility system, financial institutions can streamline their operations, mitigate risks, and make more informed lending decisions. This not only benefits the institutions themselves but also enhances the overall efficiency and integrity of the lending process.

INITIAL PROJECT MANAGEMENT REPORT:

1. Team Members:

1)Arish Panjwani - 500235989
2)Mansi Jayeshbhai Sutreja - 500238276
3)Obianuju Nonyerem Anuma - 500236077
4)Pavan Kumar Pilli - 500234994
5)Riya Kalpeshkumar Shah - 500236809
6)Siddhi Pravinbhai Patel - 500237311
7)Sri Datta Nadipolla - 500237146
8)Thejaswee Badepalle - 500236954
9)Yugahang Limbu - 500236054

2) DRAFTING A PLAN OF ACTION:

To achieve our goal and in order not to waste our time, we decided to choose the dataset as early as possible and started to work on it. As we have already picked our dataset so we worked on it and closely monitored to achieve our common goal. We had meetings to know each other and updated our progress. To make sure the report is submitted within the deadline our assigned team members will submit their project works on 24th may to the respective team member.

3) MEETING DOCUMENTATION :

Date	Meeting type (online or offline)	Content of discussion
11-05-24	Offline (recreation room)	We had discussion about skills and ideas from our team members. Choosing a dataset and assigning the roles to them.
14-05-24	Offline (student lounge)	We explored topic in detail about the problem statement and platform to work on it.
16-05-24	Online (zoom meeting)	Discussion on making a report and slides for preparation.
21-05-24	Offline (in class)	Started working together with coordination on the assigned works.
24-05-24	Offline (in class)	Implemented our inputs to produce a final report and ppt.

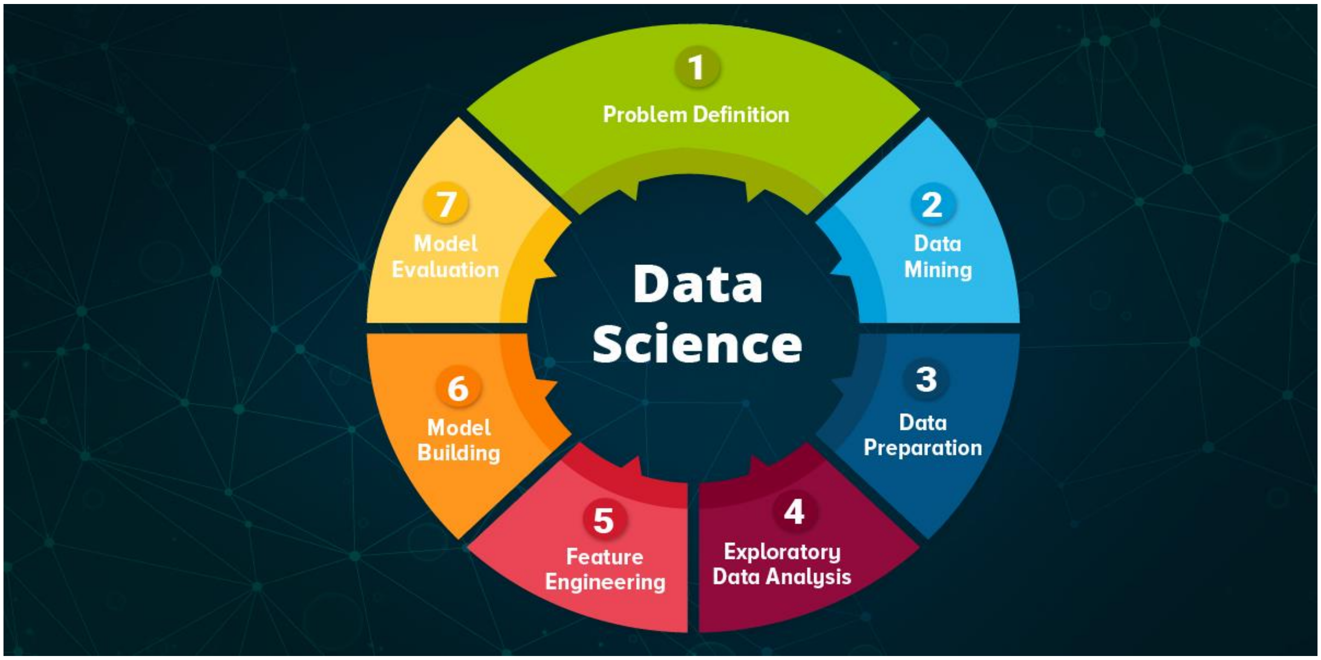
4) SOLUTION FLOW

1) Data Source: - Source and name of the dataset are “Kaggle”

https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset

2) Type of Analysis: - We have used exploratory data analysis and Prediction model to get the desired output.

Exploratory Data Analysis (EDA)



The approach involves using graphical representations and statistical summaries to analyse data. It is used to identify trends, patterns or to check assumptions.

Data scientists employ the approach of exploratory data analysis to research and analyse data collections (EDA). EDA's primary goal is to encourage data analysis before making any assumptions. You can get assistance with spotting glaring errors, understanding data trends, spotting outliers or strange occurrences, and discovering fascinating links among the variables.
To make sure the findings they create are reliable and relevant to any desired business objectives and goals, data scientists can employ exploratory analysis. EDA aids stakeholders by assuring them that they are posing the proper questions. Standard deviations, categorical variables, and confidence intervals are all topics that EDA may aid with. After EDA is finished and conclusions are made, its characteristics can be applied to more complex data analysis or modelling, including machine learning.
EDA comes in four main categories:

Non-graphical Univariate: When there is only one variable in the data being evaluated, this is the simplest type of data analysis. Since there is only one variable, no causes or correlations are discussed. Univariate analysis is mostly used to describe the data and identify any patterns.

Univariate Graphical: Non-graphical techniques don't give the whole story of the data. Therefore, graphical techniques are needed. Univariate graphics frequently used include:
The distribution's form and all of the data values are displayed in stem-and-leaf plots.
The frequency (count) or proportion (count/total count) of cases for a range of values are represented by each bar in a histogram, which is a bar plot.
Box plots, which graphically represent the minimum, first quartile, median, third quartile, and maximum five-number summary.

Multivariate nongraphical: Multivariate data is made up of multiple variables. Cross-tabulation or statistics are typically used in multivariate non-graphical EDA approaches to indicate the relationship between two or more variables of the data.

Multivariate graphical: Graphics are used with multivariate data to show connections between two or more kinds of data. The most common type of graph is a grouped bar plot or bar chart, where each group corresponds to a particular level of one of the variables and each bar inside a group to a particular level of the other variable.

The following are some of the most popular data science tools used to develop an EDA:

Python: An interpreted, object-oriented, dynamically semantic programming language.

DATA MANAGEMENT: -

Source Data Set:

The source dataset is the original set of data used as the foundation for a new dataset, or analysis. It is often transformed and cleaned, to create the final dataset used for user analysis. The quality and accuracy of the source dataset can greatly impact the accuracy and usefulness of the final dataset.

Quality Considerations:

Quality considerations for a dataset typically include:

1.**Accuracy**: The correctness and precision of the data.

2.**Consistency**: The data is consistent with itself

3.**Relevance**: The data is applicable to the problem or question at hand.

4.**Representativeness**: The data accurately reflects the population being studied.

Using Quality Consideration methods, we have checked the data set with the python libraries and it is as per the requirements of our problem statement.

DATA CLEANING:

Data cleaning is the process of preparing data for analysis by removing or modifying inaccuracies, outliers, and missing values.

1. Handling missing values: This can be done by either removing the rows with missing values or imputing missing values using techniques such as mean imputation or regression imputation.

2. Removing duplicates: Duplicate records can be removed using the drop_duplicates method in Pandas.

3. Removing outliers: Outliers can be removed by defining a range of values to consider normal and removing records outside this range.

4. Formatting data: This includes converting data types and transforming values to a consistent format.

Data Import and Exploration:

To work on the dataset, we had to first import the necessary libraries, view the dataset and explore the data.

In [1]:
```python
# Importing the library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
# Importing the dataset
df=pd.read_csv("loan_approval_dataset.csv",low_memory=False)
```

In [4]:
```python
# The dimensions of the dataset
df.shape
```

Out[4]:
```
(4269, 13)
```

In [3]:
```python
# First few rows of the dataset
df.head()
```

Out[3]:

	loan_id	no_of_dependents	education	self_employed	income_annum	loan_amount	loan_term
0	1	2	Graduate	No	9600000	29900000	12
1	2	0	Not Graduate	Yes	4100000	12200000	8
2	3	3	Graduate	No	9100000	29700000	20
3	4	3	Graduate	No	8200000	30700000	8
4	5	5	Not Graduate	Yes	9800000	24200000	20

In [5]:
```python
# Data types of the Series
df.dtypes
```

Out[5]:
```
loan_id                       int64
 no_of_dependents             int64
 education                   object
 self_employed               object
 income_annum                 int64
 loan_amount                  int64
 loan_term                    int64
 cibil_score                  int64
 residential_assets_value     int64
 commercial_assets_value      int64
 luxury_assets_value          int64
 bank_asset_value             int64
 loan_status                 object
dtype: object
```

In [6]:
```python
# Checking for duplicate records
df.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
# Checking for missing values
df.isnull().sum()
```

Out[7]:
```
loan_id                      0
 no_of_dependents            0
 education                   0
 self_employed               0
 income_annum                0
 loan_amount                 0
 loan_term                   0
 cibil_score                 0
 residential_assets_value    0
 commercial_assets_value     0
 luxury_assets_value         0
 bank_asset_value            0
 loan_status                 0
dtype: int64
```

In [9]:
```
# Statistical summary of numerical variables
df.describe()
```

Out[9]:

	loan_id	no_of_dependents	income_annum	loan_amount	loan_term	cibil_score
count	4269.000000	4269.000000	4.269000e+03	4.269000e+03	4269.000000	4269.000000
mean	2135.000000	2.498712	5.059124e+06	1.513345e+07	10.900445	599.936051
std	1232.498479	1.695910	2.806840e+06	9.043363e+06	5.709187	172.430401
min	1.000000	0.000000	2.000000e+05	3.000000e+05	2.000000	300.000000
25%	1068.000000	1.000000	2.700000e+06	7.700000e+06	6.000000	453.000000
50%	2135.000000	3.000000	5.100000e+06	1.450000e+07	10.000000	600.000000
75%	3202.000000	4.000000	7.500000e+06	2.150000e+07	16.000000	748.000000
max	4269.000000	5.000000	9.900000e+06	3.950000e+07	20.000000	900.000000

In [10]:
```
# Statistical summary of string variables
df.describe(include='object')
```

Out[10]:

	education	self_employed	loan_status
count	4269	4269	4269
unique	2	2	2
top	Graduate	Yes	Approved
freq	2144	2150	2656

To increase the accuracy of the data analysis, we must address the missing values in the data set. That is why we are using the **isnull**() method to find the gaps, where the information is missing.

The **describe**() methods returns description of the data and with the help of it we can figure out whether the data has any numeric variables or not.

Checking skewness is a crucial step in statistical modeling and data analysis. Skewness measures the asymmetry of the of the distribution of values in the dataset. Understanding skewness helps us to select appropriate statistical methods, improve model performance, and make valid conclusion. First we are selecting the column from the dataframe that have numeric datatypes and assigned to variable "numeric_cols", then we calculate skewness for each column using "df.[numeric_cols].skew()". A **skewness value of 0** indicates a **symmetric distribution** while a positive or negative value indicate **skewness (left or right respectively)**.

Likewise, checking for kurtosis helps us to understand the distribution of data and make decisions for further analysis. It helps to measure know if there are any outliers present in the dataset providing more insight into the distribution of data.

```
In [13]: # Checking for skewness
         numeric_cols = df.select_dtypes(include=[np.number]).columns
         df[numeric_cols].skew()

Out[13]: loan_id                      0.000000
          no_of_dependents            0.017971
          income_annum               -0.012814
          loan_amount                 0.308724
          loan_term                   0.036359
          cibil_score                -0.009039
          residential_assets_value    0.978451
          commercial_assets_value     0.957791
          luxury_assets_value         0.322208
          bank_asset_value            0.560725
         dtype: float64
```

click to expand output; double click to hide output

```
In [14]: # Checking for Kurtosis
         df[numeric_cols].kurt()

Out[14]: loan_id                     -1.200000
          no_of_dependents           -1.256992
          income_annum               -1.182729
          loan_amount                -0.743680
          loan_term                  -1.220853
          cibil_score                -1.185670
          residential_assets_value    0.184738
          commercial_assets_value     0.100813
          luxury_assets_value        -0.738056
          bank_asset_value           -0.397277
         dtype: float64
```

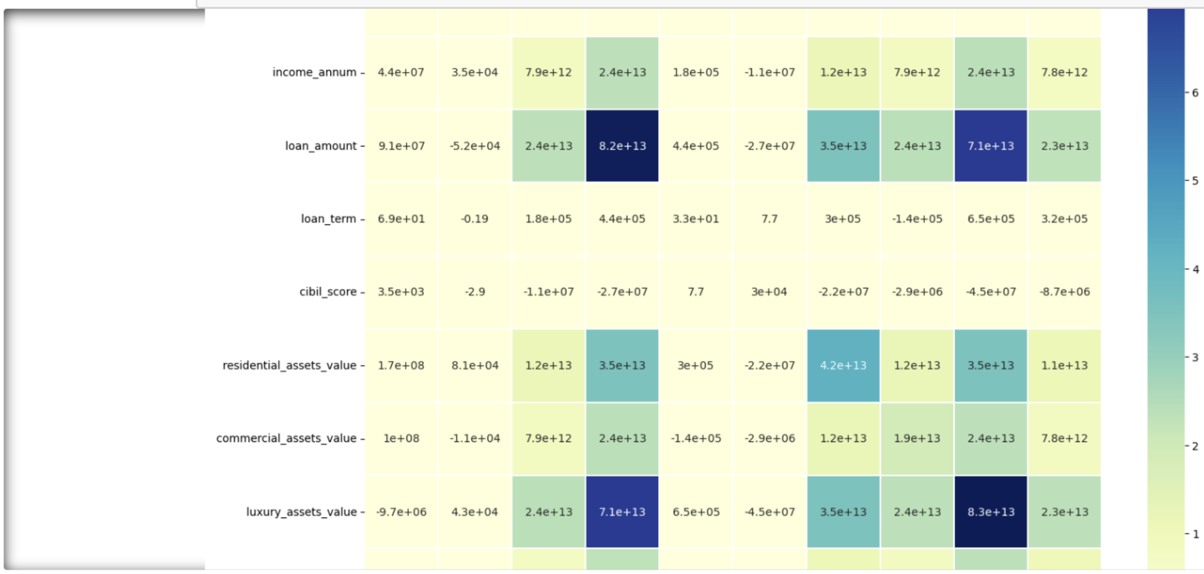
Then we calculate the covariance matrix for the selected subset of columns which shows covariance between two matrices (columns).

In [16]:
```python
# Covariance matrix, checking for relationship between variables
cov_matrix=df[numeric_cols].cov()
df[numeric_cols].cov()
```

Out[16]:

	loan_id	no_of_dependents	income_annum	loan_amount	loan_
loan_id	1.519052e+06	11.132146	4.356157e+07	9.106441e+07	69.01
no_of_dependents	1.113215e+01	2.876111	3.458884e+04	-5.162044e+04	-0.19
income_annum	4.356157e+07	34588.842910	7.878350e+12	2.354222e+13	184097.39
loan_amount	9.106441e+07	-51620.436384	2.354222e+13	8.178241e+13	435617.72
loan_term	6.901968e+01	-0.194716	1.840974e+05	4.356177e+05	32.59
cibil_score	3.469049e+03	-2.923817	-1.114830e+07	-2.656321e+07	7.68
esidential_assets_value	1.678147e+08	81349.435557	1.162533e+13	3.497100e+13	297624.65
ommercial_assets_value	1.005890e+08	-11398.230042	7.888277e+12	2.394115e+13	-137256.09
luxury_assets_value	-9.675375e+06	43485.164619	2.374225e+13	7.087780e+13	649175.26
bank_asset_value	4.312202e+07	61532.757354	7.764309e+12	2.316495e+13	318742.80

In [17]:
```python
# Covariance visualization
fid,ax=plt.subplots(figsize=(15,12))
sns.heatmap(cov_matrix,annot=True,linewidth=0.05,cmap='YlGnBu',fmt='.2')
plt.show()
```



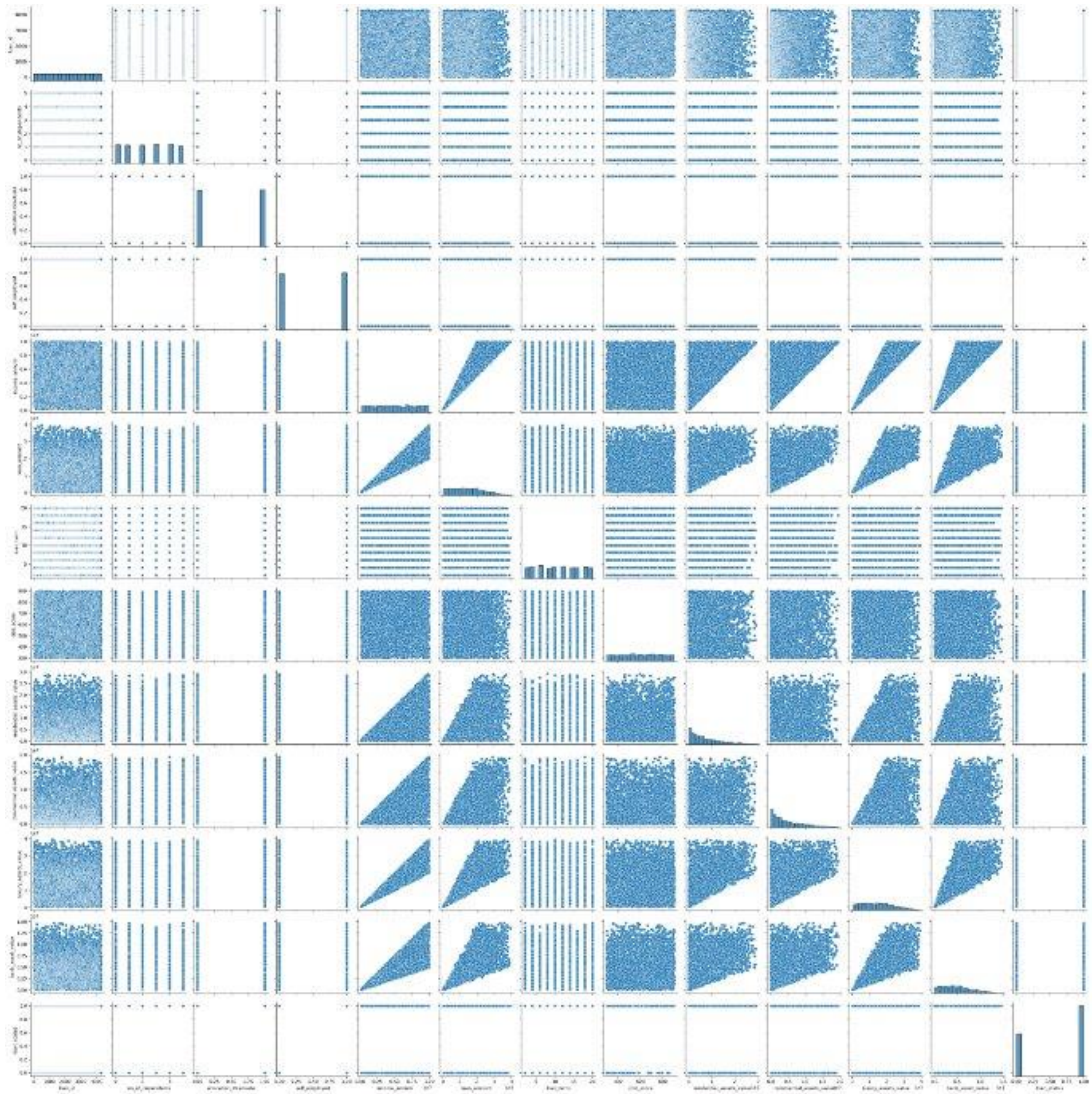
In [22]:
```python
#Checking for correlation of variables
corr_matrix=df[numeric_cols].corr()
df[numeric_cols].corr()
```

Out[22]:

	loan_id	no_of_dependents	income_annum	loan_amount	loan_term	cibil_
loan_id	1.000000	0.005326	0.012592	0.008170	0.009809	0.0
no_of_dependents	0.005326	1.000000	0.007266	-0.003366	-0.020111	-0.0
income_annum	0.012592	0.007266	1.000000	0.927470	0.011488	-0.0
loan_amount	0.008170	-0.003366	0.927470	1.000000	0.008437	-0.0
loan_term	0.009809	-0.020111	0.011488	0.008437	1.000000	0.0
cibil_score	0.016323	-0.009998	-0.023034	-0.017035	0.007810	1.0
esidential_assets_value	0.020936	0.007376	0.636841	0.594596	0.008016	-0.0
nmercial_assets_value	0.018595	-0.001531	0.640328	0.603188	-0.005478	-0.0
luxury_assets_value	-0.000862	0.002817	0.929145	0.860914	0.012490	-0.0
bank_asset_value	0.010765	0.011163	0.851093	0.788122	0.017177	-0.0

In [21]:
```python
#Corralation matrix
fid,ax=plt.subplots(figsize=(15,12))
sns.heatmap(corr_matrix,annot=True,linewidth=0.05,cmap='YlGnBu',fmt='.2'
plt.show()
```

Pair Plots



Pair Plots for Features having correlations greater than 0.3

Scatter plots can show whether there is a correlation between the two variables. If the points tend to go upwards from left to right, there is a positive correlation. If they tend to go downwards, there is a negative correlation.

The closer the points are to forming a straight line, the stronger the relationship. If the points are widely scattered, the relationship is weaker.

Scatter plots can easily highlight outliers, which are points that deviate significantly from the overall pattern. Identifying outliers is crucial for understanding anomalies, errors, or special cases in your data.

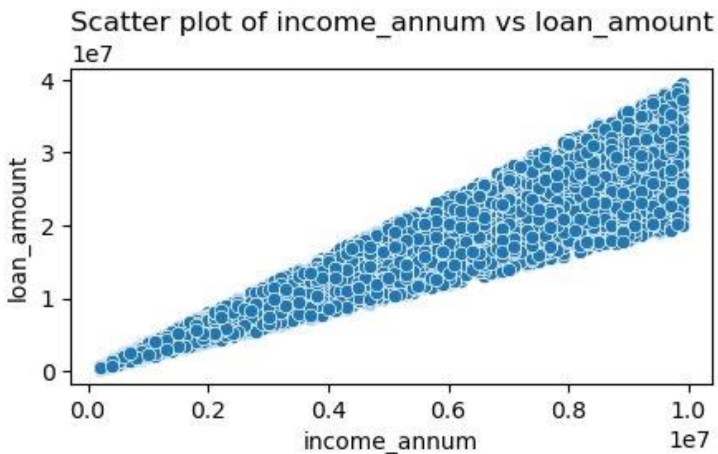
By examining how points are spread across the plot, you can get a sense of the distribution of each variable. For instance, you can see if the data is evenly spread out or if there are concentrations in certain areas.

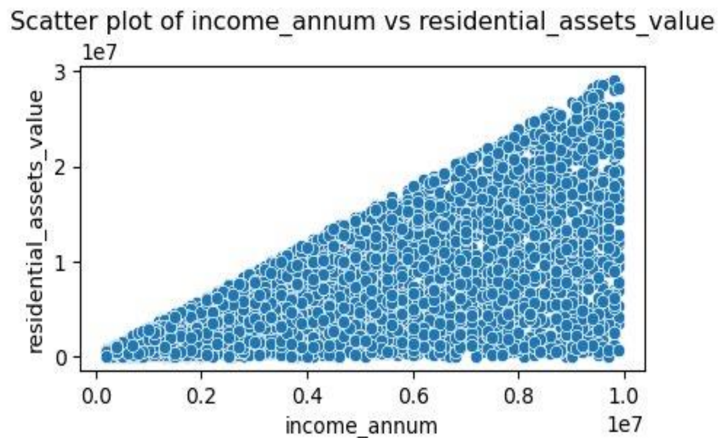
Pair Plots for Features having correlations greater than 0.3

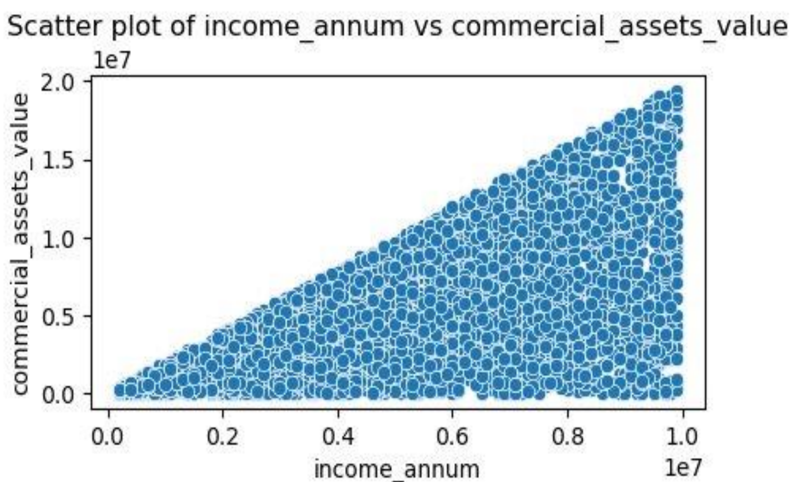
```
In [33]:  corr_matrix = df.corr()

          pairs = [(i, j) for i in range(corr_matrix.shape[0]) for j in range(i+1, corr_matrix.shape[0]) if abs(corr_matrix.iloc[i, j])

          for pair in pairs:
```

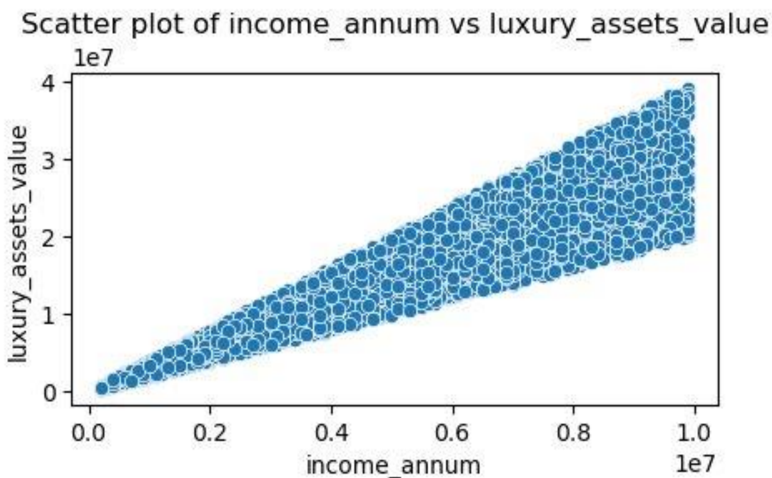
(U.S.)   Text Predictions: On   Editor Suggestions: Showing   10

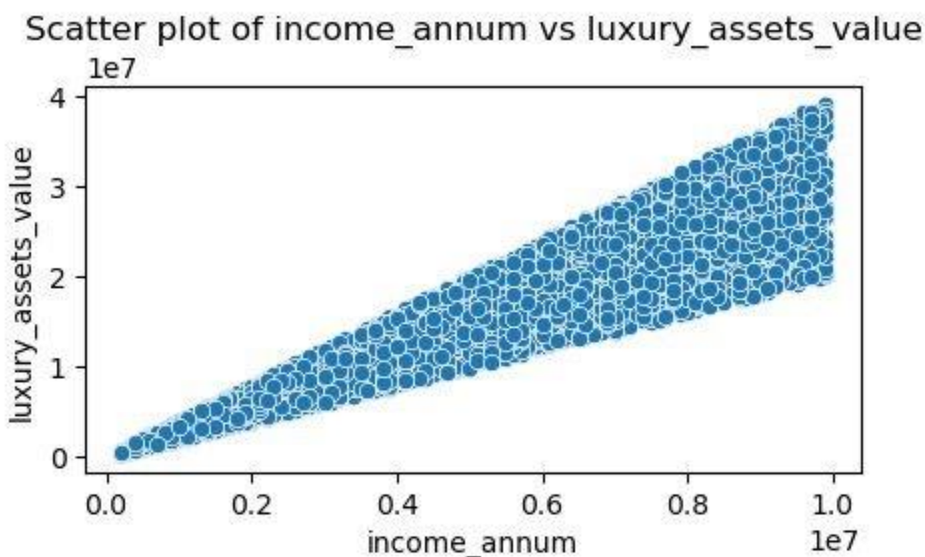
```
In [33]:  corr_matrix = df.corr()

          pairs = [(i, j) for i in range(corr_matrix.shape[0]) for j in range(i+1, corr_matrix.shape[0]) if abs(corr_matrix.iloc[i, j])

          for pair in pairs:
              plt.figure(figsize=(5, 2.5))
              sns.scatterplot(data=df, x=df.columns[pair[0]], y=df.columns[pair[1]])
              plt.title(f'Scatter plot of {df.columns[pair[0]]} vs {df.columns[pair[1]]}')
              plt.show()
```

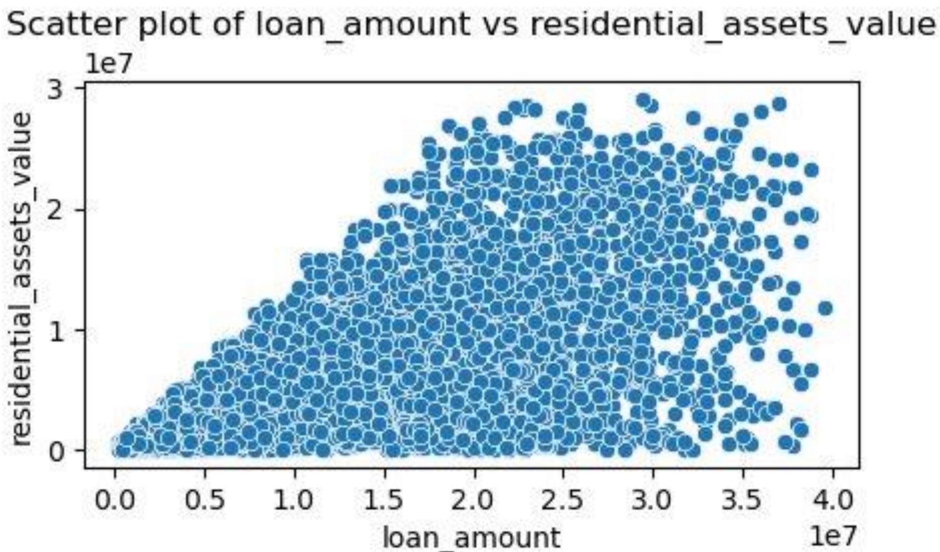

Scatter plot of income_annum vs loan_amount
1e7
loan_amount
income_annum
1e7
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0

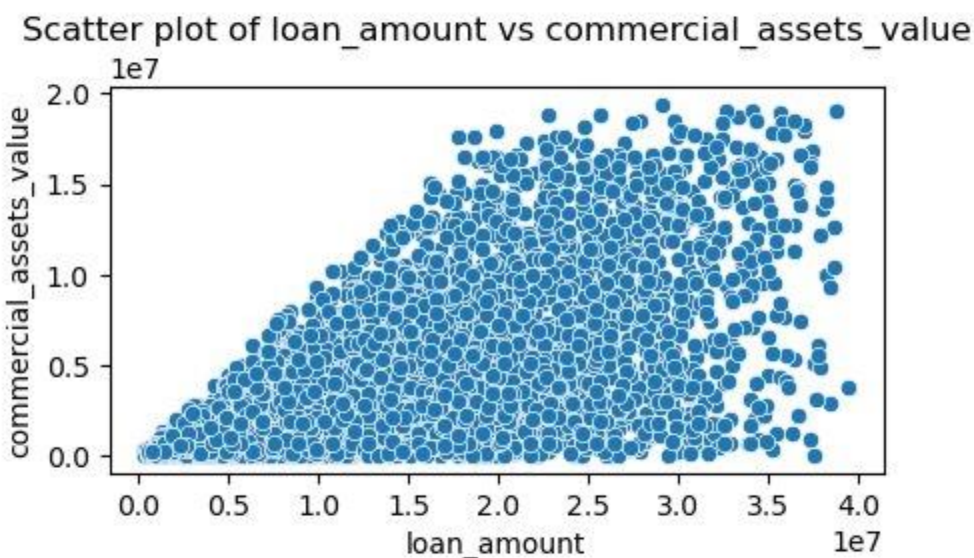

Scatter plot of income_annum vs residential_assets_value
1e7
residential_assets_value
income_annum
1e7
0
1
2
3
0.0
0.2
0.4
0.6
0.8
1.0

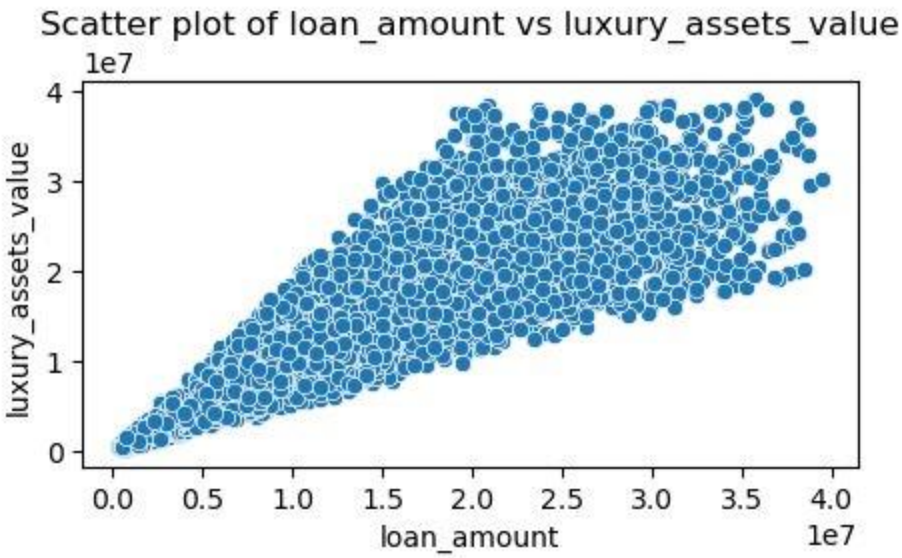

Scatter plot of income_annum vs commercial_assets_value
1e7
commercial_assets_value
income_annum
1e7
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
1.0

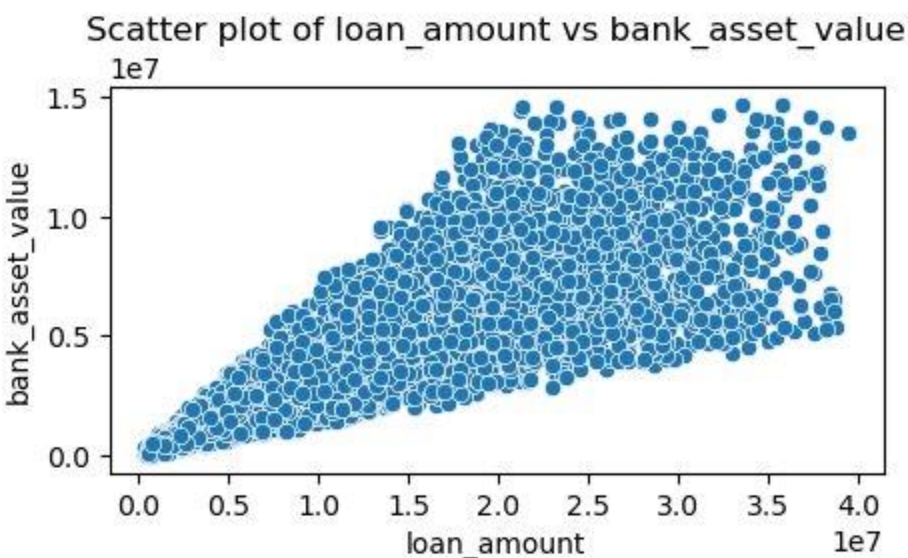

Scatter plot of income_annum vs luxury_assets_value
1e7
4
3
2
1
0
luxury_assets_value
0.0
0.2
0.4
0.6
0.8
1.0
income_annum
1e7


Scatter plot of income_annum vs luxury_assets_value
1e7
4
3
2
1
0
luxury_assets_value
0.0
0.2
0.4
0.6
0.8
1.0
income_annum
1e7

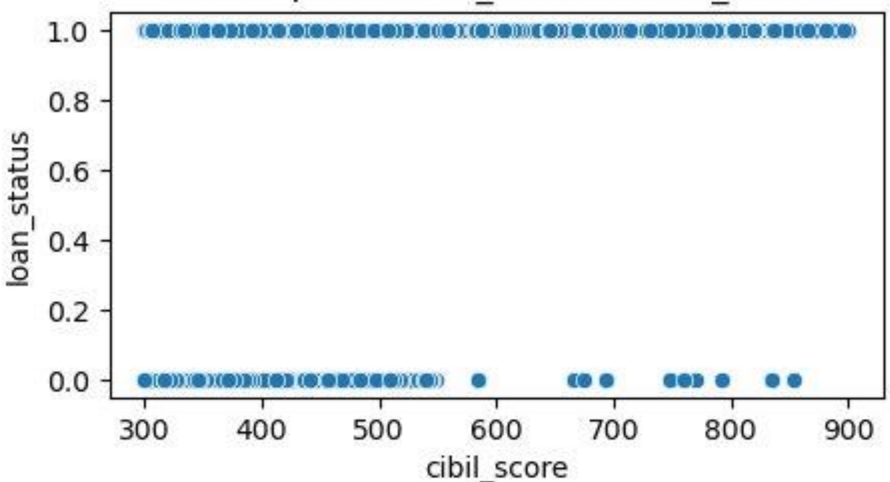

Scatter plot of loan_amount vs residential_assets_value
1e7
3
2
1
0
residential_assets_value
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
loan_amount
1e7


Scatter plot of loan_amount vs commercial_assets_value
1e7
2.0
1.5
1.0
0.5
0.0
commercial_assets_value
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
loan_amount
1e7

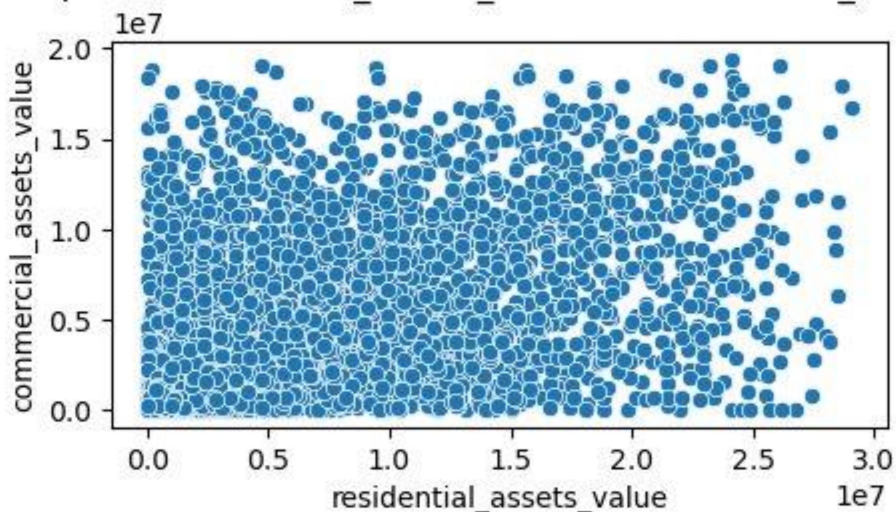

Scatter plot of loan_amount vs luxury_assets_value
1e7
4
3
2
1
0
luxury_assets_value
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
loan_amount
1e7


Scatter plot of loan_amount vs bank_asset_value
1e7
1.5
1.0
0.5
0.0
bank_asset_value
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
loan_amount
1e7

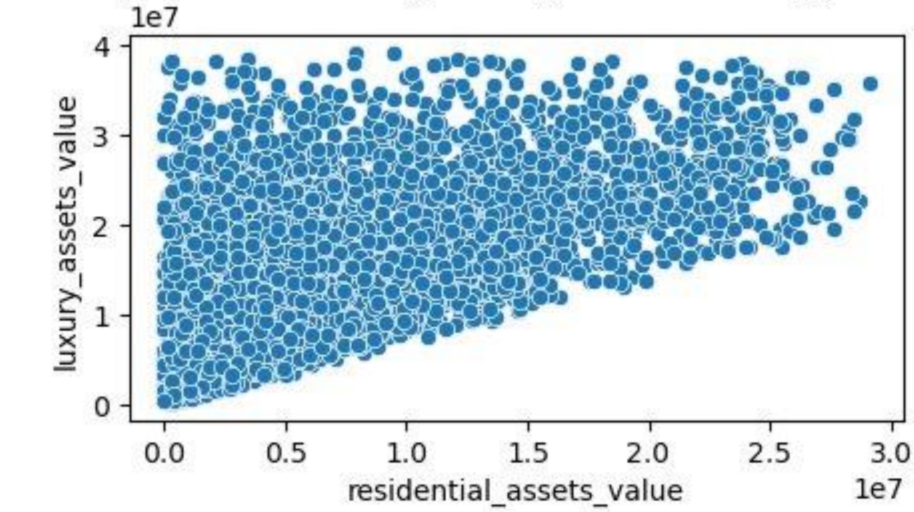
Scatter plot of cibil_score vs loan_status



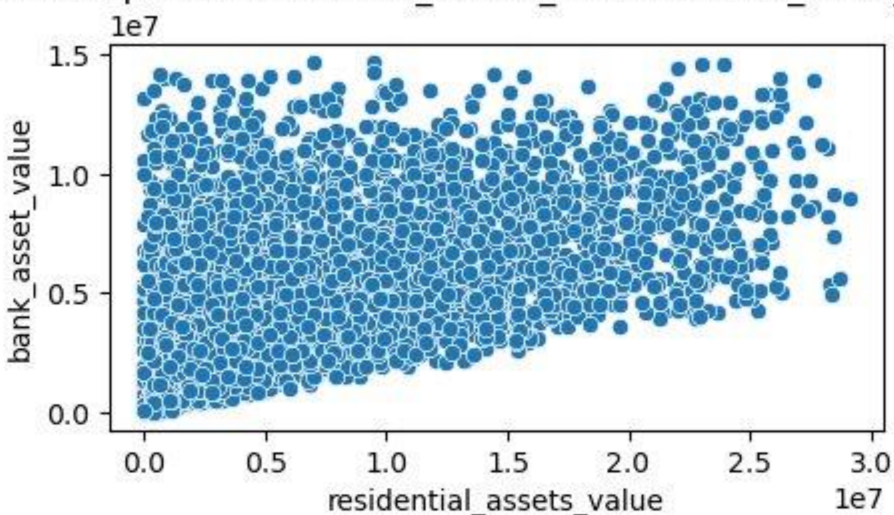
Scatter plot of residential_assets_value vs commercial_assets_value



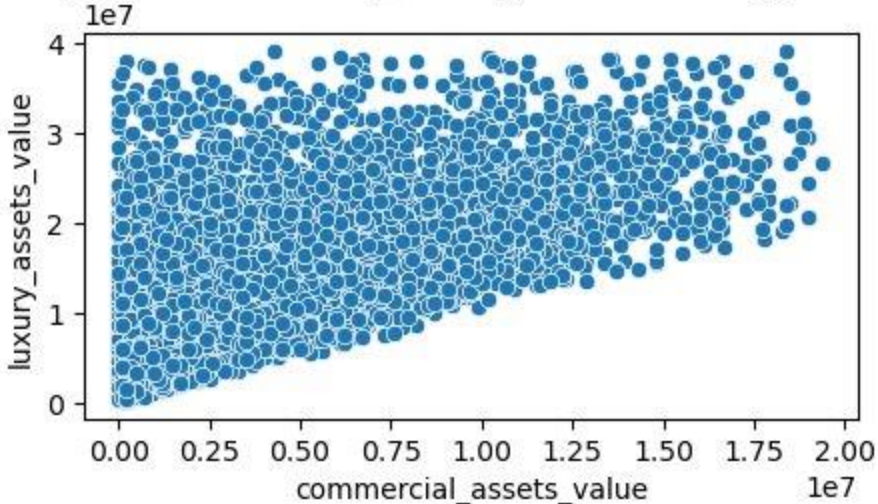
Scatter plot of residential_assets_value vs luxury_assets_value



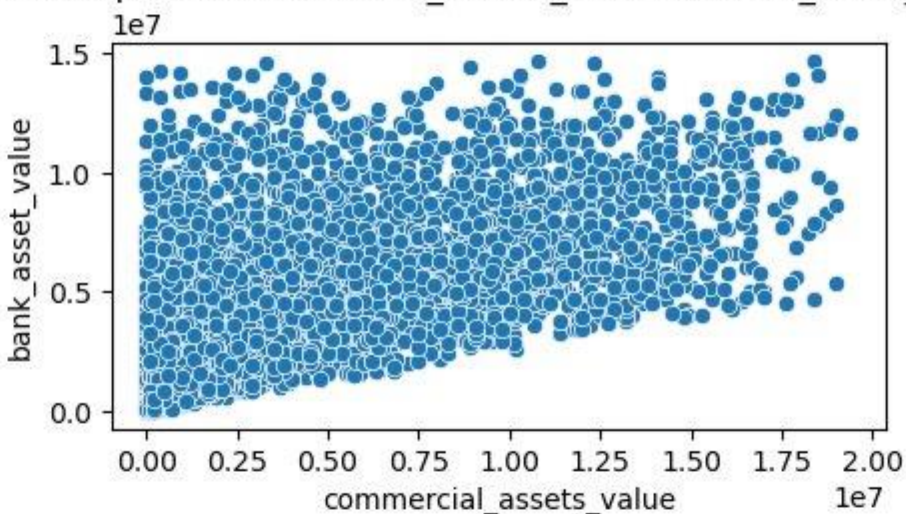
Scatter plot of residential_assets_value vs bank_asset_value

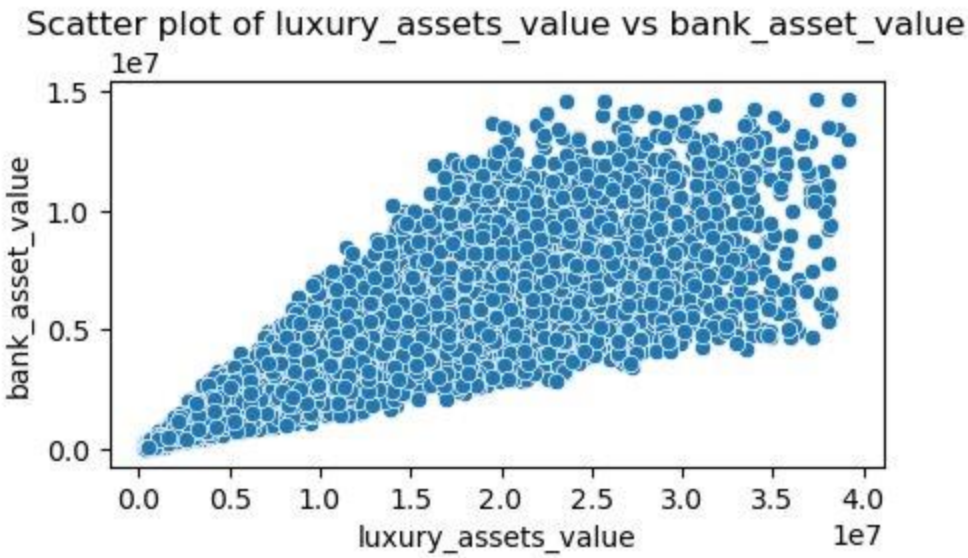


Scatter plot of commercial_assets_value vs luxury_assets_value



Scatter plot of commercial_assets_value vs bank_asset_value

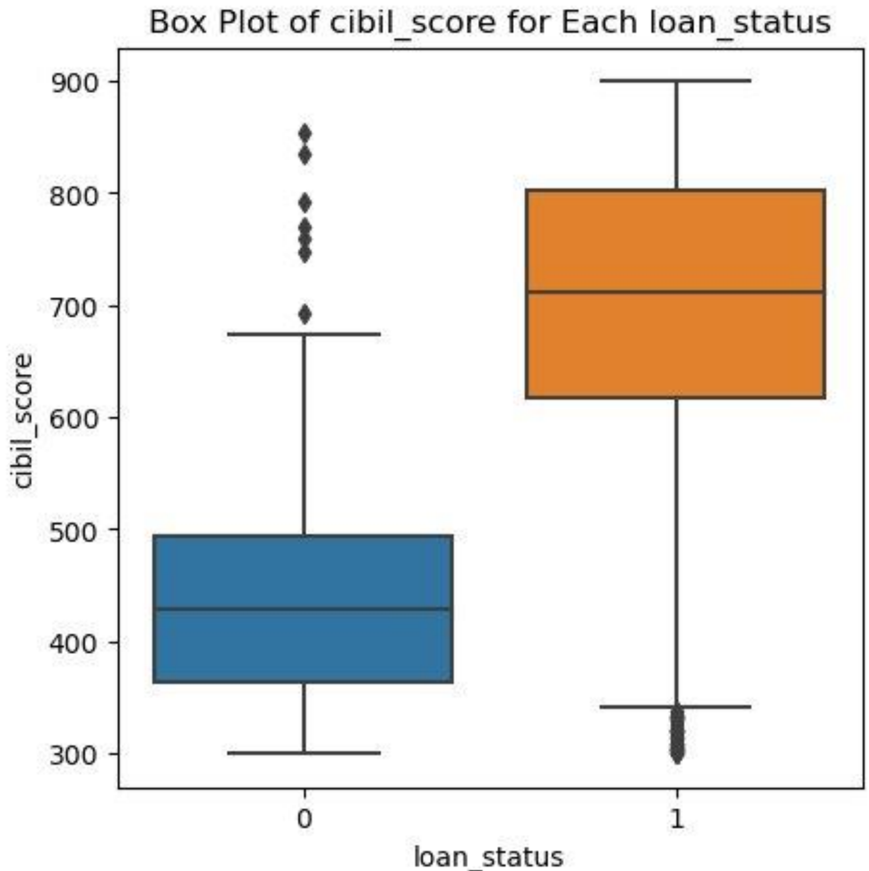




1. Income_annum and loan_amount is making positive correlation with the luxury assets value variable. The increase in annual income, loan amount clearly impacts the luxury assets value.
2. The plot between the loan amount and the annual income nearly maintains a positive correlation, which implies they are highly correlated.

Remaining features are maintaining no relation among each other.

Box Plot for Loan status vs Cibil score



REFERENCES:

*Loan-Approval-Prediction-Dataset (*2023) Kaggle. *https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset*

What is exploratory data analysis (EDA)? (2024) IBM. *https://www.ibm.com/topics/exploratory-data-analysis*